OBJECT LOCALIZATION HELPS ACTION RECOGNITION MODELS ADAPT TO NEW ENVIRONMENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Consider a real-world problem where we wish to adapt an existing action recognition (AR) model to a new environment. A common approach is to fine-tune a model on a set of labeled videos of actions performed in that environment. Such an approach is costly, since we need to record and annotate the videos, and fine-tune the model. At the same time, there has been recent interest in AR models that take an object-centric approach. In many cases these models are more structured, e.g., containing a module dedicated to object localization. Could we perform adaptation to a new environment via objects alone? We propose to re-use a previously trained AR model and *only adapt its object localization module*. Specifically, we train class-agnostic detectors that can adapt to each new environment. The idea of performing AR model adaptation via objects is novel and promising. While it requires some annotated images with the localized objects in the new environment, such supervision cost is lower than that of a conventional approach above. We conduct experiments on unseen kitchens in within- and across- dataset settings using Epic-Kitchen and EGTEA benchmarks, and show that AR models equipped with our object detectors can efficiently adapt to new environments. [1]

1 INTRODUCTION

Consider a realistic and practical problem: we wish to adapt an existing action recognition model to a new environment (e.g., adapt a smart home assistant to a particular home). A common solution is to treat this as domain adaptation problem, where we would typically require video recordings in the new environment, where video segments are annotated with action labels. We would then fine-tune the action recognition model on the annotated segments. Despite its simplicity, fine-tuning over target domain data is still considered as the upper bound for unsupervised domain adaptation performance (Munro & Damen, 2020; Kim et al., 2021). However, this approach is costly both in terms of data collection and labeling, and the required model fine-tuning (see Figure 1, A). For example, consider the effort of recording and labeling videos to demonstrate to a smart assistant how you make pancakes or cook an omelette in your specific kitchen environment.

At the same time, recently we have seen several action recognition works take an object-centric approach (Arnab et al., 2021b; Ben-Avraham et al., 2022; Herzig et al., 2022; Wang & Gupta, 2018). Their key intuition is that recognizing complex actions (such as human-object interactions) strongly benefits from explicitly modeling objects. Typically these models are more structured as opposed to "monolithic" architectures, e.g., they often contain a dedicated hand-object object localization module. However, prior works have not explored an adaptation scenario similar to ours.

All this brings us to the following question: could we perform action model adaptation to a new environment via objects alone? More concretely, could we re-use a previously trained action recognition model and *only adapt its object localization module*? Naturally, this will require some annotated images with the localized objects representative of the environment. But such supervision cost is much lower than that of a conventional approach described above. Compare the effort needed in a traditional approach (above) to capturing some images in your kitchen and drawing bounding boxes over the representative objects, like a frying pan or a spatula.

[1] Our code is included in the supplementary and we will release it upon acceptance.

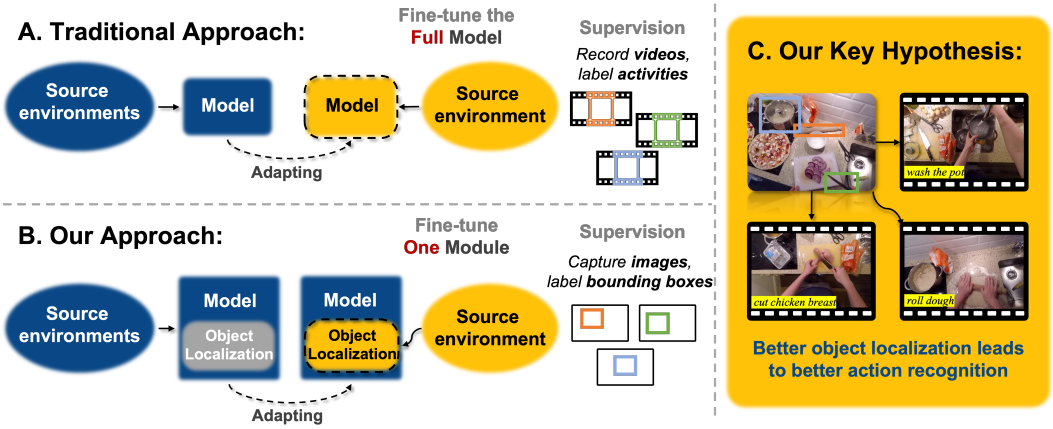


Figure 1: **The motivation behind our work:** (A) shows the traditional approach for adapting an action recognition model to a new environment, (B) shows our proposed approach for adaptation via objects, (C) illustrates our key hypothesis, i.e. that better object localization will also lead to better action recognition in the target environment.

Our proposed design to tackle this problem is as follows (Figure 1, B). Given an existing action recognition model trained on a source environment, we fine-tune its class-agnostic object detector to adapt to the target environment, leveraging a modest amount of labeled images. Note, that the action recognition part of the model is *not trained* on the target environment. Our main novelty and a key hypothesis is that better object localization in the new environment brings improvement in action recognition performance (Figure 1, C). Our particular design builds upon ORViT (Herzig et al., 2022), a recent video transformer framework that employs an object detector to provide input into the model. Note, that our presented ideas are general and could be applied to other object-centric video models.[2]

We conduct experiments in the following two challenging scenarios. 1. *Adaptation to an unseen environment within a dataset*. Here, we split the popular Epic-Kitchen dataset (Damen et al., 2018; 2020) into seen and unseen kitchens, and only evaluate on the unseen ones. This setting is somewhat "optimistic", since all the recordings in this dataset are made and annotated using the same hardware and annotation protocols. But unlike the standard split, we evaluate model performance on kitchens never seen in training. 2. *Adaptation to an unseen environment across datasets*. Here we test a more challenging scenario of adapting not only to an unseen kitchen but also to a different dataset. Specifically, we use Epic-Kitchen as source data and EGTEA Gaze+ videos (Li et al., 2018b) as targets. While these two datasets both capture egocentric cooking actions, they are distinct in terms of used hardware, recording setup, and annotation protocols.

Our results show that: (1) Our novel approach for action recognition adaptation supervised by the labeled images as opposed to labeled video segments, achieves over 8% relative boost in action recognition on a new environment within the same dataset. (2) When faced with the more challenging scenario of adaptation across datasets, it leads to an over 14% relative improvement in action recognition, showing the generalization of our method. (3) Our approach achieves cheaper (in terms of the needed number of frames) and faster adaptation while showing competitive performance to the traditional fine-tuning approach. (4) Finally, we can significantly improve the fine-tuning performance with our proposed adaptation of object localization module, showing complementarity of our idea with the traditional approach.

2 Adaptation To New Environments Via Object Localization

Our goal is to adapt an existing action recognition model trained on a source environment D_s to a target environment D_t. This problem has practical applications in the real world, for example, a system that is trained to recognize actions in one kitchen may be deployed in other kitchens where we cannot gather and annotate a comprehensive video dataset but wish to allow for some target

[2]Among the object-centric video models, ORViT is the most recent and it is easy to use.

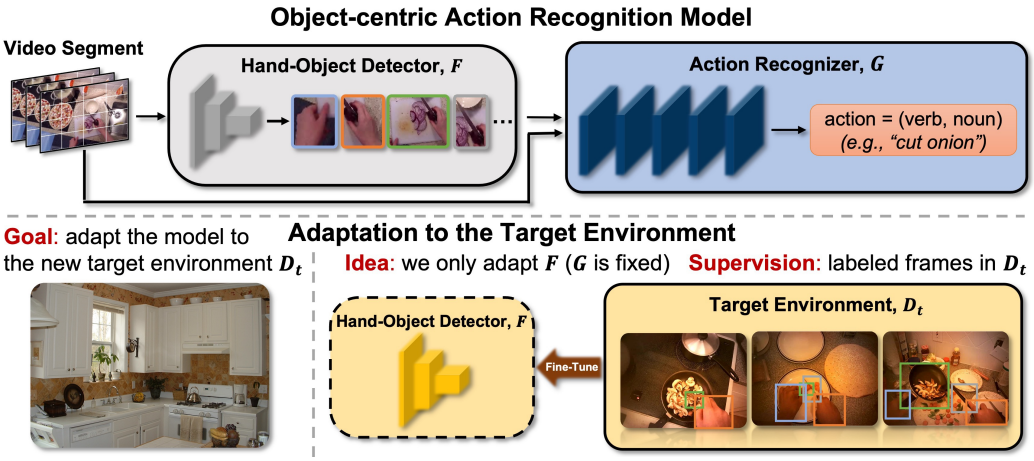


Figure 2: **The overview of the proposed approach:** (Top) we illustrate a generic object-centric action recognition model that consists of two modules: a hand-object detector F and an action recognizer G; (Bottom) we illustrate our adaptation approach via fine-tuning the object localizer F alone, based on a set of labeled images from the target environment.

adaptation. We start by describing the model (Section 2.1) and the problem statement (Section 2.2); then in Section 2.3, we describe how we adapt the model to a target environment via objects.

2.1 Object-Centric Video Model

We instantiate our framework with ORViT (Herzig et al., 2022), a recent object-centric video transformer model. Further we follow ORViT and its design choices. First, we assume that our model comprises two main modules: a hand-object detector F and an action recognizer G. We set G to be the ORViT action recognizer, and F to be the hand-object detection model of Shan et al. (2020b).

More formally, let $x = (z_1, ..., z_t) \in \mathbb{R}^{T\times H\times W\times 3}$ be an input video segment. The detector F is a function from a frame $z_i \in \mathbb{R}^{H\times W\times 3}$ to an ordered list of $N = 4$ boxes: left hand, right hand, left object, and right object.[3]For a given frame z_i, the predicted j^{th} box $\hat{b}_{ij} \in [0, 1]^4$ has an associated $\hat{p}_{ij} \in \mathbb{R}$ confidence score. Each left/right hand-object pair has an "in contact" or "no contact" binary prediction $\hat{c}_{ik} \in \mathbb{R}$ for $k \in \{1, 2\}$. The action recognizer is a mapping from a video segment $x \in \mathbb{R}^{T\times H\times W\times 3}$ and corresponding bounding boxes $b \in [0, 1]^{T\times N\times 4}$ to an action label, defined as a verb-noun pair: $G(x, b) \in V \times O$. While in this design we model both hands and objects, which is intuitive for many human-object interaction tasks, in a more general case, we talk about object localization broadly.

To predict a verb-noun pair given an input video segment $x = (z_1, ..., z_t) \in \mathbb{R}^{T\times H\times W\times 3}$, we employ the detector F to predict a list of bounding boxes in a frame-wise manner, then utilize the action recognizer G to make the final verb $\hat{v} \in V$ and noun $\hat{n} \in O$ predictions:

$$[\hat{b}, \hat{p}, \hat{c}] = F(x), \qquad [\hat{v}, \hat{n}] = G(x, \hat{b}), \tag{1}$$

where $\hat{b} \in [0, 1]^{T\times N\times 4}$ are the bounding box coordinates, $\hat{p} \in \mathbb{R}^{T\times N}$ are the confidence scores for left/right hand and left/right object, and $\hat{c} \in \mathbb{R}^{T\times 2}$ are the contact/no contact scores. In practice, $\hat{p}, \hat{c}$ are only used in F optimization. We describe how to adapt F to new environments in Section 2.3.

2.2 Adaptation Problem Statement

Given an existing action recognition model (as defined in Section 2.1) trained on some source environment D_s, our goal is to adapt it to a target environment D_t. In the target environment, we aim to classify the actions (i.e., verb-noun pairs) of new given test video segments. Importantly, we assume access to a set of n_t videos with sparse frame-level labels, $(x, b, c) \in D_t$, where each x is a

[3]Note, that we **do not** predict specific object labels, such as "knife" or "bowl".

video segment with a few associated bounding boxes and contact/no-contact information. Collecting videos (or even distinct images) of objects and sparsely annotating them with bounding boxes in the target environment is less prohibitive than recording and labeling entire demonstrations of diverse actions, and can be done by the end user of the system. Importantly, we aim to test our hypothesis that improving object localization in the target environment may improve action recognition as well.

2.3 ADAPTATION LOSS

The key idea in our work is to adapt the existing action recognition model to a target environment D_t by *only finetuning the object detector* F, leaving G frozen. Intuitively, this adaptation method deals with the uncertainty related to encountering previously unseen objects in the target environment, as well as other factors like the background and lightning conditions.

To adapt the model to the target environment, we fine-tune the hand-object detector F to localize objects in the target environment, utilizing the available annotations. Let T' be the set of annotated frames of the video. We minimize the following objective:

$$\mathcal{L}_{\text{HO}} := \sum_{i \in T'} \sum_{j=1}^{4} \text{BCE}(\text{Sigmoid}(\hat{p}_{ij}), p_{ij}) + p_{ij}(L_1(\hat{\boldsymbol{b}}_{ij}, \boldsymbol{b}_{ij})) \tag{2}$$

Where $p_{ij} \in \{0, 1\}$ is a binary indicator that receives 1 if the j^{th} object predicted in the i^{th} frame overlaps with the ground truth object.

$$\mathcal{L}_{Contact} := \sum_{i \in T'} \sum_{k=1}^{2} \text{CE}(\hat{c}_{ik}, c_{ik}) \tag{3}$$

To obtain the final loss, we compare the predictions and the ground-truth labels then add the two terms, and the total loss is:

$$\mathcal{L}_{Adaptation} := \mathcal{L}_{HO} + \mathcal{L}_{Contact} \tag{4}$$

Finally, during inference, we feed the target video segments into the adapted action recognition model: namely, we use the frame-level bounding boxes from the fine-tuned hand-object detector F as input to the frozen action recognizer G (see Equations 1).

3 EXPERIMENTS AND RESULTS

3.1 EXPERIMENTAL SETUP

Implementation details. Our training recipes and code are based on the ORViT model (Herzig et al., 2022). We train the source ORViT model for 50 epochs with the $lr = 10^{-4}$ and follow the standard learning rate decay schedule. Our method is implemented in Pytorch and we train the models on 8 NVIDIA Quadro RTX 6000 GPUs. For training we use a standard crop size of 224 and jitter the scales from 256 to 320. We perform inference on a single clip with 16 frames. For each sample, the evaluation frame is centered in frame 8. We use a crop size of 224 in test time. We take 1 spatial crop with 10 different clips sampled randomly to aggregate predictions over a single video in testing. We use the hand-object detector presented by Shan et al. (2020a), which is based on Faster R-CNN (Ren et al., 2015). The detector utilizes a ResNet-101 backbone He et al. (2016) that was pretrained on ImageNet (Russakovsky et al., 2015), and a Feature Pyramid Network (FPN) (Lin et al., 2017).

Datasets. We conduct experiments on **Epic-Kitchens** (Damen et al., 2020) and **EGTEA Gaze+** (Li et al., 2018b). **Epic-Kitchens** (EK), which is a large-scale dataset for first-person (egocentric) vision spanning a wide range of actions recorded by different participants in 4 cities belonging to 10 different nationalities in 45 native kitchen environments. **EGTEA Gaze+** (EGTEA) is another large dataset for egocentric vision actions that spans 7 cooking recipes, each performed by 4 different participants.

For training the hand-object detector we use the annotations from the **100K Frame** (Shan et al., 2020b) (100K for short), which is built upon **100 Days Of Hands** (Shan et al., 2020b) videos. It includes 99,899 frames annotated with 189K hand boxes, 189K hand states, 189K contact states,

Table 1: **Split information for the four unseen target kitchens in Epic-Kitchens dataset.**

Kitchen id	Scale	# Segments	# Frames of different fine-tuning sets						
			1%	5%	20%	35%	50%	70%	100%
4	L	7,917	17	99	392	684	979	1,368	1,955
8	M	2,650	16	80	318	557	794	1,111	1,590
22	L	10,732	19	93	363	637	913	1,276	1,826
24	S	1,804	12	59	235	411	588	821	1,174
24 (object-based)	S	1,804	61	92	259	441	603	847	1,174

Table 2: **The average of top-1 and top-5 accuracy of action (A), noun (N) and verb (V) for for action recognition in the 4 kitchens of EK.** The ↑ denotes the absolute maximum increase between the highest and base. the relative maximum increase w.r.t base is shown in the first row.

Fine-tuning Set	Top 1			Top 5		
	A	N	V	A	N	V
Relative ↑	8.49%	6.96%	5.67%	5.73%	3.80%	2.35%
100%	**20.12** (↑**1.6**)	**31.86** (↑**2.1**)	**46.29** (↑**2.5**)	**46.85** (↑**2.5**)	**55.66** (↑**2.1**)	**75.47** (↑**1.7**)
70%	20.05	31.74	45.80	46.74	55.57	75.32
50%	19.95	31.59	46.09	46.66	55.53	75.32
35%	19.75	31.68	45.86	46.56	55.48	75.17
20%	19.89	31.43	45.69	46.06	55.25	75.02
5%	19.57	31.11	45.25	45.65	54.80	74.70
1%	19.18	30.71	44.72	45.19	54.34	74.43
base	**18.54**	**29.79**	**43.81**	**44.31**	**53.62**	**73.74**

140K object boxes. We also use **Ego-centric Hands**, a collection of 42,625 frames from EK and EGTEA similarly labeled with hand-object interactions, which we obtain from Shan et al. (2020b).

Choosing target fine-tuning set. For adaptation, we fine-tune the hand-object detector on a subset of labeled frames in the target environment, which are sourced from the **Ego-centric Hands**. Naturally, we aim to capture a wide variety of objects in our target fine-tuning set. We also want to experiment with different amounts of labeled frames. Thus our data "splitting" strategy is as follows. Within each target environment, we have several video sequences that include different actions and objects; some of the frames in these sequences are sparsely labeled with hand-object bounding boxes. To cover the wide variety of scenes, we randomly select 1%, 5%, ..., 100% of the available annotated frames from each video sequence; we combine them all to get the final set. More details on the data splits are given in Sections 3.2 and 3.3.

Evaluation Metrics. We report Accuracy of top-1 / top-5 noun, top-1 / top-5 verb, top-1 / top-5 action predictions to evaluate the models' performance, where action is a verb-noun pair.

3.2 Adaptation to an unseen environment within a dataset

Goal. In this setting, our goal is to evaluate how an action recognition model adapts to a new kitchen environment, assuming that the same camera device, recording setup and annotation strategy are adopted. When adapting to new kitchen, factors like different lightning, background, objects, and behavior of the participants (people who prepare the food), result in performance degradation.

Data. We build upon the **Epic-Kitchens** (EK) dataset. We choose 4 kitchens of different scales (small, medium, large) from the total of 45 kitchens in EK as the unseen target environments; then we use the split strategy introduced in Section 3.1 to generate several fine-tuning sets. The detailed statistics are shown in Table 1.[4]

Experiment description. First, the ORViT model is trained on video segments in all the EK kitchens except the 4 target ones. Meanwhile, we pre-train the base hand-object detector on the 100K dataset and part of the frames in Ego-centric hands (namely, all EK kitchens except the 4 target ones). Next,

[4]The "24 (object-based)" split will be introduced in Section 3.4.1.

Table 3: **Split information for the 7 target recipes (all in an unseen kitchen) in EGTEA dataset.**

Recipe id	Scale	# Segments	# Frames of different fine-tuning sets						
			1%	5%	20%	35%	50%	70%	100%
BaconAndEggs	L	3,171	48	238	954	1,671	2,387	3,344	4,775
Cheeseburger	M	1,217	15	75	300	524	747	1,045	1,493
ContinentalBreakfast	M	1,209	17	82	325	572	816	1,143	1,633
GreekSalad	M	1,263	17	79	315	549	786	1,100	1,572
PastaSalad	L	4,454	65	328	1,310	2,291	3,274	4,584	6,546
Pizza	S	357	9	46	185	325	464	652	930
TurkeySandwich	M	1,123	14	69	276	487	694	972	1,388

the detector is fine-tuned for 7 epochs using different fine-tuning sets from the 4 target kitchens with the $lr = 10^{-3}$ (quantitative and qualitative results of the fine-tuning are included in the Supp. C). Finally, video segments in 4 target kitchens are used to evaluate the action recognition accuracy.

Results. We report the average accuracy for the 4 target kitchens in Table 2. Across the three sub-tasks (Action, Verb and Noun), using our approach leads to an average absolute improvement of around 2%, with relative improvement of over 8% in top-1 Action accuracy. When using only 1% of labeled frames (around 16 frames), we find that the Noun and Verb top-1 accuracy already increases by 1% while the Action top-1 accuracy increased by 0.5% compared to the base detector that was only trained on the source domain. Therefore, we find that better localization of objects in the new environment also benefits action recognition. Results per kitchen are included in the Supp. A.

3.3 Adaptation to an unseen environment across datasets

Goal. In this scenario, we perform domain adaptation from Epic-Kitchens to EGTEA dataset. This scenario is particularly challenging because recording instructions, recording hardware and annotations protocols differ between the datasets. These differences serve as extra domain gap factors besides the ones mentioned in the first scenario (different participants, physical environments, etc.), making the adaptation more challenging in comparison to the first one. Our goal here is to evaluate how our method performs when facing a larger domain gap.

Data. Since EGTEA is annotated based on 7 different recipes, here we create target environments based on each recipe recorded in the same kitchen. We again use the split strategy introduced in Section 3.1 to generate the fine-tuning sets. The detailed statistics are given in Table 3.

Table 4: **The average top-1 and top-5 accuracy of action (A), noun (N) and verb (V) for for action recognition on the 7 recipes in EGTEA.** The ↑ denotes the absolute maximum increase between the highest and base. the relative maximum increase w.r.t base is shown in the first row.

Fine-tuning Set	Top 1			Top 5		
	A	N	V	A	N	V
Relative ↑	14.68%	8.84%	5.75%	6.83%	5.54%	2.41%
100%	**11.80** (↑1.5)	**21.65** (↑1.8)	39.82	**35.42** (↑2.3)	**41.70** (↑2.2)	**71.50** (↑1.7)
70%	11.71	21.45	**39.89** (↑2.2)	35.24	41.57	71.20
50%	11.71	21.17	39.64	35.06	41.39	71.27
35%	11.53	21.19	39.48	34.80	41.14	70.90
20%	11.20	20.94	39.26	34.44	40.68	70.69
5%	11.05	20.67	38.90	34.07	40.49	70.54
1%	10.58	20.41	38.38	33.90	40.15	70.21
base	**10.29**	**19.89**	**37.72**	**33.15**	**39.51**	**69.82**

Experiment description. First, the ORViT model is trained on all the kitchens in the EK dataset. Meanwhile, we pre-train the base hand-object detector on the 100K dataset and part of the frames in Ego-centric hands (i.e., all the EK kitchens). Next, the pre-trained detector is fine-tuned for 7 epochs using different fine-tuning sets in EGTEA with the $lr = 10^{-3}$ (quantitative and qualitative results of

Table 5: **The comparison between video-based split (left) and object-based split (right) for kitchen** 24 **in EK.** The ↑ denotes the absolute maximum increase between the highest and base. the relative maximum increase w.r.t base is shown in the first row.

Video-based Split						Fine-tuning Set	Object-based Split					
Top 1			Top 5				Top 1			Top 5		
A	N	V	A	N	V		A	N	V	A	N	V
6.41%	3.46%	5.02%	4.67%	3.46%	2.06%	Relative ↑	7.56%	4.99%	4.98%	5.40%	4.18%	2.06%
↑1.6	**↑1.3**	**↑2.6**	**↑2.5**	**↑2.2**	**↑1.7**	Absolute ↑	**↑1.8**	**↑1.8**	**↑2.5**	**↑2.9**	**↑2.6**	**↑1.7**
25.89	**37.97**	53.54	**55.82**	64.36	**81.92**	100%	25.89	37.97	**53.54**	55.82	64.36	**81.92**
25.78	37.80	52.22	55.76	**64.63**	81.87	70%	**26.17**	**38.53**	52.66	**56.21**	**65.08**	81.71
25.42	37.62	**53.56**	55.71	64.47	81.65	50%	25.06	37.36	52.38	55.49	64.63	81.21
24.61	37.71	52.55	55.76	64.36	81.65	35%	24.67	37.25	52.44	55.27	64.02	81.65
25.42	37.54	52.27	55.38	64.02	81.46	20%	24.50	37.69	52.22	55.21	64.47	81.10
25.21	37.36	52.00	54.88	63.75	81.15	5%	24.83	37.14	52.72	54.93	63.80	81.60
24.97	36.97	51.47	54.16	63.36	80.93	1%	24.22	37.08	52.27	54.66	63.75	80.88
24.33	**36.70**	**51.00**	**53.33**	**62.47**	**80.27**	Base	**24.33**	**36.70**	**51.00**	**53.33**	**62.47**	**80.27**

fine-tuning are provided in the Supp. C). Finally, 7 target environments' video segments are used to test the action recognition accuracy.

Results. We report the average accuracy for the 7 target kitchens in Table 4.[5] Again, across the three sub-tasks, using our approach leads to an average absolute improvement of around 2%, with relative improvement of over 14% in top-1 Action accuracy. All the three tasks show similar trends as in the first scenario, showing the generalization of our proposed method. We notice that the average accuracy when using the base detector is around 10% lower than in the first scenario, which is expected since there is a larger domain gap across datasets, making adaptation more difficult.[6] This further highlights the robustness of the proposed method: precise localization of the objects obtained from only small number of annotated frames could help overcome a challenging domain gap and obtain a visible improvement in action recognition task.

3.4 Analysis

3.4.1 Ablation of data splitting scheme

With the video-based split approach proposed in Section 3.1, there is a risk of uneven coverage of objects due to their uneven distribution across videos. For example, a lot of videos might involve "a knife" but only one video involves "a spoon". We therefore analyze another fine-tuning set splitting strategy we refer to as "object-based", which bestows the fine-tuning set with more instance-level object information. To evaluate how the two splitting approaches compare, we manually label the categories of the available bounding boxes from one EK kitchen (24), then create finetuning sets while ensuring an even distribution of the labeled frames across objects. See Table 1 (bottom row) for split information and Supp. D for object category statistics. We evaluate the adaptation performance on the new obtained sets (see Table 3.4.1). We find that while the object-based split is intuitively preferable, our approach, which does not require knowing the object labels, achieves similar performance.

3.4.2 Adaptation via objects and/or actions

Finally, we compare finetuning of the hand-object detector with finetuning of the action recognizer, which requires video segments labeled with verb-noun pairs from the target environment. Since this approach utilizes labeled video segments whereas we utilize relatively few frames annotated with bounding boxes, we view this an an alternative and complementary approach. We consider adaptation using full fine-tuning of the model G and partial finetuning of the last FC layer that predicts verb and noun ("linear probing"). We experiment with the EK kitchen 24 from the first "within dataset" scenario and the EGTEA recipe $GreekSalad$ from the second "across datasets" scenario. As shown in Tables 6, we report the average performance over the fine-tuning sets (1%, 5%, 20%, 35%, 50%, 70%, 100%), Memory (MiB) and Time (s) of the adaptation phase; to guarantee the fairness of the comparison, the batch size is set to 1 for all the models, and we use one Quadro RTX 6000 gpu. We provide additional implementation details and results in Suppl. G.

[5]The detailed results for each recipe are included in the Supp. B.

[6]For additional analysis of the domain gap please refer to the Supp. F.

Table 6: **The comparison of the action recognition accuracy between our object-based adaptation and the traditional fine-tuning methods.** G is the action recognizer, F is the object detector. The Memory, Time and Accuracy are all average values w.r.t the fine-tuning sets 1%, 5%, 20%, 35%, 50%, 70% and 100%. Scenario 1 is on EK kitchen 24, scenario 2 is on the EGTEA recipe $GreekSalad$.

Scenario	Method	Memory /MiB	Time /s	Top 1			Top 5		
				A	N	V	A	N	V
1	Base	/	/	24.33	36.70	51.00	53.33	62.47	80.27
	G Fine-tuning	8775	7,318	25.74	37.48	53.44	55.68	63.17	84.26
	G Linear Probing	3329	5,216	25.05	36.38	52.84	53.28	61.25	82.09
	F Fine-tuning (Ours)	3027	896	25.22	37.86	53.34	55.80	64.00	82.94
	G **Fine-tuning + Ours**	8775	8,128	**26.30**	**39.20**	**54.11**	**60.02**	**66.64**	**85.76**
2	Base	/	/	10.21	19.95	41.01	33.33	39.27	73.66
	G Fine-tuning	8775	14,643	11.97	22.68	42.16	34.84	40.73	74.78
	G Linear Probing	3329	23,324	11.11	20.16	40.05	31.77	37.41	73.38
	F Fine-tuning (Ours)	3027	1,054	12.12	21.86	42.31	34.58	39.98	73.74
	G **Fine-tuning + Ours**	8775	15,698	**16.46**	**28.61**	**45.53**	**43.79**	**51.09**	**76.75**

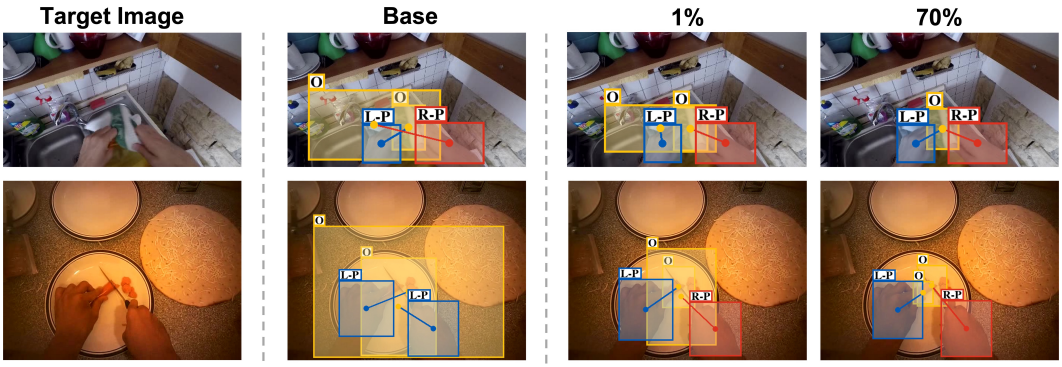


Figure 3: **Qualitative results for base detector F vs. fine-tuning F on 1% and 70% sets in the new environments.** We show the bounding boxes predicted by F. Each box has either object ("O"), left-hand ("L") or right-hand ("R") category. ("P" indicates contact between a hand and an object).

In both cases, jointly fine-tuning the detector F and action recognizer (AR) G performs the best. This indicates that the two approaches are complementary. While AR finetuning performs slightly better, it requires three times more memory and is at least seven times slower. Fine-tuning the AR model also requires recording and annotating videos of actions in a new environment, which might require tremendous effort compared to collecting and annotating frames of objects in the kitchen. While the linear probing approach to finetuning operates faster than full finetuning, it performs worse in practice. Therefore, our method achieves cheaper and faster adaptation compared to other adaptation avenues, while maintaining competitive performance. Overall, we believe that adaptation using objects is a promising direction both on its own or in combination with the traditional options.

3.4.3 Visualization of the fine-tuning of F

Figure 3 shows qualitative results for the fine-tuning of the hand-object detector F. In the first row, we randomly select an image from kitchen 22 in the EK that does not appear in our fine-tuning sets. We show the predictions of the base detector and the ones fine-tuned with different target sets. In the second row, we similarly show an image selected from the recipe $GreekSalad$ in EGTEA.

For instance, in the example of "cut carrots" in the second row, the base detector does not perform well (incorrect distinction of the left and right hands, inaccurate localization of the object), while after fine-tuning on the target environment the detector can better distinguish the left-right hands and the localization of the interacted object gets more precise. The object is better distinguished from the surrounding context (the counter, plate) to localize more fine-grained objects held by the hands (the carrot, knife.) More details including additional visualizations can be found in Supp. C.1.

4 RELATED WORK

Domain Adaptation for Action Recognition. Domain adaptation has been extensively studied outside of action recognition, including object detection (Ganin et al., 2016; Long et al., 2015; Sun & Saenko, 2016; Sun et al., 2019b), semantic segmentation (Huang et al., 2018; Zhang et al., 2017), and more (Hoffman et al., 2018; Harary et al., 2022; Tzeng et al., 2014). The topic of domain adaption for action recognition has only recently gained a great deal of attention. Most of the works focused on view-invariant action recognition (Kong et al., 2017; Li et al., 2018a; Liu et al., 2017; Rahmani & Mian, 2015; Sigurdsson et al., 2018), meaning adapting to the geometric transformations of a camera, while others have been focused on Unsupervised Domain Adaptation (UDA) for changes in environments, which has received limited attention until recently (Chen et al., 2019; Jamal et al., 2018; Pan et al., 2020; Munro & Damen, 2020). Most of these works have employed several modalities (RGB, Optical flow) (Pan et al., 2020; Munro & Damen, 2020) while some only use RGB (Chen et al., 2019; Jamal et al., 2018). Contrary to these works we focus on adapting to similar but unseen target environments by utilizing *object information* in target environments.

Action Recognition. As a long-standing problem in computer vision, various approaches have been proposed for action recognition. Ranging from the early works of optical flow (Efros et al., 2003), using recurrent networks (Donahue et al., 2015; Yue-Hei Ng et al., 2015), through to 3D spatio-temporal kernels (Ji et al., 2013; Taylor et al., 2010; Tran et al., 2015; Varol et al., 2018; Lin et al., 2019; Wang et al., 2019; Carreira & Zisserman, 2017), and two-stream networks(Feichtenhofer et al., 2016; Simonyan & Zisserman, 2014; Feichtenhofer et al., 2019)) that capture complementary signals (e.g., motion and spatial cues (Feichtenhofer et al., 2016; Simonyan & Zisserman, 2014; Feichtenhofer et al., 2019)). Recently, Vision Transformers have become the dominant approach to computer vision in general and to action recognition in particular. Several recent video transfromer works performed well on multiple video datasets, such as ViViT (Arnab et al., 2021a), MViT (Fan et al., 2021), MFormer (Patrick et al., 2021), TimeSformer (Bertasius et al., 2021), MViTv2 (Li et al., 2022), Video Swin (Liu et al., 2021). More recently, ORViT (Herzig et al., 2022) proposed to incorporate objects into video transformers for a better action recognition. In this work, we leverage the ORViT model by exploiting objects to adapt action recognition models to new environments without recording or annotating them.

Structured Models. Recently, structured models have been successfully applied to a wide range of computer vision applications, including vision and language (Chen et al., 2020; Li et al., 2019; 2020; Tan & Bansal, 2019), video relation detection (Liang et al., 2019; Santoro et al., 2017; Sun et al., 2019a), human-object interactions (Gao et al., 2020; Kato et al., 2018; Xu et al., 2019), relational reasoning (Baradel et al., 2018; Battaglia et al., 2018; Herzig et al., 2018; Krishna et al., 2018; Jerbi et al., 2020; Raboh et al., 2020; Xu et al., 2020; Zambaldi et al., 2018), and even the generation of images and videos (Bar et al., 2021; Herzig et al., 2020; Johnson et al., 2018). The advances and the success of structured models in these domains inspired various video-based tasks, such as action localization Arnab et al. (2021b); Nawhal & Mori (2021); Wu & Krähenbühl (2021), video synthesis (Bar et al., 2021), and action recognition (Ben-Avraham et al., 2022; Girdhar et al., 2019; 2017; Herzig et al., 2022; 2019; Ji et al., 2019; Materzynska et al., 2020; Ma et al., 2018; Nagarajan et al., 2020; Sun et al., 2018; Wang & Gupta, 2018). Although the works above suggest that objects are useful to video-based tasks, our work explores the adaptation of video models to novel environments.

5 CONCLUSION

In this work we have addressed a practical problem of adapting an existing action recognition model to a new unseen environment. Our key idea is that adapting object-centric video models is possible via only fine-tuning their object localization modules. We have considered two challenging scenarios, namely "within" and "across" dataset adaptation, using the available large-scale egocentric cooking benchmarks. We have shown that our proposed approach is cheaper and more efficient than the traditional fine-tuning, while providing similar accuracy. We have also offered an analysis of an alternative scheme for selecting supervisory frames, as well as a study of complementarity between the traditional and our approach. We believe this is a promising direction for researchers interested in related tasks, including domain adaptation and few-shot learning, in context of action recognition.

REFERENCES

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer, 2021a.

Anurag Arnab, Chen Sun, and Cordelia Schmid. Unified graph structured models for video understanding. In *ICCV*, 2021b.

Amir Bar, Roei Herzig, Xiaolong Wang, Anna Rohrbach, Gal Chechik, Trevor Darrell, and A. Globerson. Compositional video synthesis with action graphs. In *ICML*, 2021.

Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *ECCV*, pp. 105–121, 2018.

Peter W Battaglia, Jessica B Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinicius Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

Elad Ben-Avraham, Roei Herzig, Karttikeya Mangalam, Amir Bar, Anna Rohrbach, Leonid Karlinsky, Trevor Darrell, and Amir Globerson. Bringing image scene structure to video via frame-clip consistency of object tokens, 2022.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017.

Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6321–6330, 2019.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *ECCV*, 2020.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. In *European Conference on Computer Vision (ECCV)*, 2018.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4125–4141, 2020.

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2625–2634, 2015.

Alexei A Efros, Alexander C Berg, Greg Mori, and Jitendra Malik. Recognizing action at a distance. In *Computer Vision, IEEE International Conference on*, volume 3, pp. 726–726. IEEE Computer Society, 2003.

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*, 2021.

C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6201–6210, 2019. doi: 10.1109/ICCV.2019.00630.

Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1933–1941, 2016.

Yaroslav Ganin, E. Ustinova, Hana Ajakan, Pascal Germain, H. Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. In *J. Mach. Learn. Res.*, 2016.

Chen Gao, Jiarui Xu, Yuliang Zou, and Jia-Bin Huang. Drg: Dual relation graph for human-object interaction detection. *ArXiv*, abs/2008.11714, 2020.

Rohit Girdhar, Deva Ramanan, Abhinav Kumar Gupta, Josef Sivic, and Bryan C. Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3165–3174, 2017.

Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 244–253, 2019.

Sivan Harary, Eli Schwartz, Assaf Arbelle, Peter Staar, Shady Abu-Hussein, Elad Amrani, Roei Herzig, Amit Alfassy, Raja Giryes, Hilde Kuehne, Dina Katabi, Kate Saenko, Rogerio S. Feris, and Leonid Karlinsky. Unsupervised domain generalization by learning a bridge across domains. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5280–5290, June 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Roei Herzig, Moshiko Raboh, Gal Chechik, Jonathan Berant, and Amir Globerson. Mapping images to scene graphs with permutation-invariant structured prediction. In *Advances in Neural Information Processing Systems (NIPS)*, 2018.

Roei Herzig, Elad Levi, Huijuan Xu, Hang Gao, Eli Brosh, Xiaolong Wang, Amir Globerson, and Trevor Darrell. Spatio-temporal action graph networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

Roei Herzig, Amir Bar, Huijuan Xu, Gal Chechik, Trevor Darrell, and Amir Globerson. Learning canonical representations for scene graph to image generation. In *European Conference on Computer Vision*, 2020.

Roei Herzig, Elad Ben-Avraham, Karttikeya Mangalam, Amir Bar, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Object-region video transformers. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018.

Haoshuo Huang, Qixing Huang, and Philipp Krähenbühl. Domain transfer through deep activation matching. In *ECCV*, 2018.

Arshad Jamal, Vinay P Namboodiri, Dipti Deodhare, and KS Venkatesh. Deep domain adaptation in action space. In *BMVC*, volume 2, pp. 5, 2018.

Achiya Jerbi, Roei Herzig, Jonathan Berant, Gal Chechik, and Amir Globerson. Learning object detection from captions via textual scene attributes. *ArXiv*, abs/2009.14558, 2020.

Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as composition of spatio-temporal scene graphs. *arXiv preprint arXiv:1912.06992*, 2019.

Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.

Justin Johnson, Agrim Gupta, and Li Fei-Fei. Image generation from scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1219–1228, 2018.

Keizo Kato, Yin Li, and Abhinav Gupta. Compositional learning for human object interaction. In *ECCV*, 2018.

Donghyun Kim, Yi-Hsuan Tsai, Bingbing Zhuang, Xiang Yu, Stan Sclaroff, Kate Saenko, and Manmohan Chandraker. Learning cross-modal contrastive features for video domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13618–13627, 2021.

Yu Kong, Zhengming Ding, Jun Li, and Yun Fu. Deeply learned view-invariant features for cross-view action recognition. *IEEE Transactions on Image Processing*, 26(6):3028–3037, 2017.

Ranjay Krishna, Ines Chami, Michael S. Bernstein, and Li Fei-Fei. Referring relationships. *ECCV*, 2018.

Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Unsupervised learning of view-invariant action representations. *Advances in neural information processing systems*, 31, 2018a.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. Visualbert: A simple and performant baseline for vision and language. *ArXiv*, abs/1908.03557, 2019.

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. *ECCV 2020*, 2020.

Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *CVPR*, 2022.

Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 619–635, 2018b.

Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G. Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5718–5727, 2019.

Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944, 2017.

Mengyuan Liu, Hong Liu, and Chen Chen. Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition*, 68:346–362, 2017.

Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. *ArXiv*, abs/1502.02791, 2015.

Chih-Yao Ma, Asim Kadav, Iain Melvin, Zsolt Kira, Ghassan Al-Regib, and Hans Peter Graf. Attend and interact: Higher-order object interactions for video understanding. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6790–6800, 2018.

Joanna Materzynska, Tete Xiao, Roei Herzig, Huijuan Xu, Xiaolong Wang, and Trevor Darrell. Something-else: Compositional action recognition with spatial-temporal interaction networks. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

Jonathan Munro and Dima Damen. Multi-modal domain adaptation for fine-grained action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 122–132, 2020.

Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 160–169, 2020.

Megha Nawhal and Greg Mori. Activity graph transformer for temporal action localization, 2021.

Boxiao Pan, Zhangjie Cao, Ehsan Adeli, and Juan Carlos Niebles. Adversarial cross-domain action recognition with co-attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 11815–11822, 2020.

Mandela Patrick, Dylan Campbell, Yuki M. Asano, Ishan Misra Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and Joao F. Henriques. Keeping your eye on the ball: Trajectory attention in video transformers, 2021.

Moshiko Raboh, Roei Herzig, Gal Chechik, Jonathan Berant, and Amir Globerson. Differentiable scene graphs. In *WACV*, 2020.

Hossein Rahmani and Ajmal Mian. Learning a non-linear knowledge transfer model for cross-view action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2458–2466, 2015.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Dandan Shan, Jiaqi Geng, Michelle Shu, and David Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020a.

Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9869–9878, 2020b.

Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Actor and observer: Joint modeling of first and third-person videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7396–7404, 2018.

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pp. 568–576, 2014.

Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV Workshops*, 2016.

Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 318–334, 2018.

Xu Sun, Tongwei Ren, Yuan Zi, and Gangshan Wu. Video visual relation detection via multi-modal feature fusion. *Proceedings of the 27th ACM International Conference on Multimedia*, 2019a.

Yu Sun, Eric Tzeng, Trevor Darrell, and Alexei A. Efros. Unsupervised domain adaptation through self-supervision. *ArXiv*, abs/1909.11825, 2019b.

Hao Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*, 2019.

Graham W Taylor, Rob Fergus, Yann LeCun, and Christoph Bregler. Convolutional learning of spatio-temporal features. In *European conference on computer vision*, pp. 140–153. Springer, 2010.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 4489–4497, 2015.

Eric Tzeng, Judy Hoffman, N. Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *ArXiv*, abs/1412.3474, 2014.

Gül Varol, Ivan Laptev, and Cordelia Schmid. Long-term temporal convolutions for action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1510–1517, 2018.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(11):2740–2755, 2019.

Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, 2018.

Chao-Yuan Wu and Philipp Krähenbühl. Towards Long-Form Video Understanding. In *CVPR*, 2021.

Bingjie Xu, Yongkang Wong, Junnan Li, Qi Zhao, and M. Kankanhalli. Learning to detect human-object interactions with knowledge. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2019–2028, 2019.

Keyulu Xu, Jingling Li, Mozhi Zhang, Simon S. Du, Ken ichi Kawarabayashi, and Stefanie Jegelka. What can neural networks reason about? In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rJxbJeHFPS.

Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4694–4702, 2015.

Vinicius Zambaldi, David Raposo, Adam Santoro, Victor Bapst, Yujia Li, Igor Babuschkin, Karl Tuyls, David Reichert, Timothy Lillicrap, Edward Lockhart, et al. Relational deep reinforcement learning. *arXiv preprint arXiv:1806.01830*, 2018.

Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2039–2049, 2017.

SUPPLEMENTARY MATERIAL

In the supplementary material, we show the detail of the action recognition accuracy w.r.t each single kitchen of scenario 1 in A and each single recipe of scenario 2 in B. Then, we provide details about our fine-tuning of the hand-object detector F in C which include some quantitative and qualitative results and our analysis. Then, in D, we show some statistics about the distributions of the frame-level objects and segment-level action annotations in kitchen 24 of Epic-Kitchens as an environment. Next, we introduce the image-to-segment scheme that we use in our main paper to map a frame to its parent video segment. Next, we give some statistics about the domain gaps between two datasets: EK and EGTEA, based on which we deliver our experiment of scenario 2 (domain adaptation across datasets). Finally, details of the implementation and results of section 3.4.2 are provided in G.

A KITCHEN-LEVEL RESULTS OF THE FIRST SCENARIO

In this section, we include the kitchen-level results of the accuracy for action recognition in our 4 kitchens in EK in target domain as shown in Table 7, Table 8, Table 9 and Table 10. We also plot the accuracy of the sub-tasks with respect to the different fine-tuning sets in Figure 4, Figure 5, Figure 6 and Figure 7, which give a straightforward sense of the increase trend across fine-tuning sets and three sub-tasks.

Table 7: **The top-1 and top-5 accuracy of action (A), noun (N) and verb (V) for action recognition in kitchen 4.** The ↑ denotes the absolute maximum increase between the highest and base. the relative maximum increase w.r.t base is shown in the first row.

Fine-tuning Set	Top 1			Top 5		
	A	N	V	A	N	V
Relative ↑	15.75%	12.71%	5.88%	9.40%	7.03%	2.43%
100%	**11.39** (↑1.6)	**21.9** (↑2.5)	**33.83** (↑1.9)	**31.54** (↑2.7)	**42.96** (↑2.8)	**62.47** (↑1.5)
70%	11.32	21.76	33.82	31.34	42.68	62.35
50%	11.29	21.72	33.79	31.3	42.47	62.34
35%	11.3	21.64	33.74	31.03	42.43	62.25
20%	11.19	21.26	33.7	30.66	42.24	61.97
5%	10.6	20.94	33.5	30.15	41.78	61.4
1%	10.12	20.49	32.66	29.68	41.07	61.37
Base	**9.84**	**19.43**	**31.95**	**28.83**	**40.14**	**60.99**

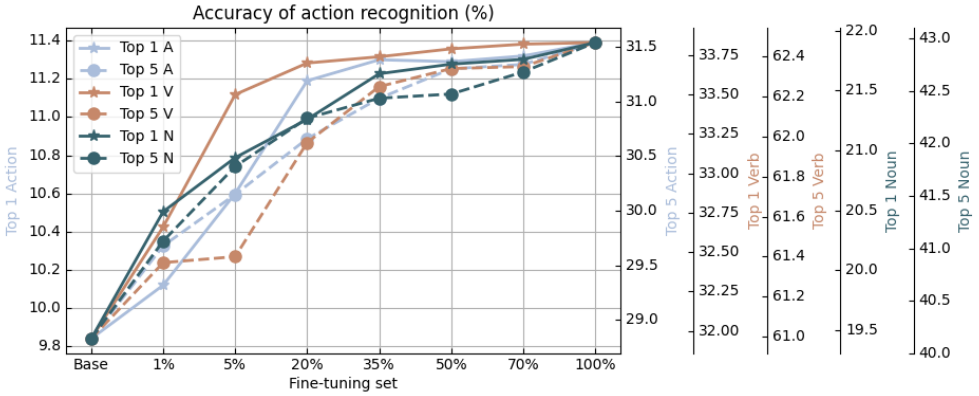


Figure 4: **The accuracy curve of top-1 and top-5 action (A), noun (N) and verb (V) in kitchen 4.**

Table 8: **The top-1 and top-5 accuracy of action (A), noun (N) and verb (V) for action recognition in kitchen 8.** The maximum increase is the difference between the highest and base. ↑ denotes the absolute maximum increase between the highest and base. the relative maximum increase w.r.t base is shown in the first row.

Fine-tuning Set	Top 1			Top 5		
	A	N	V	A	N	V
Relative (↑)	7.54%	7.68%	6.07%	6.18%	2.71%	3.29%
100%	**20.11** (↑1.4)	**33.77** (↑2.4)	**45.09** (↑2.6)	**50.72** (↑3.0)	**59.13** (↑1.6)	**77.25** (↑2.5)
70%	20.04	33.74	44.6	50.57	58.83	76.87
50%	20.01	33.36	44.45	50.38	59.11	77.06
35%	19.96	**33.77**	44.6	50.28	58.98	77.17
20%	19.89	33.43	44.3	49.26	58.79	76.6
5%	19.7	33.06	43.38	48.92	58.42	76.45
1%	19.48	32.89	43	48.68	58.06	75.98
base	**18.7**	**31.36**	**42.51**	**47.77**	**57.57**	**74.79**

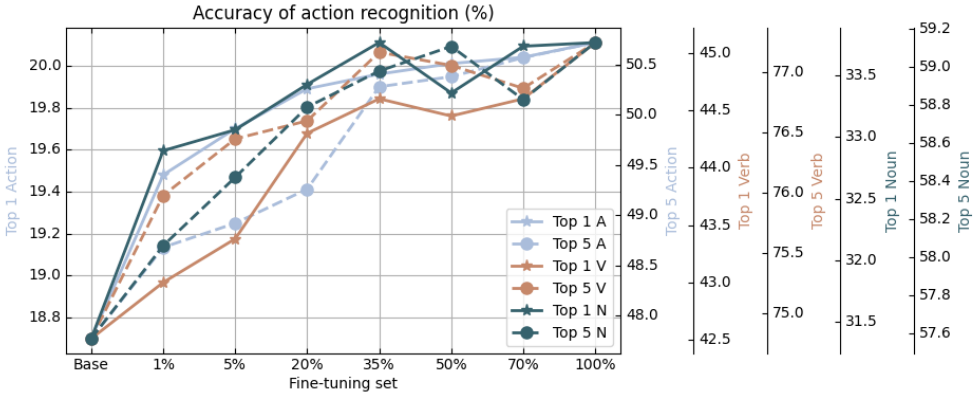


Figure 5: **The accuracy curve of top-1 and top-5 action (A), noun (N) and verb (V) in kitchen 8.**

Table 9: **The top-1 and top-5 accuracy of action (A), noun (N) and verb (V) for action recognition in kitchen 22.** The maximum increase is the difference between the highest and base. ↑ denotes the absolute maximum increase between the highest and base. the relative maximum increase w.r.t base is shown in the first row.

Fine-tuning Set	Top 1			Top 5		
	A	N	V	A	N	V
Relative (↑)	8.54%	6.76%	5.89%	4.25%	3.44%	1.69%
100%	23.08	**33.81** (↑2.1)	**52.69** (↑2.9)	**49.33** (↑2.0)	**56.17** (↑1.9)	**80.25** (↑1.3)
70%	23.07	33.65	52.57	49.3	56.14	80.2
50%	23.06	33.64	52.55	49.23	56.08	80.22
35%	**23.12** (↑1.8)	33.59	52.55	49.15	56.14	80.19
20%	23.07	33.47	52.47	48.93	55.94	80.05
5%	22.75	33.06	52.13	48.64	55.23	79.8
1%	22.15	32.5	51.73	48.23	54.86	79.42
base	**21.3**	**31.67**	**49.76**	**47.32**	**54.3**	**78.92**

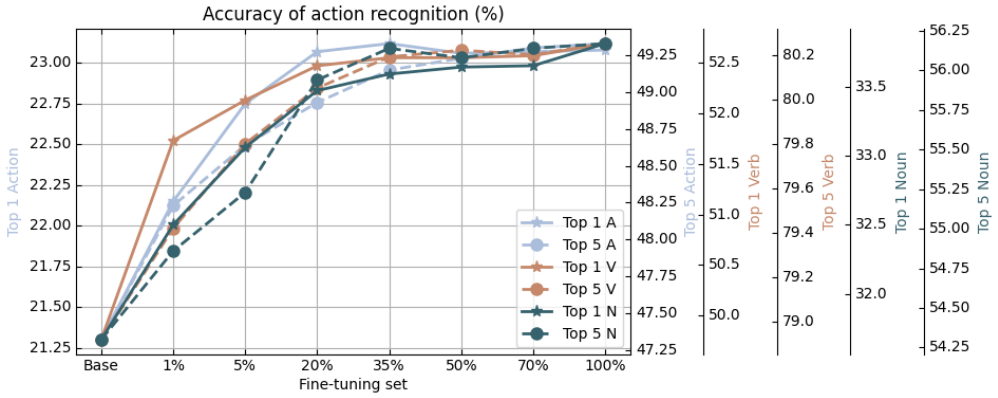


Figure 6: **The accuracy curve of top-1 and top-5 action (A), noun (N) and verb (V) in kitchen 22.**

Table 10: **The top-1 and top-5 accuracy of action (A), noun (N) and verb (V) for action recognition in kitchen 24 (video-based split).** ↑ denotes the absolute maximum increase between the highest and base. the relative maximum increase w.r.t base is shown in the first row.

Fine-tuning Set	Top 1			Top 5		
	A	N	V	A	N	V
Relative (↑)	6.41%	3.46%	5.02%	4.67%	3.46%	2.06%
100%	**25.89** (↑1.6)	**37.97** (↑1.3)	53.54	**55.82** (↑2.5)	64.36	**81.92** (↑1.7)
70%	25.78	37.80	52.22	55.76	**64.63** (↑2.2)	81.87
50%	25.42	37.62	**53.56** (↑2.6)	55.71	64.47	81.65
35%	24.61	37.71	52.55	55.76	64.36	81.65
20%	25.42	37.54	52.27	55.38	64.02	81.46
5%	25.21	37.36	52.00	54.88	63.75	81.15
1%	24.97	36.97	51.47	54.16	63.36	80.93
base	**24.33**	**36.70**	**51.00**	**53.33**	**62.47**	**80.27**

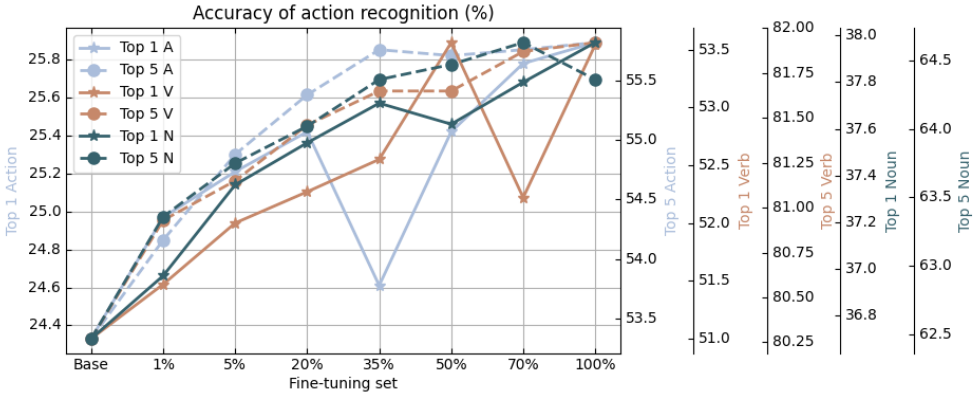


Figure 7: **The accuracy curve of top-1 and top-5 action (A), noun (N) and verb (V) in kitchen 24 (video-based split).**

B RECIPE-LEVEL RESULTS OF THE SECOND SCENARIO

In this section, we include the recipe-level results of the accuracy for action recognition in our 7 recipes in EGTEA of the target domain as shown in Table 11, Table 12, Table 13, Table 14, Table 15, Table 16 and Table 17. We also plot the accuracy of the sub-tasks with respect to the fine-tuning set in Figure 8, Figure 9, Figure 10, Figure 11, Figure 12, Figure 13 and Figure 14, which give a straightforward sense of the increase trend across fine-tuning sets and three sub-tasks.

Table 11: **The top-1 and top-5 accuracy of action, noun and verb for for action recognition of GreekSalad.** ↑ denotes the absolute maximum increase between the highest and base. the relative maximum increase w.r.t base is shown in the first row.

Fine-tuning Set	Top 1			Top 5		
	A	N	V	A	N	V
Relative (↑)	22.53%	10.33%	8.32%	7.14%	5.65%	2.86%
100%	12.51	**22.01** (↑2.1)	**44.42** (↑3.4)	**35.71** (↑2.4)	41.09	**75.77** (↑2.1)
70%	**12.59** (↑2.3)	21.85	44.1	35.24	40.93	75.38
50%	12.67	21.85	**44.42** (↑3.4)	35.33	**41.49** (↑2.2)	75.61
35%	12.43	21.69	44.03	35.31	41.41	75.06
20%	11.64	21.3	43.87	35.15	40.99	74.66
5%	11.32	20.9	43.52	34.63	40.73	74.38
1%	11.06	21.06	42.1	34.39	40.65	74.11
base	**10.21**	**19.95**	**41.01**	**33.33**	**39.27**	**73.66**

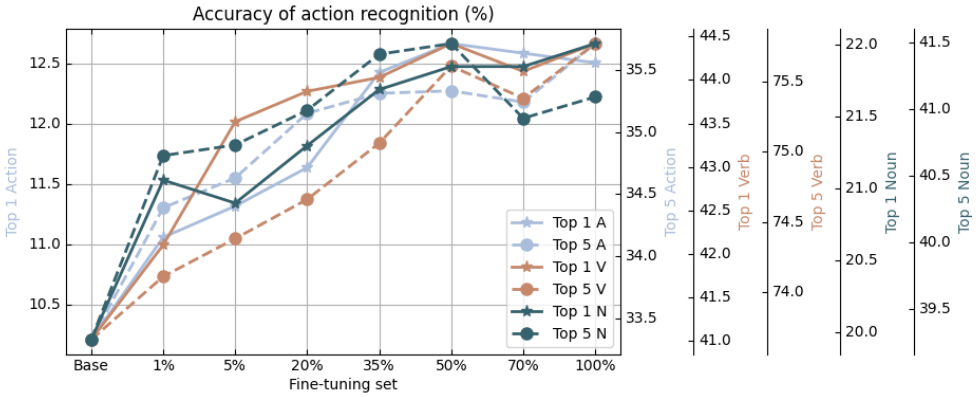


Figure 8: **The accuracy curve of top-1 and top-5 action (A), noun (N) and verb (V) in GreekSalad.**

Table 12: **The top-1 and top-5 accuracy of action (A), noun (N) and verb (V) for action recognition of Pizza.** ↑ denotes the absolute maximum increase between the highest and base. the relative maximum increase w.r.t base is shown in the first row.

Fine-tuning Set	Top 1			Top 5		
	A	N	V	A	N	V
Relative (↑)	20.08%	6.75%	8.98%	5.96%	5.54%	3.37%
100%	**12.08** (↑2.0)	39.22	**25.01** (↑2.1)	**45.54** (↑2.6)	36.81	**77.31** (↑2.5)
70%	11.96	**39.54** (↑2.5)	24.93	45.02	**36.97** (↑1.9)	76.47
50%	12.04	38.87	24.37	44.26	36.69	76.19
35%	11.76	38.66	24.65	43.98	36.01	75.63
20%	10.92	38.22	23.81	43.47	35.85	75.35
5%	11.48	38.01	23.53	43.26	35.29	75.26
1%	10.17	37.82	23.09	43.04	35.12	74.99
base	**10.06**	**37.04**	**22.95**	**42.98**	**35.03**	**74.79**

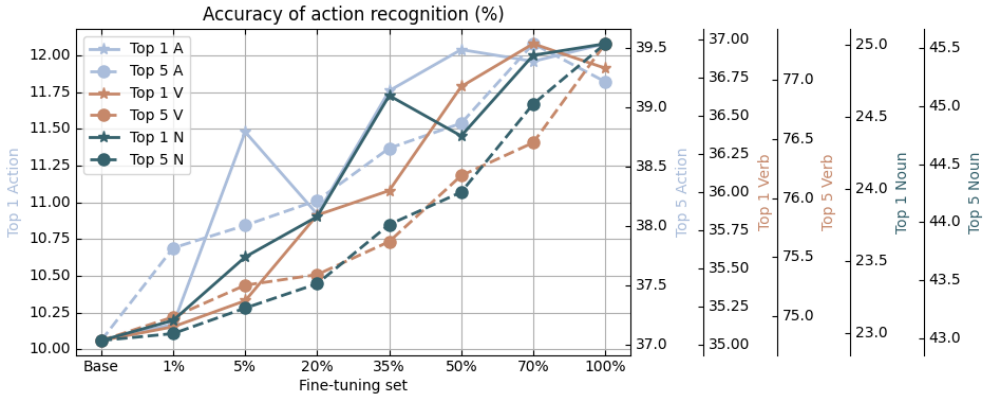


Figure 9: **The accuracy curve of top-1 and top-5 action (A), noun (N) and verb (V) in Pizza.**

Table 13: **The top-1 and top-5 accuracy of action (A), noun (N) and verb (V) for action recognition of Cheeseburger.** ↑ denotes the absolute maximum increase between the highest and base. the relative maximum increase w.r.t base is shown in the first row.

Fine-tuning Set	Top 1			Top 5		
	A	N	V	A	N	V
Relative (↑)	6.53%	6.20%	5.50%	6.51%	5.20%	1.98%
100%	**11.91** (↑0.7)	**22.43** (↑1.3)	36.98	**35.01** (↑2.1)	40.35	69.56
70%	11.50	22.19	**37.24** (↑1.9)	34.61	**40.44** (↑2.0)	69.42
50%	**11.91** (↑0.7)	22.02	36.89	34.36	40.35	**69.88** (↑1.4)
35%	11.42	21.97	36.57	34.02	39.85	69.57
20%	11.75	21.77	36.02	33.44	39.77	69.43
5%	11.59	21.42	35.73	33.20	39.41	69.35
1%	11.34	21.61	35.65	32.96	38.97	68.87
base	**11.18**	**21.12**	**35.30**	**32.87**	**38.44**	**68.52**

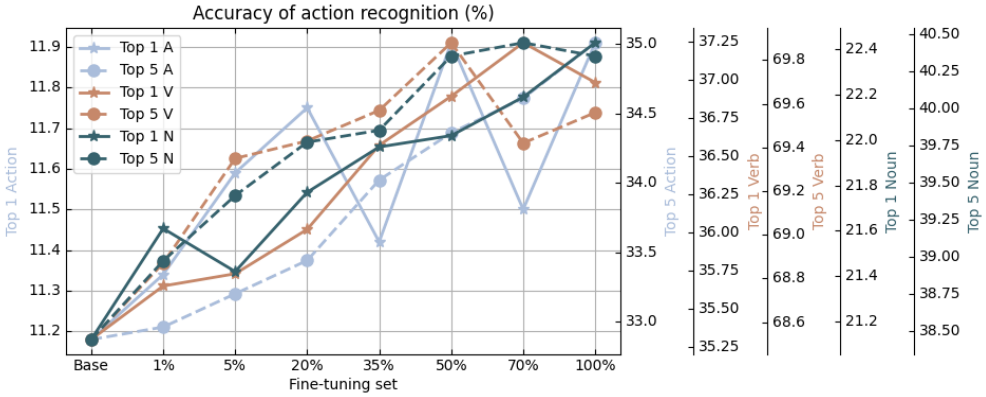


Figure 10: **The accuracy curve of top-1 and top-5 action (A), noun (N) and verb (V) in Cheeseburger.**

Table 14: **The top-1 and top-5 accuracy of action (A), noun (N) and verb (V) for action recognition of BaconAndEggs.** ↑ denotes the absolute maximum increase between the highest and base. the relative maximum increase w.r.t base is shown in the first row.

Fine-tuning Set	Top 1			Top 5		
	A	N	V	A	N	V
Relative (↑)	11.05%	8.15%	5.66%	6.05%	5.30%	2.39%
100%	**13.17** (↑1.3)	25.21	37.84	**36.62** (↑2.1)	**45.32** (↑2.3)	**68.67** (↑1.6)
70%	13.02	25.18	**37.87** (↑2.0)	36.50	45.13	68.55
50%	12.92	25.15	37.68	36.46	45.04	68.51
35%	12.76	25.15	37.64	36.04	44.88	68.52
20%	12.80	**25.34** (↑1.9)	37.50	35.56	44.38	68.48
5%	12.26	24.90	37.06	35.34	43.91	68.26
1%	12.07	24.02	36.59	34.83	43.37	67.74
base	**11.86**	**23.43**	**35.84**	**34.53**	**43.04**	**67.07**

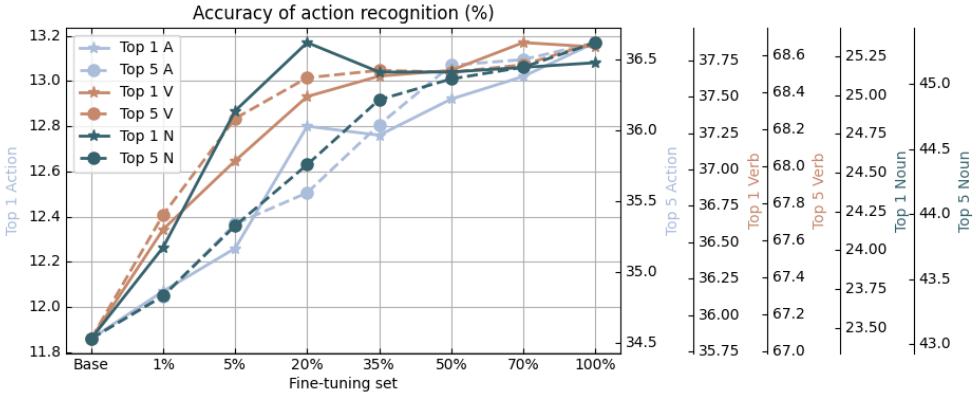


Figure 11: **The accuracy curve of top-1 and top-5 action (A), noun (N) and verb (V) in BaconAndEggs.**

Table 15: **The top-1 and top-5 accuracy of action (A), noun (N) and verb (V) for action recognition of TurkeySandwich.** ↑ denotes the absolute maximum increase between the highest and base. the relative maximum increase w.r.t base is shown in the first row.

Fine-tuning Set	Top 1			Top 5		
	A	N	V	A	N	V
Relative (↑)	19.50%	7.93%	6.28%	7.16%	5.00%	2.81%
100%	10.32	**19.32** (↑1.4)	39.18	**31.56** (↑2.1)	**37.58** (↑1.8)	**69.10** (↑1.9)
70%	**10.42** (↑1.7)	18.97	**39.27** (↑2.3)	31.08	37.49	68.92
50%	10.24	18.79	38.20	30.99	37.49	68.66
35%	10.15	18.72	38.74	30.81	36.95	68.30
20%	9.53	18.70	38.47	30.45	36.22	67.97
5%	9.44	18.61	37.76	30.28	36.49	67.68
1%	8.90	18.43	37.67	30.24	36.02	67.48
base	**8.72**	**17.90**	**36.95**	**29.45**	**35.79**	**67.21**

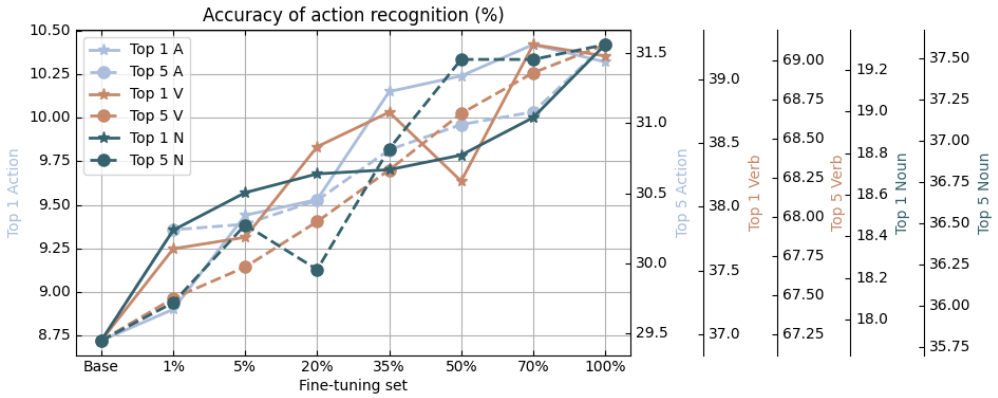


Figure 12: **The accuracy curve of top-1 and top-5 action (A), noun (N) and verb (V) in TurkeySandwich.**

Table 16: **The top-1 and top-5 accuracy of action (A), noun (N) and verb (V) for action recognition of ContinentalBreakfast.** ↑ denotes the absolute maximum increase between the highest and base. the relative maximum increase w.r.t base is shown in the first row.

Fine-tuning Set	Top 1			Top 5		
	A	N	V	A	N	V
Relative (↑)	13.03%	13.38%	3.81%	7.54%	6.05%	2.64%
100%	**10.84** (↑1.3)	**17.46** (↑2.1)	**38.72** (↑1.4)	31.69	**36.48** (↑2.1)	67.43
70%	10.59	17.12	38.61	**31.93** (↑2.2)	**36.48** (↑2.1)	67.41
50%	10.59	16.38	38.57	31.85	35.77	**67.74** (↑1.7)
35%	10.42	16.46	38.38	31.18	35.68	67.16
20%	10.09	16.05	38.21	30.77	34.82	66.97
5%	9.93	15.96	38.38	30.52	34.74	66.83
1%	9.76	15.63	37.71	30.02	34.65	66.75
base	**9.59**	**15.40**	**37.30**	**29.69**	**34.40**	**66.00**

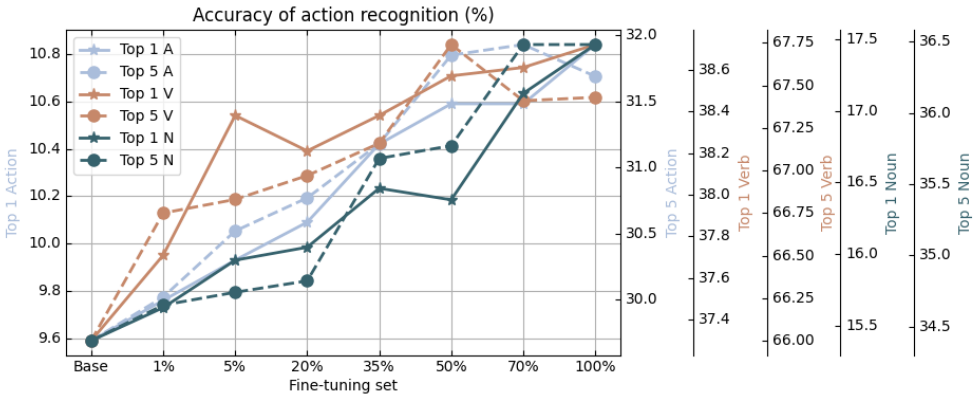


Figure 13: **The accuracy curve of top-1 and top-5 action (A), noun (N) and verb (V) in ContinentalBreakfast.**

Table 17: **The top-1 and top-5 accuracy of action (A), noun (N) and verb (V) for action recognition of PastaSalad.** ↑ denotes the absolute maximum increase between the highest and base. the relative maximum increase w.r.t base is shown in the first row.

Fine-tuning Set	Top 1			Top 5		
	A	N	V	A	N	V
Relative (↑)	14.42%	8.76%	6.00%	8.42%	6.77%	1.65%
100%	11.76	**20.12** (↑1.6)	44.77	**38.10** (↑3.0)	**45.55** (↑2.9)	**72.67** (↑1.2)
70%	**11.90** (↑1.5)	19.91	**45.20** (↑2.6)	37.81	45.53	72.23
50%	11.59	19.62	45.02	37.54	45.31	72.27
35%	11.74	19.69	45.02	37.56	45.24	72.03
20%	11.70	19.58	44.88	37.52	45.08	71.98
5%	11.32	19.40	44.59	36.54	44.87	72.01
1%	10.74	19.02	43.84	37.05	44.36	71.56
base	**10.40**	**18.50**	**42.64**	**35.14**	**42.66**	**71.49**

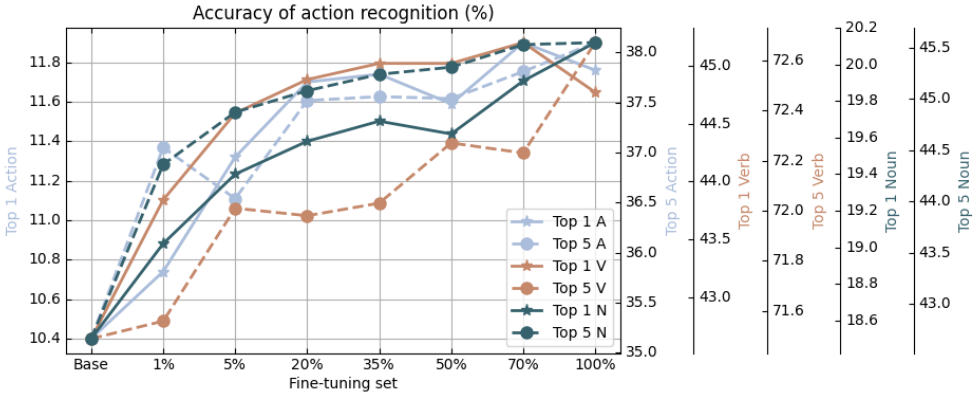


Figure 14: **The accuracy curve of top-1 and top-5 action (A), noun (N) and verb (V) in PastaSalad.**

C DETAILS OF FINE-TUNING OF THE HAND-OBJECT DETECTOR

For the fine-tuning of the hand-object detector, we attempt to adapt the base one to a new environment. We conduct extensive experiments to quantify and visualize the performance of the generated boxes. In the experiment, we adopt different epochs, freeze different number of layers, and use different fine-tuning sets. We finally find the object detector performs the best with the epoch set to 7, all the layers unfrozen, the and the learning rate (lr) set to 10^{-3} with a decay of 10^{-4}.

C.1 QUALITATIVE RESULTS FOR THE FINE-TUNING OF THE HAND-OBJECT DETECTOR.

Figure 15 shows qualitative results of the fine-tuning of the object detector, where the first two rows are images randomly selected from test split of kitchen 22 in EK and the last two rows are the ones in recipe Pizza in EGTEA. We can see that with the increase of the fine-tuning epoch, the accuracy of the boxes increases gradually.

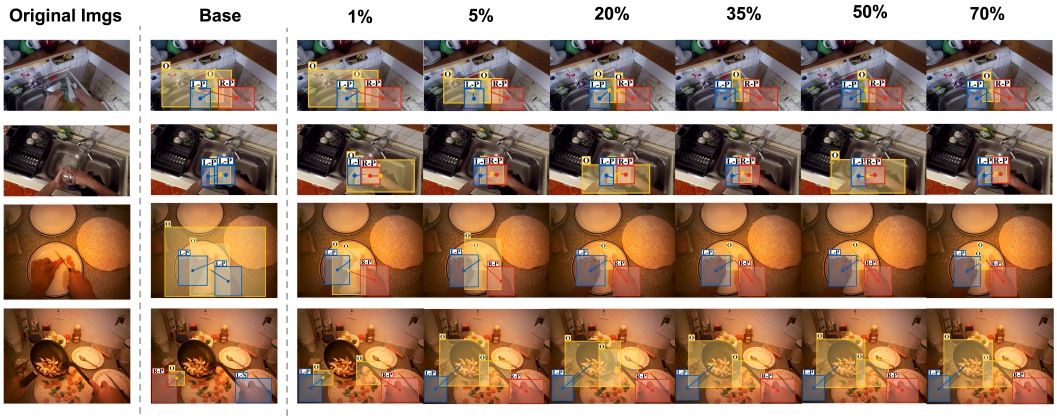


Figure 15: **Qualitative results of fine-tuning of the object detectors.** The first two rows are frames in kitchen 22 of EK, and the last two rows are examples in EGTEA of the recipe of Pizza. We show the bounding boxes predicted by the detector. Each box has either object ("O"), left-hand ("L") and right-hand ("R") category. "P" indicates contact between a hand and an object.

Table 18: **Average Precision (AP) of the fine-tuning of different sets for different new kitchens environment in EK.** AP_{obj} is the AP of the target object, AP_h is the AP of the hand, and AP_m is the mean of the two.

Kitchen id	AP	# FT frames						
		Base	1%	5%	20%	35%	50%	70%
4	AP_{obj}	0.405	0.450	0.527	0.569	0.662	0.683	**0.695**
	AP_h	0.907	0.907	0.907	0.907	0.907	0.906	0.907
	AP_m	0.656	0.679	0.717	0.738	0.784	0.795	**0.801**
8	AP_{obj}	0.460	0.475	0.530	0.618	0.648	0.652	**0.664**
	AP_h	0.908	0.908	0.908	0.908	0.909	0.908	0.908
	AP_m	0.684	0.692	0.719	0.763	0.778	0.780	**0.786**
22	AP_{obj}	0.417	0.490	0.555	0.618	0.662	0.674	**0.677**
	AP_h	0.901	0.901	0.903	0.904	0.906	0.905	0.906
	AP_m	0.659	0.695	0.729	0.761	0.784	0.789	**0.791**
24 (video-based split)	AP_{obj}	0.373	0.458	0.549	0.575	0.649	0.649	**0.657**
	AP_h	0.906	0.902	0.904	0.906	0.906	0.906	0.906
	AP_m	0.639	0.680	0.727	0.741	0.778	0.777	**0.782**

C.2 QUANTITATIVE RESULTS FOR THE FINE-TUNING OF THE OBJECT DETECTOR.

In order to quantitatively evaluate the performance of the fine-tuning of the object detector, we test the fine-tuned object detector on the rest 30% in 1 and 3.

We use the Average Precision (AP) to evaluate the fine-tuning of the object detector. The AP_{obj}, AP_h, AP_m denotes the AP for the hand-interacted object, hands and the mean of them, respectively. The results are shown in 18 and 19.

Table 19: **Average Precision (AP) of the fine-tuning of different sets for the different out-of-domain recipes.** AP_{obj} is the AP of the target object, AP_h is the AP of the hand, and AP_m is the mean of the two.

Recipes	AP	# FT frames						
		Base	1%	5%	20%	35%	50%	70%
BaconAndEggs	AP_{obj}	0.429	0.588	0.729	0.769	**0.788**	0.782	0.786
	AP_h	0.899	0.904	0.906	0.905	0.907	0.907	0.907
	AP_m	0.664	0.746	0.817	0.837	**0.847**	0.844	**0.847**
Cheeseburger	AP_{obj}	0.408	0.501	0.630	0.732	0.756	0.768	**0.785**
	AP_h	0.895	0.901	0.903	0.902	0.905	0.905	0.903
	AP_m	0.652	0.701	0.767	0.817	0.831	0.837	**0.844**
ContinentalBreakfast	AP_{obj}	0.586	0.611	0.666	0.752	0.611	0.765	**0.769**
	AP_h	0.903	0.893	0.901	0.901	0.893	0.902	0.903
	AP_m	0.744	0.752	0.783	0.827	0.752	0.833	**0.836**
GreekSalad	AP_{obj}	0.350	0.460	0.536	0.638	0.659	0.667	**0.729**
	AP_h	0.815	0.902	0.906	0.907	0.907	0.906	0.906
	AP_m	0.583	0.681	0.721	0.772	0.783	0.787	**0.818**
PastaSalad	AP_{obj}	0.307	0.470	0.597	0.668	0.677	0.684	**0.687**
	AP_h	0.815	0.903	0.904	0.906	0.907	0.907	0.906
	AP_m	0.561	0.687	0.751	0.787	0.792	0.795	**0.797**
Pizza	AP_{obj}	0.376	0.354	0.600	0.680	0.761	**0.787**	**0.787**
	AP_h	0.906	0.906	0.907	0.908	0.908	0.908	0.908
	AP_m	0.641	0.630	0.754	0.794	0.835	**0.847**	**0.847**
TurkeySandwich	AP_{obj}	0.438	0.549	0.608	0.715	0.744	0.744	**0.767**
	AP_h	0.817	0.815	0.907	0.907	0.906	0.907	0.907
	AP_m	0.627	0.682	0.758	0.811	0.825	0.825	**0.837**

C.3 FACTORS AFFECTING THE FINE-TUNING PERFORMANCE

In our exploration of fine-tuning the object detector, a set of factors play a role in its final performance. Here we show experiment results to analyze these factors.

Fine-tuning Epochs. The first factor is the number of epochs when fine-tuning the object detector. Figure 16 shows the AP of the hand-interacted object versus the epoch, different color represents different fine-tuning set. The curve indicates that the increase of the average precision with respect to the fine-tuning epoch slows down gradually, then saturates around the 7^{th} epoch, after which the box fine-tuning process becomes over-fitted. Therefore, we finally set the total number of the fine-tuning epoch to 7.

Fine-tuning Sets. Another factor that affects the fine-tuning is the number of frames in the fine-tuning set. Intuitively, when the number of the frames increases, the object detector should be able to better localize the object in a given image, in other words, the object detector can better recognize the object if we give it a glance of the specific object in the new environment. Figure 17 shows the curve of AP versus different fine-tuning sets.

It presents a unified trend that the AP for the hand-interacted object increases with the number of frames in fine-tuning set. But the increase gradually slows down after a point, for the object detector

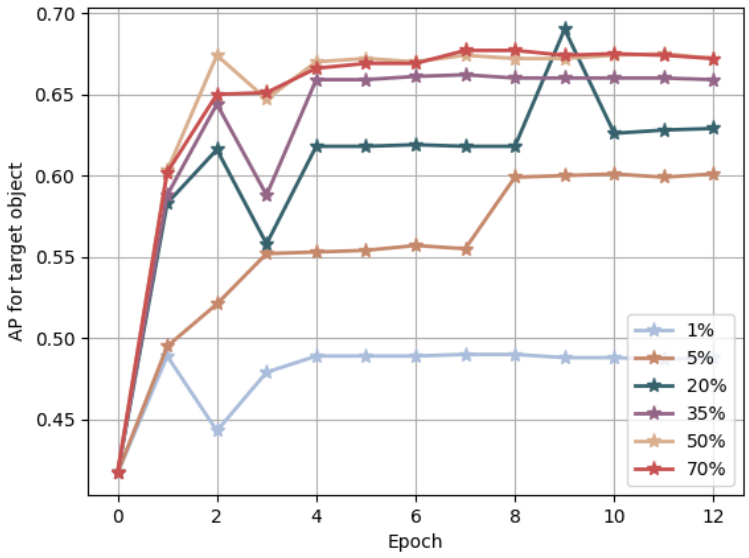
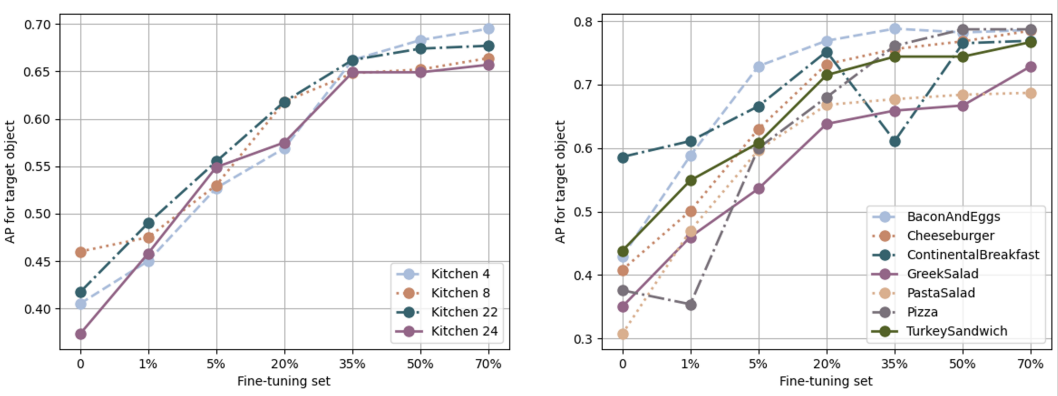


Figure 16: **AP of the object detectors with different fine-tuning epochs of kitchen 22 in EK.**



(A) (B)

Figure 17: **AP of the object detectors with different fine-tuning sets.**. (A) displays the change of AP with respect to fine-tuning set for hand-interacted object in EK, (B) shows the one in EGTEA.

has already seen most of the unique objects in the new environment and will not have further obvious improvement.

D OBJECT-LEVEL INFORMATION OF KITCHEN 24 IN EK.

D.1 OBJECT-BASED FRAME DISTRIBUTION

Since the annotated frames used in our fine-tuning set only include bounding boxes of "left hand", "right hand" and "hand-interacted object", but not the specific class of the object bounding box, we re-labeled the class of each bounding box of the annotated frames in kitchen 24 in EK to have better leverage the object-level information provided by the boxes.

First, we visualize all the unique hand-interacted objects in Figure 18 and have some statistics about their distribution in kitchen 24 in EK as shown in Figure 19,

Figure 18: **The object-based distribution of fine-tuning set of kitchen 24.**

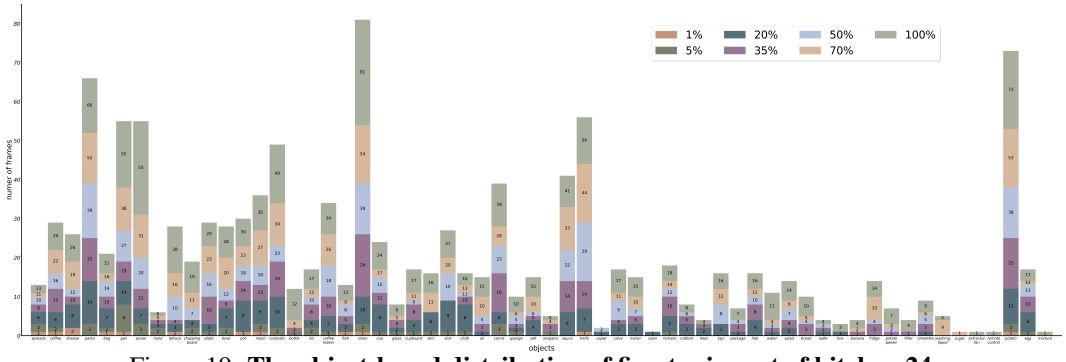


Figure 19: **The object-based distribution of fine-tuning set of kitchen 24.**

As shown in Figure 19, we visualize the 60 kinds of unique objects that appear in the annotated frames of kitchen 24 in EK. We also analyze the the distribution of these frames (grouped by object) in our video-based split fine-tuning set. Different color denotes different fine-tuning set, and the number on it denotes the absolute number of frames including that object in the corresponding fine-tuning set. For example, in our video-based split of kitchen 24 in EK, there are totally 55 frames involved with "pan", 1% fine-tuning set includes 1 of them, 5% fine-tuning set includes 8 of them, 20% fine-tuning set includes 14 of them, 35% fine-tuning set includes 19 of them, 50% fine-tuning set includes 27 of them, while 70% fine-tuning set includes 38 of them.

D.2 OBJECT-BASED SEGMENT DISTRIBUTION

As show in Table 20, we also have statistics about the number of video segments of kitchen 24 in EK based on object. For example, there are totally 69 unique object that appear in the video sequences of kitchen 24 in EK, 57 segments in these sequences are involved with fridge.

Table 20: **Object-based segment distribution in kitchen 24 of EK.** There are total 69 unique objects in kitchen 24. **Object** denotes each object and **# Segments** count the number of video segment involved with that object.

Object	# Segments	Object	# Segments	Object	# Segments
fridge	57	tap	78	colander	40
melon	6	biscuit	2	bread	13
cupboard	111	heat	5	spreads	18
skin	20	oven	5	onion	56
banana	5	spoon	85	lid	49
drawer	6	meat	66	caper	5
coffee maker	31	oregano	19	sauce	19
plate	60	salt	30	chopping board	36
filter	2	extractor fan	2	pot	19
cloth	19	salad	22	sugar	3
coffee	18	pan	78	pasta	66
hand	32	water	15	tomato	42
bag	40	glass	24	top	1
sink	23	bottle	47	bottle opener	2
knife	77	oil	15	ladle	6
hob	24	bowl	33	lettuce	22
fork	51	box	25	carrot	30
remote control	2	seed	3	olive	18
cheese	32	can	2	potato	31
freezer	4	egg	16	breadcrumb	5
washing liquid	7	paper	6	potato peeler	1
tv	2	rubbish	1	package	14
cup	58	sponge	35	mixture	1

E THE IMAGE-TO-SEGMENT MAPPING STRATEGY

In our comparison experiment, we have adopted fine-tuning and linear probing as the traditional methods. Different from what most research have covered, image-based domain adaptation, our task is targeted on video-based action recognition thus takes video segments as input. Therefore, we should first find a mapping scheme between each image we use in the fine-tuning phase and the corresponding video segment used in the action recognition phase. What we do here is we realize the mapping by associating the image with the video segment it belongs to in the raw dataset. There are also some edge cases: 1) more than one images in the fine-tuning set belong to the same video segment; 2) some images belong to no segment, which are just the frames "between" different video segments. For these two cases. we just keep them as what they are. Table 21 shows the statistics of the mapping of kitchen 24 in EK and recipe GreekSalad in EGTEA.

Table 21: The mapping details between images and corresponding video segment.

Dataset	Data Format	Fine-tuning Set						
		1%	5%	20%	35%	50%	70%	100%
Kitchen 24 (EK)	Image	61	92	259	441	602	847	1,174
	Segment	44	65	172	256	322	423	538
GreekSalad (EGTEA)	Image	17	79	315	549	786	1,100	1,572
	Segment	14	41	137	220	290	359	449

F EXISTED DOMAIN GAPS BETWEEN THE TWO SCENARIOS

As two different datasets, domain gaps exist in many aspects, here we have some detailed data to show some of them in Table 22, we have statistics about the overlapping of Verb, Noun, Action, average frame number of each video segment in EK and EGTEA. The number in the Table denotes the absolute value, for example, in EK, there are totally 97 kind of verb class, while in EGTEA, there are 19 kinds of verb class, and 19/19 them are overlapped with the ones in EK.

Table 22: **Domain gap between EpicKitchens and EGTEA.**

	Verb	Noun	Action (Verb + Noun)	Average numbers of frame / video segment
EpicKitchens	97	300	3028	179
EGTEA	19	53	282	87
Overlap	19/19	53/53	211/282	/

G ADAPTATION OF THE ADAPTION VIA OBJECTS/OR ACTIONS

In Section 3.4.2 we described different adaptaion schemes, e.g., by adapting via actions and/or objects. Here, we provide additional implementation details and detailed results regarding this experiment.

For fine-tuning, we unfreeze all the layers of action recognition model (pre-trained on source domain) and train it for another 7 epochs with the $lr = 8 \times 10^{-6}$. For linear probing, we take the same pre-trained model and freeze all the layers except the two functional heads (predictions for verb and noun), and train for 7 epochs with the $lr = 8 \times 10^{-6}$. To enable direct comparison of these approaches to our frame-wise supervision, we obtain a set of the target environment video segments that are annotated with action labels by our image-to-segment mapping scheme. Basically in which we map a specific frame to the raw video segment it belongs to in the original videos with the corresponding (verb, noun) action annotations (the details of this scheme refer to E). Then we use these obtained video segments to form the new fine-tuning sets to adapt the action recognition model.

In addition, we show the details of Memory, Time and Accuracy of action recognition task of different fine-tuning sets (1%, 5%, 20%, 35%, 50%, 70% and 100%) for Kitchen 24 of EK as shown in Table 23 and recipe GreekSalad as shown in EGTEA in Table 24.

Table 23: **The comparison between the proposed box supervision method and the traditional domain adaptation methods in kitchen 24 of the EK dataset.**

Method	Memory /MiB	Fine-tuning Set	Time /s	Accuracy of the action recognition					
				Top 1 Action	Top 5 Action	Top 1 Noun	Top 5 Noun	Top 1 Verb	Top 5 Verb
Base	/	/	/	24.33	53.33	36.70	62.47	51.00	80.27
Fine-tuning	8775	1%	938	24.88	53.16	36.18	60.82	52.61	82.62
		5%	1810	24.80	54.74	36.73	61.77	51.90	83.02
		20%	4904	25.07	55.42	36.91	62.90	53.42	83.99
		35%	7297	25.86	55.64	37.13	63.40	53.27	84.49
		50%	9186	26.57	55.74	37.92	63.98	53.98	84.81
		70%	11526	**26.67**	56.69	38.65	64.30	**54.58**	**85.52**
		100%	15562	26.30	**58.37**	**38.86**	**65.01**	54.34	85.39
Linear Probing	3329	1%	599	24.88	52.84	36.18	60.74	52.84	81.83
		5%	1304	24.96	52.84	36.26	60.66	52.84	81.91
		20%	3563	25.04	53.00	36.26	60.98	52.84	81.99
		35%	5378	25.04	53.32	36.41	61.30	52.84	82.07
		50%	6536	25.12	53.48	36.57	61.45	52.76	82.23
		70%	8406	**25.20**	53.55	**36.49**	61.61	**52.92**	82.23
		100%	10723	25.12	**53.95**	**36.49**	**62.01**	52.84	**82.39**
Ours	3027	1%	75	23.78	54.74	37.02	63.40	52.69	82.46
		5%	84	24.33	55.06	37.26	63.48	53.00	82.49
		20%	295.98	24.93	55.77	37.65	63.19	53.29	82.94
		35%	499	25.17	55.53	38.33	63.80	53.49	82.99
		50%	702	25.80	56.08	38.33	64.35	53.68	83.18
		70%	1720	**26.80**	56.64	**38.49**	**64.90**	**53.69**	83.18
		100%	2293	25.72	**56.77**	37.97	64.88	53.55	**83.33**
Ours + Fine-tuning	8775	1%	1013	24.30	54.82	35.86	62.32	52.45	82.46
		5%	1895	24.70	57.35	37.12	64.77	51.03	83.73
		20%	5200	25.51	60.43	38.55	66.43	54.27	86.26
		35%	7797	26.30	61.61	39.65	67.61	53.79	86.97
		50%	9888	27.73	61.77	40.84	68.25	**56.08**	87.05
		70%	13246	**28.36**	**63.67**	**42.26**	**70.30**	**56.08**	**87.44**
		100%	17856	27.17	60.51	40.13	66.82	55.06	86.41

Table 24: **The comparison between the proposed box supervision method and the traditional domain adaptation methods for recipe GreekSalad in EGTEA.**

Method	Memory /MiB	Fine-tuning Set	Time /s	Accuracy of the action recognition					
				Top 1 Action	Top 5 Action	Top 1 Noun	Top 5 Noun	Top 1 Verb	Top 5 Verb
Base	/	/	/	10.21	33.33	19.95	39.27	41.01	73.66
Fine-tuning	8775	1%	1620	11.04	33.29	21.74	39.43	41.28	73.59
		5%	7020	11.65	34.77	22.10	40.15	42.14	74.45
		20%	8066	11.25	34.93	22.88	40.54	42.49	74.24
		35%	12013	11.85	34.96	22.12	40.70	42.21	74.76
		50%	31506	12.20	34.93	22.70	40.79	42.09	75.01
		70%	17967	12.66	35.12	**23.66**	41.74	**42.56**	**75.75**
		100%	24312	**13.14**	**35.87**	23.59	**41.77**	42.38	75.68
Linear Probing	3329	1%	2742	11.06	31.57	20.02	37.35	39.80	73.22
		5%	5366	11.06	31.57	20.02	37.36	39.93	73.22
		20%	15074	11.08	31.70	20.15	37.35	40.05	73.34
		35%	25272	11.08	31.87	20.15	37.42	40.05	73.46
		50%	28217	11.16	31.88	20.25	37.44	40.05	73.46
		70%	38441	11.15	31.87	20.25	**37.47**	40.17	**73.52**
		100%	48153	**11.18**	**31.94**	**20.27**	**37.47**	**40.29**	73.46
Ours	3027	1%	39	11.27	33.42	20.52	39.31	41.42	72.60
		5%	144	11.69	34.03	21.02	39.56	41.82	72.97
		20%	537	11.43	34.05	21.88	39.70	42.42	73.83
		35%	930	11.92	34.92	22.01	39.57	42.37	73.83
		50%	1308	12.53	35.17	22.50	40.21	42.54	73.96
		70%	1845	12.87	35.05	22.50	40.51	42.68	**74.78**
		100%	2578	**13.16**	**35.42**	**22.62**	**41.03**	**42.91**	74.21
Ours + Fine-tuning	8775	1%	1659	12.16	34.03	21.38	40.91	41.65	72.97
		5%	7164	12.29	36.00	22.24	42.38	41.40	73.83
		20%	8603	14.37	40.79	25.92	48.28	44.23	75.55
		35%	12943	17.57	46.31	29.48	53.81	46.56	77.89
		50%	32814	17.69	47.67	30.96	55.28	46.68	78.26
		70%	19811	19.90	49.51	33.54	57.25	**49.14**	78.87
		100%	26891	**21.25**	**52.21**	**36.73**	**59.71**	49.02	**79.85**