Interactive Prototype Learning for Egocentric Action Recognition

Xiaohan Wang[1,2] Linchao Zhu[3] Heng Wang[4] Yi Yang[1]
[1]CCAI, Zhejiang University [2]Baidu Research
[3]ReLER, University of Technology Sydney [4]Facebook AI Research
wxh1996111@gmail.com linchao.zhu@uts.edu.au hengwang00@gmail.com yangyics@zju.edu.cn

Abstract

Egocentric video recognition is a challenging task that requires to identify both the actor's motion and the active object that the actor interacts with. Recognizing the active object is particularly hard due to the cluttered background with distracting objects, the frequent field of view changes, severe occlusion, etc. To improve the active object classification, most existing methods use object detectors or human gaze information, which are computationally expensive or require labor-intensive annotations. To avoid these additional costs, we propose an end-to-end ***I****nteractive* ***P****rototype* ***L****earning (IPL) framework to learn better active object representations by leveraging the motion cues from the actor. First, we introduce a set of verb prototypes to disentangle active object features from distracting object features. Each prototype corresponds to a primary motion pattern of an egocentric action, offering a distinctive supervision signal for active object feature learning. Second, we design two interactive operations to enable the extraction of active object features,* i.e.*, noun-to-verb assignment and verb-to-noun selection. These operations are parameter-efficient and can learn judicious location-aware features on top of 3D CNN backbones. We demonstrate that the IPL framework can generalize to different backbones and outperform the state-of-the-art on three large-scale egocentric video datasets,* i.e.*, EPIC-KITCHENS-55, EPIC-KITCHENS-100 and EGTEA.*

1. Introduction

Egocentric videos have become popular on social media and have attracted increasingly more attention in computer vision since the introduction of datasets, such as EGTEA [21], Charades-Ego [32], EPIC-KITCHENS [6, 5, 7]. Unlike third-person videos where actions usually happen at a distance, egocentric videos focus on person and object interactions at a closer look. Understanding egocentric videos requires to identify both the motion from the actor and the object that the actor interacts with. Recent egocentric video datasets [5, 7] are usually constructed by decomposing an action into a combination of a *verb* and a *noun*, where action recognition can be achieved by classifying the associated *verb* and *noun*. For instance, "cut potato" is divided into a verb "cut" and a noun "potato". Such a formulation helps to distinguish the subtle semantic differences among actions.

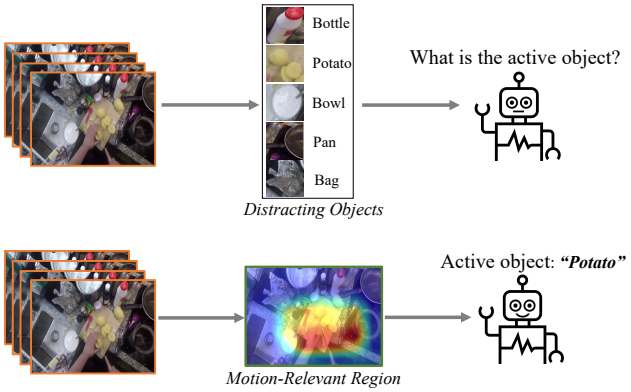

Figure 1. The motivation of **I**nteractive **P**rototype **L**earning (IPL) framework. The *noun* classification is difficult as the active object can be surrounded by a considerable number of distracting objects. Our framework aims to collaboratively learn judicious motion-relevant spatio-temporal features for more accurate *noun* (active object) classification.

Egocentric videos focus on domain-specific fine-grained actions, while the existing third-person datasets [18] are more generic and collected from various domains, like sports and daily activities. In egocentric videos, the background scene is often similar among different actions. For instance, "cutting carrot" and "peeling potato" can both happen in the same kitchen scene. Hence, the usefulness of scene context information is limited in egocentric videos, making the recognition task more challenging.

Besides the aforementioned challenges, *noun* classification is particularly difficult as the active object [9, 11] involved in the action can be surrounded by a considerable number of distracting objects, *e.g.*, the bowls and pans surrounding the active object "potato" in Fig. 1. Indeed, *noun*

classification tends to have much lower accuracy than *verb* in egocentric video datatsets [6, 40], and is the bottleneck of the whole action recognition system.

Previous methods either use off-the-shelf object detectors [40, 42] or human gaze provided by the datasets [21] as additional cues to improve *noun* recognition. However, running object detectors on high-resolution video frames is computationally expensive, and human annotations are not always available. In this paper, we propose to improve active object recognition in egocentric videos by leveraging the information learned from the actor's motion. The active objects often locate in areas where the actors perform the action. Moreover, the actor's motion carries the intent of the actor and is often the dominant signal in the egocentric videos, which can serve as a reliable supervision to improve active object recognition.

We devise an end-to-end **I**nteractive **P**rototype **L**earning (IPL) framework for joint *verb* and *noun* classification (Fig. 1). IPL learns verb prototypes using the supervision from verb labels, and each verb prototype encodes the motion pattern of a verb class. The learned verb prototypes are used to guide *noun* classification through disentangling active object features from distracting object features. This is achieved by two interactive operations, *i.e.*, *noun*-to-*verb* assignment and *verb*-to-*noun* selection. The two operations collaboratively extract judicious location-aware spatio-temporal features for *noun* classification. The *noun*-to-*verb* assignment aims to aggregate the features based on their similarities to the verb prototypes. In the *verb*-to-*noun* selection, we choose the most action-relevant features for the *noun* classification.

Some components of IPL share the same spirit as NetVLAD [1], but there are a few key differences. First, our prototypes are learned with direct supervision from verb annotations. Each prototype corresponds to each verb class, whereas the semantic meaning of the NetVLAD clusters is unclear. Second, our prototypes are shared by *verb* and *noun* classification in a multi-task setting, where the harder task (*i.e.*, *noun* classification) can benefit from the information learned from *verb* classification. Third, instead of concatenating the features from all clusters like NetVLAD, we propose a *verb*-to-*noun* selection mechanism to identify discriminative features from active objects.

With extensive experiments and detailed ablation studies, we demonstrate that IPL outperforms the state of the art on three large-scale egocentric video dataset, and is able to generalize to different video backbones [3, 38]. To summarize, we made the following major contributions:

- Propose to leverage the information learned from recognizing the actor's motion to improve active object classification, which is currently the bottleneck of egocentric video recognition.
- Design the IPL framework which allows better information flow between the *verb* and *noun* classification task by sharing the same set of feature prototypes.
- IPL shows superior results on three egocentric datasets, *i.e.*, EPIC-KITCHENS-100 [7], EPIC-KITCHENS-55 [6] and EGTEA [21], without the additional cost of object detection and human gaze annotations.

2. Related Work

Video Architectures. Early architectures [33, 8, 39] for video classification are usually based on 2D convolution adopted from the image domain. 2D convolutions are still widely used for efficient video recognition, such as TRN [46], TSM [24], ECO [47], *etc*. On the other hand, 3D convolutions [37] have gained popularity due to their spatio-temporal modeling capacity. I3D [3] initializes 3D CNNs with inflated weights from 2D CNNs pretrained on ImageNet [20]. S3D [44], R(2+1)D [38] and P3D [28] propose to decompose 3D convolutions to 2D spatial convolutions and 1D temporal convolutions. SlowFast [10] is another recent example of video architectures. These popular video backbones are designed for general video classification tasks and do not take into account the challenges of egocentric videos. Though IPL is built on existing video backbones, we focus on designing a framework that can improve the accuracy of recognizing active objects in egocentric videos.

Action Recognition in Egocentric Videos. A number of existing methods leveraged object detection to improve egocentric video recognition [40, 41, 42, 30, 26], among which [42, 30] also incorporate temporal contexts to help understanding of the ongoing action. These methods require labor-intensive object detection annotations and are computationally expensive, which may limit their applications in real-world systems. In contrast, our framework only uses the existing action labels as supervision and does not depend on costly object detectors. Recently, Shan *et al.* [31] developed a hand-object detector to locate the active object. When the detector is well-trained, it can be directly deployed on the target dataset without finetuning. However, running the detector on high resolution frames is still much more expensive than our method.

Sudhakara *et al.* [35] proposed a two-stage Long-Short Term Attention RNN model to track discriminative areas and locate active objects. Li *et al.* [21] and Liu *et al.* [25] leveraged the gaze annotations to guide deep models to focus on the interacting area and select informative features. TBN [19] fuse multi-modal information (such as optical flow and audio) to improve egocentric action recognition. Compared to these methods, IPL leverages the motion cues

learned from *verb* classification to select discriminative features for active object recognition.

Feature Aggregation. Our method is also related to the feature aggregation method such as VLAD [17] and Fisher Vectors [29]. NetVLAD [1] converts VLAD into an differentiable layer for end-to-end training. These feature aggregation methods have been applied to video recognition and achieved promising results [45, 14, 27]. Besides feature aggregation, IPL is also designed to learn prototypes shared between the *verb* and *noun* classification, and prototypes are trained with direct supervision from verb labels. This enables our prototypes to encode the motion feature of each verb class and provide information to select discriminative features to improve noun classification.

3. Interactive Prototype Learning

3.1. Overview

Given an input video clip $\boldsymbol{X}$, the goal is to classify it into M verb classes and N noun classes. The underlying action can be inferred from the verb and noun prediction results. As shown in Figure 2, we first extract the spatio-temporal feature map $\boldsymbol{\phi}_\theta(\boldsymbol{X}) \in \mathbb{R}^{T\times H\times W\times C}$ from the last convolutional layer of the 3D CNN backbone $\boldsymbol{\phi}_\theta$, where θ is the parameters of the CNN, T is the temporal length, C is the number of channels and $H \times W$ is the spatial resolution.

The core idea of IPL is to utilize verb features to guide the learning of action-centric object features. Specifically, we introduce M *verb prototypes* $\boldsymbol{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M\} \in \mathbb{R}^{M\times C}$, where each prototype corresponds to a verb class and represents one type of motion from the actor. All M prototypes are shared between the verb and noun branches in order to enable interactive learning.

In the verb branch, we obtain the C-dimensional verb feature vector by applying global average pooling on $\boldsymbol{\phi}_\theta(\boldsymbol{X})$ (Section 3.2). Unlike the conventional linear classifier implemented by a fully connected layer, we use a simple nearest neighbor classifier with a cosine similarity [13] between the verb feature and M verb prototypes. This simple strategy allows us to directly enforce the strong supervision from the verb ground-truth to learn more semantic verb prototypes.

In noun classification, we design two interactive operators to extract location-aware features from $\boldsymbol{\phi}_\theta(\boldsymbol{X})$ to perform the noun classification. In the *noun*-to-*verb* assignment operator (Section 3.3.1), we decompose $\boldsymbol{\phi}_\theta(\boldsymbol{X})$ into THW C-dimensional features, and assign each feature to M verb prototypes and one additional background prototype to catch irrelevant background information. This converts the THW features into $M+1$ feature groups. In the *verb*-to-*noun* selection operator (Section 3.3.2), we select K feature groups corresponding to the top-K verb classes with the highest classification score from the verb branch. The selected K features are then aggregated to obtain the final representation for noun classification.

3.2. Verb Classification

Verb Prototype. In egocentric videos, motion is the dominant information for action recognition and indicates the intention of the actor and which object the actor wants to interact with [6, 7]. This motivates us to leverage the actor motion information to improve active object recognition, which is an arguably harder task. Specifically, we propose to learn a prototype for each verb class. We denote the *verb prototypes* as $\boldsymbol{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_M\}$, where $\boldsymbol{P} \in \mathbb{R}^{M\times C}$. These verb prototypes are intermediate representations to facilitate interaction between the verb and noun classification. They are anchors for grouping spatio-temporal features based on their similarities.

Cosine Classifier. Inspired by recent works [4, 13], We design a simple and effective classifier: Nearest-Neighbors (NN) on top of ℓ_2-normalized features, and named as cosine classifier. Given the spatio-temporal feature map $\boldsymbol{\phi}_\theta(\boldsymbol{X})$, the verb feature is generated with global average pooling (GAP):

$$\boldsymbol{v} = GAP(\boldsymbol{\phi}_\theta(\boldsymbol{X})), \tag{1}$$

where $\boldsymbol{v} \in \mathbb{R}^{1\times C}$. After that, we calculate the cosine similarity between the verb feature and each verb prototype. The verb classification probability q_i for the i-th class is generated using a softmax activation function. Formally,

$$q_i = \frac{exp(\bar{\boldsymbol{v}}\bar{\boldsymbol{p}}_i^{\mathsf{T}}/\tau)}{\sum_{j=1}^{M} exp(\bar{\boldsymbol{v}}\bar{\boldsymbol{p}}_j^{\mathsf{T}}/\tau)}, \tag{2}$$

where $\bar{\boldsymbol{v}} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}$ and $\bar{\boldsymbol{p}}_i = \frac{\boldsymbol{p}_i}{\|\boldsymbol{p}_i\|}$ are the l_2-normalized vectors. Here we use a temperature τ to re-scale the similarities following [13, 4]. The temperature τ can help training similarity-based classifier and reduce intra-class variations [4], which is beneficial for learning discriminative video representations.

3.3. Noun Classification

3.3.1 Feature Assignment and Grouping

In egocentric videos, the motion from the actor gives strong indications about what object the actor interacts with. This inspires us to leverage the motion features to identify the features from the active objects and suppress the features from distracting objects. We design new operators that can decompose and regroup object features based on their relevance to the actor motion and learn more discriminative features for active object classification.

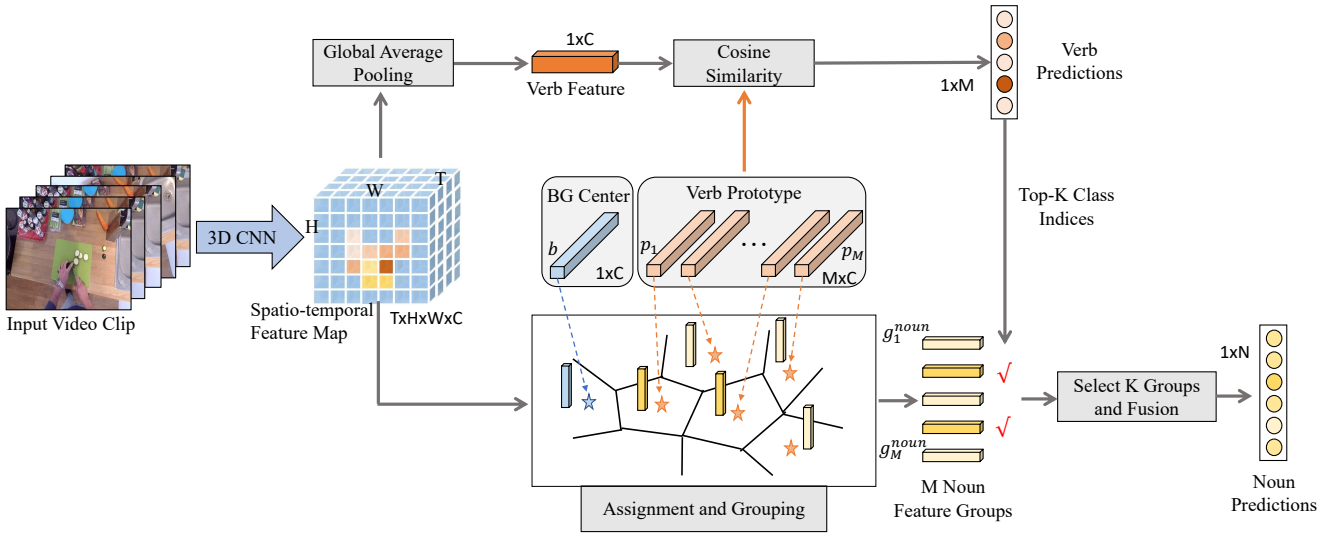

Figure 2. Our Interactive Prototype Learning (IPL) framework. The feature map of size $T \times H \times W \times C$ is extracted from the last convolutional layer of the 3D CNN backbone. To facilitate the interaction between the verb branch and the noun branch, we introduce a set of *verb prototypes* shared across the two branches. A background prototype is introduced to filter the action-irrelevant information from the spatio-temporal feature map. Each prototype is a C-dimensional vector and is randomly initialized during training. Verb prediction is obtained by computing the cosine similarity between the average pooled verb feature and the verb prototypes. For noun prediction, the feature map is decomposed and grouped by soft-assigning each feature to the prototypes. We select the most relevant K groups based on verb predictions to generate the final noun representation. The 3D CNN backbone and IPL are jointly trained in an end-to-end manner.

Feature Assignment. We propose to assign the THW C-dimensional features to the learned prototypes. Besides the aforementioned M verb prototypes, we also introduce a background prototype $\boldsymbol{b} \in \mathbb{R}^{1\times C}$ to catch all the irrelevant features that do not match to any of the motion patterns from the M verb classes. In total we have $M+1$ prototypes, noted as $\boldsymbol{P'} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_M, \boldsymbol{b}\}$, where $\boldsymbol{P'} \in \mathbb{R}^{(M+1)\times C}$ and each $\boldsymbol{c}_j \in \mathbb{R}^{1\times C}$ for $j = 1 \ldots M+1$. By assigning THW features to $M+1$ prototypes, we can disentangle the features of the active object from the features of the distracting objects, and select relevant features for noun classification.

We use a simple dot-product operator between the feature vector and the learned prototypes to measure their similarity. A softmax function is applied to the dot product to achieve the soft assignment of the feature to the $M+1$ prototype. For convenience, we reshape the spatio-temporal feature map $\phi_\theta(\boldsymbol{X}) \in \mathbb{R}^{T\times H\times W\times C}$ to a 2D tensor $\boldsymbol{Z} \in \mathbb{R}^{B\times C}$, where $B = T \times H \times W$. For a feature vector $\boldsymbol{z}_i$ from $\boldsymbol{Z}$, the assignment to the prototype $\boldsymbol{c}_j$ is defined as,

$$a_{i,j} = \frac{exp(\boldsymbol{z}_i \boldsymbol{c}_j^\mathsf{T})}{\sum_{k=1}^{M+1} exp(\boldsymbol{z}_i \boldsymbol{c}_k^\mathsf{T})}, \tag{3}$$

where $a_{i,j}$ is an element from the soft assignment matrix $\boldsymbol{A'} \in \mathbb{R}^{B\times(M+1)}$. We discard the assignments belong to the background prototype $\boldsymbol{b}$ from $\boldsymbol{A'}$ as they are considered irrelevant for active object recognition. We end up with a new assignment matrix $\boldsymbol{A} \in \mathbb{R}^{B\times M}$. Though the assignments to the background prototype $\boldsymbol{b}$ is removed from $\boldsymbol{A}$, all $M+1$ prototypes are coupled together in the denominator of Eq. 3. Thus the background prototype $\boldsymbol{b}$ is learned in the same way as $\boldsymbol{p}_i$ via back propagation.

Feature Grouping. We aggregate all the assigned features on each prototype to obtain M feature groups. The aggregation operation can be performed by a matrix multiplication as follows,

$$\boldsymbol{G} = \boldsymbol{A}^\mathsf{T} \boldsymbol{Z}, \tag{4}$$

where $\boldsymbol{G} \in \mathbb{R}^{M\times C}$ denotes the feature groups on the M prototypes. $\boldsymbol{g_i} \in \mathbb{R}^C$ is the i-th row of $\boldsymbol{G}$, and represents the aggregated feature belongs to prototype $\boldsymbol{p}_i$. $\boldsymbol{g_i}$ includes all the information from both the actor motion and the active object. To obtain the feature of the active object, we compute a residual between the grouped feature $\boldsymbol{g_i}$ and the verb prototype:

$$\boldsymbol{g}_i^{noun} = \boldsymbol{g_i} - \sum_{k=1}^{B} a_{k,i} \boldsymbol{p}_i, \tag{5}$$

where $a_{k,i}$ is the element in $\boldsymbol{A} \in \mathbb{R}^{B\times M}$. The normalization on $\boldsymbol{p}_i$ is for calibration so that it is on the same scale as $\boldsymbol{g_i}$. $\boldsymbol{g}_i^{noun}$ is the final noun feature w.r.t. prototype $\boldsymbol{p}_i$.

3.3.2 Group Selection and Noun Classification

After feature assignment and grouping, we obtain a set of features $\boldsymbol{G}^{noun} = \{\boldsymbol{g}_1^{noun}, \boldsymbol{g}_2^{noun}, \ldots, \boldsymbol{g}_M^{noun}\}$ corresponding to M verb prototypes. Given a trimmed video clip, we want to identify the features that are most related to the motion of the actor, and suppress the features that may come from irrelevant background or distracting objects.

Towards this end, we propose to simply select top-K features from $\{\boldsymbol{g}_1^{noun}, \boldsymbol{g}_2^{noun}, \ldots, \boldsymbol{g}_M^{noun}\}$ based on their verb classification scores. We sort M verb predictions in decreasing order. We denote the indices of the top-K classes with the highest scores as $\{i_1, i_2, \ldots, i_K\}$. Then the top-$K$ selected features are $\{\boldsymbol{g}_{i_1}^{noun}, \boldsymbol{g}_{i_2}^{noun}, \ldots, \boldsymbol{g}_{i_K}^{noun}\}$.

We apply l_2-normalization to each selected feature and concatenate them to generate the feature $\boldsymbol{n}' \in \mathbb{R}^{K \times C}$. We then use a layer $f_{\boldsymbol{w}}$ parameterized by $\boldsymbol{w}$ to enhance feature $\boldsymbol{n}'$, which also reduces its dimension from $K \times C$ to C. We obtain the final noun representation $\boldsymbol{n} = f_{\boldsymbol{w}}(\boldsymbol{n}')$, which can be directly used for classification. Similar to verb classification, we simply use a cosine classifier for noun classification to reduce intra-class variations. In our implementation, we instantiate $f_{\boldsymbol{w}}$ with an 1D convolutional layer with batch normalization [16] followed by ReLU activation. Note that the number of additional parameters introduced by $f_{\boldsymbol{w}}$ is negligible.

Relations to NetVLAD. The implementation of IPL shares similar components with the NetVLAD layer [1, 14, 27], if we consider the verb prototypes as the NetVLAD clusters. Unlike our verb prototypes, the clusters in NetVLAD are not trained with direct supervision from a classification loss. The semantic meaning of the NetVLAD clusters is unclear, as they only serve as anchors in the feature space for clustering. In contrast, our verb prototypes are directly optimized with a loss for verb classification. Each verb prototype can be considered as a representation to capture the motion feature of a verb class. Benefit from this design, our learned prototypes can be directly used for verb classification with a simple nearest neighbor classifier. Note that the verb prototypes are also used to assign features for noun classification and play the role of bridging the two tasks (*i.e.*, verb and noun classification) for egocentric video recognition. Additional supervision from noun classification further enhances the semantic meaning of the learned prototypes. Instead of concatenating the features from all the clusters in NetVLAD, we only select top-K aggregated features as we aim to disentangle the features from the active object and the features from the distracting objects.

3.4. Training and Inference

During training, we use cross-entropy loss for classification. The overall training objective is to minimize the sum of the verb classification loss and the noun classification loss. The 3D CNN backbone and the Interactive Prototypical Network are jointly optimized in an end-to-end manner. During inference, given an input video clip, the framework produces verb and noun predictions simultaneously. The action predictions are generated by combining verb and noun predictions.

4. Experiment

4.1. Dataset

EPIC-KITCHENS-55 [6] is a large-scale first-person video dataset. It consists of 55 hours of recordings capturing all daily activities in the kitchens. It contains 39,594 action segments which are annotated with 125 verb classes and 321 noun classes. We split the original training set to a new training and validation set following [2]. We report the top-1 accuracy on the validation set.

EPIC-KITCHENS-100 [7] is recently introduced. Compared to EPIC-KITCHENS-55 [6], the annotations are denser and more accurate. It consists of 100-hour videos and contains 89,979 segments of fine-grained actions, covering 97 verb classes and 300 noun classes. We report top-1 accuracy following the protocol of the original paper [7].

EGTEA [21] is a large-scale egocentric video dataset which consists of 10321 video clips annotated with 19 verb classes, 51 noun classes, and 106 action classes. We report mean class accuracy on the three Train/Val splits.

4.2. Implementation Details

We train two backbones with the proposed Interactive Prototype Learning framework, *i.e.*, I3D [3] and R(2+1)D-34 [38]. For I3D, we train both spatial and temporal streams with 64 RGB frames or optical flow as the input. The backbone is initialized using the Kinetics [3] pretrained weights. IPL is trained for 30 epochs using SGD with a momentum of 0.9 and a weight decay of 0.0005. The learning rate is initialized to 0.006 and then reduced by a factor of 10 in the last 10 epochs. The batch size is set to 32. During training, the spatial size of input clip is 224×224. Random scaling, random cropping and horizontal flipping are deployed as data augmentation. During inference, we resize the frame to 256×256 and feed them to the model without cropping. We average the predictions of 10 uniformly sampled clips as the final video-level predictions. For R(2+1)D-34 [38], we train the RGB stream using the IG-Kinetics [12] pretrained weights as initialization. The learning rate is set to 0.0004 and reduced by a factor of 10 every 9 epochs. We use SGD with a momentum of 0.9 and a weight decay of 0.0005 to train the model for 20 epochs. We use 32 frames

Method	Overall Top-1 Accuracy			Unseen Participants Top-1 Accuracy			Tail Classes Top-1 Accuracy		
	Verb	Noun	Act.	Verb	Noun	Act.	Verb	Noun	Act.
Chance [7]	10.68	1.79	0.55	9.37	1.90	0.59	0.97	0.39	0.12
TSN [39]	59.03	46.78	33.57	53.11	42.02	27.37	26.23	14.73	11.43
TRN [46]	63.28	46.16	35.28	57.54	41.36	29.68	28.17	13.98	12.18
TBN [19]	62.72	47.59	35.48	56.69	43.65	29.27	30.97	19.52	14.10
SlowFast [10]	63.79	48.55	36.81	57.66	42.55	29.27	29.65	17.11	13.45
TSM [24]	65.32	47.80	37.39	59.68	42.51	30.61	30.03	16.96	13.45
IPL I3D	65.66	49.74	38.43	59.12	45.26	32.17	32.17	**20.34**	**15.51**
IPL R(2+1)D-34	65.74	**50.45**	**39.17**	**61.22**	**46.01**	**33.70**	**33.02**	18.97	15.22

Table 1. The comparison with the state-of-the-art methods on the **EPIC-KITCHENS-100** Test set.

Method	Act@1	Verb@1	Noun@1
Chance [7]	0.51	10.42	1.70
TSN [39]	33.19	60.18	46.03
TRN [46]	35.34	65.88	45.43
TBN [19]	36.72	66.00	47.23
SlowFast [10]	38.54	65.56	50.02
TSM [24]	38.27	67.86	49.01
I3D† [3]	37.58	66.84	48.48
IPL I3D	39.87	67.82	50.87 (+**2.39**)
R(2+1)D-34† [12]	37.62	67.28	47.55
IPL R(2+1)D-34	40.98	68.61	51.24 (+**3.69**)

Table 2. The comparison with the baselines and state-of-the-arts on the **EPIC-KITCHENS-100** validation set. "†" indicates our implementation with two separate classifiers for noun and verb.

as input and the spatial size is 112×112 during training and 128×128 during testing. The same data augmentation and multi-crop testing strategy are used as I3D. In the verb-to-noun selection module, we set K to 5 for the two backbones on all datasets.

4.3. Comparison with State of the Arts

4.3.1 Results on EPIC-KITCHENS-100

We compare our Interactive Prototype Learning framework with the state-of-the-art methods on the largest egocentric video dataset EPIC-KITCHENS-100 [7]. TSN [39], TRN [46] and TSM [24] are based on 2D CNNs. All the three models employ a two-stream approach that use both RGB and optical flow. Besides RGB and optical flow streams, TBN [19] add audio as another modality as well. SlowFast [10] uses two RGB streams with different resolutions and frame rates.

We experiment the IPL framework using two popular backbones, *i.e.*, I3D [3] and R(2+1)D-34 [38]. As stated before, we use two streams (*i.e.*, RGB and optical flow) for I3D, and single stream (*i.e.*, RGB) for R(2+1)D-34. To evaluate the performance of the backbone itself, we train two separate classifiers implemented by FC layers for *verb* and *noun* classification. As shown in Table 2, the IPL framework is able to significantly boost the performance of the backbones for both I3D and R(2+1)D. IPL improves the overall top-1 accuracy of *noun* classification by 2.39% and 3.69% for I3D and R(2+1)D-34, respectively. The performance gain mainly comes from the *noun*-to-*verb* grouping and *verb*-to-*noun* selection operators. By introducing the interactions with verb prototypes, the most action-relevant features can be selected for *noun* classification. The *verb* recognition accuracy is also slightly improved, as the verb prototypes can also benefit from the interactive learning scheme. As expected, the dramatic improvements on *noun* recognition also lead to better accuracy for action recognition, *i.e.*, 2.29% for I3D (from 37.58% to 39.87%) and 3.36% for R(2+1)D-34 (from 37.62% to 40.98%). We outperform other state-of-the-art methods (*e.g.*, TSN [39], TRN [46], TSM [24] and SlowFast [10]) on overall top-1 accuracy. For instance, our IPL R(2+1)D-34 outperforms TSM by 2.71% on overall top-1 action accuracy.

We evaluate IPL on the test set by submitting our results to the competition server, as shown in Table 1. Our IPL R(2+1)D-34 outperforms all the state-of-the-art methods on noun classification and action classification on the Overall classes and the Unseen Participants split. Specifically, IPL R(2+1)D-34 achieves 2.65% gain for overall noun classification compared to TSM. IPL I3D has slightly better results on the Tail Classes, which may presumably due to the two-stream inputs can better improve the few-shot classes.

4.3.2 Results on EPIC-KITCHENS-55

Compare with the state of the art. We compare IPL with state-of-the-art 3D CNNs on the EPIC-KITCHENS-55 validation set in Table 3. Using the I3D backbone, the IPL framework gives an improvement of 1.9% for *noun* classification. Notably, using the R(2+1)D-34 backbone, IPL outperforms the baseline by 4.4% for *noun* classification. These results clearly show that IPL works effectively on

Method	Act@1	Verb@1	Noun@1
R50-NL [42]	19.0	49.8	26.1
R(2+1)D-34† [12]	22.5	56.6	32.7
SlowFast [43]	21.9	55.8	27.4
I3D [3]	23.5	59.6	31.3
IPL I3D	24.5	59.8	33.2 (+**1.9**)
R(2+1)D-34 [12]	23.6	60.5	31.1
IPL R(2+1)D-34	25.4	60.7	35.5 (+**4.4**)

Table 3. Comparison of 3D CNN backbones on the **EPIC-KITCHENS-55** validation set. "†" indicates [12] uses two R(2+1)D-34 backbones, one for verb classification and the other for noun. Our "IPL R(2+1)D-34" and "R(2+1)D-34" use a shared backbone for both tasks.

Method	Obj	Act@1	Verb@1	Noun@1	GFLOPs
LFB Max [42]	✓	22.8	52.6	31.8	6664
SAP [40]	✓	25.0	55.9	35.0	2871
IPL R(2+1)D-34	✗	**25.4**	**60.7**	**35.5**	**153**

Table 4. Compare with the state-of-the-art methods using object detection annotations on the **EPIC-KITCHENS-55** validation set.

EPIC-KITCHENS-55. With the clear gains in *noun* classification, the action classification accuracy is also improved on both backbones. For instance, there is a 1.6% gain when we compare IPL R(2+1)D-34 with its baseline.

Compare with methods using object detection annotations. As the EPIC-KITCHENS-55 dataset provides object detection annotations, a few works [42, 40] utilize these annotations for better egocentric video classification. Although object annotations can improve *noun* classification, they are also costly and not always available. Besides, SAP [41] and LFB [42] run a heavy-weight detector (ResNeXt-101-FPN) on high-resolution frames, which leads to much higher computational cost. In our paper, we do not use any additional annotations and only leverage a single CNN backbone for both noun and verb classification. As shown in Table 4, we obtain higher results compared to SAP [40] and LFB Max [42] with much lower FLOPs. This demonstrates the effectiveness and efficiency of IPL.

4.3.3 Results on EGTEA

EGTEA [21] provides gaze and hand mask annotations which have been used by the state-of-the-art methods to provide strong supervision on spatio-temporal attention. EgoIDT+Gaze [23] and I3D+Gaze [21] utilize gaze point to locate and select discriminative features. I3D (joint) [3] jointly optimizes the two-stream I3D networks. I3D+EgoConv [34] encodes head motion and hand masks, and further injects this information to a two-stream I3D model. Prob-ATT [22] also use gaze supervision to achieve high recognition results. Most existing methods on EGTEA

Methods	Mean Class Accuracy			
	Split1	Split2	Split3	Avg
EgoIDT+Gaze [23]	42.55	37.30	37.60	39.13
I3D (joint) [3]	55.76	53.14	53.55	54.15
I3D+Gaze [21]	53.74	50.30	49.63	51.22
I3D+EgoConv [34]	54.19	51.45	49.41	51.68
Ego-RNN-2S [36]	52.40	50.09	49.11	50.53
LSTA-2S [35]	53.00	-	-	-
Mutual Context-2S [15]	55.70	-	-	-
Prob-ATT [22]	56.50	53.52	53.58	54.53
Prob-ATT+Gaze [22]	57.20	53.75	54.13	55.03
I3D†	56.78	54.92	53.94	55.21
IPL I3D	**60.15**	**59.03**	**57.98**	**59.05**

Table 5. The comparison with the state-of-the-art methods on the **EGTEA** dataset. "†" indicates our implementation with two separate classifiers.

like I3D (joint) [22] and Prob-ATT [22] only use a single action classifier trained with action labels. In contrast, our methods utilizes two separate classifiers for *verb* and *noun* classification. And we train the models with both *verb* and *noun* labels. For a fair comparison, we also implement the I3D baseline using two separate classifiers. This leads to slightly better results compared to I3D (joint) as we use both *verb* and *noun* labels. As shown in Table 5, IPL I3D outperforms our strong baseline and the state-of-the-art methods by a large margin on all three splits, even though we do not utilize the gaze and hand mask annotations.

4.4. Ablation Studies

Compared to NetVLAD In IPL, the *verb* prototypes are used as anchors to assign features for *noun* classification and the *noun* features are selected based on the *verb* predictions. To investigate the effectiveness of this interactive learning scheme, we implement a baseline model that uses the NetVLAD [1] module for *noun* classification and discards the interaction between the *verb* and *noun* classification. As shown in Table 6, our IPL R(2+1)D achieves higher results on both *verb* and *noun* classification than the "R(2+1)D + NetVLAD" model, and boosts the action top-1 accuracy by 2.18%. It demonstrates that our interactive learning scheme can not only generate more discriminative representations for the *noun* classification but also enhance the *verb* prototypes to improve *verb* classification.

The verb-to-noun selection To disentangle the active object features from distracting object features, we propose to select top-K features based on the *verb* predictions. To demonstrate the effectiveness of this design, we implement a model that utilizes all feature groups for *noun* classification. As shown in Table 6, the "IPL R(2+1)D w/o Selection" model achieves 49.68% *noun* top-1 accuracy, which is lower than IPL R(2+1)D by 1.56%. Besides, without the

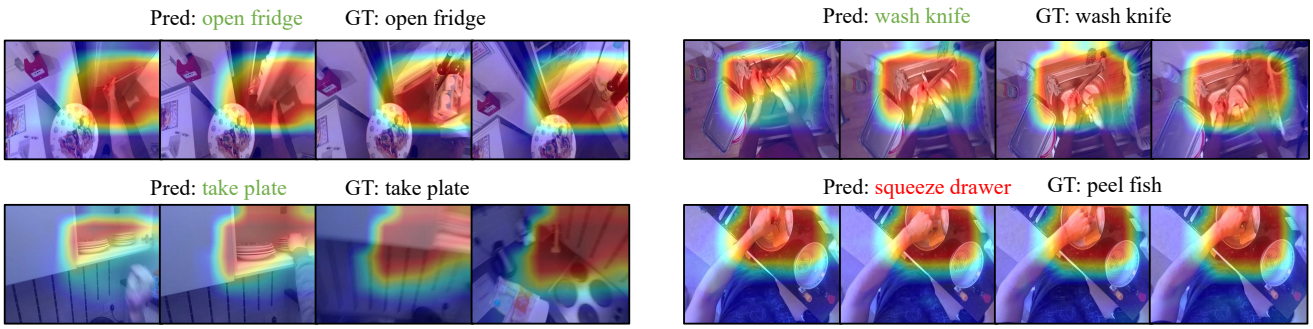

Figure 3. Assignments visualization of the IPL R(2+1)D model. We illustrate the sum of assignments on the top-K verb prototypes for each feature vector on the spatio-temporal feature map. For each input clip, we uniformly sample four frames. Higher assignment values shows in red. We also print the predictions and the ground-truth above the frames (Green for correct predictions and Red for failure cases).

GT	wash spoon	turn-on tap	pour-up oil	skin carrot	put-down knife	scoop coffee	pick-up utensil
Baseline	wash saucepan	turn-on gas	pour-up rice	put-down pasta	pick-up spoon	open lid	pick-up bin
IPL	wash spoon	turn-on tap	pour-up oil	cut carrot	pick-up knife	open coffee	pick-up lid

Figure 4. Qualitative results of the IPL R(2+1)D model and the baseline model.

Method	Act@1	Verb@1	Noun@1
R(2+1)D Baseline	37.62	67.28	47.55
R(2+1)D + NetVLAD	38.80	67.39	49.38
IPL R(2+1)D w/o Selection	38.50	66.82	49.68
IPL R(2+1)D w/o BG Center	40.02	68.20	50.30
IPL R(2+1)D	**40.98**	**68.61**	**51.24**

Table 6. Ablation studies on the EPIC-KITCHENS-100 Val. set.

verb-to-noun selection, the result on *verb* classification also drops by 1.59%. This demonstrates that the combination of the features on the irrelevant prototypes does harm to both the *verb* classification and *noun* classification.

The background center We conduct the experiment without the background center. As shown in Table 6, the top-1 accuracy of the model "IPL w/o BG Center" on *noun* recognition drops by 0.94%. This result demonstrates the effectiveness of introducing a background center. This design may reduce the noise information during the feature assignment and enhance the selected noun features.

4.5. Qualitative Results

For each input video, K noun feature groups are selected based on Top-K verb predictions for noun classification. Thus, for each feature vector in the spatio-temporal feature map, the assignments on the K verb prototypes determine its contributions to the final noun classification. We sample four frames corresponding to the final four spatial feature maps and plot the sum of assignments on Top-K verb prototypes. As shown in Fig. 3, the region with high assignment weights indicates the interacting area and the location of the active objects. These qualitative results explain why our IPL can improve the recognition accuracy of active objects.

We compare the predictions of our IPL R(2+1)D model to its baseline in Fig. 4. In the first example, the baseline model classifies the video to "saucepan" as its noun prediction, but "saucepan" is actually the distracting object next to the active object "spoon". In contrast, our IPL correctly recognizes the active object. As shown in the fifth example, although the prediction for verb classification is not correct, our IPL can recognize that the active object is "knife" rather than "spoon". This is because the verb-to-noun selection mechanism is robust to the verb classification results.

5. Conclusion

In this paper, we present an interactive prototype learning (IPL) framework for egocentric video classification. IPL introduces a set of discriminative verb prototypes to enhance the interactions between noun and verb classification. We evaluate IPL on three large-scale egocentric video classification datasets. Experimental results demonstrate that IPL is able to effectively learn action-centric noun representations. In the future work, we will take a closer look into the affordance of the active object. Besides, we will make full use of the long-range temporal context to help the active object recognition.

References

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *CVPR*, 2016. 2, 3, 5, 7

[2] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *ECCV*, 2018. 5

[3] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 2, 5, 6, 7

[4] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2018. 3

[5] Dima Damen, Hazel Doughty, Giovanni 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 T-PAMI*, 2020. 1

[6] 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 *ECCV*, 2018. 1, 2, 3, 5

[7] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Rescaling egocentric vision. *arXiv preprint arXiv:2006.13256*, 2020. 1, 2, 3, 5, 6

[8] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *IEEE T-PAMI*, 2017. 2

[9] Alireza Fathi, Xiaofeng Ren, and James M Rehg. Learning to recognize objects in egocentric activities. In *CVPR 2011*, 2011. 1

[10] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 2, 6

[11] Antonino Furnari, Sebastiano Battiato, Kristen Grauman, and Giovanni Maria Farinella. Next-active-object prediction from egocentric videos. *Journal of Visual Communication and Image Representation*, 49:401–411, 2017. 1

[12] Deepti Ghadiyaram, Matt Feiszli, Du Tran, Xueting Yan, Heng Wang, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *CVPR*, 2019. 5, 6, 7

[13] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018. 3

[14] Rohit Girdhar, Deva Ramanan, Abhinav Gupta, Josef Sivic, and Bryan Russell. Actionvlad: Learning spatio-temporal aggregation for action classification. In *CVPR*, 2017. 3, 5

[15] Yifei Huang, Minjie Cai, Zhenqiang Li, Feng Lu, and Yoichi Sato. Mutual context network for jointly estimating egocentric gaze and action. *IEEE TIP*, 2020. 7

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, pages 448–456, 2015. 5

[17] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010. 3

[18] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1

[19] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*, 2019. 2, 6

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 2

[21] Yin Li, Miao Liu, and James M Rehg. In the eye of beholder: Joint learning of gaze and actions in first person video. In *ECCV*, 2018. 1, 2, 5, 7

[22] Yin Li, Miao Liu, and James M. Rehg. In the eye of the beholder: Gaze and actions in first person video. *arXiv preprint arXiv:2006.00626*, 2020. 7

[23] Yin Li, Zhefan Ye, and James M Rehg. Delving into egocentric actions. In *CVPR*, 2015. 7

[24] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019. 2, 6

[25] Miao Liu, Siyu Tang, Yin Li, and James Rehg. Forecasting human object interaction: Joint prediction of motor attention and egocentric activity. In *ECCV*, 2020. 2

[26] Minghuang Ma, Haoqi Fan, and Kris M Kitani. Going deeper into first-person activity recognition. In *CVPR*, 2016. 2

[27] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017. 3, 5

[28] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, 2017. 2

[29] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 2013. 3

[30] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *ECCV*, 2020. 2

[31] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020. 2

[32] Gunnar A. Sigurdsson, Abhinav Gupta, C. Schmid, Ali Farhadi, and Alahari Karteek. Actor and observer: Joint modeling of first and third-person videos. In *CVPR*, 2018. 1

[33] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014. 2

[34] Suriya Singh, Chetan Arora, and CV Jawahar. First person action recognition using deep learned descriptors. In *CVPR*, 2016. 7

[35] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Lsta: Long short-term attention for egocentric action recognition. In *CVPR*, 2019. 2, 7

[36] Swathikiran Sudhakaran and Oswald Lanz. Attention is all we need: Nailing down object-centric attention for egocentric activity recognition. In *BMVC*, 2018. 7

[37] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 2

[38] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 2, 5, 6

[39] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Val Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 2, 6

[40] Xiaohan Wang, Yu Wu, Linchao Zhu, and Yi Yang. Symbiotic attention with privileged information for egocentric action recognition. In *AAAI*, 2020. 2, 7

[41] Xiaohan Wang, Linchao Zhu, Yu Wu, and Yi Yang. Symbiotic attention for egocentric action recognition with object-centric alignment. *IEEE T-PAMI*, 2020. 2, 7

[42] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *CVPR*, 2019. 2, 7

[43] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020. 7

[44] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 2

[45] Zhongwen Xu, Yi Yang, and Alex G Hauptmann. A discriminative cnn video representation for event detection. In *CVPR*, 2015. 3

[46] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018. 2, 6

[47] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *ECCV*, 2018. 2