RECOGNIZING ACTIONS USING OBJECT STATES

Nirat Saini, Bo He, Gaurav Shrivastava, Sai Saketh Rambhatla, Abhinav Shrivastava
University of Maryland, College Park
{nirat,bohe,gauravsh,rssaketh,abhinav2}@umd.edu

ABSTRACT

Object-centric actions cause changes in object states, including their visual appearance and their immediate context. We propose a computational framework that uses only two object states, *start* and *end*, and learns to recognize the underlying actions. Our approach has two modules that learn subtle changes induced by the action and suppress spurious correlations. We demonstrate that only two object states are sufficient to recognize object-centric actions. Our framework performs better than approaches that use multiple frames and a relatively large model. Moreover, our method generalizes to unseen objects and unseen video datasets.

1 INTRODUCTION

Objects in real world exists in different color, shape and structure. For instance, an apple can be red or green, cut or peeled, raw or riped. Growing interest in object state transformations is still limited to image based datasets such as Isola et al. (2015) and Yu & Grauman (2014), which study a set of object-state transformations for edible objects and shoes. However, using objects states for videos is still uncommon. In Fathi & Rehg (2013), actions are detected based on changes in object states. Similar to Fathi & Rehg (2013), we hypothesize that most object-centric actions induce specific changes in object states, where state includes object's appearance and immediate context. Humans are adept at identifying such actions simply based on the state transformations. In the commonsense reasoning literature, this notion of state transformation is also known as *fluent* of an object Liu et al. (2017); Fire & Zhu (2015); Mueller (2006), defined as *the status of an object which varies over time.* For instance, a lemon's *fluent* changes from whole→pieces over time, associated with *cut* action.

In this work, we propose a computational model that embodies the intuition of leveraging object state transformations, or *fluent* of an object, to reveal the underlying action being performed. In fact, we limit our focus to the question: Can we infer actions just from emphtwo distinct object states – *start* and *end*? Note that we do not claim that more intermediate states will not be beneficial for modeling transformations (*e.g*, see Wang et al. (2016b)). However, for object-centric actions, we show that only using two states can outperform standard approaches using many more frames. Moreover, prior works in action recognition domain Zhou et al. (2018a); Simonyan & Zisserman (2014); Yang et al. (2020a) use training and test splits based on actions, *i.e*, for label *cut apple*, few videos are part of train and some are testing videos. However, we focus to learn the action or transformation of any object. Hence, we create dataset splits where for an action, objects are split up as seen and unseen objects as training and test set respectively.

Firstly, our proposed method selects two bounding boxes, representing *start* and *end* states, and then extracts their representations. Our manipulation module uses these representations and learns a *fluent* representation used to classify actions. We use cooking video datasets: EpicKitchen55 Damen et al. (2018), YouCook2 Zhou et al. (2018b), and EpicKitchens100 Damen et al. (2020), and firstly train on just EpicKitchen55 and show that our model can generalize to YouCook2 and EpicKitches100. Second, we hold out several unseen objects for all actions while training and generalize to transformations of these unseen objects in test set. We show that our approach is able to improve performance for both *object-manipulating* and *context-manipulating* actions. To summarize, our contributions are as follows: (a) developing a manipulation module with object and context modules, that can understand object state changes, (b) demonstrating that amplifying relevant information from just two states is enough to outperform architectures that use several frames, and (c) demonstrating the generalization of our approach to unseen objects datasets.

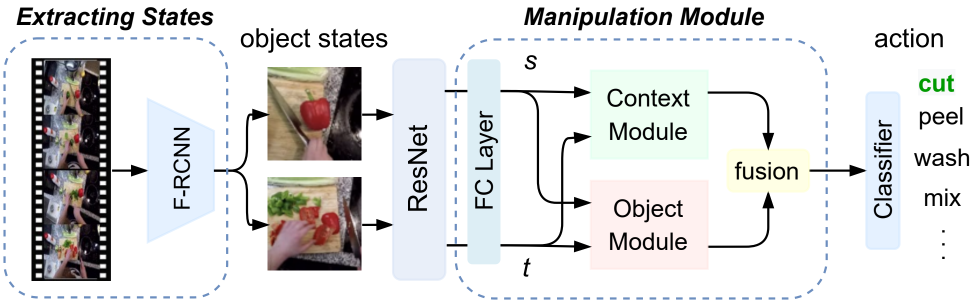

Figure 1: Our framework extracts *start* and *end* object states from the video. We use ResNet50 He et al. (2016) pre-trained on ImageNet Deng et al. (2009) as our backbone network. We introduce a novel framework to model object-transforming and context-transforming actions separately. The context module learns the subtle changes and similarities in the object environment, *i.e*, context. The object module focuses on the object states, rather than the context.

2 Approach

In this work, our goal is to recognize object-centric actions using *start→end* object state changes. A state manipulation can correspond to *object-manipulating actions* that change object's visual appearance and *context-manipulating actions* that change its context. Our approach is inspired by attention correlation between labels and images Xu et al. (2021). Our Manipulation Module shown in Figure 1 are designed to amplify similarity between these states to suppress spurious signals and amplify the differences to leverage subtle changes between *start* and *end* states.

Currently, no video dataset exists with specific object states before and after an action. Hence, as a pre-processing step, we first extract object states, or regions of the object in two distinct frames before and after the action has been performed. More details of are in provided in the appendix A. Once two distinct object states are captured, we propose our method using Manipulation Module, as shown in Figure 1. Hence, the input to our model are the *start* ($\mathcal{I}_s$) and *end* ($\mathcal{I}_t$) states. We use a pre-trained deep neural network to extract features f_s and f_t ($f_s, f_t \in \mathbb{R}^n$; $n = 2048$) from $\mathcal{I}_s$ and $\mathcal{I}_t$, respectively. We then pass f_s and f_t through an FC layer with ReLU non-linearity to obtain g_s and g_t ($g_s, g_t \in \mathbb{R}^n$). The features g_s and g_t are then passed through the Manipulation Module for further processing.

Our hypothesis is that the similarities and differences between deep features g_s and g_t capture the manipulation in objects and its context. Hence, our manipulation module consists of –- *Context Module* and *Object Module*. The object module captures and difference in object states, whereas context module learns to amplify the subtle similarity between object's context or environment that can be used for classification. Both module take g_s and g_t as input to produce representations f_c and f_o respectively. The two features f_c and f_o are then fused to produce the final representation. We describe one of the modules (the context module) in detail below and the other module is implemented in a similar fashion with minor changes.

Context Module captures the similarity between the *start* and *end* states. We first take the outer product of g_s and g_t denoted as $O = g_s \otimes g_t$ ($O \in \mathbb{R}^{n \times n}$). Element $o_{i,j}$ represents the similarity between i^{th} element of g_s with j^{th} element of g_t. Moreover, let $\mathbf{o_{i*}}$ and $\mathbf{o_{*j}}$ represent the i^{th} row and j^{th} column of O respectively. Then, $\mathbf{o_{i*}}$ captures the similarity of all the elements in g_t with respect to i^{th} element of g_s. To know the most similar element among g_t with respect to i^{th} element of g_s, we can take a row-wise `softmax` over O. Similarly, for j^{th} element of g_t, column o_{*j} represents the similarity with all the elements of g_s. Using a column-wise `softmax`, we can interpret the most similar and least similar element of g_s with respect to j^{th} element of g_t. Therefore, by applying column-wise and row-wise `softmax`, we get two matrices, $\mathcal{A}$ and $\mathcal{A}'$ ($\mathcal{A}, \mathcal{A}' \in \mathbb{R}^{n \times n}, n = 2048$).

$$\mathcal{A}[i,:] = \frac{e^{\gamma \mathbf{o_{i*}}}}{\sum_{j=1}^{d} e^{\gamma o_{ij}}} \quad \text{and} \quad \mathcal{A}'[:,j] = \frac{e^{\gamma \mathbf{o_{*j}}}}{\sum_{i=1}^{d} e^{\gamma o_{ij}}},$$

Table 1: **Quantitative results on Unseen Objects**. This table summarizes results for actions on objects not seen during training. (test set). All values are shown as mean with confidence interval (95%) over 5 splits. In the table we highlight the best and second best result in bold+underline and underline respectively.

Network Input	Method	All Actions		Object Transforming Actions		Context Transforming Actions	
		Acc.	Mean Acc.	Acc.	Mean Acc.	Acc.	Mean Acc.
8 Frames	TRN Zhou et al. (2018a)	31.49 ± 2.2	7.16 ± 1.5	18.89 ± 6.0	7.24 ± 1.6	44.87 ± 7.1	7.07 ± 1.6
	TPN Yang et al. (2020a)	35.60 ± 4.1	8.78 ± 1.3	$\underline{28.61 \pm 9.4}$	8.59 ± 2.1	41.36 ± 8.7	7.42 ± 2.5
3 Frames	TRN Zhou et al. (2018a)	29.78 ± 2.5	7.38 ± 2.8	16.26 ± 6.7	6.26 ± 2.1	44.16 ± 7.7	8.50 ± 4.0
	TPN Yang et al. (2020a)	30.59 ± 0.7	7.16 ± 1.5	18.83 ± 9.2	7.44 ± 2.4	41.14 ± 12.2	6.86 ± 1.5
8 Objects	TRN Zhou et al. (2018a)	28.75 ± 2.5	7.53 ± 1.6	15.34 ± 5.9	6.35 ± 1.7	42.91 ± 7.0	8.71 ± 2.3
	TPN Yang et al. (2020a)	34.92 ± 2.6	9.22 ± 2.7	27.65 ± 6.8	8.51 ± 2.3	41.98 ± 5.6	7.93 ± 3.3
3 Objects	TRN Zhou et al. (2018a)	27.86 ± 3.2	7.83 ± 2.5	16.70 ± 4.7	7.35 ± 1.6	39.68 ± 7.8	8.31 ± 3.5
	TPN Yang et al. (2020a)	28.84 ± 3.8	6.58 ± 2.5	12.30 ± 9.8	5.65 ± 4.0	$\underline{45.95 \pm 14.9}$	7.50 ± 1.6
2 Objects	Fathi et al. Fathi & Rehg (2013)	30.59 ± 1.1	8.17 ± 1.7	26.97 ± 2.6	$\underline{8.81 \pm 1.1}$	34.00 ± 3.5	7.53 ± 2.3
	MaxpoolResFeats	28.37 ± 4.1	9.70 ± 1.7	18.88 ± 2.1	8.56 ± 1.8	38.47 ± 9.3	$\underline{9.19 \pm 2.9}$
	ConcatResFeats	$\underline{35.71 \pm 0.7}$	$\underline{9.83 \pm 1.3}$	27.30 ± 4.5	7.40 ± 2.7	42.56 ± 6.6	8.37 ± 1.0
	Ours	38.83 ± 1.4	10.08 ± 1.6	29.29 ± 8.0	8.96 ± 2.9	47.70 ± 7.7	10.79 ± 1.4

where γ is the inverse temperature parameter. We compute row and column sum for $\mathcal{A}$ and $\mathcal{A}'$ respectively, to get raw state masks, $\mathbf{m^t}$ and $\mathbf{m^s}$ in Eq.1 To get the final masks, we normalize with ReLU (Eq.2). We compute the final representation f_c of the context module in Eq. 3, where $\odot$ is the elementwise multiplication and $W \in \mathbb{R}^{2n \times n}$ and $b \in \mathbb{R}^n$.

$$\hat{m}^t_j = \sum_{i=1}^{d} \mathcal{A}_{ij} \quad \text{and} \quad \hat{m}^s_i = \sum_{j=1}^{d} \mathcal{A}'_{ij}. \tag{1}$$

$$m^s = \text{ReLU}(\hat{m}^s - \mathbb{E}[\hat{m}^s]) \quad \text{and} \quad m^t = \text{ReLU}(\hat{m}^t - \mathbb{E}[\hat{m}^t]) \tag{2}$$

$$f_c = W^T([m_s \odot g_s; m_t \odot g_t]) + b \tag{3}$$

Similarly, we can get the object representation f_o, except instead of using similarity between states ($\mathcal{A}$ and $\mathcal{A}'$), use distance measure ($\mathcal{D}$ and $\mathcal{D}'$) computed as

$$\mathcal{D} = 1./\mathcal{A} \quad \text{and} \quad \mathcal{D}' = 1./\mathcal{A}'$$

where $1./*$ is the element-wise inversion of matrices. The final representation is the output of max-pooling of f_c and f_o. We train using a linear classifier in an end-to-end fashion using softmax Cross-Entropy loss function.

3 Experiments and Results

3.1 Dataset and Evaluation

We use Epic-kitchens-55 (EK-55) Damen et al. (2018) which has kitchen activities captured with an egocentric camera. Among all actions, only 13 actions are object-transforming and 30 actions are context-transforming. In total, we use 7342 videos, 2120 for object-transforming and 5222 for context-transforming actions. We create five train/val/test splits of the dataset and report the mean performance across splits. Each split has all 13 object-transforming actions and randomly chosen 13 context-transforming actions. Note that the test set, referred to as 'unseen object-actions' in each split contains actions involving objects unseen during training. We report overall accuracy and Mean accuracy. More details are provided in appendix A.

3.2 Quantitative Results

We compare our model with two types of action recognition models:
Spation-temporal Baselines. Our setup is RGB based, hence we compare with the best RGB-only model for action recognition Yang et al. (2020a); Zhou et al. (2018a). TPN Yang et al. (2020a)

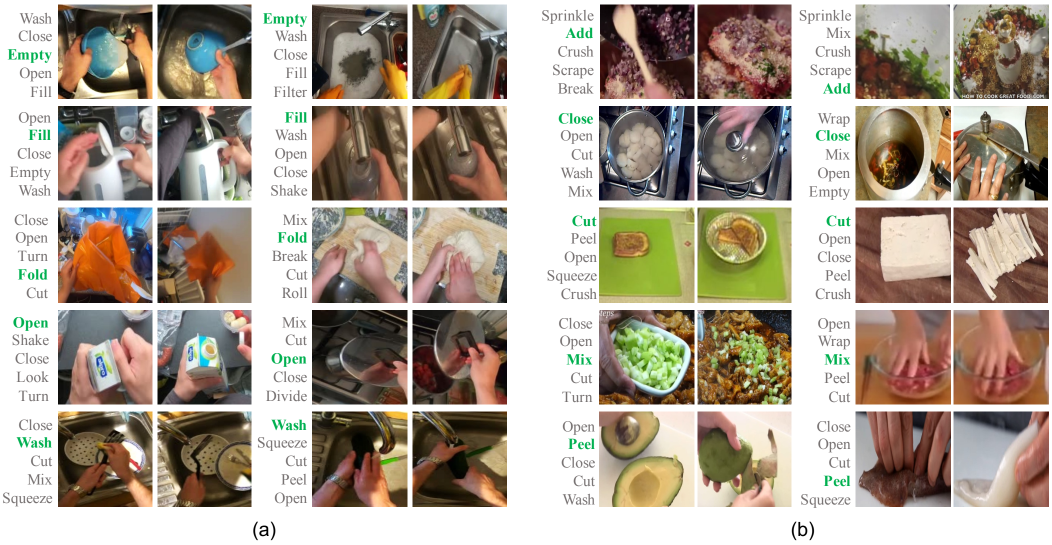

Figure 2: We show qualitative results for (a) Epic-Kitchen 100 Damen et al. (2020) and (b) Youcook2. We show top 5 predictions for object manipulating actions: empty, fill, open, peel, cut, close, and context manipulating actions: fold, wash, mix, add.

shows state-of-the-art results on multiple datasets, including EK-55. TPN Yang et al. (2020a) captures visual temporal scales of different actions. TRN Zhou et al. (2018a) accounts for relationship between frames at multiple scale (2,3,4 and 8). Both models use minimum of 3 frames, and finally consolidate the relationships between all frames. We use 3 and 8 frames for these models, with TSN Wang et al. (2016a) based sampling frames. We also train these baselines for with 3 and 8 object states, to compare with our 2 objects states setup.

Object States Baselines. We compare with three other baselines which use exactly as input as ours, *i.e*, 2 object states. Fathi et al. Fathi & Rehg (2013). detect changing regions in video and train SVMs for each action. We use our implementation for Fathi & Rehg (2013), with deep features for regions to replicate the results. We propose two more object states baselines: MaxpoolResFeats uses `maxpool`(f_s,f_t) and ConcatResFeats uses `concatenation`(f_s, f_t) for action classification. Note that we follow the notation explained in section 2, where f_s and f_t are ResNet features for *start* and *end* states. For a fair comparison, these object states baseline models have around the same number of trainable parameters as our model.

To keep it consistent with TPN Yang et al. (2020a) and TRN Zhou et al. (2018a), we use ResNet50 He et al. (2016) as our backbone network and all setups follow three-crop testing evaluation protocol. The results are reported in Table 1 for unseen object actions (test set). For 5 different splits for action classes, we denote the final accuracy as mean and confidence interval with 95% confidence, for object-transforming and context-transforming actions separately as well. Our model outperforms all the baselines for 3 and 8 frames and bounding boxes setup, not only on majority classes but also performs better for minority classes (Table 1). We show the of our object and context modules for capturing the similarity and dissimilarity, to significantly improve object-transforming actions, as compared to using max of ResNet features (MaxpoolResFeats baseline) and concatenation baseline.

3.3 Ablation Studies

We also do an ablation study on one of the validation set split (Seen object actions), to justify our architectural choices Table 2.

Object Baselines: Using only ResNet features for action recognition, either maxpool (MaxpoolResFeats in Table 2) or concatenation (ConcatResFeats) of object states, also performs better than 3 and 8 frames models.

(*start*→*end*) Attention: This is akin to using a single module between *start* and *end* states. We use a non-spatial attention block, inspired by Wang et al. (2018); He et al. (2020). More details on how

Table 2: **Ablation analysis on Seen Objects**. We show how our amplifiers and separate modeling of object-transforming and context-transforming actions help in object-centric action recognition. All the experiments are done on split1. In the table we highlight the best and second best result in bold+underline and underline respectively.

	All Actions		Object Transforming		Context Transforming	
	Acc.	Mean Acc.	Acc.	Mean Acc.	Acc.	Mean Acc.
ConcatResFeats	65.9	23.23	72.4	27.56	48.89	18.49
MaxPoolResFeats	51.86	22.75	56.80	30.78	42.39	18.12
(*start*→*end*) Attention Wang et al. (2018)	58.17	20.95	65.60	23.69	39.39	16.22
Module Alternate	60.74	24.06	64.01	29.62	48.52	18.12
Our Approach	**68.77**	**25.39**	**76.40**	**33.21**	**49.49**	**18.96**

this attention is different than our method are in appendix. This setup shows we need different ways to amplify context and object features between *start* and *end* states.

Module Alternative: For understanding the significance of our proposed Manipulation Module, we model object manipulating actions as difference between features for object states and mean of features for context transforming actions. Fusion of the two is done in the same way with maxpool. This experiment shows need for two separate object and context modules.

3.4 Qualitative Results

We show qualitatively how our method performs on out-of-domain data, such as Epic Kitchens-100 Damen et al. (2020) and YouCook2 Zhou et al. (2018b). Although, our state detector only finds a limited set of seen objects for these datasets, we achieve, 65.60% Top1 accuracy and 90.10% Top5 accuracy for Epic Kitchens-100, for unseen and seen actions. Some examples of correctly classified actions are shown in Figure 2. Note that these samples are classified correctly by our model among top 5 predictions.

4 Conclusion

We presented a framework for recognizing object-centric actions using object states. We show that by leveraging object states *object-manipulating* and *context-manipulating* actions can be learnt. We propose a novel manipulation module, with object and context blocks, that capture the subtle similarities and differences between object states. We also show generalization to unseen objects and out-of-domain datasets. In future work, we intend to extend object states for procedure planning and action segmentation.

References

Y. Atzmon, F. Kreuk, U. Shalit, and Gal Chechik. A causal view of compositional zero-shot recognition. *ArXiv*, abs/2006.14610, 2020.

Nachwa Abou Bakr, J. Crowley, and Rémi Ronfard. Recognizing manipulation actions from state-transformations. *ArXiv*, abs/1906.05147, 2019.

Gedas Bertasius and Lorenzo Torresani. Cobe: Contextualized object embeddings from narrated instructional video, 2020.

H. Chen, Z. Nan, Jiang Jing-jing, and Nanning Zheng. Learning to infer unseen attribute-object compositions. 2020.

D. Damen, H. Doughty, G. Farinella, S. Fidler, Antonino Furnari, Evangelos Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and Michael Wray. Scaling egocentric vision: The epic-kitchens dataset. *ArXiv*, abs/1804.02748, 2018.

D. Damen, H. Doughty, G. Farinella, S. Fidler, Antonino Furnari, Evangelos Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and Michael Wray. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

Jia Deng, W. Dong, R. Socher, L. Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR 2009*, 2009.

Alireza Fathi and James M Rehg. Modeling actions through state changes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2579–2586, 2013.

Christoph Feichtenhofer, A. Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1933–1941, 2016.

A. Fire and S. Zhu. Learning perceptual causality from video. *ACM Trans. Intell. Syst. Technol.*, 7: 23:1–23:22, 2015.

Rohit Girdhar, D. Ramanan, A. 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.

A. Gupta, Aniruddha Kembhavi, and L. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31:1775–1789, 2009.

Bo He, Xitong Yang, Zuxuan Wu, Hao Chen, Ser-Nam Lim, and Abhinav Shrivastava. Gta: Global temporal attention for video action understanding, 2020.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Phillip Isola, Joseph J. Lim, and E. Adelson. Discovering states and transformations in image collections. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1383–1391, 2015.

A. Karpathy, G. Toderici, Sanketh Shetty, T. Leung, R. Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1725–1732, 2014.

F. M. Khan, V. Singh, and R. Nevatia. Simultaneous inference of activity, pose and object. *2012 IEEE Workshop on the Applications of Computer Vision (WACV)*, pp. 281–288, 2012.

Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7082–7092, 2019.

Y. Liu, P. Wei, and S. Zhu. Jointly recognizing object fluents and tasks in egocentric videos. *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2943–2951, 2017.

T. McCandless and K. Grauman. Object-centric spatio-temporal pyramids for egocentric activity recognition. In *BMVC*, 2013.

I. Misra, A. Gupta, and M. Hebert. From red wine to red tomato: Composition with context. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1160–1169, 2017.

E. Mueller. Commonsense reasoning: An event calculus based approach. 2006.

Tushar Nagarajan and K. Grauman. Attributes as operators: Factorizing unseen attribute-object compositions. In *ECCV*, 2018.

Z. Nan, Y. Liu, N. Zheng, and S. Zhu. Recognizing unseen attribute-object pair with generative model. In *AAAI*, 2019.

Shaoqing Ren, Kaiming He, Ross B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

L. Shen, Serena Yeung, Judy Hoffman, G. Mori, and Li Fei-Fei. Scaling human-object interaction recognition through zero-shot learning. *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1568–1576, 2018.

K. Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

Haoliang Tan, L. Wang, Q. Zhang, Zhanning Gao, Nanning Zheng, and Gang Hua. Object affordances graph network for action recognition. In *BMVC*, 2019.

L. Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, D. Lin, X. Tang, and L. Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016a.

X. Wang, Ross B. Girshick, A. Gupta, and Kaiming He. Non-local neural networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, 2018.

Xiaolong Wang, Ali Farhadi, and Abhinav Gupta. Actions ˜ transformations. In *CVPR*, 2016b.

K. Wei, Muli Yang, H. Wang, Cheng Deng, and Xianglong Liu. Adversarial fine-grained composition learning for unseen attribute-object recognition. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3740–3748, 2019.

Jianchao Wu, L. Wang, Jie Guo, and Gangshan Wu. Learning actor relation graphs for group activity recognition. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9956–9966, 2019.

Tete Xiao, Quanfu Fan, Dan Gutfreund, Mathew Monfort, A. Oliva, and B. Zhou. Reasoning about human-object interactions through dual attention networks. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3918–3927, 2019.

Guangyue Xu, Parisa Kordjamshidi, and Joyce Yue Chai. Zero-shot compositional concept learning. *ArXiv*, abs/2107.05176, 2021.

Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and B. Zhou. Temporal pyramid network for action recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 588–597, 2020a.

Muli Yang, C. Deng, Junchi Yan, Xianglong Liu, and Dacheng Tao. Learning unseen concepts via hierarchical decomposition and composition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10245–10253, 2020b.

A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 192–199, 2014.

Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 803–818, 2018a.

Luowei Zhou, Nathan Louis, and Jason J. Corso. Weakly-supervised video object grounding from text by loss weighting and object interaction. In *BMVC*, 2018b.

Yipin Zhou and T. Berg. Learning temporal transformations from time-lapse videos. *ArXiv*, abs/1608.07724, 2016.

A Appendix

A.1 Related Work

The study of object states and how they transform from one to the other has recently garnered interest in the computer vision community Isola et al. (2015); Yu & Grauman (2014); Nagarajan & Grauman (2018); Chen et al. (2020); Nan et al. (2019); Atzmon et al. (2020); Yang et al. (2020b); Misra et al. (2017); Wei et al. (2019); Bertasius & Torresani (2020); Liu et al. (2017); Fire & Zhu (2015); Zhou & Berg (2016). Understanding these states could enable an object recognition system to generalize better to unseen states, as studied in Bertasius & Torresani (2020); Misra et al. (2017). However, most studies have focused on image-based datasets Isola et al. (2015); Yu & Grauman (2014); Misra et al. (2017), where the goal is to capture the underlying distribution of states (*e.g*, different sizes of heels in shoes, old and new laptops). Some of these states are just *attributes* of objects (*e.g*, red car and blue car) and are generally static over time; whereas other states are transformations of objects over time (*e.g*, raw→ripe, whole→pieces, on table→on plate). These different types of states provide very different information. In this work, we leverage the state transformations that happen over time for understanding transforming actions. For the rest of this paper, we will refer to changes in object states (including visual appearance and context) over time as *object state transformations*.

For action recognition, most modern approaches can be grouped into two categories: (a) models utilizing temporal and spatial convolutions over videos to classify the actions Girdhar et al. (2017); Simonyan & Zisserman (2014); Feichtenhofer et al. (2016); Karpathy et al. (2014); Lin et al. (2019); Wang et al. (2016a); Yang et al. (2020a); Zhou et al. (2018a), and (b) models which rely on human-object interactions over time to recognize actions Gupta et al. (2009); Khan et al. (2012); Shen et al. (2018); Wu et al. (2019); Tan et al. (2019); Xiao et al. (2019). In both categories, the focus is generally to leverage as much temporal information as possible; this is evident with the focus on recent works to utilize as many frames as computationally feasible to understand actions.

Similar to our approach, Bakr et al. (2019); McCandless & Grauman (2013); Fathi & Rehg (2013) propose using state transformations for action recognition; *e.g*, Bakr et al. (2019) uses five key-frames for modeling transformations for action recognition. Closest to our approach is the work of Fathi and Rehg Fathi & Rehg (2013), which also uses two object bounding boxes as states for action recognition. However, our approach explicitly amplifies affinity and differential signals between states to extract *fluent* representations, which are used for action recognition.

A.2 Dataset Splits Extension

We use Epic-kitchens-55 (EK-55) Damen et al. (2018) as our primary dataset. EK-55 has kitchen activities captured with an egocentric camera. In total, it has 352 sparsely annotated bounding boxes for objects and 125 actions. Among all actions, only 13 actions are object-transforming and 30 actions are context-transforming. We choose these actions such that each action has atleast 40 videos in total. Our subset is also impacted by pre-processing step (object bounding boxes detection), since we require two bounding boxes of the object of interest either from the detector or the ground truth. This drastically reduces the number of videos with a long-tailed distribution over the classes. In total, we use 7342 videos, 2120 for object-transforming and 5222 for context-transforming actions.

Due to imbalanced number of object-manipulating and context-manipulating actions, we create five train/val/test splits of the dataset and report the mean performance across splits. Each split has all 13 object-transforming actions and randomly chosen 13 context-transforming actions. The test set in each split contains actions involving objects unseen during training. This is different from the typical test set provided for EK-55 Damen et al. (2018). We argue that using those splits do not provide evaluation of the model learning actions or latching onto object features. Therefore, by having different objects in train and test set inspired by Nagarajan & Grauman (2018), we ensure that the model learns the action, and does not rely only on object visual features. We refer to test set as "unseen object-actions" and validation set as "seen object-actions" set.

A.3 Preprocessing: Extracting object states

As discussed above, analyzing object state provides useful information for inferring actions. For this purpose, we first identify two frames in a video, one before and the other after the action has

Table 3: **Quantitative results on Seen Objects**. This table summarizes results for seen object actions (validation set). All values are mean with confidence interval (95%). In the table we highlight the best and second best result in bold+underline and underline respectively.

Network Input	Method	All Actions		Object Transforming Actions		Context Transforming Actions	
		Acc.	Mean Acc.	Acc.	Mean Acc.	Acc.	Mean Acc.
8 Frames	TRN Zhou et al. (2018a)	48.20 ± 2.0	11.68 ± 1.6	47.20 ± 7.0	14.18 ± 1.9	48.20 ± 7.1	9.07 ± 2.1
	TPN Yang et al. (2020a)	56.13 ± 5.2	14.11 ± 1.8	13.01 ± 11.4	18.42 ± 3.2	49.67 ± 9.4	9.60 ± 3.2
3 Frames	TRN Zhou et al. (2018a)	45.68 ± 5.4	12.27 ± 4.8	44.16 ± 8.3	13.90 ± 3.7	47.50 ± 8.3	10.62 ± 6.2
	TPN Yang et al. (2020a)	35.70 ± 2.0	8.63 ± 1.0	32.64 ± 7.0	11.05 ± 3.0	39.52 ± 10.1	6.09 ± 1.0
8 Objects	TRN Zhou et al. (2018a)	57.50 ± 1.6	17.90 ± 2.9	56.16 ± 2.8	20.77 ± 2.5	59.26 ± 3.0	14.76 ± 3.7
	TPN Yang et al. (2020a)	59.92 ± 2.9	17.00 ± 4.8	65.28 ± 4.6	21.90 ± 4.6	50.84 ± 5.6	11.58 ± 5.4
3 Objects	TRN Zhou et al. (2018a)	45.95 ± 3.8	13.62 ± 3.8	44.88 ± 8.9	15.36 ± 5.3	46.29 ± 6.4	11.86 ± 5.5
	TPN Yang et al. (2020a)	34.48 ± 6.3	8.57 ± 3.5	23.28 ± 15.1	8.97 ± 6.1	50.31 ± 8.0	8.38 ± 1.6
2 Objects	Fathi et al. Fathi & Rehg (2013)	55.93 ± 0.9	15.34 ± 3.1	61.12 ± 1.7	20.15 ± 4.8	46.30 ± 3.4	10.26 ± 1.6
	MaxpoolResFeats	56.80 ± 1.0	18.47 ± 8.2	56.48 ± 2.4	21.35 ± 9.9	56.57 ± 5.9	14.40 ± 7.0
	ConcatResFeats	66.08 ± 0.9	18.80 ± 2.5	70.30 ± 1.5	22.96 ± 4.1	57.87 ± 4.3	15.26 ± 2.5
	Ours	**68.36 ± 1.5**	**20.50 ± 2.8**	**71.84 ± 2.4**	**24.91 ± 2.1**	**61.36 ± 7.0**	**16.00 ± 3.9**

been performed, corresponding to different states of the object. We choose the highest confidence bounding box, extracted using an object detector, of the object of interest from each frame as the inputs to our model. We refer to these two regions as *start* and *end* states.

In particular, we leverage the temporal annotations provided with the video to identify clips where the action is performed. We extract bounding boxes using an object detector on the first N frames of the first clip and the last N frames from the last clip, and choose the box with highest class confidence score for the object under consideration as *start* and *end* states, respectively.

A.4 Implementation Details

We use ego-centric kitchen video dataset, Epic-Kitchen-55 (EK-55) Damen et al. (2018) for training. This dataset provides temporal action localization labels and has object-centric actions. Each video can have multiple clips of same object-action pair. We consider first and last clip for the same object-action in proximity, so that accurate *start* and *end* states are captured for objects. After merging the first and last clips from the video, we use one merged clip for each object-action pair from a video.

For extracting object states, use a Faster-RCNN Ren et al. (2015) detector, pre-trained on EK-55 Damen et al. (2018), which has sparsely annotated ground truth bounding boxes for 295 object classes. We apply it on two sets of images from an object-action clip with 30fps: first and last 20 frames. From each set, the object's bounding box with the highest confidence is chosen, which should be higher than 65%. This gives use *start* and *end* object states. Features are extracted using ResNet50 He et al. (2016) pretrained on ImageNet Deng et al. (2009) from the penultimate layer, and are of dimensions 1×2048. Extracted features are then passed through our *fluent* Attention module, and classified for all actions. For our model, we use learning rate 1e-5, Adam optimizer and ReLU activation. Since our dataset follows long-tail distribution, we use weighted cross-entropy loss over the actions. We empirically choose a temperature value as 100. We use L2-normalization on output of both context and object, before fusion.

A.5 Results and Ablation

We also show results on seen object in Table 3. Similar to unseen objects, our model outperforms previous models with a margin for all categories.

(*start*→*end*) Attention: We use a non-spatial attention block, inspired by Wang et al. (2018); He et al. (2020). Given an input feature (*start*/*end* state, denoted as g_s/g_t), three linear projections are applied to compute query (q) with respect for before state g_s, and key (k) and value (v) with respect to after state g_t. The attention (a) is weighted sum of value representation, where the weight (M) is dot product between the query and the corresponding key, followed by a softmax normalization as

$$\begin{aligned} &M = \mathrm{softmax}\left(\frac{qk^T}{\lambda}\right) \\ &a = Mv \\ &o = g_s + \mathrm{BatchNorm}(a) \end{aligned}$$

where o represents attention output for *start* state with respect to *end* state, and λ is the temperature parameter. Note that $g_s, q, k, v, o, a \in \mathbb{R}^n; M \in \mathbb{R}^{n \times n}$, where $n = 2048$. This experiment shows we need different ways to amplify similarities and differences between *start* and *end* states.