EPIC-KITCHENS-100- 2022 Challenges Report

Dima Damen, Adriano Fragomeni, Toby Perrett, Daniel Whettam, Michael Wray, Bin Zhu
University of Bristol, UK

Antonino Furnari, Giovanni Maria Farinella
University of Catania, Italy

Davide Moltisanti
University of Edinburgh, UK

Abstract

This report presents the findings from the 4th EPIC-KITCHENS-100 challenges, opened from Jan 2022 and concluded on the 1st of June 2022. It serves as an introduction to all technical reports that were submitted to the 10th EPIC@CVPR2022 workshop, and an official announcement of the winners.

1. EPIC-KITCHENS-100

All challenges are based on the publicly available EPIC-KITCHENS-100 dataset. In summary, EPIC-KITCHENS-100 provides 20M frames of egocentric footage, captured in an unscripted manner, with carefully collated annotations of 90K fine-grained actions. Details of how the dataset was collected and annnotated are available in our IJCV paper [6].

This report details the submissions and winners of the 2022 edition of the five challenges available on CodaLab: Action Recognition, Action Anticipation, Action Detection, Unsupervised Domain Adaptation for Recognition and Multi-Instance Retrieval. For each challenge, submissions were limited per team to a maximum of 50 submissions in total, as well as a maximum daily limit of 1 submission. In Sec. 2, we update the statistics of dataset download and usage. The results for all challenges are provided in Sec. 3-7. The winners of the 2022 edition of these challenges are noted in Sec. 8.

A snapshot of the complete leaderboard, when the 2022 challenge concluded on the 1st of June, is available at `http://epic-kitchens.github.io/2022#results`.

Details of the three previous year's reports for 2021, 2020 and 2019 challenges are available from the technical reports [8], [9] and [10] respectively.

In 2022, EPIC-KITCHENS-100 was downloaded from the official data.bris.ac.uk servers a total of 2430 times, compared to 2335 downloads in 2021. In Table 1, we show the distribution of countries of download. Additionally, in Fig. 1 we show the number of page visits for the webpage in the same duration. Finally, in Table 2 we report the active submissions this year, reporting the change over last year using arrows. Interestingly, number of submissions has dropped for both action recognition challenges (supervised and UDA). We explain in the relevant sections how these baeslines are now harder to beat limiting the number of contributions. The remaining 3 challenges saw a significant increase over last year's contributions, particularly for Multi-Instance Retrieval.

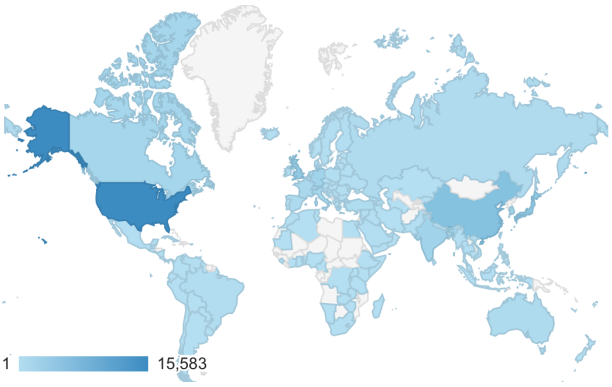


Figure 1: Heatmap of countries based on EPIC-KITCHENS-100 webpage view statistics, with an increase of 3.5K views between June 2021 and June 2022.

2. New Leaderboards

Following the recommendations of Codalab, we moved all 5 leaderboads to the new servers (`https://codalab.lisn.upsaclay.fr`) which offered increased memory and processing capacity. The *depricated* leaderboards remain available as *read-only* and all submissions have been backed up. To ensure continuity, All challenges this year match the description of last year's chal-

July 29th, 2022
Technical Report

United States	721	China	476	Germany	186
United Kingdom	160	France	100	Italy	72
Canada	63	Japan	56	Netherlands	56
Brazil	42	Spain	37	India	36
South Korea	34	Sweden	24	Belgium	23
Mexico	22	Russia	21	Indonesia	20
Poland	20	Turkey	19	Taiwan	17
Israel	16	Australia	14	New Zealand	13
South Africa	13	Argentina	10	Denmark	10
Austria	9	Hungary	9	Malaysia	8
Chile	7	Ireland	7	Bangladesh	6
Czech	6	Portugal	6	Romania	6
Croatia	5	Finland	5	Singapore	5
Slovakia	5	Ukraine	5	Bulgaria	4
Colombia	4	Egypt	4	Greece	4
Luxembourg	4	Pakistan	4	Phillipines	4
United Arab Emirates	4	Costa Rica	3	Lithuania	3
Puerto Rico	3	Thailand	3	Vietnam	3
Yemen	3	Albania	2	Algeria	2
Ecuador	2	Estonia	2	Georgia	2
Honduras	2	Iceland	2	Iran	2
Kenya	2	Latvia	2	Malta	2
Moldova	2	Morocco	2	Nigeria	2
Serbia	2	Tunisia	2	Afghanistan	1
Bahamas	1	Bosnia and Herzegovina	1	Dominican Republic	1
El Salvador	1	Ethiopia	1	Iraq	1
Kuwait	1	Libya	1	Macao	1
Macedonia	1	Maldives	1	Mauritius	1
Myanmar	1	Nicaragua	1	Oman	1
Panama	1	Papua New Guinea	1	Peru	1
Qatar	1	Seychelles	1	Slovenia	1
Syria	1	Tanzania	1	Uganda	1
Venezuela	1				

Table 1: Downloads for EPIC-KITCHENS dataset, in 2022, by country

lenges. In January 2022, we started the official challenge phase for 5 challenges available in CodaLab and along with each challenge we released codebase with pre-trained models, features and evaluation scripts:

- Action Recognition at https://github.com/epic-kitchens/C1-Action-Recognition: Five pre-trained models were made available using the codebases: TSN, TRN, TBN, TSM and SlowFast, as well as evaluation script.
- Action Detection at https://github.com/epic-kitchens/C2-Action-Detection: with pre-extracted features, a baseline using BMN model and evaluation script.
- Action Anticipation at https://github.com/epic-kitchens/C3-Action-Anticipation with pre-extracted features, RULSTM base model and evaluation script.
- Unsupervised Domain Adaptation for Recognition at https://github.com/epic-kitchens/C4-UDA-for-Action-Recognition with pre-extracted audio-visual features, TA3N model and evaluation script.
- Multi-Instance Retrieval at https://github.com/epic-kitchens/C5-Multi-Instance-Retrieval with features, JPoSE model and evaluation script.

Recall that each submission is requested to provide their level of supervision following the proposed Supervision Levels Scale (SLS) [11]. We next present the findings for each challenge and introduce the enclosed reports.

Action Recognition	11	8	64▼
Action Anticipation	19	16	138▲
Action Detection	7	7	91▲
UDA for Recognition	12	8	103▼
Multi-Instance Retrieval	9	5	32▲

Table 2: Number of registered teams, active teams and submissions on CodaLab for the five challenges

3. Action Recognition Challenge

The Action Recognition challenge has been running since 2019. In both train and test sets, the start and end times of an action are given. Correct recognition of the action includes correctly recognising the 'verb' class and the 'noun' class. Table 3 shows the entries on the challenge leaderboard for 2022. Methods are ranked based on top-1 action accuracy (noted by arrow), which was used to decide on the overall rank.

Only two submissions warranted the top winning spots. As shown in the table, other submissions could not outperform last year's winning team. Best performing method in 2022 improved over last year's winning entry by by $+0.3\%$, $+7.0\%$ and $+4.1\%$ for VERB, NOUN and ACTION Top-1 Accuracy, respectively. Of particular note is the marginal progress on the verb classification compared to the significant jump on nouns. This becomes clearer when analysing the pretraining flags for these entries. The top-ranked Google Research entry in fact reports an SLS-PT score of 5. This indicates a private large-scale dataset was used for the model's pre-training which gives this approach a significant advantage in recognising nouns primarily. In comparison, the second-ranked model only uses public datasets such as Kinetics [3] for pre-training, but relies on neighbouring action context and a language model. We describe the contributions of each of the teams, based on their technical reports.

3.1. Technical Reports

Technical reports for the Action Recognition challenge, in order of their overall rank on the public leaderboard, are: **Google Research (Rank 1)** is the top ranking entry. This work employs a multimodal transformer of images, optical flow and audio spectograms. Each modality is tokenised using 3D patches (or tubelets) of various sizes. The proposed approach is accordingly referred to as "Multi-view". To train the transformer, a large number of data augmentation approaches were incorporated and Table 1 in the report

Rank	Team	Submissions		SLS			Overall%			Unseen%			Tail%		
		Entries	Date	PT	TL	TD	VERB	NOUN	ACTION▲	VERB	NOUN	ACTION	VERB	NOUN	ACTION
4	**Google Research**	8	06/01/22	**5.0**	**3.0**	**4.0**	**70.9**	**66.2**	**52.8**	**64.5**	**61.6**	**44.8**	**39.7**	**43.7**	**28.5**
4	**Oxford+Bristol**	8	06/01/21	**2.0**	**3.0**	**4.0**	**70.7**	**63.5**	**50.9**	**64.5**	**58.2**	**42.6**	**37.8**	**37.3**	**25.3**
3	SCUT-JD	24	06/01/21	2.0	3.0	4.0	70.6	59.2	48.7	63.5	52.7	39.8	36.1	30.3	22.2
4	CNUS-HUST-THU-Alibaba	27	05/30/21	2.0	3.0	4.0	69.3	60.3	48.5	62.9	54.1	39.5	34.0	33.1	22.7
5	TCN	1	12/20/21	2.0	3.0	3.0	67.9	60.0	46.8	61.1	55.2	39.0	35.2	34.7	22.8
6	SAIC-FBK-UB	8	05/29/21	2.0	3.0	4.0	68.2	55.5	44.8	62.0	50.6	37.5	34.6	25.9	19.0
7	CTS-AI	13	06/01/22	2.0	3.0	4.0	65.0	53.1	42.8	57.2	47.9	35.1	25.6	20.3	16.2
8	MEITUAN	19	06/01/22	2.0	3.0	3.0	66.1	54.0	40.4	59.4	45.6	33.1	34.5	28.5	10.5
9	EPIC_TSM_FUSION	2	10/10/21	2.0	3.0	4.0	65.3	47.8	37.4	59.7	42.5	30.6	30.0	17.0	13.5
10	EPIC_SLOWFAST_RGB	3	01/14/21	2.0	3.0	4.0	63.8	48.6	36.8	57.7	42.6	29.3	29.7	17.1	13.5
11	EPIC_TBN_FUSION	7	01/27/21	2.0	3.0	4.0	62.7	47.6	35.5	56.7	43.7	29.3	31.0	19.5	14.1
12	EPIC_TRN_FUSION	2	10/10/20	2.0	3.0	4.0	63.3	46.2	35.3	57.5	41.4	29.7	28.2	14.0	12.2

Table 3: Results on EPIC-KITCHENS-100 Action Recognition challenge - 1 June 2022

Rank	Team	Submissions		SLS			Overall%			Unseen%			Tail%		
		Entries	Date	PT	TL	TD	VERB	NOUN	ACTION▲	VERB	NOUN	ACTION	VERB	NOUN	ACTION
1	**SCUT**	7	06/01/22	**2.0**	**3.0**	**3.0**	**37.91**	**41.71**	**20.43**	**27.94**	**37.07**	**18.27**	**32.43**	**36.09**	**17.11**
2	**NVIDIA-UNIBZ**	26	06/01/22	**1.0**	**3.0**	**4.0**	**29.67**	**38.46**	**19.61**	**23.47**	**35.25**	**16.41**	**23.48**	**31.11**	**16.63**
3	**ICL-SJTU**	22	06/01/22	**2.0**	**4.0**	**4.0**	**41.96**	**35.74**	**19.53**	**33.35**	**26.80**	**15.85**	**41.01**	**33.22**	**16.87**
4	PCO-PSNRD	7	05/30/22	2.0	4.0	3.0	30.85	41.32	18.68	25.65	35.39	16.32	24.99	35.40	16.14
10	AVT-FB-UT	13	06/01/21	2.0	4.0	4.0	25.25	32.04	16.53	20.41	27.90	12.79	17.63	23.47	13.62
11	Panasonic_CNSIC_PSNRD	10	05/27/21	1.0	4.0	3.0	30.38	33.50	14.82	21.08	27.11	10.21	24.57	27.45	12.69
13	ICL-SJTU	4	06/01/21	1.0	4.0	3.0	36.15	32.20	13.39	27.60	24.24	10.05	32.06	29.87	11.88
16	RULSTM-FUSION	1	09/30/20	1.0	4.0	3.0	25.25	26.69	11.19	19.36	26.87	9.65	17.56	15.97	7.92
17	EPIC_CHANCE_BASELINE	1	09/30/20	0.0	1.0	3.0	6.17	2.28	0.14	8.14	3.28	0.31	1.87	0.66	0.03

Table 4: Results on EPIC-KITCHENS-100 Action Anticipation challenge - 1 June 2022

details the hyperparameters for these augmentations. The model is initialised from the private large-scale web dataset (WTS) [21], hence the submission is correctly tagged with the SLS-PT score of 5, indicating a private dataset was used for pre-training. 10 Different models were trained to form the ensemble. An ablation of the various models is included in the report.

Oxford+Bristol (Rank 2) builds on a prior model [18], noted in Table 3 on Rank 5 (TCN). The approach offers novel modifications that achieve significant improvement on the test sets, improving the top-1 action recognition by 4.1%. Different from [18], temporal context is modelled by surrounding frames/clips rather than by nearby actions. The model considers two input modalities - RGB and Audio, as well as a language model that refines predictions based on probable sequences, learnt from training. The final submission is an ensemble of 16 models of varying context durations and number of clips.

As noted earlier, only two submissions outperformed last year's winning entry. Accordingly, only two teams were awarded prizes.

Meituan (Rank 4 (2022), Rank 8 (overall)) utilises spatial detections of hands and objects to build an ensemble of gloabl (full-image) as well as local (hands + objects) detectors. Additionally, the prior probability of actions as combinations of valid verbs and nouns are considered in the learning.

4. Action Anticipation Challenge

The 2022 edition of the Action Anticipation challenge has been set similarly to the past editions. Methods were asked to predict upcoming action happening after 1 second from the observed video segment. Predictions follow the same format as that of the recognition challenge, i.e., the participants provided prediction scores for verbs, nouns and actions. Table 4 shows the results achieved by the participants, along with the public leaderboard rankings. The top-3 submissions are highlighted in bold. Lines highlighted in green report the results of the winners of the CVPR 2021 competition. Shaded lines reflect the baseline models. All submissions outperformed the baselines and winners from the past edition of the challenge. Overall, the submissions have improved over the previous models by $+8.95\%$, $+6.22\%$ and 3.49% for VERB, NOUN and ACTION Overall Mean Top-5 Recall.

We next summarise the contributions of the participants based on their technical reports.

4.1. Technical Reports

Technical reports for the Action Anticipation challenge, in order of their overall rank on the public leaderboard, are:
SCUT (Rank 1) The method is based on a Causal Transformer Decoder. The performance of the baseline are improved introducing two main modules: an anticipation time knowledge distillation module and a verb-noun relation module. The distillation module uses a teacher transformer observing the whole video (past and future) to encourage

the teacher to fill the representation gap due to the unobserved future by using learned future embeddings. The verb-noun relation module is used to allow for interaction between the inferred future verb and the nouns in the observed video. Different backbones are used for feature extraction and an ensemble of 10 models is used to obtain the final predictions.

NVIDIA-UNIBZ (Rank 2) The method is an ensemble of instances of two main models: Higher Order Recurrent Space-time Transformer (HORST) and Message-Passing Neural Network with Edge Learning (MPNNEL). The HORST model processes features extracted with a 2D-CNN backbone, which are used as a query and cross-reference from historical states via a space-time decomposition attention. MNPNNEL is based on a graph structure, where the topology is inferred from the input at each time step. Multi-head attention is used for information routing between vertices. Training is performed in four stages aimed to build a strong feature extractor, focus on the anticipation task, distinguish between hard samples and tail classes. A final training stage finetunes the model on the combination of the training and validation set. Classes are weighted to cope with the long-tail distribution problem.

ICL-SJTU (Rank 3) The method is based on the TransAction architecture. The approch is designed to cope with the long-tail distribution of EPIC-KITCHENS-100 and the domain shift caused by the variability in human behavior, camera viewpoint and scene settings which leads to intra-class inter-domain variations. To deal with the domain shift, the approach considers data from the same subject at the same time as a single domain. Prototype learning is hence used to tackle domain shift and external knowledge in the form of word semantic embeddings is used to regularize learning through a contrastive loss. To make training robust to the long-tail distribution, the authors propose a two-stage training in which the model is first trained using standard cross entropy, then finetuned with class reweighting. Model ensembling is further used to improve results.

PCU-PSNRD (Rank 4) The method described in this submission is based on a Video Swin Transformer baseline, which is trained to account for the long-tail distribution characterizing the EPIC-KITCHENS-100 dataset and to provide better generalization via augmentations and model ensembling. Specificlaly, the LDAM and Logit Adjustments techniques are used to cope with the long-tail distribution, RandAugment-T is used to provide data augmentations, and model ensembling is tuned using the Optuna framework.

5. Action Detection Challenge

The Action Detection challenge has been set similarly to the 2021 challenge editions and follows similar challenges in action detection [15]. Participants have been instructed to consider the test videos as untrimmed, i.e., no temporal segment annotations can be used at test time. The goal is to detect all action instances within the untrimmed video, as in [17].

Participants provided the detected temporal segments for each test video, along with the predicted verb and noun. Results are reported using mean Average Precision (mAP) considering different Intersection over Union (IoU) thresholds ranging from 0.1 to 0.5. Results are reported on the whole test set. Table 5 shows the results achieved by the participants, along with the public leaderboard ranking. Methods are ranked by Average ACTION mAP. The Top-3 submissions among the participants are highlighted in bold. Green lines report the results of the winner of the past edition. Shaded lines reflect the baseline model. All submissions outperformed the baselines, while the top-2 submissions outperformed previous winners. Overall, the submissions have improved over the past methods by $+6.98\%$, $+3.96\%$ and $+5.17\%$ for VERB, NOUN and ACTION Avarage mAP.

We next summarise the contributions of the participants based on their technical reports.

5.1. Technical Reports

Technical reports for the Action Detection challenge, in order of their overall rank on the public leaderboard, are:

Alibaba (Rank 1). This method proposes a one-stage action detection method based on transformers. Clip-based features are first extracted using pre-trained video encoders. Features are added to clip embeddings and aggregated through a transformer encoder. Detection heads predict a fixed number of N action segments, each including a verb prediction, a noun prediction, and a pair of starting and end times. Focal losses are used for verbs and nouns, while 1D IOU losses are used to regress start and end times.

4Paradigm-UWMadison-NJU (Rank 2). This report presents results obtained with ActionFormer [24], a transformer-based action localisation method proposed by the same authors. The report shows that combining features extracted from very different architectures is beneficial for the localisation task. Specifically, the authors use both a fully convolutional model (SlowFast R101-NL [14]) and a Transformer (ViViT [1]). An interesting finding of this work is that following the two-stream architecture to predict verb and noun separately works better than attaching a verb and a noun head to a single Action-Former model.

CTC-AI (Rank 3). The described approach extracts clip features using SlowFast [14] and TimeSformer [2]. Features are then summed to positional embeddings and encoded via a neighbourhood-window-attention and a multi-head-in-head transformer to model the relationship between and within action clips. Similar to ActionFormer [24], this method combines multiscale-feature representations with

Rank	Team	Submissions		SLS			Mean Average Precision (mAP)					
		Entries	Date	PT	TL	TD	Task	@0.1	@0.2	@0.3	@0.4	@0.5	Avg.▲
1	**Alibaba**	18	06/01/22	2.0	3.0	4.0	VERB	**30.67**	**29.40**	**26.81**	**24.34**	**20.51**	**26.35**
							NOUN	**30.96**	**29.36**	**26.78**	**23.27**	**18.80**	**25.83**
							ACTION	**24.57**	**23.50**	**21.94**	**19.65**	**16.74**	**21.28**
2	**4Paradigm-UWMadison-NJU**	19	06/01/22	2.0	3.0	4.0	VERB	**26.97**	**25.91**	**24.21**	**21.77**	**18.47**	**23.47**
							NOUN	**28.61**	**27.14**	**24.92**	**22.14**	**18.69**	**24.30**
							ACTION	**23.90**	**22.98**	**21.37**	**19.57**	**16.94**	**20.95**
3	Alibaba-MMAI-Research	7	05/18/21	2.0	3.0	3.0	VERB	22.77	22.01	19.63	17.81	14.65	19.37
							NOUN	26.44	24.55	22.30	19.82	16.25	21.87
							ACTION	18.76	17.73	16.26	14.91	12.87	16.11
4	**CTC-AI**	25	06/01/22	2.0	3.0	4.0	VERB	**22.62**	**21.73**	**20.68**	**17.74**	**15.16**	**19.58**
							NOUN	**20.65**	**19.58**	**18.34**	**16.18**	**12.88**	**17.52**
							ACTION	**16.68**	**16.11**	**15.15**	**13.59**	**11.66**	**14.64**
5	Bristol-MaVi	22	06/01/22	2.0	3.0	4.0	VERB	25.33	23.99	21.91	19.61	17.08	21.58
							NOUN	18.99	17.87	16.41	14.43	11.36	15.81
							ACTION	14.71	13.98	12.86	11.56	9.85	12.59
7	LocTransformer	4	05/22/21	2.0	3.0	3.0	VERB	18.26	17.36	16.10	12.52	10.36	14.92
							NOUN	15.97	14.60	13.09	10.94	8.37	12.60
							ACTION	8.77	8.04	7.40	6.31	5.07	7.12
8	EPIC_BMN_SLOWFAST	1	01/10/21	2.0	3.0	3.0	VERB	11.10	9.40	7.44	5.69	4.09	7.54
							NOUN	11.99	8.49	6.04	4.10	2.80	6.68
							ACTION	6.40	5.37	4.41	3.36	2.47	4.40

Table 5: Results on EPIC-KITCHENS-100 Action Detection challenge - 1 June 2022

local self-attention and uses a decoder to classify each point in time and estimate action boundaries.

Bristol-MaVi (Rank 4). This method proposes an anchor-free approach using SlowFast [14] features and the transformer-based ActionFormer [24] framework. The key component of the method is the Gaussian Boundary Mechanism, where an additional head predicts the confidence score of the start/end bounds produced by the model. Confidence scores are modelled with a Gaussian. The Gaussian Boundary Mechanism improves the ranking of the candidate action segments, which in turn boosts the localisation performance of the framework.

6. Unsupervised Domain Adaptation for Recognition Challenge

The Unsupervised Domain Adaptation for Recognition challenge follows the same task as the Action Recognition, however, the labelled videos available during training (source) are collected two years before the videos for testing (target). Due to the different recording times, there is a domain gap between source and target. The different cameras used, the change in location of participants and the differing tools and activities in the domains, are all factors that contribute to the drop in performance when testing on target instead of source. The goal of this challenge is to improve action recognition performance on target with the addition of unlabelled target data during training. This reduces annotation cost as it is assumed unlabelled data is cheap to collect in the target domain compared to annotation.

Table 6 shows the results achieved from the participants. The winning entry improves over last year's methods by 5.1% top-1 action accuracy. The majority of submissions did not submit predictions for Source Test, which were optional for submission on CodaLab. This would have provided additional insights into how much each submission improves action recognition in general compared to overcoming the domain gap. We encourage next year's submissions to consider providing the Source Test scores.

6.1. Technical Reports

The technical reports for the Unsupervised Domain Adaptation for Recognition challenge, in order of their overall rank on the public leaderboard, are given in this section. Most solutions exploited multiple modalities for domain adaptation, while the best performing solutions used additional backbone architectures compared to the baselines which used TBN, and incorporate prior knowledge from action recognition model and co-occurrence matrix of verb and noun.

VI-I2R (Rank 1) The key idea of this method is to introduce prior knowledge from action recognition model to disentangle the action-aware source features for alignment with target features. In addition, the target action prediction results are further refined by co-occurrence matrix of verb and noun to eliminate the impossible actions. For the video presentation, pre-trained slowfast [14] is employed to obtain frame-level features and Graph Convolutional Network is used for temporal relation modelling to get the video-level features.

Audio-Adaptive-CVPR2022 (Rank 2) This method proposes a idea to enhance visual features by leveraging audio

Rank	Team	Submissions		SLS			Target Top-1 Accuracy (%)			Target Top-5 Accuracy (%)		
		Entries	Date	PT	TL	TD	VERB	NOUN	ACTION▲	VERB	NOUN	ACTION
1	VI-I2R	30	06/01/22	2.0	4.0	3.0	57.89	40.07	30.12	83.48	64.19	48.10
2	Audio-Adaptive-CVPR2022	4	05/12/22	2.0	3.0	3.0	52.95	42.26	28.06	80.03	67.51	44.03
3	plnet	37	05/29/22	2.0	3.0	3.0	55.51	35.86	25.25	82.77	60.65	40.09
4	CVPR2021-chengyi	1	01/17/22	2.0	3.0	3.0	53.16	34.86	25.00	80.74	59.30	40.75
5	CVPR2021-M3EM	1	01/17/22	2.0	3.0	3.0	53.29	35.64	24.76	81.64	59.89	40.73
6	CVPR2021-plnet	1	01/17/22	2.0	3.0	3.0	55.22	34.83	24.71	81.93	60.48	41.41
7	Nie-Lin	6	06/01/22	2.0	3.0	3.0	48.87	28.72	19.88	74.61	49.70	32.32
8	EPIC_TA3N	1	01/17/22	2.0	3.0	3.0	46.91	27.69	18.95	72.70	50.72	30.53
9	EPIC_TA3N_SOURCE_ONLY	1	01/17/22	2.0	3.0	3.0	44.39	25.30	16.79	69.69	48.40	29.06

Table 6: Results on the Unsupervised Domain Adaptation for Recognition challenge - 1 June 2022

to learn more domain-invariant and discriminative features in the target domain. Audio-infused recognizer is proposed to fuse information in audio and visual modalities and reduce the impact of domain-relevant visual features.

plnet (Rank 3) extends Relative Norm Alignment network (RNA-Net) [20] with optical flow, then introduces frame-level and video-level adversarial alignment between source and target domains, with adding attentive entropy loss to decrease the uncertainty of target prediction. In addition, Multiple Spatio-Temporal Adversarial Alignment (MSTAA) is proposed to reduce environmental bias. Finally, Min-Entropy Consistency (MEC) and Complement Entropy (CENT) are utilized to encourage consistency among different models and reduce the effect o uncertain predictions.

Nie-Lin (Rank 7) The main idea of this approach is to introduce a learnable patch selection method for domain adaptation. The selected patches focus on the local information, by incorporating the global information in whole frames, performance gain is achieved.

7. Multi-Instance Retrieval Challenge

This is the second year that the Multi-Instance Retrieval challenge has ran as part of the EPIC-KITCHENS-100 challenges. Apart from the ranking of the methods according to the evaluation metrics, the challenge is unchanged from the previous year. Details of the challenge can be found below: Given a query video segment, the goal of video-to-text retrieval is to rank captions in a gallery set, C, such that those with a higher rank are more semantically relevant to the action in the query video segment. On the contrary, the goal of text-to-video retrieval is to rank videos given a query caption $c_i \in C$. Differently from the other retrieval challenges, where captions are considered relevant if and only if they were collected for the same video, in this challenge the class knowledge proxy measure introduced in [7] and [22] is used to define caption relevancy (e.g. "put glass" and "place cup" are considered semantically relevant).

Video-to-text and text-to-video results are reported using mean Average Precision (mAP) and normalised Discounted Cumulative Gain (nDCG) on the whole test set. For this year, the results are ranked based on a combination of average mAP and nDCG performance, differing from last year which ranked solely on average nDCG. This led towards joint placed winners for both 1st and 3rd place as methods tended to perform better on one of the two metrics. The metrics both evaluate the ranking of the gallery set and allow for multiple correct retrievals, however, they differ in that mAP only allows for a binary relevancy, i.e. a gallery item is either considered relevant or not, whereas nDCG assigns a continuous relevance score for each gallery item, allowing for differing levels of relevancy to be taken into account within the ranking. This represents an interesting direction to look into for future work to discover why methods may perform better/worse on one of the metrics.

Table 7 shows the public results achieved by the participants. The joint Top-1 winning submissions are highlighted in bold, whereas shaded lines indicates the baselines models. The best submission(s) outperformed the baseline MI-MM [6] by a significant margin: $+21.31\%$, $+23.06\%$ and $+22.19\%$ for T2V, V2T and Average mAP and $+18.40\%$, $+19.44$ and $+18.92\%$ for T2V, V2T and Average nDCG. Note that, for this edition of the challenge, we only included the Mi-MM baseline as JPoSE was found to be a very hard baseline to beat last year, due to the chosen features for training.

We next summarise the contribution of the participants based on their technical reports.

7.1. Technical Reports

The technical reports of the submission of the Multi-Instance Retrieval challenge are:

UniUD-UB-UniBZ (Rank 1) The authors proposed an ensemble method of different models (JPoSE [23] and HGR [4]) trained with two relevance-augmented versions of the triplet loss. The JPoSE model is trained with a margin m which is proportional to the relevance value of the video and caption descriptors that should be contrasted [13]. HGR is trained with the RANP strategy [12] which uses the rele-

Rank	Team	Submissions		SLS			mean Average Precision (mAP)			normalised Discounted Cumulative Gain (nDCG)		
		Entries	Date	PT	TL	TD	T2V	V2T	Avg.▲	T2V	V2T	Avg.▲
=1	**afalcon**	3	06/01/22	**2.0**	**3.0**	**3.0**	**44.39**	**55.15**	**49.77**	58.88	**63.16**	**61.02**
=1	**kevin.lin**	3	05/30/22	3.0	**3.0**	**3.0**	40.95	53.85	47.39	**59.60**	**63.29**	**61.44**
=3	buraksatar	12	05/26/22	**2.0**	**3.0**	**3.0**	38.10	47.52	42.81	54.12	56.55	55.33
=3	haoxiaoshuai	11	05/31/22	**2.0**	**3.0**	**3.0**	38.34	49.69	44.02	51.31	54.82	53.06
4	MI-MM	4	12/10/21	2.0	3.0	3.0	23.08	32.09	27.58	40.48	43.72	42.10

Table 7: Results on EPIC-KITCHENS-100 Multi-Instance Retrieval challenge - 1 June 2022

vance function and a threshold τ to separate relevant from irrelevant samples within the batch.

Ego-VLP (Rank 1) The idea is to use a video-language model (VLP) [19] pre-trained on a clean subset of Ego4D [16] that is able to transfer its video-text representation for the MIR task. The authors proposed a modified version of the MI-MM loss [23] to fine-tune the model where the margin m is not fixed. Moreover, a dual-softmax technique [5] is used in inference to scale the similarities and filter out hard cases.

IIE-MRG (Rank 3) The proposed Cross-Modal Alignment Network (CMAN) explores the similarity information of different modalities by a semantic alignment and the bi-directional ranking loss. In addition, they examining the similarities between instances of the same modality which are exploited by the intra-modal alignment.

NTU-A*STAR (Rank 3) The method proposes a modified version of JPoSE [23] where a self-attention layer is added in the video branch to exploit contextualised visual features.

8. 2022 Challenge Winners

Accordingly, Table 8 details the winners of the 2022 EPIC challenges, announced as part of EPIC@CVPR2022 hybrid workshop. A capture of the certificate awarding ceremony for the 2022 challenges also in Fig 2—showcasing both in-person and online winning teams.

		Team	Member	Affiliations
Action Recognition	①	Google Research	Xuehan Xiong	Google Research
		(xxiong)	Anurag Arnab	Google Research
			Arsha Nagrani	Google Research
			Cordelia Schmid	Google Research
	②	Oxford-Bristol	Jaesung Huh	VGG, University of Oxford
		(Jaesung)	Evangelos Kazakos	University of Bristol
			Jacob Chalk	University of Bristol
			Dima Damen	University of Bristol
			Andrew Zisserman	VGG, University of Oxford
Action Anticipation	①	SCUT	Zeyu Jiang	South China University of Technology
		(hrgdscs)	Changxing Ding	South China University of Technology
	②	NVIDIA-UNIBZ	Tsung-Ming Tai	NVIDIA, Free University of Bozen-Bolzano
		(corcovadoming)	Oswald Lanz	Free University of Bozen-Bolzano
			Giuseppe Fiameni	NVIDIA
			Yi-Kwan Wong	NVIDIA
			Sze-Sen Poon	NVIDIA
			Cheng-Kuang Lee	NVIDIA
			Ka-Chun Cheung	NVIDIA
			Simon See	NVIDIA
	③	ICL-SJTU	Xiao Gu	Imperial College London
		(Shawn0822)	Yao Guo	Shanghai Jiao Tong University
			Zeju Li	Imperial College London
			Jianing Qiu	Imperial College London
			Benny Lo	Imperial College London
			Guang-Zhong Yang	Shanghai Jiao Tong University
Action Detection	①	Alibaba	Lijun Li	Alibaba
		(lijun)	Li'an Zhuo	Alibaba
			Bang Zhang	Alibaba
	②	4Paradigm-UWMadison-NJU	Chenlin Zhang	4Paradigm, Nanjing University
		(tzzcl1)	Lin Sui	Nanjing University
			Abrar Majeedi	University of Wisconsin-Madison
			Viswanatha Reddy Gajjala	University of Wisconsin-Madison
			Yin Li	University of Wisconsin-Madison
	③	CTC-AI	Xiaodong Dong	China Telecom
		(cuis)	Hao Sun	China Telecom
			Xuyang Zhou	China Telecom
			Qihang Wu	China Telecom
			Shun Cui	China Telecom
			Dong Wu	China Telecom
			Aigong Zhen	China Telecom
UDA for Recognition	①	A*STAR	Yi Cheng	A*STAR, Singapore
		(VI-I2R)	Dongyun Lin	A*STAR, Singapore
			Fen Fang	A*STAR, Singapore
			Hao Xuan Woon	A*STAR, Singapore, NUS
			Qianli Xu	A*STAR, Singapore
			Ying Sun	A*STAR, Singapore
	②	Uni-Amsterdam	Yunhua Zhang	University of Amsterdam
		(Audio-Adaptive)	Hazel Doughty	University of Amsterdam
			Cees Snoek	University of Amsterdam
	③	Torino	Mirco Planamente	Politecnico di Torino, Italy
		(plnet)	Gabriele Goletto	Politecnico di Torino, Italy
			Gabriele Trivigno	Politecnico di Torino, Italy
			Giuseppe Averta	Politecnico di Torino, Italy
			Barbara Caputo	Politecnico di Torino, Italy
Multi-Instance Video Retrieval	①	UniUD-UB-UniBZ	Alex Falcon	University of Udine
		(afalcon)	Giuseppe Serra	University of Udine
			Sergio Escalera	University of Barcelona
			Oswald Lanz	Free University of Bozen-Bolzano
	①	Ego-VLP	Kevin Qinghong Lin	National University of Singapore
		(kevin.lin)	Alex Jinpeng Wang	National University of Singapore
			Rui Yan	National University of Singapore
			Eric Zhongcong Xu	National University of Singapore
			Rongcheng Tu	Tencent Data Platform
			Yanru Zhu	Tencent Data Platform
			Wenzhe Zhao	Tencent Data Platform
			Weijie Kong	Tencent Data Platform
			Chengfei Cai	Tencent Data Platform
			Hongfa Wang	Tencent Data Platform
			Wei Liu	Tencent Data Platform
			Mike Zheng Shou	National University of Singapore
	③	IIE-MRG	Xiaoshuai Hao	Chinese Academy of Sciences
		(buraksatar)	Yufan Liu	Chinese Academy of Sciences
			Wanqian Zhang	Chinese Academy of Sciences
			Dayan Wu	Chinese Academy of Sciences
			Bo Li	Chinese Academy of Sciences
	③	NTU-A*STAR	Burak Satar	A*STAR, NTU, Singapore
		(haoxiaoshuai)	Zhu Hongyuan	A*STAR, Singapore
			Hanwang Zhang	NTU, Singapore
			Joo Hwee Lim	A*STAR, NTU, Singapore

Table 8: Top-3 Winners - 2022 EPIC-KITCHENS-100 challenges

Figure 2: Winners during 10th EPIC@CVPR2022 hybrid Workshop, 20 June 2022.

References

[1] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid. Vivit: A video vision transformer, 2021. 4

[2] G. Bertasius, H. Wang, and L. Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021. 4

[3] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. CVPR*, 2017. 2

[4] S. Chen, Y. Zhao, Q. Jin, and Q. Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition,CVPR*, 2020. 6

[5] X. Cheng, H. Lin, X. Wu, F. Yang, and D. Shen. Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss. *CoRR*, abs/2109.04290, 2021. 7

[6] D. Damen, H. Doughty, G. M. Farinella, , A. Furnari, J. Ma, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 130:33–55, 2022. 1, 6

[7] D. Damen, H. Doughty, G. Maria Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Scaling egocentric vision: The epic-kitchens dataset. In *Proc. ECCV*, 2018. 6

[8] D. Damen, A. Fragomeni, J. Munro, T. Perrett, D. Whettam, M. Wray, A. Furnari, G. M. Farinella, and D. Moltisanti. Epic-kitchens-100- 2021 challenges report. Technical report, University of Bristol, 2021. 1

[9] D. Damen, E. Kazakos, W. Price, J. Ma, H. Doughty, A. Furnari, and G. M. Farinella. Epic-kitchens - 2020 challenges report. Technical report, 2020. 1

[10] D. Damen, W. Price, E. Kazakos, A. Furnari, and G. M. Farinella. Epic-kitchens - 2019 challenges report. Technical report, 2019. 1

[11] D. Damen and M. Wray. Supervision levels scale (SLS). *CoRR*, abs/2008.09890, 2020. 2

[12] A. Falcon, G. Serra, and O. Lanz. Learning video retrieval models with relevance-aware online mining. In *International Conference on Image Analysis and Processing*, pages 182–194. Springer, 2022. 6

[13] A. Falcon, S. Sudhakaran, G. Serra, S. Escalera, and O. Lanz. Relevance-based margin for contrastively-trained video retrieval models. *arXiv preprint arXiv:2204.13001*, 2022. 6

[14] C. Feichtenhofer, H. Fan, J. Malik, and K. He. Slowfast networks for video recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6201–6210, 2019. 4, 5

[15] B. Ghanem, J. C. Niebles, C. Snoek, F. C. Heilbron, H. Alwassel, R. Khrisna, V. Escorcia, K. Hata, and S. Buch. Activitynet challenge 2017 summary. *arXiv preprint arXiv:1710.08011*, 2017. 4

[16] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M. Xu, E. Z. Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, A. Gebreselasie, C. González, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolář, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. Ruiz, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Z. Zhao, Y. Zhu, P. Arbeláez, D. Crandall, D. Damen, G. M. Farinella, C. Fuegen, B. Ghanem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. S. Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba,

L. Torresani, M. Yan, and J. Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18995–19012, June 2022. 7

[17] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. ActivityNet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 4

[18] E. Kazakos, J. Huh, A. Nagrani, A. Zisserman, and D. Damen. With a little help from my temporal context: Multimodal egocentric action recognition. In *British Machine Vision Conference (BMVC)*, 2021. 3

[19] K. Q. Lin, A. J. Wang, M. Soldan, M. Wray, R. Yan, E. Z. Xu, D. Gao, R. Tu, W. Zhao, W. Kong, C. Cai, H. Wang, D. Damen, B. Ghanem, W. Liu, and M. Z. Shou. Egocentric video-language pretraining. *CoRR*, abs/2206.01670, 2022. 7

[20] M. Planamente, C. Plizzari, E. Alberti, and B. Caputo. Cross-domain first person audio-visual action recognition through relative norm alignment. In *arXiv*, 2021. 6

[21] J. Stroud, D. Ross, C. Sun, J. Deng, R. Sukthankar, and C. Schmid. Learning video representations from textual web supervision. *CoRR*, abs/2007.14937, 2020. 3

[22] M. Wray, H. Doughty, and D. Damen. On semantic similarity in video retrieval. In *CVPR*, 2021. 6

[23] M. Wray, D. Larlus, G. Csurka, and D. Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 6, 7

[24] C. Zhang, J. Wu, and Y. Li. Actionformer: Localizing moments of actions with transformers. *arXiv preprint arXiv:2202.07925*, 2022. 4, 5

M&M Mix: A Multimodal Multiview Transformer Ensemble

Xuehan Xiong, Anurag Arnab, Arsha Nagrani, Cordelia Schmid
Google Research
{xxman,aarnab,anagrani,cordelias}@google.com

Abstract

This report describes the approach behind our submission to the 2022 Epic-Kitchens Action Recognition Challenge from team Google Research Grenoble . Our approach builds upon our recent work, Multiview Transformer for Video Recognition (MTV), and adapts it to multimodal inputs. Our final submission consists of an ensemble of Multimodal MTV (M&M) models varying backbone sizes and input modalities. Our approach achieved 52.8% Top-1 accuracy on the test set in action classes, which is 4.1% higher than last year's winning entry.

1. Introduction

Transformers have replaced Convolutional Networks (CNNs) as the de facto backbone for video understanding. The state-of-the-art results on popular datasets (e.g., Kinetics [2], Moments in Time [19], Epic-Kitchens [4], etc) are all obtained using a pure transformer-based approach. Our approach is built upon a very recent state-of-the-art method for video classification, Multiview Transformers for Video Recognition (MTV) [29]. MTV proposed a multi-stream architecture to process video data in a multiscale fashion where each stream takes in different-sized tubelets of RGB frames, however no other modalities (such as sound) were used for making a prediction.

Epic-Kitchens is a large-scale dataset of first-person (egocentric) videos recorded in kitchen environments. Contestants of the Action Recognition challenge are required to predict a verb and a noun for each video clip. Videos in this dataset are multimodal (they contain an audio track) and the egocentric domain consists of rich sounds resulting from the interactions between humans and objects, as well as the proximity of the wearable microphone to the undergoing action. Sound is a hence a discriminative feature for identifying actions [17, 20], for example, the sound of running water provides important cues to predict actions such as "wash glass". Optical flow is another modality that is complementary to RGB frames as shown in previous work [22]. As we will show later in the experiments, this observation remains true for state-of-the-art video transformer models, such as MTV. In this work, we extend MTV to process multimodal inputs where each stream encodes input data from one temporal resolution and from one modality.

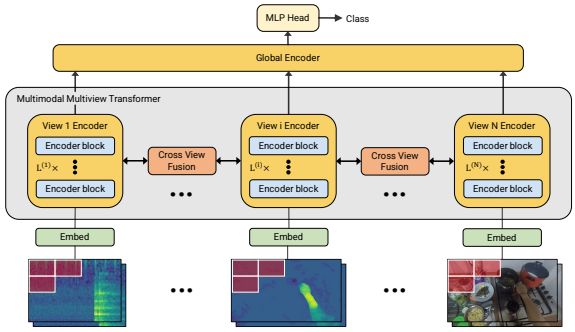


Figure 1. Overview of our Multimodal Multiview Transformer (M&M). The input video consists of three modalities, spectrogram, optical flow, and RGB frames (from left to right) and we create multiple representations or "views" by tokenizing each input modality using tubelets of different sizes. These tokens are fed into separate encoders and further fused through a Cross View Fusion module, and finally aggregated by a global encoder. Note that each encoder can vary in architecture.

2. Multimodal Multiview Transformers

2.1. Background (MTV)

This section presents a brief overview of Multiview Transformers (MTV) [29]. It consists of separate transformer encoders for each view which are connected by lateral connections to fuse cross-view information. A view is defined as a video representation expressed by a set of fixed-sized tubelets. A larger view corresponds to a set of larger tubelets (and thus fewer tokens) and a smaller view corresponds to smaller tubelets (and thus more tokens). Each transformer layer within the encoders follows the same design as the original transformer of Vaswani et al. [27]. Furthermore, within each transformer layer, self-attention is computed only among tokens extracted from the same temporal index, following the Factorised Encoder of [1]. This significantly reduces the computational cost of the model. We chose cross-view attention as the fusion method as it

gives the best performance as shown in [29]. Finally, the classification tokens from each view are extracted and processed with another transformer encoder that aggregates information from all views.

2.2. M&M

The overall architecture of M&M (shown in Figure. 1) remains the same as MTV except for the input tokenization step. In this example, the input video has three modalities, RGB, optical flow, and short-term magnitude spectrograms derived from audio. For each modality, we can have multiple representations or "views" by tokenizing the frames from this modality using different tubelet sizes. An alternative design is to use a single encoder that takes in tokens from all modalities [11, 16, 20]. Our design of utilizing a separate encoder for each multimodal view is more flexible. As Yan et al. [29] have shown, it is sufficient to use a smaller encoder to learn representations from larger views of RGB frames. Feichtenhofer et al. [10] applied a smaller CNN (e.g., a smaller number of channels) to learn motion information and a larger one for encoding the semantics of frames. One advantage of our design is that our architecture also supports multiscale processing within each modality.

3. Experiments

3.1. Experimental setup

Model notation For the backbone of each view, we consider four ViT variants, "Tiny", "Small", "Base", and "Large". Their settings strictly follow the ones defined in BERT [7] and ViT [8], *i.e.* number of transformer layers, number of attention heads, hidden dimensions. For convenience, each model variant is denoted with the following abbreviations indicating the backbone size, tubelet length, and input modality. For example, B/2:R+S/4:S+Ti/8:F denotes a three-view model, where a "Base", "Small", and "Tiny" encoders are used to processes tokens from RGB tubelets of sizes $16 \times 16 \times 2$, spectrogram tubelets of sizes $16 \times 16 \times 4$, and optical flow tubelets of sizes $16 \times 16 \times 8$, respectively. Note that we omit 16 in our model abbreviations because all our models use 16×16 as the spatial tubelet size following ViT [8]. If we omit the modality in the notation, we assume all views use RGB frames as the modality. All model variants use the same global encoder which follows the "Base" architecture, except that the number of heads is set to 8 instead of 12. The reason is that the hidden dimension of the tokens should be divisible by the number of heads for multi-head attention, and the number of hidden dimensions across all standard transformer architectures (from "Tiny" to "Large" [8, 23]) is divisible by 8.

Optical flow and spectrogram extraction We compute optical flow using the FlowNet [9] algorithm. Audio spectrograms are extracted in a similar manner to [13]. All audio is converted to monochannel and resampled to 16kHz. Spectrograms are then extracted using short-term Fourier transforms with a Hann window of 25ms with 15ms hop. The resulting spectrogram is integrated into 64 mel-spaced frequency bins (lower cutoff 125 Hz and upper corner frequency 7500 Hz) and the squared magnitude is extracted. This gives us mel spectrograms of 96×64 bins for 0.96 seconds of audio. For the entire clip, we run the above procedure in a sliding window fashion with a temporal hop of 40ms to align with RGB frame rate (25FPS). Spectrograms are normalized to [-1, 1] before feeding into the model.

Data augmentation	
Random crop probability	1.0
Random flip probability	0.5
Scale jitter probability	1.0
Maximum scale	1.33
Minimum scale	0.9
Colour jitter probability	0.8
Rand augment number of layers [3]	3
Rand augment magnitude [3]	10
Regularisation	
Stochastic droplayer rate [14]	0.1
Label smoothing [25]	0.1

Table 1. Data augmentation and regularization parameters.

Initialization We trained two RGB-only models B/2+S/4+Ti/8 and L/2+B/4+S/8+Ti/16 on WTS [24] and use them to initialize multimodal models. Optical flow images have two input channels and spectrogram images only have one so the initial tubelet embedding layer has a different shape than the pretrained RGB models. To address this issue, we simply average the kernel of the embedding layer along the input channel axis and perform tiling.

Training and inference All models are trained on 64 frames with a temporal stride of 1. In Epic-Kitchens, each video is labeled with a "verb" and a "noun". We predict both categories using a single network with two "heads". We train all our models for 50 epochs with a global batch size of 128 using synchronous SGD with momentum of 0.9 following a cosine learning rate schedule with a linear warm up. The initial learning rates for all models are set to 0.4. We follow [1, 6, 29] and apply the same data augmentation and regularization schemes [3, 14, 25], which were used by [26] to train vision transformers more effectively. For spectrograms we use SpecAugment [21] with a max time mask length of 96 frames and max frequency mask length of 16 bins following MBT [20]. See Table 1 for detailed settings. During single-model inference, we adopt the standard evaluation protocol by averaging over four temporal crops. To produce the final predictions from the model ensemble, we simply average the logits produced by each model.

Pretraining datasets	Top-1 Action	Top-1 Noun	Top-1 Verb
K400	46.7	60.5	67.8
K700	48.0	61.2	69.1
WTS	49.3	63.0	69.4

Table 2. Effects of different pretraining datasets. All models are trained and evaluated on 224×224 crops.

Spatial resolution	Top-1 Action	Top-1 Noun	Top-1 Verb
224p	49.3	63.0	69.4
280p	50.5	63.9	69.9
432p	52.7	66.1	71.2

Table 3. Effects of increasing spatial resolution. All models are finetuned from a WTS-pretrained checkpoint.

3.2. Ablation study

We use a RGB-only model B/2+S/4+Ti/8 for the studies in Table 2 and 3. We report Top-1 accuracies on Action, Noun, and Verb classes obtained from averaging predictions across four temporal crops. All numbers reported in this section are from the validation set.

Effects of pretraining Table 2 presents the finetuning results from models pretrained on Kinetics 400 [15], Kinetics 700 [15], and WTS [24] datasets. Kinetics 400 and 700 consist of 230,000 and 530,000 10s video clips focusing on human actions with each clip labeled with one of the 400 and 700 classes, respectively. WTS contains 60M videos with only video-level labels. All three pretraining datasets are from a different domain than Epic-Kitchens that is composed of egocentric videos. Table 2 shows that it is more beneficial to pretrain on a large-scale weakly supervised dataset than on a smaller set of trimmed video clips.

Effects of input resolution As Table 3 shown, as spatial resolution increases so does top-1 accuracy for nouns. Accuracies for verbs are also improved and this is likely due to the increased number of tokens that help the model better understand motion in the scene.

Effects of combining different modalities The first two rows in Table 4 present the Top-1 accuracies of the RGB-only and the Flow-only models. Changing input modality of the "Small" encoder from RGB to flow and to spectrogram improves Top-1 accuracy on action from 52.7 to 53.4 and 53.2, respectively. Combining all three modalities gives the best performance on action with a score of 53.6. All models share similar FLOPs with the only difference being the initial embedding layers. RGB is the most informative modality for predicting "nouns", there is little gain by adding flow and audio. However, optical flow and audio provide complimentary information to RGB for predicting "verbs".

Models	Top-1 Action	Top-1 Noun	Top-1 Verb
B/2:R+S/4:R+Ti/8:R	52.7	66.1	71.2
B/2:F+S/4:F+Ti/8:F	40.5	50.1	68.1
B/2:R+S/4:F+Ti/8:R	53.4	**66.5**	71.9
B/2:R+S/4:S+Ti/8:R	53.2	66.3	**72.0**
B/2:R+S/4:S+Ti/8:F	**53.6**	66.3	**72.0**

Table 4. Effects of combining different modalities. All models are trained and evaluated on 432×432 crops. As an example of our naming convention, B/2:R+S/4:S+Ti/8:F denotes a three-view model, where a "Base", "Small", and "Tiny" encoders are used to processes tokens from RGB tubelets of sizes $16 \times 16 \times 2$, spectrogram tubelets of sizes $16 \times 16 \times 4$, and optical flow tubelets of sizes $16 \times 16 \times 8$, respectively.

Data split	Models	Top-1 Action	Top-1 Noun	Top-1 Verb
validation	MoViNet [18]	47.7	57.3	**72.2**
	MeMViT [28]	48.4	60.3	71.4
	Omnivore [12]	49.9	61.7	69.5
	M&M-B	**53.6**	**66.3**	72.0
test	[5]	48.7	59.2	**70.6**
	M&M-B	**49.6**	**63.7**	68.0

Table 5. Comparisons to state-of-the-art. M&M-B refers to our three-view multimodal MTV model, B/2:R+S/4:S+Ti/8:F (no ensembling). The gray row is the winning entry from last year's challenge, which uses a 6-model ensemble. All other rows are from a single-model evaluation.

3.3. Comparison to the state-of-the-art

Table 5 compares our best single model to the previous state-of-the-art on the Epic-Kitchens dataset and last year's winning entry of the challenge. Our M&M-B model improves over the previous state-of-the-art [12] by a margin of 3.7% in Top-1 action accuracy and also outperforms last year's winning method [5], which uses a 6-model ensemble.

3.4. Model ensemble

To create the final submission, we generated two model ensembles one for predicting the verbs and the other for nouns. Table 6 lists all individual models used in this challenge and their corresponding performance on the validation set. Table 7 shows which models we used for verbs and nouns. Using this model ensembling strategy, we improve the Top-1 action accuracy from 53.6 (from our single best model) to 56.9 on the validation set. Our final submission scored 52.8 on Epic-Kitchens test set, which is 4.1% higher than last year's winning entry.

4. Conclusions

In this report, we present the approach behind our submission to the 2022 Epic-Kitchens Action Recognition challenge. We proposed M&M, a transformer backbone

Model indices	Model variants	Pretraining datasets	Resolution	Top-1 Action	Top-1 Noun	Top-1 Verb
0	B/2:R+S/4:R+Ti/8:F	WTS → K700	432p	53.4	66.4	71.8
1	B/2:R+S/4:F+Ti/8:R	WTS → K700	432p	53.4	66.5	71.9
2	L/2:R+B/4:F+S/8:F+Ti/16:R	WTS → K700	320p	53.0	66.7	71.1
3	L/2:R+B/4:R+S/8:R+Ti/16:R	WTS	352p	52.6	67.2	69.8
4	B/2:F+S/4:F+Ti/8:F	WTS → K700	432p	40.5	50.1	68.1
5	B/2:R+S/4:R+Ti/8:R (128×1)	WTS	304p	52.4	65.6	71.3
6	L/2:F+B/4:F+S/8:F+Ti/16:F	WTS → K700	352p	40.9	50.6	67.2
7	L/2:R+B/4:F+S/8:S+Ti/16:R	WTS	320p	53.6	67.0	71.7
8	B/2:R+S/4:S+Ti/8:F	WTS	432p	53.6	66.3	72.0
9	B/2:R+S/4:S+Ti/8:R	WTS	432p	53.2	66.3	72.0
10	B/2:R+S/4:R+Ti/8:S	WTS	432p	53.4	66.6	72.0

Table 6. All model variants used in our final ensemble and their respective performance on the validation set. WTS→K700 denotes a pretraining strategy where we first pretrain the model on WTS and then finetune on Kinetics 700. Model 5 is trained and evaluated on 128 frames instead of 64 for all other models.

Model indices	Top-1 Action (val/test)	Top-1 Noun	Top-1 Verb
0,1,2,3,5,6,7,8,9,10	56.9/52.8	69.2/66.2	
4,5,6,7,8,9,10			75.0/70.9

Table 7. Results from our final model ensemble on both validation/test sets. Different sets of models are used for predicting nouns and verbs.

that learns a multimodal and multiscale representation of videos. Our final submission is an ensemble of M&M models with varying backbone sizes and modality mixes. It scored 52.8 in top-1 accuracy on action classes on the test set, which is 4.1% higher than the last year's winner.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021. 1, 2

[2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 1

[3] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. In *NeurIPS*, 2020. 2

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. In *IJCV*, 2021. 1

[5] Dima Damen, Adriano Fragomeni, Jonathan Munro, Toby Perrett, Daniel Whettam, Michael Wray, Antonino Furnari, Giovanni Maria Farinella, and Davide Moltisanti. Epic-kitchens-100- 2021 challenges report. Technical report, University of Bristol, 2021. 3

[6] Mostafa Dehghani, Alexey Gritsenko, Anurag Arnab, Matthias Minderer, and Yi Tay. Scenic: A JAX library for computer vision research and beyond. In *arXiv preprint arXiv:2110.11403*, 2021. 2

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 2

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[9] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2758–2766, 2015. 2

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

[11] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *European Conference on Computer Vision*, pages 214–229. Springer, 2020. 2

[12] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. *arXiv preprint arXiv:2201.08377*, 2022. 3

[13] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135, 2017. 2

[14] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016. 2

[15] 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. In *arXiv preprint arXiv:1705.06950*, 2017. 3

[16] Evangelos Kazakos, Jaesung Huh, Arsha Nagrani, Andrew Zisserman, and Dima Damen. With a little help from my

temporal context: Multimodal egocentric action recognition. *BMVC*, 2021. 2

[17] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5492–5501, 2019. 1

[18] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. Movinets: Mobile video networks for efficient video recognition. In *CVPR*, 2021. 3

[19] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Tom Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. Moments in time dataset: one million videos for event understanding. In *PAMI*, 2019. 1

[20] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. In *NeurIPS*, 2021. 1, 2

[21] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779*, 2019. 2

[22] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014. 1

[23] Andreas Steiner, Alexander Kolesnikov, , Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? Data, augmentation, and regularization in vision transformers. In *arXiv preprint arXiv:2106.10270*, 2021. 2

[24] Jonathan C. Stroud, Zhichao Lu, Chen Sun, Jia Deng, Rahul Sukthankar, Cordelia Schmid, and David A. Ross. Learning video representations from textual web supervision. In *arXiv 2007.14937*, 2020. 2, 3

[25] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 2

[26] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 2

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1

[28] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. *arXiv preprint arXiv:2201.08383*, 2022. 3

[29] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *CVPR*, 2022. 1, 2

Oxford+Bristol Submission to the Epic-Kitchens-100 Action Recognition 2022 Challenge

Jaesung Huh[1] Evangelos Kazakos[2] Jacob Chalk[2] Dima Damen[2] Andrew Zisserman[1]
[1]Visual Geometry Group, University of Oxford [2]University of Bristol

Abstract

This report presents the technical details of our submission on Epic-Kitchens Action Recognition Challenge 2022 from Oxford+Bristol team. We propose a novel visual-audio transformer model to use the temporal context from the clips outside the action boundaries to boost the performance of the recognition performance. We also introduce a temporal relative localization loss, which enables the model to learn the temporal order of the clip sequences without additional manual labels. For the challenge, we aggregate predictions from the multiple variants of this model, along with the language model to utilize the temporal context within the output modality. Our final submission obtains 50.6% of top-1 action accuracy on the challenge test set, ranked as 2^{nd} on the leaderboard only using publicly available data.

1. Introduction

Video action recognition is the task of understanding what the person is doing within a given temporal boundary. It is one of the fundamental tasks of video understanding and researchers have introduced a plethora of works on diverse datasets such as Kinetics [4]. Epic-kitchens [2] is recently gaining its popularity as the largest egocentric video dataset with the development of VR / AR technologies. However, there are additional particular challenges on this dataset including: (1) the actions are fine-grained (e.g. put spoon), (ii) rapid movement of RGB frames due to the egocentric recording condition, and (3) some actions are relatively short, often less than a second. In order to improve the performance, we can employ *temporal context* by encoding information about the past and future of the ongoing actions. For example, in Figure 1, we can easily infer that the action 'pour water in kettle' will happen between 'open kettle' and 'close kettle'. The objects are sometimes persistent within certain time intervals and there might be the possible relationships between verbs as well.

There have been a few works which consider the temporal context to improve the recognition performance of an ongoing action. [9] utilizes the clips outside the boundaries of the action of interest, but only considers visual modality and does not employ other modalities such as audio or language. The closest work of ours is [5] which leverages visual, audio and language within a sequence of actions. However, it requires the temporal boundaries of the neighbouring actions in an untrimmed video, which are not available in real-world scenarios. Our work does not require any temporal information of actions other than the action of interest, while effectively learn the relations between the neighbouring clips.

In this work, we propose a novel multimodal framework which uses the temporal context within visual, audio and language modalities to improve recognition performance. We also propose *temporal relative localization loss* which enables the network to learn the temporal order of clip sequences without additional supervision. Inspired by [7], we also show that this objective helps the model to learn the temporal relationships between clips effectively.

2. Model

In this report, we define the *temporal context* as the video clips outside the action (segment) of interest, including clips that precede and succeed the action in an untrimmed video. We propose a model which utilizes the visual features and corresponding audio features of these clips as well as the features from the action of interest to improve the recognition performance. The N clips are equally sampled within certain time period t sec both before and after the ongoing action. Visual & auditory features of these clips are ingested into the multimodal transformer with the features inside the ongoing action and produces the 'verb' and 'noun' classes. We also introduce *temporal relative localization loss* which encourages the transformer to learn temporal relations within the clips, both in visual and audio.

In addition to using the temporal context in data stream, we also use the temporal context within the labels of the untrimmed video by employing the language model which is first introduced in [5]. The final submission is made by applying ensemble of different models by varying the time t the number of sampled clips N within this context which is explained in Section 5. Please refer to Figure 2 for the

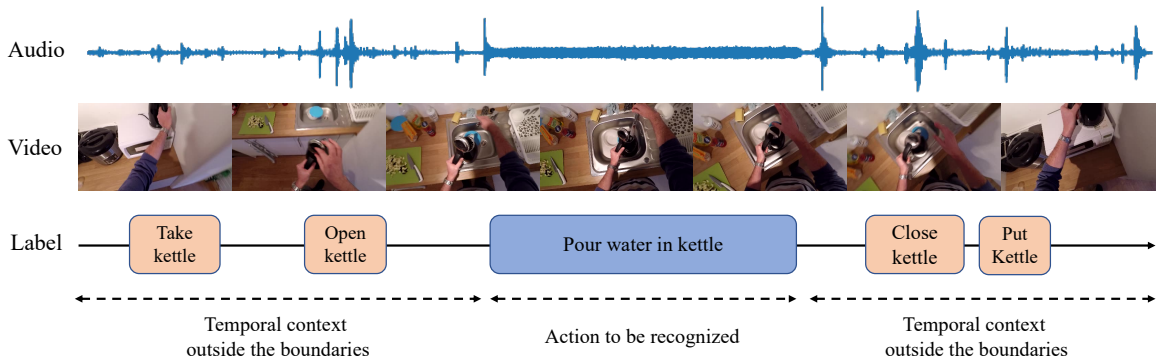


Figure 1. Example of temporal context. Prediction of 'pour water in kettle' is improved by referring to the nearby clips which contain the actions of 'open kettle' and 'close kettle'.

detailed architecture.

2.1. Visual-audio transformer

Let $\tilde{X}_T = [X_{T-N}, \cdots, X_{T+N}]$ be the video clips centered at the current segment at timestamp T within the 'window' size of $2N+1$. X_T is the clip randomly sampled within the temporal boundary of ongoing action, whereas $[X_{T-N}, \cdots, X_{T-1}]$ are N clips equally sampled along the time axis from the t sec before the action and $[X_{T+1}, \cdots, X_{T+N}]$ are clips from the t sec after the action. Let $\tilde{V}_T = [V_{T-N}, \cdots, V_{T+N}]$ be the visual inputs of $\tilde{X}_T$ and $\tilde{A}_T = [A_{T-N}, \cdots, A_{T+N}]$ are corresponding audio inputs.

Encoding Layer Visual and audio encoding layer project $\tilde{V}_T$ and $\tilde{A}_T$ to D-dimensional vectors which serve as inputs to the transformer. Visual-audio transformer then learns the relations between these features and aggregates the temporal context between the clips both inside and outside the action of interest. Since the self-attention operations are permutation-invariant, we add modality-invariant positional embeddings to the encoder outputs to make use of the temporal order of feature sequence. Two modality embeddings m_v, m_a are also added to the encoder outputs respectively to discriminate between visual and audio tokens. Two [CLS] tokens, both for verb and noun, are injected to the transformer for classifying the action.

In addition to the positional and modality embedding, we introduce **center embedding** which allows the network to know the action of interest. We add this learnable vector to visual & audio encodings from the action of interest and also add to two [CLS] tokens to ensure that the network needs to classify the indicated action. We observe that without the center embedding, the network does not know which clip it needs to focus on, and the performance drops drastically.

Transformer and classifier We use a transformer encoder to learn the relations between visual and audio inputs. Transformer blocks share weights to reduce the computational overhead. Output from the [CLS] tokens are fed into the two-head classifier to predict the action.

2.2. Loss function

During training, we use a standard cross-entropy loss L_{ce} for classifying action at the center of our temporal context using the output of the classifier from the [CLS] tokens. In addition, we introduce a novel **temporal relative localization loss** in order to learn the temporal information without additional manual annotation.

Temporal relative localization loss Inspired by [7], we densely sample the feature pairs from the transformer outputs and ask the network to predict the relative distance between two features. Let the $\{\mathbf{v}_i\}$ be the $2N+1$ transformer outputs from the visual inputs and $\{\mathbf{a}_i\}$ be the corresponding audio outputs. $(i = T-N, \cdots, T+N)$ We randomly sample a set of m (e.g. 32) pairs B from these features and concatenate them and put into small MLP $f_t(\cdot)$ to produce the normalized relative distance between two vectors. For each randomly sampled pair $\{e_i, e_j\} \in B$, we can calculate a relative distance $d_{i,j}$ by

$$d_{i,j} = \frac{|i-j|}{2N+1} \tag{1}$$

and compute the L1 loss between the output of $f_t(\cdot)$ and relative distance $d_{i,j}$:

$$L_{loc} = \sum_{\{e_i,e_j\}\in B} |f_t(e_i \oplus e_j) - d_{i,j}| \tag{2}$$

where $\oplus$ indicates concatenation.

The final objective is the weighted sum of two losses with a hyperparameter λ:

$$L_{total} = L_{ce} + \lambda L_{loc} \tag{3}$$

We investigate the effect of λ in Section 4.

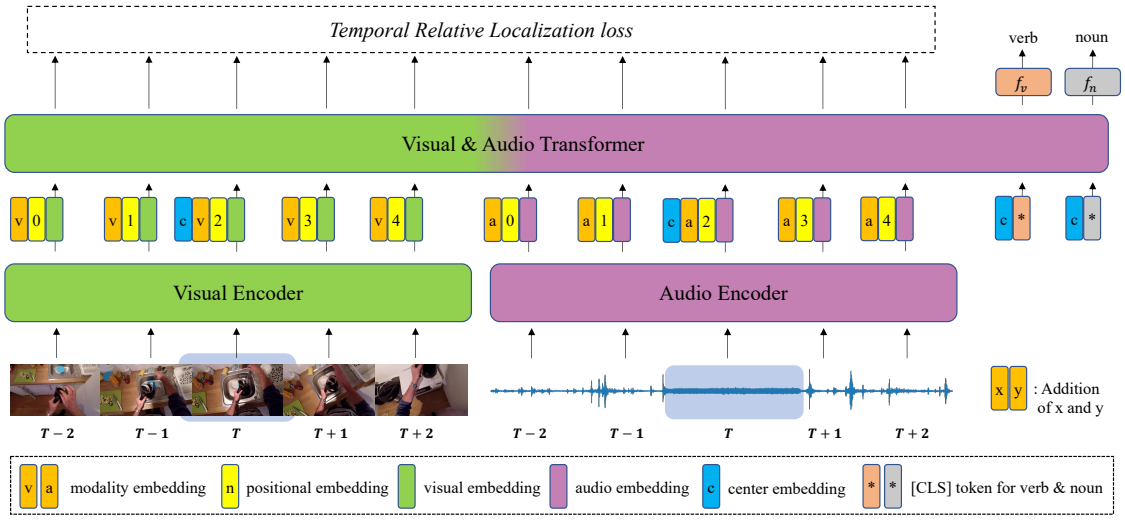


Figure 2. Model overview. The figure illustrates the case when the input is $2N+1=5$ clips at time T. Blue region is the action of interest.

2.3. Language model

We apply the language model to leverage the temporal context between action labels in untrimmed video. We adopt the Masked Language Model (MLM) introduced in [5], which is trained only with the labels of Epic-kitchens-100 train set. Please refer to [5] for more detailed information. We use the released model and inference code. [1]

3. Experiments

In this section, we explain the details of feature extraction, model architecture and training hyperparameters.

Feature extraction We utilize two variants of visual encoder, MotionformerHR [8] and Omnivore [3]. Both works provide the model that are finetuned with Epic-kitchens-100 so we adopt their model and inference code. The former produces $D_v = 768$ dimensional features while the latter produces $D_v = 1024$ dimensional features. For audio encoder, we use the Auditory Slow-Fast [6] pretrained with VGGSound [1] and finetuned with Epic-kitchens train set. The model takes a 2 sec of audio and produces a $D_a = 2304$ dimensional vector.

Model architecture Both visual and audio encoding layers project the input features into 512-dimensional vectors. The visual-audio transformer consists of 4 self-attention encoder layer which share weights to each other. Each layer has 8 attention heads and a hidden unit dimension of 512. We apply the dropout with $p = 0.5$ on encoding layer and $p = 0.1$ within the transformer. All of the additional embeddings, including positional embedding, modality embedding, center embedding and [CLS] tokens, are 512-dimensional learnable vectors. The architecture of language model is identical to the model introduced in [5].

Train/Test details The model is trained with LAMB optimizer [10] with an initial learning rate 0.005. All models are trained for 100 epochs and the learning rate is reduced by a factor of 0.1 at 50 and 75 epochs. We choose the model which provides the best top-1 action accuracy on validation set. We use a batch size of 32 and a weight decay of 0.0005. Mixup [11] with $\alpha = 0.2$ is used as a data augmentation.

During training, we randomly sample 1 clips per action of interest and N clips within t sec both before and after the action uniformly spaced along time axis. For testing, we sample 10 clips equally sampled within the action segment and average 10 corresponding predictions as a final prediction.

4. Results

Table 1 shows the performance of the model by varying the temporal context t and the number of sampled clips N. We also compare the models which use two different visual encoders, MotionformerHR [8] and Omnivore [3]. All of the results are measured in Epic-kitchens validation set except the "challenge submission" which the performance are computed on challenge test set.

In general, MotionformerHR performs slightly better than Omnivore in our implementation. Using $t = 10$ sec and $N = 10$ clips performs best regardless of the visual encoder. The ensemble method boosts the performance by 3% on top-1 action accuracy from the best single model. Applying the language model on top of these action scores results in 0.2% of extra gain on top-1 action accuracy.

Ablation Studies Table 2 shows the influence of temporal relative localization loss by varying the value of λ in Equa-

[1] https://github.com/ekazakos/MTCN

Visual encoder	t (sec)	N	Overall						Unseen Participants			Tail-classes		
			Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-1 Accuracy (%)		
			Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
MformerHR [8]	5	5	71.37	62.79	50.58	91.58	83.77	68.51	65.26	50.80	40.38	42.73	39.32	27.99
	5	10	71.26	62.90	50.74	90.97	82.94	67.90	66.57	50.99	40.66	42.05	37.74	27.92
	5	20	70.51	62.19	49.95	90.60	82.02	66.24	66.20	51.08	40.09	41.53	37.37	26.73
	10	10	71.77	63.55	51.38	91.32	83.83	68.43	65.73	51.55	40.56	44.32	39.95	28.76
	10	20	71.34	63.11	50.73	90.88	83.41	68.15	65.82	51.64	40.56	42.84	38.74	27.15
	50	10	70.43	62.40	49.92	91.32	83.92	67.79	64.41	51.83	39.91	42.84	38.58	27.86
	50	20	71.30	63.21	50.85	91.30	83.69	68.03	65.73	53.05	40.66	43.07	39.37	28.57
	50	50	71.47	63.66	51.27	91.09	83.63	68.16	65.73	51.27	40.09	42.39	39.26	27.73
Omnivore [3]	5	5	71.59	60.04	49.07	91.49	80.58	65.67	66.39	49.39	39.62	45.80	33.84	26.38
	5	10	71.94	59.13	48.69	91.20	80.42	65.53	64.41	48.45	37.56	43.47	32.58	24.96
	5	20	71.00	59.14	48.17	90.77	79.60	65.02	65.92	49.86	39.25	43.69	32.84	24.61
	10	10	71.82	60.17	49.01	91.34	81.19	66.30	65.54	50.23	40.28	45.40	35.63	26.34
	10	20	71.69	59.70	48.79	91.13	80.44	65.72	64.88	50.33	39.62	45.00	33.05	24.70
	50	10	69.90	59.80	47.59	91.07	80.88	65.23	64.23	49.95	37.93	45.06	34.63	26.41
	50	20	71.09	59.75	48.38	91.20	80.93	65.69	65.92	49.30	39.34	43.81	34.21	25.83
	50	50	71.45	59.46	48.55	91.01	80.63	65.88	64.79	51.46	39.72	43.52	33.26	24.73
Ensemble			**74.81**	65.76	54.42	**92.89**	**85.90**	72.53	**69.58**	55.02	43.57	**46.36**	40.47	**30.56**
Ensemble + LM			74.80	**66.16**	**54.61**	92.85	85.88	**72.55**	69.39	**55.31**	**43.66**	45.51	**40.84**	30.47
Challenge submission			70.65	63.53	50.94	91.07	85.16	69.49	64.50	58.20	42.55	37.83	37.33	25.31

Table 1. Result table. All of experiments use the same audio encoder and model architecture. 'Ensemble' and 'Ensemble+LM' are results before and after applying the language model, respectively. Please note that the performance are measured on Epic-kitchens validation set, except the 'challenge submission', which reports the challenge test set results.

λ	Top-1 Accuracy (%)		
	Verb	Noun	Action
0	71.25	62.42	50.14
0.1	**71.89**	63.33	50.95
0.3	71.77	**63.55**	**51.38**
0.5	71.28	63.01	50.73

Table 2. Ablation studies on λ

tion 3. The experiments are performed using $N = 10$ clips within $t = 10$ sec and pretrained MotionformerHR is used as a visual encoder. We prove that introducing the temporal localization loss ($\lambda > 0$) improves the recognition performance. We also observe that $\lambda = 0.3$ produces the best result. Therefore we use this value when training our model for our final submission.

5. Final submission

For our final submission, we incorporate Epic-kitchens validation set in our training data. We randomly choose 1000 samples from the validation set to track the performance and use rest of the validation set during training stage . We ensemble 16 different models, 8 using MotionformerHR encoder and another 8 using Omnivore encoder and average the softmax outputs.(See Table 1) We apply the language model on top of this result for our final submission. Our last submission is ranked 2^{nd} on the leaderboard. We only use publicly available datasets for training our model.

References

[1] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. VGGSound: A large-scale audio-visual dataset. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020. 3

[2] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 2021. 1

[3] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A Single Model for Many Visual Modalities. In *CVPR*, 2022. 3, 4

[4] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017. 1

[5] Evangelos Kazakos, Jaesung Huh, Arsha Nagrani, Andrew Zisserman, and Dima Damen. With a little help from my

temporal context: Multimodal egocentric action recognition. In *British Machine Vision Conference (BMVC)*, 2021. 1, 3

[6] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Slow-fast auditory streams for audio recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021. 3

[7] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco Nadai. Efficient training of visual transformers with small datasets. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 2

[8] Mandela Patrick, Dylan Campbell, Yuki M. Asano, Ishan Misra Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and Joo F. Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. In *Advances in Neural Information Processing Systems*, 2021. 3, 4

[9] 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. 1

[10] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes. *arXiv preprint arXiv:1904.00962*, 2019. 3

[11] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of International Conference on Learning Representations ICLR*, 2018. 3

Meituan Submission to the EPIC-Kitchens-100 Action Recognition Challenge 2022

Yiyang Gan, Weixin Luo, Bairui Wang, Lin Ma
Meituan, Beijing
{ganyiyang, luoweixin, wangbairui}@meituan.com forest.linma@gmail.com

Abstract

In this report, we briefly introduce the technical details of our submission to the EPIC-Kitchens-100 Action Recognition Challenge 2022. Considering the characteristics of different tasks, we design different input styles for different tasks. We then deploy feature extraction and classification based on several transformer models and perform a model ensemble on them. Finally, we propose a frequency-prior learning strategy to generate the prediction scores. Our submission achieves an action accuracy of 40.4% on the test set.

1. Introduction

In recent years, video analysis is one of the most popular areas of computer vision, which has a wide range of applications in automated surveillance, human-computer interaction, vehicle navigation, etc. With the rapid development of wearable cameras, such as GoPro, Google Glass, a large number of egocentric videos are being captured and stored. How to effectively analyze these videos becomes an important study topic.

EPIC-Kitchens-100 [2–4] is the largest dataset in egocentric vision, which contains a collection of 100 hours, 20M frames, 90K actions in 700 variable-length videos, capturing long-term unscripted activities in 45 environments, using head-mounted camera.

In this work, we separately train different models for verb classification, noun classification, and verb-noun classification. We then perform model ensemble to get the verb and noun prediction scores. Finally, we use the prior of the verb-noun pair to train a probability matrix and yield the action prediction scores.

2. Methods

2.1. Global-to-Local Image Collection

Video inputs always implicate lots of temporal information thus benefiting the verb classification task. However, the noun classification task doesn't heavily rely on temporal information. We design a global-to-local image collection as inputs for the noun classification task. As shown in Fig. 1, each mini-batch of the inputs are composed of 3 different images (c_1 full images, c_2 hand-object unions and c_3 object bounding boxes):

$$c_1 + c_2 + c_3 = B, c_1, c_2, c_3 = 1, 2, ..., B \tag{1}$$

B is the batch size of inputs when training. The hands and objects bounding boxes are extracted from [8]. For the verb classification task and verb-noun classification task, we simply use the extracted frames as inputs.

2.2. Video Classification

In order to get effective features and precise classification predictions from videos, we use 2 different networks, MViT [5] and Uniformer [7], for the 3 tasks: verb classification, noun classification and verb-noun classification.

To take advantage of complementary predictions from different models, we ensemble all the models and calculate the final verb/noun prediction scores by averaging the prediction scores from each model.

2.3. Frequency-Prior Learning

In Epic-Kithchens-100 dataset, there are 97 classes of verbs and 200 classes of nouns, thus there are theoretical 20580 classes of actions, which is much greater than the real class number 4053. In fact, most of actions will never occurs in the real scene, e.g. *cook clothes*, *wear potatoes*. Denote $\mathbf{S}_{\text{verb}} \in [0,1]^{v\times 1}$ as the scores of verbs predicted by the network, and $\mathbf{S}_{\text{noun}} \in [0,1]^{n\times 1}$ as the scores of nouns predicted by the network, where v, n is the number of verb class and noun class. Given the score $\mathbf{S}_{\text{verb}}(i)$ of the i-th classes of verb prediction, and the score $\mathbf{S}_{\text{noun}}(j)$ of the j-th classes of noun prediction. We compute the score of each class of action by:

$$\begin{gathered}\mathbf{S}_{\text{action}}(i,j) = \mathbf{M}(i,j)\times\mathbf{S}_{\text{verb}}(i) \times \mathbf{S}_{\text{noun}}(j), \\ i = 1, 2, ..., v, j = 1, 2, ..., n\end{gathered} \tag{2}$$

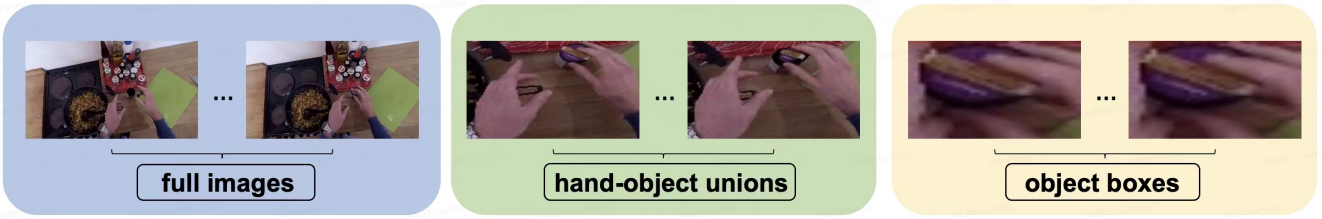


Figure 1. An illustration of the global-to-local image collection.

Table 1. Action recognition results on test set.

Model	Overall						Unseen Participants			Tail Classes		
	Top-1 Accuracy (%)			Top-5 Accuracy (%)			Top-1 Accuracy (%)			Top-5 Accuracy (%)		
	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action	Verb	Noun	Action
Ensemble	66.1	54.0	40.4	89.1	79.1	59.6	59.4	45.6	33.1	34.5	28.5	10.5

where $\mathbf{M} \in \mathbb{R}^{+^{v \times n}}$ is the probability matrix, which contains learnable parameters to calculate the probability of each verb-noun combination. We simply initial $\mathbf{M}$ with the count $\mathbf{C} \in \mathbb{N}^{v \times n}$ of verb-noun pairs in training annotations and validation annotations, and utilize cross-entropy loss to calculate the cost between the action prediction and the ground-truth. The proposed scheme can be summarized in Algorithm 1:

Algorithm 1: Frequency-Prior Learning Scheme

Input: $\mathbf{S}_{\text{verb}} \in [0,1]^{v \times 1}$, $\mathbf{S}_{\text{noun}} \in [0,1]^{n \times 1}$, $\mathbf{C} \in \mathbb{N}^{v \times n}$.

Initialization: $\mathbf{S}_{\text{action}}, \mathbf{M} \leftarrow \mathbf{S}_{\text{verb}} \mathbf{S}_{\text{noun}}^{\top}, \mathbf{C} * \mathbf{E}$, where $\mathbf{E} \sim \mathbf{U}(1 - 1/\sqrt{v+n}, 1 + 1/\sqrt{v+n})$ is a random matrix, and $\mathbf{U}$ represents the uniform distribution.

for *epoch in* $1 : max\_epoch$ **do**

  $\mathbf{M} = \mathbf{Relu}(\mathbf{M})$.

  $\mathbf{S}_{\text{action}} = \mathbf{M} * \mathbf{S}_{\text{action}}$.

  $\mathcal{L} = \mathbf{L}(\mathbf{S}^{\text{gt}}_{\text{action}}, \mathbf{S}_{\text{action}})$, where $\mathbf{L}$ is the cross-entropy loss function.

  Update $\mathbf{M}$ using AdamW optimizer.

Output: $\mathbf{S}_{\text{action}} \in \mathbb{R}^{v \times n}$

3. Experiments

Video Classification Details. We use MViT and Uniformer for verb classification and noun classification. The MViT model uses MViT-B architecture in [5], and the model is pretrained on Kinetics-600 [1]. The Uniformer model uses UniFormer-B architecture in [7], and the model is pretrained on Kinetics-400 [6]. We use only MViT for verb-noun classification, and the model is pretrained on the verb classification task.

At the training stage, we firstly use random short side scale jittering, random crop, random horizontal flipping, and mixup [9] with $\alpha = 0.8$ as data augmentation. The resolution of inputs is 320, and the temporal sampling is 32×2. We train both MViT and Uniformer using the AdamW optimizer with an initial learning rate of 1×10^{-4} and a weight decay of 5×10^{-2}. We also use a cosine learning rate schedule when training. In the verb classification task, we train the MViT model for 50 epochs with a batch size of 32, and train the Uniformer model for 40 epochs with a batch size of 48. In the noun classification task, we train the MVIT model for 35 epochs with a batch size of 32, and train the Uniformer model for 70 epochs with a batch size of 48. In the verb-noun classification task, we train the MViT model for 25 epochs with a batch size of 32.

At the testing stage, we sample 10 clips consisting 32 frames. For each frame, 3 spatial crops are generated. That is, we take 10×3 views from each video.

Frequency-Prior Learning. To simplify the training, we only train the network on the validation set. We train the layer using the AdamW optimizer with an initial learning rate of 1×10^{-2} and a weight decay of 5×10^{-4}. The training procedure last for 2 epochs with a batch size of 64.

Results We achieve an action recognition top1-accuracy of 40.4%, and perform well with unseen participants. Table 1 shows the reported results on all metrics.

4. Conclusion

This report presents our solution for the EPIC-Kitchens-100 action recognition challenge. To address the problem, we have designed a global-to-local image collection as inputs to promote the noun classification task. We simultaneously trained three networks for noun classification, verb classification and verb-noun classification tasks, and perform model ensemble on trained models. To make better use of the prior information, we have also proposed a frequency-prior learning strategy to calculate the probability matrix for action classification prediction. We finally achieve a good rank of action recognition accuracy on the leaderboard.

References

[1] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 2

[2] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *IJCV*, 2021. 1

[3] 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

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. The epic-kitchens dataset: Collection, challenges and baselines. *IEEE TPAMI*, 43(11):4125–4141, 2021. 1

[5] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *ICCV*. 1, 2

[6] 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. 2

[7] Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning. *arXiv preprint arXiv:1808.01340*, 2022. 1, 2

[8] Dandan Shan, Jiaqi Geng, Michelle Shu, and David F. Fouhey. Understanding human hands in contact at internet scale. In *CVPR*, 2020. 1

[9] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2

1st Place Solution to the EPIC-Kitchens Action Anticipation Challenge 2022

Zeyu Jiang[1] Changxing Ding[1,2*]
[1] South China University of Technology [2] Pazhou Lab, Guangzhou
jzy_scut@outlook.com, chxding@scut.edu.cn

Abstract

In this report, we describe the technical details of our submission to the EPIC-Kitchens Action Anticipation Challenge 2022. In this competition, we develop the following two approaches. 1) Anticipation Time Knowledge Distillation using the soft labels learned by the teacher model as knowledge to guide the student network to learn the information of anticipation time; 2) Verb-Noun Relation Module for building the relationship between verbs and nouns. Our method achieves state of the art results on the test set of EPIC-Kitchens Action Anticipation Challenge 2022.

1. Introduction

EPIC-KITCHENS is a large annotated egocentric dataset [1, 2]. Action anticipation is an important task in EPIC-KITCHENS.

We summarize our main contributions as follows:

1) Aiming at the problem that the missing information of anticipation time affects the performance of egocentric action anticipation, we propose Anticipation Time Knowledge Distillation to distill the information of anticipation time.

2) Because of the lack of consideration of the relationship between verbs and nouns in the existing research work on Egocentric Action Anticipation, we propose a verb-noun relationship interaction module to model the relationship between verbs and nouns.

3) Our approaches show superior results on EPIC-KITCHENS-100.

2. Our approach

2.1. Base Model

We use Causal Transformer Decoder (like AVT-h)[5] as base model. We use a 4-head, 4-layer model as our baseline.

2.2. Anticipation Time Knowledge Distillation

The temporal gap between the past observations and the future action (Anticipation Time)[4, 12] will result in missing information. To solve the problem that the missing information of anticipation time, we propose a knowledge distillation method to distill the information of anticipation time. Fig. 1 shows the student model. We initialize the future video embedding with learnable parameter. Fig. 2 shows the overview of Anticipation Time Knowledge Distillation. The input of teacher model is full video and the input of student model is the concatenation of the observed video and future video embedding. In teacher model, if the anticipation time clip (frame) don't have label, we use the label of the closest labeled clip (frame) as its label. The teacher model can distill the soft label of anticipation time to student model.

Finally, we use a multi-scale block to improve the perfor-

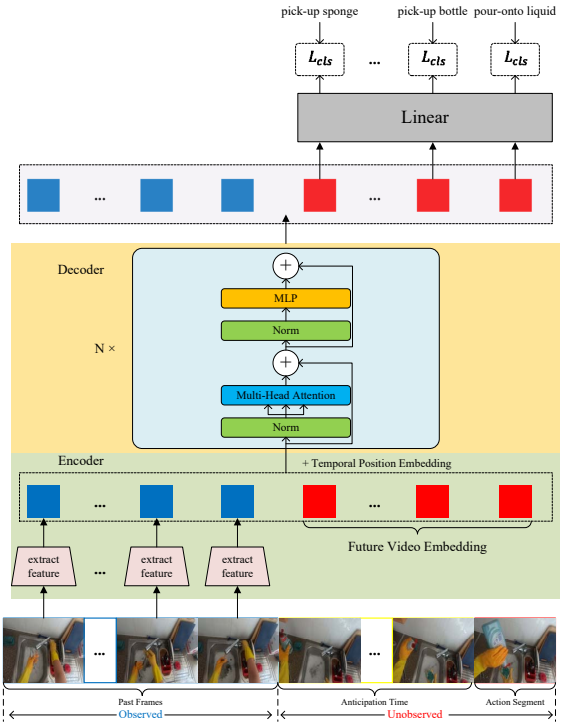


Figure 1. Student Model.

*Corresponding author.

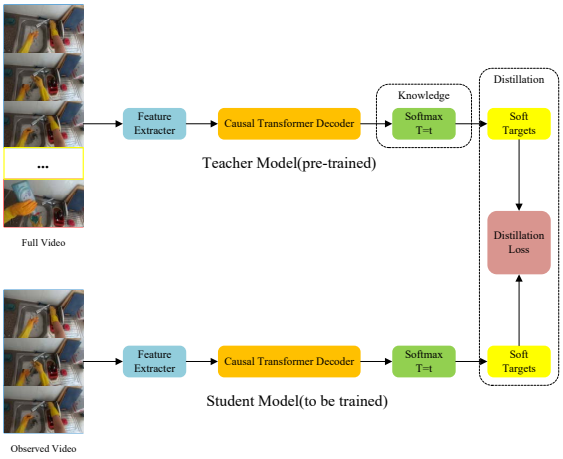


Figure 2. Overview of Anticipation Time Knowledge Distillation.

mance. Fig. 3 shows the architecture of student model with multi-scale block. Fig. 4 shows the details of multi-scale block.

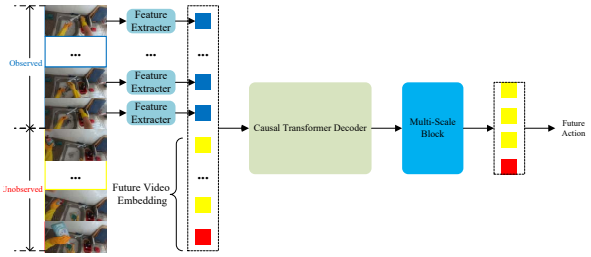


Figure 3. Student model with multi-scale block .

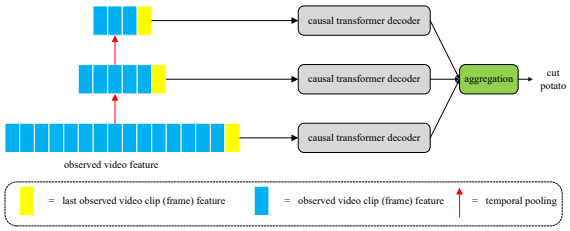


Figure 4. Multi-scale Block .

2.3. Verb-Noun Relation Module

Inspired by [10] and [11], we propose a verb-noun relationship interaction module to model the relationship between verbs and nouns. The verb-noun relation interaction module guides the features of the nouns interacting with the wearer in the observed videos to represent the features of the nouns interacting with the wearer in the future through

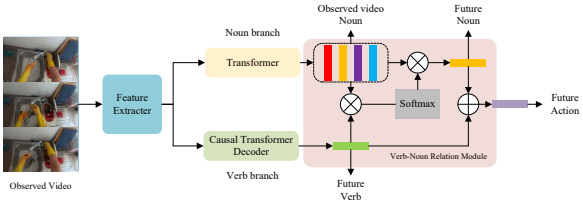


Figure 5. Overview of Verb-Noun Relation Module.

the features of the predicted future verbs. Fig. 5 shows the overview of Verb-Noun relation module.

Same as Anticipation Time Knowledge Distillation, if the clip (frame) don't have label, we use the label of the closest labeled clip (frame) as its label.

Finally, we use knowledge distillation to improve the performance. The input of teacher model's verb branch is the full video and the input of teacher model's noun branch is only the observed video.

2.4. Feature Extraction

We use some action recognition models as backbone to extract features.

The backbones as follow:

Model A SlowFast 16×8, R101+NL[3], predicting verb and noun
Model B SlowFast 8×8, R101[3], predicting verb and noun
Model C TSN(BNInception)[9, 4]
Model D Mformer-L[7], with temporal stride 4
Model E Mformer-HR[7], with temporal stride 8
Model F Mformer-HR[7], with temporal stride 4
Model G SlowFast 16×8, R101+NL[3], predicting verb, noun and action
Model H SlowFast 8×8, R101[3], predicting verb, noun and action

2.5. Ensemble

We use an ensemble of a set of 10 models as final result for testing set.

3. Experiments

3.1. Implementation Details

We train the networks using AdamW[6], using a batch size of 128, label smoothing[8] of 0.4, an l2 weight decay of $5e-4$, and an initial learning rate of $1e-4$. The maximum number of training iterations is set to 300 epochs. A cosine annealing with warm up restart schedule (20 cycles) is used. The cycles is set to 15 epochs with 1 epochs of linear warmup. All 10 models which we use an ensemble for testing set are trained on the same hyper parameters with same random seed.

Table 1. Results of Ablation Studies(Anticipation Time Knowledge Distillation).

method	kd	multi-scale	backbone	backbone(teacher)	verb	noun	action
base model			F	\	32	32.3	15.9
ATKD			F	\	31.2	34.6	16.7
ATKD	✓		F	F	32.2	35.3	17.3
ATKD	✓	✓	F	F	31.7	36.4	18.1
ATKD	✓	✓	F	B	33.7	36.3	19.1
ATKD	✓	✓	F	B+F(average soft label)	36.5	36.8	18.7

Table 2. Results of Ablation Studies(Verb-Noun Relation Module).

method	backbone	backbone(teacher)	verb	noun	action
base model	F	\	32	32.3	15.9
VNRM(w/o kd)	F	\	33.9	34.7	16.8
VNRM	F	F	31.7	37	17.5
VNRM	F	B	34.7	38.4	18.7
VNRM	F	B+F(avreage soft label)	32.9	39.7	19.2

Table 3. 10 model for ensemble.

#	method	backbone	backbone(teacher)	verb	noun	action	
1	base model	C	\	27.1	27.4	12.9	
2	ATKD	F	B	33.7	36.3	19.1	
3	ATKD	E	E	31.5	35.8	17.7	
4	ATKD	A	A	32.6	34.6	17	
5	ATKD	B	B	32.6	35.4	16.9	
6	ATKD	F	\	31.2	34.6	16.7	only student model w/o multi-scale
7	VNRM	G	G	29.6	36	16.3	
8	VNRM	H	H	33.1	34.4	15.9	
9	VNRM	D	B+F(average soft label)	31.7	38.2	17.1	
10	VNRM	F	B+F(average soft label)	32.9	39.7	19.2	
ensemble	ensemble	\	\	41	44.2	22.7	

Table 4. Action Anticipation results on test set.

	Overall			Unseen			Tail		
	Mean Top-5 Recall			Mean Top-5 Recall			Mean Top-5 Recall		
	Verb	Noun	Act.	Verb	Noun	Act.	Verb	Noun	Act.
Ensemble	37.91	41.71	20.43	27.94	37.07	18.27	32.43	36.09	17.11

3.2. Results

The result of ablation study can be found in Table 1 and Table 2. The result of 10 models for ensemble is shown in Table 3.

The final ensemble result on test set are presented in Table 4. Our algorithm achieved the best performance.

4. Conclusion

In this paper, we propose two novel methods. The validation and testing results show that our proposed method can achieve excellent performance.

References

[1] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018. 1

[2] 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: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, 130(1):33–55, 2022. 1

[3] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6202–6211, 2019. 2

[4] Antonino Furnari and Giovanni Maria Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6252–6261, 2019. 1, 2

[5] Rohit Girdhar and Kristen Grauman. Anticipative video transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13505–13515, 2021. 1

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[7] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[8] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016. 2

[9] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016. 2

[10] Xiaohan Wang, Yu Wu, Linchao Zhu, and Yi Yang. Symbiotic attention with privileged information for egocentric action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12249–12256, 2020. 2

[11] Xiaohan Wang, Linchao Zhu, Heng Wang, and Yi Yang. Interactive prototype learning for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8168–8177, 2021. 2

[12] Yu Wu, Linchao Zhu, Xiaohan Wang, Yi Yang, and Fei Wu. Learning to anticipate egocentric actions by imagination. *IEEE Transactions on Image Processing*, 30:1143–1152, 2020. 1

NVIDIA-UNIBZ Submission for EPIC-KITCHENS-100 Action Anticipation Challenge 2022

Tsung-Ming Tai[1,2], Oswald Lanz[2], Giuseppe Fiameni[1], Yi-Kwan Wong[1]
Sze-Sen Poon[1], Cheng-Kuang Lee[1], Ka-Chun Cheung[1], Simon See[1]

[1]NVIDIA, [2]Free University of Bozen-Bolzano

{tstai,oswald.lanz}@unibz.it
{gfiameni,gwong,spoon,ckl,chcheung,ssee}@nvidia.com

Abstract

In this report, we describe the technical details of our submission for the EPIC-Kitchen-100 action anticipation challenge. Our modelings, the higher-order recurrent space-time transformer and the message-passing neural network with edge learning, are both recurrent-based architectures which observe only 2.5 seconds inference context to form the action anticipation prediction. By averaging the prediction scores from a set of models compiled with our proposed training pipeline, we achieved strong performance on the test set, which is 19.61% overall mean top-5 recall, recorded as second place on the public leaderboard.

1. Introduction

Forecasting future events based on evidence of current conditions is an innate skill of human beings, and key for predicting the outcome of any decision making. Anticipating "*what will happen next?*" is a natural skill for human beings, but not for machines. In computer vision, the same question arises in video action anticipation. It is a long standing and widely studied problem to recognize the human actions given a video clip. However, to further predict the future action based on the given observations has just attracted increasing interests in recent years. Unlike action recognition, in action anticipation, the target action only stays in causal relation to the signal in the sub-clip, but is not directly observable. It must be forecast as one possible consequence of the already observed video context. EPIC-Kitchen-100 [2] is the largest dataset containing the definition of the video action anticipation task. It considers 97 verbs and 300 nouns. Unique verb-noun pairs define 3807 action categories. The dataset is provided with the pre-extracted RGB, optical flow, object bounding box, and object mask modalities in this competition.

We participated in the video action anticipation challenge by considering two different proposed models:

- *Higher-Order Recurrent Space-Time Transformer* [8]: A recurrent network with space-time decomposition attention and higher order recurrent designs.

- *Message-Passing Neural Network with Edge Learning* [9]: A recurrent network based on the message-passing framework. It models the sequential structure as a graph with a set of vertices and edges and learns the edge connectivity by different strategies.

Both Higher-Order Recurrent Space-Time Transformer (HORST) and Message-Passing Neural Network with Edge Learning (MPNNEL) are recurrent architectures, and learn the spatial-temporal dependencies in different ways. HORST builds the n-gram temporal modeling by considering the higher-order recurrence with temporal attention and dynamically attends the relevant spatial information by spatial attention. On the other hand, MPNNEL projects the spatial contexts of frame input from each timestep onto the internal graph representation, and leverages the message-passing framework to capture the temporal propagation. MPNNEL also learns to augment the edge connectivity by using different end-to-end learning strategies. Both modelings are based on the extracted feature from 2D-CNN frame-based backbone. The final score for this competition was deployed by late-fusion of all the training variants from HORST and MPNNEL, and averaging the prediction scores of individual models across different modalities.

The remaining parts of this report are organized as follows. The description of applied models is presented in Section 2, proposed training techniques are in Section 3. The experimental results are included in Section 4. Finally, Section 5 contains concluding remarks of this technical report.

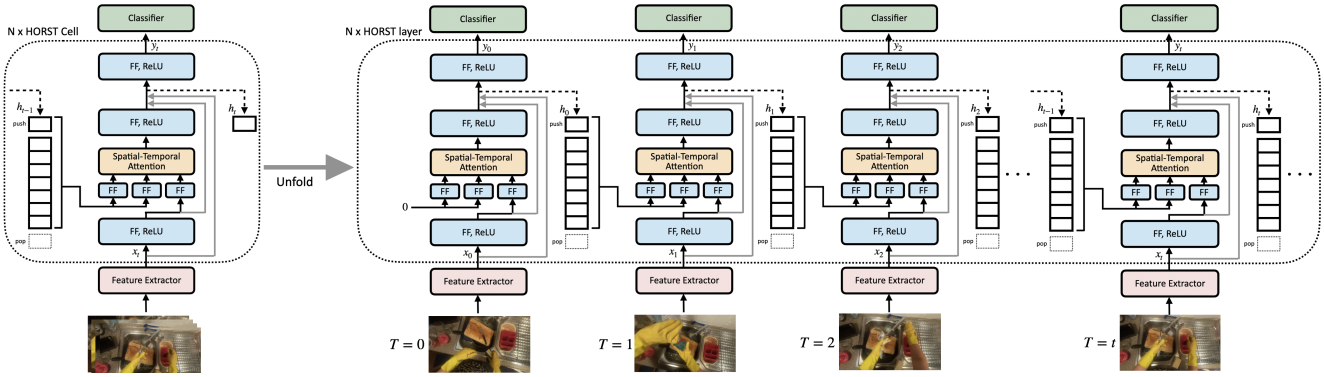


Figure 1. The overview HORST architecture. The HORST cell consist of a light-weighted spatial-temporal attention, and an internal first-in first-out queue to maintain the previous states for higher-order recurrence design.

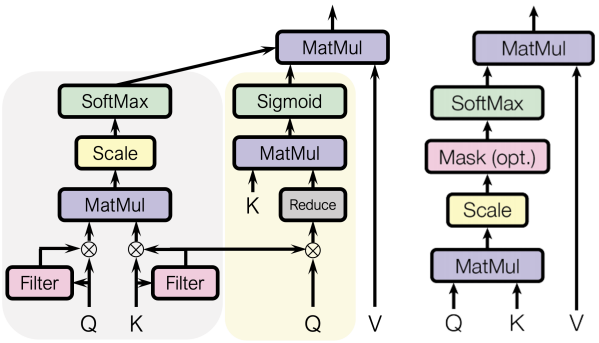


Figure 2. *Left:* The space-time decomposition attention used in HORST; *Right:* The self-attention proposed in [10].

2. Model Architecture

We briefly introduce HORST and MPNNEL architectures, the two modelings we used in this competition.

2.1. HORST Model

To exploit the effective information in space-time structure, we proposed *space-time decomposition attention* – a light-weighted and computation-efficient attention, which integrates spatial and temporal operators from separated branches as shown in Figure 2. To define spatial and temporal branch operators, *spatial filter* was introduced to recognize the relevant spatial information by the max and mean pooled features of inputs:

$$f_{\mathcal{X}}(X) = \text{sigmoid}(\theta_{\mathcal{X}} * [X_{max}, X_{avg}] + b_{\mathcal{X}}) \quad (1)$$

where $*$ is convolution, X_{avg}, X_{max} are channel mean and max pooled, $\theta_{\mathcal{X}}$ and $b_{\mathcal{X}}$ are convolution kernels and biases.

The general higher-order recurrent network [6, 7, 11] is with the following form:

$$h_t = f(x_t, \phi(h_{t-1:t-S})), \quad (2)$$

where the hidden state at time t, h_t, is computed by the cell function f on input x_t and S orders states $h_{t-1:t-S}$ aggregated by the function ϕ.

The HORST cell can be viewed as instantiating ϕ with *space-time decomposition attention* and maintain the previous states $h_{t-1:t-S}$ in an internal queue by the first-in first out update policy. The overall design is shown in Figure 1. At each step t, we process the video frame by a 2D-CNN backbone to obtain the feature map and encode it to the intermediate representation. Such representation is served as query and cross-reference from the historical states via the space-time decomposition attention. The attention output is then pushed to the queue while releasing the oldest state. Cell output finally propagate to the classifier. More details are found in [8] and we build our HORST models for this competition based on the codebase published at https://github.com/CorcovadoMing/HORST.

2.2. MPNNEL Model

MPNNEL translates the anticipation problem into a message passing scheme, producing a graph-structured space-time representation. The connectivity of the graph structure is inferred from the input at each time step. The readout function is called when the prediction is required at any timestep. The proposed model utilizes only multi-head self-attention for information routing between vertices. The overall architecture definition is illustrated in Figure 4. Note that the resulting spatial graph is either bi-directed, when an adjacency matrix A is provided, or else it is un-directed.

Without any prior knowledge, we assume each vertex in the graph can be accessible by any other vertices. In this case the scaled dot-product in the self-attention computes the pairwise similarity of all vertices from the inputs can be viewed as an *implicit* edge estimation. This can be extended by optionally providing the edge estimation *explicitly* by one of following strategies, also shown in Figure 3:

- *Template Bank (TB)*, which forms the estimation of

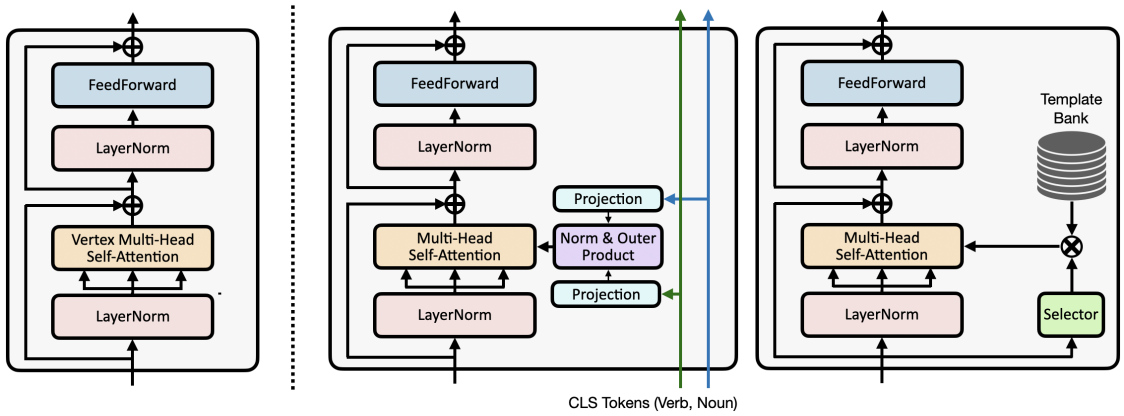


Figure 3. *Left:* The implicit edge estimation by multi-head self-attention; *Middle:* The augmented edge learning by outer product the class tokens supervised by the verb and noun annotations; *Right:* The augmented edge learning by introducing a joint learnable template bank.

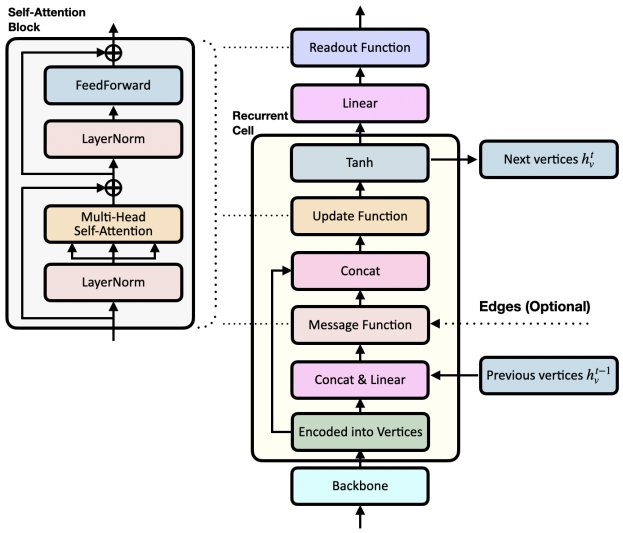


Figure 4. The overview architecture of MPNNEL. The message function is extendable with the explicit edge estimation by different edge learning strategies.

edge connections by soft-fusing a set of learnable templates using weights computed from the frame input.

- *Class Token Projection (CTP)*, which performs the outer-product of class tokens to construct the edge estimation. The class tokens are supervised from provided verb and noun labels.

More details are found in [9] and we build our MPNNEL models for this competition based on the codebase published at https://github.com/CorcovadoMing/MPNNEL.

3. Model Training

In this section, we describe the 4 phases training pipeline used to efficiently train all our models in this competition, and also the class weightings applied in the loss function to cope with imbalanced class distribution.

3.1. Training Phases

We trained all of our models by having them experience four learning phases, where they are (i) warmup phase; (ii) ordinary training; (iii) finetune; and (iv) finetune with joint validation set. The demonstration is shown in Figure 5.

The details of different training phases are:

- *Warmup Phase*: The model is end-to-end trainable on the target dataset. The model can access the actual action frames beyond the anticipation limitation only in this training phase.
- *Ordinary Phase*: The model is trained with backbone freeze, and the action frames are not accessible in this and following training stages.
- *Finetune Phase*: The model is trained with backbone freeze, under the lower learning rate, and with the class weightings adjusted in the loss functions.
- *Finetune with joint validation set*: The model is trained with backbone freeze, under the lower learning rate, and with the class weightings adjusted. Additionally, the validation samples are joint together in the supervised learning.

The warmup phase is targeted to build a strong feature extractor for the competition, the backbone model is able to receive gradients and the action frames are allowed to be observable exclusively in this phase. The ordinary phase focus on the anticipation task and trains the HORST and

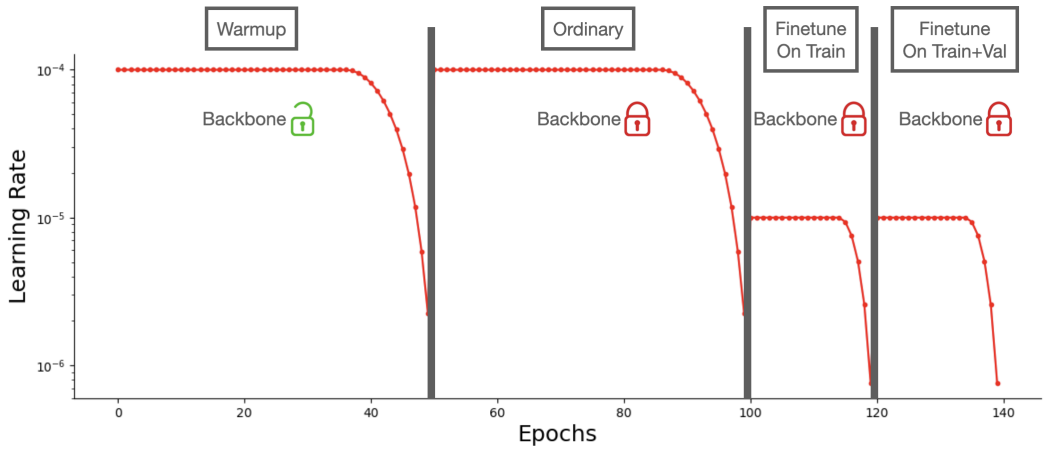


Figure 5. Demonstration of different training phase. The Backbone model is only trainable in the warmup phase and remains freeze in rest of the phases.

MPNN architectures with the feature extractor kept frozen. The finetune phases learn to distinguish the hard samples and tailed cases by class weighting adjustments, also with the validation set jointly in the last training. Every phases are resumed from its previous step and each model training experiences the complete learning rate scheduling.

3.2. Class Weightings

We adjusted the class weightings of the cross-entropy loss for individual verbs, nouns, and actions during the finetuning stages. The adjustment is based on the label frequency summarized from the training set. Note the action distribution defined in EPIC-Kitchen-100 is composed of joint probability of verbs and nouns, however, the label frequency of the action class could be different than the individual frequency belonging to verbs and nouns. Therefore, we empirically found this adjustment brings additional regularization to the model learning and results in noticeable gains on the validation up to 4% improvements.

4. Experiments

We provide implementation details and discuss the choices that led to our public record in the competition leaderboard.

4.1. Implementation Details

We prepared each input modality as follow: RGB frames are resized to 224x224 and the pixel values are scaled from [0, 256] to [-1, 1]. The Flow modality came with the two maps described for horizontal and vertical optical changes, we stacked the two maps in channel dimension and resized them to 224x224 with pixel intensity scaled to [-1, 1]. As inspired from [3], the Obj modality is formed by summarizing the object detection confidences from the officially provided object features, and discards the location information of the bounding boxes. Masked-RGB is the modality which multiplies the masking, extracted from a pretrained MaskedRCNN, with the RGB input.

The RandAugment [1] is applied for RGB, Masked-RGB, and Flow inputs. The video clip for training and inference are all sampled at 4 FPS (i.e., step 0.25s), as inherited from RU-LSTM baseline [3]. Each sample contains 14 sequential frames during training from 3.5s to 0.25s before action starts. However the last 3 frames are strictly not allowed to access in this competition since the anticipation time set to 1s. The total length of the inference context in our models are 2.5s (observed from 3.5s to 1s).

We trained our model using batch size 32 on 4 $\times$ NVIDIA A100 GPUs. AdaBelief [13] in combination with the look-ahead optimizer [12] is adopted. Weight decay is set to 0.001. The learning rate is set to 1e-4 and decreased to 1e-6 for warmup and ordinary training, and 1e-5 decreased to 1e-7 for finetune phases. The learning rate scheduling uses FlatCosine, which keeps the initial learning rate for the first 75% of total epochs and switches to cosine schedule for the last 25% epochs (see also Figure 5). The total epochs for warmup and ordinary training are set to 50, and 20 for finetune phases.

4.2. Individual Models

Unlike other modalities which are in an spatial-temporal structure, the Obj modality is presented as a temporal sequence of frame vectors. Each such vector represents the frame-level object scores computed from an object detection pretrained model. We modified HORST and MPNNEL models for supporting the 1D object vector representation, by replacing the 2D Convolution in HORST with the fully-connected layer; and by replacing the object entities with

Table 1. Individual model performance on validation set, measured in mean top-5 action recall (MT5R) at 1s, of various modalities using different modelings and backbones.

Model	Modality	Backbone	MT5R (%)
HORST	RGB	Swin-B	18.42
HORST	RGB	ConvNeXt	17.09
MPNNEL	RGB	Swin-B	17.05
MPNNEL (CTP)	RGB	Swin-B	18.18
MPNNEL (TB)	RGB	Swin-B	17.05
MPNNEL	RGB	ConvNeXt	17.18
MPNNEL (CTP)	RGB	ConvNeXt	18.54
MPNNEL (TB)	RGB	ConvNeXt	18.09
HORST	Flow	Swin-B	7.95
HORST	Flow	ConvNeXt	7.36
HORST	Flow (Snippets)	Swin-B	6.61
HORST	Flow (Snippets)	ConvNeXt	8.06
MPNNEL	Flow	Swin-B	-
MPNNEL (CTP)	Flow	Swin-B	6.66
MPNNEL (TB)	Flow	Swin-B	-
MPNNEL	Flow	ConvNeXt	7.59
MPNNEL (CTP)	Flow	ConvNeXt	8.74
MPNNEL (TB)	Flow	ConvNeXt	8.18
HORST	Obj	None	8.72
MPNNEL	Obj	None	9.69
MPNNEL (CTP)	Obj	None	8.80
MPNNEL (TB)	Obj	None	8.99
HORST	Masked-RGB	Swin-B	12.03
HORST	Masked-RGB	ConvNeXt	11.30
MPNNEL	Masked-RGB	Swin-B	9.22
MPNNEL (CTP)	Masked-RGB	Swin-B	7.87
MPNNEL (TB)	Masked-RGB	Swin-B	9.57
MPNNEL	Masked-RGB	ConvNeXt	9.65
MPNNEL (CTP)	Masked-RGB	ConvNeXt	8.53
MPNNEL (TB)	Masked-RGB	ConvNeXt	10.30

Table 2. Test accuracy of model ensemble.

Model	MT5R (%)
(a) HORST Family with all modalities	17.47
(b) MPNNEL Family with all modalities	18.19
(a) + (b)	19.52
(a) + (b) and weightings 1.2x on all RGB models	**19.61**

learnable vectors multiplied by corresponding object scores to defined the vertices in MPNNEL. Some models apply on Flow modality by snippets, as suggested in [3], where the previous 5 sequential Flow features are stacked.

For all of our models we considered the Swin Transformer (i.e., base configuration, Swin-B) [4], and ConvNeXt [5] as backbones. We showed validation results of each representative category in Table 1. Note the validation results reported in Table 1 are before training with the joint validation set, in order to keep the numbers meaningful.

4.3. Model Ensemble

We manually selected the strong models from each individual variant, and tried to balance between HORST and MPNNEL instances to maintain the diversity among the ensembled models. Our best submission, 19.61% overall accuracy, was achieved by an ensemble of in total 54 models. Those models consisted of 30 RGB models, 10 Flow models, 8 Obj models, and 6 Masked-RGB models.

Table 2 reports on the trajectory we stepped to our highest score submission. Averaging the prediction scores in the HORST family resulted in 17.47% overall test accuracy, and 18.19% in the MPNNEL family. Combining both HORST and MPNNEL further improved the score significantly, to 19.52%, indicating some degree of complementarity of the two recurrent models. We also empirically found that emphasizing the prediction scores of all RGB models can have additional performance gains. In our best submission we weighted all RGB models by a factor 1.2x higher than other modalities.

5. Conclusion

In this report, we presented the technical details of our submission, achieving an overall 19.61% mean top-5 recall on the EPIC-Kitchen-100 anticipation challenge 2022. Our method considered the Higher-Order Recurrent Space-Time Transformer (HORST) and Message-Passing Neural Network with Edge Learning (MPNNEL) architectures, which are both recurrent-based networks and only observed 2.5s inference context for the action anticipation. Combined with the proposed training pipeline and by averaging the prediction scores from the models trained from various modalities, our submission recorded the second place on the public leaderboard.

References

[1] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 4

[2] 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: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, 130(1):33–55, 2022. 1

[3] Antonino Furnari and Giovanni Maria Farinella. Rolling-unrolling lstms for action anticipation from first-person video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):4021–4036, 2020. 4, 5

[4] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In

Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 10012–10022, 2021. 5

[5] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *arXiv:2201.03545*, 2022. 5

[6] Rohollah Soltani and Hui Jiang. Higher order recurrent neural networks. *arXiv:1605.00064*, 2016. 2

[7] Jiahao Su, Wonmin Byeon, Jean Kossaifi, Furong Huang, Jan Kautz, and Anima Anandkumar. Convolutional tensor-train lstm for spatio-temporal learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 13714–13726, 2020. 2

[8] Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, and Oswald Lanz. Higher order recurrent space-time transformer for video action prediction. *arXiv:2104.08665*, 2021. 1, 2

[9] Tsung-Ming Tai, Giuseppe Fiameni, Cheng-Kuang Lee, Simon See, and Oswald Lanz. Unified recurrence modeling for video action anticipation. *arXiv:2206.01009*, 2022. 1, 3

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. 2

[11] Rose Yu, Stephan Zheng, Anima Anandkumar, and Yisong Yue. Long-term forecasting using tensor-train rnns. *arXiv:1711.00073*, 2017. 2

[12] Michael Zhang, James Lucas, Jimmy Ba, and Geoffrey E Hinton. Lookahead optimizer: k steps forward, 1 step back. In *NeurIPS*, volume 32, 2019. 4

[13] Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C. Tatikonda, Nicha C. Dvornek, Xenophon Papademetris, and James S. Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. In *NeurIPS*, 2020. 4

LTDS: ICL-SJTU Submission to EPIC-Kitchens Action Anticipation 2022

Xiao Gu[1], Yao Guo[2], Zeju Li[1], Jianing Qiu[1], Benny Lo[1], and Guang-Zhong Yang[2]
Imperial College London[1]
Shanghai Jiao Tong University[2]
xiao.gu17@imperial.ac.uk

Abstract

In this report, the technical details of ICL-SJTU submission to EPIC-Kitchens Action Anticipation Challenge CVPR 2022 are presented. We considered egocentric action anticipation as a long-tailed distribution problem entangled with domain shift problem. The coexistence of these two issues significantly degrades model performance in real-world deployment, which however was overlooked in most previous research. To participate in this challenge, we proposed a novel framework, denoted as ***LTDS****, to simultaneously handle long-tailed distribution and domain shifts. Our final submission to the test server achieves 42.0% for overall Verb, 35.7% for overall Noun, and 19.5% for overall Action, in terms of the class-mean recall@5.*

1. Introduction

Anticipating future actions from egocentric videos is an important task for human behaviour understanding. This is potentially beneficial to a variety of applications including assistive robotics, virtual reality, and autonomous driving. Recently, considerable research efforts have been devoted to egocentric action anticipation, ranging from large-scale dataset curation [6] to dedicated computational model design [5, 8]. However, this real-world task is associated with two inherent challenges, which have always been overlooked in previous works.

On one hand, human action categories are in nature long-tailed distributed, where a few classes account for the majority of sample classes, yet many more classes only present a few samples. For instance, as shown in Fig. 1, one of the most frequent daily actions in the kitchen is *"turn on tap"*, whereas other actions like *"move broccoli"* may occur much less frequently. In fact, an ideal action anticipation model is expected to perform well on all action classes, rather than get biased towards the majority classes.

On the other hand, the intentions of human actions are highly heterogeneous, prone to large variations caused by several factors. For different subjects, two totally different past observations may lead to the same future action category. For example, as shown in Figure 1, for the "*turn-on tap*" action, the past observation of two different subjects are distinct. Apart from subject behaviours, other factors including scenarios, camera types, viewpoints may also contribute to the data heterogeneity, making the domain gap a challenging problem [2].

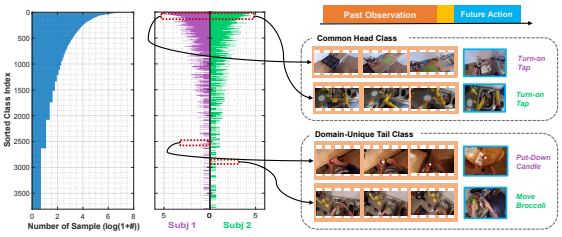


Figure 1. Illustrations of challenges associated with egocentric action anticipation. (i) First of all, the long-tailed distribution of the overall dataset poses challenge to achieving consistently good performance across all the classes. (ii) Secondly, the domain shift caused by human behaviour, camera viewpoint, scene settings, etc. leads to significant intra-class inter-domain variations. (iii) Furthermore, the coexistence issue of domain shift and long-tailed distribution limits the occurrence of most tail classes only in specific domains, where short-cuts associated with spurious correlation might be learned.

Although either long-tailed distribution [13] or domain shift [11] has been investigated a lot in the existing literature, they have so far not been addressed simultaneously. Existing long-tailed solutions usually only consider data from a homogeneous source, without taking domain shifts into considerations [13]. Moreover, existing solutions on domain generalization [11] mostly assume an identical distribution across domains. These can hardly be directly adopted in a real-world scenario where categorical distribution is imbalanced. In fact, such combination reveals a more challenging yet practical problem in the real world. Typically, the long-tailed distribution leads to a large number of tail classes unique only to a few domains, where the learned classifier may be optimized to certain short-cut decision boundaries if not carefully addressed.

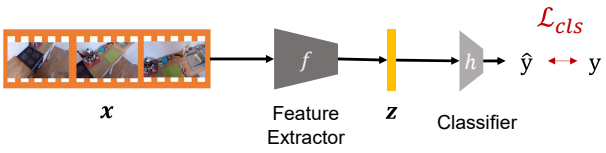


Figure 2. Basic model for egocentric action anticipation.

In this report, we presented our solution, referred to as **LTDS**, to effectively resolve the **long-tailed distribution (LT)** along with **domain shifts (DS)** inherent in egocentric action anticipation. We validated our proposed solution on EPIC-Kitchens 100 Action Anticipation track.

2. Methods

We denote the input past observations as $\boldsymbol{x} \in \mathbb{R}^{T \times D_f}$ and the ground truth future action as y, where T refers to the input frame number, D_f the feature dimension. Additionally, we considered data from each subject at the same time as a single domain, and the domain number is denoted as $d \in \{1, 2, ..., \#domain\}$. Given the inference model as g, the final prediction $\hat{y}$ can be derived by $g(\boldsymbol{x})$. Similar to previous domain generalization works that aim to learn domain invariant features, we further decompose g into a feature extractor f and a head classifier h, as shown in Fig. 2. To deal with the combined issue of long-tailed distribution and domain shifts, our goal is to make sure that f is able to learn domain invariant and unbiased representations.

2.1. Cross-Modality Matching to Enforce Unbiased Representation Learning

Due to the long-tailed category distribution and domain shifts, the domain-unique classes (mostly tail classes) may lead to spurious correlation between domain-specific and class-specific features. This would unfortunately lead to deriving biased representations. To tackle this, we leveraged external knowledge, word semantic embeddings $\boldsymbol{s} \in \mathbb{R}^{C \times D_s}$, to facilitate unbiased representation learning.

Instead of directly utilizing $\boldsymbol{z}$, as shown in Figure 3, we apply another multi-layer perceptron (MLP) to non-linear project $\boldsymbol{z}$ to $\boldsymbol{e}$ [1], and then utilize contrastive loss with large margin penalty to enforce the alignment between visual feature and semantic embeddings by Eq (1) as below,

$$\mathcal{L}_{cm} = -\log \frac{e^{(\boldsymbol{e}_i^\intercal \boldsymbol{s}_{y_i} - \alpha)/\tau}}{e^{(\boldsymbol{e}_i^\intercal \boldsymbol{s}_{y_i} - \alpha)/\tau} + \sum_{j \neq y_i} e^{\boldsymbol{e}_i^\intercal \boldsymbol{s}_j/\tau}}, \tag{1}$$

where α is the margin penalty and τ is a scale constant.

2.2. Prototype Learning to Tackle Domain Shifts

To further reduce the domain shifts for domain-invariant categorical features, we build domain-specific categorical prototype per domain, denoted as $\boldsymbol{\mu}^m$ for domain m. To deal with the missing classes in each domain, those corresponding entries are filled with their corresponding semantic embeddings. For those seen classes, we update their value in a online manner as shown in Eq (2),

$$\boldsymbol{\mu}_c^m|_{t+1} = \alpha \frac{1}{|\Lambda_c^m|} \sum_{\substack{y_i=c, \\ d_i=m}} \boldsymbol{e}_i^m + (1-\alpha)\boldsymbol{\mu}_c^m|_t, \tag{2}$$

where $|\Lambda_c^m|$ refers to the number of samples belonging to class c domain m in each batch.

During training, we align the feature of each sample to the prototype of other domains ($m \neq d_i$) by contrastive loss with large-margin penalty as in Eq (3).

$$\mathcal{L}_{proto} = \mathop{\mathbb{E}}_{m \neq d_i} -\log \frac{e^{(\boldsymbol{e}_i^\intercal \boldsymbol{\mu}_{y_i}^m - \alpha)/\tau}}{e^{(\boldsymbol{e}_i^\intercal \boldsymbol{\mu}_{y_i}^m - \alpha)/\tau} + \sum_{j \neq y_i} e^{\boldsymbol{e}_i^\intercal \boldsymbol{\mu}_j^m/\tau}}. \tag{3}$$

It should be noted that to aovid memory issues during the computation, we randomly selected the available domains in each training batch to perform the cross-domain alignment.

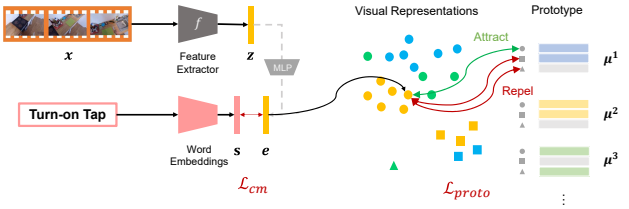


Figure 3. Illustration of cross-modality matching and cross-domain prototype matching. The feature $\boldsymbol{z}$ is projected to e by a MLP layer, and subsequently $\mathcal{L}_{CM}$ is leveraged to perform cross-modality matching, thus ensuring learning unbiased representations. Furthermore, to tackle domain shifts, we build prototype per domain in an online manner, and fill those missing entries with their corresponding semantic features. Then, $\mathcal{L}_{proto}$ is applied to align each visual feature to the visual prototype of other domains.

2.3. Two-Stage Training to Promote Tail Performance

To further improve the performance on tail classes, we incorporate two-stage training [9] into the whole training procedure. With the same cross entropy loss as in Eq (4), in the first stage, we follow the original long-tailed distribution. Subsequently, in the second stage, we apply class reweighting to derive a balanced distribution, and then perform optimization only on the head classifier h with the same $\mathcal{L}_{cls}$ loss.

$$\mathcal{L}_{cls} = CE(\hat{y}, y). \tag{4}$$

3. Experiments and Results

The whole framework was deployed with Pytorch on RTX 3090. The optimization was based on SGD (lr 0.01,

Table 1. Our results on validation set using all modalities (RGB+Flow+Obj). The metrics used are the % class-mean recall@5 for overall, unseen and tail splits, in terms of Verb, Noun and Action.

Input	Timestep	Overall			Unseen Kitchen			Tail Classes		
		Verb	Noun	Act	Verb	Noun	Act	Verb	Noun	Act
21.0s	1.0s	38.3	34.8	17.0	33.6	24.0	13.2	38.7	32.9	16.4
11.0s	0.5s	34.5	35.5	17.5	23.1	25.0	14.0	33.1	33.3	16.7
5.5s	0.25s	35.4	34.6	17.2	22.5	24.1	14.3	34.2	31.2	16.4
3.5s	0.2s	36.6	34.5	17.0	29.8	21.0	14.4	35.7	32.0	16.3
LTDS		41.1	37.0	19.5	34.0	27.8	15.4	40.1	34.2	18.8

momentum 0.9), with learning rate decayed by 0.5 per 10 epochs. We set the milestone for second stage training as epoch 25. The TransAction architecture proposed in our previous work [8] was utilized as the backbone, with the same input feature from [4]. In practice, we applied GloVe as the semantic embedding feature and set the margin α value as 0.1. Below, we present our detailed results on the validation set, as well as our submission to the test server.

3.1. Results on Validation Set

We train our model on four settings with varied input time and time step, as listed in Table1. The final version ensembled from these four models is denoted as **LTDS**. The results of each individual model and the final model are presented in Table 1.

In addition, we compared our results on the validation set with state-of-the-art methods, including RULSTM [4], AVT [5], Panasonic [3], and DCR [12], as presented in the Val split of Table 2. The results demonstrate the overall superior performance of our method compared to others.

3.2. Results on the Test Set

3.2.1 Trained on training set only

Meanwhile, we also compared the performance of these methods on the testing set when trained only on the training set, as shown in upper part of the test split in Table 2. As shown in Table 2, our method outperforms other state-of-the-art methods in most cases.

3.2.2 Final test server submission

To submit the final result to submission, we made an ensemble prediction as follows.

We trained our **LTDS** with train+val set, and also aggregate result from AVT++. This leads to our version **LTDS+ V1**. On top of **LTDS+ V1**, we made additional ensemble to improve the performance of Action. We replaced the RGB feature with TSM provided by [12], and also incorporated the result from [12]. This leads to the version **LTDS+ V2**. We compared the performance in the lower part of the test split of Table 2. Our final submission to the test server is **LTDS+ V2**.

Table 2. Comparison of results on val and test sets using all modalities (RGB+Flow+Obj). The metrics used are the % class-mean recall@5 for overall, unseen and tail splits, in terms of Verb, Noun and Action.

Split	Method	Overall			Unseen Kitchen			Tail Classes		
		Verb	Noun	Act	Verb	Noun	Act	Verb	Noun	Act
Val	RULSTM [2]	27.8	30.8	14.0	28.8	27.2	14.2	19.8	22.0	11.1
	AVT+ [5]	28.2	32.0	15.9	29.5	23.9	11.9	21.1	25.8	14.1
	Panasonic [3]	32.5	36.4	18.3	32.9	26.9	**15.4**	26.5	31.4	17.1
	DCR [12]	-	-	18.3	-	-	14.7	-	-	15.8
	LTDS	**41.1**	**37.0**	**19.5**	**34.0**	**27.8**	**15.4**	**40.1**	**34.2**	**18.8**
Test	RULSTM [2]	25.3	26.7	11.2	19.4	26.9	9.7	17.6	16.0	7.9
	AVT+ [5]	25.6	28.8	12.6	20.9	22.3	8.8	19.0	22.0	10.1
	TempAgg [10]	21.8	30.6	12.6	17.9	27.0	10.5	13.6	20.6	8.9
	TransAction [8]	36.2	32.2	13.4	27.6	24.2	10.1	32.1	29.9	11.9
	Panasonic [3]	30.4	33.5	14.8	21.1	**27.1**	10.2	24.6	27.5	12.7
	LTDS	**42.3**	**34.6**	**17.0**	**33.4**	25.9	**12.8**	**42.5**	**31.4**	**15.6**
	AVT++ [5]	26.7	32.3	16.7	21.0	**27.6**	12.9	19.3	24.0	13.8
	DCR* [12]	29.9	30.4	17.4	25.1	26.1	14.2	24.6	23.7	14.3
	LTDS+ V1	**42.0**	**35.7**	18.9	**33.4**	26.8	15.2	**41.0**	**33.2**	16.4
	LTDS+ V2	-	-	**19.5**	-	-	**15.9**	-	-	**16.9**

*Ensemble with AVT++

Table 3. Results of model/modality-agnostic effectiveness on validation set using different modals and single-modalities. The metrics used are the % class-mean recall@5 for overall, unseen and tail splits, in terms of Verb, Noun and Action.

Modal	Method	Overall		
		Verb	Noun	Act
RGB	TempAgg [10]	24.2	29.8	13.0
	↪+LTDS	29.1	32.0	13.8
	TransAction [8]	28.3	30.8	13.8
	↪+LTDS	34.2	32.7	16.5
Flow	TempAgg [10]	18.9	18.7	7.3
	↪+LTDS	22.7	19.3	8.1
	TransAction [8]	24.1	19.5	8.5
	↪+LTDS	26.6	19.6	8.8
Obj	TempAgg [10]	20.5	27.6	10.5
	↪+LTDS	28.9	32.2	12.0
	TransAction [8]	20.8	27.3	9.5
	↪+LTDS	28.6	29.4	10.2

3.3. Model/Modality-Agnostic Effectiveness

To validate the effectiveness of our framework on different modalities and models, we applied a single-modality version of TransAction [7], and the TempAgg model [10] on three different modalities provided by [4].

As the results shown in Table 3, performance gains can be achieved after integrating our framework for training, across both models and all three modalities; this highlights the compatibility of our proposed framework.

4. Discussion

In this report, we presented our solution to Epic-Kitchen Action Anticipation Challenge 2022. To handle the co-existence of long-tailed distribution and domain shifts, we proposed cross-modality matching and cross-domain prototype alignment to ensure learning domain-invariant unbiased representations. In addition, we leveraged two-stage training to improve the performance of tail classes. Our final submission to the testing server achieves 42.0% for overall Verb, 35.7% for overall Noun, and 19.5% for overall Action.

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[2] 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, 3

[3] Dima Damen, Adriano Fragomeni, Jonathan Munro, Toby Perrett, Daniel Whettam, Michael Wray, Antonino Furnari, Giovanni Maria Farinella, and Davide Moltisanti. Epic-kitchens-100- 2021 challenges report. Technical report, University of Bristol, 2021. 3

[4] Antonino Furnari and Giovanni Maria Farinella. What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6252–6261, 2019. 3

[5] Rohit Girdhar and Kristen Grauman. Anticipative Video Transformer. In *ICCV*, 2021. 1, 3

[6] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. *arXiv preprint arXiv:2110.07058*, 2021. 1

[7] Xiao Gu, Yao Guo, Fani Deligianni, Benny Lo, and Guang-Zhong Yang. Cross-subject and cross-modal transfer for generalized abnormal gait pattern recognition. *IEEE Transactions on Neural Networks and Learning Systems*, 32(2):546–560, 2020. 3

[8] Xiao Gu, Jianing Qiu, Yao Guo, Benny Lo, and Guang-Zhong Yang. Transaction: Icl-sjtu submission to epic-kitchens action anticipation challenge 2021. *arXiv preprint arXiv:2107.13259*, 2021. 1, 3

[9] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019. 2

[10] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long-range video understanding. In *European Conference on Computer Vision*, pages 154–171. Springer, 2020. 3

[11] Jindong Wang, Cuiling Lan, Chang Liu, Yidong Ouyang, Tao Qin, Wang Lu, Yiqiang Chen, Wenjun Zeng, and Philip Yu. Generalizing to unseen domains: A survey on domain generalization. *IEEE Transactions on Knowledge and Data Engineering*, 2022. 1

[12] Xinyu Xu, Yong-Lu Li, and Cewu Lu. Learning to anticipate future with dynamic context removal. In *CVPR*, 2022. 3

[13] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *arXiv preprint arXiv:2110.04596*, 2021. 1

Technical Report for EPIC-100 Action Anticipation Challenge 2022

Yutaro Yamamuro[1] Shinji Takenaka[1] Yuji Sato[2] Takeshi Fujimatsu[2]

{yamamuro.yutaro, takenaka.shinji-k, sato.yuji, fujimatsu.takeshi}@jp.panasonic.com
[1]Panasonic System Networks R&D Lab. Co., Ltd.
[2]Panasonic Connect Co., Ltd. R&D Division Advanced Research Lab.

Abstract

This report explains the method that we submitted to the EPIC-KITCHENS-100 Action Anticipation Challenge 2022, in which we adopted Video Swin Transformer (VST) as the base model. While VST has been shown to produce outstanding results in action recognition, it also produced excellent results in action anticipation from first-person visuals. Further, to counter the fact that EpicKitchens100 is an imbalanced dataset, we used the label-distribution-aware margin (LDAM) loss method, and logit adjustment to maintain the margin between logits with rare versus dominant labels. We also used the RandAugment-T method, which considers the temporal perturbation of videos, for data augmentation. The model that we ultimately submitted recorded 18.68% on the public leaderboard.

1. Introduction

Anticipating future action using first-person visuals is an important task for computer vision. It is a technology that can have future value through use in systems that warn about dangerous actions, and systems that support users' next actions.

For conventional action anticipation models, Furnari et al. [1] propose RULSTM, an LSTM-based method that integrates RGB, optical flow, and video object information. Recently, for various tasks in the computer vision field, there have been more reports using the Transformer method than conventional methods. This is not uncommon in tasks for action anticipation, however. For example, Girdhar et al. [2], who won last year's Action Anticipation Challenge [3], proposed AVT [2] to model the continuity of visuals. This combined a Vision Transformer [4] that models spatial information with a Transformer to model chronological information. In doing so, the team achieved the highest performance in datasets such as Epic-Kitchens [5], EGTEA, and 50-Salads.

In our method, we used Vision Swin Transformer (VST) [6] as the action anticipation base model. VST is a Transformer for action recognition proposed by Liu et al., and this method has achieved the highest performance in datasets including Kinetics-400 [7], Kinetics-600 [8], and Something-Something v2. Unlike AVT [2], VST can simultaneously model spatial and temporal information through embedded patches using 3DCNN, Shifted Window Multi-head Self Attention, and Patch Merging. Using VST that we fine-tuned through pre-training on Kinetics and Epic-Kitchens [5], we achieved an anticipation score comparable to conventional methods.

Epic-Kitchens-100 is a large-scale dataset that includes video of work in the kitchen filmed on a head-mounted camera, with labels for 3,806 different kinds of actions. With such a large number of labels, there is an imbalance in the amount of data for each label. For example, for "turn-on tap", the label with the largest of amount of data, there are 1,900 pieces of data. On the other hand, for "take pear" and other labels, which have the lowest amounts, there is only one piece of data. When learning imbalanced data such as this, overfitting occurs for labels with large amounts of data, and performance drops for those with less data. We adopted the LDAM [9] and Logit Adjustment [10] methods to improve general efficiency for this imbalanced data. LDAM is a loss function method proposed by Cao et al. [9] that determines decision bounds so that labels with more data have smaller distance margins from other classes of data distributions and labels with fewer data have larger margins. LDAM can replace regular cross-entropy loss, and can be used alongside other countermeasures for imbalanced data, such as re-weighting and re-sampling. Logit Adjustment is a method proposed by Menon et al. [10], and by adjusting logits from the model, it can minimize balanced error, which treats errors from labels with large and small amounts of data equally, and in turn promote improvement in overall data performance. Adjustments can be made to logits after learning and during learning, and in our experiment, the latter excelled. Further, Logit Adjustment and LDAM can be used simultaneously, and in papers written by Menon et al. [10], they were shown to be effective in image recognition tasks for imbalanced datasets.

In the same way, our experiment showed that simultaneous use of LDAM [9] and Logit Adjustment [10] was also effective for action anticipation. To improve the anticipation score, we also used RandAugment-T [11] for data augmentation. RandAugment-T is a data augmentation method for

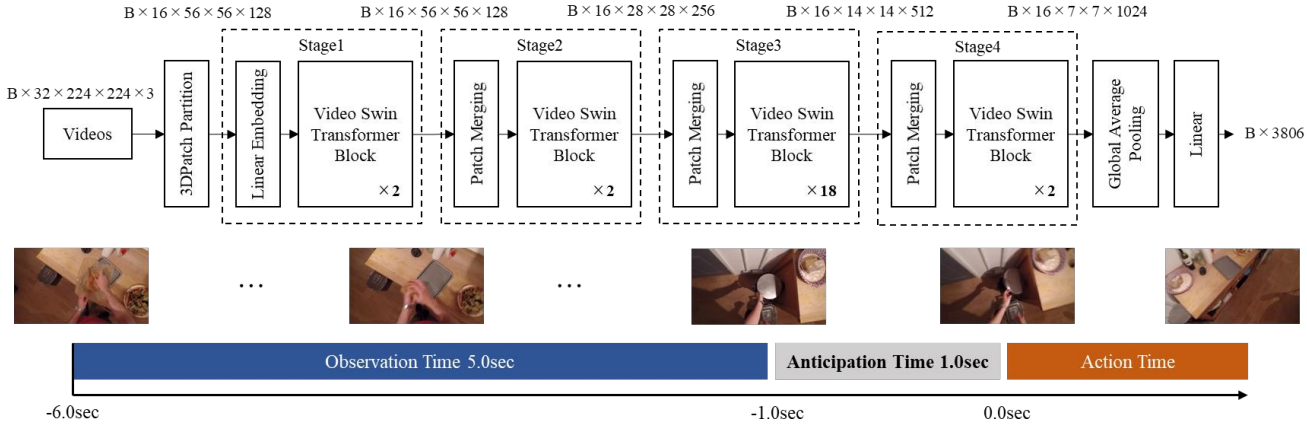


Fig. 2: Overview of Video Swin Transformer based on Liu 2021 [6] (top) and input data(bottom).

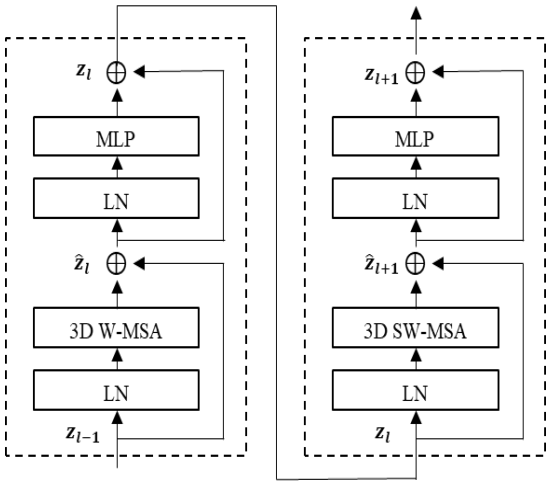


Fig. 1: An illustration of Video Swin Transformer Block based on Liu 2021 [6].

video recognition proposed by Kim et al. [11], extending the RandAugment, [12] a data augmentation method frequently used in image recognition, to the temporal dimension in videos. As time passes in video, the position of the object that is the subject of the video changes, while the brightness of the overall image changes, too. RandAugment-T expresses these temporal changes by changing the strength of the data augmentation in each frame.

Our report comprises the following.

- Section 2: Details of the method adopted for the competition
- Section 3: Experiments and results
- Section 4: Conclusion

2. Our Approach

2.1. Video Swin Transformer

For our base model, we used the VST-B [6] publicly available at the link[1] below. The composition of VST-B can be seen in Fig 1. It comprises four stages, with 2, 2, 18, and 2 as the number of blocks for each stage. The composition of the blocks can be seen in Fig 2. z shows the embedding features, LN shows LayerNormalization, MLP shows the fully connected layer, and 3D(S)W-MSA shows 3D shifted window multi head attention.

2.2. Long-tail Learning

2.2.1 LDAM

LDAM [9] is shown in Eq. 1, while the margin for each label is shown in Eq. 2. Through Eq. 1, we can see that the margin calculated in Eq. 2 is applied only to the labels subject to anticipation. In Eq. 2, C is the constant not dependent on the amount of data, and n_j shows the amount of data for the labels j subject to anticipation.

$$L_{LDAM}((x,y);f) = -\log\frac{e^{z_y-\Delta_y}}{e^{z_y-\Delta_y}+\Sigma_{j\neq y}e^{z_j}} \quad (1)$$

$$where\ \Delta_j = \frac{C}{n_j^{\frac{1}{4}}}\ for\ j \in \{1,\dots,k\} \quad (2)$$

2.2.2 Logit Adjustment

Of the two Logit Adjustment [10] methods, the Logit Adjusted Loss adjusted during learning is shown in Eq. 3. y shows the labels subject to anticipation, $f(x)$ shows the logits obtained from the model, and π_y shows the ratio of relevant y labels in the overall amount of data. τ is the hyperparameter that shows the strength of adjustment.

$$L_{LA} = -\log\frac{e^{f_y(x)+\tau\cdot\log\pi_y}}{\Sigma_{y'\in[L]}e^{f_{y'}(x)+\tau\cdot\log\pi_{y'}}}$$
$$= -\log\left[1+\Sigma_{y'\neq y}\left(\frac{\pi_{y'}}{\pi_y}\right)^{\tau}\cdot e^{\left(f_{y'}(x)-f_y(x)\right)}\right] \quad (3)$$

2.3. RandAugment-T

RandAugment-T [11] has two hyperparameters: n for the number of data augmentations and m for the strength. The strength (m) changes linearly from the head frame to the tail frame. Whether the strength gets stronger or

[1] https://github.com/SwinTransformer/Video-Swin-Transformer

Table1 : Results of changing pre-training. IN22, K400, and K600 indicate the type of prior learning, ImageNet22K, Kinetics-400, and Kinetics-600, respectively.

	Mean Top5 Recall[%]		
pre-train	Overall	Unseen	Tail
IN22	13.28	13.82	10.03
K400	14.54	14.89	11.12
K600	14.92	16.13	11.50

weaker from the head to the tail is determined randomly. The types of data augmentation are the same as the conventional RandAugment [12]: the video's geometrical changes are modeled by adjusting rotate, shear-x, shear-y, translate-x, and translate-y, while changes in the brightness and colors of the video are modeled using solarize, posterize, contrast, and brightness. Whereas usually it is important to look for the optimal data augmentation method through grid research, due to time constraints, in this experiment we only adjusted the hyperparameters.

2.4. Model Ensemble

For the model ensemble, weighted additions are made to the output from each model. The weight of each model is automatically tuned using Optuna [13]. The objective function of Optuna is set at the maximum anticipation score for the validation data, and the number of trials undertaken for tuning was 200.

3. Experiment

Epic-Kitchens-100 [5] video data was resized to 456 in height and 256 in width and converted to 30 fps. The data input into the VST [6] was post-conversion data. We sampled five-second-long clips of an equal thirty-two frames from six seconds before the action to one second before the action, and each was cropped to a height of 224 and a width of 224. Regarding the basic learning pipeline, AdamW was used for the Optimizer and the learning rate was set at 3e-4. The parameter betas for AdamW were set at 0.9 and 0.999, and the weight decay was set at 0.05. Cosine Annealing was used for the warmup, and the learning rate was raised linearly over approximately 42,000 iterations from the start of the learning process. The training epoch was set to 30, the minibatch size was 16, and learning was undertaken using four NVIDIA P100GPU units in a row. Random Resize Crop and Flip were used as data augmentation standards.

3.1. Compare pretrain

To clarify the effect of other items under examination, we checked the performance of the baseline VST [6]. Table 1 shows the results when pre-training for VST-B was changed. When comparing the results of pre-training, the model that used Kinetics-600 [8] for pre-training achieved the highest performance at 14.92%, showing a 1.64% improvement over ImageNet21K and 0.38% over Kinetics-400 [7].

Table2 : Results of Long-tail Learning. CE and LA denote Cross Entropy and Logit Adjustment, respectively. + indicates a combination of the two methods.

		Mean Top5 Recall[%]		
pre-train	Loss	Overall	Unseen	Tail
K400	CE	14.54	14.89	11.12
K400	LDAM	14.71	16.99	10.92
K400	CE+LA	18.59	14.80	16.96
K400	LDAM+LA	19.74	15.33	18.05
K600	LDAM+LA	19.91	15.74	17.93

3.2. Long-tail Learning

Table 2 shows the results from applying LDAM [9] and Logit Adjustment [10] both individually and simultaneously. The parameters τ for Logit Adjustment are uniform at 1.0. When comparing CE and LDAM, at 14.71%, LDAM exceeds CE by 0.17%. When using Logit Adjustment alongside CE and LDAM, the results were 18.59% and 19.74%, respectively.

3.3. RandAugment-T

Table 3 shows the results when applying RandAugment-T [11]. n=1 and m=10 are the optimal parameters, and there is a 0.74% improvement compared to before application. Further, when comparing RandAugment [12] using the same parameters, RandAugment-T scores 0.36% higher. It is also effective when used in conjunction with RandAugment-T and LDAM or Logit Adjustment. When using LDAM and Logit Adjustment, the results were 20.54%. This result was best in our experiments under the condition of no model ensemble.

3.4. Submission Model

3.4.1 Test Time Augmentation

At the time of inference, we applied Three Crop for spatial data augmentation. When preprocessing and cropping the frame to 224 px horizontally and vertically, we used Three Crop to crop the frame into three to cover the whole frame. Images cropped into three were input into the model before outputting the average logit for each.

3.4.2 Model Ensemble

Table 4 shows the subject of the model ensemble and the weight of each model tuned by Optuna [13]. The parameters for the models shown on the table were adjusted using validation data, and the final validation data was added to the training data before learning. The

Table3 : Results of RandAugment-T. n and m respectively indicate the number and intensity of data augmentation.

					Mean Top5 Recall[%]		
pre-train	Augmentation	n	m	Loss	Overall	Unseen	Tail
K400	RandAugment-T	1	7	CE	14.75	14.92	11.53
K400	RandAugment-T	1	10	CE	15.28	14.22	11.82
K400	RandAugment-T	1	13	CE	15.04	15.29	11.77
K400	RandAugment	1	10	CE	14.92	15.01	11.42
K400	RandAugment-T	1	10	CE+LA	19.41	13.06	18.68
K400	RandAugment-T	1	10	LDAM+LA	20.54	16.34	19.06

Table4 : Details of models registered on public leaderboards.

pre-train	Loss	RandAugment-T	Three Crop	weight	valid_score
K600	LDAM + LA			9.14	19.91
K600	LDAM + LA		✓	5.41	20.20
K600	CE + LA			2.60	19.74
K600	CE + LA		✓	7.41	19.68
K400	LDAM + LA	✓		6.05	20.54
K400	LDAM + LA	✓	✓	6.37	20.23

model that we finally submitted recorded 18.68% on the public leaderboard.

4. Conclusion

This report outlined the method that we submitted for the action anticipation category in EPIC-KITCHENS-100 2022 CHALLENGES. For our base model, we adopted Video Swin Transformer [6], which boasts state-of-the-art performance in action recognition. To counter the imbalanced dataset, we also adopted LDAM [9] and Logit Adjustment [10]. We used RandAugment-T [11] as a data augmentation method suited to video recognition, and confirmed the effectiveness of improvement methods for each. For the model that we ultimately submitted, we put together a weighted ensemble using anticipations from a trained model based on multiple conditions. For the weight of each model, we used Optuna [13] to make automatic adjustments to maximize performance metrics.

References

[1] Antonino Furnari and Giovanni Maria Farinella, "Rolling-Unrolling LSTMs for Action Anticipation from First-Person Video," arXiv preprint arXiv:2005.02190, 2020.

[2] Rohit Girdhar, Kristen Grauman, "Anticipative Video Transformer," arXiv preprint arXiv:2106.02036, 2021.

[3] Dima Damen, Adriano Fragomeni, Jonathan Munro, Toby Perrett, Daniel Whettam, Michael Wray, Antonino Furnari, Giovanni Maria Farinella, Davide Moltisanti, "EPIC-KITCHENS-100- 2021 Challenges Report," https://epic-kitchens.github.io/Reports/EPIC-KITCHENS-Challenges-2021-Report.pdf, 2021.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," arXiv preprint arXiv:2010.11929, 2020.

[5] Damen, Dima and Doughty, Hazel and Farinella, Giovanni Maria and Furnari, Antonino and Ma, Jian and Kazakos, Evangelos and Moltisanti, Davide and Munro, Jonathan and Perrett, Toby and Price, Will and Wray, Michael, "Rescaling Egocentric Vision," arXiv preprint arXiv:2006.31256, 2020.

[6] Liu, Ze and Ning, Jia and Cao, Yue and Wei, Yixuan and Zhang, Zheng and Lin, Stephen and Hu, Han, "Video Swin Transformer," arXiv preprint arXiv:2106.13230, 2021.

[7] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, Andrew Zisserman, "The Kinetics Human Action Video Dataset," arXiv preprint arXiv:1705.06950, 2017.

[8] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, Andrew Zisserman, "A Short Note about Kinetics-600," arXiv preprint arXiv:1808.01340, 2018.

[9] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, Tengyu Ma, “Learning Imbalanced Datasets with Label-Distribution-Aware Margin Loss,” arXiv preprint arXiv:1906.07413, 2019.

[10] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, Sanjiv Kumar, “Long-tail learning via logit adjustment,” arXiv preprint arXiv:2007.07314, 2020.

[11] Taeoh Kim, Hyeongmin Lee, MyeongAh Cho, Ho Seong Lee, Dong Heon Cho, Sangyoun Lee, “Learning Temporally Invariant and Localizable Features via Data Augmentation for Video Recognition,” arXiv preprint arXiv:2008.05721, 2020.

[12] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, Quoc V. Le, “RandAugment: Practical automated data augmentation with a reduced search space,” arXiv preprint arXiv:1909.13719, 2019.

[13] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, Masanori Koyama, “Optuna: A Next-generation Hyperparameter Optimization Framework,” arXiv preprint arXiv:1907.10902, 2019.

One-stage Action Detection Transformer @ EPIC-KITCHENS Action Detection Challenge

Lijun Li, Li'an Zhuo, Bang Zhang
Alibaba Group
{shenfei.llj, lianzhuo.zla, zhangbang.zb}@alibaba-inc.com

Abstract

In this work, we introduce our solution to the EPIC-KITCHENS-100 2022 Action Detection challenge. One-stage Action Detection Transformer (OADT) is proposed to model the temporal connection of video segments. With the help of OADT, both the category and time boundary can be recognized simultaneously. After ensembling multiple OADT models trained from different features, our model can reach 21.28% action mAP on the test-set of the Action detection challenge.

1. Introduction

With the explosion of video contents, video understanding has gained lots of interest from computer vision researchers [10,11,13,14,20]. In this field, action related tasks form the basis of video understanding. Compared with traditional action recognition [12, 18, 21], action detection not only recognizes action classes, but also detects the temporal boundaries simultaneously. Although only solve one another task, it is much difficult to distinguish the boundary since the action interval is ambiguous. In order to solve the action detection task, most traditional works firstly generate action proposals sorted by confidence score, then use another separate module to classify the proposals. With the great success of transformer in vision, a few works start to insert transformer into the action detection pipeline [22,23]. We follow similar pipeline and propose a one-stage network OADT for action detection.

2. Our Approach

The overall structure is showed in Fig. 1. The network is composed of three parts: video encoder, transformer neck and detection heads. In the following, we will describe each part in details.

2.1. Video Encoders.

Limited by the device memory, the raw video cannot be directly fed to the network. Therefore, the clip-level features are extracted from the untrimmed video using the video encoders. The video encoders are adapted from action recognition without the classification head. In this work, five superior action recognition methods are implemented.

Omnivore [8]. Omnivore is based on the swin-transformer, which leverages the flexibility of transformer-based architectures and is trained jointly on classification tasks from different modalities. For action recognition, the videos are converted into spatio-temporal tubes, and then these tubes are projected into embeddings using the linear layer.

MVit [6]. Multiscale Vision Transformers create a multi-scale pyramid of features on the vision transformer, which hierarchically expands the feature complexity while reducing visual resolution.

Motionformer [17]. Motionformer introduces the trajectory attention that aggregates information along implicitly determined motion paths on the video transformer.

Slowfast [7]. SlowFast proposes a two-pathway architecture for video recognition. A slow pathway with a low frame rate is designed to capture spatial semantics. In contrast, a fast pathway, operating at high temporal resolution, is responsible for dealing with rapid motion.

TimeSformer [2]. TimeSformer is a transformer-based approach built exclusively on self-attention over space and time, where temporal attention and spatial attention are separately applied within each block.

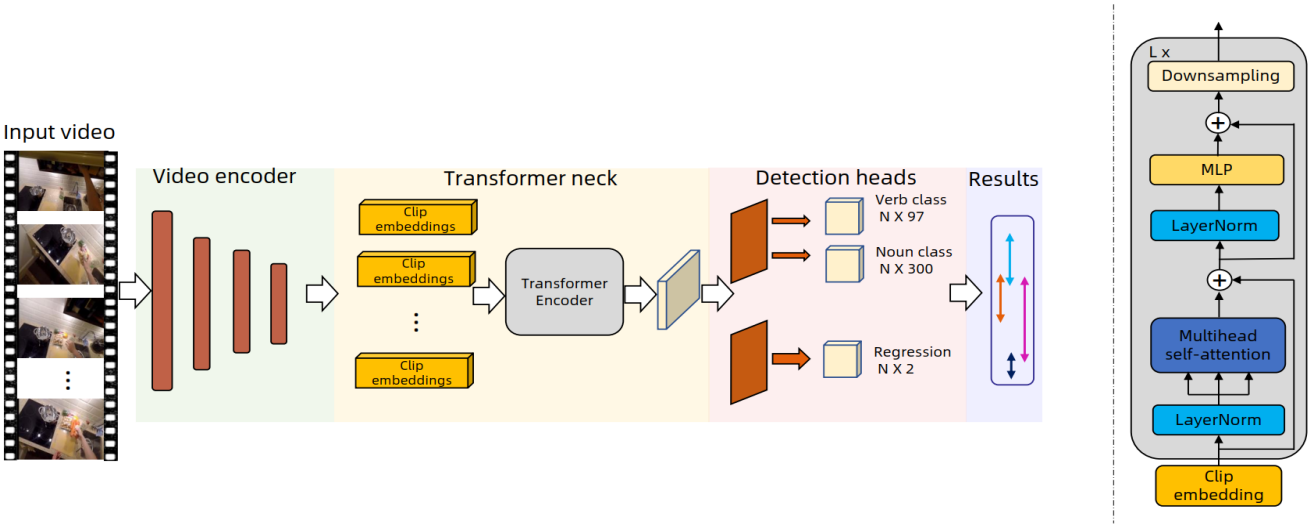


Figure 1. Overview of our proposed OADT. It is composed of three parts: video encoder which extracts clip-level features from untrimmed videos, transformer neck that takes in the clip embeddings and performs self-attention and detection heads which classify the clips and regress the time boundary.

2.2. Transformer Neck

The transformer neck is composed of a sequence of transformer [19] layers. It takes in the clip embeddings obtained by the video encoder and performs self-attention. As is shown in the right of Fig. 1, the basic transformer layer includes the layer norm (LN) operations [1], multi-head self-attention (MHSA), residual connections [9], multi-layer perceptron (MLP) and the downsampling operation. Furthermore, a feature pyramid with different temporal resolutions is created to capture the various temporal range of actions.

2.3. Detection Heads

Different from the two-stage approaches that generate the segment proposals firstly, The detection heads solve the action classification and segment regression in a synchronous manner. The detection heads predict N results directly, where N is the predefined maximum of the proposals. For the regression head, the segments including the begin and end time are predicted by the several full-connection layers. For the classification head, verb and noun are also predicted by the full-connection layers separately for corresponding proposals, and then both are combined into action classification using the simple operations, i.e., addition or multiplication. The focal losses [15] are employed on optimizing verb, noun and action classification, and the 1D IOU losses are used for segment regression.

3. Experiments

Epic-KITCHENS-100 [5] is a large-scale egocentric action dataset. The dataset is very challenging because it contains various kinds of verb and noun classes from fine-grained action videos which capture all daily activities in the kitchen.

3.1. Experimental Details

In this challenge, we employ the video classification methods and pretrain them on Kinetics600 [4] dataset firstly. Then they are finetuned on EPIC-KITCHENS-100 dataset for action recognition. After finetuning, clip-level features are generated with sliding windows. For each sliding window, the time interval is 32 frames and the temporal stride is 16 frames. In the training stage of action detection, the model is trained for 27 epochs and the input resolution is 456$\times$ 256. AdamW [16] optimizer is used with weight decay of 0.0005. The batch size is 2 and the learning rate is set to 0.0001 with the cosine scheduler. We generate the action labels by combining verb and noun predictions. The corresponding time intervals are obtained from the regression head. In inference, Soft-NMS [3] is used for post-processing to suppress redundant action segments.

Evaluation metrics. Mean Average Precision (mAP) is used to evaluate verbs, nouns and actions at different temporal IOU thresholds as well as average mAP. In EPIC-KITCHENS-100 dataset, temporal IOU thresholds range from 0.1 to 0.5 with a step of 0.1. We follow the official split of training, validation and test. For test submission,

Team	Label	Test mAP(%) @0.1	@0.2	@0.3	@0.4	@0.5	Avg
richard61	Verb	22.78	21.68	20.14	18.34	15.54	19.69
	Noun	19.33	17.98	16.55	14.69	12.28	16.17
	Action	14.33	13.63	12.80	11.53	9.93	12.44
Bristol-MaVi	Verb	25.33	23.99	21.91	19.61	17.08	21.58
	Noun	18.99	17.87	16.41	14.43	11.36	15.81
	Action	14.71	13.98	12.86	11.56	9.85	12.59
CTC-AI	Verb	22.62	21.73	20.68	17.74	15.16	19.58
	Noun	20.65	19.58	18.34	16.18	12.88	17.52
	Action	16.68	16.11	15.15	13.59	11.66	14.64
Alibaba-MMAI-Research	Verb	22.77	22.01	19.63	17.81	14.65	19.37
	Noun	26.44	24.55	22.30	19.82	16.25	21.87
	Action	18.76	17.73	16.26	14.91	12.87	16.11
4Paradigm-UWMadison-NJU	Verb	27.11	26.07	24.38	21.96	18.59	23.62
	Noun	28.71	27.27	25.19	22.33	**18.82**	24.47
	Action	23.73	22.87	21.36	19.53	**16.86**	20.87
Ours	Verb	**30.67**	**29.40**	**26.81**	**24.34**	**20.51**	**26.35**
	Noun	**30.96**	**29.36**	**26.78**	**23.27**	18.80	**25.83**
	Action	**24.57**	**23.50**	**21.94**	**19.65**	16.74	**21.28**

Table 1. Final results on EPIC-KITCHENS-100 test set.

Video encoder	Val mAP(%) for Action @0.1	@0.2	@0.3	@0.4	@0.5	Avg
TimeSformer [2]	20.47	19.75	18.69	17.02	14.82	18.15
SlowFast [7]	21.01	20.15	19.02	17.66	15.13	18.59
MVit [6]	22.41	21.44	20.16	18.50	16.10	19.72
Motionformer [17]	22.99	22.08	20.64	18.73	16.09	20.11
Omnivore [8]	25.38	24.50	23.09	21.18	18.72	22.57
Ensemble	**27.19**	**26.23**	**24.38**	**22.47**	**19.82**	**24.02**

Table 2. Detection results on EPIC-KITCHENS-100 validation set.

our model is first trained on the training&validation set and then test on the test set.

Ensemble models. In order to further boost our performance, we apply the five action recognition methods mentioned in Section 2.1 as video encoder separately and train each OADT model. To make full use of different models, we ensemble OADT model trained from different features as our final model.

Results. Tab. 2 shows our results on validation set. From the table, we can see that OADT using Omnivore as the video encoder performs best. While Motionformer and MVit perform slightly worse and are roughly 2% lower. In the end, we ensemble all the five models and can reach 24.02% which is about 1.45% higher than the single best model in action mAP. In Tab. 1, we compare our result to existing state-of-the-art results on the test set. Our solution can get 21.28% mAP which is 5% higher than the winning solution of last year. We outperform prior work especially on verb class by a large margin of +3%.

4. Conclusion

We present our OADT model used in the EPIC-KITCHENS-100 2022 Action Detection challenge. Our model is a one-stage transformer-based architecture composed of the video encoder, transformer block and multiple heads for classification and regression. After ensembling five models, our model can reach the state-of-the-art result of 21.28% average mAP on the EPIC-KITCHENS-100 test set.

Acknowledgement We would like to thank the whole EPIC-KITCHENS team for hosting such a great challenge.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 2

[2] 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*, 2021. 1, 3

[3] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5561–5569, 2017. 2

[4] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018. 2

[5] 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: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, 130(1):33–55, 2022. 2

[6] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6824–6835, 2021. 1, 3

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6202–6211, 2019. 1, 3

[8] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A Single Model for Many Visual Modalities. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2022. 1, 3

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[10] 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, 2012. 1

[11] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 1

[12] Lijun Li and Shuling Dai. Action recognition with deep network features and dimension reduction. *KSII Trans. Internet Inf. Syst.*, 13(2):832–854, 2019. 1

[13] Lijun Li and Boqing Gong. End-to-end video captioning with multitask reinforcement learning. In *IEEE Winter Conference on Applications of Computer Vision*, pages 339–348, 2019. 1

[14] Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. NATTACK: learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 3866–3876, 2019. 1

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2

[16] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 2

[17] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. *Advances in Neural Information Processing Systems*, 34, 2021. 1, 3

[18] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in Neural Information Processing Systems*, 27, 2014. 1

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[20] 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, 2018. 1

[21] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019. 1

[22] Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. Long short-term transformer for online action detection. *Advances in Neural Information Processing Systems*, 34, 2021. 1

[23] Chen-Lin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. *CoRR*, abs/2202.07925, 2022. 1

Detecting Egocentric Actions with ActionFormer

Chenlin Zhang[1,2] Lin Sui[2] Abrar Majeedi[3] Viswanatha Reddy Gajjala[3] Yin Li[3,4]
[1]4Paradigm Inc., Beijing, China
[2]State Key Laboratory for Novel Software Technology, Nanjing University, China
[3]Department of Biostatistics and Medical Informatics, University of Wisconsin-Madison
[4]Department of Computer Sciences, University of Wisconsin-Madison

Abstract

This report describes our submission to EPIC Kitchens 100 action detection challenge 2022. Our submission builds on ActionFormer – our previous work on temporal action localization [15], and integrates latest video features from SlowFast [7] and ViViT [1]. Our solution achieves 21.36 mAP on the validation set and 20.95 mAP on the test set, outperforms previous best results from the 2021 challenge by 4.84 absolute percentage points in average mAP, and is ranked 2nd on the public leaderboard of the 2022 challenge. Our code is available at https://github.com/happyharrycn/actionformer_release.

1. Introduction

Temporal action detection seeks to simultaneously localize action instances in time and recognize their categories. Many prior works have studied action detection in third person videos [2,4,9,10,12,14,16], yet few has focused on egocentric videos. Key challenges arise for egocentric action detection, as manifested in the EPIC-Kitchens dataset [6]. For example, most previous works have considered using action proposals [9] or anchor windows [10] to represent actions in time. An egocentric video, however, often contains hundreds of action instances from many categories spanning from a few seconds to a few minutes, making it difficult to design proposals or anchors.

Our solution instead considers an anchor-free model from our previous work [15]. Our work of ActionFormer presents one of the first Transformer based single-stage anchor-free model, capable of localizing moments of actions in a single shot without using action proposals or predefined anchor windows [15]. ActionFormer adapts local self-attention to model temporal context in untrimmed videos, classifies every moment in an input video, and regresses their corresponding action boundaries.

We explore the integration of different video features in ActionFormer, including SlowFast [7] and ViViT [1] (used by the winning team in the 2021 challenge [11]). We train two separate models for detecting the motion in the action (defined by verbs) and the active objects (defined by nouns), and further combine their outputs for action detection. Our submission achieves 21.36 mAP on the validation set and 20.95 mAP on the test set, outperforms previously best results from 2021 challenge by 4.84 absolute percentage points in average mAP. Our results are ranked 2nd on the public leaderboard of 2022 challenge, with a gap of 0.32 average mAP to the top ranked solution.

2. Our Approach

Our solution firsts extract clip-level video features using pre-trained video backbones. Each clip is thus represented as a feature vector, and each video a sequence of feature vectors. This sequence is further used by ActionFormer for action detection. ActionFormer considers every moment within the sequence as an action candidate, classifies their action category, and regress their action boundaries. We train two separate models to detect motion (verbs) and active objects (nouns), and combine their outputs. In what follows we describe the details of our approach.

2.1. Encoding Video Features

To extract video features, we consider two different video backbones, including (a) a variant (SlowFast R101-NL using 3D ResNet 101 with non-local blocks) of the SlowFast network [7] widely used for video understanding; and (b) a more recent video Transformer model (ViViT [1]) that has proven to be effective on EPIC-Kitchens dataset [8]. Both backbones are pre-trained on third person videos using Kinetics-600 [5]. Following [8], we further fine-tune the backbones on EPIC-Kitchens Action Recognition task, allowing the models to better adapt to egocentric videos. The fine-tuned backbones are then used to extract clip-level video features for action detection.

Fine-tuning on EPIC-Kitchens Action Recognition. Our first step is to fine-tune SlowFast R101-NL and ViViT for

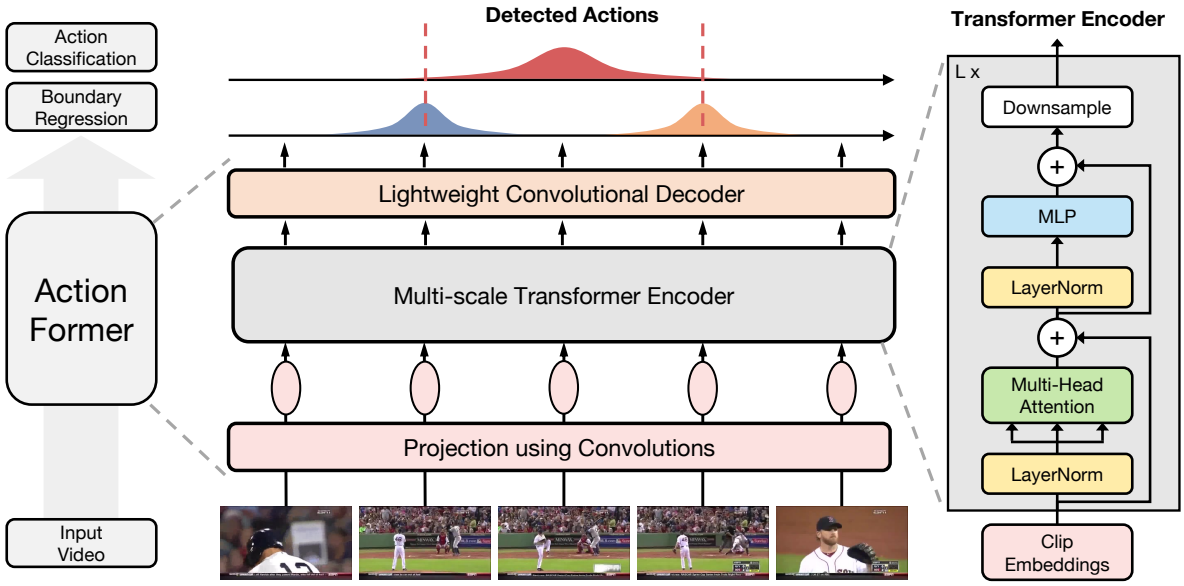


Figure 1. Overview of ActionFormer (taken from our paper [15]). Our method builds a Transformer based model to detect action instances in time by classifying every moment and estimating action boundaries, thereby providing a single-stage anchor-free model for temporal action localization.

action recognition on the training set of EPIC-Kitchens 100.

- **SlowFast R101-NL**: We attach a verb and a noun head to the pre-trained model, and fine-tune all weights on EPIC-Kitchens. Specifically, we randomly sample 32 frames with a temporal stride of 1 from downsampled videos (512×288 at 30 FPS). The model is fine-tuned by 30 epochs with batch size 64, weight decay 0.0001, and initial learning rate 0.01. The learning rates decays by 0.1 at 20th and 25th epoch. The fine-tuned model has 51.6% top-1 noun accuracy and 65.3% top-1 verb accuracy on the validation set with single-crop test.
- **ViViT**: We take the released model from [8], which are already fine-tuned on EPIC-Kitchens. Similar to SlowFast R101-NL, this version of ViViT include separate verb and noun heads for classification. The model reaches 58.9% top-1 noun accuracy and 67.4% top-1 verb accuracy on the validation set with multi-crop test. We refer to [8] for the training details.

Video Feature Extraction. Given the fine-tuned backbones, our next step is to extract clip-level video features for action detection. For both SlowFast and ViViT, we extract a feature vector for every clip of 32 RGB frames with a temporal stride of 8. Optical flow is not used for computing video features.

- **SlowFast R101-NL**: SlowFast network is fully convolutional. Thus, we input video frames with a higher resolution of 512×288, and perform an average pooling before the classification heads to extract a feature vector for each clip. The feature vector is of dimension 2304.
- **ViViT**: ViViT from [8] is trained on a resolution of 320×320 with 60 FPS, yet takes every other frames in the video (temporal stride 2). Altering the input resolution will require interpolating the learned position embeddings. Thus, we downsample the videos to 320×569 at 30 FPS, and feed 32 consecutive frames along with 3 horizontal crops each of size 320×320. The model processes these 3 crops independently, and feature vectors from the CLS token are further averaged to produce a 768-D clip-level feature.

We experimented with using individual features for action detection, yet found that a simple concatenation of the features yields the best performance.

2.2. Temporal Action Detection with ActionFormer

The extracted video features are further used by our ActionFormer for temporal action detection. ActionFormer first embeds each of the clip-level features. The embedded features are further encoded into a feature pyramid using a multi-scale transformer. The resulting feature pyramid is then examined by shared classification and regression heads, predicting action candidates at every time step. Our method is illustrated in Figure 1. We refer the readers to our paper for more technical details [15].

A Two Stream Model. While it is possible to attach separate verb and noun heads in a single ActionFormer model, we found it helpful to train individual models to detect motion (verbs) and active objects (nouns) and then combine their outputs, resembling the key idea of a two stream net-

Split	Method	Feature	Task	mAP@tIoU					
				0.1	0.2	0.3	0.4	0.5	mean
Val	BMN [6, 9]	SlowFast [7]	Verb	10.83	9.84	8.43	7.11	5.58	8.36
			Noun	10.31	8.33	6.17	4.47	3.35	6.53
			Action	6.95	6.10	5.22	4.36	3.43	5.21
	Huang [11]	ViViT [1]	Verb	22.92	21.86	20.89	18.33	15.66	19.93
			Noun	30.09	27.59	25.81	22.80	19.26	25.11
			Action	21.14	20.10	19.02	17.32	15.11	18.53
	Ours (ActionFormer [15])	ViViT [1]	Verb	23.23	22.35	21.28	19.69	16.50	20.61
			Noun	28.85	27.33	25.52	23.01	18.92	24.73
			Action	22.48	21.39	20.24	18.57	16.20	19.78
	Ours (ActionFormer [15])	SlowFast [7]+ViViT [1]	Verb	25.98	24.80	23.26	21.22	18.08	22.67
			Noun	30.49	29.14	26.88	24.77	20.70	26.40
			Action	23.87	22.91	21.70	20.28	18.04	21.36
Test	BMN [6, 9]	SlowFast [7]	Verb	11.10	9.40	7.44	5.69	4.09	7.54
			Noun	11.99	8.49	6.04	4.10	2.80	6.68
			Action	6.40	5.37	4.41	3.36	2.47	4.40
	Huang [11]	ViViT [1]	Verb	22.77	22.01	19.63	17.81	14.65	19.37
			Noun	26.44	24.55	22.30	19.82	16.25	21.87
			Action	18.76	17.73	16.26	14.91	12.87	16.11
	Ours (ActionFormer [15])	SlowFast [7]+ViViT [1]	Verb	26.97	25.90	24.21	21.77	18.47	23.46
			Noun	28.61	27.14	24.92	22.13	18.69	24.30
			Action	23.90	22.98	21.37	19.57	16.94	20.95

Table 1. Results of action detection on EPIC Kitchens 100. All results on the test set are evaluated on the test server. Our method achieves an average mAP of 20.95 for the 2022 challenge, surpassing previous best results from [11].

work [13]. A possible explanation is that doing so facilities implicit model ensemble. Specifically, each stream of ActionFormer predicts the classifications scores ($\boldsymbol{p}(verb)$ or $\boldsymbol{p}(noun)$) and regresses the temporal boundaries ($\boldsymbol{d}(verb)$ or $\boldsymbol{d}(noun)$) at each time step on the feature pyramid. We combine the outputs by using

$$\begin{aligned} \boldsymbol{p}(action) &= \boldsymbol{p}(verb)^{\alpha}\, \boldsymbol{p}(noun)^{(1-\alpha)}, \\ \boldsymbol{d}(action) &= \omega \boldsymbol{d}(verb) + (1-\omega)\boldsymbol{d}(noun), \end{aligned} \tag{1}$$

where $\alpha = 0.45$ (selected based on validation results) is used to "calibrate" the classification scores, and $\omega = \boldsymbol{p}(verb)/(\boldsymbol{p}(verb) + \boldsymbol{p}(noun))$ is used to re-weighted the regression outputs.

Implementation Details. Our model takes the concatenated features (3072-D for each clip with a temporal stride of 8) as the input, uses 6 levels of feature pyramid, and samples a sequence with maximum length of 4608 steps (approximately 20 minutes) for each video during training. The training epochs is 12 and 16 for verb and noun, respectively, as we observed overfitting issues with pro-longed training schedule. The results are further processed using multiclass SoftNMS [3]. We set the maximum predictions of each video to 15,000. Our code will be released in our public repository available at https://github.com/happyharrycn/actionformer_release.

3. Action Detection Results

We now present our results on EPIC Kitchens dataset.

Dataset. Our results are reported on EPIC Kitchens 100 action detection dataset [6]. EPIC Kitchens 100 is the largest egocentric action dataset with more than 100 hours of videos from 700 sessions capturing cooking activities across several kitchen environments. The dataset has an average 128 actions from a large array of categories per session. Each action is defined as a combination of a verb (action) and a noun (active object).

Evaluation Protocol and Metrics. We follow the official splits of train, validation and test set. When reporting results on validation set, we train our model on the training set. For the results on test set, we combine both training and validation sets for training and evaluate the results using the official server. Our results are reported for noun, verb and action, respectively. The metrics include the mean average precision (mAP) at different tIoU thresholds [0.1:0.1:0.5], as well as the average mAP , following [6].

Results. Table 1 summarizes our results on on the validation and test set. When using the same ViViT backbone and evaluated on the validation set, our method reaches an average mAP of 19.73% for action detection in comparison to the previous best result of 18.53% from Huang et al. [11] (also last year's winning solution). Adding SlowFast fea-

tures further improves the average mAP to 22.67%, 26.40%, and 21.36% for verb, noun, and action, respectively, largely outperforming the previous best [11] by 2.74%, 1.29%, and 2.83%. On the test set, our final model achieves 23.46%, 24.30%, and 20.95% mAP on verb, noun, and action, which is 4.09%, 2.43% and 4.84% higher than the previous best results [11]. Our average mAP for actions is slightly lower than the best ranked solution in the 2022 challenge, with a small gap of 0.32%.

4. Conclusion

In this report, we presented our solution using ActionFormer and latest video backbones for temporal action detection in egocentric videos. Notwithstanding its simplicity, our approach has demonstrated strong performance on the EPIC Kitchens dataset, ranked 2nd on the public leaderboard of 2022 challenge, surpassing previous best results and with a gap of 0.32 average mAP to the top ranked solution. We hope that our model can shed light on temporal action localization and egocentric vision, and the more broader problem of video understanding.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. In *Int. Conf. Comput. Vis.*, 2021. 1, 3

[2] Yueran Bai, Yingying Wang, Yunhai Tong, Yang Yang, Qiyue Liu, and Junhui Liu. Boundary content graph neural network for temporal action proposal generation. In *Eur. Conf. Comput. Vis.*, volume 12373 of *LNCS*, pages 121–137, 2020. 1

[3] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-NMS–improving object detection with one line of code. In *Int. Conf. Comput. Vis.*, pages 5561–5569, 2017. 3

[4] Shyamal Buch, Victor Escorcia, Bernard Ghanem, and Juan Niebles Carlos. End-to-end, single-stream temporal action detection in untrimmed videos. In *Brit. Mach. Vis. Conf.*, pages 93.1–93.12, 2017. 1

[5] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the Kinetics dataset. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4724–4733, 2017. 1

[6] 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, 3

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *Int. Conf. Comput. Vis.*, pages 6202–6211, 2019. 1, 3

[8] Ziyuan Huang, Zhiwu Qing, Xiang Wang, Yutong Feng, Shiwei Zhang, Jianwen Jiang, Zhurong Xia, Mingqian Tang, Nong Sang, and Marcelo H Ang Jr. Towards training stronger video vision Transformers for EPIC-Kitchens-100 action recognition. *arXiv preprint arXiv:2106.05058*, 2021. 1, 2

[9] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: Boundary-matching network for temporal action proposal generation. In *Int. Conf. Comput. Vis.*, pages 3889–3898, 2019. 1, 3

[10] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 344–353, 2019. 1

[11] Zhiwu Qing, Ziyuan Huang, Xiang Wang, Yutong Feng, Shiwei Zhang, Jianwen Jiang, Mingqian Tang, Changxin Gao, Marcelo H Ang Jr, and Nong Sang. A stronger baseline for ego-centric action detection. *arXiv preprint arXiv:2106.06942*, 2021. 1, 3, 4

[12] Zheng Shou, Jonathan Chan, Alireza Zareian, Kazuyuki Miyazawa, and Shih-Fu Chang. CDC: Convolutional-de-convolutional networks for precise temporal action localization in untrimmed videos. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5734–5743, 2017. 1

[13] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Adv. Neural Inform. Process. Syst.*, pages 568–576, 2014. 3

[14] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-TAD: Sub-graph localization for temporal action detection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10156–10165, 2020. 1

[15] Chenlin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. *arXiv preprint arXiv:2202.07925*, 2022. 1, 2, 3

[16] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *Int. Conf. Comput. Vis.*, pages 2914–2923, 2017. 1

A Gaussian Boundary Mechanism for Ego-Centric Action Detection

Hanyuan Wang, Dima Damen, Majid Mirmehdi, Toby Perrett
University of Bristol
Bristol, UK
{hanyuan.wang, dima.damen, toby.perrett}@bristol.ac.uk, majid@cs.bris.ac.uk

Abstract

In this technical report, we present an anchor-free model for the EPIC-KITCHENS-100 action detection challenge. It predicts boundaries and confidence scores for each temporal location via three parallel prediction heads. Specifically, we explore the importance the boundary scores in the ranking of candidate proposals. We use a Gaussian Boundary Mechanism to generate the boundary scores for each proposal, which allows for small boundary errors to be penalized less than large errors during training. Finally, we submitted our results to the EPIC Kitchens 100 action detection challenge under the team name of Bristol-MaVi, and achieve 21.6%, 15.8% and 12.6% average mAP for verb, noun and action detection. This outperforms methods from the previous challenge using the same features.

1. Introduction

Temporal action detection aims to predict the start and end timestamp of each action segment in an untrimmed video, and classify them. Most current methods [8,9,11,12] mainly focus on localizing action segments with sliding windows and pre-defined anchors. However, these anchor-based methods are not very suitable for challenging datasets such as EPIC Kitchens 100, which contains short and dense actions. In contrast, anchor-free methods [7, 16, 17] only generate one candidate proposal with classification scores and a pair of relative distances to boundaries.

Evaluating the quality of anchor-free proposals is an open problem. In [7, 17], classification confidence scores are the only criterion used to rank candidate proposals. However, only using classification confidence ignores boundary information. This means precise boundaries may be predicted, but with a low ranking, resulting in high quality predictions being missed.

In order to solve this deficiency, we propose a Gaussian Boundary Mechanism for anchor-free methods to predict the probability that a temporal location is at the start or end of an action. We incorporate our proposal into the ActionFormer action detection method [17].

2. Our Approach

An overview of our method is illustrated in Figure 1. Given an untrimmed video, a feature pyramid is extracted and put in three parallel prediction heads Two parallel classification heads output verb and noun class scores separately. The Gaussian boundary head produces a set of candidate boundaries via a simple 1D convolutional network, and predicts corresponding boundary scores using a Gaussian Boundary Mechanism. The final confidence scores are obtained by multiplying classification scores and boundary scores, and used to rank and filter candidate proposals in Soft-NMS [1].

Section 2.1 offers a formal problem formulation. Section 2.2 introduces how to extract video features and construct the feature pyramid. Section 2.3 shows the network of two classification heads. Section 2.4 presents our main contribution, the Gaussian boundary prediction head. Finally, Section 2.5 gives the details of training and inference.

2.1. Problem Definition

Given an untrimmed video $V = \{v_1, v_2, v_3, ..., v_T\}$ with length T, the annotations of action segments in video V can be denoted as $\Psi = \{(s,\ e,\ v,\ n)\}_{k=1}^{K}$, where K is the total number of ground truth action segments, and s, e, v and n are starting time, ending time and class label of action segments for verb and noun, respectively. The goal of the temporal action detection task is to predict a set of possible action segments $\hat{\Phi} = \{(\hat{s},\ \hat{e},\ \hat{c},\ \hat{v},\ \hat{n})\}_{m=1}^{M}$. Here, $\hat{s}$ and $\hat{e}$ are the starting and ending boundaries, $\hat{c}$ is a confidence score for the proposal, $\hat{v}$ and $\hat{n}$ are the predicted class for verb and noun task separately, and M is the number of predicted action segments. The annotation set Ψ is used to assign training labels. The predicted segments set $\hat{\Phi}$ is expected to cover Ψ with high overlap and recall, so M is likely to be larger than K.

Our model uses an anchor-free presentation to predict action segments, which is also used in [7, 16, 17]. For each temporal location t, it regresses the relative distance $[\hat{r}_t^s, \hat{r}_t^e]$

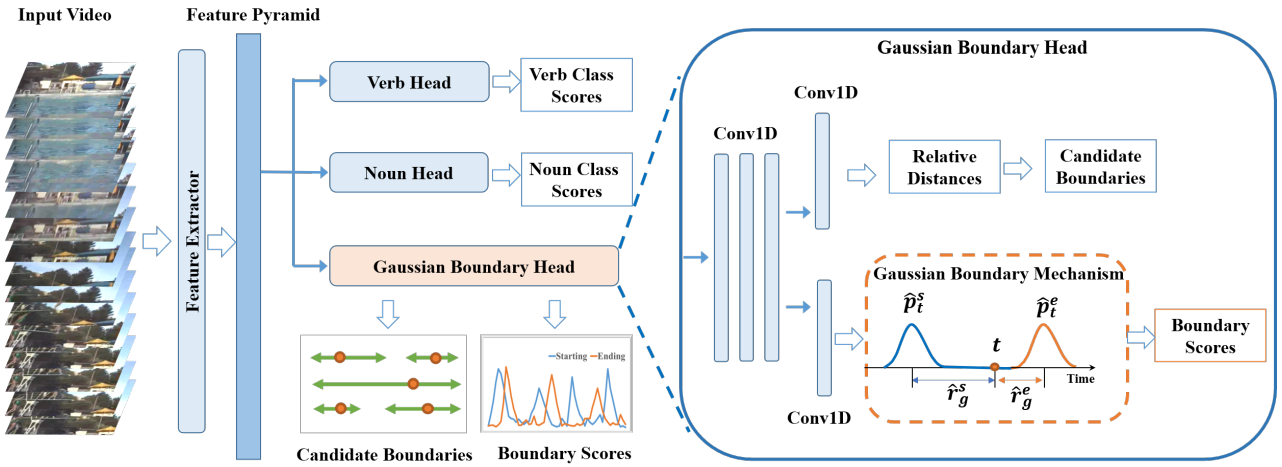


Figure 1. The Overview of our method. Given an untrimmed video, the feature pyramid is extracted and put in three parallel prediction heads. The Gaussian boundary head predicts candidate boundaries, and uses the Gaussian Boundary Mechanism to predict corresponding boundary scores. Two parallel classification heads output verb and noun class scores separately. The final confidence scores are obtained by multiplying classification scores and boundary scores, and used to rank and filter candidate proposals in Soft-NMS.

between the location and corresponding action boundaries. Therefore, the ground-truth relative distance is defined as $r_t^s = t - s$ for starting, and $r_t^e = e - t$ for ending. The starting and ending boundaries of the predictions can be calculated as follows:

$$\hat{s} = t - \hat{r}_t^s \quad \text{and} \quad \hat{e} = t + \hat{r}_t^e \tag{1}$$

This anchor-free manner can get rid of redundant predefined anchors and capture more potential candidate action segments.

2.2. Feature Extraction

Following [17], we use SlowFast [4] trained on the EPIC Kitchens action recognition task [2] to extract video features. A Transformer [15] network uses these features to build the multi-scale feature pyramid $F = \{F_1, F_2, F_3, ..., F_L\}$, where L is the number of layers of the feature pyramid.

2.3. Classification Head

Given F as input, two parallel heads are used to predict the classification scores of each temporal location t for verb and noun separately. These heads consist of two 1D-convolutional layers with ReLU activation function, and a sigmoid function is used to output classification scores $\hat{p}_{c,t}^v$, $\hat{p}_{c,t}^n$ and label $\hat{v}_t$, $\hat{n}_t$. The classification scores are used to calculate the final confidence scores in the inference stage.

2.4. Gaussian Boundary Head

For each temporal location, the Gaussian boundary head predicts the relative distance to starting and ending points, as well as the boundary scores.

Two branches share the same three 1D-convolutional layers. The first branch predicts the relative distance tuple $(\hat{r}_t^s, \hat{r}_t^s)$ by attaching a ReLU at the end, and the second calculates the boundaries $\hat{s}$ and $\hat{e}$ using equation Eq. (1). It first produces the relative distance tuple $(\hat{r}_{g,t}^s, \hat{r}_{g,t}^e)$, similar to the first branch, then uses the relative distance tuple to measure the probabilities that the current temporal location t is an action starting or ending point. Specifically, the boundary scores of temporal location t are defined as:

$$\hat{p}_t^s = e^{-(\hat{r}_{g,t}^s)^2/2\sigma^2} \quad \text{and} \quad \hat{p}_t^e = e^{-(\hat{r}_{g,t}^e)^2/2\sigma^2} \tag{2}$$

where σ is the variance of a Gaussian curve [6]. A temporal location with a large value of $\hat{r}^s$ or $\hat{r}^s$ will have low boundary confidence $\hat{p}_t^s$ or $\hat{p}_t^e$, which indicates that this location is far away from the staring or ending point and thus is unlikely to be a boundary.

2.5. Training and Inference

Label assignment. For the classification head, we directly use the provided verb and noun labels. To supervise the prediction of relative distance, we calculate the distance between the temporal location and corresponding starting and ending points. It's worth noting that only temporal locations around the duration of an action center are selected for training [14, 17, 18]. For the Gaussian Boundary Mechanism, we generate the temporal boundary probabilities p_t^s and p_t^e as the supervision signal. Following BSN [9], we calculate the maximum overlap ratio of ground truth action region to the starting and ending regions, where the starting and ending regions are defined as a specific duration around starting and ending points respectively.

Loss function. We use a multi-task learning strategy to optimize the following loss function:

$$L_{total} = \alpha * L_c + \beta * L_r + \gamma * L_g \tag{3}$$

where L_c is a typical focal loss [10], L_r is the IoU loss [13], L_g is the Gaussian Boundary Mechanism loss, and α, β, γ are the weighting parameters.

For Gaussian Boundary Mechanism loss, we calculate the loss for starting and ending separately using the following MSE losses:

$$L_g^s = \frac{1}{T'} \sum_{t=1}^{T'} (\hat{p}_t^s - p_t^s)^2 \quad \text{and} \quad L_g^e = \frac{1}{T'} \sum_{t=1}^{T'} (\hat{p}_t^e - p_t^e)^2. \tag{4}$$

where T' is the number of temporal locations around the duration of an action center.

Inference. For each location t, we fuse the classification score $\hat{p}_{c,t}^v$, $\hat{p}_{c,t}^n$ and the boundary score $\hat{p}_t^s$, $\hat{p}_t^e$ to calculate the final confidence score, which is used to rank candidate proposals. The final confidence score for each proposal is:

$$\hat{c}_t = \hat{p}_{c,t}^v * \hat{p}_{c,t}^n * \hat{p}_t^s * \hat{p}_t^e \tag{5}$$

These predicted candidate proposals are further processed by Soft-NMS [1] to remove redundant predictions, and obtain the final prediction set $\hat{\Phi} = \{(\hat{s},\ \hat{e},\ \hat{c},\ \hat{v},\ \hat{n})\}_{m=1}^{M}$ for evaluation.

3. Experiments and Results

In this section, we experimentally evaluate the proposed model for EPIC Kitchens 100 action detection task. The dataset and evaluation metrics will be introduced first, followed by the implementation details and results analysis.

3.1. Dataset.

We conduct experiments on EPIC Kitchens 100 dataset [2], which is the largest egocentric action dataset. EPIC Kitchens 100 consists of 700 variable-length videos with 100 hours. The dataset has 90K actions and each action is annotated as a combination of verb and noun.

3.2. Evaluation Metrics.

We use mean Average Precision (mAP) at different Intersection over Union (tIoU) thresholds to evaluate the performance of action detection. Following the official setting, we use IoU thresholds from 0.1 to 0.5 at step size of 0.1. A predicted segment will be defined as a true positive sample if the tIoU with ground truth is greater than the specific threshold.

3.3. Baselines

We compare our approach with two baselines: BMN [8] and LocTransformer [3], both using the same features as ours. BMN is the baseline provided in [2], and LocTransformer is rank 2 for the previous EPIC Kitchens 100 Action Detection challenge, which also uses an anchor-free manner to detect actions.

3.4. Implementation Details.

For feature extraction, the length and stride of sliding window are set to 32 and 16 respectively. For the classification heads, the number of action classes is set to 97 and 300 for verb and nouns. σ in Equation 2 is set to 5.5. In Equation 3, the weighting of total loss function α, β, γ are all set to 0.5. We train the network using the Adam optimizer [5] with a learning rate 0.0001 for 70 epochs.

3.5. Results.

Table 1 presents the performance of our model. On the validation set, our model achieved comparable performance with an average mAP of 14.22% for the main ranking task "action". On the test set, our model reached an average mAP of 12.6% for action task. This outperforms the baselines and shows that our method can localize segments with more precise boundaries. Table 2 shows how performance can be effected by controlling σ. We tried different values from 4.5 to 6.5 for σ in Equation 2, the best is $\sigma = 5.5$.

4. Conclusion

This technical report presents a anchor-free action detector with three parallel prediction heads for EPIC Kitchen 100 action detection challenge. Our detector incorporates a novel Gaussian Boundary Mechanism, which generates boundary scores for each temporal location based on the boundary prediction head. The boundary scores are used in candidate proposals ranking to capture more potential high quality proposals, and further improve the detection performance.

References

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S. Davis. Soft-nms improving object detection with one line of code. In *International Conference on Computer Vision*, 2017. 1, 3

[2] 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. 2, 3

[3] Dima Damen, Adriano Fragomeni, Jonathan Munro, Toby Perrett, Daniel Whettam, and Michael Wray. Epic-kitchens-100- 2021 challenges report. Report, 2021. 3, 4

Method	Task	Val (mAP@IoU)						Test (mAP@IoU)					
		0.1	0.2	0.3	0.4	0.5	Avg.	0.1	0.2	0.3	0.4	0.5	Avg.
BMN [8]	Verb	10.8	9.8	8.4	7.1	5.6	8.4	11.1	9.4	7.4	5.7	4.1	7.5
	Noun	10.3	8.3	6.2	4.5	3.4	6.5	12.0	8.5	6.0	4.1	2.8	6.7
	Action	7.0	6.1	5.2	4.4	3.4	5.2	6.4	5.4	4.4	3.4	2.5	4.4
LocTransformer [3]	Verb	16.1	15.2	14.2	12.7	10.3	13.7	18.3	17.4	16.1	12.5	10.4	14.9
	Noun	15.1	14.1	12.9	10.9	8.7	12.3	15.3	14.3	12.8	10.9	8.4	12.6
	Action	8.5	8.1	7.5	6.5	5.5	7.2	8.8	8.0	7.4	6.3	5.1	7.1
Ours	Verb	22.5	21.5	20.4	18.9	16.5	**20.0**	25.3	24.0	21.9	19.6	17.1	**21.6**
	Noun	21.6	20.3	18.7	16.4	14.3	**18.3**	19.0	17.9	16.4	14.4	11.4	**15.8**
	Action	16.3	15.4	14.5	13.2	11.7	**14.2**	14.7	14.0	13.0	11.6	9.9	**12.6**

Table 1. Action detection results on EPIC Kitchens 100 validation and test set. BMN and LocTransformer are included as baselines. **Bold** for best.

σ	Task	mAP@IoU					
		0.1	0.2	0.3	0.4	0.5	Avg.
4.5	Verb	21.9	20.9	19.9	18.5	16.3	19.5
	Noun	21.3	20.0	18.2	16.5	14.3	18.1
	Action	16.2	15.3	14.4	13.2	11.8	**14.2**
5.5	Verb	22.5	21.5	20.4	18.9	16.5	**20.0**
	Noun	21.6	20.3	18.7	16.4	14.3	**18.3**
	Action	16.3	15.4	14.5	13.2	11.7	**14.2**
6.5	Verb	22.1	21.2	20.3	18.7	16.3	19.7
	Noun	21.0	19.7	18.2	15.9	13.6	17.7
	Action	16.1	15.2	14.5	13.1	11.7	14.1

Table 2. The effect of varying σ on EPIC Kitchens 100 validation set. **Bold** for best.

[4] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 2

[5] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2014. 3

[6] Yanghao Li, Cuiling Lan, Junliang Xing, Wenjun Zeng, Chunfeng Yuan, and Jiaying Liu. Online human action detection using joint classification-regression recurrent neural networks. In *European conference on computer vision*, pages 203–220. Springer, 2016. 2

[7] Chuming Lin, Chengming Xu, Donghao Luo, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Feiyue Huang, and Yanwei Fu. Learning salient boundary feature for anchor-free temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3320–3329, 2021. 1

[8] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. BMN: Boundary-matching network for temporal action proposal generation. In *International Conference on Computer Vision*, 2019. 1, 3, 4

[9] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. BSN: Boundary sensitive network for temporal action proposal generation. In *European Conference on Computer Vision*, 2018. 1, 2

[10] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 3

[11] Qinying Liu and Zilei Wang. Progressive boundary refinement network for temporal action detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11612–11619, 2020. 1

[12] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Gaussian temporal awareness networks for action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 344–353, 2019. 1

[13] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019. 3

[14] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceed-*

ings of the IEEE/CVF international conference on computer vision, pages 9627–9636, 2019. 2

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[16] Le Yang, Houwen Peng, Dingwen Zhang, Jianlong Fu, and Junwei Han. Revisiting anchor mechanisms for temporal action localization. *IEEE Transactions on Image Processing*, 29:8535–8548, 2020. 1

[17] Chenlin Zhang, Jianxin Wu, and Yin Li. Actionformer: Localizing moments of actions with transformers. *arXiv preprint arXiv:2202.07925*, 2022. 1, 2

[18] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9759–9768, 2020. 2

Team VI-I2R Technical Report on EPIC-KITCHENS-100 Unsupervised Domain Adaptation Challenge for Action Recognition 2022

Yi Cheng[1], Dongyun Lin[1], Fen Fang[1], Hao Xuan Woon[1,2], Qianli Xu[1], Ying Sun[1]
[1]Institute for Infocomm Research, Agency for Science, Technology and Research, Singapore
[2]National University of Singapore
{cheng_yi,lin_dongyun,fang_fen,qxu,suny}@i2r.a-star.edu.sg, haoxuan.woon@u.nus.edu

Abstract

In this report, we present the technical details of our submission to the EPIC-KITCHENS-100 Unsupervised Domain Adaptation (UDA) Challenge for Action Recognition 2022. This task aims to adapt an action recognition model trained on a labeled source domain to an unlabeled target domain. To achieve this goal, we propose an action-aware domain adaptation framework that leverages the prior knowledge induced from the action recognition task during the adaptation. Specifically, we disentangle the source features into action-relevant features and action-irrelevant features using the learned action classifier and then align the target features with the action-relevant features. To further improve the action prediction performance, we exploit the verb-noun co-occurrence matrix to constrain and refine the action predictions. Our final submission achieved the first place in terms of top-1 action recognition accuracy.

1. Introduction

The EPIC-KITCHENS-100 dataset is a large-scale video dataset, capturing daily cooking activities in different kitchens using head-mounted cameras [5]. It mainly contains fine-grained actions involving extensive hand object interactions, and each action in the dataset is defined by the combination of a verb and a noun. The Unsupervised Domain Adaptation (UDA) for Action Recognition Challenge aims to learn an action recognition model on a labeled source domain and generalize it to an unlabeled target domain. It has attracted increasing attention from the community as it can significantly alleviate the annotation burden when applying a trained model to other unannotated datasets.

Compared with UDA for image-based tasks, such as image classification and object detection, UDA for video-based tasks is more challenging as both spatial features and temporal dynamics should be aligned during the adaptation. In the task of UDA for Action Recognition, adversarial learning is the dominant approach that aims to learn domain-invariant features for action recognition [2]. Although rapid progress has been made, these methods have one intrinsic limitation, *i.e.*, they directly align source and target features which may degrade the performance of action recognition. It is known that the essence of action recognition is to learn discriminative action-relevant features. Similarly, for UDA for action recognition, it is desirable to ensure that the target features are discriminative enough for correct prediction. However, as the source video features contain both action-relevant and action-irrelevant features, directly aligning source and target features would introduce extra noise and reduce the discriminability of learned features. Therefore, it is important to align the target features with only action-relevant source features.

To address this limitation, we propose to leverage the prior knowledge generated from the action recognition task for video domain adaptation. Specifically, the source feature is first disentangled into action-relevant and action-irrelevant source features using the action classifier learned on the source data, and then the target feature is aligned with the action-relevant source feature. In this manner, the model can learn discriminative domain-invariant features for action recognition. Besides, as each action class is defined as the combination of a verb and a noun, some combinations may be invalid (*e.g.*, rinse & table). We exploit the verb-noun co-occurrence matrix generated from the source domain to constrain and refine the action predictions.

2. Our Approach

In this section, we describe the technical details of our proposed approach. As illustrated in Fig. 1, the overall framework mainly contains two stages: video representation learning and action-aware domain adaptation. We will describe each stage in the following subsections.

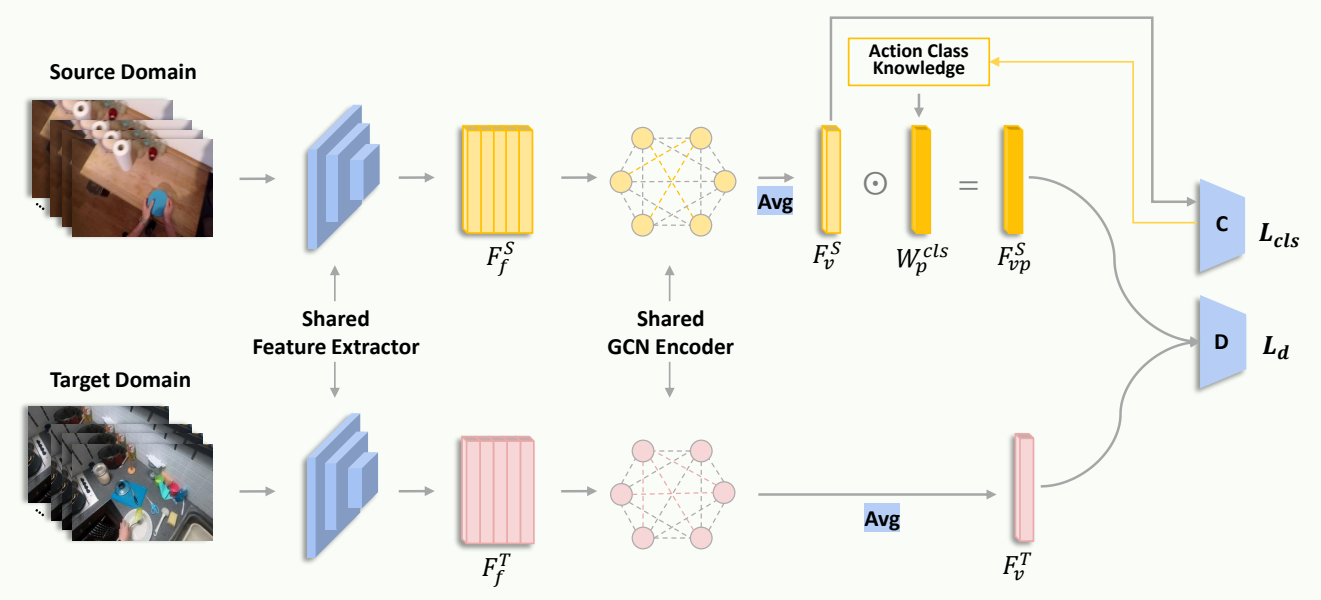


Figure 1. Overall architecture of the proposed framework. First, the frame-level source/target features ($\boldsymbol{F}_f^S$/$\boldsymbol{F}_f^T$) are extracted from video frames using a pre-trained feature extractor. Then, the extracted features are passed to a Graph Convolutional Networks (GCNs), followed by an average operation, to generate the video-level source/target feature ($\boldsymbol{F}_v^S$/ $\boldsymbol{F}_v^T$). Next, $\boldsymbol{F}_v^S$ is passed to the action classifier, and then the action-discriminative features $\boldsymbol{F}_{vp}^S$ are generated using the action class knowledge learned by the action classifier. Lastly, we align $\boldsymbol{F}_v^T$ with $\boldsymbol{F}_{vp}^S$ for video domain adaptation. L_{cls} and L_d denote action classification loss and domain classification loss, respectively. This figure is best viewed in color.

2.1. Video Representation Learning

To learn a robust video feature representation that can generalize across domains for action recognition, it is essential to mine the intrinsic temporal relations within videos. Therefore, we design a video representation learning module that consists of a pre-trained feature extractor for frame feature encoding and a GCN encoder for temporal relation modeling.

Feature extractors. To generate powerful feature representations from the input videos, we explore SlowFast [6], a model based on the 3D Convolutional Neural Network, to extract features from the input video frames. The extracted features are used to generate the video-level features for video domain adaptation.

GCN encoder. As the feature extractor maps individual video frames into the corresponding frame-level features, it does not fully explore the intrinsic temporal structure in videos. Therefore, we apply a fully-connected GCN encoder to model the temporal relations between different video frames. Concretely, we first embed the extracted features from both the source and target domains into the graph space using an FC layer, where the dimension of output features is D. Then, the GCN encoder takes the embedded features as input and outputs a sequence of frame-level features containing rich temporal relation information. Then, we perform average pooling on the output features to generate the video-level feature representations $\boldsymbol{F}_v^S$ and $\boldsymbol{F}_v^T$.

2.2. Action-aware Domain Adaptation

In the task of UDA for action recognition, it is essential to ensure that the shared feature embeddings across domains are discriminative enough for action classification. Therefore, we propose disentangling the action-relevant features from the holistic source features to enable the action-aware alignment with target features.

Grad-CAM [1] is a popular technique to identify the discriminative features for CNN-based classification models [4, 8]. It has been explored in [9–11] that weights of the learned classifier with respect to the ground-truth class can help to identify the critical features for correct class prediction. Motivated by this observation, we propose to use weights of the learned action classifier for the ground-truth action class to generate the action-relevant features that are discriminative for action classification. Concretely, with the video-level source feature $\boldsymbol{F}_v^S$ and the weights of learned action classifier $\boldsymbol{W}_p^{cls}$ for ground-truth class p, the action-relevant feature is computed as:

$$\boldsymbol{F}_{vp}^S = \boldsymbol{W}_p^{cls} \odot \boldsymbol{F}_v^S, \tag{1}$$

where $\odot$ is the Hadamard product, $\boldsymbol{F}_{vp}^S$ is the action-relevant feature containing critical information for action classification.

After obtaining the action-relevant features $\boldsymbol{F}_{vp}^{S}$ from the source domain, features from the target domain are aligned with $\boldsymbol{F}_{vp}^{S}$ using a domain classifier to discriminate whether the sample is from the source or target domain. Following [2], we insert a gradient layer between the domain classifier and the main model for gradient back-propagation. As shown in Fig. 1, the overall framework is optimized using two loss functions: the action classification loss L_{cls} using source action labels and the domain classification loss L_d.

2.3. Verb-noun Co-occurrence Prior

During inference, the video-level target feature $\boldsymbol{F}_{v}^{T}$ is passed to the learned action classifier for action prediction. Since each action class is defined as a combination of a verb and a noun, we design two classification branches: one for predicting the verb probabilities $\boldsymbol{P}_V$ and the other for predicting the noun probabilities $\boldsymbol{P}_N$. Therefore, the action probabilities are computed as:

$$\boldsymbol{P}_A = \boldsymbol{P}_V \boldsymbol{P}_N^T, \tag{2}$$

As mentioned in Section 1, some action classes are invalid because certain verb classes and noun classes are incompatible, such as rinse and table. Therefore, we propose to utilize the co-occurrence of verb and noun as prior knowledge to refine the final predictions on target samples. Concretely, we compute $\boldsymbol{M}$, the co-occurrence matrix of verb and noun, from the statistics of the source domain, where $\boldsymbol{M}_{i,j}$ denotes the number of co-occurrence times of the i-th verb class and the j-th noun class. With the assumption that action classes never appearing in the source domain are highly likely to be invalid, we refine the action probabilities on target samples by reducing the probabilities of invalid action classes:

$$\boldsymbol{P}_A^{'} = \boldsymbol{P}_A \odot \boldsymbol{M}^{'}, \boldsymbol{M}^{'} = \begin{cases} 1, & \text{if } \boldsymbol{M}_{i,j} > 0, \\ 0.01, & \text{otherwise.} \end{cases} \tag{3}$$

3. Experiments

3.1. Implementation Details

Feature extractors. We train three variants of the SlowFast [6], including SlowFast with ResNet50, SlowFast with ResNet101, and SlowOnly (using only the slow path in SlowFast) with ResNet50. For each of the three variants, the model is trained for 60 epochs using synchronized SGD training as in [6]. The input number of frames is set as 32 and 8 for the fast and slow paths, respectively. The batch size is set as 64. During feature extraction, we extract the features from the last convolutional layer and apply average pooling to generate the frame-level feature representations. The feature dimension for SlowOnly-ResNet50 is 2048, while the feature dimensions for SlowFast-ResNet50 and SlowFast-ResNet101 are 2304.

Action-aware domain adaptation. We follow the guidelines posted by the challenges to train the action-aware domain adaptation model. The model is first trained on the validation set for algorithm validation and hyper-parameters tuning. Then, the model is retrained on the training set using the selected hyper-parameters. Finally, the model is applied to predict the action labels of target samples in the testing set, followed by a refinement on the action predictions, to generate the final results. During training, the parameters of the feature extractors are fixed, while the other parameters are learned using an initial learning rate at 3×10^{-3}. The model is trained for 60 epochs, and the learning rate is multiplied by 0.1 after 30 and 45 epochs. We empirically set the dimension of embedded feature vectors as $D = 512$.

3.2. Results

Table 1 demonstrates the recognition performance on the target validation set using the RGB and Flow features extracted from the pre-trained SlowOnly-ResNet50 model. It is observed that by leveraging the action-relevant information from the learned action classifier, the performance of model can be improved by 1.72% in terms of top-1 action accuracy on the validation set. Moreover, the refinement using verb-noun co-occurrence prior information can further improve the top-1 action accuracy by 0.59%. As the EPIC-KITCHENS-100 dataset is highly imbalanced with many tail classes containing very few training samples, the learned action classifier may not be informative enough for the tail classes. Therefore, we expect a higher performance gain on a balanced dataset with enough training samples.

Table 1. The comparison of model performance on the EPIC-KITCHENS-100 validation set. "Baseline" denotes the general domain adaptation without using the information from the learned action classifier. "Baseline+ADA" denotes the Action-aware Domain Adaptation (ADA) proposed in the report. "Baseline+ADA+AF" denotes our proposed method including the Action Refinement (AF). The best performance is marked in bold.

Method	Top-1 Accuracy (%)			Top-5 Accuracy (%)		
	Verb	Noun	Action	Verb	Noun	Action
Baseline	50.33	34.30	22.63	79.75	56.15	48.41
Baseline+ADA	52.75	34.76	24.35	81.33	58.08	50.57
Baseline+ADA+AF	52.75	34.76	**24.94**	81.33	58.08	**51.62**

Table 2. The performance of different models on the EPIC-KITCHENS-100 validation set.

Feature extractor	Input	Top-1 Accuracy (%)			Top-5 Accuracy (%)		
		Verb	Noun	Action	Verb	Noun	Action
SlowOnly (R50)	RGB+Flow	52.75	34.76	24.94	81.33	58.08	51.62
SlowFast (R50)	RGB	49.55	33.35	23.01	80.57	56.03	49.82
SlowFast (R101)	RGB	46.79	34.81	23.24	78.24	56.02	49.70

Table 3. The final results of UDA for domain adaptation on the EPIC-KITCHENS-100 test set.

Method	Top-1 Accuracy (%)			Top-5 Accuracy (%)		
	Verb	Noun	Action	Verb	Noun	Action
Ensemble	57.89	40.07	**30.12**	83.48	64.19	48.10

3.3. Model Ensemble

As model ensemble helps exploit the complementary nature of predictions from different models [12], we ensemble the results from models shown in Table 2. These models are trained on features extracted using the three variants of the SlowFast [6] action recognition model. To further improve the performance, we also ensemble the results from HC-VDA [3] which leverages the hand bounding boxes to generate hand-centric features for video domain adaptation. Following [7], we first calculate the action predictions for each model and then aggregate the results in terms of action probabilities. The final results on the test set are shown in Table 3, and it ranks first in terms of the top-1 action accuracy in the EPIC-KITCHENS-100 UDA Challenge for Action Recognition 2022.

4. Conclusion

In this report, we describe the technical details of our approach to the EPIC-KITCHENS-100 UDA Challenge for Action Recognition 2022. To leverage the action-relevant information that are invariant across domains, we propose an action-aware domain adaptation framework for action recognition. To the best of our knowledge, this is the first work to exploit the prior knowledge induced from the learned action classifier in the task of UDA for action recognition. Moreover, we utilize the verb-noun co-occurrence matrix computed from the source domain data to refine the action predictions. With further performance increase from the model ensemble, our final submission ranks first on the leaderboard in terms of top-1 action recognition accuracy.

Acknowledgments

We would like to thank Dr. Joo Hwee Lim for his continuous support and useful guidance. This research is supported by the Agency for Science, Technology and Research (A*STAR) under its AME Programmatic Funding Scheme (Project #A18A2b0046).

References

[1] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. pages 839–847, 03 2018. 2

[2] 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*, pages 6321–6330, 2019. 1, 3

[3] Yi Cheng, Fen Fang, and Ying Sun. Team VI-I2R technical report on epic-kitchens-100 unsupervised domain adaptation challenge for action recognition 2021. 2021. 4

[4] Yi Cheng, Ying Sun, Hehe Fan, Tao Zhuo, Joo-Hwee Lim, and Mohan Kankanhalli. Entropy guided attention network for weakly-supervised action localization. *Pattern Recognition*, 129:108718, 2022. 2

[5] Dima Damen, Hazel Doughty, Giovanni Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision*, 130:1–23, 01 2022. 1

[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. pages 6201–6210, 10 2019. 2, 3, 4

[7] Ziyuan Huang, Zhiwu Qing, Xiang Wang, Yutong Feng, Shiwei Zhang, Jianwen Jiang, Zhurong Xia, Mingqian Tang, Nong Sang, and Marcelo H Ang Jr. Towards training stronger video vision transformers for epic-kitchens-100 action recognition. *arXiv preprint arXiv:2106.05058*, 2021. 4

[8] Dongyun Lin, Yiqun Li, Yi Cheng, Shitala Prasad, Tin Lay Nwe, Sheng Dong, and Aiyuan Guo. Multi-view 3d object retrieval leveraging the aggregation of view and instance attentive features. *Knowledge-Based Systems*, 247:108754, 2022. 2

[9] Dongyun Lin, Yiqun Li, Shitala Prasad, Tin Lay Nwe, Sheng Dong, and Zaw Min Oo. Cam-unet: Class activation map guided unet with feedback refinement for defect segmentation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 2131–2135. IEEE, 2020. 2

[10] Dongyun Lin, Yiqun Li, Shitala Prasad, Tin Lay Nwe, Sheng Dong, and Zaw Min Oo. Cam-guided multi-path decoding u-net with triplet feature regularization for defect detection and segmentation. *Knowledge-Based Systems*, 228:107272, 2021. 2

[11] Dongyun Lin, Yiqun Li, Shitala Prasad, Tin Lay Nwe, Sheng Dong, and Zaw Min Oo. Cam-guided u-net with adversarial regularization for defect segmentation. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 1054–1058. IEEE, 2021. 2

[12] Y. Sun, Yi Cheng, M. Leong, Hui Li Tan, and Kenan E. Ak. Team VI-I2R technical report on epic-kitchens action anticipation challenge 2020. 2020. 4

Audio-Addaptive Activity Recognition Across Video Domains

Yunhua Zhang Hazel Doughty Cees G. M. Snoek
University of Amsterdam

Abstract

This report strives for activity recognition under domain shift, caused by change of scenery. The leading approaches reduce the shift in activity appearance by fusing RGB and optical flow modalities. Different from these vision-focused works we leverage activity sounds for domain adaptation as they have less variance across domains and can reliably indicate which activities are not happening. We propose an audio-adaptive encoder and associated learning methods that discriminatively adjust the visual feature representation as well as addressing shifts in the semantic distribution. To further eliminate domain-specific features and include domain-invariant activity sounds for recognition, an audio-infused recognizer is proposed, which effectively models the cross-modal interaction across domains. Experiments on the unsupervised domain adaptation challenge show the effectiveness of our approach. Specifically, we achieve the best accuracy in the noun and the action predictions, over all methods that report the results on both target and source domains. The full version of this work has been accepted at CVPR 2022 with more domain shift scenarios. Project page: https://xiaobai1217.github.io/DomainAdaptation.

1. Introduction

The goal of this paper is to recognize activities such as *washing pan*, *cutting onion* or *wiping sink* under domain shift caused by change of scenery, as shown in Figure 1. Existing solutions align distribution-shifted domains inside a single visual video network by adversarial training [3, 16, 22, 24] and self-supervised learning [6, 18, 28]. Although successful, projecting the visual features from different source and target domains into a shared space can make the ability of the model to distinguish between classes in the target domain suffer. We observe that activity sounds can act as natural domain-invariant cues, as they carry rich activity information while exhibiting less variance across domains. We thus propose a video model which adapts to video distribution shifts with the aid of sound.

Many have considered sound in addition to visual analysis for activity recognition within a single domain [13, 20, 21, 23, 26, 31, 32, 35, 36, 38]. For instance, both Gao *et al*. [13] and Korbar *et al*. [20] reduce the computational cost by previewing the audio track, while Lee *et al*. [21] show that combining visual features with audio can better localize actions. However, the cross-modal correspondences become harder to discover when shifting domains, causing existing cross-modal fusion schemes to degrade in performance. Yang *et al*. [37] and Planamente *et al*. [25] propose to directly fuse visual and audio features or predictions for cross-domain activity classification. However, the effectiveness of these methods is reduced when not all activities make a characteristic sound. Different from previous works, we introduce audio-adaptive learning methods and a cross-modal interaction that utilizes the reliable domain-invariant cues within sound to help the video model adapt to the distribution shift.

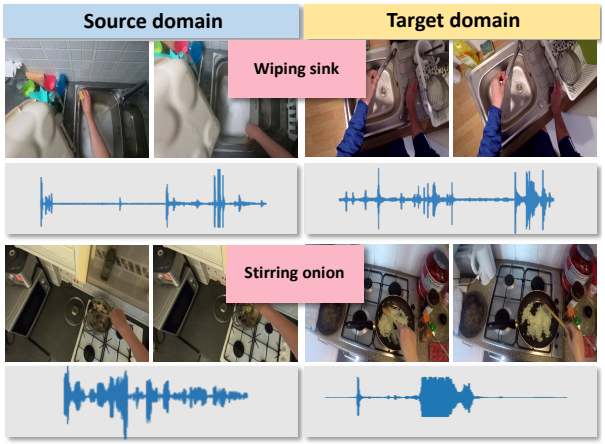


Figure 1. We recognize activities under domain shifts, caused by change of scenery, with the aid of sound.

We introduce two technical contributions in this report. First, we propose an audio-adaptive encoder which exploits the rich information from sound to adjust the visual feature representation causing the model to learn more discriminative features in the target domain. This is done by preventing the model from over-fitting to domain-specific visual content, while simultaneously dealing with imbalanced semantic distributions between domains. Second, we introduce an audio-infused recognizer, which eliminates domain-specific

features further and allows effective cross-modal interaction across domains by considering domain-invariant activity information within sound. Experiments on EPIC-Kitchens unsupervised domain adaptation challenge [8] demonstrate the advantage of our approach under scenery shift. While achieving good performance int the target domain, we are the best in the source among submitted results.

2. Related Work

Sound for activity recognition. Many works have utilized sound for within-domain activity recognition in videos, *e.g.*, [13,17,20,21,31,32]. Since there is a natural correlation between the visual and auditive elements of a video, Korbar *et al*. [19] and Asano *et al*. [1] learn audio-visual models in a self-supervised manner. As processing audio signals is much faster than video frames, both Gao *et al*. [13] and Korbar *et al*. [20] reduce computation by previewing the audio track for video analysis. Cross-modal attention is widely used in activity localization [21, 32, 36] and audiovisual video parsing [31, 35] to guide the visual model to focus on the audible regions. Zhang *et al*. [38] conduct repetitive activity counting by using audio signals to decide the sampling rate and predict the reliability of the visual features. As opposed to most works which rely on sound for within-domain activity recognition, we consider its domain-invariant nature for activity recognition across different domains.

Video domain adaptation by vision. The field of vision-focused domain adaptation is extensive (see recent surveys [34, 40]). Here, we focus on video domain adaptation for activity recognition. State-of-the-art visual-only solutions learn to reduce the shift in activity appearance by adversarial training [3–6, 16, 22, 24] and self-supervised learning techniques [6, 18, 22, 28]. While Jamal *et al*. [16] and Munro and Damen [22] directly penalize domain specific features with an adversarial loss at every time stamp, Chen *et al*. [3], Choi *et al*. [6] and Pan *et al*. [24] attend to temporal segments that contain important cues. Self-supervised learning objectives are also incorporated in [22] and [6] to better align the features across domains by utilizing the correspondences between RGB and optical flow or the temporal order of video clips. Song *et al*. [28] and Kim *et al*. [18] obtain remarkable performance by contrastive learning for self-supervised learning to align the feature distributions between video domains. Instead of relying on the vision modality only, which may present large activity appearance variance, we consider the domain-invariant information within sound to help the model adapt to the visual distribution shift.

Video domain adaptation by vision and audio. As audio signals contain valuable domain-invariant cues, some recent works recognize activities across domains with the aid of sound. Yang *et al*. [37] directly fuse the features from visual and audio modalities before classification. However, this can lead to the visual features dominating the classification since many activities are silent and the audio features are less discriminative. As a result, the complementary information from sound may not be considered. Planamente *et al*. [25] instead align the two modalities with an audio-visual loss. Nonetheless, the audio predictions for silent activities remain unreliable and limit their performance improvements. Instead, we propose audio-adaptive learning that exploits the supervisory signals from sound to adjust to the distribution shift and handle both audible and silent activities.

3. Approach

For activity recognition under domain shift, we consider unsupervised domain adaptation where we have: a set of labeled source videos $\mathcal{S}=\{(X_1^{\mathcal{S}}, y_1^{\mathcal{S}}), \ldots, (X_N^{\mathcal{S}}, y_N^{\mathcal{S}})\}$ and a set of unlabeled target videos $\mathcal{T}=\{X_1^{\mathcal{T}}, \ldots, X_M^{\mathcal{T}}\}$. In each domain, X and y indicate a video sample and the corresponding activity class label, while N and M are the number of samples in the source and target domain. Using all available training data from the source and the target domains, the task is to train an activity recognition model, which performs well on (unseen) videos from the target domain.

We train our audio-adaptive model in two stages using videos from source and target domains with accompanying audio. In the first stage we train our audio-adaptive encoder (Section 3.1) that uses audio to adapt a visual encoder to be more robust to distribution shifts. In the second stage we train our audio-infused recognizer (Section 3.2) using pseudo-labels from the audio-adaptive encoder for the target domain and the ground-truth labels for the source domain. The audio-infused recognizer maps the source and target domains into a common space and fuses audio and visual features to produce an activity prediction for either domain.

3.1. Stage 1: Audio-Adaptive Encoder

Our audio-adaptive encoder $\mathcal{E}(\cdot)$, detailed in Figure 2, consists of a visual encoder $\mathcal{V}(\cdot)$, an audio encoder $\mathcal{A}(\cdot)$ and an audio-based attention module $\psi(\cdot)$. Since the sounds of activities have less variance across domains, $\mathcal{E}(\cdot)$ aims to extract visual features that are invariant but discriminative under domain shift with the aid of $\mathcal{A}(\cdot)$ pre-trained for audio-based activity recognition. To this end, we train $\mathcal{V}(\cdot)$ and $\psi(\cdot)$ with two audio-adaptive learning methods: absent-activity learning for unlabeled target data and audio-balanced learning for labeled source data. The former aims to remove irrelevant parts of the visual features while the latter helps to handle the differing label distribution between domains. Once trained, for each video, we can extract an audio feature vector from $\mathcal{A}(\cdot)$ and a series of visual features from $\mathcal{V}(\cdot)$ with which to train our audio-infused recognizer (Section 3.2) for activity classification.

Audio-based attention. We use an audio-based attention module $\psi(\cdot)$ to adapt the visual encoder to focus on activity-relevant features. For example, the visual model may pre-

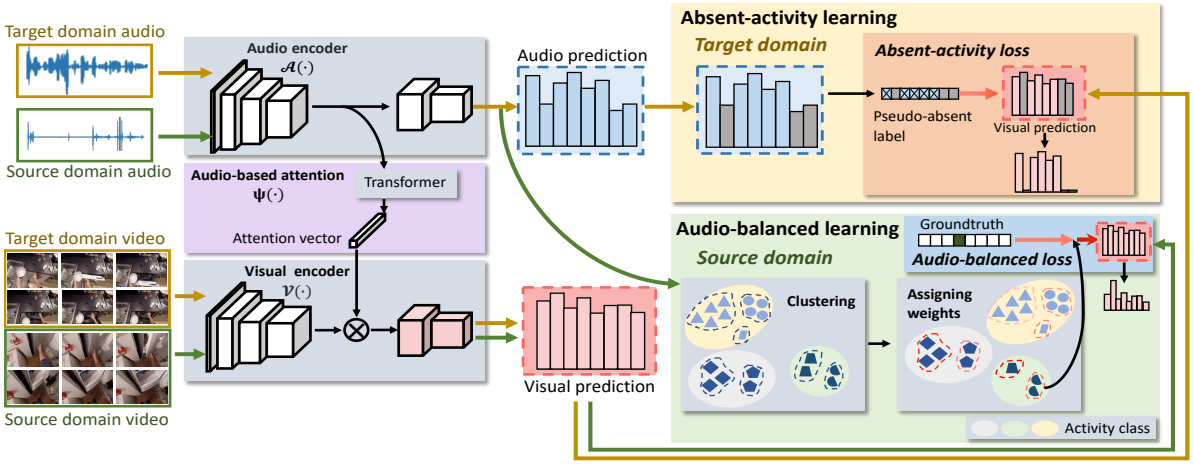


Figure 2. **Audio-adaptive encoder for activity recognition under domain shift.** With a pre-trained audio encoder, we train the visual encoder and audio-based attention module, which guides the visual encoder to focus on the activity relevant features. We do this with two audio-adaptive learning methods: absent-activity learning and audio-balanced learning. The absent activity learning operates in the target domain and uses the audio predictions to indicate which activities cannot be heard in the video. The visual predictions are then encouraged to have low probabilities for these 'pseudo-absent' activities. The audio-balanced learning uses audio in the source domain to cluster samples in each activity class into clusters according to the sounds of the object/environment interacted with. In the audio-balanced loss the rare activities and interactions are weighted higher to handle the semantic shift between domains.

dict the activity *washing* because of the presence of a sink. However, without the sound of water the attention module suppresses the channels encoding the sink thus increasing the prediction of the correct class. The attention module is based on the transformer encoder [9, 10, 33]. It takes the audio features as input and outputs the channel attention feature vector, which is multiplied with the visual features.

Absent-activity learning. The absent-activity learning uses audio in the target domain to train the attention module and visual encoder. Naively, we could treat the class with the highest probability from the visual encoder as the pseudo label. However, doing so can create biased pseudo-labels as irrelevant objects often appear in a scene. Instead, we use the audio predictions to guide the visual pseudo-labels. While we may not be confident which activity is happening in a video, particularly for silent videos, we can often be confident that certain activities with distinctive sounds are *not* occurring in a video. We call these "absent activities". To learn from these absent activities, we generate pseudo-absent labels for the unlabeled target domain videos, which indicate the activities with the lowest probabilities from the audio encoder. The visual encoder is then encouraged to predict these unlikely classes with low probability.

Specifically, for an unlabeled video $X^{\mathcal{T}}$ in the target domain, we obtain the audio-based activity probability distribution $\mathbf{p}_a^{\mathcal{T}} \in \mathbb{R}^K$ (K is the number of classes) from the audio encoder $\mathcal{A}(\cdot)$ trained on labeled source data. From this we obtain the set of absent activities $\mathcal{Q}$ by taking the lowest r predictions in $\mathbf{p}_a^{\mathcal{T}}$, *i.e.*, the classes with the lowest probabilities from the audio encoder. We also extend this to multi-label classification by instead assuming the $(1-\alpha_k)\gamma$ percent videos with the lowest probabilities do not contain class k, where $\gamma \in (0, 1]$ and α_k is the percentage of videos containing each activity class in the labeled source domain.

Our loss for absent-activity learning is formulated as:

$$l_{\mathrm{A}}(\mathbf{p}_v^{\mathcal{T}}, \mathcal{Q}) = -\sum_{q \in \mathcal{Q}} \log(1 - p_{v,q}^{\mathcal{T}}), \tag{1}$$

where $p_{v,q}^{\mathcal{T}}$ is the probability output for the qth class for the video $X^{\mathcal{T}}$. With this loss, the visual encoder is able to ignore confounding visual features and generate less-noisy pseudo-labels for the target domain. This allows our model to better capture high-level semantic information between domains based on both appearance and motion cues.

Audio-balanced learning. Besides a change in visual appearance, domain shift can also be caused by a change in label distributions [22] and frequencies of objects/environments. For example, the *open* activity may commonly occur on a 'cupboard' in the source domain but be more common with a 'can' in the target. These two cases result in different audio-visual activity appearances. We address such challenges with our audio-balanced learning, which not only handles imbalance in activity classes, but also imbalance in terms of the objects or the environment being interacted with.

To this end, we first use k-means to group the video samples inside each activity class by their audio feature $\mathbf{f}_a^{\mathcal{S}}$ with the assumption that each group represents a different type of object or environment. We use audio features for clustering as they can indicate the material of the interacted objects or the environment the action is performed in, while being

invariant to appearance changes. The number of interaction clusters per activity class is determined by the Elbow method [30], which favours a small number while obtaining a low ratio of dispersion both between and within clusters.

We based our *audio-balanced* loss on the class-balanced loss by Cui *et al*. [7]. When using the original class-balanced loss on a source domain video $X^{\mathcal{S}}$ with visual probabilities $\mathbf{p}_v^{\mathcal{S}}$ we can balance over our activity classes:

$$l_{\text{CB}}(\mathbf{p}_v^{\mathcal{S}}, y^{\mathcal{S}}) = \frac{1-\beta}{1-\beta^{n_y}} \mathcal{L}(\mathbf{p}_v^{\mathcal{S}}, y^{\mathcal{S}}), \tag{2}$$

where $\mathcal{L}$ is a classification loss, *e.g*., softmax cross-entropy loss and n_y is the number of training samples of ground-truth activity class y. $\beta \in [0, 1)$ is a hyper-parameter which controls the weighting factor $\frac{1-\beta}{1-\beta^{n_y}}$. As $\beta \to 1$, this weighting factor becomes inversely proportional to the effective number of samples inside each class so that tail classes in the source domain are weighted higher in training.

With our *audio-balanced loss* we include an additional weighting factor so the long tail of object interactions are also accounted for with our interaction clusters:

$$l_{\text{B}}(\mathbf{p}_v^{\mathcal{S}}, y^{\mathcal{S}}) = \frac{1-\beta}{1-\beta^{n_{y,j}}} l_{\text{CB}}(\mathbf{p}_v^{\mathcal{S}}, y^{\mathcal{S}}). \tag{3}$$

$n_{y,j}$ is the number of samples for the jth interaction cluster that video $X^{\mathcal{S}}$ is assigned within ground-truth activity $y^{\mathcal{S}}$. By this loss, both rare activities and rare interactions from frequent activities are given a high weight during training. This means the classifier can generalize well to the target domain where the distribution of activities and interactions may not be the same.

Audio-adaptive encoder loss. The absent-activity loss and the audio-balanced loss are combined to obtain the overall loss for training the visual encoder $\mathcal{V}(\cdot)$ and audio-based attention $\psi(\cdot)$ inside the audio-adaptive encoder $\mathcal{E}(\cdot)$:

$$l_{\mathcal{E}} = \sum_{(X_i)\in\mathcal{T}} l_{\text{A}}(\mathbf{p}_{i,v}^{\mathcal{T}}, \mathcal{Q}_i) + \sum_{(X_j,y_j)\in\mathcal{T}} l_{\text{B}}(\mathbf{p}_{j,v}^{\mathcal{S}}, y_j^{\mathcal{S}}). \tag{4}$$

3.2. Stage 2: Audio-Infused Recognizer

While audio can help focus on the activity-relevant visual features, there is still a large difference between the appearance of activities in different domains. To further eliminate domain-specific visual features and fuse the activity cues from the audio and visual modalities we propose the audio-infused recognizer $\mathcal{R}(\cdot)$, visualized in Figure 3.

Transformer with domain embedding. We adopt a transformer encoder since its core mechanism, self-attention, can efficiently encode multi-modal representations [12, 29, 39]. For a vanilla version, we take the input sequence:

$$\mathbf{z}^m = [\mathbf{z}_{\text{cls}}^m; \mathbf{f}_{v,1}\mathbf{E}_v; \cdots; \mathbf{f}_{v,n}\mathbf{E}_v; \mathbf{f}_{a,1}\mathbf{E}_a, ; \cdots; \mathbf{f}_{a,n}\mathbf{E}_a], \tag{5}$$

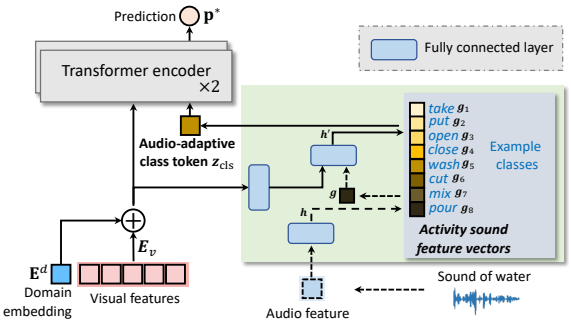


Figure 3. **Audio-infused recognizer.** We add domain embedding E_d to encourage a common visual representation across domains. Then, an audio-adaptive class token is obtained from a series of activity sound feature vectors, considering both audio and visual features. It is sent into the transformer together with the visual features. By the transformer's self attention, this token aggregates information from visual features with the domain-invariant audio activity cues for activity classification.

where $\mathbf{z}_{\text{cls}}^m$ is the learnable class token defined as in [10], and $\{\mathbf{f}_{v,1}, \cdots, \mathbf{f}_{v,n} | \mathbf{f}_{v,\cdot} \in \mathbb{R}^{C_v}\}$ and $\{\mathbf{f}_{a,1}, \cdots, \mathbf{f}_{a,n} | \mathbf{f}_{a,\cdot} \in \mathbb{R}^{C_a}\}$ are the visual and audio features of n clips from video X. $\mathbf{E}_v \in \mathbb{R}^{C_v \times D}$ and $\mathbf{E}_a \in \mathbb{R}^{C_a \times D}$ are linear projections to map the visual and audio features to D dimensions. To map source and target domains into a common space, we first learn a domain embedding $\mathbf{E}^d \in \mathbb{R}^D$ ($d \in \{\mathcal{S}, \mathcal{T}\}$), which contains both positive and negative values and is added to suppress domain-specific visual features. Then, the input sequence for the transformer becomes:

$$\mathbf{z}' = [\mathbf{z}_{\text{cls}}^m; \mathbf{f}_{v,1}\mathbf{E}_v + \mathbf{E}^d; \cdots; \mathbf{f}_{v,n}\mathbf{E}_v + \mathbf{E}^d; \mathbf{f}_{a,1}\mathbf{E}_a, ; \cdots; \mathbf{f}_{a,n}\mathbf{E}_a]. \tag{6}$$

Audio-adaptive class token. Ideally, the transformer's self attention will aggregate audio and visual features with the class token to predict the correct activity. However, the cross-modal correspondences are difficult to find under distribution shift, meaning the prediction may rely on the more discriminative, but less domain-invariant, visual features. To address this, we propose to generate an audio-adaptive class token, which is initialized from the audio activity class prediction and gradually aggregates the visual features while keeping its own audio-based activity information through the transformer. As shown in Figure 3, the audio-adaptive class token is obtained from a series of activity sound vectors $\{\mathbf{g}_k \in \mathbb{R}^D\}_{k=1}^K$, with each representing an activity class. They capture global context information and serve as the representation bottleneck to provide regularization for model learning [2, 27]. For selection, the feature vector from the audio adaptive encoder $\mathcal{A}(X)$ is first processed by a fully connected layer to give the activity probabilities $\mathbf{h} \in \mathbb{R}^K$. Then, an initial vector is obtained by $\mathbf{g} = \sum_{k=1}^K h_k * \mathbf{g}_k$. We include visual features to help silent activities select the representative vector. To avoid the visual features dominating, we project them to a lower dimension with a fully connected

layer before concatenating them with the initial vector $\mathbf{g}$. The concatenated vector is given to another fully connected layer which outputs the probabilities $\mathbf{h}'$ for each type of activity sound. Finally, we obtain the audio representation $\mathbf{z}_{\text{cls}}=\sum_{k=1}^{K} h'_k * \mathbf{g}_k$, which serves as the class token. Consequently, the input sequence for the transformer becomes:

$$\mathbf{z}=[\mathbf{z}_{\text{cls}}; \mathbf{f}_{v,1}\mathbf{E}_v + \mathbf{E}^d, ; \cdots ; \mathbf{f}_{v,n}\mathbf{E}_v + \mathbf{E}^d], \quad (7)$$

where $\mathbf{z}_{\text{cls}}$ is the audio-adaptive class token. The class token output state is further sent to a fully connected layer to get the final prediction $\mathbf{p}^*$. For audible activities, the activity sound vector can be accurately selected and kept discriminative for audiovisual interaction. For silent activities, the vector is obtained from environmental sound, which indicates the presence of multiple possible activities. The vector becomes more discriminative as the transformer progressively enhances it through the visual features.

Audio-infused recognizer loss. We train the audio-infused recognizer on both source and target videos with the loss:

$$l_{\mathcal{R}} = \sum_{(X_i,y_i)\in\{\mathcal{S},\mathcal{T}\}} \mathcal{L}(\mathbf{p}_i^*, y_i) + \eta\Big(\mathcal{L}(\mathbf{h}_i, y_i) + \mathcal{L}(\mathbf{h}'_i, y_i)\Big), \quad (8)$$

where hyperparameter η balances the loss terms and y_i is the groundtruth or, in the case of the unlabeled video, the hard pseudo-label. $\mathbf{p}_i^*$ is the final classification prediction, and $\mathbf{h}_i$ and $\mathbf{h}'_i$ are the probabilities for the activity sound vectors outputted by the first and second fully connected layers. The first term $\mathcal{L}(\mathbf{p}_i^*, y_i)$ optimizes the transformer to predict the correct activity class, while the second term $\mathcal{L}(\mathbf{h}_i, y_i) + \mathcal{L}(\mathbf{h}'_i, y_i)$ optimizes the activity sound vectors.

4. Results

We first describe the implementation details before ablating the components of our method. More ablations and comparisons with state-of-the-art methods can be found in our project page.

4.1. Implementation Details

For our visual encoder $\mathcal{V}(\cdot)$ we use SlowFast [11] for verb prediction while use the Omnivore [14] for the noun prediction. For the audio encoder $\mathcal{A}(\cdot)$ we use ResNet-18 [15]. The audio-based attention module $\psi(\cdot)$ consists of eight transformer encoder layers [10] with a final fully connected layer to obtain the attention vector for the visual encoder. The inputs are intermediate audio features from $\mathcal{A}(\cdot)$ (conv3) along with a learnable class token defined as in [10] (note this is different from our audio-adaptive class token used in $\mathcal{R}(\cdot)$). The output state of the class token passes through the fully connected layer to obtain the attention vector for $\mathcal{V}(\cdot)$. We set the parameters of our absent activity loss to r=75 for verb prediction, while r=270 for noun prediction, and set γ=0.05 and β=0.999. Our audio-infused recognizer $\mathcal{R}(\cdot)$ consists of two transformer encoder layers [10] and three fully connected layers for generating the class token. The sequence dimension D is 512 and each layer has 8 self-attention heads. We train four audio-adaptive models with the SlowFast architecture as the backbone for the visual encoder independently to predict the verbs, and let them take the inputs of different numbers of frames and sampling rates, *i.e*. 32 frames with stride 4, 64 frames with stride 2, 32 frames with stride 2 and 64 frames with stride 1. For the noun prediction, three audio-adaptive models with the Omnivore [14] backbone for the visual encoder are also learned, taking 32 frames with stride 4, 64 frames with stride 21, 32 frames with stride 2 as inputs respectively. We use the average prediction as the final prediction.

Model	EPIC-Kitchens Top-1 (%) ↑
Stage 1: Audio-adaptive encoder $\mathcal{E}(\cdot)$	
Visual encoder $\mathcal{V}(\cdot)$	48.0
+ Audio-based attention $\psi(\cdot)$	51.2
+ Absent-activity learning	53.7
+ Audio-balanced learning	55.7
Stage 2: Audio-infused recognizer $\mathcal{R}(\cdot)$	
+ Vanilla multi-modal transformer $\mathbf{z}^m$	56.1
+ Domain embedding $\mathbf{z}'$	57.2
+ Audio-adaptive class token $\mathbf{z}$	59.2

Table 1. **Model components ablation.** All components in the audio-adaptive encoder and the audio-infused recognizer contribute to performance improvement under distribution shift and the improvements over a vanilla SlowFast visual encoder are considerable.

4.2. Ablation Study

In ablations we use RGB and audio modalities on EPIC-Kitchens under the same setting described in [22] for predicting the verbs. Since EPIC-Kitchens [22] contains multiple adaptation settings, we report the average.

Stage 1: Audio-adaptive encoder. We report results in Table 1. We first consider the audio-adaptive encoder alone. Initially, we train only the visual encoder with a standard softmax cross-entropy loss on the source domain. Simply generating channel attention for the visual features with our audio-based attention module already improves performance by 3.2% top-1 accuracy on EPIC-Kitchens. Since audio contains useful activity information, this attention helps the visual encoder focus on relevant features. Adding the absent-activity learning results in 2.5% improvements, demonstrating that the pseudo-absent labels increase the discriminative ability of the model in the target domain. We observe that adopting the audio-balanced learning and replacing the softmax cross-entropy with our audio-balanced loss delivers a further 2.0% increase. This highlights the importance of addressing the label distribution shift in domain adaption.

Stage 2: Audio-infused recognizer. For the audio-infused recognizer, we first consider a vanilla transformer. It takes as input $\mathbf{z}^m$ (Eq. 5), *i.e*. the audio and visual features from the audio-adaptive encoder, mapped by $\mathbf{E}_v$ and $\mathbf{E_a}$ into a common space, alongside a learnable class token. This only gives a marginal improvement in results. Adding the domain embedding $\mathbf{E}^d$ to reduce domain-specific visual features in $\mathbf{z}'$ (Eq. 6) gives a benefit of 1.1% on EPIC-Kitchens. This is because the cross-modal correspondences become easier to discover. When we replace the plain audio features and single learnable class token with our audio-adaptive class token to get $\mathbf{z}$ (Eq. 7), we observe further improvements of 2.0%. This is expected, as the audio-adaptive class token better incorporates complementary information from sound for the final activity classification, with a standard learnable class token the visual features will dominate the fusion inside the transformer.

5. Discussion

Limitations. During training, our method needs videos from both source and target domains, and all should have an audio track with decent quality, limiting our approach to multi-modal video training sets. While audio at test-time is not required, it benefits activity recognition results considerably.

Potential negative impact. When deployed our approach will have to record, store and process video and audio information related to human activities, which will have privacy implications for some application domains.

Conclusions. We propose to recognize activities under domain shift with the aid of sound, using a novel audiovisual model. By leveraging the domain-invariant activity information within sound, our model improves over both silent and audible activities as well as rare activities in the source domain. Experiments on the unsupervised domain adaptation challenge demonstrate that our approach has better adaptation ability than previous visual-only solutions and audio-visual method with late fusion. Specifically, among all highly ranked methods that report results on both target and source domains, we perform best in the noun and action prediction.

References

[1] Yuki M. Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. In *NeurIPS*, 2020. 2

[2] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In *ICML*, 2018. 4

[3] Min-Hung Chen, Zsolt Kira, Ghassan AlRegib, Jaekwon Yoo, Ruxin Chen, and Jian Zheng. Temporal attentive alignment for large-scale video domain adaptation. In *ICCV*, 2019. 1, 2

[4] Min-Hung Chen, Baopu Li, Yingze Bao, Ghassan Al-Regib, and Zsolt Kira. Action segmentation with joint self-supervised temporal domain adaptation. In *CVPR*, 2020. 2

[5] Jinwoo Choi, Gaurav Sharma, Manmohan Chandraker, and Jia-Bin Huang. Unsupervised and semi-supervised domain adaptation for action recognition from drones. In *WACV*, 2020. 2

[6] Jinwoo Choi, Gaurav Sharma, Samuel Schulter, and Jia-Bin Huang. Shuffle and attend: Video domain adaptation. In *ECCV*, 2020. 1, 2

[7] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019. 4

[8] 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. 2

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 3, 4, 5

[11] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 5

[12] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *ECCV*, 2020. 4

[13] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. In *CVPR*, 2020. 1, 2

[14] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. *arXiv preprint arXiv:2201.08377*, 2022. 5

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5

[16] Arshad Jamal, Vinay P Namboodiri, Dipti Deodhare, and KS Venkatesh. Deep domain adaptation in action space. In *BMVC*, 2018. 1, 2

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

[18] 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 *ICCV*, 2021. 1, 2

[19] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018. 2

[20] Bruno Korbar, Du Tran, and Lorenzo Torresani. SCSampler: Sampling salient clips from video for efficient action recognition. In *ICCV*, 2019. 1, 2

[21] Jun-Tae Lee, Mihir Jain, Hyoungwoo Park, and Sungrack Yun. Cross-attentional audio-visual fusion for weakly-supervised action localization. In *ICLR*, 2021. 1, 2

[22] Jonathan Munro and Dima Damen. Multi-modal domain adaptation for fine-grained action recognition. In *CVPR*, 2020. 1, 2, 3, 5

[23] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. In *NeurIPS*, 2021. 1

[24] Boxiao Pan, Zhangjie Cao, Ehsan Adeli, and Juan Carlos Niebles. Adversarial cross-domain action recognition with co-attention. In *AAAI*, 2020. 1, 2

[25] Mirco Planamente, Chiara Plizzari, Emanuele Alberti, and Barbara Caputo. Cross-domain first person audio-visual action recognition through relative norm alignment. *arXiv preprint arXiv:2106.01689*, 2021. 1, 2

[26] Nishant Rai, Haofeng Chen, Jingwei Ji, Rishi Desai, Kazuki Kozuka, Shun Ishizaka, Ehsan Adeli, and Juan Carlos Niebles. Home action genome: Cooperative compositional action understanding. In *CVPR*, 2021. 1

[27] Alexandre Rame and Matthieu Cord. Dice: Diversity in deep ensembles via conditional redundancy adversarial estimation. In *ICLR*, 2021. 4

[28] Xiaolin Song, Sicheng Zhao, Jingyu Yang, Huanjing Yue, Pengfei Xu, Runbo Hu, and Hua Chai. Spatio-temporal contrastive domain adaptation for action recognition. In *CVPR*, 2021. 1, 2

[29] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019. 4

[30] Robert L Thorndike. Who belongs in the family? *Psychometrika*, 18(4):267–276, 1953. 4

[31] Yapeng Tian, Dingzeyu Li, and Chenliang Xu. Unified multisensory perception: weakly-supervised audio-visual video parsing. In *ECCV*, 2020. 1, 2

[32] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *ECCV*, 2018. 1, 2

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 3

[34] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 2

[35] Yu Wu and Yi Yang. Exploring heterogeneous clues for weakly-supervised audio-visual video parsing. In *CVPR*, 2021. 1, 2

[36] Yu Wu, Linchao Zhu, Yan Yan, and Yi Yang. Dual attention matching for audio-visual event localization. In *ICCV*, 2019. 1, 2

[37] Lijin Yang, Yifei Huang, Yusuke Sugano, and Yoichi Sato. EPIC-KITCHENS-100 unsupervised domain adaptation challenge for action recognition 2021: Team M3EM technical report. *arXiv preprint arXiv:2106.10026*, 2021. 1, 2

[38] Yunhua Zhang, Ling Shao, and Cees GM Snoek. Repetitive activity counting by sight and sound. In *CVPR*, 2021. 1, 2

[39] Linchao Zhu and Yi Yang. ActBERT: Learning global-local video-text representations. In *CVPR*, 2020. 4

[40] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76, 2020. 2

PoliTO-IIT-CINI Submission to the EPIC-KITCHENS-100 Unsupervised Domain Adaptation Challenge for Action Recognition

Mirco Planamente[1,2,3] Gabriele Goletto [1] Gabriele Trivigno [1] Giuseppe Averta [1] Barbara Caputo[1,3]

[1] Politecnico di Torino
name.surname@polito.it

[2] Istituto Italiano di Tecnologia
name.surname@iit.it

[2] Consortium Cini

Abstract

In this report, we describe the technical details of our submission to the EPIC-Kitchens-100 Unsupervised Domain Adaptation (UDA) Challenge in Action Recognition. To tackle the domain-shift which exists under the UDA setting, we first exploited a recent Domain Generalization (DG) technique, called Relative Norm Alignment (RNA). Secondly, we extended this approach to work on unlabelled target data, enabling a simpler adaptation of the model to the target distribution in an unsupervised fashion. To this purpose, we included in our framework UDA algorithms, such as multi-level adversarial alignment and attentive entropy. By analyzing the challenge setting, we notice the presence of a secondary concurrence shift in the data, which is usually called environmental bias. It is caused by the existence of different environments, i.e., kitchens. To deal with these two shifts (environmental and temporal), we extended our system to perform Multi-Source Multi-Target Domain Adaptation. Finally, we employed distinct models in our final proposal to leverage the potential of popular video architectures, and we introduced two more losses for the ensemble adaptation. Our submission (entry 'plnet') is visible on the leaderboard and ranked in 2nd position for 'verb', and in 3rd position for both 'noun' and 'action'.

1. Introduction

First person action recognition offers a wide range of opportunities and challenges, thanks to the use of wearable devices to capture the current state of the user and of the environment. Very often, indeed, the actions of the subject are captured through a video-camera placed on the head of the user. As a consequence, in contrast with most CV tasks, the major feature of this scenario is that source data are intrinsically characterized by rich multi-modal information, thanks to the proximity of the sensor to the action scene. As a result, sensor fusion between visual and auditory cues can be a powerful method to fully exploit the knowledge available in the data. However, the particular setup of data collection also comes with several difficulties: i) ego-motions represents a significant source of noise for the dataset, because changes in head posture cause a shift in the point-of-view and background. While from one side this effect can be exploited as an intrinsic attention mechanism, it may also introduce confusion between ego-motion and the real action of the subject. An approach to mitigate this effect could be to complement RGB data with other motion-related sources, such as the optical flow; ii) model predictions tend to be strongly correlated with the surrounding environment, which represents a bias in the dataset (usually referred to as *environmental bias*), thus resulting in decreased performances when the environment changes (e.g. different kitchens). In this report, we discuss the idea that, to fully exploit the potential of data sources, and to mitigate the performances drop across domains, it is crucial to properly combine several sensing modalities, including audio, video, and motion. This is particularly true for cross-domain scenarios, where test data are extracted from a different distribution w.r.t. the training data (i.e. different users and/or kitchens). Indeed, the effect of domain shift is not consistent across different sensing modalities, and some of them may suffer in some cases where others are more robust.

The reason is that domain shifts are not all of the same nature. For instance, the optical flow is more focused on the motion in the scene, rather than appearance, and is therefore less sensitive to environmental changes, thus showing higher robustness than the visual modality when changing environment [12]. On the other side, the domain shift of auditory information is very different from the visual one (e.g., the sound of 'cut' will differ from a plastic to a wooden cutting board). For all those reasons, the classifier should be able to assess - depending on the conditions - which modality is more informative, and therefore should be considered more for the final prediction.

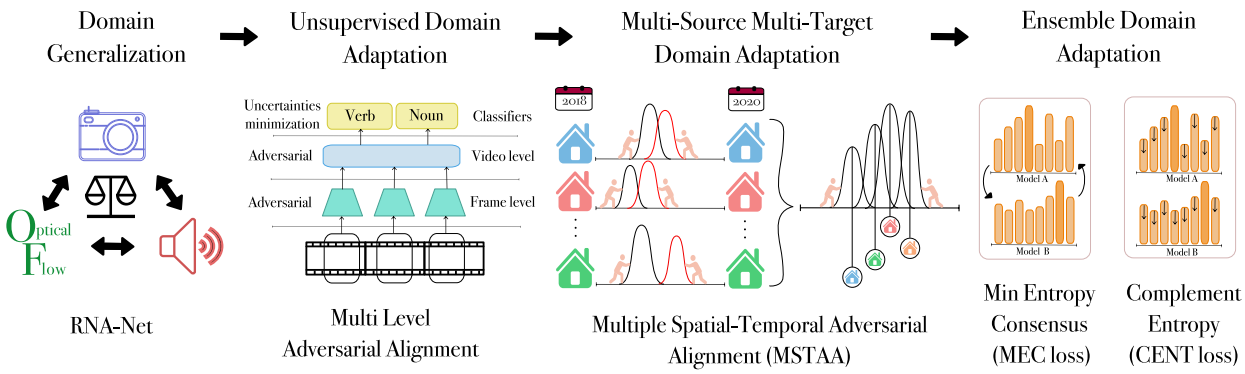


Figure 1: An overview of the proposed approach. It can be summarized in four main aspects: **1.** Domain Generalization through RNA-Net [15], **2.** Unsupervised Domain Adaptation via Multi-Level Adversarial Alignment and entropy minimization, **3.** Multi-Source Multi-Target Domain Adaptation extension and **4.** Ensemble Domain Adaptation losses.

To this purpose, authors of [15] recently proposed a multi-modal framework, called Relative Norm Alignment network (RNA-Net), which aims at progressively aligning the feature norms of audio and visual (RGB) modalities among multiple sources in a Domain Generalization (DG) setting, where target data are not available during training. Interestingly, the authors showed that *merely feeding all the source domains to the network without applying any adaptive techniques leads to sub-optimal performance, while a multi-source domain alignment allows the network to promote domain-agnostic features.*

Including all the aforementioned considerations, we developed the method adopted in the challenge with the following steps (see also Figure 1):

1. RNA-Net was extended to the Flow modality, obtaining remarkable results without accessing target data;
2. with further modifications, RNA-Net was adapted to work with unlabelled target data under the standard Unsupervised Domain Adaptation (UDA) setting;
3. the challenge's setting was revisited by identifying a new concurrent shift denominated "environmental bias". Our framework was modified accordingly to perform Multi-Source Multi-Target Domain Adaptation;
4. the final submission was obtained by combining different model streams by means of DA-based losses, namely Min-Entropy Consistency (MEC) and Complement Entropy (CENT).

2. Our Approach

In this section, we first describe the DG approach used. Then, we show our UDA framework and its extension for Multi-Source Multi-Target Domain Adaptation. Finally, we demonstrate how to re-define existing DA-based losses to induce consistency between different architectures.

2.1. Domain Generalization

The multi-source nature of the proposed challenge setting makes it perfect to deal with the domain shift using DG techniques. Thus, we first exploited a method which has been recently proposed to operate in this context, called Relative Norm Alignment (RNA) [15]. This methods consists of an *audio-visual domain alignment* at feature-level through the minimization of a cross-modal loss function ($\mathcal{L}_{RNA}$). The latter aims at minimizing the *mean-feature-norm distance* between the audio and visual features norms among all the source domains, and it is defined as

$$\mathcal{L}_{RNA} = \left(\frac{\mathbb{E}[h(X^v)]}{\mathbb{E}[h(X^a)]} - 1 \right)^2, \tag{1}$$

where $h(x_i^m) = (\|\cdot\|_2 \circ f^m)(x_i^m)$ indicates the L_2-norm of the features f^m of the m-th modality, $\mathbb{E}[h(X^m)] = \frac{1}{N} \sum_{x_i^m \in \mathcal{X}^m} h(x_i^m)$ for the m-th modality and N denotes the number of samples of the set $\mathcal{X}^m = \{x_1^m, ..., x_N^m\}$.

Authors of [15] proved that the norm unbalance between different modalities might cause the model to be biased towards the source domain that generate features with greater norm, thus causing wrong predictions. Contrarily, by simultaneously solving the problem of classification and relative norm alignment on different domains, the network extracts a shared knowledge between the different sources, resulting in a domain-agnostic model.

In our submission to the EPIC-Kitchen UDA challenge, we extended the RNA-Net framework to the optical flow modality, in order to exploit the multiple sources available from the official training splits while showing the effectiveness of RNA loss in a multi-source DG setting.

2.2. Domain Adaptation

The UDA techniques embedded into our pipeline can be divided in two main groups: *feature*-level and *classifier*-level. The first aims at aligning the distribution of source and target, and works at different levels of representation (frames- and video-level); the latter, instead, reduces the classifier's uncertainty on target data.

Multi-Level Adversarial Alignment.

Following popular practices in unsupervised video domain adaption techniques, we integrate into our framework an adversarial approach [3, 12], consisting of an extension of the DANN [8] standard UDA image-based method. We apply it at two different feature levels; frame- and video-level. It entails the introduction of two separate branches in our framework. Down-stream of said branches there are discriminators that try to distinguish the two domains (source and target). Contrarily, by maximising the corresponding discriminator losses, the network learns feature representations invariant to both domains.

Attentive Entropy. In order to reduce the uncertainty of the classifier on the target data, we minimize the attentive entropy loss proposed in [3] as in [17]. This action minimizes the entropy, resulting in a refinement of the classifier adaptation. The term "attentive" refers to a loss reweighting approach that prioritizes videos with low domain discrepancy by focusing on minimizing entropy for these videos.

2.3. Multi-Source Multi-Target Domain Adaptation

The previous Epic Kitchen challenges [6, 5], as well as the literature on unsupervised domain adaptation for first person action recognition [13, 15, 14, 18, 16], reveal a strong dependency of the models on the environment where the actions are recorded. This problem, known as "environmental bias", causes a decrease in performance in occurrence of environment switches. As regards past action recognition challenges, we see this behavior by comparing performances of the models when tested on S1 (seen) and S2 (unseen). In the setting proposed in [13], similar behavior is observed, demonstrating the model's low generalization ability when tested on different kitchens.

The above considerations allow us to identify a secondary shift in this challenge, that occurs along with the temporal shift. Indeed, the training data are collected from different environments i.e. kitchens, thus introducing an environmental shift. As a result, we may rename the challenge setting *Multi-Source Multi-Target Unsupervised Domain Adaptation*.

To deal with this new setting we propose a novel framework, which we call Multiple Spatio-Temporal Adversarial Alignment (MSTAA), combining Multiple Temporal Adversarial Alignment (MTAA) and Multiple Spatial Adversarial Alignment (MSAA). MTAA is obtained by adopting 2K domain adversarial branches (where K indicates the number of kitchens), aligning the source and the target distribution both at video- and frame-level for each kitchen. Instead, MSAA consists in adding another adversarial branch with a k-dimension discriminator in order to align the distribution of different kitchens and alleviate the environmental bias issue.

2.4. Ensemble UDA losses

For our final submission different models have been used in order to fully exploit the potentiality of popular video architectures. However, training individually each backbone with standard UDA protocols would result in independently adapted feature representations, which consequently vary between different streams. Our intuition is that this aspect could impact negatively the training process and the performance on target data. Indeed, since the domain adaption process acts on each architecture independently, naively training the backbones separately would yield mismatching prediction logits on target data, which, when combined, could increase the level of uncertainty of the model. For this reason, we use the Min Entropy Consensus (MEC) loss, to impose a consistency constraint between feature representations from various models. Then, re-purposing the existing Complement Entropy (CENT) loss, we attempt to exploit the target data samples based on the assumption that there are some conditions in which it is easier to answer the question *"Which classes does this action not belong to?"* rather than *"Which class does this action belong to?"*.

Min Entropy Consensus (MEC loss). We extended the loss proposed in [19] to encourage coherent predictions between different models. The resulting loss is defined as:

$$\mathcal{L}_{MEC} = -\frac{1}{m}\sum_{i=1}^{m}\frac{1}{b}\max_{y\in\mathcal{Y}}\sum_{b}\log p_b(y|x_i^t) \qquad (2)$$

where m is the cardinality of the batch size of the target set, y is the predicted class, and $\log p_b(y|x_i^t)$ is the prediction probability of the b-th backbone network. The intuitive idea behind the proposed approach is to encourage different backbones to have a similar predictions.

Complement Entropy (CENT). The Complement Entropy (CENT) loss aims at neutralizing the negative effects on the final prediction of clips whose logits present high degrees of uncertainty. It accomplishes this by "flattening" the predicted probabilities of "complement classes", i.e., all classes except the predicted one. As a result, when predictions are ensembled, the noise due to uncertainty on complement classes is reduced. We refer to this loss as "complement entropy" objective, as it consists in maximizing the entropy for low-confident classes rather than minimizing it for the most confident one, as standard entropy minimization does. It is defined as:

Unsupervised Domain Adaptation Leaderboard							
	Rank	Verb Top-1	Noun Top-1	Action Top-1	Verb Top-5	Noun Top-5	Action Top-5
VI-I2R	1	**57.89**	40.07	**30.12**	**83.48**	64.19	**48.10**
Audio-Adaptive-CVPR2022	2	52.95	**42.26**	28.06	80.03	**67.51**	44.03
plnet	3	55.51	35.86	25.25	82.77	60.65	40.09
CVPR2021-chengyi	4	53.16	34.86	25.00	80.74	59.30	40.75
CVPR2021-M3EM	5	53.29	35.64	24.76	81.64	59.89	40.73
CVPR2021-plnet	6	55.22	34.83	24.71	81.93	60.48	41.41
EPIC_TA3N [4]	8	46.91	27.69	18.95	72.70	50.72	30.53
EPIC_TA3N_SOURCE_ONLY [4]	9	44.39	25.30	16.79	69.69	48.40	29.06

Table 1: Leaderboard results of EPIC-Kitchens Unsupervised Domain Adaptation Challenge. The results obtained by the top-3 participants and the provided baseline methods are reported. **Bold:** highest result Underline: second highest result; Green: our final submission.

Unsupervised Domain Adaptation			
	Verb	Noun	Action
Ensemble (E) *Source Only*	53.64	32.65	22.98
E-UDA	53.88	33.10	23.22
E+MEC	53.67	34.32	23.91
E+MEC+CENT	54.20	33.92	23.99
E-SMR+MEC+CENT	**54.55**	34.72	24.22
E-SMR+MEC+CENT+MTAA	54.09	33.72	23.77
E-SMR+MEC+CENT+MSTAA	54.01	**34.82**	**24.24**

Table 2: Results on the EPIC-Kitchen validation set.

Domain Generalization			
	Target	Verb Top-1	Verb Top-5
Source Only	✗	44.39	69.69
EPIC_TA3N [4]	✓	46.91	72.70
RNA-Net [15]	✗	47.96	79.54
EPIC_TA3N+RNA-Net	✓	**50.40**	**80.47**

Table 3: Results on the EPIC-Kitchen test set.

$$\mathcal{L}_{CENT} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{H}(\hat{y}_{i\bar{c}}) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1,j\neq p}^{\mathcal{C}}\left(\frac{\hat{y}_{ij}}{1-\hat{y}_{ip}}\log\frac{\hat{y}_{ij}}{1-\hat{y}_{ip}}\right) \quad (3)$$

where N is the total number of samples in the batch, $\hat{y}_{ip}$ represents the predicted probability of the class p with the higher score for the i-th sample, i.e., $\hat{y}_{ip} = max_j(\hat{y}_{ij})$, and $\mathcal{H}(\cdot)$ is the entropy function computed on the prediction of complement classes $\hat{y}_{i\bar{c}}$ ($\bar{c} \neq p$). The formulation is similar to the one in [2], and we extend it to operate in an unsupervised fashion.

3. Framework

In this section, we describe the architectures of the feature extractors used to produce suitable multi-modal video embeddings, and the fusion stategies adopted to combine them. Finally, we deepen the analysis describing the hyper-parameters used for the training.

3.1. Architecture

Backbone. For our submission, we adopted three different network configurations. In the first one, corresponding to the RNA-Net framework in [15], we used the Inflated 3D ConvNet (I3D), pre-trained on Kinetics [1], for RGB and Flow streams, and a BN-Inception model [10] pre-trained on ImageNet [7] for the auditory information. Each feature extractor produces a 1024-dimensional representation which is fed to an action classifier. In the second configuration, we used BN-Inception models for all the three streams, using pre-extracted features from a TBN [12] model trained on EPIC-Kitchens-55. In the last configurations, we used standard ResNet-50 architectures [9] equipped with the Temporal Shift Module [11] pre-trained on EPIC-Kitchens-55 [1].

Multi-modal fusion strategies. In all the above mentioned configurations, each modality is processed by its own backbone, and the corresponding extracted representations are then fused following different strategies. For RNA-Net, we followed a standard late fusion strategy, consisting in averaging the final score predictions obtained from two different fully-connected layers (verb, noun) from each modality. In the other configurations, we adopted the recent mid-fusion strategy, called Semantic Mutual Refinement submodule (SMR), proposed in [20], to generate a common frame-embedding among the modalities. Then, using tem-

[1] https://github.com/epic-kitchens/epic-kitchens-55-action-models

λ_{RNA}	λ_{CENT}	λ_{MEC}	γ	β
1	0.31	0.22	0.003	0.75, 0.75, 0.75

Table 4: UDA losses hyper-parameters used during training.

poral pooling, we obtain a final video-embedding that is sent to the verb and noun classifiers.

3.2. Implementation Details

We trained I3D and BN-Inception models with SGD optimizer, with an initial learning rate of 0.001, dropout 0.7, and using a batch size of 128, following [15]. Instead, when using pre-extracted features from ResNet50 or BN-Inception, we trained the SMR modules on top of them for 45 epochs with an initial learning rate of 0.03, decayed after epochs 25 and 35 by a factor of 0.1. We used a batch size of 128 with SGD optimizer. In Table 4 we report the other hyper-parameter used. Specifically, we indicate with λ_{RNA}, λ_{CENT} and λ_{MEC} the weights of RNA, CENT and MEC losses respectively. In addition, we report the values used to weight the attentive entropy loss (γ) and the domain losses at different levels (β) for MSTAA.

4. Results and Discussion

In Table 1 we report our best performing model on the target test, achieving the **2st** position on 'verb', and the **3rd** on 'noun' and 'action'. Meanwhile, in Tables 2 and 3 we show an ablation of the proposed UDA and DG methods described in section 2.

How well do DG approaches perform? The results in Table 3 are obtained under the multi-source DG setting, when target data are not available during training. Noticeably, RNA outperforms the baseline Source Only by up to 3% on Top-1 and 10% on Top-5, highlighting the importance of using ad-hoc alignment techniques to deal with multiple sources in order to effectively extract a domain-agnostic model. Moreover, it outperforms the recent UDA technique TA^{3}N [3] without accessing target data. Interestingly, when combined with EPIC_TA3N, it further improves performance, proving the complementarity of RNA to other existing UDA approaches.

In Table 2 it can be seen how the proposed UDA approaches improve Top-1 accuracy on all categories by up to 1%. Although using an additional adversarial branch for each kitchen does not appear to provide a significant improvement on the validation set, it increases the top-1 action accuracy on the test set, allowing us to obtain the third position in the challenge. Without MSTAA, the accuracy on the action top-1 reaches just 24.83%. This outcome was predictable given that the validation set is populated with a different set of kitchens than the test set, whereas the kitchens in the test set are the same as those used for the target and source training. This aspect confirms the *Multi-Source Multi-Target Unsupervised Domain Adaptation* setting and the presence of two different shifts, the *temporal* shift (2018-2020) and the *environmental* shift (among the kitchens).

References

[1] 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*, pages 6299–6308, 2017.

[2] Hao-Yun Chen, Pei-Hsin Wang, Chun-Hao Liu, Shih-Chieh Chang, Jia-Yu Pan, Yu-Ting Chen, Wei Wei, and Da-Cheng Juan. Complement objective training. *arXiv preprint arXiv:1903.01182*, 2019.

[3] 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 International Conference on Computer Vision*, pages 6321–6330, 2019.

[4] 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.

[5] Dima Damen, Evangelos Kazakos, Will Price, Jian Ma, and Hazel Doughty. Epic-kitchens-55 - 2020 challenges report. https://epic-kitchens.github.io/Reports/EPIC-KITCHENS-Challenges-2020-Report.pdf, 2020.

[6] Dima Damen, Will Price, Evangelos Kazakos, Antonino Furnari, and Giovanni Maria Farinella. Epic-kitchens - 2019 challenges report. https://epic-kitchens.github.io/Reports/EPIC-Kitchens-Challenges-2019-Report.pdf, 2019.

[7] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[8] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. volume 37 of *Proceedings of Machine Learning Research*, pages 1180–1189, Lille, France, 07–09 Jul 2015. PMLR.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456. PMLR, 2015.

[11] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings*

of the IEEE International Conference on Computer Vision, pages 7083–7093, 2019.

[12] 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*, pages 122–132, 2020.

[13] 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 (CVPR)*, June 2020.

[14] Mirco Planamente, Andrea Bottino, and Barbara Caputo. Self-supervised joint encoding of motion and appearance for first person action recognition. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8751–8758. IEEE, 2021.

[15] Mirco Planamente, Chiara Plizzari, Emanuele Alberti, and Barbara Caputo. Domain generalization through audio-visual relative norm alignment in first person action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1807–1818, 2022.

[16] Mirco Plananamente, Chiara Plizzari, and Barbara Caputo. Test-time adaptation for egocentric action recognition. In *International Conference on Image Analysis and Processing*, pages 206–218. Springer, 2022.

[17] Chiara Plizzari, Mirco Planamente, Emanuele Alberti, and Barbara Caputo. Polito-iit submission to the epic-kitchens-100 unsupervised domain adaptation challenge for action recognition. *arXiv preprint arXiv:2107.00337*, 2021.

[18] Chiara Plizzari, Mirco Planamente, Gabriele Goletto, Marco Cannici, Emanuele Gusso, Matteo Matteucci, and Barbara Caputo. Eˆ 2 (go) motion: Motion augmented event stream for egocentric action recognition. *arXiv preprint arXiv:2112.03596*, 2021.

[19] Subhankar Roy, Aliaksandr Siarohin, Enver Sangineto, Samuel Rota Bulo, Nicu Sebe, and Elisa Ricci. Unsupervised domain adaptation using feature-whitening and consensus loss. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9471–9480, 2019.

[20] Lijin Yang, Yifei Huang, Yusuke Sugano, and Yoichi Sato. Epic-kitchens-100 unsupervised domain adaptation challenge for action recognition 2021: Team m3em technical report. *arXiv preprint arXiv:2106.10026*, 2021.

EPIC-KITCHENS-100 Unsupervised Domain Adaptation Challenge for Action Recognition 2022: Team HNU-FPV Technical Report

Nie Lin, Minjie Cai
College of Computer Science and Electronic Engineering, Hunan University
Changsha, China
{nielin,caiminjie}@hnu.edu.cn

Abstract

In this report, we present the technical details of our submission to the 2022 EPIC-Kitchens Unsupervised Domain Adaptation (UDA) Challenge. Existing UDA methods align the global features extracted from the whole video clips across the source and target domains but suffer from the spatial redundancy of feature matching in video recognition. Motivated by the observation that in most cases a small image region in each video frame can be informative enough for the action recognition task, we propose to exploit informative image regions to perform efficient domain alignment. Specifically, we first use lightweight CNNs to extract the global information of the input two-stream video frames and select the informative image patches by a differentiable interpolation-based selection strategy. Then the global information from videos frames and local information from image patches are processed by an existing video adaptation method, i.e., TA3N, in order to perform feature alignment for the source domain and the target domain. Our method (without model ensemble) ranks 4th among this year's teams on the test set of EPIC-KITCHENS-100.

1. Introduction

With the rapid development of deep learning techniques, how to develop deep neural networks to understand human's daily interactions with surrounding environments from the first-person perspective has gained increasing interests from researchers. The EPIC-KITCHENS-100 dataset is a large video dataset of first-person perspective, and the videos record most of the common actions that would happen in a kitchen scene [2]. The dataset provides fine-grained action labels, and each action is composed by a pair of verb and noun labels. In order to meet the task of EPIC-KITCHENS-100 Unsupervised Domain Adaptation (UDA) Challenge for Action Recognition, the model needs to be trained on the labeled source domain (EPIC-KITCHENS-2018) and adapted to the unlabeled target domain (EPIC-KITCHENS-100). The UDA for action recognition is more challenging than the action recognition task since the adapted model needs to overcome the domain discrepancy represented in complex video features between the source domain and the target domain. Therefore, how to effectively model the shared feature representation of the source and target domains is one of the keys to solve this challenge.

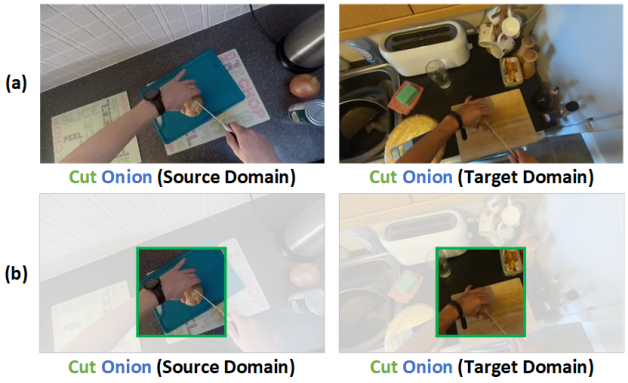


Figure 1. Illustration of fine-grained action recognition on EPIC-KITCHENS-55 (source domain) and EPIC-KITCHENS-100 (target domain). (a) Due to the differences in shooting time and indoor environment, there are many different background objects in the video clip of the same action (*e.g.*, "cutting onion") between the source/target domains, which are irrelevant to the action recognition task. (b) By selecting the most informative image regions for processing, the domain discrepancy between the source domain and the target domain can be effectively reduced.

As recorded by a wearable camera from the first-person perspective, egocentric video is characterized by rapidly changing background between consecutive actions and cluttered background containing multiple objects irrelevant to the ongoing action. Furthermore, for videos in different domains, the same actions may present huge differences of image appearance, especially in the background. As a

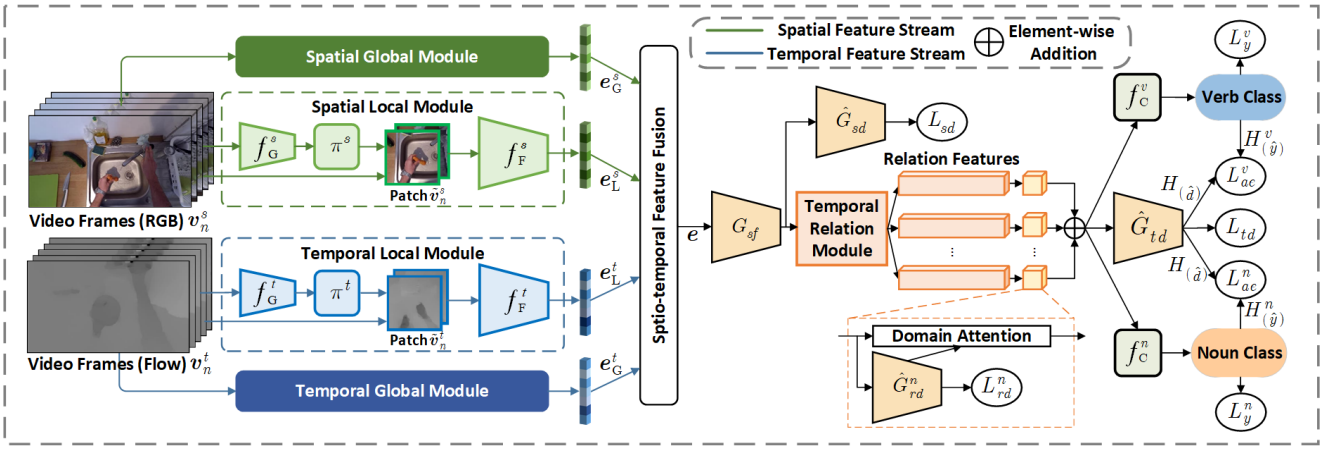


Figure 2. Overview of the proposed method. The method includes two main parts: spatio-temporal feature extraction and video domain adaptation. In spatio-temporal feature extraction, it is composed by global feature extraction branches and local feature extraction branches for both RGB and optical flow inputs. f^s_{G}, f^s_{F} and π^s denote the glancer, focuser and policy networks for the spatial local module, respectively. Similar notations are used for the temporal local module. In video domain adaptation, $\hat{G}_{sd}$, $\hat{G}_{td}$ and $\hat{G}^n_{rd}$ denote the spatial, temporal and relation domain classifiers, respectively. L_{sd}, L_{td} and L^n_{rd} denote the spatial, temporal and relation domain classification loss. f^v_{C} and f^n_{C} denote the verb classifier and noun classifier. L^v_y and L^n_y denote the verb and noun classification loss, respectively. L^v_{ae} and L^n_{ae} denote the attentive entropy loss for verb and noun, respectively.

result, directly modeling shared feature representation between different domains is challenging due to spatial redundancy in the original video features. Figure 1 shows examples of video frames of the same action from two different domains. It can be seen that the action of "cutting onion" in the source domain shows quite different visual appearance compared with the target domain. One exception is the region around hands which show certain consistency between two domains. Actually, information of the verb "cutting" and the noun "onion" is fully encoded in such informative regions of video frames. So the challenge of action recognition in UDA lies in the frequent scene switching between each action and the difference in the background of the same action in different domains. Therefore, instead of straightforward domain alignment of original video features, exploiting the most informative regions in video frames for feature extraction shows a promising way of efficient domain adaptation for egocentric action recognition.

In this work, we incorporate a learning-based patch selection strategy into an existing video domain adaption framework. The patch selection strategy is implemented as a lightweight CNN and a policy network which helps locate the task-related regions and extract local features for each video frame. We consider both RGB and optical flow images as input to capture the spatial and temporal characteristic of an action. After spatial-temporal feature fusion with both global and local features, we adopt an existing video domain adaptation method TA3N [1] to do feature alignment for the source and target domains. The experimental results on EPIC-KITCHEN-100 demonstrate the effectiveness of the proposed method in UDA for action recognition.

2. Method

As an overview of our approach is described in Figure 2, the overall model is divided into two parts. The first part of the model extracts the spatio-temporal features of the video from the input RGB frames and optical flow frames and contains both global and local branches in the process. For the local branch, inspired by the latest work in video-based action recognition [6, 7], we build a spatio-temporal local feature extraction. After extracting the global and local features of the original video, the model will fuse the spatio-temporal features extracted from different domains through spatio-temporal feature fusion. In the second part, the model is used to align the spatio-temporal features extracted from the source domain and the target domain and finally complete the action prediction of the target domain. We will introduce the above component in detail in the following sections.

2.1. The Spatio-temporal Feature Extraction

Given a RGB stream of video frames $\{\boldsymbol{v}^s_1, \boldsymbol{v}^s_2, ...\}$ and a optical flow stream of video frames $\{\boldsymbol{v}^t_1, \boldsymbol{v}^t_2, ...\}$, the model will extract the spatio-temporal features of the two different video streams. For the local feature extraction, the model takes a glance at each frame in the video with the corresponding glancer network f_{G}. Then the cheap and coarse feature will be fed into the corresponding policy network π to select the area that contributes the most to the task:

$$\begin{aligned} \tilde{\boldsymbol{v}}^s_n &= \pi^s(f^s_{\mathrm{G}}(\boldsymbol{v}^s_n)), \quad n = 1, 2, ..., \\ \tilde{\boldsymbol{v}}^t_n &= \pi^t(f^t_{\mathrm{G}}(\boldsymbol{v}^t_n)), \quad n = 1, 2, ..., \end{aligned} \tag{1}$$

Table 1. The recognition performance of different models on target validation set. FeatDim: the dimension of shared features of TA3N; NumSeg: the number of input frames between the global and local branches is from left to right. The left and right side of "+" indicates the input into the glancer network and the focuser network. "-" indicates that local branches are not used for feature extraction.

Method	Backbone		FeatDim	NumSeg	Top-1 Accuracy (%)			Top-5 Accuracy (%)		
	Global	Local			Verb	Noun	Action	Verb	Noun	Action
TA3N	TBN	-	512	6 / -	48.10	26.74	18.72	77.98	47.50	41.87
TA3N	TBN	-	1024	6 / -	48.28	27.30	19.25	76.71	47.39	41.65
TA3N	TBN	MN2/RN	1024	6 / 4+6	48.70	27.87	19.61	76.18	48.52	42.01
TA3N	TBN	MN2/RN	2048	12 / 8+12	49.42	28.33	20.11	77.06	47.52	41.82

Table 2. The recognition performance of different models on target test set. All results on the test set were evaluated on the test server. Table column definitions are the same as in Table 1.

Method	Backbone		FeatDim	NumSeg	Top-1 Accuracy (%)			Top-5 Accuracy (%)		
	Global	Local			Verb	Noun	Action	Verb	Noun	Action
TA3N	TBN	MN2/RN	1024	6 / 4+6	47.71	27.74	19.41	73.38	48.91	31.26
TA3N	TBN	MN2/RN	2048	12 / 8+12	48.87	28.72	19.88	74.61	49.70	32.32

where $\tilde{\boldsymbol{v}}_n^s$, $\tilde{\boldsymbol{v}}_n^t$ are the selected patch of RGB video frames and optical flow video frames of the n^{th} frame. And the selected patch $\tilde{\boldsymbol{v}}_n^s$, $\tilde{\boldsymbol{v}}_n^t$ will be fed into the corresponding focuser network f_{F} to extract the local feature maps $\boldsymbol{e}_{\mathrm{L}}^s$, $\boldsymbol{e}_{\mathrm{L}}^t$:

$$\begin{aligned} \boldsymbol{e}_{\mathrm{L}}^s &= f_{\mathrm{F}}^s(\tilde{\boldsymbol{v}}_n^s), \quad n = 1, 2, ..., \\ \boldsymbol{e}_{\mathrm{L}}^t &= f_{\mathrm{F}}^t(\tilde{\boldsymbol{v}}_n^t), \quad n = 1, 2, ..., \end{aligned} \tag{2}$$

Finally, the global spatio-temporal features $\boldsymbol{e}_{\mathrm{G}}^s$, $\boldsymbol{e}_{\mathrm{G}}^t$ and audio feature $\boldsymbol{e}_{\mathrm{G}}^a$ extracted from the global branch. Note that our model considers the global features corresponding to the audio modalities, which are not represented in the figure for the sake of simplicity. Then the global features are concatenated with the local spatio-temporal features $\boldsymbol{e}_{\mathrm{L}}^s$, $\boldsymbol{e}_{\mathrm{L}}^t$ extracted from the local branch are concatenated to serve as the input final feature $\boldsymbol{e}$ to the video domain adaptation:

$$\boldsymbol{e} = Concat(\boldsymbol{e}_{\mathrm{G}}^s, \boldsymbol{e}_{\mathrm{L}}^s, \boldsymbol{e}_{\mathrm{G}}^t, \boldsymbol{e}_{\mathrm{L}}^t, \boldsymbol{e}_{\mathrm{G}}^a). \tag{3}$$

2.2. The Video Domain Adaptation

After the global and local spatio-temporal features are obtained, it will be more beneficial for the model to perform efficient domain alignment. In the video domain-adapted training of global-local features from the source and target domains, we adapt an existing video domain adaptation method for action recognition tasks, *i.e.*, TA3N [1]. As shown in Figure 2, model first aligns frame-level features from the source and target domain inputs through the adversarial discriminators $\hat{G}_{sd}$ and generates the corresponding domain loss L_{sd}. At the same time, the frame-level features of the input are modeled in the temporal relation module of TA3N, and these relation features are aggregated to obtain the video-level features. In aggregating these relational features, the domain attention mechanism is added to pay more attention to the alignment of local temporal features that have larger domain discrepancy. In the domain attention mechanism, the adversarial discriminators $\hat{G}_{rd}^n$ are used to align the relational features from the source and target domains, and the corresponding domain loss L_{rd}^n is generated. Then, the adversarial discriminators $\hat{G}_{td}$ are also used to align the video-level features from the source and target domains, and the corresponding domain loss L_{td} is generated. Finally, the model classifies the video-level features through two corresponding classifiers f_{C}^v and f_{C}^n, and generates the predicted verb classification and noun classification.

3. Experiments

3.1. Implementation Details

Spatio-temporal feature extraction. Since the network of spatial feature extraction and temporal feature extraction are the same in parameter settings, the following description will not distinguish between spatial and temporal feature extraction. For the global feature of spatio-temporal feature extraction, we use RGB, flow and audio features provided by the organizers that were extracted with Temporal Binding Network (TBN) [4] pretrained in the source domain. And we follow the model setting in [7] to extract the local feature of the spatio-temporal feature. We also adopt MobileNet-V2 (MN2) [5] and ResNet-50 (RN) [3] as the glancer network f_{G} and focuser network f_{F}, respectively. And the same policy network is used to select the image patch that contributes most to the task from the input video frames by the differentiable bilinear interpolation. The network parameters are learned with SGD optimizer with momentum 0.9 and weight decay 5×10^{-4}. For each network of the local branches, the initial learning rates of f_{G}, f_{F} and π are set to 0.005, 0.01, and $1e-4$, respectively. For the video frames that are input into the model, we adopt the

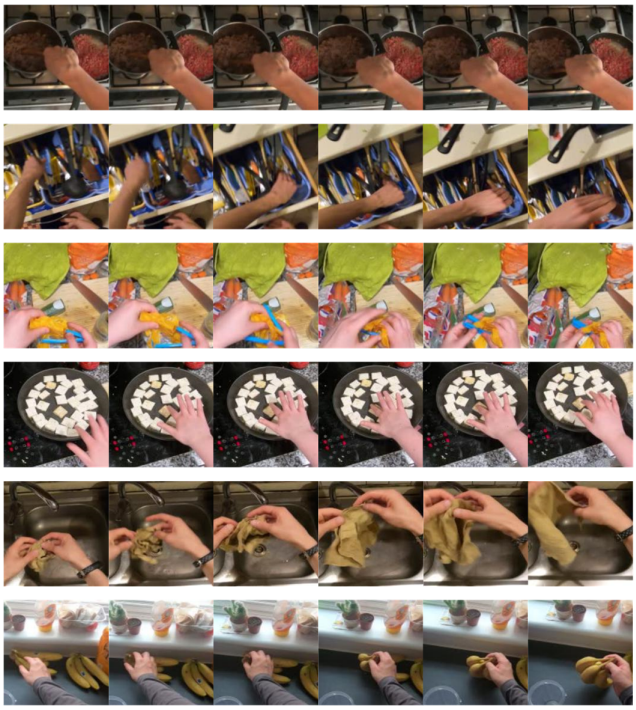


Figure 3. Visualization results of the image patches selected from the spatial local module.

same processing method as [7] and set the size of the selected image patch to 176×176.

Video domain adaptation. After obtaining the spatio-temporal features of the source and target domains, TA3N [1] is used to align the input features and generate the prediction results of the model. The network parameters are also learned with SGD optimizer with momentum 0.9 and weight decay 5×10^{-4}. During training, the parameters in the spatio-temporal feature extraction are freezed. The initial learning rate is set at 3e-3 and decayed by a factor of 0.1 at epochs 10 and 20.

3.2. Result

Table 1 shows the action recognition effect of the model on the target validation set under different input and hyper-parameter settings. The table shows that the accuracy can be improved under the same hyperparameter setting by using the local spatio-temporal branch to extract the local feature. We tried two groups of models trained under different hyper-parameters, and their performance on the target test set is shown in Table 2. Our proposed method performs favorably against TA3N by 0.93% in the top-1 action accuracy. In our final submission, we use RGB, Flow and Audio modalities, and the shared feature dimension of the model is set as 2048. The number of input frames of the glancer network and focuser network is set as 8 and 12, respectively.

The visualization results of the image patches selected from the test set by the proposed method are shown in Figure 3. Each line shows a number of image patches selected from consecutive video frames by the spatial local module of the method. It should be noticed that the spatial local module is fixed after training with source domain data. It can be seen that the model can also be well applied to the videos of the target domain.

4. Conclusion

This paper presents the technical details of our solution for the EPIC-KITCHENS-100 UDA for Action Recognition Challenge. By incorporating a learning-based patch selection strategy into an existing video domain adaption framework, the proposed method can effectively improve the domain adaptation performance of action recognition. Our work empirically verifies the importance of exploiting informative regions for egocentric videos and provides some new inspirations for domain adaptive action recognition.

References

[1] 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*, pages 6321–6330, 2019. 2, 3, 4

[2] 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

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 3

[4] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5492–5501, 2019. 3

[5] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 3

[6] Yulin Wang, Zhaoxi Chen, Haojun Jiang, Shiji Song, Yizeng Han, and Gao Huang. Adaptive focus for efficient video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16249–16258, 2021. 2

[7] Yulin Wang, Yang Yue, Yuanze Lin, Haojun Jiang, Zihang Lai, Victor Kulikov, Nikita Orlov, Humphrey Shi, and Gao Huang. Adafocus v2: End-to-end training of spatial dynamic networks for video recognition. *arXiv preprint arXiv:2112.14238*, 2021. 2, 3, 4

UniUD-FBK-UB-UniBZ Submission to the EPIC-Kitchens-100 Multi-Instance Retrieval Challenge 2022

Alex Falcon
Fondazione Bruno Kessler and University of Udine
afalcon@fbk.eu

Giuseppe Serra
University of Udine
giuseppe.serra@uniud.it

Sergio Escalera
University of Barcelona and Computer Vision Center
sergio@maia.ub.es

Oswald Lanz
Free University of Bozen-Bolzano
lanz@inf.unibz.it

Abstract

This report presents the technical details of our submission to the EPIC-Kitchens-100 Multi-Instance Retrieval Challenge 2022. To participate in the challenge, we designed an ensemble consisting of different models trained with two recently developed relevance-augmented versions of the widely used triplet loss. Our submission, visible on the public leaderboard, obtains an average score of 61.02% nDCG and 49.77% mAP.

1. Introduction

Retrieving the most relevant videos based on a user query is a difficult task involving joint visual and textual understanding. The EPIC-Kitchens-100 dataset [2] offers a challenging benchmark, comprising more than 70k egocentric video clips capturing activities from 45 kitchens. Differently from standard benchmarks in text-video retrieval, the EPIC-Kitchens-100 Multi-Instance Retrieval Challenge uses rank-aware metrics, such as the nDCG and the mAP, to assess the quality of the solutions. This is made possible by the introduction of a relevance function [2] which is defined in terms of the noun and verb classes found within the captions.

To participate in the challenge, we designed an ensemble of multiple models trained with two relevance-augmented versions [4, 5] of the standard triplet loss [7]. The final results show the effectiveness of the training techniques we recently proposed, as well as the ensemble version. In particular, when compared to the public leaderboard from last year, we observe improvements of almost 8% in nDCG and 5% in mAP. Moreover, when compared to the current public leaderboard, we obtain the best result in terms of mAP with a margin of more than 2%, and the second best result in nDCG with only 0.4% difference.

In Section 2 we provide details about the two optimization strategies [4, 5] which we recently developed. In Section 3 we describe the two architectures [1,9] which we used as the basis of our study. Implementation details and a brief overview describing how we ensemble the different models are provided in Section 4. Finally, we conclude the report in Section 5.

2. Optimization strategies

We describe the details concerning two different optimization strategies which use the relevance function introduced in [2] to improve the contrastive loss functions commonly used to learn text-video retrieval models.

2.1. Relevance-Margin

To train a text-video retrieval model with the triplet loss function, the same fixed margin is enforced on the similarity between the anchor-positive pair and the anchor-negative pair. This strategy makes it possible to maximize the similarity of the descriptors of the video and caption pairs in the dataset. Yet, the negative examples may have different relevance values when compared to the anchor, and in particular they may be even partially relevant. Therefore, in [5] we proposed to replace the fixed margin with a relevance-based margin, that is a margin which is proportional to the relevance value of the video and caption descriptors which are to be contrasted. Given the anchor a, the positive p, and the negative n, it is defined as follows:

$$\Delta_{a,p,n} = 1 - \mathcal{R}(a, n) \tag{1}$$

2.2. RANP

Due to the sampling mechanisms used to form the triplets, all the negative examples are treated as equally irrelevant when compared to the anchor. Yet, as mentioned

before, not all the negatives are actually irrelevant, and therefore those which are not completely irrelevant should not be treated as if they were. In [4] we proposed RANP, a strategy which uses the relevance function and a threshold τ to separate relevant from irrelevant samples (up to a degree τ) within the batch. By doing so, the negatives can be picked from a smaller negatives' pool which only contains irrelevant samples. Similarly, we introduced an additional triplet loss term which increases the similarity of the anchor with dissimilar yet relevant samples in the current network state during training.

3. Models

We briefly describe here the two network architectures which we used as the base models.

3.1. JPoSE

To have a fine-grained understanding of the actions in a retrieval setting, Wray et al. [9] introduced JPoSE, which disentangles the Part-of-Speech (PoS) in the captions in order to learn a multi-modal embedding space for each PoS tag. The PoS-restricted embeddings are then used to perform action retrieval in a joint embedding space. All the embedding spaces are finally learned by using a mix of PoS-restricted and PoS-agnostic losses.

3.2. HGR

Chen et al. [1] propose to deal with fine-grained retrieval by means of hierarchical structures and graph reasoning. First of all, for each natural language description they build a graph of the semantic roles occurring between each noun and the associated verb phrase [8]. A global-to-local graph is then built by using these textual features as the nodes, which are then aggregated through graph message passing and aligned to the visual features with a bidirectional global loss term.

4. Experiments

In this section, we detail the experimental settings of the models considered within the ensemble and the description of the ensembling strategy.

4.1. Implementation details

Details of the models. We briefly point out for each model some technical aspects related to the implementation details.

- *Model 1.* JPoSE trained with the relevance-margin.
- *Model 2.* HGR trained with RANP, using $\tau = 0.15$, $\Delta_p = 0.2$ (see [4] for more details about the margin Δ_p).
- *Model 3.* HGR trained with RANP, using $\tau = 0.15$, $\Delta_p = 0.2$ with a lower size for the embedding space (512).
- *Model 4.* HGR trained with RANP, using $\tau = 0.4$, $\Delta_p = 0.25$.
- *Model 5.* HGR trained with RANP, using $\tau = 0.4$, $\Delta_p = 0.15$

Training details. We trained JPoSE by using the relevance-based margin within each of the triplet loss terms used in the method, including both cross-modal and within-modality losses, both at the global and at the PoS level. When dealing with the losses at the noun (respectively, verb) level, we set the verb IoU (respectively, noun IoU) to 1 during the computation of the relevance. The optimizer used is SGD with a momentum of 0.9 and learning rate 0.01. The model was trained for 100 epochs with a batch size of 64.

In the case of HGR, we used RANP as the training loss function. We employed Adam as the optimizer with a learning rate of 0.0001. We trained the model for 50 epochs with a batch size of 64.

Dataset. We used the full training set to train the models and we used a small validation set taken from the training set to keep track of the learning. We used the RGB, flow, and audio features extracted with TBN [6] which were provided by the dataset authors.

4.2. Ensembling strategy

After learning the aforementioned models, we created the similarity matrix of each of the five models. These are then summed before taking the mean similarity values. By doing so, the similarity of video v_i and caption q_j is computed as the mean of the five similarity values predicted by the models. Finally, we use this mean similarity matrix in the submission.

4.3. Results

In Table 1 we report the performance obtained by the various models used within the ensemble on the validation set. The final model which we submitted to the leaderboard (**Ens.**) obtained the best results on the validation set. Moreover, when compared to previous state-of-the-art approaches (from the EPIC-Kitchens-100 Multi-Instance Retrieval Challenge 2021 [3]), our ensemble shows considerable improvements: in fact, both Wray et al. (JPoSE trained without the relevance-based margin) and Hao et al. [3] obtained on average around 53% nDCG and 44% mAP, whereas our ensemble obtains around 61% nDCG and almost 50% mAP. On the other hand, when comparing to the current public leaderboard, we achieve top-1 mAP performance (49.77% compared to 47.39% obtained by the second best) and top-2 nDCG (61.02% compared to 61.44%).

	Validation					
	nDCG (%)			mAP (%)		
Mod.	v2t	v2t	avg	v2t	v2t	avg
1	74.6	71.1	72.8	78.7	74.1	76.4
2	81.2	77.4	79.3	85.4	75.4	80.4
3	81.4	77.5	79.5	85.0	74.3	79.7
4	81.7	77.7	79.7	85.0	72.6	78.8
5	82.0	78.1	80.1	86.4	75.8	81.1
Ens.	**82.8**	**79.5**	**81.2**	**88.2**	**78.7**	**83.5**
	Official test					
	nDCG (%)			mAP (%)		
Ens.	**63.16**	**58.88**	**61.02**	**55.15**	**44.39**	**49.77**

Table 1. Performance of the five considered models on the validation set (top) and test set (bottom) of EPIC-Kitchens-100. The similarity scores predicted by the ensemble are obtained by averaging the predictions made by each individual model.

5. Conclusion

In this report, we summarized the details of our submission to the EPIC-Kitchens-100 Multi-Instance Retrieval Challenge 2022. The proposed ensemble, comprising several models trained with relevance-augmented version of the standard triplet loss, achieves considerable improvements when compared to last year challenge competitors. Moreover, the result we obtain is visible on the public leaderboard and obtains top-1 performance in mAP (with a margin of 2.4%) and top-2 performance in nDCG (with a difference of 0.4%).

Acknowledgements

We gratefully acknowledge the support from Amazon AWS Machine Learning Research Awards (MLRA) and NVIDIA AI Technology Centre (NVAITC), EMEA. We acknowledge the CINECA award under the ISCRA initiative, which provided computing resources for this work.

References

[1] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10638–10647, 2020. 1, 2

[2] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 2021. 1

[3] Dima Damen, Adriano Fragomeni, Jonathan Munro, Toby Perrett, Daniel Whettam, Michael Wray, Antonino Furnari, Giovanni Maria Farinella, and Davide Moltisanti. Epic-kitchens-100- 2021 challenges report. Technical report, University of Bristol, 2021. 2

[4] Alex Falcon, Giuseppe Serra, and Oswald Lanz. Learning video retrieval models with relevance-aware online mining. In *International Conference on Image Analysis and Processing*, pages 182–194. Springer, 2022. 1, 2

[5] Alex Falcon, Swathikiran Sudhakaran, Giuseppe Serra, Sergio Escalera, and Oswald Lanz. Relevance-based margin for contrastively-trained video retrieval models. *ACM International Conference on Multimedia Retrieval*, 2022. 1

[6] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5492–5501, 2019. 2

[7] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 1

[8] Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*, 2019. 2

[9] Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 450–459, 2019. 1, 2

Egocentric Video-Language Pretraining @ EPIC-KITCHENS-100 Multi-Instance Retrieval Challenge 2022

Kevin Qinghong Lin[1] Alex Jinpeng Wang[1] Rui Yan[1] Eric Zhongcong Xu[1] Rongcheng Tu[2]
Yanru Zhu[2] Wenzhe Zhao[2] Weijie Kong[2] Chengfei Cai[2] Hongfa Wang[2]
Wei Liu[2] Mike Zheng Shou[1*]

[1]Show Lab, National University of Singapore [2]Tencent Data Platform

{kevin.qh.lin, yanrui6019, turongcheng}@gmail.com, {jinpengwang, zhongcongxu}@u.nus.edu
{yizhizhu, carsonzhao, jacobkong, fletchercai, hongfawang}@tencent.com
wl2223@columbia.edu, mike.zheng.shou@gmail.com

Abstract

In this report, we propose a video-language pretraining (VLP) based solution [8] for the EPIC-KITCHENS-100 Multi-Instance Retrieval (MIR) challenge. Especially, we exploit the recently released Ego4D dataset [6] to pioneer Egocentric VLP from pretraining dataset, pretraining objective, and development set. Based on the above three designs, we develop a pretrained video-language model that is able to transfer its egocentric video-text representation to MIR benchmark. Furthermore, we devise an adaptive multi-instance max-margin loss to effectively fine-tune the model and equip the dual-softmax technique for reliable inference. Our best single model obtains strong performance on the challenge test set with 47.39% *mAP and* 61.44% *% nDCG. The code will be available at* https://github.com/showlab/EgoVLP.

1. Introduction

Video-Language Pretraining (VLP) has prevailed in the regime of Vision + Language, aiming to learn strong and transferable video-language representation for powering a broad spectrum of video-text downstream tasks, video-text retrieval, video question answering, video-captioning. The successes of VLP mainly stems from the availability of large-scale open-world video-text datasets such as HowTo100M [9], which scrapes 134K hours of instructional videos from the YouTube accompanied by text yielded from Automatic Speech Recognition.

Despite reaching an impressive data scale, videos in the existing video-text pretraining datasets [1, 9] are often of 3rd-person views and might have been edited before posting on the web. Yet, there is a noticeable domain gap between the existing video-text pretraining datasets and 1st-person view videos such as those videos captured by wearable cameras or smart glasses. Egocentric video has received increasing interests from academia (e.g., activity anticipation [4]) and industry (various applications in robotics and augmented reality). But, due to such a domain gap, directly transferring the existing VLP models to egocentric downstream tasks cannot fully unleash the potential of large-scale pretraining approaches. Roused by the favorable scale and diversity of recently released Ego4D [6] dataset, we are motivated to develop Egocentric VLP models [8], which can greatly benefit various egocentric video downstream applications.

In this report, we leverage our Egocentric VLP [8] for powering EPIC-KITCHENS-100 Multi-Instance Retrieval (MIR) challenge. We provide a comprehensive analysis of the impact of different VLPs on this task, e.g., without VLP, 3rd-person VLP, and 1st-person VLP. Furthermore, to effectively transfer the video-text representation to MIR task, we devise an adaptive multi-instance maxmargin loss for fine-tuning. Besides, we introduce the dual-softmax technique for reliable inference.

2. Approach

2.1. VLP Model

We choose Frozen [1] as our pretraining architecture. As depicted in the Fig. 1(b), Frozen [1] design encompasses an elegant and simple dual encoder strategy (one per modality) which has favorable characteristics (e.g., indexability and efficiency [1]). Note that this allows the pretrained model for single-modality tasks (e.g., video-only tasks). In prac-

*Corresponding Author.

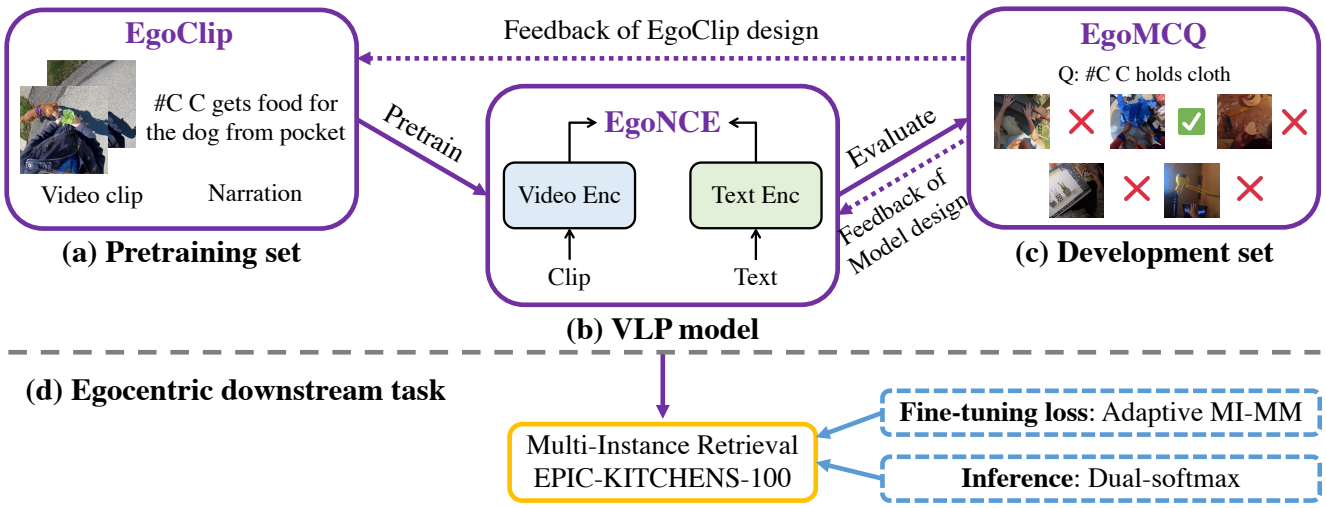


Figure 1. **Top:** Our Egocentric VLP framework, which includes: (a) pretraining set EgoClip; (b) VLP model; and (c) development set EgoMCQ. We use EgoClip to pretrain a VLP model with EgoNCE loss and then evaluate on EgoMCQ. According to the feedback, we iteratively refine our designs of (a) and (b). **Down:** We transfer our pretrained model to EPIC-KITCHENS-100 Multi-Instance Retrieval task by equipping two techniques: the adaptive multi-instance max-margin loss for fine-tuning and the dual-softmax for inference.

tice, the video encoder is a TimeSformer [2] architecture while the text encoder builds upon DistillBERT [10]. We adopt this notation: $(\mathcal{V}_i, \mathcal{T}_i)$ represent the video-test input to the model, while $\mathbf{v}_i$ and $\mathbf{t}_i$ are used to identify the L2 normalized video and text embedding with d dimension.

2.2. Egocentric Pretraining

As illustrated in Fig. 1 [8], our pretraining framework includes three designs: EgoClip, EgoNCE, and EgoMCQ. We use EgoClip dataset for pretraining, which comprises 3.85M video-text pairs well-chosen from Ego4D, covering a large variety of human daily activities. Next, we employ EgoNCE as the model pretraining objective, which extends video-text InfoNCE [1] via positive and negative sampling strategies with formulation:

$$\mathcal{L}^{\text{ego}} = \mathcal{L}^{\text{ego}}_{\text{v2t}} + \mathcal{L}^{\text{ego}}_{\text{t2v}}. \tag{1}$$

We formulate $\mathcal{L}^{\text{ego}}_{\text{v2t}}$ for simplicity whereas $\mathcal{L}^{\text{ego}}_{\text{t2v}}$ is defined in a symmetry way.

$$\mathcal{L}^{\text{ego}}_{\text{v2t}} = \frac{1}{|\widetilde{\mathcal{B}}|} \sum_{i \in \widetilde{\mathcal{B}}} \log \frac{\sum_{k \in \mathcal{P}_i} \exp(\mathbf{v}_i^T \mathbf{t}_k / \tau)}{\sum_{j \in \mathcal{B}} \left(\exp(\mathbf{v}_i^T \mathbf{t}_j / \tau) + \exp(\mathbf{v}_i^T \mathbf{t}_{j'} / \tau) \right)}, \tag{2}$$

where the numerator term corresponds to our proposed **action-aware positive samples**, which select the positive sample within a batch by identifying narrations nouns and verbs. Then, batch samples that shared at least one noun and at least one verb are treated as positive samples: $\mathcal{P}_i = \{j \in \mathcal{B} \mid \text{noun}(j) \cap \text{noun}(i) \neq \varnothing, \text{verb}(j) \cap \text{verb}(i) \neq \varnothing\}$. While the denominator term corresponds to our proposed **scene-aware negative samples**. For each video clip i, we sample an adjacent clip $i' \in \mathcal{N}(i)$, which is close to i in time (less than 1 min) within the same video. Hence the batch is updated as $\widetilde{\mathcal{B}} = \{\underbrace{1, 2, \cdots N}_{\mathcal{B}}, \underbrace{1', 2', \cdots, N'}_{\mathcal{N}(\mathcal{B})}\}$. EgoNCE provides a general extension to adapt the existing VLP models for video-text pretraining datasets in the egocentric domain.

We evaluate our designs of EgoClip and EgoNCE on EgoMCQ, which contains 39K video-text multi-choices questions that are closer to pretraining domains and benchmark model video-text alignment, powering us to accurately validate and quickly iterate our decisions.

2.3. Task-specific Transferring

In this section, we focus on effectively transferring pretrained video-text representations to EPIC-KITCHENS-100 Multi-Instance Retrieval task. In this task, a narration may be jointly associated with multiple clips, so a multi-instance learning mechanism can better handle such a situation. And this dataset provides the action label to calculate the correlation $c_{ij} \in [0, 1]$ between two clip-text pairs (i, j), which supports the application of Multi-Instance MaxMargin loss (MI-MM), as recommended in baseline [11].

$$\mathcal{L} = \sum_{(i,j,k) \in \Omega} \max\left(\gamma + \mathbf{v}_i^T \mathbf{t}_j - \mathbf{v}_i^T \mathbf{t}_k\right) + \left(\gamma + \mathbf{t}_i^T \mathbf{v}_j - \mathbf{t}_i^T \mathbf{v}_k\right), \tag{3}$$

where $\Omega = \{(i, j, k) \mid j \in i^+, k \in i^-\}$ is a triple, which indicates a positive instance j and a negative instance k for i. In our setting, we define the positive set as $i^+ = \{j | c_{ij} > 0.1\}$ and the negative as the remains sample within the batch. The γ is a constant margin factor.

However, different combinations are shared with the

Methods	Vis Enc Input	# Frames	Vis-text PT	mAP (%)			nDCG (%)		
				V→T	T→V	Avg	V→T	T→V	Avg
Random	-	-	-	5.7	5.6	5.7	10.8	10.9	10.9.
MI-MM	S3D	32	HowTo100M	34.8	23.6	29.2	47.1	42.4	44.7
MME [11]	TBN † [7]	25	-	43.0	34.0	38.5	50.1	46.9	48.5
JPoSE [11]	TBN † [7]	25	-	49.9	38.1	44.0	55.5	51.6	53.5
Frozen	Raw Videos	4	-	38.8	29.7	34.2	50.5	48.3	49.4
Frozen	Raw Videos	4	HowTo100M	39.2	30.1	34.7	50.7	48.7	49.7
Frozen	Raw Videos	4	CC3M+WebVid2M	41.2	31.6	36.4	52.7	50.2	51.4
Frozen	Raw Videos	4	EgoClip	44.5	34.7	39.6	55.7	52.9	54.3
Frozen+EgoNCE	Raw Videos	4	EgoClip	**45.1**	**35.3**	**40.2**	**56.2**	**53.5**	**54.8**
Frozen	Raw Videos	16	CC3M+WebVid2M	45.8	36.0	40.9	57.2	54.3	55.8
Frozen+EgoNCE	Raw Videos	16	EgoClip	**49.9**	**40.5**	**45.0**	**60.9**	**57.9**	**59.4**
Frozen	Raw Videos	4	HowTo100M.	6.8	6.3	6.5	11.6	12.8	12.2
Frozen	Raw Videos	4	CC3M+WebVid2M.	8.6	7.4	8.0	14.5	14.6	14.5
Frozen	Raw Videos	4	EgoClip.	17.9	13.1	15.5	23.0	21.2	22.1
Frozen+EgoNCE	Raw Videos	4	EgoClip	**19.4**	**13.9**	**16.6**	**24.1**	**22.0**	**23.1**

Table 1. Performance of the EPIC-KITCHENS-100 Multi-Instance Retrieval. Note that TBN † feature [7] are a combination of three modalities: RGB, Flow and Audio. Conversely, our approach only relies on RGB input. The grey rows correspond to **zero-shot evaluation**.

same margin γ in Eq. 3 and thus are treated equally when fine-tuning. Intuitively, if two sample (i, j) are highly similar, they should be pulled closer with a larger margin surpassing the (i, k). Otherwise, they should be pulled with a small margin if not very similar. Thus, we devise the following **Adaptive MI-MM** to extend the Eq.3.

$$\mathcal{L}^{\dagger} = \sum_{(i,j,k)\in\Omega} \max\left(c_{ij}\gamma + \mathbf{v}_i^T\mathbf{t}_j - \mathbf{v}_i^T\mathbf{t}_k\right) + \left(c_{ij}\gamma + \mathbf{t}_i^T\mathbf{v}_j - \mathbf{t}_i^T\mathbf{v}_k\right), \tag{4}$$

where c_{ij} adaptively control the marginal, e.g., two instances (i, j) that are semantically identical ($c_{ij} = 1$) will be assigned a largest marin 1.0γ. Otherwise, a less margin 0.1γ is given when they are not very similar ($c_{ij} = 0.1$).

Inference. After we finalize the fine-tuning, we use the model to encode video and text embeddings for all samples within the test set. To obtain the cross-modal retrieval results, a common way is to calculate the similarity score between a text embedding $\mathbf{t}_i$ and a video embedding $\mathbf{v}_j$ and index the maximum as the top retrieval result. Here, motivated by [3], we introduce the dual softmax techniques to better scale the similarities and filter the hard case, thus reaching more reliable prediction results. We show the PyTorch-like pseudo-code in Alg. 1 to compare the two inference way. Notably, the dual-softmax only works on inference and thus does not introduce additional training costs, and it is flexible to different models.

Algorithm 1 Pseudo-code for Dual-softmax (PyTorch-like)

```
# Input(embeddings): T_{Nxd}, V_{Mxd}
# Output(scores): res_{NxM}

# (1) the common way
sim = torch.mm(T,V)
res = F.softmax(sim, axis=0)

# (2) dual-softmax
sim = torch.mm(T,V)
prior = F.softmax(sim/500, axis=1)
res = F.softmax(prior*sim, axis=0)
```

3. Experiments

3.1. Implementation Details

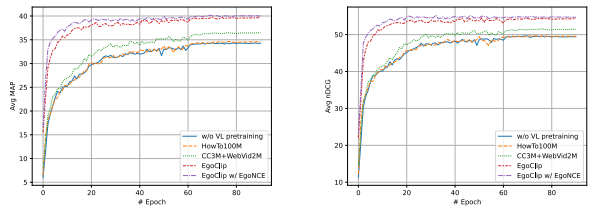
Following the settings of official Frozen [1][1], the video encoder is initialized with ViT [5] weights trained on ImageNet-21K with sequence dimension $D = 768$. The text encoder is based on huggingface's `distilbert-base-uncased`. The dimension of common feature space is set as 256, and the temperature parameter τ is set to 0.05. During pretraining, each video is resized to 224×224 as input with sample frames number 4 and batch size 512. We use the Adam optimizer with a learning rate of 3×10^{-5} with a total epoch of 10. When transferring to MIR task, we select the checkpoints with the best score on EgoMCQ benchmark and fine tune the VLP model on the MIR training set with 67.2K clips. We set the training epoch as 100 and keep other settings the same as pretraining. In the next Sec. 3.2, we use the MI-MM loss

[1]https://github.com/m-bain/frozen-in-time

with γ equal to 0.2 for fine-tuning. And we validate our proposed Adaptive MI-MM and dual-softmax in Sec. 3.3. Since most correlation c_{ij} equal 0.5 in the MIR dataset, we double the γ of Adaptive MI-MM to 0.4 to align with the margin of vanilla MI-MM loss.

3.2. Pretraining Effects

In Tab. 1, we report both zero-shot and fine-tuning evaluation results of different VLP. In the zero-shot setting, pretraining with EgoClip (3.8M), despite being smaller in scale, still outperforms CC3M+WebVid-2M (5.5M) and HowTo100M (136M), validating the unique benefit of pretraining on egocentric data. When fine-tuned with 4 frames, EgoClip pretraining maintains a margin over the best baseline CC3M+WebVid-2M, further verifying the viewpoint domain gap within fine-tuning. Lastly, we increase the sample frames of our finalized model as well as best competitor CC3M+WebVid-2M pretraining to 16. As expected, performance gains accompany the frame increase. We deem that notable benefits come from better temporal modeling for frequent action interactions in the 1st-person view. Overall, our pretraining model outperforms the best baseline (JPoSE) by 1.0 mAP and 5.9% nDCG while requiring fewer frames and input modalities.



(a) mAP with training epoch (b) nDCG with training epoch

Figure 2. Training curves of MIR task.

In Fig 2, we display training curves of MIR under different VLP discussed in Tab.3.2.We can found that: These models with video-text pretraining have a faster rise in performance. Except for HowTo100M, which is close to baseline without pretraining. With EgoClip for egocentric pretraining, the VLP model achieves nearly convergent performance with only a small number of epochs (less than 20). Especially with EgoNCE as the pretraining objective, this positive effect is further enhanced.

3.3. Transferring Ablations

In Tab.2, we validate different fine-tuning strategies when transfer the best pretrained model (Frozen+EgoNCE in Tab.1) to Multi-Instance Retrieval task, and we adopt the common way to calculate the similarity scores by default. It shows that InfoNCE performs poorly as a fine-tuning loss despite it being widely used in 3rd-person datasets e.g., Frozen [1] fine-tune on MSR-VTT. When replacing InfoNCE with MI-MM (Eq.3), there is a signifcant improvement, since MI-MM is well aligned with the mutli-positive characteristic of the EPIC-KITCHENS-100. Moreover, Adaptive MI-MM pushes the performance beyond MI-MM by introducing an adaptive margin (Eq.4), thus serving as a better fine-tune objective in MIR. By equipping dual-softmax to scale similarities, we reach extra 1.2% mAP and 1.0 nDCG performance gains, which is our best single-model performance.

Methods	mAP (%)			nDCG (%)		
	V→T	T→V	Avg	V→T	T→V	Avg
InfoNCE	40.9	34.9	37.9	57.8	56.0	56.9
MI-MM	49.9	**40.5**	45.0	60.9	57.9	59.4
Adaptive MI-MM	**52.3**	40.1	**46.2**	**62.2**	**58.6**	**60.4**
w/ Dual softmax	**53.8**	**40.9**	**47.4**	**63.2**	**59.6**	**61.4**

Table 2. Ablation of different transferring strategies.

4. Conclusion and Limitations

We present an egocentric video-language pretraining solution [8] for the EPIC-KITCHENS-100 MIR challenge. Specifically, we develop a video-language transformer model and exploit the recently released Ego4D dataset [6] to reach strong video-text representation. Furthermore, for this challenge, we devise an Adaptive MI-MM loss to fine-tune and adopt dual-softmax techniques to improve inference. Extensive experimental results validate the effectiveness of our Egocentric VLP and the transferring strategies.

Limitations: VLP requires a large training cost (1,536 GPU hrs for our model). and may be limited by the model architecture thus not flexible for a specific task.

References

[1] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, pages 1728–1738, 2021. 1, 2, 3, 4

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding. *arXiv preprint arXiv:2102.05095*, 2(3):4, 2021. 2

[3] Xing Cheng, Hezheng Lin, Xiangyu Wu, Fan Yang, and Dong Shen. Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss. *arXiv preprint arXiv:2109.04290*, 2021. 3

[4] 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: Collection, pipeline and challenges for epic-kitchens-100. *IJCV*, 130(1):33–55, 2022. 1

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner,

Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 3

[6] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. *arXiv preprint arXiv:2110.07058*, 2021. 1, 4

[7] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *ICCV*, pages 5492–5501, 2019. 3

[8] Kevin Qinghong Lin, Alex Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Eric Zhongcong Xu, Difei Gao, Rongcheng Tu, Weijie Kong Wenzhe Zhao, Chengfei Cai, Hongfa Wang, Bernard Ghanem Dima Damen, Wei Liu, and Mike Zheng Shou. Egocentric video-langugae pretraining. *arXiv preprint arXiv:2206.01670*, 2022. 1, 2, 4

[9] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*, pages 2630–2640, 2019. 1

[10] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019. 2

[11] Michael Wray, Diane Larlus, Gabriela Csurka, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *ICCV*, pages 450–459, 2019. 2, 3

Team IIE-MRG: Technical Report for EPIC-KITCHENS-100 2022 Multi-Instance Retrieval Challenge

Xiaoshuai Hao, Yufan Liu, Wanqian Zhang, Dayan Wu, Bo Li
Institute of Information Engineering, Chinese Academy of Sciences
{haoxiaoshuai, liuyufan,zhangwanqian, wudayan, libo}@iie.ac.cn

Abstract

In this report, we present a solution to the EPIC-KITCHENS-100 2022 Multi-Instance Retrieval Challenge. The task of retrieving relevant videos with natural language queries plays a critical role in effectively indexing large-scale video data. The primary goal of cross-modal video-text retrieval is to map text and video features into a joint embedding space, where semantically similar texts and videos are closer and vice versa. However, existing methods fail to exploit the co-occurrence information, i.e., the intrinsic connections between videos and their corresponding descriptions (text modality). In this report, we propose a novel method named Cross-Modal Alignment Network (CMAN) for video-text retrieval, which sufficiently utilizes the co-occurred video- text pairs. CMAN explores the similarity information of different modalities with introduced semantic alignment and the bi-directional ranking loss, which effectively aligns the similarities and bridges the modality gap. Meanwhile, the similarities between instances of each single modality is exploited by the intra-modal alignment. Moreover, to further utilize the intrinsic co-occurrence information, inter-modal alignment is proposed to align features of one modality with features of the other within each pair. This novel method allowed us to achieve the 3rd place in the CVPR 2022 workshop of EPIC KITCHENS-100 Multi- Instance Retrieval Challenge.

1. Multi-Instance Retrieval Challenge

In this report, we present the method that we implemented for the EPIC-KITCHENS-100 2022 Multi-Instance Retrieval Challenge. This challenge tackles the task of caption-to-video retrieval. Specifically, given a query action segment, the aim of video-to-text retrieval is to rank captions in a gallery set, C, such that those with a higher rank are more semantically relevant to the action in the video. Conversely, text-to-video retrieval uses a query caption $c_i \in C$ to rank videos. The challenge uses EPIC-KITCHENS-100 dataset [4]. The EPIC-KITCHENS-100 dataset is an unscripted egocentric action dataset collected from 45 kitchens from 4 cities across the world. Submissions are evaluated on the test set for action retrieval. This Challenge uses two evaluation metrics: mean Average Precision (mAP) and normalised Discounted Cumulative Gain (nDCG).

2. Motivation

With the rapid development of internet and social networks, multimodal data, e.g. videos and texts, become more and more popular in modern search engines. Cross-modal video-text retrieval aims at searching semantically relevant videos with text queries and vice versa [1, 8, 7, 9]. The realistic and challenging scenario of video-text retrieval is that videos usually co-occur with their descriptions in pairs, however it's difficult to obtain their labels or categories.

However, these methods ignore the intrinsic co-occurrence information in the video-text pairs, which further leads to the failure of capturing the precise relations among data in different modalities. Specifically, the videos and their corresponding texts often co-occur in one pair, indicating that the features for the video and the text within one pair should have the smallest distance, or equivalently the maximum degree of similarity. Meanwhile, the similarity between instances within each single modality is overlooked in previous methods. Besides, the semantic correlation of the features from one modality and the corresponding ones from another modality is also ignored by existing methods, which hinders the coherent interactions of features from both modalities within each pair.

To tackle these three issues, in this paper we propose a novel cross-modal video-text retrieval method, named Cross-Modal Alignment Network (CMAN). CMAN seamlessly integrates the co-occurence information and the semantic correlation of different modalities in a unified framework, which is illustrated in Fig. 1. CMAN first extracts the feature vectors from the original data for each modality. Then, our model utilizes the bi-directional ranking loss

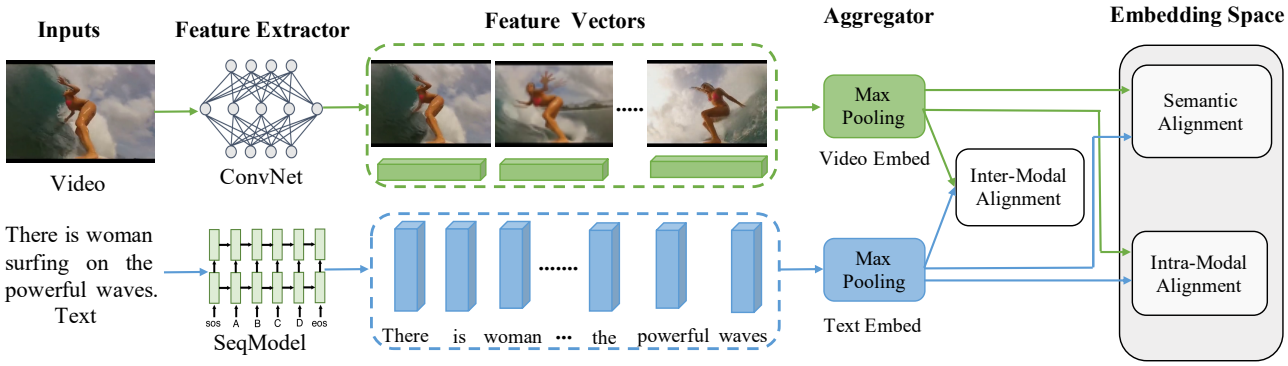


Figure 1. The overall framework of Cross-Modal Alignment Network (CMAN).

to guide the semantic alignment. Moreover, we utilize the intra-modal alignment to exploit the similarities between instances of each single modality. Last but not least, we introduce the inter-modal alignment to make use of the semantic correlation of the feature vectors from both modalities within one pair.

3. Proposed Method

3.1. Notations and Definitions

Let $\{(v_i, t_i) \,|\, v_i \in V, t_i \in T\}$ be a set of videos with v_i being the visual representation of the i^{th} video sequence and t_i the corresponding textual caption. Our goal of video-text retrieval is to learn a pair of functions $\varphi(v)$ and $\psi(t)$ to map videos and texts into a joint embedding space, in which embeddings for matched texts and videos should lie close together, while embeddings for mismatched texts and videos should lie far apart. Next, we elaborate details of the proposed method, including the feature extracting, alignment modules and the overall loss function.

Given a video, following the widely-adopted approach [**?**], we utilize a CNN model pre-trained on ImageNet, wherein outputs of the last pooling layer are considered as the video frame features. By a simple max pooling operation, we can aggregate all frame features into one global video feature. Similarly, given a sentence, we employ a bi-directional LSTM and also adopt a max pooling layer to aggregate the hidden states of all time steps, and the output is regarded as the sentence feature. Once obtained these fixed features, we utilize a video encoder $\varphi(\cdot)$ and a text encoder $\psi(\cdot)$ to map each video sample v and text description t into a joint embedding space. The visual embedding $\varphi(v) \in \mathbb{R}^M$ and text embedding $\psi(t) \in \mathbb{R}^M$ are semantically relevant if the text describes the video, where M denotes the dimension in the shared embedding space.

3.2. Semantic Alignment

Videos and their corresponding texts often co-occur in one pair, indicating that the features for the video and the text within one pair should have the smallest distance, or equivalently the maximum degree of similarity. To tackle this, CMAN first extracts the feature vectors from the original data for each modality and introduces the semantic alignment. To be specific, we utilizes the bi-directional ranking loss to guide the semantic alignment learning. While bridging the gap between an anchor and a positive sample, bi-directional ranking loss can also maximize the distance between an anchor and a negative sample. The expression of the bi-directional ranking loss for the video is as follows:

$$\begin{aligned} \mathcal{L}_{v,t} &= \sum_{(i,j,k) \in \mathcal{T}_{v,t}} max\left(m - s\left(v_i, t_j\right) + s\left(v_i, t_k\right), 0\right) \\ s.t. \quad \mathcal{T}_{v,t} &= \{(i,j,k) \,|\, v_i \in V, t_j \in T_{i+}, t_k \in T_{i-}\}. \end{aligned} \tag{1}$$

Analogously, given a text input, we set the bi-directional ranking loss for the text as follows:

$$\begin{aligned} \mathcal{L}_{t,v} &= \sum_{(i,j,k) \in \mathcal{T}_{t,v}} max\left(m - s\left(t_i, v_j\right) + s\left(t_i, v_k\right), 0\right) \\ s.t. \quad \mathcal{T}_{t,v} &= \{(i,j,k) \,|\, t_i \in T, v_j \in V_{i+}, v_k \in V_{i-}\}, \end{aligned} \tag{2}$$

where m is a constant margin, T_{i+}, T_{i-} respectively define sets of relevant and non-relevant captions and V_{i+}, V_{i-} the sets of relevant and non-relevant videos sequences for multi-modal object (v_i, t_i), respectively. $s(.)$ is the similarity scores in the embedded space. We calculate similarity scores with the cosine similarity, which is a widely-used similarity metric and has been proved effective [5, 3]:

$$s\left(v_i, t_j\right) = \frac{\varphi\left(v_i\right) \cdot \psi\left(t_j\right)}{\|\varphi\left(v_i\right)\| \|\psi\left(t_j\right)\|}, \tag{3}$$

where $\varphi(v_i)$ and $\psi(t_j)$ are the corresponding mapped features, and $\|\cdot\|$ denotes the l_2 norm of vectors and the Frobenius norm of matrices.

Rank	Team	Submissions		SLS			mean Average Precision(mAP)			normalised Discounted Cumulative Gain(nDCG)		
		Entries	Date	PT	TL	TD	T2V	V2T	Avg.	T2V	V2T	Avg.▲
1	kevin.lin	3	05/30/22	3.0	3.0	3.0	40.95	53.84	47.39	59.60	63.29	61.44
2	afalcon	3	06/01/22	2.0	3.0	3.0	44.39	55.15	49.77	58.88	63.16	61.02
3	**haoxiaoshuai**	**11**	**05/31/22**	**2.0**	**3.0**	**3.0**	**38.34**	**46.69**	**44.02**	**51.31**	**54.82**	**53.06**
4	buraksatar	12	05/26/22	2.0	3.0	3.0	38.10	47.52	42.81	54.12	56.55	55.33
5	MI-MM	1	12/10/21	2.0	3.0	3.0	23.08	32.09	27.58	40.48	43.72	42.10

Table 1. Video-to-Text and Text-to-Video retrieval results on the EPIC-KITCHENS-100 dataset.

3.3. Intra-Modal Alignment

Existing works train the embedding network only with the consideration of the semantic alignment between different modalities, which makes the semantically similar texts and videos become closer and vice versa. However, the similarity between instances within each single modality is often overlooked in previous methods.

To that end, we further propose the intra-modal alignment to exploit the similarities between instances of each single modality. The intra-modal alignment ensures that the neighborhood structure within each modality is preserved in the newly built joint embedding space. Specifically, in the learned video embedding space, we enable similar videos to be close to each other. Similarly, in the learned text embedding space, texts of the same videos are expected to be close to each other. This can also provide a useful regularization term for the cross-view matching task. Thus, the expression of the intra-modal alignment loss for the video modality can be defined as:

$$\begin{aligned} \mathcal{L}_{v,v} &= \sum_{(i,j,k) \in \mathcal{T}_{v,v}} max\left(m - s\left(v_i, v_j\right) + s\left(v_i, v_k\right), 0\right) \\ s.t. \quad \mathcal{T}_{v,v} &= \left\{(i,j,k) \,|\, v_i \in V, v_j \in V_{i+}, v_k \in V_{i-}\right\}. \end{aligned} \tag{4}$$

Similarly, we define the intra-modal alignment loss for the text modality as:

$$\begin{aligned} \mathcal{L}_{t,t} &= \sum_{(i,j,k) \in \mathcal{T}_{t,t}} max\left(m - s\left(t_i, t_j\right) + s\left(t_i, t_k\right), 0\right) \\ s.t. \quad \mathcal{T}_{t,t} &= \left\{(i,j,k) \,|\, t_i \in T, t \in T_{i+}, t \in T_{i-}\right\}. \end{aligned} \tag{5}$$

3.4. Inter-Modal Alignment

During the whole training procedure, merely utilizing the cross-modal semantic alignment will lead to the ignorance of inherent characteristics within each modality. We address that the semantic correlation of the features from one modality and the corresponding ones from another modality is also ignored by existing methods, which hinders the coherent interactions of features from both modalities within each pair.

To tackle this issue, in this paper we introduce the inter-modal alignment to make use of the semantic correlation of the feature vectors from both modalities within one pair. To ensure that the pairwise structure within each pair is preserved in the newly built joint embedding space, formally, the inter-modal alignment can be formulated as:

$$\mathcal{L}_c = \Sigma_{i=1}^{N} ||v_i - t_i||, \tag{6}$$

where N is the batch size.

Combining the above loss terms together, the overall objective function can be formulated as:

$$\mathcal{L} = \mathcal{L}_{v,t} + \mathcal{L}_{t,v} + \lambda_1 \mathcal{L}_{v,v} + \lambda_2 \mathcal{L}_{t,t} + \lambda_3 \mathcal{L}_c \,, \tag{7}$$

where λ_1, λ_2 and λ_3 are hyper parameters for balancing these terms.

4. Experiments

4.1. Datasets and Evaluation Metrics

Datasets. To show the effectiveness of the proposed method, we conduct experiments on EPIC-KITCHENS-100 dataset [4]. As access to the captions are required for both video-to-text and text-to-video retrieval, the Val set is used for evaluating this challenge to allow the held-out Test set for all other challenges to remain intact. We consider all the videos in Val, and all unique captions, removing repeats.

Evaluation Metrics. We uses two evaluation metrics: mean Average Precision (mAP) and normalised Discounted Cumulative Gain (nDCG) in the CVPR 2022 workshop of EPIC KITCHENS-100 Multi-Instance Retrieval Challenge. Mean Average Precision (mAP) has also been used for retrieval baselines [10, 5] as it allows for the full ranking to be evaluated. nDCG has been used previously for information retrieval [2, 6, 10]. It requires similarity scores between all items in the test set.

4.2. Implementation Details

We set $\lambda_1 = \lambda_2$ to 0.01 and set λ_3 to 0.005, the margin of the bi-directional ranking loss to 0.2, and the mini-batch size to 128.

4.3. Result

Results are shown in Table 1. CMAN verifies the effectiveness of simultaneously considering the semantic alignment, intra-modal alignment and the inter-modal alignment. At the closing of the challenge, CMAN(haoxiaoshuai) is ranked 3rd on the leaderboard. Table 1 shows the reported results on all metrics.

5. Conclusion

In this report, we present a solution to the EPIC-KITCHENS-100 2022 Multi-Instance Retrieval Challenge. In this report, we propose a novel method named Cross-Modal Alignment Network (CMAN) for video-text retrieval, which sufficiently exploit the co-occurrence information, i.e., the intrinsic connections between videos and texts. With the introduced semantic alignment and the bi-directional ranking loss, CMAN effectively aligns the similarities and bridges the modality gap. Moreover, intra-modal alignment and inter-modal alignment are proposed to utilize the similarities between instances of single modality and those of both modalities within one pair, respectively. This novel method allowed us to achieve the 3rd place in the CVPR 2022 workshop of EPIC KITCHENS-100 Multi-Instance Retrieval Challenge.

References

[1] Samuel Albanie, Yang Liu, Arsha Nagrani, Antoine Miech, Ernesto Coto, Ivan Laptev, Rahul Sukthankar, Bernard Ghanem, Andrew Zisserman, Valentin Gabeur, Chen Sun, Karteek Alahari, Cordelia Schmid, Shizhe Chen, Yida Zhao, Qin Jin, Kaixu Cui, Hui Liu, Chen Wang, Yudong Jiang, and Xiaoshuai Hao. The end-of-end-to-end: A video understanding pentathlon challenge (2020). *CoRR*, abs/2008.00744, 2020. 1

[2] Olivier Chapelle and Mingrui Wu. Gradient descent optimization of smoothed information retrieval metrics. *Information retrieval*, 2010. 3

[3] Shizhe Chen, Yida Zhao, Qin Jin, and Qi Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[4] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Antonino Furnari, Evangelos Kazakos, Jian Ma, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision. *CoRR*, abs/2006.13256, 2020. 1, 3

[5] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9346–9355, 2019. 2, 3

[6] FilipRadlinski and NickCraswell. Comparingthesensitivity of information retrieval metrics. *ACM SIGIR*, 2010. 3

[7] Xiaoshuai Hao, Yucan Zhou, Dayan Wu, Wanqian Zhang, Bo Li, and Weiping Wang. Multi-feature graph attention network for cross-modal video-text retrieval. In *Proceedings of the ACM on International Conference on Multimedia Retrieval*, 2021. 1

[8] Xiaoshuai Hao, Yucan Zhou, Dayan Wu, Wanqian Zhang, Bo Li, Weiping Wang, and Dan Meng. What matters: Attentive and relational feature aggreggation network for video-text retrieval. In *IEEE International Conference on Multimedia and Expo*, 2021. 1

[9] Michael Wray, Gabriela Csurka, Diane Larlus, and Dima Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *ICCV*, pages 450–459, 2019. 1

[10] M. Wray, H. Doughty, and D. Damen. On semantic similarity in video retrieval. *CVPR*, 2021. 3

Exploiting Semantic Role Contextualized Video Features for Multi-Instance Text-Video Retrieval EPIC-KITCHENS-100 Multi-Instance Retrieval Challenge 2022

Burak Satar[1,2] Zhu Hongyuan[1] Hanwang Zhang[2] Joo Hwee Lim[1,2]

[1]Institute for Infocomm Research, A*STAR, Singapore
[2]School of Computer Science and Engineering, NTU, Singapore
{burak_satar, zhuh, joohwee}@i2r.a-star.edu.sg, hanwangzhang@ntu.edu.sg

Abstract

In this report, we present our approach for EPIC-KITCHENS-100 Multi-Instance Retrieval Challenge 2022. We first parse sentences into semantic roles corresponding to verbs and nouns; then utilize self-attentions to exploit semantic role contextualized video features along with textual features via triplet losses in multiple embedding spaces. Our method overpasses the strong baseline in normalized Discounted Cumulative Gain (nDCG), which is more valuable for semantic similarity. Our submission is ranked 3rd for nDCG and ranked 4th for mAP.

1. Introduction

With the rise of videos uploaded by users via social media channels, cross-modal retrieval of video data and natural language descriptions has gained popularity. The goal of video-to-text retrieval, given a query action segment, is to rank captions in a gallery set so that those with a higher rank are more semantically related to the video action. Text-to-video retrieval, on the other hand, ranks videos based on a query caption.

While most methods [3, 8, 10] use one joint embedding space to align video and text features, recent methods [1,14] use multiple embedding spaces to match video features into the noun and verb embedding spaces along with the textual features but did not consider their interactions. Moreover, it is relatively easy to parse the textual features into verb and noun levels since an off-the-shelf toolkit could be used. However, mapping a video feature into the object and action levels is still challenging, which corresponds to noun and verb levels in text.

Inspired by [5, 11], we implement self-attentions to exploit visual features on top of the baseline, JPoSE [14] by leveraging the contexts from nouns and verbs of the text query with details in the following section. While we outperform the strong baseline in normalized Discounted Cumulative Gain (nDCG), which is more beneficial for semantic similarity, we fall short in mean Average Precision (mAP), a traditional technique for binary relevance. Our approach is ranked third for nDCG, while it is ranked fourth for mAP. We also analyze various failure examples to save the time of the following researchers on this task.

2. Method

We follow the baseline work [14]: we create a pair of functions that map videos and texts into a joint embedding space, in which embeddings for matched texts and videos should be close together, and embeddings for mismatched texts and videos should be far apart, given a video and a query text. A suitable embedding space should also ensure that related videos/texts stay close together.

With this motivation, we first parse caption into the noun t_i^1 and verb t_i^2 levels, followed by linear layers. We utilize linear layers to embed corresponding video features v_i^1, v_i^2 and use a self-attention layer to exploit contextualized features. Then, we concatenate textual and visual features to compute the distance between these representations, $\hat{v_i}$ and $\hat{t_i}$. L1 and L2 refer to triplet losses. The more details of the loss functions are in Eq. 1 and baseline paper [14] and the architecture details are in Fig. 1

In Eq. 1, the first two rows refer to cross-modal losses, and the last two rows indicate within-modal losses. θ function denotes two fully connected layers. δ function signifies two linear layers and one self-attention layer. m refers to the constant margin, while d is the distance function. While i refers to the selected video, j and k denote positive and negative samples, respectively.

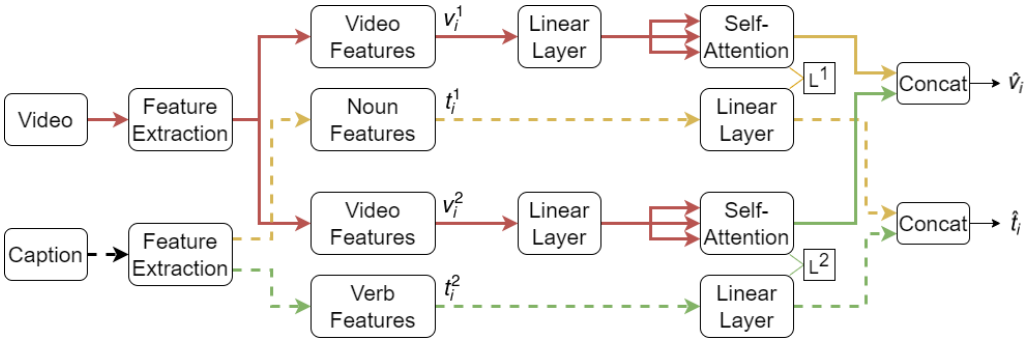


Figure 1. We first parse caption into the noun t_i^1 and verb t_i^2 levels, followed by linear layers. We utilize linear layers to embed corresponding video features v_i^1, v_i^2 and use a self-attention layer to exploit contextualized features. Then, we concatenate textual and visual features to compute the distance between these representations, $\hat{v}_i$ and $\hat{t}_i$. L1 and L2 refer to triplet losses.

$$
\begin{aligned}
L = \lambda_{v,t} \sum_{i,j,k} \max(0, d(\delta_{v_i}, \theta_{t_j}) - d(\delta_{v_i}, \theta_{t_k}) + m) \\
+ \lambda_{t,v} \sum_{i,j,k} \max(0, d(\theta_{t_i}, \delta_{v_j}) - d(\theta_{t_i}, \delta_{v_k}) + m) \\
+ \lambda_{v,v} \sum_{i,j,k} \max(0, d(\delta_{v_i}, \delta_{v_j}) - d(\delta_{v_i}, \delta_{v_k}) + m) \\
+ \lambda_{t,t} \sum_{i,j,k} \max(0, d(\theta_{t_i}, \theta_{t_j}) - d(\theta_{t_i}, \theta_{t_k}) + m)
\end{aligned} \tag{1}
$$

Eq. 2 shows that the visual features V_i are fed into the self-attention layer to encode into z_s. Then, a feed-forward layer FF outputs the final contextualized appearance feature. Normalization of the layer is done under the Norm function.

$$
\begin{aligned}
z_s &= \text{Norm}\big(\text{MultiHead}(V_i, V_i, V_i) + V_i\big) \\
E_v &= \text{Norm}\big(\text{FF}(z_s) + z_s\big)
\end{aligned} \tag{2}
$$

For multi-headed attention layers, we follow [12], as formulated in Eq. 3. All of the W matrices are learned during the training procedure. Since it is a self-attention layer, query Q is the same as key K and value V. After each attention layer, layer normalization and the residual connection are implemented.

$$
\begin{aligned}
&\text{MultiHead}(Q, K, V) = \text{Concat}(\text{Head}_1, ..., \text{Head}_h) W^O \\
&\text{Head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \\
&\text{Attention}(Q, K, V) = \sigma\Big(\frac{QK^T}{\sqrt{d}} V\Big)
\end{aligned} \tag{3}
$$

3. Experiments

3.1. Implementation Details

While $\lambda_{t,v}$ equals 2.0, the other constant margins equal 1.0. The batch size is 64, and the learning rate is 0.01.

Dataset. We undertake experiments on the EPIC-KITCHENS-100 dataset [2], which is a collection of unscripted egocentric action data across the world, to demonstrate the efficiency of our strategy.

Features. We use the video features extracted by TBN [7]. Each one is an nx25x1024 matrix holding a python dictionary containing the RGB, flow, and audio features, where n is the number of video clips. The number of training and test set pairs is 67217 and 9668, respectively. Each feature is followed by temporal mean pooling, making the shape nx1x1024. We utilize the textual features given by [14] using a Word2Vec model trained on the Wikipedia corpus. spaCy parser [4] is used to disentangle the text caption into different PoS tags. The model is trained with the default values of the baseline.

Evaluation Metrics. We utilize two assessment metrics, mAP and nDCG, on the test set to evaluate submissions for action retrieval. Mean Average Precision (mAP) was employed for retrieval baselines because it allows the whole ranking to be analyzed on binary relevance. nDCG has already been used to retrieve information [13]. It necessitates the use of similarity scores throughout the entire test set.

3.2. Results

Table 1 shows the comparison between our method and the baselines. It also compares with the methods attended to this year's challenge. While our method overpass all the baselines on nDCG, it falls short on mAP. The MI-MM approach projects both modalities onto a shared action space using linear layers via max-margin loss, which is a simplified version of [9]. The JPoSE approach [14] uses a triplet loss to separate captions into the verb and noun spaces. The JPoSE* refers to our implementation. DCRL [6] con-

Table 1. Multi-instance retrieval results on the EPIC-KITCHENS-100 test split. T2V and V2T stand for Text-to-Video and Video-to-Text retrieval, respectively. While the above part of the table compares with baselines, the lower part shares the result of this year's competition.

Comparison to Baselines						
Method	**mean Average Precision (mAP)**			**normalised Discounted Cumulative Gain (nDCG)**		
	Average	T2V	V2T	Average	T2V	V2T
MI-MM	27.58	23.08	32.09	42.10	40.48	43.72
JPoSE*	43.95	38.18	49.71	53.40	51.60	55.21
JPoSE [14]	44.01	38.11	49.91	53.53	51.55	55.51
DCRL [6]	**44.23**	**38.49**	**49.96**	53.56	51.83	55.28
Our Method	42.81	38.10	47.52	**55.33**	**54.12**	**56.55**
Comparison to Other Users						
User	**mean Average Precision (mAP)**			**normalised Discounted Cumulative Gain (nDCG)**		
	Average	T2V	V2T	Average	T2V	V2T
haoxiaoshuai	44.02 (3)	38.34 (3)	49.69 (3)	53.06 (4)	51.31 (4)	54.82 (4)
Our Method	42.81 (4)	38.10 (4)	47.52 (4)	55.33 (3)	54.12 (3)	56.55 (3)
afalcon	49.77 (1)	44.39 (1)	55.15 (1)	61.02 (2)	58.88 (2)	63.16 (2)
kevin.lin	47.39 (2)	40.95 (2)	53.84 (2)	61.44 (1)	59.60 (1)	63.29 (1)

siders both inter-modal and intra-modal constraints at the same time to retain both cross-modal semantic similarity and modality-specific consistency in the embedding space.

Failure cases. We also share failure cases which could be helpful for other researchers. For every experiment, we give the results approximately compared to baseline JPoSE [14]. 1) If we implement self-attention to the textual features as it is done to the video features, the results decrease around 2-3%. 2) When we increase the batch size or embedding size, the results decrease 1-2%. 3) We get 1-2% lower results when applying temporal max-pooling rather than mean pooling.

4. Conclusion

In this report, we propose an approach to exploit contextualized video features via self-attentions and disentangling them into multiple embedding spaces. It also parses text into corresponding embedding spaces, and then the similarity between representations is calculated via triplet loss. While our strategy outperforms the strong baseline in normalized Discounted Cumulative Gain (nDCG), a semantic similarity measurement, it falls short in mean Average Precision (mAP), a standard measure of binary relevance. For nDCG, our proposal is ranked third, and for mAP, it is ranked fourth. We plan to exploit each video feature separately via novel fusion methods as well as utilize domain-specific features such as hand-object relations for future work.

References

[1] S. Chen, Y. Zhao, Q. Jin, and Q. Wu. Fine-grained video-text retrieval with hierarchical graph reasoning. In *CVPR*, 2020. 1

[2] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, , Antonino Furnari, Jian Ma, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, and Michael Wray. Rescaling egocentric vision: Collection, pipeline and challenges for epic-kitchens-100. *International Journal of Computer Vision (IJCV)*, 2021. 2

[3] J. Dong, X. Li, C. Xu, S. Ji, Y. He, G. Yang, and X. Wang. Dual encoding for zero-example video retrieval. In *CVPR*, pages 9338–9347, 2019. 1

[4] English spaCy parser. https://spacy.io/. 2

[5] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal Transformer for Video Retrieval. In *European Conference on Computer Vision (ECCV)*, 2020. 1

[6] Xiaoshuai Hao, Yucan Zhou, Dayan Wu, Wanqian Zhang, Bo Li, and Weiping Wang. Multi-feature graph attention network for cross-modal video-text retrieval. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, ICMR '21, page 135–143, New York, NY, USA, 2021. Association for Computing Machinery. 2, 3

[7] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2

[8] Y. Liu, S. Albanie, A. Nagrani, and A. Zisserman. Use what you have: Video retrieval using representations from collaborative experts. In *arXiv*, 2019. 1

[9] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-End

Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*, 2020. 2

[10] N. C. Mithun, J. Li, F. Metze, and A. K. Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *Proceedings of the 2018 ACM on ICMR*, page 19–27, 2018. 1

[11] Burak Satar, Zhu Hongyuan, Xavier Bresson, and Joo Hwee Lim. Semantic role aware correlation transformer for text to video retrieval. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 1334–1338, 2021. 1

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. In *NIPS*, 2017. 2

[13] Michael Wray, Hazel Doughty, and Dima Damen. On semantic similarity in video retrieval. In *CVPR*, 2021. 2

[14] M. Wray, D. Larlus, G. Csurka, and D. Damen. Fine-grained action retrieval through multiple parts-of-speech embeddings. In *ICCV*, 2019. 1, 2, 3