EPIC-KITCHENS-100 Unsupervised Domain Adaptation Challenge: Mixed Sequences Prediction

EPIC-KITCHENS-100 Unsupervised Domain Adaptation Challenge: Mixed Sequences Prediction

Amirshayan Nasirimajd, Simone Alberto Peirone, Chiara Plizzari, Barbara Caputo
Politecnico di Torino
amirshayan.nasirimajd@studenti.polito.it, {simone.peirone, chiara.plizzari, barbara.caputo}@polito.it

Abstract

This report presents the technical details of our approach for the EPIC-Kitchens-100 Unsupervised Domain Adaptation (UDA) Challenge in Action Recognition. Our approach is based on the idea that the order in which actions are performed is similar between the source and target domains. Based on this, we generate a modified sequence by randomly combining actions from the source and target domains. As only unlabelled target data are available under the UDA setting, we use a standard pseudo-labeling strategy for extracting action labels for the target. We then ask the network to predict the resulting action sequence. This allows to integrate information from both domains during training and to achieve better transfer results on target. Additionally, to better incorporate sequence information, we use a language model to filter unlikely sequences. Lastly, we employed a co-occurrence matrix to eliminate unseen combinations of verbs and nouns. Our submission, labeled as 'sshayan', can be found on the leaderboard, where it currently holds the 2nd position for 'verb' and the 4th position for both 'noun' and 'action'.

1. Introduction

Egocentric action recognition, aiming to understand human activities from a first-person view, has gained significant attention with the rising popularity of wearable devices. Those have provided access to a vast amount of data, creating opportunities for applications in augmented reality, robotics, and human-computer interaction.

However, achieving high accuracy on new data becomes challenging due to the so called *domain shift*, which refers to the difference in terms of distribution between the training (source) and test set (target), caused by changes in visual appearance or contexts, leading to a significant performance gap. To address this challenge, the Unsupervised Domain Adaptation (UDA) setting has been proposed, where an unlabeled set of samples from the test distribution is available during training for adaptation.

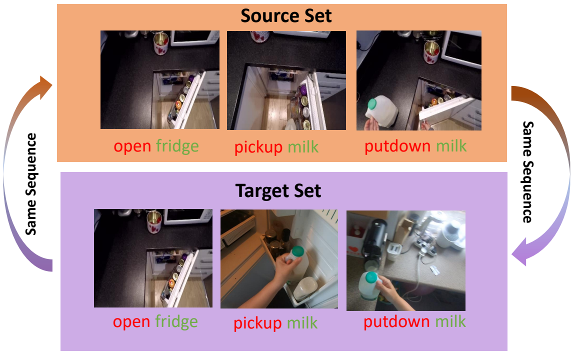

Figure 1. Example of action sequence similarities across two different kitchens. When "*picking something out of the fridge*", the sequence "open", "pick-up", and "put-down" repeats across the two domains.

Human activities tend to follow patterns as actions are strongly influenced by the temporal context in which they are performed. For example, regardless of the person and the environment, *putting something in the fridge* always involves the same steps, as shown in Figure 1. Based on this simple observation, we propose to use sequential prediction and the inherent relationships between action sequences to mitigate the negative effects of domain shift in EAR.

Our approach consists in replacing random action clips in a source domain sequence with action clips from the target domain that represent the same action. By training the model to predict the sequence of actions in the modified sequences, we encourage the model to learn common sequence patterns that exist across different domains, aiming at mitigating the influence of domain shift. Additionally, by taking inspiration from [5], we integrate a language model into our framework to better incorporate the context derived from the surrounding actions. Finally, we compute a co-occurrence matrix on the action of the source domain to filter out improbable predictions of verb and noun pairs, thus further refining the prediction process, as done in [1].

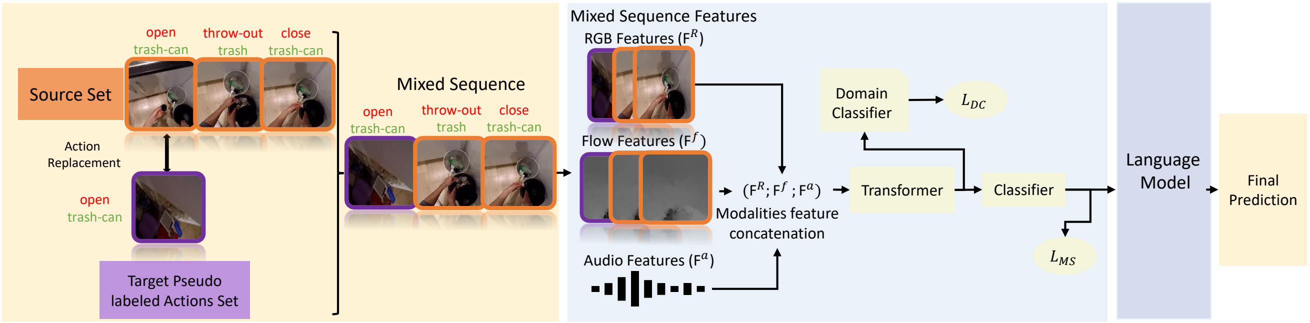

Figure 2. Overall architecture of the proposed framework. First, we randomly choose an action among the n actions of the sequence from the source domain. Next, we use pseudo-labels on the target to find a substitute for the selected action sharing the same verb and noun labels. RGB, optical flow and audio features of the samples in the sequence are fed to a transformer to enable information sharing between neighbouring actions. A domain classifier encourages the network to produce domain agnostic features. Finally, we use the language model to refine the sequence predictions.

We evaluate our approach on the EPIC-Kitchens Unsupervised Domain Adaptation challenge [2], showcasing the effectiveness of exploiting sequential information in addressing the domain gap.

2. Our Approach

In this section, we describe our method in details. We introduce our approach to model temporal dependencies between actions and to adapt action sequences across different domains. Finally, we refine the predictions of our model using a language model and we reduce the likelihood of actions not seen during training. Figure 2 presents the different components of our method.

2.1. Mixed Sequence Generation

The main objective of our method is to augment action sequences from the source domain with samples from the target sharing the same verb and noun labels. As labels are not available in the target domain, we assign *pseudo-labels* to the target data, based on a confidence threshold λ. For each target sample, we define the confidence of the model predictions as $(p_v + p_n)/2$, where p_v and p_n represent the probabilities of the top verb and noun predictions respectively. Samples whose prediction confidence falls below the threshold are not considered in the mixing process.

Given a sequence consisting of w actions, our goal is to predict the central action i within the temporal window $(i - w/2 < i < i + w/2)$, while exploiting the context provided by both preceding and subsequent actions.

To incorporate target domain information, we introduce a domain adaptation approach for replacing action clips from the source domain with action clips from target distribution representing the same action. More specifically, for each sequence of actions $\mathcal{S}_i = \{s_{-w/2}, \ldots, s_0, \ldots s_{w/2}\}$ from the source domain, we randomly select one or more item (s_i) - excluding the central action - and replace them with an equivalent t_i sampled from the target domain, with the constraint that t_i pseudo-labels match verb and noun labels of s_i. The model is trained to predict the verb and noun labels of the central action of the mixed sequence $\tilde{\mathcal{S}}_i = \{s_{-w/2}, ..., t_i, ..., s_{w/2}\}$.

2.2. Sequence Predictor

To integrate sequences in the action recognition process, we take inspiration from [5] to model temporal relations within the actions window using a transformer model.

Positional Embedding In the first step, we concatenate input features of different modalities and project the output to a lower dimension D. Then, we apply learnt positional embedding to tag the position of each action in the temporal window. Finally, we also append to the sequence two separate tokens for verb and noun prediction of central action. The resulting sequence is indicated by X_e.

Transformers Encoding Once temporal consistency is encoded in positional embeddings, the resulting sequence is fed to a transformer encoder $f(.)$. The attention layers in the transformer allow actions inside the sequence to attend to all the surrounding actions, possibly exchanging relevant clues for the classification of the central action. This operation can be expressed as $Z = f(X_e)$, where Z represents the output of the transformer module, and X_e is the output of positional embedding.

Action Classifier Finally, samples in the sequence are classified using two heads, to predict the verb and noun labels separately. As in [5], we define two classification losses. The primary loss function is defined as the cross entropy of the central action of the sequence, computed separately for verb and noun predictions. Then, to further encourage the network to model temporal relations between all neighbouring actions in the sequence, we also ask the model to predict the labels for all the actions in the sequence through standard cross entropy loss, which we refer to as the mixed sequence prediction loss ($\mathcal{L}_{MS}$).

To encourage the network to learn domain agnostic representations, we also include a domain adversarial module with Gradient Reversal Layer [4] to produce a domain classification loss $\mathcal{L}_{DC}$ for all the actions in the sequence. The resulting loss is thus determined by the summation of the domain discriminator loss $\mathcal{L}_{DC}$ and the sequence prediction loss $\mathcal{L}_{MS}$.

2.3. Language Model

Inspired by MTCN [5], we introduce a Language Model (LM) to filter out unlikely action sequences. We adopt the Masked Language Model (MLM) approach as in [5], in which the model is trained to predict the masked actions within a sequence from the source data, thus encouraging the model to derive high-level dependencies between the actions that make up the sequence. At inference time, to integrate the LM into our framework, we generate all possible sequences of size w using the top-k predictions of the mixed sequence model for each action and select the most probable sequence according to the LM. The final prediction of the model is a linear combination of the predictions from the mixed sequence y_{MS} and the output of the language model for the most probable sequence y_{LM}:

$$y_{final} = (1 - \beta)y_{MS} + \beta y_{LM}. \qquad (1)$$

2.4. Co-Occurrence

To further refine the model predictions, we compute a co-occurrence matrix $\mathcal{M}_{CO}$ of verbs and nouns labels in the source domain, as introduced in [1]. Each entry $\mathcal{M}_{CO_{i,j}}$ stores the occurrences of verb class i and noun class j together in the source data. At test time, the probability of predicting verb and noun pairs which are not present in the source data reduces by a factor of 0.01, i.e. we use the simple assumption that actions not present in the source data are unlikely to be found in the target data.

3. Experiments

3.1. Implementation Details

Sequence Predictor Architecture. To train the mixed sequence model, we use the pre-extracted Temporal Binding Network (TBN) [6] features from the Unsupervised Domain Adaptation (UDA) splits of the EPIC-Kitchens dataset. These features were available for three modalities: RGB, Flow, and Audio. Each modality comprises 25 clips with features size 1024 and we use TRN [8] to temporally aggregate the sample clips.

The model is trained for 100 epochs using a SGD optimizer, with learning rate 0.005. Pseudo-labels for the target data are generated with a confidence threshold of λ = 0.75. During each epoch, we mix actions between the source and target domains. Specifically, we replace one action from the source domain with a corresponding action from the target domain using a temporal window of length 5.

Language Model. The language model is trained for an additional 100 epochs using the Adam optimizer, with a loss value set to 0.001, and fine-tuned on the source training set labels. The training process incorporates a temporal window (w) of size 5. The weight coefficient (β) that combines the predictions of the language model and sequence predictor is set to 0.25. To score sequences, the top-5 actions predicted by the model are taken into account.

3.2. Results

Table 1 presents an ablation study of our method, using three modalities and TBN features. The sequence attention mechanism from MTCN [5] enhances the accuracy compared to the baseline, highlighting the significance of temporal context reasoning even in the presence of domain shift. Furthermore, the inclusion of target and source mixing further boosts the accuracy. Additionally, the introduction of a domain classifier aids in better integration of target information into our model, resulting in further improvement. Subsequently, we apply to filter improbable predictions using the language model, leading to enhanced action accuracy, and demonstrate the role of temporal context in alleviating the domain gap. Lastly, the inclusion of the co-occurrence matrix yields a final improvement to the model's performance, especially on the *action* metric as it weakens the probability of the unlikely verb and noun pairs.

3.3. Ablations

Sequence length. Table 2 shows the effect of the number of actions w in the temporal window on verb and noun classification accuracy, without mixing actions between source and target. Best results on action category are obtained by using a sequence of length $w = 5$.

Number of replacements. Table 3 shows the effect of replacing one or more actions in the sequence with the target before the sequence is sent to the transformer. Improvements in accuracy on all metrics are observed when at least one sample is substituted, while the substitution of more samples only improves on individual categories.

3.4. Model Ensemble

For the final submission, we ensemble our technique with different backbones, using SlowFast [3] and Temporal Shift Module [7] trained on EPIC-Kitchens-55 with ResNet50. The performances of the individual backbones and the ensemble are presented in Table 4. Our approach is visible on the official leaderboard and shown in Table 5, along with the top five teams' performance.

4. Conclusion

In conclusion, this report presents our sequential approach to the EPIC-Kitchens-100 unsupervised domain adaptation challenge. We propose a mixed sequence strategy to improve the transferability of our model across domains. Furthermore, we leverage a language model and

Sequence len > 1	$\mathcal{L}_{MS}$	$\mathcal{L}_{DC}$	LM	$\mathcal{M}_{CO}$	Verb	Noun	Action
✗	✗	✗	✗	✗	51.17	28.88	20.93
✓	✗	✗	✗	✗	51.27	30.48	21.59
✓	✓	✗	✗	✗	53.22	32.68	23.40
✓	✓	✓	✗	✗	54.30	33.31	24.58
✓	✓	✓	✓	✗	**55.22**	34.07	24.99
✓	✓	✓	✓	✓	54.95	**34.15**	**25.46**

Table 1. Ablation study of the different components of our architecture on TBN features. The table shows the Top-1 accuracies starting from the baseline, e.g. each action sample is classified individually, and adding sequence predictions with a temporal window of 5 actions, mixing with the target data ($\mathcal{L}_{MS}$), adversarial alignment using a domain classifier ($\mathcal{L}_{DC}$), predictions refinement using a language model (LM) and pruning of the unlikely verb and noun pairs using a Matrix of Co-Occurrences ($\mathcal{M}_{CO}$).

Sequence length	Verb	Noun	Action
1	51.17	28.88	20.93
3	50.97	30.23	21.18
5	51.27	**30.48**	**21.59**
9	**51.31**	30.34	21.41

Table 2. Top-1 accuracy using different numbers of actions in the temporal window w on the EPIC-Kitchens-100 validation set.

# Replacements	Verb	Noun	Action
0	51.27	30.48	21.59
1	53.22	32.68	23.40
2	53.13	32.07	**23.73**
3	**53.64**	**32.70**	23.59

Table 3. Top-1 accuracy using a different number of target replacements results within a sequence of $w = 5$ actions. Results reported on the EPIC-Kitchens-100 validation set.

Backbone	Verb	Noun	Action
TBN [6]	54.30	33.31	24.58
TSM [7]	54.13	33.25	24.53
SlowFast [3]	54.44	30.74	23.38
Ensemble(E)	56.97	35.64	26.50
E + LM	**57.46**	**36.44**	27.25
E + LM + $\mathcal{M}_{CO}$	57.24	36.42	**27.63**

Table 4. Top-1 accuracy using our domain adaptation method with different backbones on the EPIC-Kitchens-100 validation set. LM: Language Model. $\mathcal{M}_{CO}$: Matrix of Co-Occurrences.

co-occurrence matrix to integrate contextual information by filtering out improbable combinations of verbs and nouns. Through these techniques, our model achieves significant performance improvements.

Rank	Method	Verb	Noun	Action
1	Ns-LLM	**58.22**	40.33	**30.14**
2	VI-I2R	57.89	40.07	30.12
3	Audio-Adaptive-CVPR2022	52.95	**42.26**	28.06
4	**sshayan**	58.11	35.89	27.72
5	plnet	55.51	35.86	25.25

Table 5. Top-1 accuracy on the official leaderboard of the UDA challenge EPIC-Kitchens-100. Our submission is highligthed.

References

[1] Yi Cheng, Fen Fang, and Ying Sun. Team vi-i2r technical report on epic-kitchens-100 unsupervised domain adaptation challenge for action recognition 2021. *arXiv preprint arXiv:2206.02573*, 2022. 1, 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)*, 130:33–55, 2022. 2

[3] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. Pyslowfast. https://github.com/facebookresearch/slowfast, 2020. 3, 4

[4] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. Proceedings of Machine Learning Research. PMLR, 2015. 2

[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, 2, 3

[6] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 3, 4

[7] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7083–7093, 2019. 3, 4

[8] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proceedings of the European conference on computer vision (ECCV)*, pages 803–818, 2018. 3