Supplementary Material for "Domain Generalization using Action Sequences for Egocentric Action Recognition"

Amirshayan Nasirimajd[a,b], Chiara Plizzari[a,c], Simone Alberto Peirone[a,c], Marco Ciccone[a,c], Giuseppe Averta[a,c], Barbara Caputo[a,c]

[a] *Politecnico di Torino, Department of Control and Computer Engineering, Corso Castelfidardo, 34/d, Turin, 10138, TO, Italy*
[b] *name.surname@studenti.polito.it*
[c] *name.surname@polito.it*

Abstract

This document supplements the paper "Domain Generalization using Action Sequences for Egocentric Action Recognition" with additional experimental results and more in-depth analysis of the SeqDG architecture. Section 1 presents additional experiments regarding the SeqMix component, the integration of all input modalities, and the time complexity of SeqDG. Section 2 more details about the implementation of the method are presented. Finally, Section 3 discusses the limitations of SeqDG and how these could be addressed in future works.

1. Additional Experiments

1.1. SeqMix Effect

We show in Fig 1 the effect of SeqMix on the features space of the model in different stages of our framework. In the baseline (left), we observe a separation between the two domains as features are clustered according to their domain. Mixing actions (right) from different domains through SeqMix effectively results in a better alignment between the features from the source and target domains, making them more domain-agnostic and thus improving generalization.

Fig. 2 provides additional insights into the benefits of SeqMix by focusing on the top five most frequent classes and illustrating both per-class and per-domain clusters. SeqMix improves inter-class separability on target data, leading to better generalization performance. This is because mixing samples from different domains helps the network become less biased by the domain and focus more on the action itself, thereby learning better decision boundaries.

1.2. Time Complexity

We present an additional analysis on training and inference time complexity to better understand the model's ability to perform in real-time situations.

1.2.1. Latency

SeqDG requires both past and future actions to provide sufficient context to classify the middle action of the sequence. Although this approach works for offline action classification, it is not optimal for real-time applications because it incurs inherent latency as the system has to wait for the entire sequence to be available. However, SeqDG can be easily adapted to real-time

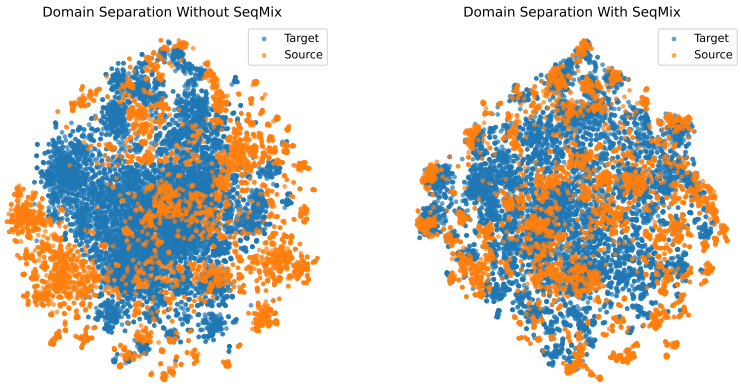

Figure 1: Features without SeqMix (left), and features with SeqMix (right). When using SeqMix, the source and target domains are more aligned (less distinguishable) in the feature space, which translates to improved generalization.

Figure 2: t-SNE plots of the features space for the top five most frequent (*verb*, *noun*) pairs. The plots on the left show the feature space without SeqMix, separately for each class and domain. The plots on the right show the feature space after the application of SeqMix, showing a better separability of action classes and more aligned representations across different domains.

applications by using only the previous actions in the sequence, which eliminates the wait time for future actions and thereby reduces latency. We observe consistent improvements w.r.t. the *Source Only* baseline, as shown in Table 1, with a slight drop compared to classification on the middle action. This results in a good trade-off between better classification accuracy achieved by looking at future actions and lower latency.

1.2.2. Time and memory complexities

We present a detailed analysis of training and inference times in Table 2 and Table 3. Table 2 illustrates the effect of the sequence length on training and inference times, as well as the total number of network parameters. Our results show that using sequences introduces some overhead compared to the baseline without sequences, both in terms of latency and model complexity.

Table 1: Comparison of SeqDG performance on EPIC-KITCHENS-100, depending on the classification target, i.e., middle or last action in the sequence.

Method	CLS Target	Modalities			Top-1 Accuracy (%)			Top-5 Accuracy (%)		
		RGB	Flow	Audio	Verb	Noun	Action	Verb	Noun	Action
Source Only	-	✓	✗	✗	33.5	21.6	11.6	70.2	41.6	33.7
SeqDG	Last	✓	✗	✗	33.5	22.4	12.1 (+0.5)	**73.1**	43.9	36.5 (+2.8)
SeqDG	Middle	✓	✗	✗	**34.3**	**24.2**	**12.8** (+1.2)	72.6	**44.9**	**36.8**(+3.1)
Source Only	-	✓	✓	✓	46.4	26.6	18.2	76.8	51.9	42.2
SeqDG	Last	✓	✓	✓	47.6	28.5	19.6 (+1.4)	78.3	49.9	43.3 (+1.1)
SeqDG	Middle	✓	✓	✓	**49.1**	**29.8**	**20.6** (+2.4)	**79.7**	**52.6**	**45.8** (+3.6)

Table 2: A comparison of the average training time and inference time for different sequence lengths of our method versus the baseline. Although SeqDG introduces some overhead in terms of training time and model size compared to the baseline, at inference time the impact is much more limited as only the visual encoder is used.

Methodology	Sequence Length	Training Time		Inference Time	
		Avg Time/Sample	N. Parameters	Avg Time/Sample	N. Parameters
Source Only	**NoSeq**	10 ***ms***	15.86 M	3.3 ***ms***	15.86 M
Source Only	3	15 ***ms***	19.39 M	4.0 ***ms***	19.39 M
Source Only	5	16 ***ms***	19.40 M	4.1 ***ms***	19.40 M
Source Only	7	17 ***ms***	19.40 M	4.1 ***ms***	19.40 M
SeqDG	3	176 ***ms***	42.75 M	4.0 ***ms***	19.39 M
SeqDG	5	180 ***ms***	42.75 M	4.1 ***ms***	19.40 M
SeqDG	7	196 ***ms***	42.76 M	4.1 ***ms***	19.40 M

However, this overhead is minimal, and increasing the sequence length by a small number of actions does not lead to additional delays. SeqDG incurs some overhead in training time and model size compared to the baseline, but this is only during training since the text and the decoders are not used during inference.

In Table 3, we provide a detailed analysis of SeqDG's components. The results indicate that most of the complexity comes from the visual and text decoders. However, since these decoders are not used during inference, they do not add any additional complexity at that stage.

2. Training and Implementation

2.1. Variation on replacement probability

Figure 3 shows the effect of replacement probability in SeqMix, where the replacement of the central action follows a Bernoulli distribution with a probability of p. Overall, we observe the best performance when p is equal to 0.5. This is because, at low probabilities, the model lacks diversity. However, if the replacement probability increases—therefore with the number of replacements—it prevents the model from effectively learning the coherence and connection between continuous frames within the same domain.

2.2. Modality Processing

We focus our analysis primarily on RGB, as this is the modality most affected by the domain shift. Nonetheless, we include experiments using all the available input modalities on the

Table 3: Ablation study on the effect of different components of SeqDG on training time and number of parameters.

Sequence	Visual Decoder	Text Decoder	Training Time	
			Avg Time/Sample	N. Parameters
-	-	-	10 ms	15.86 M
✓	-	-	16 ms	19.40 M
✓	✓	-	115 ms	28.96 M
✓	-	✓	110 ms	24.33 M
✓	✓	✓	**180 ms**	42.75 M

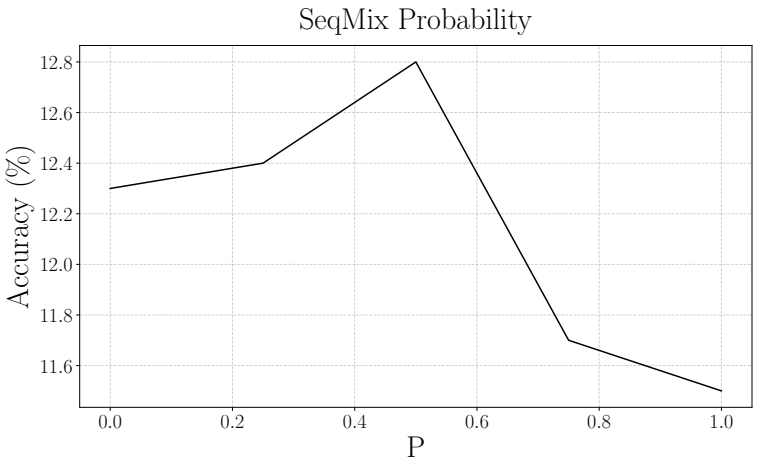

Figure 3: Parameter analysis of the SeqMix replacement Probability p of SeqDG on EPIC-KITCHENS-100 (Action accuracy, RGB-only).

EPIC-KITCHENS-100 dataset to compare with previous state-of-the-art methods and to show the adaptability of SeqDG to modalities other than RGB.

Specifically, we used pre-extracted features for all input modalities using the TBN [1] model. These features are provided as part of the EPIC-KITCHENS-100 dataset[1]. The TBN model [1] processes raw audio signals after transforming them into high-level features suitable for integration into the network. Authors of [1] extract 1.28 seconds of audio from untrimmed videos, convert it to single-channel, and resample it to 24kHz. Using a short-time Fourier transform (STFT) with a window length of 10ms, a hop length of 5ms, and 256 frequency bands, they create a 2D log-spectrogram matrix of size 256×256. The TBN model then utilizes convolutional neural networks (CNNs) to extract relevant audio features, capturing temporal and spectral characteristics essential for action recognition tasks.

We concatenate these modality-specific features into a single feature vector, which is then fed into a transformer encoder. Transformers have demonstrated efficacy in handling multiple modalities, as evidenced by their performance in the MTCN method [2]. The transformer encoder's self-attention mechanism effectively captures dependencies across different modalities, enhancing the overall performance of our method.

[1] https://github.com/epic-kitchens/C4-UDA-for-Action-Recognition?tab=readme-ov-file

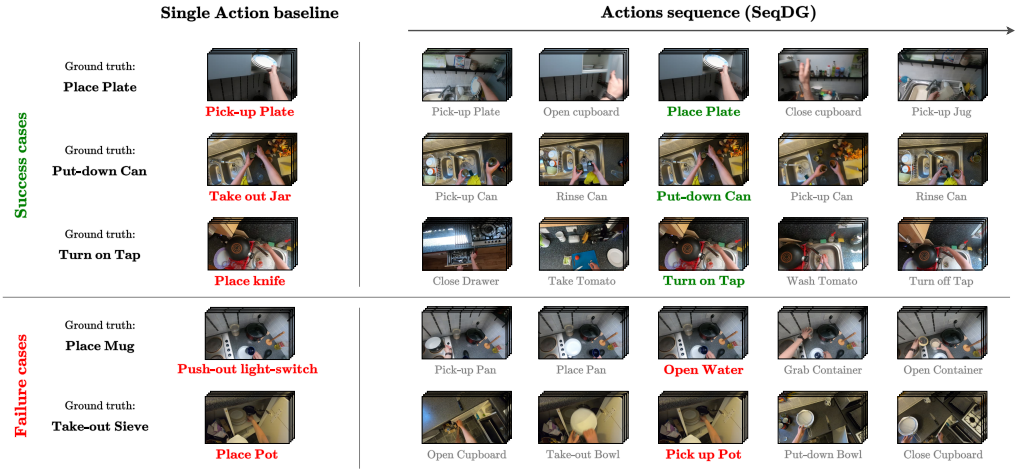

Figure 4: Qualitative examples showing success and failure cases of SeqDG.

3. Limitations and future works

We show in Figure 4 some qualitative examples where a baseline model not using sequences would fail, while our approach is able to predict the correct action, by leveraging the temporal context provided by the surrounding actions. We also present some failure cases where the central action of the sequence is weakly related to the surrounding actions, e.g. when taking out a set of objects from the cupboard, resulting in an incorrect prediction. Consistency in action sequences is a key aspect of our method, assuming human activities are procedural and interconnected. This works well in scenarios with predictable sequences but is less effective in dynamic environments with isolated actions. Future work could focus on identifying action patterns of varying lengths that generalize well across different scenarios while distinguishing user-specific behaviors. This would enhance our model's adaptability to more dynamic environments.

References

[1] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen, "Epic-fusion: Audio-visual temporal binding for egocentric action recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 5492–5501.

[2] E. Kazakos, J. Huh, A. Nagrani, A. Zisserman, and D. Damen, "With a little help from my temporal context: Multimodal egocentric action recognition," in *BMVC*, 2021.