Dual-Head Contrastive Domain Adaptation for Video Action Recognition

Victor G. Turrisi da Costa[1], Giacomo Zara[1], Paolo Rota[1], Thiago Oliveira-Santos[2],
Nicu Sebe[1], Vittorio Murino[3,4] and Elisa Ricci[1,5]
[1]University of Trento, [2]Universidade Federal do Espírito Santo
[3]University of Verona, [4]Huawei Technologies, Ireland Research Center, [5]Fondazione Bruno Kessler
{vg.turrisidacosta,giacomo.zara,paolo.rota,niculae.sebe,e.ricci}@unitn.it
todsantos@inf.ufes.br, vittorio.murino@iit.it

1. The Mixamo dataset

This Section describes additional details and statistics about our Mixamo dataset. In Table 1, we report the number of videos and frames for each class in our dataset. For comparison purposes, we also report the same information for the corresponding Kinetics subset. Figures 1 and 2 provides a visual overview of the distribution of the number of frames and the number of videos across the two datasets.

Table 1: Number of videos and frames in Mixamo and Kinetics

Class	# videos		# frames	
	Mixamo	Kinetics	Mixamo	Kinetics
backflip	959	844	83,717	51,879
breakdancing	2304	829	238,464	63,613
capoeira	3456	940	326,304	75,102
clapping	1344	934	175,488	74,469
golf putting	1037	650	100,370	55,567
jogging	2304	719	143,424	60,808
punching	2016	577	108,784	52,114
salsa dancing	960	517	326,880	42,544
shouting	1248	680	148,224	52,407
side kick	2304	970	142,272	73,998
squat	2081	888	265,833	74,906
swing dancing	1304	750	599,055	64,723
texting	1296	548	432,000	48,690
throwing	1920	1816	221,760	155,869

Furthermore, the dataset presents a rich internal variability within each action class. That is, each class is divided into a number of sub-classes, each associated to a different way of performing the same action. For instance, the *jogging* class includes 8 sub-actions, which consists of unique animations: *jog forward*, *jogging with box*, *jog forward diagonal*, *injured jog*, *slow jog*, *jogging*, *jog in circle*, *jogging stumble*. Figure 3 shows the number of unique sub-classes for each one of the original 14 categories in our synthetic dataset.

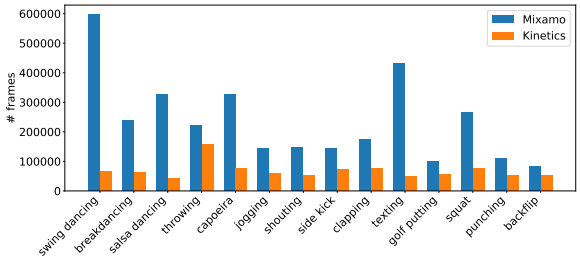


Figure 1: Distribution of frames per class across Mixamo and Kinetics

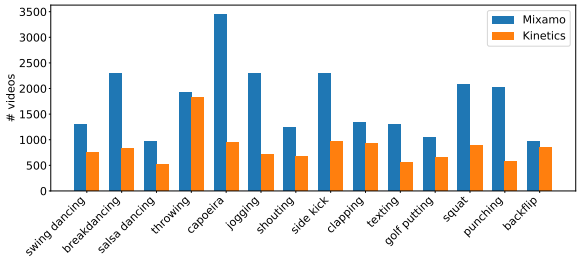


Figure 2: Distribution of videos per class across Mixamo and Kinetics

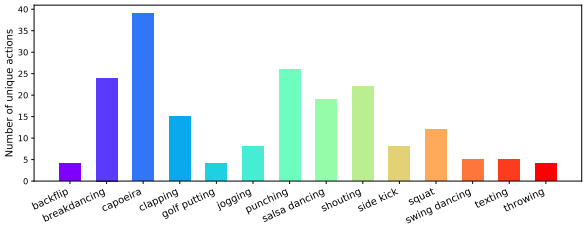


Figure 3: Distribution of unique actions per class

Table 2: Ablation of single-head versus multi-head on *HMDB↔UCF*

Method	**H→U**	**U→H**
CO^{2}A dual-head w/o $\mathcal{L}_{\mathcal{ST}}$	94.4	82.4
CO^{2}A single-head w/o $\mathcal{L}_{\mathcal{ST}}$	91.4	82.9
CO^{2}A (full)	**95.8**	**87.8**

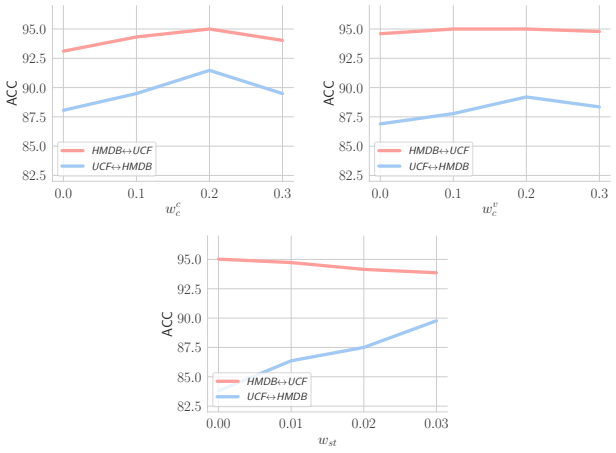


Figure 4: Sensitivity analysis of the weights of the losses $\mathcal{L}_{\mathcal{C}_c}$, $\mathcal{L}_{\mathcal{C}_v}$ and $\mathcal{L}_{\mathcal{SC}}$.

2. Single-head versus dual-head ablation

For completeness, we ablate the usage of the proposed dual-head architecture. Although it is not possible to apply $\mathcal{L}_{\mathcal{ST}}$ without the two heads, it is possible to use the other losses in a single-head architecture. By doing so, the contrastive losses can directly influence the classification head, which could lead to better feature representations. However, in Table 2, we show that, by doing so, the contrastive losses are detrimental for the performance of the model. Although on *UCF→HMDB* the performance of a single-head model is slightly better than the dual-head model, in the other direction, *HMDB→UCF*, there is a large drop in accuracy. Lastly, the single-head approach is greatly outperformed by the full model.

3. Additional sensitivity analysis

In Figure 4, we also ablated the sensitivity of our method to different weights considering the losses $\mathcal{L}_{\mathcal{C}_c}$ and $\mathcal{L}_{\mathcal{C}_v}$. Note that we decoupled w_c into w_c^c for the clip-level loss and w_c^v for the video-level loss. First, considering $\mathcal{L}_{\mathcal{C}_c}$, we can see that our method achieves the best performance for a value of the weight equal to 0.2. This indicates a trade-off between a condition in which the loss has enough weight to guide representation learning at the clip-level and a condition where it dominates other losses. A very similar behaviour is observed for $\mathcal{L}_{\mathcal{C}_v}$, even if the impact of this loss on *HMDB→UCF* is less pronounced. Nonetheless, for the other direction, where the domains are quite different, using a good value for the weight of $\mathcal{L}_{\mathcal{C}_v}$ results in an accuracy of around 2% higher.

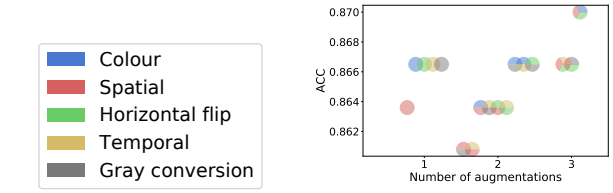
4. Augmentation details

Video-based augmentations were applied similarly to [1]. More specifically, frame-wise augmentations are performed keeping time consistency, *i.e.*, for each video, the parameters for the augmentations are randomised once, and then applied to all frames equally. Colour, spatial and random horizontal flip augmentations were applied only to the target data. The colour augmentation parameters for *torchvision* were 0.15 for the brightness, contrast and saturation, and 0.05 for hue. Spatial augmentation was performed by resizing the image to 256 by 256 and randomly cropping it to be of size 224 by 224 (using the default parameters). In *Mixamo→Kinetics* we also applied a temporal augmentation, which simply samples the total amount of frames, orders and then divides them into the K clips. For source data, we applied the same augmentations only in *Mixamo→Kinetics*. In settings where no augmentations were applied, images are simply resized to 256 and are centrally cropped with a size of 224. Horizontal flip is applied with a 50% probability.

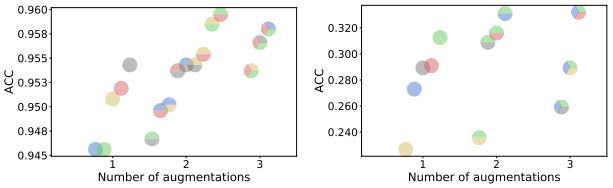
For completeness, we also ablated our method with different combinations of augmentations on *HMDB↔UCF* and *Kinetics→NEC-Drone* in Figure 5. In Figure 5 (a) we can observe that our method is not sensitive to the choice of the augmentations and different combinations of augmentations achieve very similar performance. However, In Figure 5 (b), we can see that the combination of *colour* + *spatial* + *horizontal* outperforms other configurations. Lastly, in the more challenging setting of *Kinetics→NEC-Drone* (Figure 5 (c)), we can see that the choice of augmentations is more important, with *colour* + *horizontal* and *colour* + *spatial* + *horizontal* performing very similarly. Because *colour* + *spatial* + *horizontal* performs best on *UCF→HMDB* and *Kinetics→NEC-Drone* and is only slightly inferior to the best combination on *HMDB→UCF* we selected it as a good default combination across these datasets.

5. Visualisation of the learned representations

In Figure 6 we visualise the features before the linear classifier on the test data for *HMDB↔UCF* when considering a source only model and CO2A. First, considering *HMDB→UCF*, our model produces more compact clusters when considering the majority of classes. Also, it is able to better separate some classes, e.g., *golf* from *shoot bow* and *pull-up* from *fencing*. On *UCF→HMDB* the classes' clusters are more compact, but we can also observe that some clusters became even more compact, e.g., *punch* and *kick ball*. Likewise, it better-separated *fencing* and *shoot bow*. On both directions, we observe that the circles and crosses

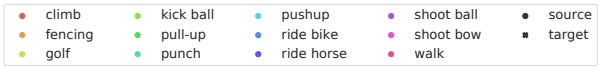


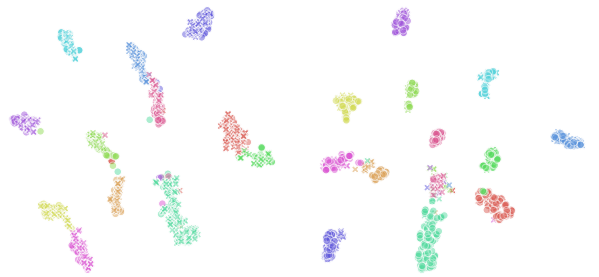
(a) *HMDB→UCF*



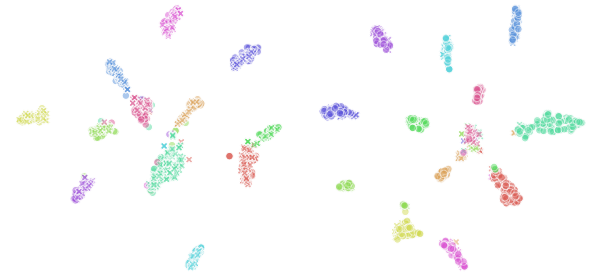
(b) *UCF→HMDB* (c) *Kinetics→NEC-Drone*

Figure 5: Ablation of different augmentations using CO^{2}A on *HMDB↔UCF* and *Kinetics→NEC-Drone*.





(a) Source only (*HMDB→UCF*) (b) Source only (*UCF→HMDB*)



(c) CO^{2}A (*HMDB→UCF*) (d) CO^{2}A (*UCF→HMDB*)

Figure 6: t-SNE plots of test data on *HMDB↔UCF* for a source only model versus CO2A.

(source and target domains) have more overlap, indicating that the adaptation procedure better aligns both domains.

References

[1] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *arXiv preprint arXiv:2008.03800*, 2020.