AutoLabel: CLIP-based framework for Open-set Video Domain Adaptation

Supplementary Material

Giacomo Zara[1], Subhankar Roy[3], Paolo Rota[1], Elisa Ricci[1,2]
[1]University of Trento, Italy [2]Fondazione Bruno Kessler, Italy
[3]LTCI, Télécom Paris, Institut polytechnique de Paris, France
{giacomo.zara,paolo.rota,e.ricci}@unitn.it, subhankar.roy@telecom-paris.fr

The supplementary material is organized as follows: in Section A we provide further details on our fine-tuning process on the target domain. In Section B we describe in depth the pipeline for attributes extraction and matching. In Section C we provide the pseudo-code for the most relevant routines of our AutoLabel framework. In Section D we provide useful statistics of our considered benchmarks. In Section E we provide a more detailed description of the baseline methods included in our experimental evaluation. In Section F we report additional results and ablation study experiments.

A. Target fine-tuning

In this Section we provide details of how the target pseudo-labelling and consequent fine-tuning steps are carried out in our AutoLabel framework. ActionCLIP [8] performs inference by projecting the test video to the CLIP space, and assigning the label corresponding to the textual prompts whose embedding is the most similar to the video embedding. In order to fine-tune on the target domain, we freeze the network and apply such inference step to the unlabelled target training batch, obtaining a pseudo-label for each target instance. After that, we filter out all those predictions that are not included in the top-$k\%$ most confident ones for that specific label. In order to measure the confidence of a given pseudo-label, we consider the similarity in the CLIP space with the closest set of textual prompts. On the instances of the target training batch passing the filtering process, we simply carry out a standard supervised training step with the ActionCLIP loss.

B. Attributes extraction and matching

In this Section we detail the implementation of the attributes extraction and matching pipeline mentioned in the main paper. In particular, we provide the formal details of the `tfidf` and $sim(\cdot,\cdot)$ functions from Sections 3.2.2 and 3.2.3, respectively. As mentioned in the paper, the off-the-shelf image captioning model ViLT [6] extracts a set of attributes from a selection of frames in a given input video sequence. For our experiments, we set up the model in order to extract 5 attributes for each of 5 frames out of each video sequence. After that, we select the 5 most frequent attributes across the frame selection in order to build the final set of attributes for the sequence.

B.1. `tfidf` function

After the extraction carried out by ViLT, we are provided with a set of attributes for each video sequence. The following steps demand the extraction of a set of attributes for a given class (source domain) and for a given video cluster (target domain). In both cases the pipeline is the same, and we only change the set of instances given as input. Given the sets of instances, we apply the `tfidf` module, which is implemented as follows. We firstly compute the most frequent attributes across all the input instances; at this point, we compute the *Term-frequency and Inverse Document Frequency (tf-idf)* [4, 7] score of each attribute. Given a set of text documents and the corresponding token vocabulary, the *tf-idf* value of a given token with respect of a given document is designed in order to quantify how relevant that token is for that document. Formally, this score is defined as the product of two different statistics, namely *Term frequency (tf)* and *Inverse document frequency (idf)*, defined as follows:

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{1}$$

$$idf(t,D) = \log \frac{N}{|\{d \in D : t \in d\}} \tag{2}$$

$$tfidf(t,d,D) = tf(t,d) \cdot idf(t,D) \tag{3}$$

where $f_{t,d}$ is the raw count of of term t in the document d, i.e., the number of occurrences of term t in d. N is the total number of documents considered, and D is the set of documents.

The conceptual intuition behind this score lies in the fact that a given term is most likely to be relevant for a given document if (i) it occurs often in the document and (ii) it occurs seldom in any other document. In our case, we consider one document for each class ($N = K$, source domain) or for each cluster ($N = C$, target domain): each document comprises the most common attributes across that specific set of instances. Consequently, by thresholding the *tf-idf* of each attribute, we end up with the most relevant terms for each source class and for each target cluster. We set this threshold to 0.5 for all experiments.

B.2. Matching

Once provided with a set of relevant terms for each source class and for each target cluster, we carry out the matching step as described in the main document. This step relies on the $sim(\cdot,\cdot)$ function, which takes as input two sets of attributes, ordered by confidence during the extraction, and computes a similarity score. Given a_s attributes in the source set , this function defines a_s weights in decreasing order normalized between 0 and 1. For each common occurrences, it adds up to the score value (initially 0) the weight corresponding the absolute distance between the positions of such occurrences in the two input sets. Informally, the intuition behind this score consists in the idea of taking into account both the number of co-occurrences of attributes between the two sets and their position, the latter accounting for their frequency in the corresponding input set. At the end of this loop, the score is again normalized and used to fill the S matrix mentioned in Sec. 3.2.3.

C. Pseudo-code

In this Section we provide the pseudo-code for the different modules of our framework. In particular, Alg. 1 presents the attribute extraction step presented in Section 3.2.1 of the main document; Alg. 2 presents the discovery process of candidate target classes presented in 3.2.2, Alg. 3 details the similarity function $sim(\cdot,\cdot)$ referenced in 3.2.3 and Alg. 4 provides the pipeline for the attribute matching process described in 3.2.3.

D. Datasets statistics

In this Section we provide detailed statistics about the benchmarks considered in our experimental evaluation. In particular, we report for each dataset in Table 1 the number of shared and private classes and the number of source training, target training and test samples.

E. Baseline details

In this Section we provide additional details about the baseline methods we implemented autonomously.

CEVT-CLIP [1] This baseline has been implemented by simply modifying the original code provided by the authors in [1], replacing the ResNet [5] backbone with the ActionCLIP [8] encoder.

ActionCLIP [8] This baseline is obtained by modifying our own framework, itself based on the ActionCLIP architecture, in order to apply a different open-set rejection protocol. In order to make the choice of whether to assign a

Algorithm 1: Attribute extraction

```
Input: Source video sequences X^S,
       Target video sequences X^T, Prompt z,
       ViLT model ViLT(),
       Number of selected frames F, Number of
       selected attributes k, tf-idf module tfidf
Output: Set of source attributes Λ̄^{l^S},
        Set of target attributes Λ̄^T

for X ∈ {X^S, X^T}, d ∈ {l^S, T} do
    for i ← 0 to |X| do
        x ← X[i]
        video_attributes ← [ ]
        for j ∈ F do
            A(x_j) ← ViLT(x_j, z)
            Append attributes in A(x_j) to
              video_attributes
        end
    end
    mc ← argtop_k(video_attributes)
    filtered ← tfidf(mc)
    Add attributes in filtered to Λ̄^d
end
```

Algorithm 2: Discovering candidate classes

```
Input: Target video sequences X^T, Video encoder
       G_V, Number of target clusters |C|,
       Set of target attributes Λ̄^T, Clustering
       function Cluster
Output: Target candidate labels Y^{cand,T}

v^T ← G_V(X^T)
C ← Cluster(v^T)
Y^{cand,T} ← ∅
for c ← 0 to |C| do
    Λ̄^{c,T} ←
      Attributes for videos belonging to cluster c
    l_c^{cand,T} = Λ̄_1^{c,T} || ... || Λ̄_t^{c,T}
    Add l_c^{cand,T} to Y^{cand,T}
end
```

known or unknown label to a test target sample, this method simply thresholds the similarity, in the CLIP space, between the video embedding and the closest set of label prompts. This threshold has been set to 0.9 for *HMDB↔UCF* and to 0.5 for *Epic-Kitchens*.

ActionCLIP-ZOC [2] This baseline is implemented as a modification of the open-set rejection protocol of our method: instead of extending the target label set with newly discovered labels extracted by unmatched target clusters, this method extends it for each individual test instance with

Algorithm 3: Similarity function $sim(\cdot, \cdot)$

Input: Set of source attributes $\bar{\Lambda}^{l^{\text{S}}}$,
Set of target attributes $\bar{\Lambda}^{\text{T}}$

Output: Similarity score s

```
/* Compute normalized weights */
```
$\mathbf{ref} \leftarrow reverse(range(len(\bar{\Lambda}^{l_i^{\text{S}}})))$
$\mathbf{w} \leftarrow (\mathbf{ref} - min(\mathbf{ref}))/(max(\mathbf{ref}) - min(\mathbf{ref}))$
```
/* Incrementally compute score */
```
$s \leftarrow 0$
for $i_s \leftarrow 0$ **to** $len(\bar{\Lambda}^{l^{\text{S}}})$ **do**
  for $i_t \leftarrow 0$ **to** $len(\bar{\Lambda}^{\text{T}})$ **do**
    if $\bar{\Lambda}^{l^{\text{S}}}[i_s] = \bar{\Lambda}^{\text{T}}[i_t]$ **then**
      $s \leftarrow s + \mathbf{w}[abs(i_t - i_s)]$
    end
  end
end
$s \leftarrow s/len(\bar{\Lambda}^{l_i^{\text{S}}})$

Algorithm 4: Attribute matching

Input: Target candidate labels $\mathcal{Y}^{\text{cand},\text{T}}$,
Similarity function $sim(\cdot, \cdot)$, Threshold γ,
Number of shared classes K,
Number of target clusters $|\mathcal{C}|$,
Number of tokens t

Output: Target private labels $\mathcal{Y}^{\text{priv},\text{T}}$

$\mathcal{Y}^{\text{priv},\text{T}} \leftarrow \emptyset$
for $i \leftarrow 0$ **to** $|\mathcal{C}|$ **do**
  $\mathbf{match} \leftarrow \mathbf{False}$
  for $j \leftarrow 0$ **to** K **do**
    if $sim(\bar{\Lambda}^{l_j^{\text{S}}}, \bar{\Lambda}^{i,\text{T}}) < \gamma$ **then**
      $\mathbf{match} \leftarrow \mathbf{True}$
    end
  end
  if $\neg\mathbf{match}$ **then**
    $l^{\text{cand},\text{T}} = \mathcal{Y}^{\text{priv},\text{T}}[i]$
    Add $l^{\text{cand},\text{T}}$ to $\mathcal{Y}^{\text{priv},\text{T}}$
  end
end

the names of the objects detected by ViLT [6] in that specific sequence. The detection process is carried out in the same way as in `AutoLabel`.

ActionCLIP-*Oracle* [3] We implement this baseline by extending the label set with the ground truth names of the target private categories. For fair comparison, all hyper-parameters for these baselines match those employed for `AutoLabel` in each setting.

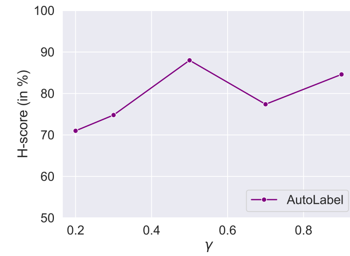

Figure 1. Sensitivity study on the threshold *gamma* for *HMDB→UCF*

F. Additional results

F.1. Detailed *Epic-Kitchens* results

We report in Table 2 the complete results of our method and its competitors on the *Epic-Kitchens* setting, including the **ALL**, **OS***, and **UNK** metrics, omitted in the main document for space issues. it is possible to observe in the complete Table that, especially when compared to the *HMDB↔UCF* case, this benchmark is characterized by a significant instability. In particular, it is evident that, across different methods considered, the **HOS** score is affected by a strong tendency of most methods to either over-accept, resulting in a higher **OS*** score, or over-reject, producing a higher **UNK** score. However, it is possible to observe that the results obtained with our proposed `AutoLabel` method, when compared to most competitors, are characterized by a better balance between **OS*** and **UNK**, indicating a more controlled training process.

F.2. Ablation analysis

We provide in this Section a further ablation analysis omitted from the main document for space issue. In particular, we report a sensitivity score on the matching threshold γ with respect to the reference **HOS** metric, for *HMDB→UCF* (Fig. 1) and for *Epic-Kitchens D1→D2* (Fig. 2). From this study, it emerges that the score consistently oscillates around 80% for *HMDB→UCF* and around 40% for *Epic-Kitchens*.

F.3. Discovered candidate classes

We provide in this Section an overview of the ground-truth and discovered target-private classes for the *HMDB→UCF* and *Epic-Kitchens D1→D2* settings, in Tables 3 and 6, respectively. In the left column of the Tables we report the actual names of the private classes of the target domain, and on the right one we report the names of the candidate target-private labels identified by our proposed `AutoLabel` framework, which are composed by concatenating the most relevant attributes extracted from each cluster that was not matched with any shared class. We can

Dataset	# shared classes	# private classes	# source train samples	# target train samples	# test samples
HMDB	6	6	375	781	337
UCF	6	6	865	1438	571
EK-D1	8	75	1543	2021	625
EK-D2	8	84	2495	3755	885
EK-D3	8	82	3897	5847	1230

Table 1. Statistics of the considered benchmarks for the experimental evaluation

Setting $\rightarrow$	D2→D1				D3→D1				D1→D2			
Method $\downarrow$	**ALL**	**OS***	**UNK**	**HOS**	**ALL**	**OS***	**UNK**	**HOS**	**ALL**	**OS***	**UNK**	**HOS**
CEVT [1]	**30.5**	7.2	**76.8**	13.2	**31.8**	8.1	**76.8**	14.7	18.7	4.5	67.4	8.4
CEVT-CLIP [1]	26.8	7.3	68.9	13.2	24.4	10.0	67.8	17.3	16.6	7.3	71.8	13.3
ActionCLIP [8]	24.6	**32.2**	48.1	31.3	19.5	29.2	27.5	28.3	21.3	25.6	**74.5**	38.1
ZOC [2]	22.0	18.4	43.6	25.9	20.9	29.2	24.7	26.8	**23.6**	24.7	44.4	31.7
`AutoLabel` (ours)	28.5	26.1	52.3	**34.8**	29.6	**30.0**	52.9	**38.3**	23.3	**33.9**	63.1	**44.1**
Oracle [3]	*25.6*	*23.8*	*55.0*	*33.2*	*21.9*	*26.0*	*45.5*	*33.1*	*31.7*	*33.1*	*42.1*	*37.1*
Setting $\rightarrow$	D3→D2				D1→D3				D2→D3			
Method $\downarrow$	**ALL**	**OS***	**UNK**	**HOS**	**ALL**	**OS***	**UNK**	**HOS**	**ALL**	**OS***	**UNK**	**HOS**
CEVT [1]	25.2	8.9	**78.5**	16.0	21.6	4.3	**71.0**	8.1	25.5	6.1	**77.7**	11.3
CEVT-CLIP [1]	21.3	8.0	67.4	14.3	21.5	5.5	69.1	10.2	19.8	5.5	65.2	10.1
ActionCLIP [8]	24.6	35.8	55.2	43.4	26.3	20.4	50.4	29.0	30.6	16.7	44.2	24.2
ZOC [2]	23.8	34.1	52.5	41.3	23.5	21.3	41.0	28.0	**31.1**	**24.1**	22.2	23.1
`AutoLabel` (ours)	25.7	**39.9**	68.4	**50.4**	**29.6**	**28.5**	36.2	**31.9**	27.7	21.1	50.8	**29.8**
Oracle [3]	*16.3*	*31.7*	*75.4*	*44.6*	*21.2*	*17.8*	*37.6*	*24.2*	*28.4*	*18.8*	*62.8*	*28.9*

Table 2. Results of all considered methods for the *Epic-Kitchens* settings. We include in this Table all the open-set metrics, included those omitted from the main document for space issues. Our proposed `AutoLabel` method is shown to achieve the best **HOS** score in all settings by achieving an effective balance of **OS*** and **UNK**.

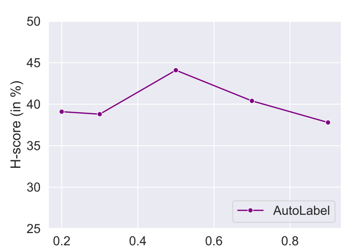

Figure 2. Sensitivity study on the threshold $gamma$ for *Epic-Kitchens D1→D2*

firstly observe that the discovered classes on *HMDB→UCF* show a significant diversity, especially when looking at the first attributes for each candidate label name, which are the most relevant ones. On the other hand, discovered classes on *Epic-Kitchens* appear to be significantly more noisy and generic. As mentioned in the main document, we associate this behavior to the fact that video sequences in each domain of the *Epic-Kitchens* dataset are all constrained to the same kitchen environment, thus characterized by the same (or similar) objects across multiple categories.

F.4. Cluster attributes

We show in Tables 4 and Tables 5, respectively, two examples of the attributes extracted from sample target cluster for the *HMDB→UCF* setting, along with the final target description obtained with the `tfidf` module. It is possible to observe in these tables how the final attributes are able to reduce redundancy and provide an effective description for the candidate unknown class.

F.5. Output visualization

We provide in this Section examples of correct and incorrect predictions of our model, on both shared and target private categories. We include an example for *HMDB→UCF* (Fig. 3) and one for *Epic-Kitchens D1→D2* (Fig. 4). It

Ground truth	Discovered
pushup	water AND horse AND fence
ride bike	rope AND table AND table AND window
ride horse	bike AND street AND car
shoot ball	basketball AND building AND fence
shoot bow	rock AND rope AND window
walk	road AND bike AND car
	sign AND net AND court
	horse AND field AND building
	floor AND chair AND table
	refrigerator AND bed AND door
	field AND dog AND grass
	boxers AND men AND referee
	horse AND building AND fence
	dog AND grass AND fence
	hoop AND basketball AND net
	rack AND door AND mirror
	house AND grass AND building
	soccer AND field AND net
	stick AND grass AND fence

Table 3. List of the actual names of the ground truth target private classes (left) and a selection of candidate target-private label names identified by AutoLabel (right) for the *HMDB→UCF* setting

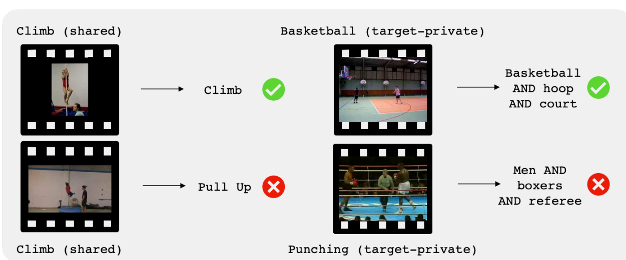

Figure 3. Example of correct and incorrect predictions of AutoLabel on both shared and private categories on the *HMDB→UCF* setting

is possible to observe in Fig. 4 how the high similarity among distinct *Epic-Kitchens* categories easily leads to incorrect prediction on both shared and unknown classes. On the other hand, it emerges from the example in Fig. 3 how the model may fail in correctly classifying sequences from *HMDB↔UCF*, despite its ability to extract a useful description (e.g. see bottom right example in Fig. 3).

Original cluster attributes		Final cluster attributes
horse	fence	horse
fence	person	person
field	people	fence
man	dirt	man
grass	horse	tree
horse	sand	water
zebra	horse	
mountain	person	
bush	fence	
horse	man	
tree	sand	
water	horse	
fence	beach	
person	people	
water	man	
person	hat	
horse	shirt	
bush	sky	
person	horse	
man	mountain	
road	bush	
horse	sand	
zebra	person	
horse	water	
beach	man	
water	fence	
woman	horse	
building	tree	
horse	person	
beach	bunch	
horse	tree	
sand	fence	
hat	person	
water	car	
beach	horse	

Table 4. List of original and final attributes extracted from a sample target cluster for the *HMDB→UCF* setting. We emphasize each of the final attributes in a different color in order to highlight occurrences among the original ones

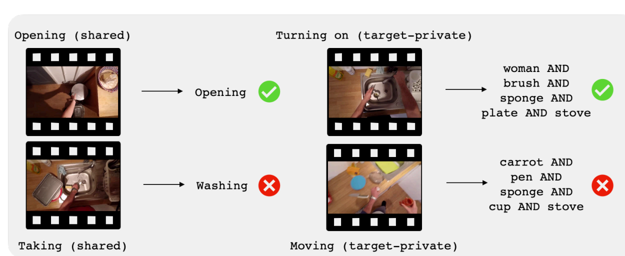

Figure 4. Example of correct and incorrect predictions of AutoLabel on both shared and private categories on the *Epic-Kitchens D1→D2* setting

Original cluster attributes			Final cluster attributes
ball	ball	net	ball
hoop	people	basketball	male
male	sign	ball	basketball
basketball	light	gym	hoop
white	kite	male	net
ball	male	door	court
male	female	rack	
net	ball	ball	
court	rack	female	
hoop	boy	male	
gym	ball	people	
hoop	basketball	basketball	
ball	men	hoop	
male	court	ball	
basketball	net	male	
ball	ball	door	
male	hoop	hoop	
rack	male	ball	
people	basketball	male	
table	white	male	
ball	ball	court	
hoop	net	man	
net	people	hoop	
basket	court	net	
person	basketball	basketball	
hoop	basketball	court	
ball	ball	court	
male	men	men	
gym	net	male	
female	basket	hoop	
ball	ball	basket	
ball	basketball	gym	

Table 5. List of original and final attributes extracted from a sample target cluster for the *HMDB→UCF* setting. We emphasize each of the final attributes in a different color in order to highlight occurrences among the original ones

Ground truth			Discovered
turn-on	shake	compress	glass AND brush AND plate AND cup AND fork
drop	knead	scrape	brush AND sponge AND plate AND cup AND fork
grate	extract	crush	chair AND sponge AND plate AND cup AND fork
throw-into	spread	move around	microwave AND brush AND plate AND cup AND fork
turn	throw	remove from	refrigerator AND sponge AND plate AND cup AND fork
see	set	wrap	woman AND carrot AND plate AND cup AND fork
adjust	hang	gather	phone AND sponge AND plate AND cup AND fork
fold	separate	wrap around	brush AND chair AND plate AND cup AND fork
wait-for	flip	press	brush AND chair AND sponge AND cup AND fork
scoop	eat	wrap with	pizza AND glass AND plate AND cup AND fork
taste	heat	rotate	carrot AND brush AND plate AND cup AND fork
drink	wait	fix	mirror AND microwave AND plate AND cup AND fork
turn-off	check	crack	phone AND chair AND sponge AND plate AND cup
drain	look for	read	glass AND chair AND plate AND cup AND fork
squeeze	sprinkle	split	mirror AND sponge AND plate AND cup AND fork
dry	roll	seal	book AND glass AND brush AND chair AND cup
move	peel	press down	cookie AND brush AND sponge AND plate AND fork
empty	unroll	break	book AND woman AND plate AND cup AND fork
unfold	hold	distribute	glass AND chair AND sponge AND plate AND fork
switch-on	spread onto	serve	refrigerator AND chair AND plate AND cup AND fork
put-in	flatten	pat	
spoon	pull down	throw in	
sprinkle-onto	take out	lower	
put-into	remove	take off	
move-into	lift	throw off	
attach-onto	pat down	grind	
twist-off	immerge	spray	
hand	move onto	tap	

Table 6. List of the actual names of the ground truth target private classes (left) and list of candidate target-private label names identified by `AutoLabel` (right) for the *Epic-Kitchens D1→D2* setting

References

[1] Zhuoxiao Chen, Yadan Luo, and Mahsa Baktashmotlagh. Conditional extreme value theory for open set video domain adaptation. In *ACM Multimedia Asia*. 2021. 2, 4

[2] Sepideh Esmaeilpour, Bing Liu, Eric Robertson, and Lei Shu. Zero-shot out-of-distribution detection based on the pretrained model clip. In *AAAI*, 2022. 2, 4

[3] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 3, 4

[4] Carol Friedman, Thomas C. Rindflesch, and Milton Corn. Natural language processing: State of the art and prospects for significant progress, a workshop sponsored by the national library of medicine. *Journal of Biomedical Informatics*, 2013. 1

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[6] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*, 2021. 1, 3

[7] Katsuya Masuda, Takuya Matsuzaki, and Jun'ichi Tsujii. Semantic search based on the online integration of nlp techniques. *Procedia - Social and Behavioral Sciences*, 2011. 1

[8] Mengmeng Wang, Jiazheng Xing, and Yong Liu. Actionclip: A new paradigm for video action recognition. *arXiv preprint arXiv:2109.08472*, 2021. 1, 2, 4