Supplementary material: Temporal and cross-modal attention for audio-visual zero-shot learning

In the supplementary material, we provide additional details about baselines (Section 1), and present further model ablations (Section 2). Additionally, we study t-SNE visualisations for TCaF and [3] (Section 3), and provide a comparison of the computational complexity of TCaF and some of the baselines (Section 4).

1 Additional details about baselines

In the following, we detail our adaptations of Attention Fusion [1] and of the Perceiver [2] to the (G)ZSL setting (which we briefly summarised in Section 4.2 of our manuscript).

1.1 Attention Fusion

In order to use Attention Fusion [1] in the (G)ZSL setting, we take the same temporal audio and visual features as inputs as TCaF. Following TCaF, we embed the input features into the same feature dimension using A_{enc} and V_{enc}. Instead of directly mapping to the number of classes, as the authors originally proposed, A_{enc} and V_{enc} map the features to $\mathbb{R}^{d_{dim}}$. The embedded features are then temporally averaged to obtain a single $d_{dim}-$dimensional feature vector for each modality. The attention weight α, which is used for fusing both modalities, is computed using the channel-wise concatenation of the audio and visual embeddings through a linear layer $f_{attn} : \mathbb{R}^{2*d_{dim}} \rightarrow \mathbb{R}^{d_{dim}}$, followed by a sigmoid function. Both modalities are then fused to create the output token o_c through $o_c = \alpha \odot \phi_{a,avg} + (1 - \alpha) \odot \phi_{v,avg}$, where $\phi_{a,avg}$ and $\phi_{v,avg}$ are the temporally averaged audio and visual features. o_c is then projected using the same projection function O_{proj}, decoder D_o, and text embedding projections as in TCaF. We train Attention Fusion using the same learning rate and loss functions as TCaF.

1.2 Perceiver

The Perceiver [2] takes the same audio and visual features as input as TCaF. For consistency between frameworks, we again embedded the input features to the same feature dimension using A_{enc} and V_{enc}, and equip both TCaF and the Perceiver with the same temporal and modality information by adding positional embeddings as described in the main paper. Our goal was to directly compare our cross-attention mechanism with the Perceiver attention. Therefore, we adapted the cross-attention, self-attention and dense layer blocks of the Perceiver to use the same internal dimensions as TCaF. We also added a dropout layer at the end of dense layer blocks to match the dense blocks in TCaF. For the randomly initialised latent array, we use 64 latent tokens with dimension $\mathbb{R}^{d_{dim}}$ for all datasets. Increasing the number of latent tokens did not provide a boost in performance, but significantly increased the computational costs. One of the latent tokens is used as the output classification token c_o. We use one cross-attention block and one self-attention block per layer without weight sharing and use the same number of layers as TCaF. This results in just a slightly higher number of parameters for the Perceiver than for our TCaF. The output token c_o is projected using the projection function O_{proj} and the decoder D_o. The computations for the text embeddings are analogous to TCaF. We train the Perceiver using the same learning rate and loss functions as our model.

2 Additional model ablations

In this section, we first study the impact of using temporal embeddings (Section 2.1) and of the number and design of the cross-attention layers in TCaF (Section 2.2). Next, we evaluate the impact on performance when adding noise to the audio modality (Section 2.3). Finally, we present results of transforming TCaF to [3] (Section 2.4).

2.1 Influence of using temporal information

In the following, we investigate the influence of using temporal information when learning multi-modal video representation for (G)ZSL with TCaF. Since the operations in our audio-visual transformer layers (cf. Section 3.2 in the manuscript) are invariant to permutation, the feature tokens are additionally equipped with temporal information through the addition of positional embeddings pos_t. Without temporal embeddings, the model is unable to put data from one time step in temporal relation to information from the other time steps. Temporal embeddings therefore allow the model to understand the concept of time.

Table 1 shows results for training and evaluating TCaF with (+) and without (−) temporal embeddings (pos_t). The highest harmonic mean is achieved when using temporal embeddings. For instance for ActivityNet-GZSL^{cls}, our model that does not use temporal embeddings ($-pos_t$) obtains only a HM of 8.69% and a ZSL score of 5.53%, compared to a HM of 12.20% and a ZSL score of 7.96% when using temporal embeddings. Similar observations can be made for VGGSound-GZSL^{cls} and UCF-GZSL^{cls}, showing the importance of temporal information for learning strong video representations.

Positional embeddings	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	S	U	HM	ZSL	S	U	HM	ZSL	S	U	HM	ZSL
$-pos_t$	**15.78**	4.66	7.19	4.97	27.35	26.02	26.67	28.06	21.80	5.43	8.69	5.53
$+pos_t$ (TCAF)	12.63	**6.72**	**8.77**	**7.41**	**67.14**	**40.83**	**50.78**	**44.64**	**30.12**	**7.65**	**12.20**	**7.96**

Table 1. Influence of temporal information provided through positional embeddings (pos_t) on the (G)ZSL performance on the VGGSound-GZSL^{cls}, UCF-GZSL^{cls}, and ActivityNet-GZSL^{cls} datasets.

Layer configurations	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	S	U	HM	ZSL	S	U	HM	ZSL	S	U	HM	ZSL
1 layer w/o FF	19.70	4.47	7.29	4.66	63.30	26.45	37.31	27.85	15.10	4.59	7.04	4.63
1 layer	**17.95**	4.78	7.55	5.13	40.07	29.40	33.92	29.74	28.22	4.85	8.27	4.89
1/2∗(all layers) w/o FF	11.33	4.25	6.18	4.59	38.72	23.17	28.99	23.28	8.13	3.35	4.75	3.40
1/2∗(all layers)	12.08	4.69	6.75	5.12	**77.19**	30.18	43.40	34.18	28.65	6.04	9.98	6.25
1/2∗(all layers) $+A_{self}$	14.62	4.56	6.96	4.97	53.05	34.83	42.05	35.84	**31.38**	5.93	9.97	6.51
all layers w/o FF	14.41	4.28	6.60	4.59	32.57	25.77	28.78	28.86	7.44	3.27	4.54	3.33
all layers	12.63	**6.72**	**8.77**	**7.41**	67.14	**40.83**	**50.78**	**44.64**	30.12	**7.65**	**12.20**	**7.96**

Table 2. Varying the number of cross-attention layers in TCAF and the use of feed forward (FF) functions in the cross-attention layers.

2.2 Impact of using different amounts of cross-attention layers and of varying the cross-attention layer design

In Table 2, we present ablations on the number of cross-attention layers used in our model. Furthermore, we investigate the relevance of using feed forward functions (FF) in our cross-attention layers.

For TCAF, we used 8 cross-attention layers on VGGSound-GZSL^{cls} (all layers). On the UCF-GZSL^{cls} and ActivityNet-GZSL^{cls} datasets, we used 6 layers (all layers). We observe that using more layers is beneficial for GZSL and ZSL performance across all datasets. Moreover, we observe that, in general, eliminating the feed forward functions leads to a decrease in performance. Finally, using only half of the layers jointly with self-attention (1/2∗(all layers) $+A_{self}$) leads to worse overall HM performance than using half of the layers without self-attention (1/2∗(all layers)). This is in line with the experiments in the main paper, where adding the self-attention leads to worse results.

This ablation shows that using only cross-attention is beneficial even when using a different number of layers. Furthermore, using more cross-attention layers that are equipped with feed forward functions brings a boost in performance.

2.3 Impact of noise in audio stream on GZSL performance

In this section, we study how the GZSL performance (HM) of TCAF decreases when noise is added to increasing temporal portions of the audio signal on all three datasets. We study both TCAF and TCAF $+A_{self}$ in Fig. 1. It can be observed that an increase in the proportion of noise leads to a decrease in the GZSL performance for both models on all three datasets. Furthermore, it can be observed that TCaF is significantly more robust to perturbations on UCF-GZSL^{cls} and slightly more robust on VGGSound-GZSL^{cls}. On the other hand, we can observe that on ActivityNet-GZSL^{cls} the trend is reversed, with TCaF $+A_{self}$ being slightly more robust. Overall, it can be argued that TCaF is more robust across all three datasets than TCaF $+A_{self}$.

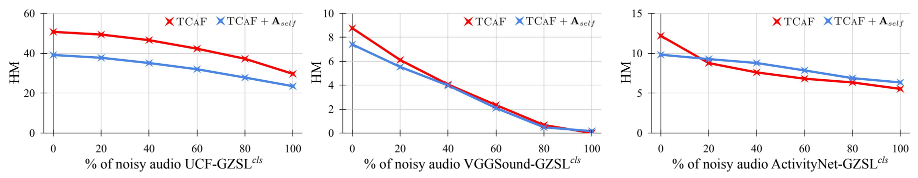

Fig. 1. Robustness of TCaF and TCaF $+A_{self}$ to noise added to different proportions of the audio stream on UCF-GZSL^{cls}, VGGSound-GZSL^{cls} and ActivityNet-GZSL^{cls}.

2.4 Transforming TCaF into [3]

Our TCaF builds on the AVCA [3] framework for audio-visual GZSL. To highlight the benefits of TCaF compared to AVCA, we show results for transforming TCaF into AVCA [3] in Table 3.

TCaF exploits temporal information and obtains a HM performance of 8.77% on VGGSound-GZSL^{cls} compared to a HM of 7.65% (TCaF avg input) when using temporally averaged inputs. Moreover, TCaF uses an enhanced cross-modal attention to effectively gather multi-modal information. On the other hand, the attention mechanism of [3] uses temporally averaged feature inputs, which leads to a HM of 6.82% on VGGSound-GZSL^{cls} ([3]). Additionally, TCaF uses a single output branch and a classification token to aggregate the multi-modal information. In contrast, [3] uses two branches and no classification token which leads to a HM of 6.27% (w/o class. token) on VGGSound-GZSL^{cls}. Finally, our training objective avoids triplet losses, i.e. there is no overhead to train with positive and negative pairs. Using triplet losses similar to those used in [3] leads to a lower performance (TCaF + $l_{triplet}$) than TCaF. The same trend can be observed for the other datasets, proving that our architectural choices are more suitable for the audio-visual (G)ZSL task.

3 t-SNE comparison between TCaF and [3]

We show t-SNE visualisations that highlight the difference between TCaF and [3] in Fig. 2. It can be observed that in the case of [3], the classes overlap more than in the case of TCaF. In particular, this can be observed for the unseen classes. Moreover, for [3], the clusters are less concentrated than for TCaF.

Model	VGGSound-GZSL^{cls}				UCF-GZSL^{cls}				ActivityNet-GZSL^{cls}			
	S	U	HM	ZSL	S	U	HM	ZSL	S	U	HM	ZSL
[3]	12.63	6.19	8.31	6.91	63.15	30.72	41.34	37.72	16.77	7.04	9.92	7.58
TCaF +att from [3]	10.08	5.16	6.82	5.41	39.47	28.85	33.33	29.79	5.58	2.37	3.33	2.43
TCaF avg input	11.69	5.69	7.65	6.16	12.00	20.46	15.13	20.59	16.43	3.26	5.44	3.42
w/o class. token	**18.40**	3.78	6.27	4.25	31.70	32.57	32.13	33.26	11.87	3.80	5.75	3.90
TCaF +$l_{triplet}$	14.51	4.78	7.19	5.06	**71.61**	35.91	47.83	40.00	18.74	6.58	9.74	6.63
TCaF	12.63	**6.72**	**8.77**	**7.41**	67.14	**40.83**	**50.78**	**44.64**	**30.12**	**7.65**	**12.20**	**7.96**

Table 3. Transforming TCaF into [3]

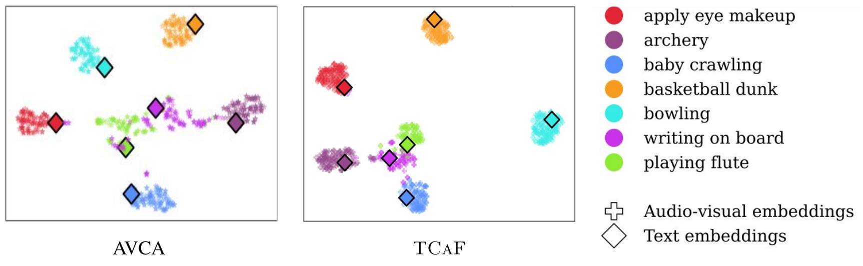

Fig. 2. t-SNE visualisations for five seen (*apply eye makeup, archery, baby crawling, basketball dunk, bowling*) and two unseen (*playing flute, writing on board*) test classes from the UCF-GZSL dataset, showing the difference between TCaF and [3]. Textual class label embeddings are visualised with a square.

4 Computational complexity

The computational complexity increases with the length of the temporal sequence. Using the average duration of the data in UCF-GZSL^{cls} and a single forward pass for a batch of 256 samples, TCaF requires 51.8 GFLOPS vs 174.1 for [2] and 4.4 for [3]. The Perceiver [2] uses a transformer architecture along with the temporal dimension, while [3] does not use the temporal dimension. Thus, it can be observed that TCaF is more resource-efficient than the most similar baseline. TCaF was trained on a single NVIDIA 2080Ti GPU.

References

1. Fayek, H.M., Kumar, A.: Large scale audiovisual learning of sounds with weakly labeled data. In: IJCAI (2020)
2. Jaegle, A., Gimeno, F., Brock, A., Vinyals, O., Zisserman, A., Carreira, J.: Perceiver: General perception with iterative attention. In: ICML (2021)
3. Mercea, O.B., Riesch, L., Koepke, A.S., Akata, Z.: Audio-visual generalised zero-shot learning with cross-modal attention and language. In: CVPR (2022)