



# Improving Audio-Text Retrieval via Hierarchical Cross-Modal Interaction and Auxiliary Captions

Yifei Xin, Yuexian Zou\*

School of ECE, Peking University, Shenzhen, China

xinyifei@stu.pku.edu.cn

## Abstract

Most existing audio-text retrieval (ATR) methods focus on constructing contrastive pairs between whole audio clips and complete caption sentences, while ignoring fine-grained cross-modal relationships, e.g., short segments and phrases or frames and words. In this paper, we introduce a hierarchical cross-modal interaction (HCI) method for ATR by simultaneously exploring clip-sentence, segment-phrase, and frame-word relationships, achieving a comprehensive multi-modal semantic comparison. Besides, we also present a novel ATR framework that leverages auxiliary captions (AC) generated by a pretrained captioner to perform feature interaction between audio and generated captions, which yields enhanced audio representations and is complementary to the original ATR matching branch. The audio and generated captions can also form new audio-text pairs as data augmentation for training. Experiments show that our HCI significantly improves the ATR performance. Moreover, our AC framework also shows stable performance gains on multiple datasets.

**Index Terms:** audio-text retrieval, hierarchical cross-modal interaction, auxiliary captions

## 1. Introduction

Given a caption or an audio clip as a query, the audio-text retrieval (ATR) task aims at retrieving a paired item from a set of candidates in another modality. To compute the similarity between the two modalities, a common technique is to embed the whole audio clip and the complete caption sentence into a joint latent space and then adopt a distance metric like the cosine similarity to measure the relevance of the audio and text. However, in the human sense, due to the intrinsic hierarchical semantic structure in audio and text data, we recognize a audio-text pair by simultaneously analyzing audio-text, segment-phrase, and frame-word interactions. Therefore, most existing ATR methods [1–3] only considering the single cross-modal interaction between whole audio clips and complete caption sentences would result in a biased retrieval.

In this paper, we introduce a hierarchical cross-modal interaction (HCI) approach for ATR, which hierarchically investigates clip-sentence, segment-phrase, and frame-word interactions to understand audio-text contents comprehensively. To explore fine-grained cross-modal interactions, HCI first constructs hierarchical audio representations and text embeddings at respective frame-segment-clip and word-phrase-sentence granularities, as shown in Fig. 1. Taking the audio modality as an

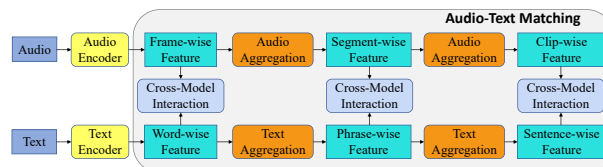


Figure 1: *The overview of our hierarchical cross-modal interaction method for ATR.*

illustration, HCI performs attention-based pooling to aggregate semantically correlated frames into several segment representations, which are then fused into a global clip representation. Similar to the audio modality, a sentence also has multi-level representations consisting of words and phrases, which can be expressed in a word-phrase-sentence manner. Thus, based on hierarchical audio and text representations, HCI employs cross-modal contrastive learning to learn inter-modal correlations at frame-word, segment-phrase, and clip-sentence granularities respectively, thereby accomplishing a more comprehensive cross-modal comparison.

Besides, many video-extracted audio clips (e.g., over 2 million audio clips of AudioSet [4] are collected from YouTube videos) come with associated text information such as titles and tags, which can be utilized to match textual queries. This motivates us to generate associated captions from audio clips to benefit ATR. To achieve this, we present a novel ATR framework that utilizes the pretrained audio captioner (e.g., the widely used CNN10 audio encoder from PANNs [5] and the GRU decoder [6]) to generate captions for each audio clip. The generated captions can be utilized from three aspects. First, the provided audio clip and its generated caption are a matched pair, so they can be used as extra positive sample pairs in addition to the initial audio-text pair as data augmentation during the training stage. Second, we perform cross-modal interaction between the audio and generated captions to enhance audio features. Specifically, we can make use of the information complementarity between audio clips and captions to reduce redundant features from audio clips and learn more discriminative audio representations. Third, we can leverage the text-caption matching to complement the original text-audio matching for ATR, thus reducing the bias of the model and yielding more robust retrieval results.

In a nutshell, our contributions are threefold:

- We introduce a hierarchical cross-modal interaction method for ATR, which explores multi-level cross-modal interactions at clip-sentence, segment-phrase, and frame-word granularities to understand audio-text contents comprehensively.
- We present a novel ATR framework that makes full use of the generated captions from three aspects (i.e., data augmentation, feature interaction, complementary text-caption match-

This paper was partially supported by NSFC (No: 62176008) and Shenzhen Science & Technology Research Program (No:GXWD20201231165807007-20200814115301001).

\* Yuexian Zou is the corresponding author.

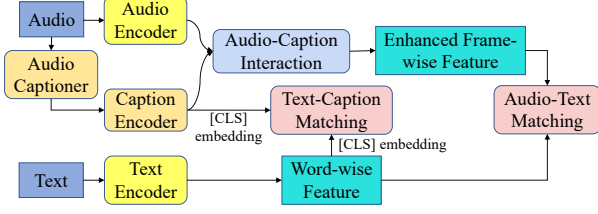


Figure 2: The overview of our auxiliary captions (AC) framework for ATR.

ing) to produce more robust retrieval results.

- Experiments show that our HCI effectively improves the ATR performance. Moreover, our AC framework also shows stable performance gains on multiple datasets.

## 2. Problem Formulation

Let  $D = \{(a_i, t_i)\}_{i=1}^N$  be an audio retrieval dataset containing  $N$  samples, where  $a_i$  is an audio clip and  $t_i$  is the paired text. Therefore,  $(a_i, t_i)$  is regarded as a positive pair while  $(a_i, t_j, j \neq i)$  is a negative pair. The ATR models usually consist of a text encoder  $f_t(\cdot)$  [7] and an audio encoder  $f_a(\cdot)$  [5,8,9] pretrained on the sound event detection task [10, 11], which project the text and audio into a shared embedding space, respectively. For an audio-text pair  $(a_i, t_j)$ , the similarity of the audio and text can be measured by the cosine similarity of their embeddings:

$$s(a_i, t_i) = \frac{f_a(a_i) \cdot f_t(t_i)}{\|f_a(a_i)\|_2 \|f_t(t_i)\|_2}. \quad (1)$$

Currently, the NT-Xent loss [12, 13] based on symmetrical cross-entropy is widely employed, which has been shown to consistently outperform the previous triplet-based losses [14, 15]. Therefore, we adopt it as the baseline loss function for our work. The NT-Xent loss is formulated as below:

$$\mathcal{L}_{at} = -\frac{1}{N} \left( \sum_i \log \frac{\exp(s(a_i, t_i)/\tau)}{\sum_j \exp(s(a_i, t_j)/\tau)} + \sum_j \log \frac{\exp(s(a_i, t_i)/\tau)}{\sum_j \exp(s(a_j, t_i)/\tau)} \right), \quad (2)$$

where  $\tau$  is a temperature hyper-parameter for scaling. Following the previous work [12], we set  $\tau = 0.07$  in our experiments. The training objective is to maximize the similarity of the positive pair relative to all negative pairs, and the ultimate loss is calculated in both directions.

## 3. Proposed methods

### 3.1. Hierarchical Cross-Modal Interaction

We denote  $A_i^f \in \mathbb{R}^{N_f \times D}$  as the frame representations extracted from the audio encoder, and  $T_i^w \in \mathbb{R}^{N_w \times D}$  as the word embeddings extracted from the text encoder, where  $N_f$  is the number of audio frames,  $N_w$  is the number of words, and  $D$  is the feature dimension. To further extract features that capture temporal audio information and long-term word dependence, HCI leverages self-attention [16–18] to aggregate semantically related frames into segment representations and related words into phrase embeddings. Taking audio modality as an example, the aggregation function  $g_a(\cdot)$  is defined as:

$$A_i^s = g_a(A_i^f) = \text{softmax}(A_i^f W_s)^T h(A_i^f), \quad (3)$$

where  $W_s \in \mathbb{R}^{D \times N_s}$  ( $N_s$  is the number of audio segments), and  $\text{softmax}(A_i^f W_s)^T$  projects  $A_i^f$  into normalized frame weights with the dimension of  $N_s \times N_f$ .  $h(\cdot)$  is a two-layer FC-ReLU with feature channel changes  $D-2D-D$ . Thus,  $g_a(\cdot)$  aggregates frame representations  $A_i^f$  into segment representations  $A_i^s \in \mathbb{R}^{N_s \times D}$ , where  $N_s < N_f$ . We denote  $A_i^s = \{A_{i,1}^s, \dots, A_{i,N_s}^s\}$  as a set of  $N_s$  audio segments.  $A_{i,j}^s$  aggregates several semantically related frame representations into a single one, which contains segment information. Similarly, the text aggregation function  $g_t(\cdot)$  is given as:

$$T_i^p = g_t(T_i^w) = \text{softmax}(T_i^w W_p)^T h(T_i^w), \quad (4)$$

where  $W_p \in \mathbb{R}^{D \times N_p}$  ( $N_p$  is the number of phrases), and  $T_i^p \in \mathbb{R}^{N_p \times D}$ .  $A_i^s$  and  $T_i^p$  can be further aggregated into the clip-level representation  $A_i^c \in \mathbb{R}^{1 \times D}$  and the sentence-level embedding  $T_i^s \in \mathbb{R}^{1 \times D}$  using  $g_a(\cdot)$  and  $g_t(\cdot)$ . Based on  $\{A_i^f, A_i^s, A_i^c\}$  and  $\{T_i^w, T_i^p, T_i^s\}$ , HCI can perform comprehensive semantic comparison of audio-text pairs at frame-segment-clip and word-phrase-sentence granularities, respectively.

For  $\{A_i^f, T_i^w\}$ , the contrastive loss of frame-word interaction  $\mathcal{L}_{f-w}$  and the cross-modal interaction (CI) function are given by:

$$\mathcal{L}_{f-w} = -\frac{1}{N} \left( \sum_i \log \frac{\exp(CI(A_i^f, T_i^w)/\tau)}{\sum_j \exp(CI(A_i^f, T_j^w)/\tau)} + \sum_j \log \frac{\exp(CI(A_i^f, T_i^w)/\tau)}{\sum_j \exp(CI(A_j^f, T_i^w)/\tau)} \right), \quad (5)$$

$$CI(A_i^f, T_i^w) = \left( \frac{1}{N_w} \sum_{n=1}^{N_w} \max_{m=1}^{N_f} s(A_{i,m}^f, T_{i,n}^w) + \frac{1}{N_f} \sum_{n=1}^{N_f} \max_{m=1}^{N_w} s(A_{i,n}^f, T_{i,m}^w) \right) / 2, \quad (6)$$

where  $s(\cdot, \cdot)$  denotes the cosine similarity of the two embeddings.  $CI(A_i^f, T_i^w)$  first computes a pair-wise similarity matrix between frames and words and then aggregates all frame-word similarities into an overall score.  $\mathcal{L}_{f-w}$  is a symmetric cross-modal contrastive loss that measures the cross-modal similarity between a set of frames and words.

Similarly, for  $\{A_i^s, T_i^p\}$ , the contrastive loss of segment-phrase interaction  $\mathcal{L}_{s-p}$  and the CI module are denoted as:

$$\mathcal{L}_{s-p} = -\frac{1}{N} \left( \sum_i \log \frac{\exp(CI(A_i^s, T_i^p)/\tau)}{\sum_j \exp(CI(A_i^s, T_j^p)/\tau)} + \sum_j \log \frac{\exp(CI(A_i^s, T_i^p)/\tau)}{\sum_j \exp(CI(A_j^s, T_i^p)/\tau)} \right), \quad (7)$$

$$CI(A_i^s, T_i^p) = \left( \frac{1}{N_s} \sum_{n=1}^{N_s} \max_{m=1}^{N_p} s(A_{i,n}^s, T_{i,m}^p) + \frac{1}{N_p} \sum_{n=1}^{N_p} \max_{m=1}^{N_s} s(A_{i,n}^s, T_{i,m}^p) \right) / 2. \quad (8)$$

For the cross-model interaction between  $\{A_i^c, T_i^s\}$ ,  $\mathcal{L}_{c-s}$  is actually the original NT-Xent loss, which uses cosine similarity to measure the cross-modal similarity between the global clip and sentence representations as the baseline method does. Note that following the previous work [12], we use the [CLS]

Table 1: Performance comparison of our HCI method with  $L_{hci}$  and previous methods with the NT-Xent loss.

Methods	Text-to-Audio		Audio-to-Text	
	R@1	R@10	R@1	R@10
<b>AudioCaps</b>				
ResNet38+NTXent [12]	33.9	82.6	39.4	83.9
CNN14+NTXent	31.4	78.8	38.2	81.9
<b>ResNet38+HCI</b>	<b>36.6</b>	<b>85.6</b>	<b>41.9</b>	<b>85.8</b>
<b>CNN14+HCI</b>	<b>33.9</b>	<b>81.4</b>	<b>41.1</b>	<b>84.3</b>
<b>Clotho</b>				
ResNet38+NTXent [12]	14.4	49.9	16.2	50.2
CNN14+NTXent	13.9	48.2	14.3	49.9
<b>ResNet38+HCI</b>	<b>16.8</b>	<b>52.8</b>	<b>19.1</b>	<b>52.7</b>
<b>CNN14+HCI</b>	<b>15.9</b>	<b>50.1</b>	<b>16.2</b>	<b>51.6</b>

text embedding [19] for the global sentence representation, as it performs better than the sentence-level embedding aggregated from the word-level embeddings.

Finally, the loss function for our HCI is:

$$\mathcal{L}_{hci} = \mathcal{L}_{c-s} + \alpha \mathcal{L}_{f-w} + \beta \mathcal{L}_{s-p}, \quad (9)$$

where  $\alpha$  and  $\beta$  are the hyper-parameters.

### 3.2. Auxiliary Captions

In this section, we first introduce the way to generate associated captions for audio clips. Then, we detail how to make full use of the generated captions to improve the ATR performance. Fig. 2 illustrates the overview of our auxiliary caption (AC) framework.

#### 3.2.1. Caption Generation

To get the auxiliary caption for the given audio clip, we utilize the pretrained encoder-decoder model to generate correlated captions. Specifically, we utilize a 10-layer CNN (CNN10) [5] as the audio encoder and a standard shallow single layer unidirectional GRU [6] as the decoder, which are all commonly used on the audio captioning task [20–22]. Both the encoder and decoder pretrained on the audio captioning dataset are frozen when generating captions for our auxiliary captions (AC) framework. Besides, we adopt the BERT [7] as our caption encoder to generate the caption embedding.

#### 3.2.2. Data Augmentation with Auxiliary Captions

Given the generated captions, the most obvious use for the auxiliary captions is to augment training data. For instance, given a dataset including  $N$  audio clips and corresponding texts, each audio clip and its generated caption are a matched pair, so they can be treated as extra positive sample pairs in addition to the audio-text pair for training. As a result, we can increase  $N$  pairs as additional data augmentation during the training stage.

#### 3.2.3. Audio-Caption Cross-Modal Interaction

We also perform cross-modal interactions between the audio and generated caption to enhance audio representations. Our motivation is to make use of the information complementarity between audio clips and captions to reduce redundant features from audio and learn more discriminative audio representations. Specifically, we feed the frame-level audio embeddings (as query) and the [CLS] embedding of the generated caption (as key and value) into the audio-caption interaction module. The audio-caption interaction module employs one co-attention

Table 2: Performance comparison of our auxiliary captions (AC) framework with previous methods.

Methods	Text-to-Audio		Audio-to-Text	
	R@1	R@5	R@1	R@5
<b>AudioCaps</b>				
Audio-Text [12]	33.9	69.7	39.4	72.0
<b>Audio-Text+AC</b>	<b>35.4</b>	<b>71.3</b>	<b>41.1</b>	<b>73.6</b>
<b>Audio-Text+AC+HCI</b>	<b>37.2</b>	<b>72.7</b>	<b>43.3</b>	<b>75.2</b>
<b>Clotho</b>				
Audio-Text [12]	14.4	36.6	16.2	37.5
<b>Audio-Text+AC</b>	<b>16.8</b>	<b>38.7</b>	<b>18.7</b>	<b>39.9</b>
<b>Audio-Text+AC+HCI</b>	<b>18.2</b>	<b>39.4</b>	<b>19.9</b>	<b>41.3</b>

transformer layer [23–25] to facilitate cross-modality information exchange, which passes the keys and values (caption embeddings) from the caption modality to the queries (frame-level embeddings) of the audio modality, followed by a standard transformer layer to model temporal information, thereby obtaining the enhanced frame-wise audio representations.

#### 3.2.4. Complementary Text-Caption Matching

In addition to the uses of captions for data augmentation and audio representation enhancement mentioned above, the generated caption itself can also reflect the content of the audio, allowing us to leverage the generated caption for text-caption retrieval. Specifically, each caption generated by the audio clip is then passed through the caption encoder to obtain its [CLS] embedding. Then, the cosine similarity between the caption embedding and the text embedding is calculated to complement the audio-text matching.

We denote  $T_i^c$ ,  $C_i^c$  as the text and the caption [CLS] embeddings of the  $i$ -th sample, respectively. For the text-caption branch, it is preferable that the text embedding  $T_i^c$  and the caption embedding  $C_i^c$  be close when they are related and far apart when they are not during the training stage. We follow the common practice [12, 13] to consider the bidirectional learning objective, which uses the symmetric cross-entropy loss to maximize the similarity between matched text-caption pairs while minimizing the similarity for other pairs:

$$\mathcal{L}_{tc} = -\frac{1}{N} \left( \sum_i \log \frac{\exp(s_{tc}(T_i^c, C_i^c)/\tau)}{\sum_j \exp(s_{tc}(T_i^c, C_j^c)/\tau)} + \sum_i \log \frac{\exp(s_{tc}(T_i^c, C_i^c)/\tau)}{\sum_j \exp(s_{tc}(T_j^c, C_i^c)/\tau)} \right), \quad (10)$$

where  $s_{tc}(\cdot, \cdot)$  represents the text-caption matching similarity function. The total loss  $\mathcal{L}_{total}$  is the sum of audio-text loss  $\mathcal{L}_{at}$  and the text-caption loss  $\mathcal{L}_{tc}$ :

$$\mathcal{L}_{total} = \mathcal{L}_{at} + \mathcal{L}_{tc}. \quad (11)$$

Notably, the  $\mathcal{L}_{at}$  can be replaced by our HCI loss  $\mathcal{L}_{hci}$ , which can achieve a higher performance.

## 4. Experiments and Results

### 4.1. Datasets

We evaluate our methods on two publicly available datasets: AudioCaps [26] and Clotho [27] datasets. AudioCaps contains about 50K audio samples, which are all 10-second long. The training set consists of 49274 audio clips, each with one human-annotated caption. The validation and test sets contain 494 and

Table 3: Ablation study of our  $\mathcal{L}_{hci}$  loss.

Methods	Text-to-Audio		Audio-to-Text	
	R@1	R@10	R@1	R@10
$\mathcal{L}_{c-s}$ [12]	33.9	82.6	39.4	83.9
+ $\mathcal{L}_{f-w}$	35.9	85.1	41.1	85.2
+ $\mathcal{L}_{s-p}$	34.4	84.3	40.6	84.6
+ $\mathcal{L}_{f-w}$ + $\mathcal{L}_{s-p}$	<b>36.6</b>	<b>85.6</b>	<b>41.9</b>	<b>85.8</b>

Table 4: Influences of different text embeddings, and segment-pharse numbers on the AudioCaps dataset.

Methods	Text-to-Audio		Audio-to-Text	
	R@1	R@10	R@1	R@10
HCI (Avg)	36.1	84.9	41.4	85.2
<b>HCI ([CLS])</b>	<b>36.6</b>	<b>85.6</b>	<b>41.9</b>	<b>85.8</b>
HCI ( $N_s=8$ )	36.3	85.5	41.5	85.3
<b>HCI (<math>N_s=10</math>)</b>	<b>36.6</b>	<b>85.6</b>	<b>41.9</b>	<b>85.8</b>
HCI ( $N_s=12$ )	36.2	85.2	41.4	85.6

957 audio clips, each with five human-annotated captions. The Clotho v2 dataset contains 6974 audio samples between 15 and 30 seconds in length. Each audio sample is annotated with 5 sentences. The numbers of training, validation, and test samples are 3839, 1045, and 1045, respectively.

#### 4.2. Training Details and Evaluation metrics

In our work, we follow the same pipeline in [12] to train our networks. We adopt BERT [7] as the text encoder, while employing the ResNet-38 and CNN14 in pre-trained audio neural networks (PANNs) [5] as the audio encoder. The dataset for captioner pretraining is consistent with the dataset for the ATR task (e.g., the captioner are pretrained on the AudioCaps when training and evaluating the retrieval performance on the AudioCaps dataset). We conduct experiments by fine-tuning the pre-trained text and audio encoders, while freezing the pretrained captioner and the caption encoder to generate the caption and its embedding for our AC framework. The hyper-parameters are set as  $N_s = N_p = 10$ ,  $\alpha = 0.5$ ,  $\beta = 0.1$ . Recall at rank  $k$  (R@k) is utilized as the evaluation metric, which is a popular cross-modal retrieval evaluation protocol. R@k measures the proportion of targets retrieved within the top-k ranked results, so a higher score means better performance. The results of R@1, R@5, and R@10 are reported.

#### 4.3. Experimental Results

As shown in Table 1, we first compare the performance of our hierarchical cross-modal interaction (HCI) method (using  $\mathcal{L}_{hci}$ ) with previous baselines (using the NT-Xent loss). We adopt either the ResNet-38 or the CNN14 as the audio encoder on the AudioCaps and Clotho datasets. It can be seen that our HCI brings significant gains with different audio encoders on both datasets, thus demonstrating the effectiveness of our method.

To evaluate our auxiliary captions (AC) framework, we compare it with the baseline method of just using original audio-text pairs to compute the similarity with the NT-Xent loss. Here, we choose ResNet-38 as the audio encoder. As can be seen in Table 2, our AC framework also achieves performance boosts by a large margin. Moreover, when using our  $\mathcal{L}_{hci}$  to replace the NT-Xent loss, the performance can be further improved, which strongly proves the effectiveness and robustness of our HCI method and AC framework.

Table 5: Influences of each part of our AC framework.

Methods	Text-to-Audio		Audio-to-Text	
	R@1	R@5	R@1	R@5
Audio-Text [12]	14.4	36.6	16.2	37.5
+DA	15.1	37.2	16.8	38.2
+ACFI	15.9	37.8	17.6	39.1
<b>+TCM</b>	<b>16.8</b>	<b>38.7</b>	<b>18.7</b>	<b>39.9</b>

#### 4.4. Ablation Study

In this part, we discuss the influence of each term of our  $\mathcal{L}_{hci}$  loss, the selection of text embeddings with the [CLS] token and the aggregated sentence-level embedding, different segment-pharse numbers, and each component of our AC framework. Here, we use the ResNet-38 as the audio encoder.

**Results of each term of our  $\mathcal{L}_{hci}$  loss.** To show the effectiveness of each term of our  $\mathcal{L}_{hci}$  loss, we present ablation results on the AudioCaps dataset in Table 3, where  $\mathcal{L}_{c-s}$  is our baseline method using the NT-Xent loss. It can be seen that compared to the segment-pharse interaction, the frame-word interaction brings a larger performance benefit, while combining the frame-word and segment-pharse interactions can further boost the performance, indicating that performing the multi-level interaction between audio and text can effectively benefit the ATR task.

**Results of different text embeddings, and segment-pharse numbers.** As shown in Table 4, HCI (\*) means that we only change the setting of \*, and the rest of the settings are the same as the experimental best settings, which we show in bold. We first conduct experiments for comparing the performance of the [CLS] text embedding and the aggregated sentence-level embedding. It is clear that the [CLS] embedding consistently performs better than the aggregated sentence-level embedding. Then, we experiment with different  $N_s$  ( $= N_p$ ) numbers, indicating how many segments or phrases HCI obtains from frames or words. It is clear that  $N_s = N_p = 10$  achieves the best performance.

**Results of each part of our AC framework.** As shown in Table 5, we provide ablation results to clarify the effects of our AC framework on the Clotho dataset, where +DA represents that we use auxiliary captions only for data augmentation based on our baseline models, +ACFI denotes that we add the audio-caption interaction module based on +DA, and +TCM means to add the text-caption matching module based on +ACFI. It can be seen that our method achieves consistent performance gains by merging each component of our AC framework step by step, which highly verifies the effectiveness of our AC framework.

## 5. Conclusions

In this paper, we present a hierarchical cross-modal interaction (HCI) method for ATR, which simultaneously explores clip-sentence, segment-pharse, and frame-word relationships to understand audio-text contents. Besides, we also develop a framework that leverages auxiliary captions (AC) generated by a pretrained captioner to benefit the audio-text matching from three aspects, i.e., data augmentation, audio-caption feature interaction for enhancing audio representations, and text-caption matching to complement the original ATR matching branch. Experiments show that our HCI significantly improves the ATR performance. Moreover, our AC framework also shows stable performance gains on the AudioCaps and Clotho datasets.

## 6. References

- [1] A. S. Koepke, A.-M. Oncescu, J. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries: A benchmark study," *IEEE Transactions on Multimedia*, 2022.
- [2] S. Lou, X. Xu, M. Wu, and K. Yu, "Audio-text retrieval in context," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 4793–4797.
- [3] Y. Xin, D. Yang, and Y. Zou, "Improving text-audio retrieval by text-aware attention pooling and prior matrix revised loss," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [4] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.
- [5] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.
- [6] X. Xu, H. Dinkel, M. Wu, Z. Xie, and K. Yu, "Investigating local and global information for automated audio captioning with transfer learning," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 905–909.
- [7] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of naacl-HLT*, 2019, pp. 4171–4186.
- [8] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 646–650.
- [9] Y. Xin, D. Yang, and Y. Zou, "Audio pyramid transformer with domain adaption for weakly supervised sound event detection and audio classification," *Proc. Interspeech 2022*, pp. 1546–1550, 2022.
- [10] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound event detection: A tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, 2021.
- [11] Y. Xin, D. Yang, F. Cui, Y. Wang, and Y. Zou, "Improving weakly supervised sound event detection with causal intervention," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [12] X. Mei, X. Liu, J. Sun, M. D. Plumbley, and W. Wang, "On metric learning for audio-text cross-modal retrieval," *arXiv preprint arXiv:2203.15537*, 2022.
- [13] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.
- [14] H. Xie, S. Lipping, and T. Virtanen, "Dcase 2022 challenge task 6b: Language-based audio retrieval," *arXiv e-prints*, pp. arXiv:2206.2022, 2022.
- [15] T. L. de Gail and D. Kicinski, "Take it easy: Relaxing contrastive ranking loss with cider," DCASE2022 Challenge, Tech. Rep., Tech. Rep., 2022.
- [16] H. Touvron, M. Cord, A. El-Nouby, P. Bojanowski, A. Joulin, G. Synnaeve, and H. Jégou, "Augmenting convolutional networks with attention-based aggregation," *arXiv preprint arXiv:2112.13692*, 2021.
- [17] Y. Wang, J. Li, and F. Metze, "A comparison of five multiple instance learning pooling functions for sound event detection with weak labeling," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 31–35.
- [18] J. Jiang, S. Min, W. Kong, H. Wang, Z. Li, and W. Liu, "Tencent text-video retrieval: Hierarchical cross-modal interactions with multi-level representations," *IEEE Access*, 2022.
- [19] X. Mei, X. Liu, H. Liu, J. Sun, M. D. Plumbley, and W. Wang, "Language-based audio retrieval with pre-trained models," Tech. rep., DCASE2022 Challenge, Tech. Rep., 2022.
- [20] Y. Zhang, H. Yu, R. Du, Z. Ma, and Y. Dong, "Caption feature space regularization for audio captioning," *arXiv preprint arXiv:2204.08409*, 2022.
- [21] X. Mei, X. Liu, M. D. Plumbley, and W. Wang, "Automated audio captioning: An overview of recent progress and new challenges," *EURASIP journal on audio, speech, and music processing*, vol. 2022, no. 1, pp. 1–18, 2022.
- [22] Z. Zhou, Z. Zhang, X. Xu, Z. Xie, M. Wu, and K. Q. Zhu, "Can audio captions be evaluated with image caption metrics?" in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 981–985.
- [23] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *Advances in neural information processing systems*, vol. 32, 2019.
- [24] R. J. Chen, M. Y. Lu, W.-H. Weng, T. Y. Chen, D. F. Williamson, T. Manz, M. Shady, and F. Mahmood, "Multimodal co-attention transformer for survival prediction in gigapixel whole slide images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4015–4025.
- [25] W. Wu, H. Luo, B. Fang, J. Wang, and W. Ouyang, "Cap4video: What can auxiliary captions do for text-video retrieval?" *arXiv preprint arXiv:2301.00184*, 2022.
- [26] C. D. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 119–132.
- [27] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 736–740.