Interspeech 2024
1-5 September 2024, Kos, Greece

GPA: Global and Prototype Alignment for Audio-Text Retrieval

*Yuxin Xie, Zhihong Zhu, Xianwei Zhuang, Liming Liang, Zhichang Wang, Yuexian Zou**

School of ECE, Peking University, China

{yuxinxie, zhihongzhu, xwzhuang, limingliang, wzcc}@stu.pku.edu.cn
zouyx@pku.edu.cn

Abstract

Recent Audio-Text Retrieval (ATR) models have achieved progressive results, which pursue semantic interaction upon audio and text pairs. To clarify this coarse-grained global interaction and move a step further, we have to encounter challenging shell-breaking interactions for fine-grained cross-modal learning between audio and text. In this paper, we present GPA for ATR to achieve both Global (coarse-grained) and Prototype (fine-grained) Alignment. In detail, apart from performing vanilla global contrast between audio and text pairs, we model the frames in audio and words in text as prototypes, and align the prototypes to generate a prototype similarity matrix. Based on this, we introduce a Learnable Attention Similarity Scoring module, which can fully consider the information between different prototype pairs and obtain the retrieval score. Finally, we incorporate the Sinkhorn-Knopp algorithm to modify the retrieval score. Experimental results on two benchmark datasets with superior performance justify the efficacy of our proposed GPA.

Index Terms: audio-text retrieval, fine-grained alignment, learnable attention

1. Introduction

Audio-Text Retrieval (ATR) is a significant and challenging task in cross-modal interaction [1, 2, 3, 4, 5], which has received widespread attention in recent years [6]. Given a text (audio) query, the goal of ATR is to retrieve the corresponding audio (text) in the candidate pool. Existing ATR methods [7, 8, 9] predominantly focus on devising cross-modal interactions under a joint latent space and calculating the global cosine similarity score for retrieval. Therein, [7] introduced audio retrieval benchmarks and provides baseline results through multi-modal video retrieval methods. [8] demonstrated that audio features extracted using pre-trained model PANNs [10] outperform commonly used static features such as log-mel spectrogram (LMS) and mel-frequency cepstral coefficients (MFCC) [11]. [9] evaluated the impact of different metric learning objectives on retrieval performance based on pre-trained models.

Although these studies have made significant progress, we find that they still face two key issues: **(1) Neglecting fine-grained alignment**. As shown in Figure 1, previous studies utilize vanilla contrastive learning strategy to perform global alignment between acoustic and textual features obtained through audio and text encoders, respectively. As illustrated in Figure 1(b), there exists a more compact and complex correspondence relationship between audio clips and textual tokens, Capturing this correspondence is essential for establishing the intrinsic relationship between audio and text modalities. However, previous methods neglect this fine-grained alignment of clip-level acoustic features and token-level textual features. **(2) Retrieval score imbalance in Audio-to-Text Retrieval.** A large number of previous experiments [8, 12] and empirical studies have identified an imbalance phenomenon [13] within the Audio-to-Text retrieval task. Specifically, the cumulative retrieval scores associated with a particular text across all audio instances significantly surpass those for other texts. This imbalance leads to an over-selection of certain texts in the retrieval process, which adversely impacts the overall retrieval performance.

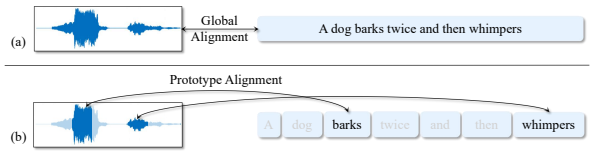

Figure 1: *An example of Audio-Text Retrieval. (a) audio text alignment adopted by existing methods (Global Alignment). (b) there is a correspondence between the fine-grained information in audio and text.*

To solve the aforementioned issues, in this paper, we present GPA for ATR to achieve both Global (coarse-grained) and Prototype (fine-grained) Alignment. **For the first issue**, we model the collections of audio frames as audio prototypes and the collections of words as text prototypes and consider fine-grained alignment at the prototype level. As shown in Figure 2, we use the mask-based method to generate prototypes and get the similarity matrix by aligning them. Furthermore, we propose a Learnable Attention Similarity Scoring module (LASS) that assigns a weight to each value in the prototype similarity matrix and transforms the prototype similarity matrix into a prototype similarity score. The retrieval score can be obtained by adding the global similarity score and the prototype similarity score. **For the second issue**, we introduce the Sinkhorn-Knopp algorithm [14] to correct imbalance issues and further improve retrieval performance. In detail, we use the training query set as an approximation of the test set and apply the Sinkhorn-Knopp algorithm to calculate instance bias to adjust the cross-modal retrieval score so that each instance is fairly represented during the retrieval process. We outperform the state-of-the-art performance on the AudioCaps [15] and Clotho [16] datasets by 6.7% and 9.0% relative improvements on Text-to-Audio retrieval, and 12.2% and 14.8% on Audio-to-Text retrieval.

Overall, our contributions in this work are three-fold:

- We capture the fine-grained information in audio and text to

This paper was partially supported by NSFC(No:62176008).
* Yuexian Zou is the corresponding author.

10.21437/Interspeech.2024-1642

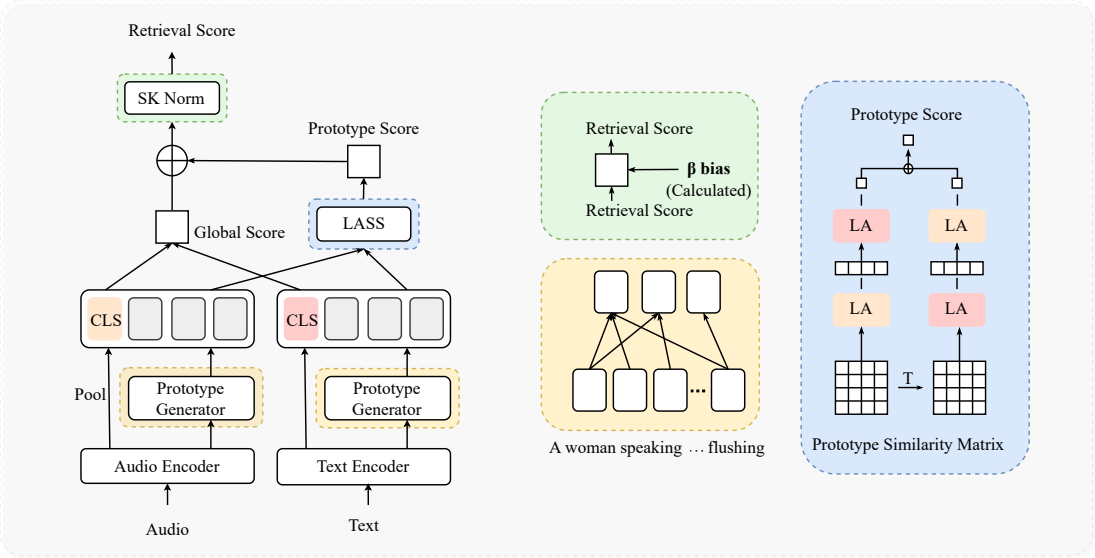

Figure 2: *Illustration of the proposed* GPA *model. The input sentences are processed by a text encoder and a prototype generator to generate global-level and prototype-level text representations. The input audio is processed by an audio encoder and a prototype generator to generate global-level and prototype-level audio representations. Based on these representations, we compute the similarity score at the global level and prototype level, and introduce the Sinkhorn-Knopp algorithm to correct the retrieval score.*

generate prototypes and align the prototypes to obtain the prototype similarity matrix. Our proposed LASS can well aggregate prototype similarity matrix and obtain the similarity score for audio and text prototypes.

- We aggregate the similarity scores at the global and prototype levels to get the retrieval score, and we are the first to introduce the Sinkhorn-Knopp algorithm to correct the imbalance problem in the Audio-to-Text task.
- Experimental results show that our algorithm effectively improves the performance of the ATR task, and shows steady improvement on two benchmark datasets.

2. Method

In this section, we elaborate on each component of our proposed GPA, whose architecture is shown in Figure 2.

2.1. Problem Definition

Let $D = \{(a_i, t_i)\}_{i=1}^{N}$ be an audio captioning dataset of N examples, where a_i is an audio clip and t_i is the paired caption. Therefore, (a_i, t_i) is regarded as a positive pair while $(a_i, t_{j,j\neq i})$ is a negative pair. The goal of the ATR task is to train retrieval models by calculating the audio-text retrieval score and making the retrieval score of positive pairs r_{ii} higher than that of negative pairs r_{ij}.

2.2. Prototype and Global Feature Alignment

Audio and Text Encoder. Following [9], the ResNet-38 [17] in PANNs [10] is employed as the audio encoder, where the last two linear layers are discarded. And pre-trained BERT [18] is employed as the text encoder.

Prototype Generator. We denote $a \in \mathbb{R}^{N_a \times D_a}$ as the frame representations extracted from the audio encoder, and $t \in \mathbb{R}^{N_t \times D_t}$ as the word embeddings extracted from the text encoder, where N_a is the number of audio frames and N_t is the number of words. D_a and D_t are the feature dimensions.

There are many ways to generate text and audio prototypes, including attention [19, 20, 21], graph neural networks [22, 23, 24, 25], and other methods [26, 27, 28, 29, 30, 31]. In this work, we use a simpler mask-based method to generate prototypes. The kth text prototype is formulated as follows:

$$p_k^t = \sum_{j=1}^{N_t} t^j \cdot f^{mask}(t^j)_k, \tag{1}$$

where t^j, f^{mask} represents the jth token feature of t, mask-generating function. f^{mask} is implemented by a linear layer, followed by the relu function. In general, the class token feature obtained by identifier `[CLS]` can represent the final output of BERT. So we expand $p_{k+1}^t = t^{cls}$, which means we have $K+1$ text prototypes. Similarly, we use the output of the average and max pooling layers as audio `[CLS]` to generate $K+1$ audio prototypes. Through the prototype generator, we obtain the prototypes of the audio $p^a \in \mathbb{R}^{(K+1)\times D_a}$ and text $p^t \in \mathbb{R}^{(K+1)\times D_t}$. Finally, the MLP block is used to project audio and text features into the shared embedding space.

Global and Prototype Similarity Matrix. For an audio-text pair, the global similarity score s_g is obtained by the cosine similarity of global features. Similarly, we calculate cosine similarity for different prototypes and obtain the fine-grained cosine similarity matrix $s_p \in [K+1, K+1]$.

2.3. Learnable Attention Similarity Scoring Module

For the similarity matrix at the prototype level, using mean pooling to aggregate prototype information ignores the importance of different acoustic and textual prototypes. We propose the Learnable attention similarity scoring module (LASS), which captures the interaction between different audio and text prototypes and assigns different weights to different similarity values. Since the prototype similarity matrix s_p contains the

similarity of $K+1$ audio prototypes and $K+1$ text prototypes, we perform two learnable attention (LA) operations on the matrix. The first attention goal is to obtain audio prototype and text prototype similarity vectors. Audio prototype similarity vector s_a can be formulated as follows:

$$w_a = \frac{exp(s_{p(i,*)})}{\sum_{j=1}^{K+1} exp(s_{p(j,*)})}. \tag{2}$$

We add a linear layer L_a based on the weight w_a, then we get the learnable weights w_{Linear_a}:

$$w_{Linear_a} = \frac{exp(L_a(w_a)_{(i,*)})}{\sum_{j=1}^{K+1} exp(L_a(w_a)_{(j,*)})}. \tag{3}$$

We assign the weight w_{Linear_a} to each value in the prototype similarity matrix:

$$s_a = \sum_{i=1}^{K+1} w_{Linear_a(i,*)} s_{p(i,*)}, \tag{4}$$

where $*$ represents all content in the dimension, $s_a \in \mathbb{R}^{1\times(K+1)}$ is the audio-level similarity vector. Specifically, $s_a \in \mathbb{R}^{1\times(K+1)}$ shows the similarity score between the audio and $K+1$ text prototypes in the sentence. To obtain audio prototype-level similarity score s'_a, we conduct the second LA operation on the audio-level vector s_a:

$$w_t = \frac{exp(s_{a(1,i)})}{\sum_{j=1}^{K+1} exp(s_{a(1,j)})}, \tag{5}$$

$$w_{Linear_t} = \frac{exp(L_t(w_t)_{(1,i)})}{\sum_{j=1}^{K+1} exp(L_t(w_t)_{(1,j)})}, \tag{6}$$

$$s'_a = \sum_{i=1}^{K+1} w_{Linear_t(1,i)} s_{a(1,i)}. \tag{7}$$

Then we gain audio prototype-level similarity score $s'_a \in \mathbb{R}^1$. Text prototype-level similarity score s'_t can be formulated in the same way: perform first LA use L_t, then gain text-level vector s_t, then perform second LA use L_a, then gain prototype-level similarity score $s'_t \in \mathbb{R}^1$. We use the sum value as the prototype-level similarity score s'_p:

$$s'_p = s'_a + s'_t. \tag{8}$$

2.4. Retrieval Score Correction

As shown above, we describe the similarity score of an audio-text pair, with a global-level score s_g and a prototype-level score s'_p. The actual situation is that we have Q audio queries and J texts, and we calculate the global similarity scores as $S_g \in \mathbb{R}^{Q\times J}$ and prototype similarity scores as $S'_p \in \mathbb{R}^{Q\times J}$ of all audio queries and texts, where S^{ij} is the score for the i^{th} audio query and j^{th} text. We add the global-level and prototype-level similarity scores to get retrieval scores:

$$R = S_g + S'_p. \tag{9}$$

In order to solve the imbalance phenomenon in the Audio-to-Text task, we follow [13] and introduce the Sinkhorn-Knopp algorithm [14] to correct the imbalance. We refer the readers to Appendix for more details about Sinkhorn-Knopp algorithm.

Even though we can access all text and audio during the testing phase, we can only get one audio query at a time for audio-to-text retrieval. Thus we use the training audio query to simulate the test audio. Specifically, assuming we have G training queries and J test texts, we obtain the retrieval score R' between the training query and the test text by the aforementioned method. We further add the test text bias to the retrieval score matrix $R \in \mathbb{R}^{Q\times J}$ generated from Q test queries and J test texts. Specifically, we calculate the text bias in an alternating iterative manner to obtain the modified matrix R^* as:

$$R^{ij*} = R^{ij} + \mathrm{SK}_{norm}(R')^j, \tag{10}$$

where, $\mathrm{SK}_{norm}(\cdot)$ means the Sinkhorn-Knopp operation. Please refer to the supplementary material for the specific algorithm. Note that we only apply Equation 10 in the inference phase, Text-to-audio retrieval does not suffer from similar problems, and we verify the effectiveness of our Sinkhorn-Knopp algorithm in ablation experiments.

2.5. Training Objective

During training, given a batch of B audio-text pairs, the model generates a $B\times B$ retrieval score matrix. [9] evaluate the effects of multiple loss functions, and we select two of them to train our model. We use Triplet-sum and NT-Xent [32] loss functions respectively to optimize the retrieval model.

$$\mathcal{L}_{triplet} = \frac{1}{B}\sum_{i=1}^{B}\sum_{j\neq i}[m + R^{ij} - R^{ii}]_+ + [m + R^{ji} - R^{ii}]_+, \tag{11}$$

where $[x]_+ = \max(0, x)$ and m is a distance margin.

$$\mathcal{L}_{nt-xent} = -\frac{1}{B}\left(\sum_{i=1}^{B}\log\frac{\exp(R^{ii}/\tau)}{\sum_{j=1}^{B}\exp\left(R^{ij}/\tau\right)} + \sum_{i=1}^{B}\log\frac{\exp(R^{ii}/\tau)}{\sum_{j=1}^{B}\exp\left(R^{ji}/\tau\right)}\right), \tag{12}$$

where τ is a temperature hyper-parameter.

3. Experiments

3.1. Datasets, Metrics and Implementation Details

Datasets. We evaluate GPA on two ATR benchmarks: AudioCaps [15] and Clotho [16]. AudioCaps is a large captioning dataset containing approximately 50K 10-second long audio clips. Its training set has 49274 audio clips, and each audio clip is equipped with one human-annotated caption; the validation set and test set have 494 and 957 audio clips respectively, and each audio clip is equipped with five human-annotated captions. Clotho v2 contains 6974 audio samples between 15 and 30 seconds in length. Each audio sample has five human-annotated captions. The number of training samples, validation samples, and test samples are 3839, 1045, and 1045 respectively.

Metrics. The evaluation metric R@k we used is fully named Recall at rank k, including R@1, R@5, and R@10 to validate the effectiveness of our GPA. R@k evaluates the proportion of desired outcomes that appear within the top k-ranked results, indicating that a higher score reflects superior performance.

Implementation Details. We follow the same process as in [9] to train our network, adopting ResNet-38 in PANNs as the audio encoder and pre-trained BERT as the text encoder. The model is trained using the Adam [33] optimizer for 50 epochs, and the learning rate is attenuated to 0.1 of itself every 20 epochs. For the AudioCaps dataset, the batch size is set to

Table 1: *Results of the experiments. DP (baseline) represents existing methods that use global similarity to represent retrieval scores.*

Dataset	Methods	Objective	Text-to-Audio			Audio-to-Text		
			R@1	R@5	R@10	R@1	R@5	R@10
AudioCaps	DP [9]	Triplet-sum	$32.2_{\pm 0.3}$	$68.2_{\pm 0.6}$	$81.6_{\pm 0.5}$	$36.1_{\pm 1.2}$	$69.2_{\pm 1.3}$	$81.4_{\pm 1.7}$
		NT-Xent	$33.9_{\pm 0.4}$	$69.7_{\pm 0.2}$	$82.6_{\pm 0.3}$	$39.4_{\pm 1.0}$	$72.0_{\pm 1.0}$	$83.9_{\pm 0.6}$
	GPA	Triplet-sum	$\mathbf{35.3_{\pm 0.2}}$	$\mathbf{71.0_{\pm 0.1}}$	$\mathbf{83.2_{\pm 0.3}}$	$\mathbf{42.6_{\pm 0.1}}$	$\mathbf{74.5_{\pm 0.4}}$	$\mathbf{86.7_{\pm 0.2}}$
		NT-Xent	$\mathbf{36.2_{\pm 0.2}}$	$\mathbf{71.4_{\pm 0.1}}$	$\mathbf{82.9_{\pm 0.2}}$	$\mathbf{44.2_{\pm 0.2}}$	$\mathbf{75.9_{\pm 0.3}}$	$\mathbf{86.7_{\pm 0.4}}$
Clotho	DP [9]	Triplet-sum	$14.2_{\pm 0.5}$	$36.6_{\pm 0.5}$	$49.3_{\pm 0.7}$	$16.1_{\pm 0.7}$	$37.5_{\pm 1.2}$	$50.7_{\pm 1.0}$
		NT-Xent	$14.4_{\pm 0.4}$	$36.6_{\pm 0.2}$	$49.9_{\pm 0.2}$	$16.2_{\pm 0.7}$	$37.5_{\pm 0.9}$	$50.2_{\pm 0.7}$
	GPA	Triplet-sum	$\mathbf{15.7_{\pm 0.1}}$	$\mathbf{39.0_{\pm 0.1}}$	$\mathbf{51.3_{\pm 0.3}}$	$\mathbf{18.2_{\pm 0.3}}$	$\mathbf{40.9_{\pm 0.8}}$	$\mathbf{53.9_{\pm 0.4}}$
		NT-Xent	$\mathbf{15.7_{\pm 0.1}}$	$\mathbf{39.1_{\pm 0.1}}$	$\mathbf{50.9_{\pm 0.1}}$	$\mathbf{18.6_{\pm 0.1}}$	$\mathbf{42.4_{\pm 0.4}}$	$\mathbf{55.3_{\pm 0.3}}$

Table 2: *Ablation study of Prototype Generator.*

Methods	K	Text-to-Audio		Audio-to-Text	
		R@1	R@5	R@1	R@5
Baseline	0	33.9	69.7	39.4	72.0
Attention	3	35.1	70.6	44.3	75.4
Mask	3	**36.2**	**71.4**	44.2	**75.9**
Mask	1	35.1	70.8	43.2	74.9
Mask	2	35.5	70.8	45.1	74.7
Mask	4	35.3	70.1	**45.6**	74.2
Mask	5	36.1	70.6	43.9	75.8
Mask	10	35.7	70.5	43.2	74.8

32 and the learning rate is set to 1×10^{-4}. For the Clotho dataset, the batch size is set to 24 and the learning rate is set to 5×10^{-5}. For Triplet-sum loss, the distance margin m is set to 0.2, and the temperature hyper-parameter τ in the NT-Xent loss is set to 0.07. In the Sinkhorn-Knopp algorithm, we set the number of iterations as 10 across all datasets. All experiments are conducted on 8 RTX3090 GPUs.

3.2. Main Results

ATR is divided into Text-to-Audio retrieval (T2A) and Audio-to-Text retrieval (A2T) tasks. As shown in Table 1, we compare the proposed GPA with the baseline on the AudioCaps and Clotho datasets, where DP [9] (baseline) represents existing methods that use global similarity to represent retrieval scores.

As can be seen from Table 1, whether using NT-Xent or Tript-sum loss, our method is better than the baseline. When NT-Xent is selected as the learning objective, on the dataset AudioCaps, the R@1 index of T2A has improved by 6.7% and the R@1 index of A2T has improved by 12.2%. The size of the training data in the Clotho dataset is limited, and the text corresponding to the audio is more diverse, making the ATR task on the Clotho dataset more difficult. On the dataset Clotho, the R@1 index of T2A has improved by 9.0%, and the R@1 index of A2T has improved by 14.8%. Our model brings significant gains on different loss functions and different datasets, proving the effectiveness of our model.

3.3. Model Analysis

The experiments in this section are conducted on the AudioCaps dataset and use NT-Xent as the loss function.

Different Prototype Generator and the number of prototypes. Our GPA introduces a Mask-based prototype generation method. To verify the effect of this Mask-based method, we design an Attention-based prototype generation method. The Attention-based method uses the Self-Attention method to generate K prototypes. As shown in Table 2, the Mask-based method is significantly better than the Attention-based method on T2A Task, increasing by 3.1% on R@1 and 1.1% on R@5.

Table 3: *Ablation study of prototype similarity scoring model.*

Methods	Text-to-Audio		Audio-to-Text	
	R@1	R@5	R@1	R@5
Mean Pooling	34.7	69.1	41.6	73.5
Softmax Weight	35.3	70.7	43.7	74.4
LASS	**36.2**	**71.4**	**44.2**	**75.9**

Table 4: *Ablation study of Sinkhorn-Knopp algorithm.*

Methods	Audio-to-Text		
	R@1	R@5	R@10
w/o SK norm	43.5	74.9	86.2
w SK norm	**44.2**	**75.9**	**86.7**

The performance of the method of adding prototypes is better than the baseline, which shows that there is a connection between the rich fine-grained information in audio and text, which is beneficial to improving retrieval performance. The selection of the number of prototypes is crucial. $K = 3$ performs better than $K = 1, 2, 4, 5, 10$. An appropriate number of prototypes can capture key information in audio and text.

Different ways to calculate prototype scores. In the above, we propose LASS to aggregate the prototype similarity matrix into the prototype similarity score. We utilize the non-learnable average pooling aggregation method and softmax weight aggregation method to verify the effectiveness of LASS. As shown in Table 3, our proposed LASS outperforms average pooling and softmax weight in various indicators, which shows that our proposed LASS can effectively capture the interaction between different audio and text prototypes, and reasonably assign different weights to different similarity values.

Effectiveness of Sinkhorn-Knopp algorithm. We compare our GPA model with removing Sinkhorn-Knopp algorithm in the GPA. As can be seen from Table 4, introducing Sinkhorn-Knopp algorithm can effectively improve the performance of the Audio-to-Text task, improving the R@1 by 1.6% and the R@5 1.3%. The Sinkhorn-Knopp algorithm we introduced can effectively correct the imbalance problem in the A2T task.

4. Conclusions

In this paper, we propose GPA, an end-to-end framework for Audio-Text Retrieval. It automatically generates prototypes on both audio and text sides to represent rich information and proposes a learnable attention similarity scoring module to calculate prototype similarity score. In addition, we also introduce the Sinkhorn-Knopp algorithm into the ATR task, further improving the performance of the model. Experimental results on AudioCaps and Clotho datasets demonstrate the effectiveness and versatility of our proposed model.

5. References

[1] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei *et al.*, "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*. Springer, 2020, pp. 121–137.

[2] V. Gabeur, C. Sun, K. Alahari, and C. Schmid, "Multi-modal transformer for video retrieval," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IV 16*. Springer, 2020, pp. 214–229.

[3] K. Li, Y. Zhang, K. Li, Y. Li, and Y. Fu, "Visual semantic reasoning for image-text matching," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 4654–4662.

[4] Z. Wang, Y.-L. Sung, F. Cheng, G. Bertasius, and M. Bansal, "Unified coarse-to-fine alignment for video-text retrieval," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 2816–2827.

[5] C. Lin, A. Wu, J. Liang, J. Zhang, W. Ge, W.-S. Zheng, and C. Shen, "Text-adaptive multiple visual prototype matching for video-text retrieval," 2022.

[6] 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.

[7] A.-M. Oncescu, A. Koepke, J. F. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries," *arXiv preprint arXiv:2105.02192*, 2021.

[8] 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.

[9] 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.

[10] 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.

[11] D. Li, I. K. Sethi, N. Dimitrova, and T. McGee, "Classification of general audio data for content-based retrieval," *Pattern Recogn. Lett.*, vol. 22, no. 5, p. 533–544, apr 2001. [Online]. Available: https://doi.org/10.1016/S0167-8655(00)00119-7

[12] X. Cheng, H. Lin, X. Wu, F. Yang, and D. Shen, "Improving video-text retrieval by multi-stream corpus alignment and dual softmax loss," 2021.

[13] Y. Park, M. Azab, B. Xiong, S. Moon, F. Metze, G. Kundu, and K. Ahmed, "Normalized contrastive learning for text-video retrieval," 2022.

[14] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transportation distances," 2013.

[15] 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*, 2019, pp. 119–132.

[16] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 736–740.

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 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. 4171–4186.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[20] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," 2020.

[21] X. Zhuang, X. Zhu, H. Hu, J. Yao, W. Li, C. Yang, L. Wang, N. Feng, and D. Xu, "Residual swin transformer unet with consistency regularization for automatic breast ultrasound tumor segmentation," in *2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 3071–3075.

[22] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017.

[23] Z. Zhu, X. Cheng, H. Li, Y. Li, and Y. Zou, "Dance with labels: Dual-heterogeneous label graph interaction for multi-intent spoken language understanding," in *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, ser. WSDM '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 1022–1031. [Online]. Available: https://doi.org/10.1145/3616855.3635782

[24] X. Zhuang, X. Cheng, and Y. Zou, "Towards explainable joint models via information theory for multiple intent detection and slot filling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 17, 2024, pp. 19 786–19 794.

[25] Z. Zhu, X. Zhuang, Y. Zhang, D. Xu, G. Hu, X. Wu, and Y. Zheng, "Tfcd: Towards multi-modal sarcasm detection via training-free counterfactual debiasing," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, 2024.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997.

[27] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," 2021.

[28] Z. Wang, Y. Gou, J. Li, Y. Zhang, and Y. Yang, "Region semantically aligned network for zero-shot learning," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, ser. CIKM '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 2080–2090. [Online]. Available: https://doi.org/10.1145/3459637.3482471

[29] Z. Wang, Y. Gou, J. Li, L. Zhu, and H. T. Shen, "Language-augmented pixel embedding for generalized zero-shot learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 33, no. 3, pp. 1019–1030, 2023.

[30] X. Zhuang, Z. Wang, X. Cheng, Y. Xie, L. Liang, and Y. Zou, "Macsc: Towards multimodal-augmented pre-trained language models via conceptual prototypes and self-balancing calibration," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2024.

[31] Z. Chen, Z. Zhao, Z. Zhu, R. Zhang, X. Li, B. Raj, and H. Yao, "Autoprm: Automating procedural supervision for multi-step reasoning via controllable question decomposition," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2024.

[32] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.