Interspeech 2022
18-22 September 2022, Incheon, Korea

Visually-aware Acoustic Event Detection using Heterogeneous Graphs

Amir Shirian[1], *Krishna Somandepalli*[2], *Victor Sanchez*[1], *Tanaya Guha*[3]

[1]The University of Warwick, UK
[2]Google Research, USA
[3]The University of Glasgow, UK

Abstract

Perception of auditory events is inherently multimodal relying on both audio and visual cues. A large number of existing multimodal approaches process each modality using modality-specific models and then fuse the embeddings to encode the joint information. In contrast, we employ heterogeneous graphs to explicitly capture the spatial and temporal relationships between the modalities and represent detailed information about the underlying signal. Using heterogeneous graph approaches to address the task of visually-aware acoustic event classification, which serves as a compact, efficient and scalable way to represent data in the form of graphs. Through heterogeneous graphs, we show efficiently modelling of intra- and inter-modality relationships both at spatial and temporal scales. Our model can easily be adapted to different scales of events through relevant hyperparameters. Experiments on *AudioSet*, a large benchmark, shows that our model achieves state-of-the-art performance. Our code is available at `github.com/AmirSh15/VAED_HeterGraph`

Index Terms: Acoustic event classification, graph neural network, heterogeneous graph, multimodal data.

1. Introduction

Audio perception by humans is inherently multimodal in nature. It involves processing both aural and visual cues. Visual cues are important not only for audio source localization [1], but also for improving audio perception [2]. Perceptual studies have also revealed that visual cues can even change how sound is heard [3].

The majority of existing works on learning audiovisual representations rely on maintaining a tight temporal synchrony between the visual and audio modalities [4, 5, 6]. Consider a scene of a bike moving away from the camera. The revving sound of the bike fades as it moves away. While an audio-only-based model may not be capable of detecting the fading sound as 'bike', taking into account the bike as a visual cue, it is possible to identify the event as 'motorbike running'. Computer vision-inspired models are common [7, 8, 9], where two augmented views of a given audio/audiovisual sample are fed to a shared 'backbone', followed by optimizing a contrastive loss [10, 11, 12, 13, 14], distillation [14, 15], quantization [4] or information maximization [16, 17]. However, the vision-inspired audio representation learning methods do not take full advantage of the temporal information available in video data or the complementary knowledge between modalities. Another difficult aspect of such approaches is that data augmentation functions, being vision-inspired, are not often well-suited to a multimodal input.

Heterogeneous graphs are a compact, efficient. and scalable way to represent data involving multiple different entities and their relations [18, 19]. Modelling the interaction of entities (including modalities) with heterogeneous graphs is a relatively new paradigm. Multimodal heterogeneous graphs have been successfully used to address various problems in computer vision and natural language processing, such as visual-question answering [20], multimedia recommendation [21, 18], audio-visual sentiment analysis [22], and cross-modal retrieval [19]. Multimodal heterogeneous graphs lead to a closer coupling between concepts in multiple modalities, resulting in a significant performance improvement over previous methods [20, 21, 22, 18]. Motivated by the success of graph-based methods in multimodal problems , we propose a heterogeneous graph-based approach to learn visually-aware audio representations.

In this paper, we propose a visually-aware audio representation learning approach based on heterogeneous graphs (see Fig.1 for an overview) in the context of acoustic event classification. Our heterogeneous graph model creates a shared space for audio and visual modalities that takes advantage of their spatial and temporal relationships explicitly. We first model the input audiovisual clip as a heterogeneous graph with two subgraphs, one for each modality with edges capturing inter- and intra-modality relationships. We next develop a heterogeneous graph neural network which is able to capture rich audio representation incorporating complementary information from the visual information. Our contributions are as follows:

- We develop a graph construction method for converting an audiovisual clip to a multimodal heterogeneous graph.
- We propose a novel heterogeneous graph neural network (HGNN) that can capture modality-specific information as well as complementary information between modalities.
- We demonstrate improved performance by our model for the task of acoustic event classification on the large benchmark AudioSet dataset.

2. Proposed Approach

This section describes our proposed approach for visually-aware audio representation learning. First, we construct heterogeneous graphs to represent the audiovisual data consisting of modality-specific subgraphs and inter-modality edges. Next, we propose a heterogeneous graph neural network (HGNN) architecture that performs graph classification in the context of acoustic event classification.

2.1. Heterogeneous graph construction

Our first task is to construct a heterogeneous graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, O, R)$, where $\mathcal{V}$ represents the set of nodes, $\mathcal{E}$ the set of edges, O is the set of node types (object/modality), and R is the

10.21437/Interspeech.2022-10670

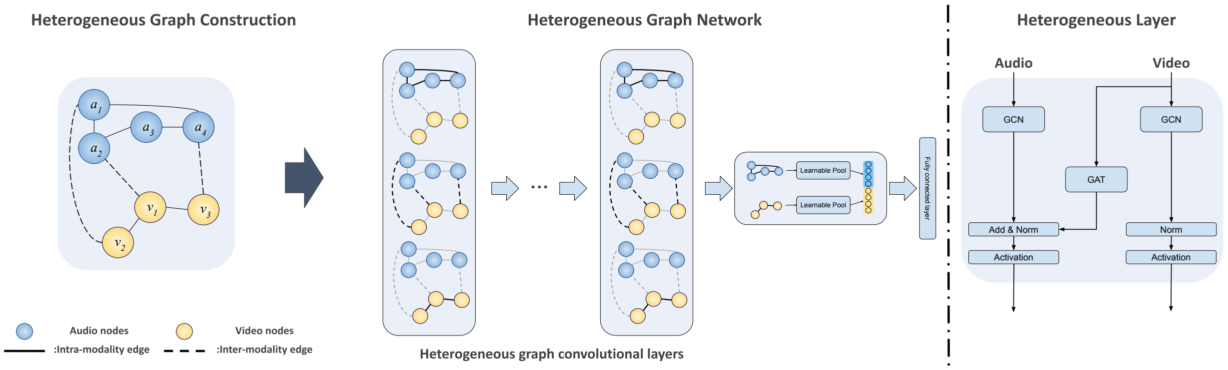


Figure 1: *(Left)* ***Heterogeneous graph architecture****. We split the input audio and video clip into Q and P overlapping segments and then construct the heterogeneous graph containing intra- and inter-modality edges between nodes. Each edge type is considered and processed by the corresponding GNN. For both audio and video modalities, heterogeneous graph convolution layers are utilised to extract the embedding for each node. Separate learnable pooling modules are then used to capture the overall graph representation. (Right)* ***Heterogeneous graph layer*** *has two independent audio and video flows taking into account intra-modality edges, as well as an attention layer connecting video nodes to audio nodes considering inter-modality edges.*

set of edge types, where $|O| + |R| > 2$. Each node $v \in \mathcal{V}$ is associated with a node type and each edge $e \in \mathcal{E}$ is associated with an edge type.

Given an audiovisual input, we uniformly divide the video frames and the audio into P and Q segments (see Fig.2). The segments are used for feature extraction. Then, given the video and the audio segments, we construct a heterogeneous graph with node sets $\mathcal{V}^v = \{v_i\}_{i=1}^{P}$ and $\mathcal{V}^a = \{a_i\}_{i=1}^{Q}$, with edge sets $\mathcal{E} = \{\mathcal{E}_{vv}, \mathcal{E}_{aa}, \mathcal{E}_{va}\}$, which represent edges between video-only nodes, audio-only nodes, and between audio-video nodes respectively. These corresponding adjacency matrices are denoted as $\mathbf{A}_v$, $\mathbf{A}_a$, and $\mathbf{A}_{va}$. Each node $v_i \in \mathcal{V}^v$ corresponds to a video segment and its associated feature vector is $\mathbf{n}_i^v \in \mathbb{R}^{d_v}$. Similarly, an audio node $a_i \in \mathcal{V}^a$ is associated with feature vector $\mathbf{n}_i^a \in \mathbb{R}^{d_a}$. Since the graph structure is not naturally defined here, we propose to add inter- and intra-modality edges (see Fig. 2). Additionally, Our graph has two parameters for each edge type, i.e, for $\mathcal{E}_{vv}, \mathcal{E}_{aa}, \mathcal{E}_{va}$: (i) *span across time* and (ii) *dilation*. The former denotes the number of nodes connected to each node in the temporal direction, whereas the latter denotes leaps between nodes. In total, we have six hyperparameters for graph construction.

2.2. Heterogeneous graph neural network (HGNN)

Given heterogeneous graphs $G_1, ..., G_N$ and their ground-truth labels $\mathbf{y}_1, ..., \mathbf{y}_N$, the task is to learn a d-dimensional graph representation $\mathbf{h}_{G_i} \in \mathbb{R}^d$ that captures rich structural and semantic information in G_i.

The key idea of most GNNs is to aggregate feature information from a node's neighbours and then update the node feature vector:

$$\mathbf{H}^{k+1} = \sigma\big(\mathbf{A}\mathbf{H}^k\mathbf{W}\big) \tag{1}$$

where $\mathbf{W}^{(k)}$ is the weight matrix for the k^{th} layer of the GNN, σ is a non-linear activation function, such as ReLU, and k is the layer number ($k = 0, \cdots K$). Because of the various node and edge types, this approach is not directly applicable to our heterogeneous graphs. Previous studies utilise meta-paths for processing heterogeneous graphs [23, 24], which has been shown to be inadequate to properly exploit the information provided by node and edge types [25]. To overcome this, we use separate GNNs for processing different edge types.

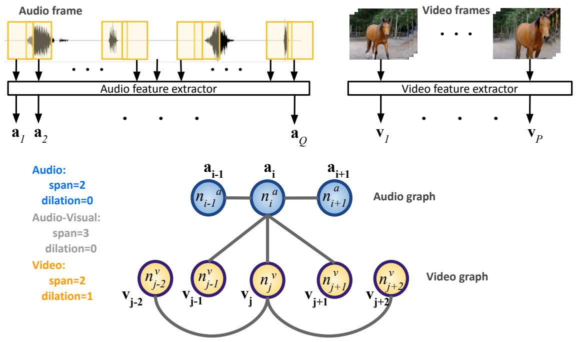


Figure 2: *Heterogeneous graph construction process. For simplicity, the edges are only shown for v_i and v_j. Similar connections are added for each node.*

Our HGNN has three flows of information corresponding to the intra- and inter-modality edges as shown in Fig.1. Audio and video flow process the audio and video nodes by considering only intra-modality edges ($\mathcal{E}_{vv}, \mathcal{E}_{aa}$) between audio and video nodes, respectively. The third flow carries audio-related information from video nodes to audio nodes for the inter-modality edges ($\mathcal{E}_{va}$):

$$\begin{aligned} \mathbf{n}_{l+1}^a &= \mathrm{GNN}_{\theta_1}\big(\mathbf{n}_l^a, \mathbf{A}_a\big) + \mathrm{GNN}_{\theta_2}\big(\mathbf{n}_l^v, \mathbf{A}_{va}\big) \\ \mathbf{n}_{l+1}^v &= \mathrm{GNN}_{\theta_3}\big(\mathbf{n}_l^v, \mathbf{A}_v\big) \end{aligned} \tag{2}$$

where $\mathbf{n}_l^a$ and $\mathbf{n}_l^v$ are audio and video node features in layer l, and GNN is a graph-based neural network such as GCN [26] or GAT [27]. The video nodes are only updated using video nodes from the previous layer, as demonstrated in the Eq. 2. As audio is the primary source of information in this application, unlike the video branch, the audio nodes are updated using both the audio and video nodes from the preceding layer, bringing information from the video to the audio modality.

Our objective is to classify entire graphs, as opposed to the more common task of classifying each node. Hence, we seek a *graph-level* representation $\mathbf{h}_G \in \mathbb{R}^d$ as the output of our network. This can be obtained by pooling the node-level representations $\mathbf{n}_K^a$, $\mathbf{n}_K^v$ at the K-th layer before passing them to the classification layer (see Fig.1). Common choices for pooling functions in the graph domain are `mean`, `max`, and `sum` pooling [26]. Max and mean pooling often fail to preserve the underlying information about the graph structure, while sum pooling has been shown to be a better alternative [28]. However, all these pooling functions treat adjacent nodes with equal importance, which may not be optimal. To this end and following [29], we propose to *learn* a pooling function Ψ that combines the node embeddings from the K-th layer to produce an embedding for the entire graph. The pooling layer for each modality is thus defined as follows:

$$\mathbf{h}_G = \Big[\Psi_a(\mathbf{n}_K^a) \,|\, \Psi_v(\mathbf{n}_K^v)\Big] = \mathbf{n}_K^a \mathbf{p}^a + \mathbf{n}_K^v \mathbf{p}^v \qquad (3)$$

where $\mathbf{p}^a$ and $\mathbf{p}^v$ are learnable weights to combine node-level embeddings to obtain a graph-level embedding for audio and video nodes. The overall heterogeneous graph network is trained with focal loss $\mathcal{L}$ as we have a unbalanced dataset:

$$\mathcal{L} = -\sum_n (1 - \mathbf{y}_n)^\gamma \log \tilde{\mathbf{y}}_n. \qquad (4)$$

3. Experiments

In this section, we first discuss the dataset used for benchmarking and feature extraction details. We then present experimental results and analysis to evaluate the performance of the proposed HGNN architecture.

3.1. Dataset

We use a large scale weakly labelled dataset **AudioSet** [30], which contains audio segments from YouTube videos. We work with 33 categories from the balanced set that have high rater confidence score ($\{0.7, 1.0\}$). This yields a training set of 82,410 clips. For a fair comparison with baseline methods, we also use the original evaluation set, which has 85,487 test clips.

3.2. Feature Encoder

Audio Encoder. To extract the audio node features, each audio clip is divided into 960 ms segments with 764 ms overlap. For each segment, a log-mel spectrogram is computed by taking its short-time Fourier transform using a frame of 25 ms with 10ms overlap, 64 mel-spaced frequency bins, and log-transforming the magnitude of each bin. This creates log-mel spectrograms of dimensions 96 × 64, which are the input to the pre-trained VGGish network [31]. We use the 128-dimensional features extracted by the VGGish network for each log-mel spectrogram. **Video Encoder.** Each video is segmented into non-overlapping 250 ms chunks to extract the video node features. The 1024-dimensional feature is then obtained by feeding each segment into an off-the-shelf 3D convolution network, S3D [32] (trained with self-supervision[33]). Note that our method is not limited to these pre-trained embeddings and can work with any generic embeddings for both audio and video.

3.3. Implementation Details

Each video clip produces a heterogeneous graph with $P = 40$ audio and $Q = 100$ video nodes, where each node corresponds to a 960 ms length audio or 250 ms length video segment. We repeat our experiments 10 times with different seeds and report both mAP (mean average precision) and ROC-AUC (area under the ROC curve) values. Our network weights are initialized following the Xavier initialization. We used Adam optimizer with a learning rate of 0.005, a decay rate of 0.1 after 1500 iterations, and 1000 warm-up iterations for all experiments. We set $\gamma = 2$ (see Eq. 4). The graph construction hyper-parameters are explored heuristically and set to *span audio* = 6, *dilation audio* = 3, *span video* = 4, *dilation video* = 4, *span audio-visual* = 3, and *dilation audio-visual* = 1 for all experiments. For graph neural network, we select regular GCNs [26] for each modality branch and a GAT [27] for fusing information from video to audio branch, resulting in 4 heterogeneous layers (Fig. 1) with a hidden size of 512 for all layers. We use Pytorch on an NVIDIA RTX-2080Ti GPU.

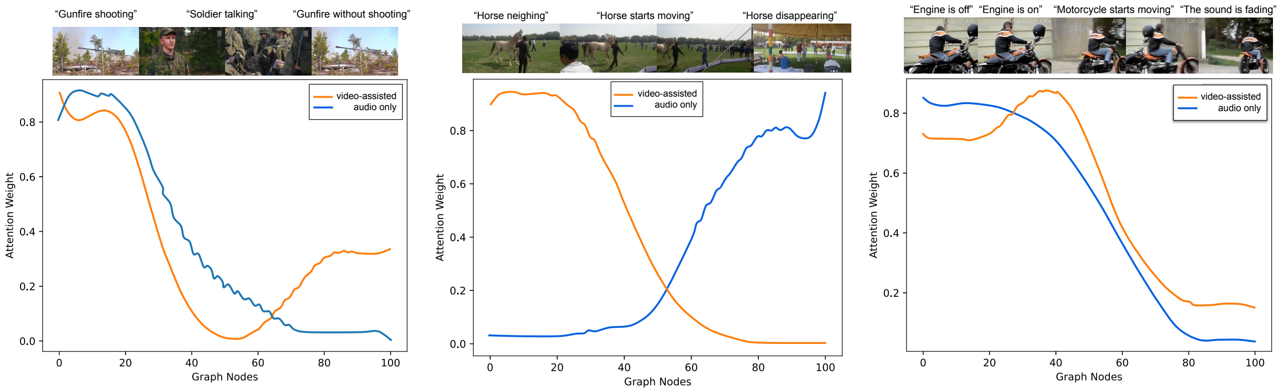
Table 1: *Acoustic event classification results on* ***AudioSet***

Model	**mAP**	**ROC-AUC**	**Params**
Ours audio only	0.42 ± 0.01	0.90 ± 0.00	1.4M
Ours video only	0.15 ± 0.02	0.75 ± 0.01	1.5M
Ours both	$\mathbf{0.50} \pm 0.01$	0.93 ± 0.00	2.1M
Baselines			
ResNet-1D audio only	0.35 ± 0.01	0.90 ± 0.00	40.4M
ResNet-1D both	0.38 ± 0.03	0.89 ± 0.02	81.2M
LSTM audio only	0.40 ± 0.00	0.90 ± 0.00	0.8M
State-of-the-art			
DaiNet [34]	0.25 ± 0.07	-	1.8M
Spectrogram-VGG	0.26 ± 0.01	-	6M
VATT [35]	0.39 ± 0.02	-	87M
SSL graph [36]	0.42 ± 0.02	-	218K
Wave-Logmel [37]	0.43 ± 0.04	-	81M
AST [38]	0.44 ± 0.00	-	88M

3.4. Results and analysis

Baselines. We compare our method with a number of fully and self-supervised models, as tabulated in Table 1. The Spectrogram-VGG model is the same as configuration A in [39], with only one change: the final layer is a softmax with 33 units. The feature for each audio input to the VGG model is a log-mel spectrogram of dimensions 96×64 computed by averaging across non-overlapping segments of length 960ms. We also compared our method with a graph-based work. Each node in this work represents an audio clip, and a KNN subgraph has been created, as well as a GNN that is trained using graph self-supervised proxy tasks [36]. We also use the two popular spatial and temporal network architectures, ResNet-1D [40] and LSTM, with pretrained embedding features for both audio and video as input, to further investigate the superiority of our graph modelling. All baseline hyper-parameters are set to the values published in the original papers. Note that we do not utilise any data augmentation, despite the fact that other methods used powerful data augmentations. Additionally, all of the baselines have been retrained using the same classes as our model.

Results. Table 1 reports the mAP and ROC-AUC (averaged over 10 runs with different seeds) values with standard deviation for each model and their variants. It compares the performance of our model with different independent modalities and strong baselines with that of the heterogeneous model in terms of mAP. The heterogeneous graph model outperforms the homogeneous graph and non-graph models. Our method leverages the pre-trained features as node attributes. Thus, to check the performance of our graph-based model, two strong baselines, ResNet-1D and LSTM, have been selected. Compared to these methods, our homogeneous graph sub-models achieve a superior mAP score that demonstrates the effectiveness of our graph-based



(a) Gunshot, gunfire

(b) Horse neighing

(c) Motorcycle starting

Figure 3: *Qualitative results showing attention weights corresponding to the audio nodes for with (in orange) and without (in blue) video supervision. Each node represents a segment of 100-millisecond duration and the ground-truth label for each video is provided below. Attention values were normalized and re-scaled to [0,1] range. (a) This video begins with a strong machine firing sound. After that, a soldier is interrogated, followed by footage of troops. Finally, the machine appears again but is not fired. Even without the associated sound of shooting, the video-assisted audio nodes are able to recognise the firing machine towards the end by assigning higher attention weights to these moments. (b) A horse begins neighing and moves away from the camera. As it moves, the sound fades. The horse is no longer visible or audible as time elapses. The audio-only model incorrectly detects these moments by assigning high attention values, while the video-assisted model correctly discards these moments. (c) A video of a motorcycle moving. The video-assisted attention weights suggest that our model can capture additional meaningful patterns, such as the engine start. Furthermore, as the engine sound fades, the attention weights corresponding to the audio-only model decrease, and the video-assisted attention weights have relatively higher values indicating that the video information extracted by our model is complementary to the audio event information.*

modelling strategy. Furthermore, when compared to other baselines, our heterogeneous graph-based model achieves the greatest ROC-AUC score (0.93), implying more trustworthy predictions at various thresholds. When compared with the other supervised models, our heterogeneous graph model outperforms Spectrogram-VGG and DaliNet [34]. Our model also has significantly fewer learnable parameters compared with the recent transformer-based architectures, VATT and AST.

Ablation experiments. We perform exhaustive ablation experiments to investigate the contribution of each component we propose to build our heterogeneous graph neural network. Table 2 presents the ablation results on the AudioSet dataset. We observe that each new component brings improvement. In all experiments, model performance is measured with mAP to quantify the recognition rate. The introduction of the heterogeneous graph increases the recognition rate by about 9%; when combined with our new graph attentional convolution layer between modalities (right half of Fig. 1), the performance increases to 0.49. Adding the learnable pooling brings up the mAP score to 0.50. Removing the learnable pooling however reduces the performance by about 3% and 1% for audio-only and video-only models, respectively. The ablation results show that each of the proposed components in our architecture is important, and contributes positively towards the overall model performance.

Qualitative results. We display how our model attends to different nodes to gain insights into its learning process. Because each video clip is divided into 100ms segments, each node represents a 100ms time window. In Fig. 3, we show the attention weights corresponding to audio nodes in cases of with and without video supervision for three input videos from the test set with distinct acoustic classes. Then, for each video, we sample four frames and display them on top of each figure to provide more visual information. This gives rise to *salient* nodes for each input. The results show that the proposed model can extract visually complementary information to an audio event from heterogeneous graphs as input.

Table 2: *Ablation experiments on the AudioSet dataset. Each new component in our heterogeneous network contributes towards its performance.*

Audio	Video	Attn	Learned p	mAP
✓	-	-	-	0.38
✓	-	-	✓	0.41
-	✓	-	-	0.12
-	✓	-	✓	0.13
✓	✓	-	-	0.49
✓	✓	✓	-	0.49
✓	✓	✓	✓	**0.50**

4. Conclusion

In this paper, we introduced the idea of hetergeneous graphs to model audio data with visual cues. We proposed a compact and efficient graph-based architecture that learns audio representations effectively in the context of acoustic event classification. We transformed an audiovisual input to a heterogeneous graph with different learnable hyper-parameters capturing intra and inter modalities connections in both spatial and temporal domains. Our heterogeneous graph model produces higher or comparable performance to the state-of-the-art on a popular benchmark dataset, the AudioSet. Our current model relies on pre-trained embeddings, which gives the flexibility of choosing any suitable embeddings. Nevertheless, our model can be made end-to-end trainable, which will be addressed as part of our future work.

5. References

[1] H. Atilgan, S. M. Town, K. C. Wood, G. P. Jones, R. K. Maddox, A. K. Lee, and J. K. Bizley, "Integration of visual information in auditory cortex promotes auditory scene analysis through multisensory binding," *Neuron*, vol. 97, no. 3, pp. 640–655, 2018.

[2] B. Shi, W.-N. Hsu, and A. Mohamed, "Robust self-supervised audio-visual speech recognition," *arXiv preprint arXiv:2201.01763*, 2022.

[3] H. McGurk and J. MacDonald, "Hearing lips and seeing voices," *Nature*, vol. 264, no. 5588, pp. 746–748, 1976.

[4] H. Alwassel, D. Mahajan, B. Korbar, L. Torresani, B. Ghanem, and D. Tran, "Self-supervised learning by cross-modal audio-video clustering," *NeurIPS*, vol. 33, pp. 9758–9770, 2020.

[5] Y. Asano, M. Patrick, C. Rupprecht, and A. Vedaldi, "Labelling unlabelled videos from scratch with multi-modal self-supervision," *NeurIPS*, vol. 33, pp. 4660–4671, 2020.

[6] B. Korbar, D. Tran, and L. Torresani, "Cooperative learning of audio and video models from self-supervised synchronization," *NeurIPS*, vol. 31, 2018.

[7] J.-B. Alayrac, A. Recasens, R. Schneider, R. Arandjelović, J. Ramapuram, J. De Fauw, L. Smaira, S. Dieleman, and A. Zisserman, "Self-supervised multimodal versatile networks," *NeurIPS*, vol. 33, pp. 25–37, 2020.

[8] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, and Y. Cui, "Spatiotemporal contrastive video representation learning," in *CVPR*, 2021, pp. 6964–6974.

[9] S. Ma, Z. Zeng, D. J. McDuff, and Y. Song, "Active contrastive learning of audio-visual video representations," in *ICLR*, 2021.

[10] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," in *ICASSP*. IEEE, 2021, pp. 3875–3879.

[11] S. Ma, Z. Zeng, D. McDuff, and Y. Song, "Contrastive learning of global and local video representations," *NeurIPS*, vol. 34, 2021.

[12] J. Jiao, Y. Cai, M. Alsharid, L. Drukker, A. T. Papageorghiou, and J. A. Noble, "Self-supervised contrastive video-speech representation learning for ultrasound," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2020, pp. 534–543.

[13] S. Jenni and H. Jin, "Time-equivariant contrastive video representation learning," in *ICCV*, 2021, pp. 9970–9980.

[14] Y. Chen, Y. Xian, A. Koepke, Y. Shan, and Z. Akata, "Distilling audio-visual knowledge by compositional contrastive learning," in *CVPR*, 2021, pp. 7016–7025.

[15] M. Liu, X. Chen, Y. Zhang, Y. Li, and J. M. Rehg, "Attention distillation for learning video representations," in *31st British Machine Vision Conference 2020, BMVC*, 2020.

[16] A. Shukla, S. Petridis, and M. Pantic, "Learning speech representations from raw audio by joint audiovisual self-supervision," *arXiv preprint arXiv:2007.04134*, 2020.

[17] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, "Barlow twins: Self-supervised learning via redundancy reduction," in *ICML*. PMLR, 2021, pp. 12 310–12 320.

[18] Q. Wang, Y. Wei, J. Yin, J. Wu, X. Song, L. Nie, and M. Zhang, "Dualgnn: Dual graph neural network for multimedia recommendation," *IEEE Transactions on Multimedia*, 2021.

[19] S. Qian, D. Xue, H. Zhang, Q. Fang, and C. Xu, "Dual adversarial graph neural networks for multi-label cross-modal retrieval," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI*, 2021, pp. 2440–2448.

[20] R. Saqur and K. Narasimhan, "Multimodal graph networks for compositional generalization in visual question answering," *NeurIPS*, vol. 33, pp. 3070–3081, 2020.

[21] Y. Wei, X. Wang, L. Nie, X. He, R. Hong, and T.-S. Chua, "Mmgcn: Multi-modal graph convolution network for personalized recommendation of micro-video," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1437–1445.

[22] X. Yang, S. Feng, Y. Zhang, and D. Wang, "Multimodal sentiment detection based on multi-channel graph neural networks," in *IJCNLP*, 2021, pp. 328–339.

[23] X. Fu, J. Zhang, Z. Meng, and I. King, "Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding," in *Proceedings of The Web Conference 2020*, 2020, pp. 2331–2341.

[24] Z. Hu, Y. Dong, K. Wang, and Y. Sun, "Heterogeneous graph transformer," in *Proceedings of The Web Conference 2020*, 2020, pp. 2704–2710.

[25] Q. Lv, M. Ding, Q. Liu, Y. Chen, W. Feng, S. He, C. Zhou, J. Jiang, Y. Dong, and J. Tang, "Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1150–1160.

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

[27] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[28] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *ICLR*, 2019.

[29] A. Shirian, S. Tripathi, and T. Guha, "Dynamic emotion modeling with learnable graphs and graph inception network," *IEEE Transactions on Multimedia*, 2021.

[30] 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 *ICASSP*, 2017, pp. 776–780.

[31] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "Cnn architectures for large-scale audio classification," in *ICASSP*, 2017, pp. 131–135.

[32] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning for video understanding," *arXiv preprint arXiv:1712.04851*, vol. 1, no. 2, p. 5, 2017.

[33] T. Han, W. Xie, and A. Zisserman, "Self-supervised co-training for video representation learning," *NeurIPS*, vol. 33, pp. 5679–5690, 2020.

[34] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, "Very deep convolutional neural networks for raw waveforms," in *ICASSP*. IEEE, 2017, pp. 421–425.

[35] H. Akbari, L. Yuan, R. Qian, W.-H. Chuang, S.-F. Chang, Y. Cui, and B. Gong, "Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text," *arXiv preprint arXiv:2104.11178*, 2021.

[36] A. Shirian, K. Somandepalli, and T. Guha, "Self-supervised graphs for audio representation learning with limited labeled data," *arXiv preprint arXiv:2202.00097*, 2022.

[37] 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.

[38] Y. Gong, Y.-A. Chung, and J. Glass, "Ast: Audio spectrogram transformer," *arXiv preprint arXiv:2104.01778*, 2021.

[39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, Y. Bengio and Y. LeCun, Eds., 2015.

[40] S. Hong, Y. Xu, A. Khare, S. Priambada, K. Maher, A. Aljiffry, J. Sun, and A. Tumanov, "Holmes: health online model ensemble serving for deep learning models in intensive care units," in *ACM SIGKDD*, 2020, pp. 1614–1624.