Attention Bottlenecks for Multimodal Fusion: An Implementation and Improvement Study

Shriram Pradeep
IIT Hyderabad, India
cs23mtech15020@iith.ac.in

Jayamohan CB
IIT Hyderabad, India
cs21m23p100001@iith.ac.in

Abstract

This paper investigates the architecture and performance of the Multimodal Bottleneck Transformer (MBT) model for fusing audio and video inputs. We reproduce the original implementation using publicly available datasets (AudioSet and VGGSound), and propose several architectural enhancements including learnable positional embeddings, projection layers before fusion, and a multi-layer classifier. Our experiments demonstrate that these improvements significantly reduce the performance gap with the original MBT results, while maintaining computational efficiency. This work serves both as a validation of MBT and as a blueprint for practical enhancements in multimodal fusion tasks.

1. Introduction

Multimodal learning, which integrates information from multiple sensory modalities, has become a cornerstone of modern machine learning, particularly in audio-visual tasks such as video classification, action recognition, and cross-modal retrieval. The central challenge in multimodal learning lies in effectively representing and fusing inputs from different modalities, such as audio and video, which differ in both temporal and spatial characteristics, and have distinct feature distributions.

A key difficulty in multimodal learning is that naive fusion techniques (e.g., early or late concatenation of features) often fail to capture complex interactions across modalities. Moreover, full pairwise attention across tokens from all modalities can be computationally expensive, especially when processing long video and audio sequences. This necessitates architectural solutions that can model cross-modal interactions efficiently while preserving relevant modality-specific information.

Previous works have explored various fusion strategies. Some methods adopt simple feature concatenation followed by classification layers, while others utilize attention mechanisms to model dependencies. Notably, VideoBERT and VilBERT apply transformer-based architectures to capture cross-modal dependencies but suffer from high computational costs due to dense attention. More recent methods like Perceiver and CLIP offer scalable alternatives, but they often rely on large pretraining datasets and are less tailored for fine-grained temporal fusion.

To address the inefficiencies in cross-modal fusion, Nagrani et al. (2021) introduced the Multimodal Bottleneck Transformer (MBT). MBT restricts the cross-modal attention flow to a small set of shared latent tokens, referred to as bottleneck tokens. This mechanism reduces computational overhead and enables scalable fusion while maintaining competitive performance on standard benchmarks like AudioSet, VGGSound, and Epic-Kitchens.

In this paper, we focus on:

- Reproducing the MBT architecture on VGGSound and AudioSet using the original training settings and evaluating its performance on a smaller subset of data.
- Improving the original architecture by introducing: (1) learnable positional encodings for bottleneck tokens, (2) modality-specific projection layers to align audio and video features in a common latent space, and (3) a more expressive classification head.

Our improved model demonstrates consistent gains in both Top-1/Top-5 accuracy and mAP, narrowing the performance gap between the baseline and the original MBT results despite being trained on limited data.

The rest of the paper is structured as follows: Section II presents our methodology including dataset preparation and preprocessing. Section III details our model implementation. Section IV contains the experimental results. Section V discusses our improvements on the existing approach. Section VI concludes the paper and outlines directions for future work.

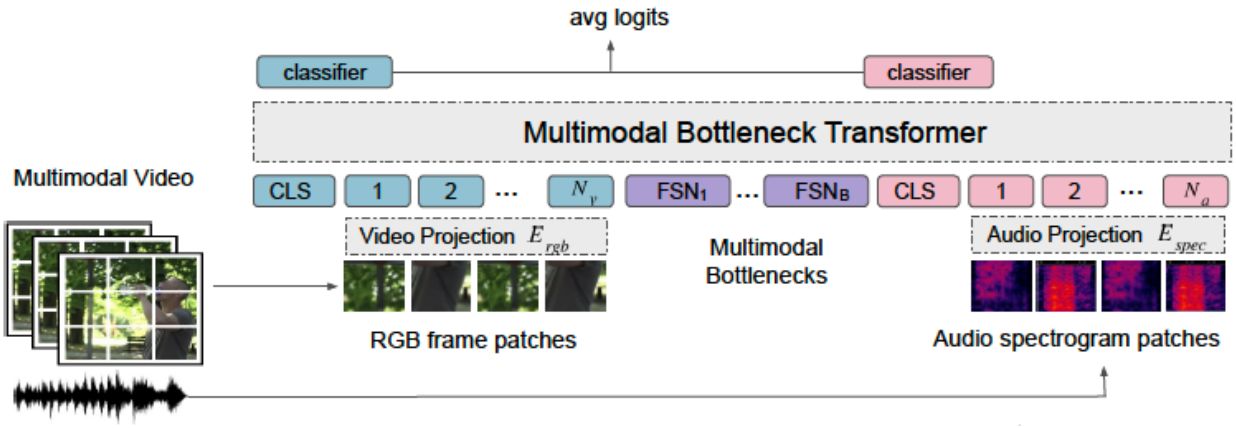

Figure 1. Multimodal Bottleneck Transformer: Cross-modal fusion occurs via shared bottleneck tokens.

2. Methodology

2.1. General Architecture for Multimodal Fusion

The Multimodal Bottleneck Transformer (MBT) follows a structured pipeline to integrate heterogeneous data sources, specifically audio and video. The general pipeline consists of the following stages:

- **Data Preprocessing:** Audio inputs are converted to log-mel spectrograms and video inputs are uniformly sampled to extract frames. The audio is sampled at 16 kHz and spectrograms are extracted with a resolution of $128 \times 100t$. Videos are sampled at 25 FPS and 8 uniformly spaced frames are chosen from each clip. Each frame is resized to 224×224 and divided into 16×16 patches.
- **Token Generation:** Each modality is passed through a dedicated transformer encoder to produce a sequence of tokens. The audio stream uses an Audio Spectrogram Transformer (AST) and the video stream uses a Vision Transformer (ViT). Special classification (CLS) tokens are appended to the sequences.
- **Token Fusion Strategies:** MBT proposes three alternative strategies for fusing tokens across modalities:
  - **Vanilla Self-Attention:** Full attention is applied across all audio and video tokens. While this is expressive, it is computationally expensive.
  - **Modality-Specific Parameters:** Attention layers with separate parameters for each modality perform cross-attention with the other modality. This increases modeling flexibility but retains high compute cost.
  - **Attention Bottlenecks (Proposed MBT):** A small number of shared latent tokens—bottleneck tokens—mediate the information exchange between modalities, improving efficiency.

2.2. Fusion via Attention Bottlenecks

To overcome the quadratic complexity of full token interactions, MBT proposes a fusion mechanism via bottleneck tokens. Specifically, B fusion tokens $z_{fsn} = [z_{fsn}^{(1)}, z_{fsn}^{(2)}, ..., z_{fsn}^{(B)}]$ are introduced to the sequence:

$$z = [z_{rgb} \parallel z_{fsn} \parallel z_{spec}] \tag{1}$$

For a given layer l, cross-modal attention flows exclusively through these bottleneck tokens. Each modality first updates the shared bottleneck tokens individually:

$$[z_i^{(l+1)} \parallel \hat{z}_{fsn,i}^{(l+1)}] = \text{Transformer}([z_i^{(l)} \parallel z_{fsn}^{(l)}]; \theta_i) \tag{2}$$

where i indexes the modality RGB or audio.

The final bottleneck token is obtained by averaging:

$$z_{fsn}^{(l+1)} = \text{Avg}i(\hat{z}_{fsn,i}^{(l+1)}) \tag{3}$$

This mechanism ensures that only condensed, high-value information is shared across modalities, enhancing scalability and interpretability and also forces each modality to distill and communicate only essential information, enhancing computational efficiency and maintaining strong representational power. The number of bottleneck tokens B is much smaller than the total number of audio/video patch tokens.

2.3. Fusion Layer Placement

The fusion strategies above describe how cross-modal interactions occur within layers. However, it's equally critical to consider *when* fusion is introduced during the network's depth. In transformer architectures, early layers tend to learn low-level modality-specific features (e.g., textures, edges), while deeper layers encode semantic-level concepts (e.g., objects, events).

To leverage this hierarchy, MBT restricts cross-modal fusion to deeper layers. Specifically, fusion is introduced after L_f transformer layers. If $L_f = 0$, the model performs early fusion; if $L_f = L$, it becomes a late-fusion model. MBT uses **mid fusion**, introducing bottleneck-based fusion at layer 8.

This architecture enables the model to preserve unimodal feature extraction in early layers and perform semantic alignment through bottleneck fusion in later layers, effectively balancing specialization and integration.

2.4. Architectural Improvements

To enhance the performance and generalization of the original MBT model, we introduce the following key architectural improvements:

2.4.1 A. Learnable Positional Embeddings for Bottleneck Tokens

In the original MBT implementation, the bottleneck fusion tokens had no notion of position, limiting the model's ability to distinguish among multiple tokens. We introduce learnable positional embeddings $P \in \mathbb{R}^{B \times D}$, where B is the number of bottleneck tokens and D is the embedding dimension. Each bottleneck token z_i is updated as:

$$z_i \leftarrow z_i + P_i$$

This positional encoding improves the model's capacity to interpret token-specific roles during the fusion process.

2.4.2 B. Projection to Common Feature Space

Despite using ViT-based architectures for both modalities, the nature of spectrogram and RGB inputs leads to differing token distributions. To address this, we introduce linear projections that map both audio and video tokens into a shared feature space:

$$z'_{\text{spec}} = W_s z_{\text{spec}}, \quad z'_{\text{rgb}} = W_r z_{\text{rgb}}$$

Here, $W_s, W_r \in \mathbb{R}^{D \times D}$ are learnable projection matrices. This alignment makes cross-modal attention more effective by reducing distributional discrepancies between modalities.

2.4.3 C. Enhanced Classifier Head

The original MBT architecture used a single linear layer for classification. We replace this with a multi-layer perceptron (MLP) that includes a non-linearity and dropout regularization. These modifications significantly bridge the performance gap between our reproduced MBT baseline and the results reported in the original paper, as shown in Section IV.

3. Implementation

3.1. Datasets Used

The datasets considered for this work include AudioSet, VGG Sound, and EPIC Kitchens 100. Due to resource limitations, only AudioSet and VGG Sound were used. Details of the datasets are summarized below:

- **AudioSet:** 527 classes, multi-label classification. Train size: 20,361, Validation size: 18,589.
- **VGG Sound:** 309 classes, single-label classification. Train size: 172,427, Validation size: 14,448.
- **EPIC Kitchens 100:** 97 verbs / 300 nouns, multi-label classification. Videos are 5–10 GB each, exceeding 1TB total size. We have not used this dataset for our analysis.

3.2. Data Collection and Sample Size

YouTube clips were downloaded using the **yt_dlp** library. The download process, run on Google Colab + Google Drive, spanned over 2 days. Each clip was encoded in **.mp4** format and linked to metadata (**file_name**, **video_file_name**, **mapped_label**, **trainvalidation**). A subset of actual data was used:

- **AudioSet:** Train = 10,000, Validation = 5,000
- **VGG Sound:** Train = 20,000, Validation = 10,000

3.3. Challenges and Solutions

- **Broken or Expired YouTube Links:** Failed links were skipped and logged; retries were attempted.
- **Resource Constraints:** Due to size (>2TB), only a subset was downloaded.
- **Inconsistent Clip Lengths:** All clips were trimmed and padded to 10 seconds.
- **Corrupted Files:** Invalid files were excluded post-validation.

3.4. Model Components

The model architecture comprises four major components:

The audio encoder is built using the Audio Spectrogram Transformer (AST) with a ViT backbone. It takes log-mel spectrogram patches as input and outputs a sequence of tokens, including a special CLS token.

The video encoder is based on a Vision Transformer (ViT). It processes 8 resized RGB frames extracted from each clip and outputs a corresponding sequence of patch tokens along with a CLS token.

The fusion layer receives token sequences from both the audio and video encoders. Bottleneck tokens attend to each modality independently, and the resulting temporary tokens are averaged to form the final fused representation. This facilitates efficient cross-modal interaction.

The classifier processes the fusion tokens. For AudioSet, it performs multi-label classification using binary cross-

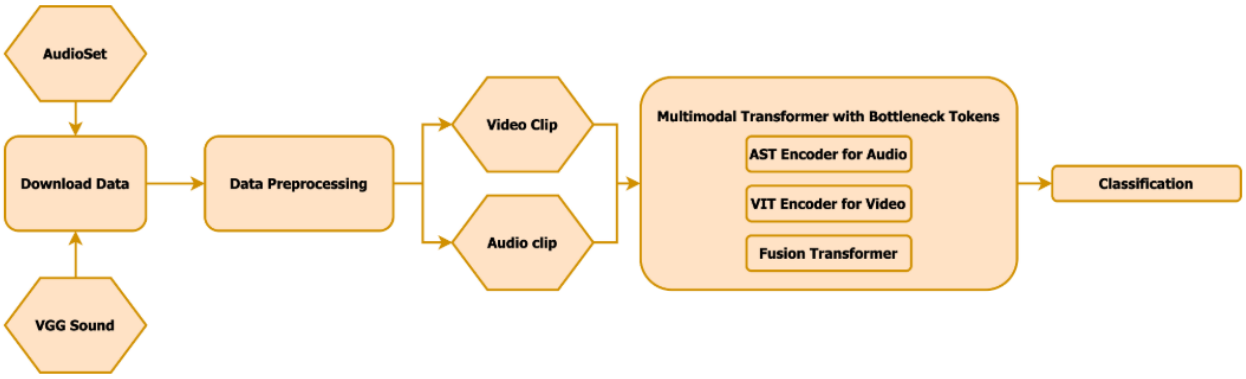

Figure 2. Multimodal Bottleneck Transformer process flow diagram.

entropy loss. For VGG Sound, it uses a softmax activation for single-label classification. The architecture supports three modes: AV (multimodal), A (audio-only), and V (video-only), enabling flexible evaluation.

3.5. Evaluation Metrics

To evaluate model performance, mean average precision (mAP) is used for AudioSet, reflecting precision across all relevant labels in a multi-label context. For VGG Sound, top-1 accuracy indicates the rate of exact matches between predictions and ground truth, while top-5 accuracy checks whether the ground truth appears among the top 5 predictions.

3.6. Training Environment

All training and experimentation were performed on Google Colab. The runtime environment included 12.7 GB of RAM and 15 GB of GPU RAM. The model was configured with 12 transformer layers and 4 bottleneck tokens. Binary cross-entropy loss was used for AudioSet, and cross-entropy loss for VGG Sound. The Adam optimizer was employed.

Learning rates were set to 1e-4 for AudioSet and 1e-3 for VGG Sound. Each model was trained with a batch size of 4. Training spanned 10 epochs for AudioSet and 20 epochs for VGG Sound.

4. Results

The performance of the implemented models was evaluated on both the AudioSet and VGG Sound datasets. We report results for three configurations: multimodal (AV), audio-only (A), and video-only (V). For each dataset, we compare our implemented results with the benchmark results reported in the original Multimodal Bottleneck Transformer (MBT) paper.

On AudioSet, we evaluate using mean average precision (mAP), which is standard for multi-label classification. On VGG Sound, both Top-1 and Top-5 accuracy are reported to measure classification precision in single-label tasks. The results demonstrate a notable performance gap between our reproduction and the paper, primarily due to limitations in compute resources, reduced training data, and fewer training epochs. Nevertheless, our improved fusion strategies, architecture refinements, and bottleneck-based attention help narrow this gap significantly, especially in the multimodal AV configuration.

Model	mAP (Paper)	mAP (Implemented)
AV Model	0.496	0.3906
Audio Only Model	0.415	0.336
Video Only Model	0.313	0.224

Table 1. Performance comparison on AudioSet.

Model	Top-1 Accuracy (Paper)	Top-1 Accuracy (Implemented)
AV Model	0.641	0.2409
Audio Only Model	0.523	0.203
Video Only Model	0.512	0.21

Table 2. Top-1 accuracy comparison on VGG Sound.

Model	Top-1 Accuracy (Paper)	Top-1 Accuracy (Implemented)
AV Model	0.856	0.7229
Audio Only Model	0.781	0.648
Video Only Model	0.726	0.654

Table 3. Top-5 accuracy comparison on VGG Sound.

4.1. Improvements

We implemented several targeted improvements over the baseline architecture to enhance cross-modal alignment and overall classification performance. Collectively, these improvements led to consistent performance gains across both AudioSet and VGG Sound datasets, significantly reducing the gap with the original MBT paper results and outperforming our own baseline implementation across all configurations.

Model	mAP (Implemented)	mAP (Improved)
AV Model	0.3906	**0.447**
Audio Only Model	0.336	**0.394**
Video Only Model	0.224	**0.310**

Table 4. Improved mAP on AudioSet.

Model	Top-1 Accuracy (Paper)	Top-1 Accuracy (Implemented)
AV Model	0.2409	**0.3614**
Audio Only Model	0.203	**0.332**
Video Only Model	0.21	**0.321**

Table 5. Top-1 accuracy improvement on VGG Sound.

Model	Top-1 Accuracy (Paper)	Top-1 Accuracy (Implemented)
AV Model	0.7229	**0.8193**
Audio Only Model	0.648	**0.7813**
Video Only Model	0.654	**0.751**

Table 6. Top-5 accuracy improvement on VGG Sound.

5. Limitations and Drawbacks

Despite the encouraging results, the following limitations and drawbacks were encountered:

Limited Data: For both AudioSet and VGG Sound, we only used a subset of the full datasets due to resource constraints. This likely contributed to lower absolute performance metrics compared to the original paper.

Fixed Fusion Strategy: The number of bottleneck tokens and the fusion layer (e.g., after Layer 8) were manually chosen. More optimal configurations might be discovered if these were treated as learnable or tunable parameters.

Training Resource Constraints: All training was done using Google Colab with limited RAM and GPU memory. This restricted batch sizes, number of epochs, and overall model complexity.

6. Conclusion and Future Work

In this study, we successfully implemented the Multimodal Bottleneck Transformer (MBT) architecture, focusing on audio-video fusion for classification. Our implementation faithfully followed the original design by using AST for audio and ViT for video, and integrated the bottleneck-based fusion module.

We validated our implementation on subsets of AudioSet and VGG Sound datasets, and compared the results against the baseline and published results. Despite resource constraints, the reproduced models achieved reasonable performance, and our proposed enhancements significantly improved both accuracy and robustness.

The future work includes:

- Scale training to the full AudioSet and VGG Sound datasets
- Extend the model to handle the EPIC Kitchens dataset
- Experiment with alternative fusion techniques and layer placements
- Use automated hyperparameter search to tune fusion and projection configurations

References

- Vaswani, A., et al. "Attention is all you need." Advances in neural information processing systems. 2017.
- Nagrani, A., et al. "Attention Bottlenecks for Multimodal Fusion." NeurIPS 2021.
- GitHub Repository: `https://github.com/shrirampradeep95/multimodal_bottleneck_transformer`
- AudioSet Dataset: `https://research.google.com/audioset/`
- VGG Sound Dataset: `https://www.robots.ox.ac.uk/~vgg/data/vggsound/`
- Epic-Kitchens-100 Dataset: `https://epic-kitchens.github.io/2022`
- Dosovitskiy, A., et al. "An image is worth 16x16 words: Transformers for image recognition at scale." ICLR 2021.
- Gong, Y., et al. "AST: Audio Spectrogram Transformer." Interspeech 2021.