Tackling Data Bias in MUSIC-AVQA: Crafting a Balanced Dataset for Unbiased Question-Answering

Xiulong Liu *†
University of Washington
xl1995@uw.edu

Zhikang Dong *†
Stony Brook University
zhikang.dong.1@stonybrook.edu

Peng Zhang
Bytedance Inc
zhang.peng@bytedance.com

1. Implementation Details

1.1. Video Pre-processing

We adhere to the LAVISH open-source code guidelines to preprocess the audio and visual frames of videos in MUSIC-AVQA and v2.0. The majority of these videos have a duration of 60 seconds. For videos shorter than 60 seconds, we extend them by repeating the last visual frame and the corresponding 1-second audio until reaching 60 seconds, as per the LAVISH guidelines. Visual frames are extracted from videos at a rate of 1fps, yielding 60 frames for each video. For audio, we sample the waveforms at a 16kHz rate. Given the large size of audio and visual frames, models cannot process all frames from a video. Thus, we implement the same down-sampling method as LAVISH, extracting every 10th visual frame from the start of the video, each paired with its corresponding 2-second audio segment. After down-sampling, we are left with 10 visual frames and 10 associated audio waveforms for each video. To accommodate the input size of the Swin-Transformer-V2 [9] (the backbone of LAVISH), we resize the image of each frame to $192 \times 192 \times 3$. For every 2-second audio segment, we compute the mel-spectrogram using a 5.2-millisecond frameshift and a kaldi fbank with 192 triangular mel-frequency bins. We then triple the mel-spectrogram along the channel dimension, resulting in a tensor of size $192 \times 192 \times 3$, same dimensions as the image input in each frame.

1.2. LAVISH [8]

For LAVISH, we use the official open-source code implementations and detail them below. The backbone of LAVISH consists of a 2-tower Swin-Transformer-V2-Large pretrained on ImageNet: one tower for visual input and the other for mel-spectrogram input. Within each layer of the two towers, two LAVISH adapters (a type of adapter [5] for audio-visual learning) are inserted. One is positioned as the residual of the Multi-Head Attention [11] module, and the other as the residual of the MLP module. Both the visual and audio branches undergo cross-attention within these adapters. For details about the LAVISH adapter, please refer to the original paper.

The output from both the visual and audio backbones is a 6×6 feature map with a channel size of 512. The audio feature map is then mean-pooled across its spatial dimensions to produce a 512-dimensional vector. This vector is forwarded to the spatial grounding module, which attends to the visual feature map for audio-visual fusion. The result of the spatial grounding is a 512-dimensional vector for each frame, with a total of 10 frames. This is then directed to the temporal grounding module, which is a single-layer Multi-Head Attention between the 512-dimensional question vector from a 2-layer LSTM [4] encoder and the spatial grounding outputs on the temporal axis. The final output from the temporal grounding module is a single 512-dimensional vector, which is subsequently concatenated with outputs from other branches. In LAVISH, both the spatial and temporal grounding modules are consistent with the methods described in the AVST work [7]. For further details, please refer to it.

1.3. AST branch

We apply the pretrained checkpoint of Audio-Spectrogram-Transformer (AST) on AudioSet ("Full AudioSet, 10 tstride, 10 fstride, with Weight Averaging (0.459 mAP)") as the backbone for our "AST" branch. For the audio spectrogram input, we follow the processing steps outlined in the AST paper. Each 2-second audio waveform segment is converted into a series of 128-dimensional log Mel filterbank (fbank) features, using a 25ms Hamming window at 10ms intervals. This results in a $128x204$ mel-spectrogram, which is then used as the input for the AST. Additionally, since the original pretrained AST is designed for a 10-second mel-spectrogram input, whose position embeddings are larger than those for a 2-second input, we adjust it by symmetrically trimming the leftmost and rightmost portions of the embedding matrix. This

*These authors contributed equally.
†Work done while interning at Tiktok

ensures the position embeddings are compatible with our 2-second inputs.

Once the audio inputs are processed through AST, we extract the hidden states from the final layer, selecting only the last hidden state to produce a 768-dimensional audio vector. A Linear layer is then added to project this audio vector down to 512 dimensions, aligning with the channel size of the visual feature maps in LAVISH. Subsequently, we employ the same grounding operations as LAVISH, using shared weights in the grounding modules. This involves computing the spatial grounding between the audio vector and the LAVISH visual feature maps. The resulting output is then grounded with the 512-dimensional question vector from the LSTM encoder (as in LAVISH) along the temporal axis. The grounding output from the AST branch matches the LAVISH branch in dimension, producing a 512-dimensional vector. This grounding output from the AST branch is concatenated with outputs from other branches to form an ensemble vector.

1.4. Cross-modal Pixel-wise Attention

The module receives two feature maps as inputs: an audio spatial map and a visual spatial map. Both maps have dimensions of $6 \times 6 \times 512$. These maps are flattened to 36×512, and cross-attention is computed between them along the spatial axis, as detailed in Section 4 of the main paper. The module's output is a 512-dimensional vector, which is then directed to the temporal grounding module (with shared weights) for question-related attention, consistent with the other two branches. This temporal grounding output remains a 512-dimensional vector and is concatenated with outputs from the other branches to form an ensemble vector. Finally, the concatenated outputs from all branches are forwarded to a 2-layer MLP with hidden sizes of 512 and 42 (the vocabulary size of candidate answers) respectively, producing the logits of the answer.

1.5. Training Details

We implement all models using PyTorch [10]. For LAVISH and AVST, we train the models using cross-entropy loss between the predicted and the ground truth answers, along with an audio-visual matching loss by sampling non-matching visual frames from other videos, as proposed in AVST [7]. Following AVST and LAVISH, we assign a weight of 0.5 to the audio-visual matching loss and 1.0 to the cross entropy loss. For our "LAST" and "LAST-Att" models, we use cross entropy loss only without audio-visual matching loss. During training, we freeze the parameters of all backbones, including the 2 Swin-Transformers in LAVISH and the AST audio encoder. For LAVISH adapters, we follow the paper to set a small learning rate of 8e-5. And we set the learning rate of 3e-6 for the grounding modules including our cross-modal pixel-wise attention module, and the final prediction layer in AVQA. We use Adam [6] optimizer to train all models. In terms of hardware configuration, models are trained on 8 NVIDIA-V100 32GB GPUs in data parallel mode. We configure the batch size to 24 for data loading.

2. Data Collections and Statistics

2.1. Data Quality Control

To ensure the quality of our collected data, we annotate all labels in conjunction with QA pairs by ourselves. Prior to data collection, we meticulously review the QA pairs across 33 templates. This helps us accurately understand the questions and their corresponding videos, minimizing inconsistent annotations stemming from misunderstandings. Upon review, we discover significant inconsistency in a predominant question template within the Audio-Visual Existential category: "Is there a voiceover". This inconsistency arises from varying interpretations of the term "voiceover" by previous annotators in MUSIC-AVQA. From the labels, it is evident that some annotators perceived a human voice layered over instrument sounds as a voiceover, while others interpreted it as a generic "off-screen sound". For consistency, we adopt the latter definition for our annotations. As a result, we adjust 13% of the annotations from this question template in the training set. For each QA pair among our additionally collected 8,136 samples, we have three individuals verify the annotation and only accept those that received unanimous agreement.

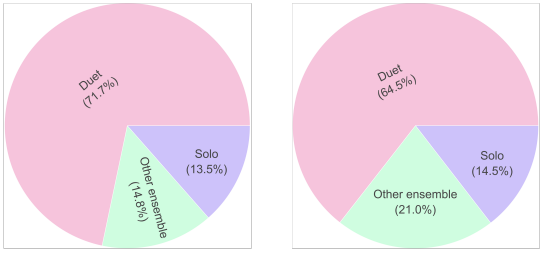
2.2. Details of Distribution After Balance

Additional QA data We provide detailed statistics for our additional QA pairs. As shown in Fig. 1, among our 8.1k QA pairs (+17.8% additional QA pairs to 45.6k in MUSIC-AVQA [7] (updated version in their work)), Audio-Visual questions constitute 52.1% of the total. This includes Audio-Visual Existential at 15.3%, Audio-Visual counting at 28.1%, and Audio-Visual Temporal at 8.7%. Visual questions account for 23.8%, with Visual Counting 17.3% and Visual Location 6.5%. Audio questions comprise 24%, with Audio Counting 21.8% and Audio Comparative 2.2%.

Additional Video Data We collect 1230 additional real videos (+16.6% more real videos than MUSIC-AVQA (7422)), sourcing from YouTube and YouTube-8M [1]. Among 1230 videos, 715 videos contain 3 or more instruments, 249 are duets and the remaining 266 are solos. With these additional videos mainly focusing on musical ensembles, we enrich the dataset with more diverse videos and a less skewed distribution of scene types. As illustrated in Fig. 2, after collection, the proportion of other ensembles has seen a large increase of 6.2%, moving from 14.8% to 21.0%.

Distribution of Each Bias Question Template Before

Figure 1. An overview of the distribution of our collected QA pairs in each question type and question template.



Scene Types Distribution: **MUSIC-AVQA** (left) v.s **MUSIC-AVQA v2.0** (right)

Figure 2. Distribution of Scene Types

and After Balance We further show distribution of each bias question template before and after balance, grouped by question type and modality type, Fig. 3, Fig. 4 and Fig. 5 show Audio-Visual questions, Visual questions and Audio questions respectively. Within each figure, we show specific counts for each answer category within the templates. As evident from the data, our QA collection considerably rectifies the majority of the skewed distribution present in the original dataset. However, we acknowledge that certain counting question templates remain biased towards fewer counts. For instance, questions asking about the number of distinct instrument types inherently lean towards smaller counts, which makes the data collection of larger counts very challenging.

3. Ablation Studies

To study the effectiveness of our proposed modules, the "AST branch" and "Cross-Modal Pixel-wise Attention", we conduct an ablation study. In this study, we implement two additional models that exclude these proposed modules.The

Table 1. Component Ablation Overview: A breakdown of components used in each method. A (✓) indicates the component is included in the method, while a (-) indicates its absence. "Swin-A" denotes the LAVISH audio branch, "AST" denotes the AST audio encoder, "CM-P-Attn" denotes cross-modal pixel-wise attention module.

Model	Swin-A	AST	CM-P-Attn
LAVISH [8]	✓	-	-
Swin-AST	-	✓	-
LAST	✓	✓	-
LAVISH-Att	✓	-	✓
LAST-Att	✓	✓	✓

Table 2. Ablation Study: Evaluation Results on Balanced Test Set: "Swin-AST" and "LAVISH-Att" v.s. Our 2 baselines and existing methods. (Ext: Existential. Cnt: Counting. Temp: Temporal. Comp: Comparative.)

Model	Total	Audio-Visual					Visual		Audio	
		Ext	Temp	Cnt	Loc	Comp	Cnt	Loc	Cnt	Comp
LAVISH [8]	73.18	73.83	60.81	73.28	65.00	63.49	81.99	80.57	84.37	58.48
Swin-AST	74.63	75.88	61.84	74.31	68.26	64.49	83.06	83.63	84.52	59.1
LAST	74.85	74.08	59.15	75.17	**69.02**	**66.12**	83.19	83.41	85.75	61.59
LAVISH-Att	75.32	75.47	**63.39**	74.37	68.37	64.94	83.72	**84.08**	**86.32**	61.74
LAST-Att	**75.44**	**76.21**	**60.60**	**75.23**	68.91	65.60	**84.12**	84.01	**86.03**	**62.52**

first model, termed "Swin-AST" model, retains only the vision branch of the LAVISH backbone, excluding the audio branch and the LAVISH adapter. This model can be viewed as an advanced version of the "AVST" model, given that it substitutes more robust backbones (In AVST, vision branch uses ResNet-18 [2] and audio branch uses Vgg-ish [3] pretrained on AudioSet). The second model, "LAVISH-Att", preserves just the LAVISH components (the 2-tower backbone, spatial grounding, and temporal grounding) and integrates our cross-modal pixel-level attention module. We train and validate both models on MUSIC-AVQA v2.0, and evaluate on the balanced test split. The ablated model components are detailed in Table 1, and evaluation results are summarized into Table 2.

As shown in the table, we observe that "Swin-AST" achieves nearly on-par results with our "LAST" baseline, and performs better than LAVISH with a +1.45% improvement. This suggests the benefits of applying robust pretrained backbones for both the visual and audio branches for AVQA task. Moreover, "LAVISH-Att", even without using a pretrained audio backbone, surpasses the LAVISH baseline by +2.14%. However, it falls short by a mere 0.12% compared to our full model, "LAST-Att". This confirms our hypothesis that integrating a fine-grained spatial cross-attention module across feature maps of both modalities can improve performance, especially when combined with the existing spatial grounding module. In our experiments, when we add only the cross-modal pixel-wise attention module and excluded the spatial grounding branch, the model underperform from the outset, plateauing at a total

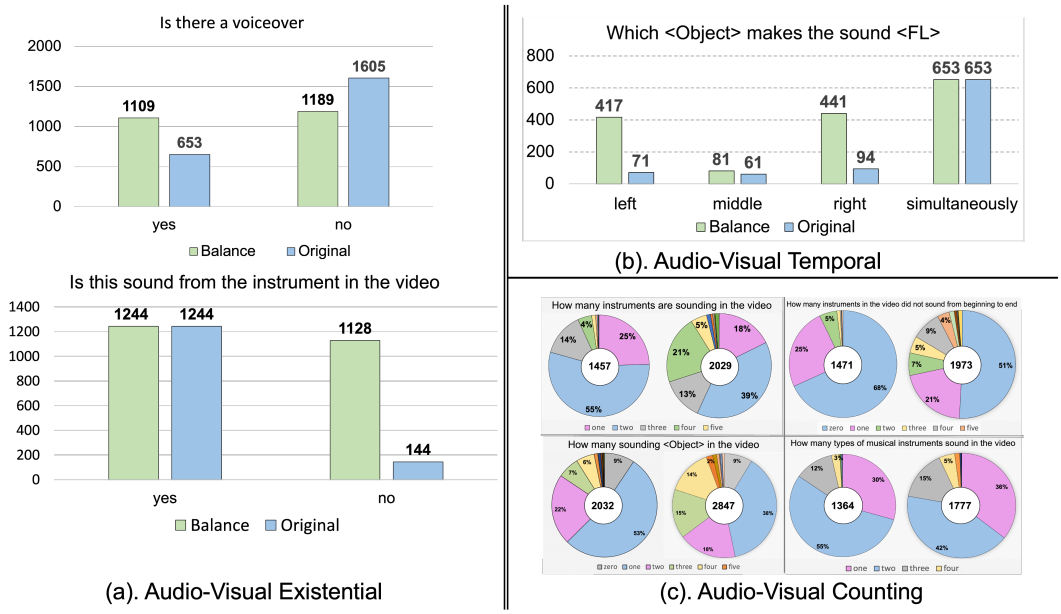


Figure 3. Distribution of Bias Audio-Visual Questions Before and After Balance

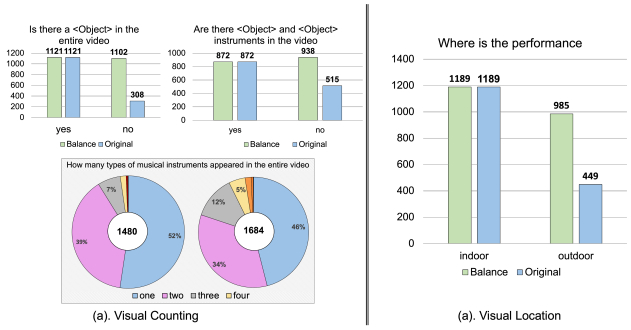


Figure 4. Distribution of Bias Visual Questions Before and After Balance

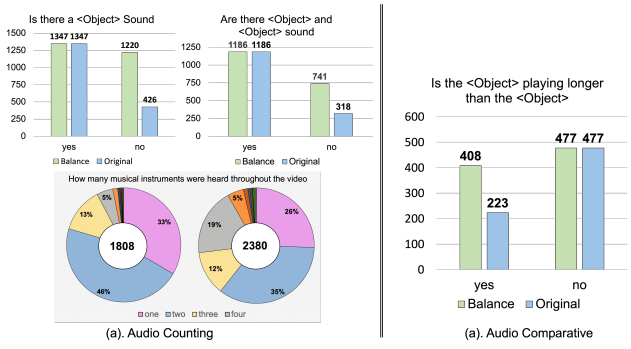


Figure 5. Distribution of Bias Audio Questions Before and After Balance

accuracy of 69.3%. We hypothesize that the fine-grained attention module captures low-level feature details, while the spatial grounding module abstracts high-level features. They complement each other to bring the optimal result.

References

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark, 2016. 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 3

[3] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson. Cnn architectures for large-scale audio classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017. 3

[4] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. 1

[5] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp, 2019. 1

[6] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. 2

[7] Guangyao Li, Yake Wei, Yapeng Tian, Chenliang Xu, Ji-Rong Wen, and Di Hu. Learning to answer questions in dynamic audio-visual scenarios, 2022. 1, 2

[8] Yan-Bo Lin, Yi-Lin Sung, Jie Lei, Mohit Bansal, and Gedas Bertasius. Vision transformers are parameter-efficient audio-visual learners, 2023. 1, 3

[9] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution, 2022. 1

[10] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. 2

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 1