MEERKAT: Audio-Visual Large Language Model for Grounding in Space and Time

Appendix

In this appendix we provide additional details about the following:

A Data Preparation Strategy

A.1 Adaptation of Public Datasets.

To collect the image-audio pairs from video-based datasets and adapt them to our setup, we carefully choose one representative image from the video. We add task-wise dataset details in Fig. 1. To this end, we design a semi-automated strategy as explained later in each task section.

A.2 Fine-grained Data Preparation

Audio Referred Image Grounding (ARIG). For this task, the dataset collection consists of image-audio pairs from Openimages-AudioSet, Openimages-VGGSound, AVSBench, VGGSS, PASCAL Sound, and Flickr-Soundnet. Among these, for Openimages-AudioSet, Openimages-VGGSound and VGGSS we first obtain the top 3 image frames with the highest image-text CLIP similarity scores [48] and subsequently select the most suitable frame by manual inspection to form the image-audio pair. The frames are extracted from the video

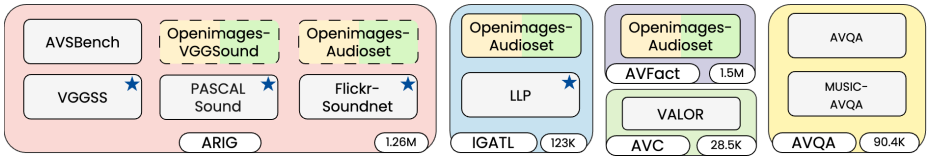


Fig. 1: Task-wise dataset distribution. The bi-coloured cells denote collections of paired image-audio samples from public datasets following our data curation strategy while single-coloured cells signify direct adaptation. Datasets with dashed outlines are used only during model training while the ones with ★ are reserved for zero-shot evaluations. Other datasets have a defined train/test split. Numbers in the bottom right represent the total #samples present in each task.

segment of interest (denoted in dataset annotation). Please refer to Tab. 1 for the Openimages-AudioSet / VGGSound classwise associations. We refer to this look-up table while matching the corresponding classes.

- **Openimages-AudioSet:** For every sample, we obtain the [`start`,`end`] time interval of the audio event of interest from the AudioSet dataset. Each sample is associated with an audio category. We use this class label while calculating the CLIP score with the image frames. We zero-pad and make length of each audio piece 30s.
- **Openimages-VGGSound:** We obtain the onset (`start`) of an event from the VGGSound dataset annotation and extract $min(\texttt{start} + 30, \text{len(audio)})$ second snippet. If the len(audio) is less than 30s we zero pad to maintain the audio sequence length.
- **AVSBench:** AVSBench comes with 5 to 6 frames along with the audio snippet. We choose the best frame that most closely relates to the audio event under consideration through manual inspection.
- **VGGSS:** We follow a similar strategy as that of VGGSound.
- **PASCAL Sound:** We choose 566 image samples from the PASCAL dataset [18] ranging from 12 sounding classes and carefully pair them with AudioSet samples using the same protocol as Openimages-AudioSet.
- **Flickr-Soundnet [53]:** Here we directly obtain the image audio pairs as released by the authors.

For all these cases we augment the image-audio pairs with our instruction tuning templates (refer to Section B).

Image Guided Audio Temporal Localization (IGATL).

- **Openimages-AudioSet (Strong):** While curating the image samples we follow a similar strategy as before. To ensure a fair assessment we choose audio snippets that are considerably longer than the event of interest (EoI). However, through manual inspections, we ensure that the EoI lies within the extracted audio piece.
- **LLP:** The LLP dataset provides fine-grained temporal annotations of the audio events in the format [`onset`,`offset`]. One representative image is chosen

Openimages Label Name	Audioset Label Name	VGGSound Label Name
Aircraft	Aircraft	airplane
Alarm clock	Alarm clock	alarm clock ringing
Ambulance	Ambulance (siren)	ambulance siren
Bicycle	Bicycle, tricycle	–
Bird	Bird	bird chirping, tweeting
Blender	Blender, food processor	electric blender running
Boat	Boat, Water vehicle	sailing
Bus	Bus	helicopter
Camera	Camera	–
Cannon	–	firing cannon
Car	Car	race car, auto racing
Cat	Cat	cat meowing
Cattle	–	cattle mooing
Ceiling fan	Mechanical fan	running electric fan
Cello	–	playing cello
Chainsaw	Chainsaw	chainsawing trees
Cheetah	Roaring cats (lions, tigers)	cheetah chirrup
Chicken	Fowl	–
Chime	Chime	wind chime
Clock	Clock	–
Computer keyboard	Computer keyboard	typing on computer keyboard
Computer mouse	Mouse	–
Corded phone	Dial tone	cell phone buzzing
Cutting board	Chopping (food)	chopping food
Dagger	Knife	–
Digital clock	–	alarm clock ringing
Dog	Dog	dog baying
Door	Door	–
Door handle	Doorbell	door slamming
Drill (Tool)	Drill	–
Drum	–	playing drum kit
Duck	Quack	duck quacking
Eagle	–	eagle screaming
Elephant	–	elephant trumpeting
Fireplace	Fire	–
Fixed-wing aircraft	Fixed-wing aircraft, airplane	–
Fountain	Waterfall	–
Fox	Canidae, wild dogs, wolves	fox barking
French horn	–	playing french horn
Frog	Frog	frog croaking
Girl	Female speech, woman speaking	–
Glasses	Glass	–
Goat	Goat	goat bleating
Golf cart	Cart	–
Goose	Ducks, geese, waterfowl	goose honking
Grinder	–	electric grinder grinding
Guitar	–	playing acoustic guitar
Hair dryer	Hair dryer	hair dryer drying
Hammer	Hammer	–
Hand dryer	Hair dryer	–
Handgun	Gunshot, gunfire	machine gun shooting
Harmonica	–	playing harmonica
Harp	–	playing harp
Harpsichord	–	playing harpsichord
Helicopter	Helicopter	helicopter
Horse	Horse	horse neighing
Human face	Female speech, woman speaking|Male speech, man speaking	–
Infant bed	–	baby crying
Ipod	Music	–
Jaguar (Animal)	Roaring cats (lions, tigers)	cheetah chirrup
Jet ski	Jet engine	skiing
Kettle	Steam whistle	–
Kitchen knife	Knife	–
Kitchen utensil	Kitchen and dining room sounds	–
Knife	Knife	–
Land vehicle	Vehicle	car passing by
Laptop	Typing	typing on computer keyboard
Leopard	Roar	–
Light switch	Clicking	–
Limousine	Car	–
Lion	Roar	lions roaring

Magpie	Bird	magpie calling
Mammal	Animal	–
Man	Male speech, man speaking	–
Mechanical fan	Mechanical fan	running electric fan
Microphone	Microphone	–
Microwave oven	Microwave oven	–
Missile	–	missile launch
Mixer	Blender, food processor	–
Mobile phone	Telephone	cell phone buzzing
Motorcycle	Motorcycle	driving motorcycle
Mouse	Mouse	–
Musical instrument	Music	orchestra
Musical keyboard	–	playing piano
Oboe	–	playing oboe
Otter	–	otter growling
Owl	Owl	owl hooting
Paper cutter	Scissors	ripping paper
Parrot	Bird	parrot talking
Person	Female speech, woman speaking|Male speech, man speaking	–
Piano	–	playing piano
Pig	Pig	pig oinking
Popcorn	Burst, pop	popping popcorn
Power plugs and sockets	Power tool	–
Pressure cooker	Steam	–
Printer	Printer	printer printing
Rabbit	Rodents, rats, mice	–
Ratchet (Device)	Ratchet, pawl	–
Raven	Crow	crow cawing
Reptile	Snake	–
Rifle	Machine gun	machine gun shooting
Rocket	–	missile launch
Saxophone	–	playing saxophone
Sea lion	–	sea lion barking
Segway	Non-motorized land vehicle	–
Sewing machine	Sewing machine	using sewing machines
Sheep	Sheep	–
Shotgun	Gunshot, gunfire	–
Shower	Shower	–
Skateboard	Skateboard	skateboarding
Ski	–	skiing
Snail	–	hail
Snake	Snake	snake rattling
Snowboard	Skateboard	skiing
Snowmobile	Motorcycle	–
Snowplow	Lawnmower	–
Spoon	Kitchen and dining room sounds	–
Stationary bicycle	Bicycle, tricycle	driving motorcycle
Swan	Quack	–
Swimming pool	Water	–
Sword	Knife	–
Table tennis racket	–	playing table tennis
Tablet computer	Computer keyboard	typing on computer keyboard
Tap	Tap	–
Taxi	Car	hail
Telephone	Telephone	telephone bell ringing
Television	Television	–
Tiger	Roar	–
Toilet	Toilet flush	toilet flushing
Train	Train	–
Truck	Truck	–
Trombone	–	playing trombone
Trumpet	–	playing trumpet
Turkey	Turkey	–
Unicycle	Bicycle bell	–
Van	Car	–
Vehicle	Vehicle	vehicle horn, car horn, honking
Violin	–	playing violin, fiddle
Wall clock	Clock	alarm clock ringing
Washing machine	Washing machine	–
Watch	Clock	–
Wine glass	Glass	–
Whale	–	whale calling
Woman	Female speech, woman speaking|Male speech, man speaking	–
Woodpecker	Wood	woodpecker pecking tree

Table 1: Image audio class mapping. We associate the image and audio classes from the Openimages and the AudioSet / VGGSound datasets and prepare a lookup table through careful manual inspection.

Task	Example Instruction
ARIG	Given the audio and image pair, identify the object category of the audio. Now, provide a bounding box for that object in the image. The answer should be in the form [<obj>,xLeft,yTop,xRight,yBottom]. <obj> represents the object category. xLeft,yTop are coordinates of the top-left corner and xRight,yBottom are coordinates of the bottom-right corner of the bounding box. The coordinates should be within the range 0 to 1. From the given audio and image pair first identify the object category of the audio. Then localize the corresponding object in the image by providing a bounding box. The answer should be in the form [<obj>,xLeft,yTop,xRight,yBottom]. <obj> represents the object category. xLeft,yTop are coordinates of the top-left corner and xRight,yBottom are coordinates of the bottom-right corner of the bounding box. The coordinates should be within the range 0 to 1. Given the audio and image pair, identify the object category of the audio. Now, localize the object in the image by providing a bounding box. The answer should be in the form [<obj>,xLeft,yTop,xRight,yBottom]. <obj> represents the object category. xLeft,yTop are coordinates of the top-left corner and xRight,yBottom are coordinates of the bottom-right corner of the bounding box. The coordinates should be within the range 0 to 1. Considering the audio and image pair, determine the object class of the audio. Next, localize the same object in the image by providing a bounding box. The answer should be in the form [<obj>,xLeft,yTop,xRight,yBottom]. <obj> represents the class of the object. xLeft,yTop are coordinates of the top-left corner and xRight,yBottom are coordinates of the bottom-right corner of the bounding box. The coordinates should be within the range 0 to 1. Considering the audio and image pair, recognize the object category of the audio. Subsequently, draw a bounding box around that object shown in the image. The answer should be in the form [<obj>,xLeft,yTop,xRight,yBottom]. <obj> represents the category of the object. xLeft,yTop are coordinates of the top-left corner and xRight,yBottom are coordinates of the bottom-right corner of the bounding box. The coordinates should be within the range 0 to 1. Considering the audio and image pair, recognize the object category of the audio. Next, draw a bounding box around that object in the image. The answer should be in the form [<obj>,xLeft,yTop,xRight,yBottom]. <obj> represents the category of the object. xLeft,yTop are coordinates of the top-left corner and xRight,yBottom are coordinates of the bottom-right corner of the bounding box. Ensure the bounding box is within the range 0 to 1.
IGATL	Identify the object category from the image. Now, find the time duration in the audio where that object is making the sound. The output should be in the form (tStart,tEnd) where tStart and tEnd are the start and end times respectively. tStart is less than tEnd. The minimum value of tStart is 0. The maximum value of tEnd is 30. Given the image, identify the object category. Next, output the time window in the audio where that object is making the sound. The output should be in the form (tStart,tEnd) where tStart and tEnd are the start and end times respectively. tStart is less than tEnd. The minimum value of tStart is 0. The maximum value of tEnd is 30. Which object do you see in the image? Please find the time window in the audio where that object is making the sound. The output should be in the form (tStart,tEnd) where tStart and tEnd are the start and end times respectively. tStart is less than tEnd. The minimum value of tStart is 0. The maximum value of tEnd is 30. Recognise the object category from the image. Now, indicate the time duration in the audio where that object is making the sound. The output should be in the form (tStart,tEnd) where tStart and tEnd are the start and end times respectively. tStart is less than tEnd. The minimum value of tStart is 0. The maximum value of tEnd is 30. What is the category of the object that you see in the image? Now, indicate the temporal duration in the audio where that object is making the sound. The output should be in the form (tStart,tEnd) where tStart and tEnd are the start and end times respectively. tStart is less than tEnd. The minimum value of tStart is 0. The maximum value of tEnd is 30.

AVFact	Does the object inside the bounding box <placeholder_bbox> of the image produce the same sound as in the given audio? Answer in True or False. Given the image, does the object inside the bounding box <placeholder_bbox> produce the same sound as in the given audio? Answer in True or False. The object inside the bounding box <placeholder_bbox> of the image produces the same sound as in the given audio. True or False? From the audio-image pair, verify if the object inside the bounding box <placeholder_bbox> produces the same sound as present in the given audio. Answer in True or False. The object in the given audio between time duration <placeholder_time> is present in the image. True or False? Listen to the audio in the time window <placeholder_time>. Does this object exist in the image? Answer in True or False. Listen to the audio in the time window <placeholder_time>. Verify if the same object is present in the image. True or False? The time segment <placeholder_time> contains the object as present in the image. True or False? Listen to the audio in the time window <placeholder_time>. The same object is within the bounding box <placeholder_bbox> in the image. True or False? Does the object inside the bounding box <placeholder_bbox> of the image produce the same sound as within the time duration <placeholder_time> in the given audio? Answer in True or False. The object inside the bounding box <placeholder_bbox> of the image produces the same sound as in the time segment <placeholder_time> of the audio. True or False? The time segment <placeholder_time> contains the object in the bounding box <placeholder_bbox> of the image. True or False? Here is an audio-image pair. Does the given audio correspond to the object shown in the image? Answer in True or False. Does the given audio correspond to the object shown in the image? Answer in True or False. Does the given audio associate with the object shown in the image? Answer in True or False. Here is an audio-image pair. Does the given image associate with the object sounding in the audio? Answer in True or False.
AVQA	How many instruments are sounding in the image? Which is the musical instrument that sounds at the same time as the <Object>? Is the <Object> on the <LR> louder than the <Object> on the <LR>? Is there a voiceover? Is the <Object> playing longer than the <Object>?
AVC	Considering the audio input, generate a caption for the image.

Table 2: Task wise instructions template.

from within this time segment. While preparing our test set, we restrict ourselves to one category per video and their corresponding `onset` and `offset` values to rule out overlapping events within the same time interval.

Audio-Visual Fact-checking (AVFact).

– **Openimages-AudioSet:** For *Type 1* we collect samples from the AudioSet split while for *Type 2* and *Type 3* we choose samples from AudioSet Strong split as it consists of time-sensitive grounding information which is used in these three types of queries. For the image collection, we follow the same strategy as before.

A.3 Coarse-grained Data Preparation

For the coarse-grained tasks, we resort to direct adaptations of publicly available datasets.

Audio-Visual Question Answering. In the absence of audio class labels, we manually inspect the video to obtain the most suitable frame for each sample.

– **AVQA:** The AVQA dataset contains the `start` time stamp which denotes the onset of the event of interest. We follow the same train/test split as proposed by the authors [62].
– **MUSIC-AVQA:** We crop the 30s from the original 1-minute-long video sequence within which the event of interest lies.

Audio-Visual Captioning (AVC).

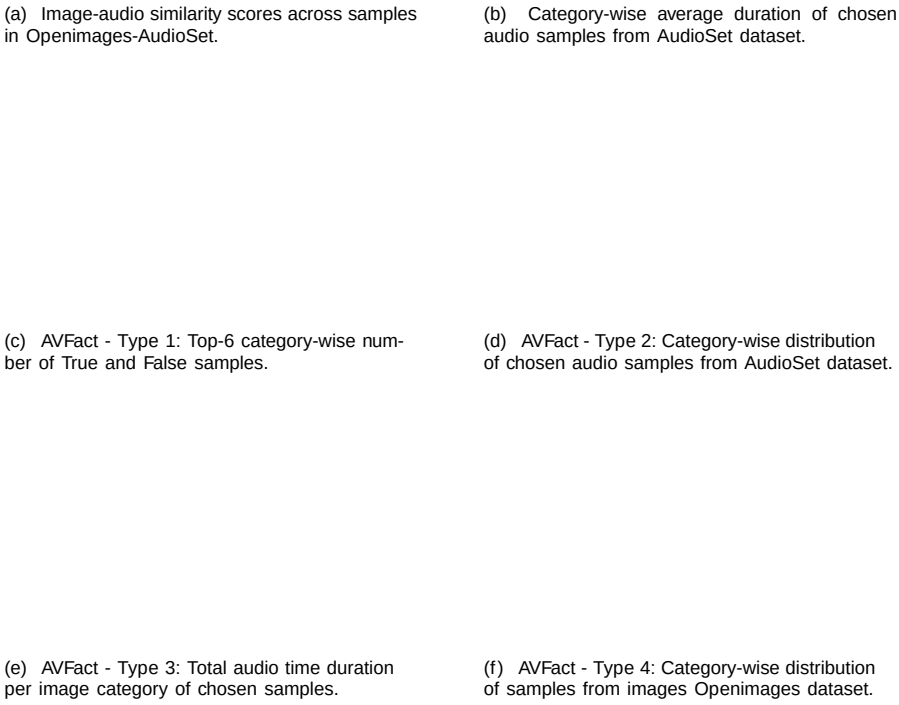
– **VALOR-32K:** Each sample in the VALOR dataset comprises an elaborate caption of the audio-visual scene. We leverage this caption to calculate the CLIP similarity score between the image-text pair and obtain the top 3 most relevant frames from within the 10s long annotation as provided by the authors. Finally, we choose one representative frame through manual inspection.

B Dataset Instruction Templates

We add task-wise sample instruction templates in Tab. 2. To make the instruction tuning robust and incorporate sufficient diversity, we manually write a few instructions and prompt GPT-3.5 [5] to generate different variants. We further refine the instruction templates using GPT-4 [1]. Note that for AVQA and AV captioning tasks, we restrict ourselves to the questions and captions provided by the authors.

C Dataset Statistics and Analysis

Image Audio Similarity. To study the similarity between the image-audio pairs [13] from Openimages-AudioSet, we utilize the CLIP [48] and CLAP [17] scores by calculating $S_{\mathsf{CLIP}}\ S^{T}_{\mathsf{CLAP}}$, where $S\ 2\ \mathbb{R}^{N\ \ N}$ and denotes the pairwise cross-modal similarity scores for a batch of size N. The CLIP similarity is calculated between the chosen image and the audio class label, similarly, the CLAP score is calculated between the audio class label and the audio snippet. The text modality acts as the bridging modality in this case. Fig. 2a reports the image-audio similarity scores over the most frequent 9 categories while 'others' denotes aggregation of all the remaining ones. Note the range of the scores is normalized between [0,1] with 0 being the lowest. The average score of image-audio pairs across all samples collected for the audio referred image grounding task is 0.77, supporting a strong association between the two modalities.
Audio Duration. We report the category-wise mean duration (in sec.) of the audio samples from the AudioSet dataset in Fig. 2b for image-guided audio temporal localization task. The 'Train' class has the overall highest value with an average duration of 9.83 sec while the 'Clicking' category has the lowest average duration at 0.32 sec.
Class wise Robustness. We report class-wise (top 6 classes based on occurrence) *True/False* sample count from the Openimages dataset for AVFact - Type



(a) Image-audio similarity scores across samples in Openimages-AudioSet.

(b) Category-wise average duration of chosen audio samples from AudioSet dataset.

(c) AVFact - Type 1: Top-6 category-wise number of True and False samples.

(d) AVFact - Type 2: Category-wise distribution of chosen audio samples from AudioSet dataset.

(e) AVFact - Type 3: Total audio time duration per image category of chosen samples.

(f) AVFact - Type 4: Category-wise distribution of samples from images Openimages dataset.

Fig. 2: AVFIT statistics and analysis.

1 set in Fig. 2c. We maintain a good balance of matched and mismatched pairs to ensure our model is robust to deceptive queries.

AudioSet Distribution. Fig. 2d reports the class-wise distribution of samples present in the AVFact Type-2 set as collected from the AudioSet dataset.

Audio Duration Per Image Class. In Fig. 2e we present the duration of audio samples across various image classes from Openimages in AVFact Type-4 split. This demonstrates the overall balanced mix of image-audio distributions across di erent pairings.

Category Wise Distribution. Fig. 2f presents image category-wise distributions of samples from the Openimages dataset for AVFact Type-4 samples.