

Temporal Reasoning via Audio Question Answering

Haytham M. Fayeck and Justin Johnson

Abstract—Multimodal question answering tasks can be used as proxy tasks to study systems that can perceive and reason about the world. Answering questions about different types of input modalities stresses different aspects of reasoning such as visual reasoning, reading comprehension, story understanding, or navigation. In this paper, we use the task of Audio Question Answering (AQA) to study the temporal reasoning abilities of machine learning models. To this end, we introduce the Diagnostic Audio Question Answering (DAQA) dataset comprising audio sequences of natural sound events and programmatically generated questions and answers that probe various aspects of temporal reasoning. We adapt several recent state-of-the-art methods for visual question answering to the AQA task, and use DAQA to demonstrate that they perform poorly on questions that require in-depth temporal reasoning. Finally, we propose a new model, Multiple Auxiliary Controllers for Linear Modulation (MALiMo) that extends the recent Feature-wise Linear Modulation (FiLM) model and significantly improves its temporal reasoning capabilities. We envisage DAQA to foster research on AQA and temporal reasoning and MALiMo a step towards models for AQA.

Index Terms—Audio, Question Answering, Reasoning, Temporal Reasoning

I. INTRODUCTION

A central goal of artificial intelligence research has been to build systems that can perceive and reason about the world. In recent years, question answering has been used as a proxy task for this goal, where systems must answer natural language questions about another input such as an image [1]–[3], a piece of text [4], [5], an interactive environment [6], or a movie [7]. Answering questions about different types of inputs stresses different aspects of reasoning such as visual reasoning, reading comprehension, story understanding, or navigation.

The world is dynamic, with events unfolding after each other over time. Systems that interact with the world must therefore be adept at *temporal reasoning*. However the input modalities used by prior question answering benchmarks are ill-suited for isolating temporal aspects of reasoning. Images offer only a static snapshot of the world. Text may describe temporal sequences, but lacks intrinsic temporality. Navigating environments requires sequences of actions, but recent benchmarks navigate only static worlds. Movies are fundamentally temporal, but processing hours-long videos introduces computational barriers to studying temporal reasoning.

We believe that *audio* is an appealing modality for investigating temporal reasoning, as it is fundamentally temporal and requires no spatial reasoning. In this paper, we therefore investigate the task of *Audio Question Answering* (AQA) where

H. M. Fayeck is with Facebook Reality Labs, Redmond, WA, USA.
E-mail: haythamfayeck@fb.com.

J. Johnson is with Facebook AI Research, Menlo Park, CA, USA.
E-mail: jcjohns@fb.com.
Manuscript under review.

systems must answer natural language questions pertaining to audio clips. In addition to being a testbed for temporal reasoning, advances in AQA have the potential to assist hearing-impaired individuals, much as advances in visual question answering could be used to assist visually-impaired individuals [8]–[10].

Towards these goals, this paper introduces the Diagnostic Audio Question Answering (DAQA) dataset for studying temporal reasoning in a controlled manner. DAQA provides 100,000 audio sequences ranging in length from approximately 10 seconds to approximately 3 minutes, constructed from a library of 400 natural sound events, and approximately 600,000 programmatically generated questions and answers that probe various aspects of temporal reasoning; see Figure 1 for examples. Inspired by synthetic reasoning benchmarks for other domains [2], [5], [11], [12], the design of DAQA minimizes bias that can be exploited by machine learning models to correctly answer questions without examining the question or the audio, and allows for fine-grained targeting of specific temporal reasoning skills.

We use DAQA to investigate the temporal reasoning abilities of several baseline models as well as adaptations of several recent state-of-the-art methods for visual question answering [13], [14]. We demonstrate that although these methods show strong performance for visual question answering, they perform comparatively poorly on questions requiring in-depth temporal reasoning.

As a step towards making methods more adept at temporal reasoning, we introduce **Multiple Auxiliary Controllers for Linear Modulation** (MALiMo), which extends and enhances the Feature-wise Linear Modulation (FiLM) model [14]. FiLM uses an auxiliary controller to process a supplementary input, e.g. the question, and predict parameters to modulate processing in a convolutional network that ingests the principal input, e.g. the image or the audio, to predict an answer. MALiMo augments FiLM with an additional auxiliary controller that receives subsampled features of the principal input and predicts parameters that modulate subsequent stages of processing in the convolutional network. This allows processing in the convolutional network to be altered as a function of both the principal and supplementary inputs, which facilitates relational and temporal reasoning. Experiments on DAQA show that MALiMo significantly outperforms other competing approaches, and that it leads to the largest improvements on questions that require in-depth temporal reasoning.

Code to generate the DAQA dataset and our models for the AQA task are publicly available¹. We hope that this will foster research both on AQA and broadly on temporal reasoning.

¹The code to generate the DAQA dataset and our models for the AQA task are available at <https://github.com/facebookresearch/daqa>.

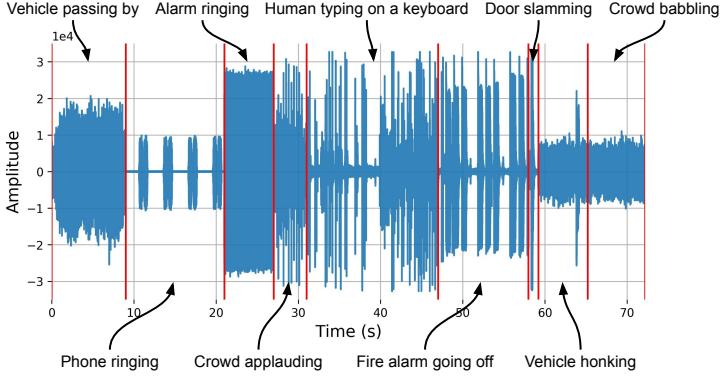


Fig. 1: DAQA provides audio clips composed of natural sound events (left) and programmatically generated questions and answer about those audio clips that test various aspects of temporal reasoning (right). Note that the annotations on the audio clip are for illustration purposes only.

Outline. The outline of the paper is as follows. Section II briefly reviews related work. Section III delineates the design and analysis of the DAQA dataset. Section IV outlines oracle benchmarks and baseline models devised to divulge bias and assess the difficulty of the DAQA dataset. Section V presents our novel MALiMo model. Section VI discusses the DAQA dataset and the MALiMo model as well as avenues for future work. Section VII concludes the paper.

II. RELATED WORK

Question Answering Datasets. The AQA task and the DAQA dataset draw inspiration from datasets of natural language questions and answers about different input modalities, such as SQuAD [4], [15] for text; VQA [1], [16], VizWiz [9], Visual Genome [17], DAQUAR [18], and Visual7W [3] for images; TVQA [19], MovieQA [7], Video-QA [20], and [21] for videos; and EQA [6] for embodied environments.

Diagnostic Reasoning Datasets. DAQA is synthetic and programmatically generated, allowing us to control bias in the dataset and assess fine-grained temporal reasoning skills of AQA models. We draw inspiration from synthetic datasets for diagnosing other reasoning skills, such as bAbI [5] for reading comprehension, CLEVR [2], COG [12], and GQA [22] for visual reasoning, and PGM [11] for abstract reasoning.

Audio Question Answering. There has been some work on answering questions about images using spoken rather than written questions [23], and answers to some questions about videos might require jointly reasoning about visual and audio cues [7], [19]–[21]. Most related to our work is the concurrent CLEAR dataset [24], which similar to DAQA, is a synthetic dataset of audio sequences and questions. All audio sequences in CLEAR are of fixed length and consist of 10 musical notes, while DAQA provides variable-length audio sequences with a variable number of more general audio events. CLEAR also adapts the question templates directly from the synthetic visual question answering dataset, CLEVR [2], while DAQA uses question types custom-built to emphasize temporal reasoning. We believe that these differences make DAQA a significantly more challenging dataset; [24] achieves 89.97% accuracy on CLEAR with a FiLM [14] baseline, while our

best and significantly improved FiLM model on DAQA only achieves 78.33% accuracy.

Question Answering Models. There is a plethora of work on answering questions of various modalities, and a full survey is beyond the scope of this paper. Our work draws most directly from recent methods for visual question answering [13], [25]–[28] and visual reasoning [14], [29]–[31]. Our MALiMo model is most closely related to FiLM [14] and conditional batch normalization [32], though its auxiliary audio controller is related to transformers [33], non-local networks [34], and SENets [35].

III. THE DIAGNOSTIC AUDIO QUESTION ANSWERING (DAQA) DATASET

The DAQA dataset is a synthetically generated dataset of `<audio, question, answer>` triplets. The dataset requires temporal reasoning capabilities to identify, locate, compare, and count relevant audio events in a sequence of audio events, in order to correctly provide an answer to the posed question. The DAQA dataset is envisaged to be useful for understanding, assessing, and benchmarking models for temporal reasoning in an AQA framework.

Each question in DAQA pertains to an *audio clip* composed of several atomic *audio events* arranged in sequence. We draw from a library of 400 audio events organized into 20 *event types*. Each question is instantiated from one of 54 *question templates* that test different aspects of temporal reasoning. Instantiating questions from templates involves several heuristics to reduce dataset bias. The answer to each question is one of 36 distinct values, so answering DAQA questions can be viewed as a classification problem. Overall, DAQA provides 100,000 audio clips and 599,294 questions and answers divided into standard training, validation, and test splits with 80,000; 10,000; and 10,000 audio clips and 399,924; 99,702; and 99,668 questions and answers each.

Audio Events. Audio events are the atoms from which our audio clips are composed. We use 20 different *event types*² chosen to span a wide variety of audio sources, locations, and sound types. Each event type consists of a *source* and an *action* (e.g. *crowd applauding*, *dog barking*). DAQA provides

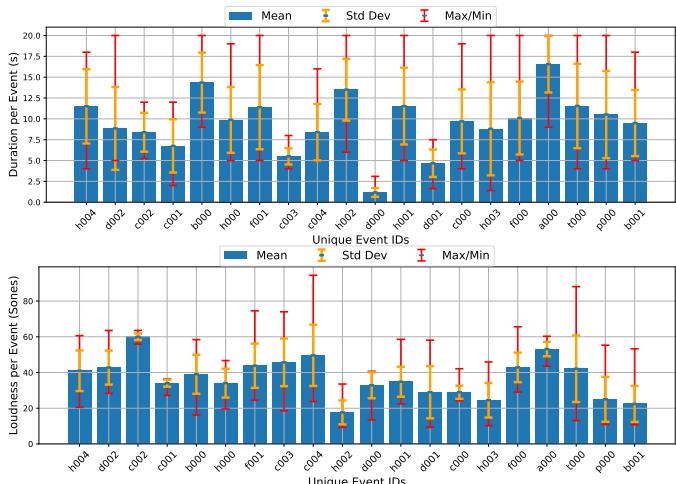


Fig. 2: Mean, standard deviation, and limits of the duration (seconds) of the 20 instances for each of the 20 audio events (top). Mean, standard deviation, and limits of the loudness (sones) of the 20 instances for each of the 20 audio events (bottom).

20 unique instances of each event type, for a total of 400 unique audio event instances. 91 of these events were recorded by the authors, while the other 309 were manually curated from the balanced training subset of AudioSet [36]. As shown in Figure 2, our 400 audio events vary significantly in duration and loudness³, both within each event type and across different event types.

We divide our 20 event types into 5 *discrete* types (*aircraft flying over, car passing by, door slamming, human speaking, and human laughing*) and 15 *continuous* types (all others). A sequence of discrete events of the same type (e.g. two consecutive *door slamming* events) can be easily segmented by a listener into disjoint events, while sequential continuous events of the same type (e.g. two consecutive *crowd babbling* events) would likely be heard as a single continuous event.

Audio Clips. Audio clips are generated by concatenating between 5 and 12 random audio events. We only allow consecutive *discrete* events of the same type; clips with adjacent *continuous* events of the same type are rejected. We randomly overlap successive events by up to 500 ms, and add normally distributed background noise to half of our audio clips. We generated 100,000 audio clips, divided into training, validation, and test splits of 80,000; 10,000; and 10,000 clips respectively. We ensured that no validation or test clips have the exact same sequence of events instances as any training clip.

Since each audio clip contains a variable number of audio events and each audio event varies in length, our generated

²Events are referred to by a unique ID as follows; a000: *aircraft flying over*, b000: *band playing*, b001: *bird singing*, c000: *crowd babbling*, c001: *crowd applauding*, c002: *crowd rioting*, c003: *car honking*, c004: *car passing by*, d000: *door slamming*, d001: *doorbell ringing*, d002: *dog barking*, f000: *fire engine passing by*, f001: *fire alarm going off*, h000: *human speaking*, h001: *human laughing*, h002: *human typing on a keyboard*, h003: *human whistling*, h004: *human operating a machine*, p000: *phone ringing*, and t000: *storm thundering*.

³We compute loudness following the time-varying loudness model [37] using the Genesis Loudness Toolbox.

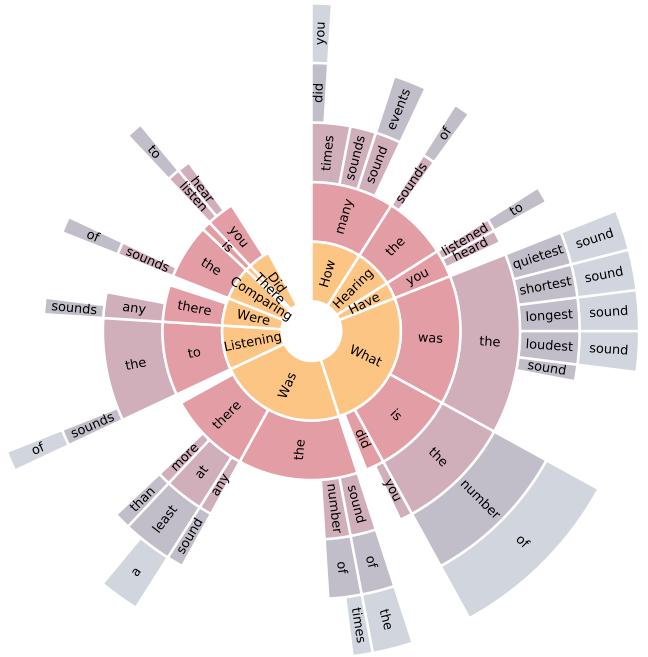


Fig. 3: Distribution of the first five words for all questions in the training set of the DAQA dataset. The innermost ring represents the first words and radiating rings represent subsequent words. Arc lengths are proportional to the number of questions with the word. Words accounting for less than 1% were omitted for clarity.

audio clips vary significantly in length: clips in the training set vary from 10.5 to 178.2 seconds, with a mean and standard deviation of 80.8 ± 26.3 seconds. We believe that this wide variety in clip lengths is necessary for evaluating temporal reasoning.

Each audio clip is annotated with the order, identity, duration, and loudness of its constituent events. We use these annotations only for analysis and oracle baseline models; in particular our AQA models are trained end-to-end for question answering and are not supervised with this information.

Question Templates. Questions are generated programmatically from one of 54 manually designed *question templates*, Each template contains several *placeholder* values of various types; given annotations for an audio clip, the template can be *instantiated* by choosing values for each placeholder, giving rise to a (question, answer) pair for the clip. For example, the template What did you hear <RO> the <S> <A>? has placeholders <RO> for a preposition (*before* or *after*), <S> and <A> for the source and action of an event type. To increase linguistic diversity, each template provides several logically equivalent English phrasings, such as What was the sound <RO> the <S> <A>? for the template above; we also randomly replace some words with synonyms (e.g. *person* for *human*). Associated with each template are short Python programs to generate or verify answers given placeholder values and audio clip annotations.

To stress temporal reasoning, templates use a variety of mechanisms to refer to audio events. Audio events can be

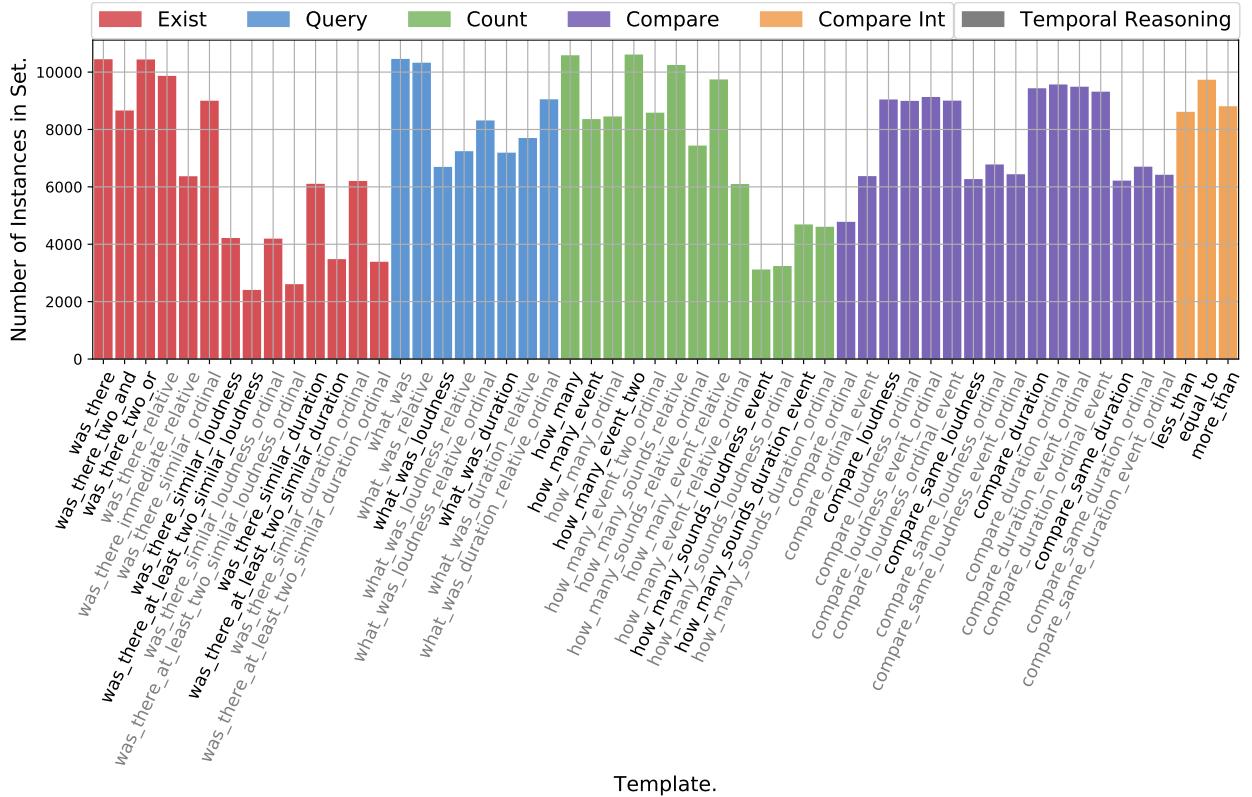


Fig. 4: Number of examples per template in the training set of the DAQA dataset sorted by color with respect to reasoning skills. Templates that require temporal reasoning, e.g. contain prepositions: before / after, or ordinals, are highlighted in gray.

referenced using event type (*the alarm ringing*), absolute ordinal position (*the third sound*), relative position (*the sound immediately after the alarm ringing*), absolute duration or loudness, (*the loudest sound*), or relative duration or loudness (*sounds longer than the alarm ringing*).

Templates may also be grouped by reasoning skills: *exist* templates ask whether an event is present (*was there an alarm ringing?*), *query* templates ask about event properties (*What was the loudest sound?*), *compare* templates require comparisons between two events (*was the first sound louder than the vehicle honking?*), *count* templates require counting events meeting some condition (*how many events were louder than the alarm ringing?*), and *compare integer* templates require comparing sizes of two event sets (*were there more alarms ringing than sounds before than the door slamming?*).

Template Instantiation. Instantiating a template requires choosing a value for each of its placeholders. This process involves several subtle issues that require special attention.

Some choices of placeholders give questions that are *invalid* for some audio clips – for example the question *What was the sound before the human speaking?* is valid only for clips with exactly one *human speaking* event. For some (template, clip) pairs, most possible template instantiations will be invalid, so randomly sampling placeholder values and rejecting invalid questions would be inefficient. We overcome this by incorporating logic into each template to sample only from valid placeholder values, and we terminate early if a placeholder has no valid values.

Sampling questions uniformly at random from all valid template instantiations can lead to severe answer imbalance. For example, consider questions of the form *Was there a <S> <A> immediately after the door slamming?* – as long as the clip contains a door slamming, any values for <S> and <A> give valid questions; however answers will be heavily biased towards *No*, introducing a significant question-conditional bias and allowing models to achieve high accuracy for these questions without examining the audio clip. We overcome this problem heuristically by rejecting questions that lead to a large difference (> 5%) between the most and least common answers for each template. Figure 5 shows the frequency of each answer in the training set; the relatively uniform distribution within each answer type is a result of this heuristic.

We attempt to instantiate 5 questions for each training clip and 10 questions for each validation and test clip; however due to the above heuristics, we are unable to sample the desired number of questions for all clips. Our training set thus contains 80,000 clips and 399,924 questions (172,126 of which are unique); the validation set has 10,000 clips and 99,702 questions, and the test set has 10,000 clips and 99,668 questions. Questions in the training set range in length from 5 to 27 words, with a mean length and standard deviation of 12.89 ± 3.86 words.

Answers. Each question's answer is one of 36 possible values: *yes*, *no*, the 20 event types, *nothing*, and integers 0 to 12 (inclusive). Figure 5 shows the frequency of these answers on the training set.

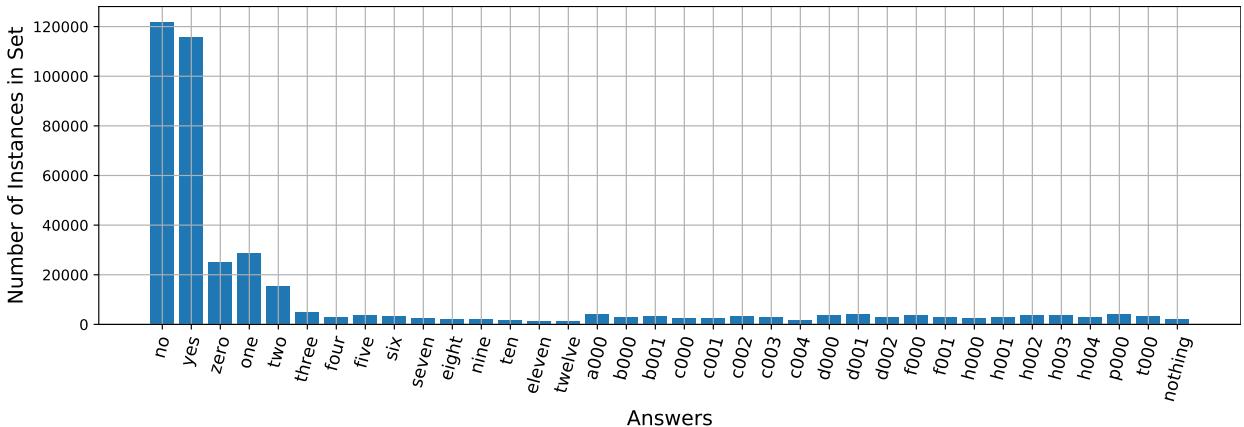


Fig. 5: Frequency of answers in the training set of the DAQA dataset.

IV. BENCHMARK EXPERIMENTS

We use several baselines as well as adaptations of recent state-of-the-art models for visual question answering to investigate and benchmark DAQA and the AQA task.

Experimental setup. The AQA task, *Audio+Question \rightarrow Answer*, is formulated as a multi-class classification problem, and all models are trained to maximize the likelihood of the correct answer $p(\text{Ans}|\text{A}, \text{Q})$. DAQA is split into training, validation, and test sets; see Section III. The validation set was used to design experiments and tune hyperparameters, such as the number and size of the layers, independently per model via grid search. Each model was evaluated on the test set only once after finalizing all experimental details and hyperparameters.

Each audio clip was split into 25 ms frames with a stride of 10 ms, and a Hamming window was applied, then 64 log-Mel-Frequency Spectral Coefficients (MFSCs) were extracted from each frame. The mean and standard deviation were normalized per coefficient to zero and one respectively using the mean and standard deviation computed from the training set only.

All models were trained from scratch, i.e. no pre-trained word embeddings or models were used. Neural models were trained via stochastic gradient descent with a mini-batch size of 40 using Adam [38] with learning rate $\alpha = 1 \times 10^{-4}$ and weight decay $\lambda = 1 \times 10^{-5}$ for 100 epochs, and the validation set was used to select the best model during training.

The following is a detailed description of the baselines and models. The results are listed in Table I.

Random / Mode Answer. We choose either a random or the most frequent (mode) answer from the training set. We also evaluate per-template (P/T) versions which output either a random valid answer for each question’s template, or the most frequent training set answer for each template.

Audio Only. This model ignores the question, and uses a Fully Convolutional Network (FCN) on the audio to predict the answer. The FCN model has five convolutional blocks similar to those in VGG [39]; each block comprises two convolution layers with Batch Normalization (BatchNorm) and Rectified Linear Unit (ReLU) nonlinearities, and a max pooling layer with a 2×2 window with stride 2 after the second convolution layer; followed by two standard convolution layers with BatchNorm and ReLUs, a final convolution layer, then

global average pooling across each channel. The convolution layers in the first convolutional block have 32 filters of size 3×12 with stride 1×9 , whereas the number of filters double every convolutional block up to 512 filters, all of which are of size 3×3 with stride 1, except the penultimate layer, which has 1024 filters of size 1×1 and the final layer which has $K = 36$ filters of size 1×1 , where K corresponds to the number of classes in DAQA.

We train models both for the entire dataset as well as separate models per-template (P/T). The overall accuracy of the audio only model for all templates is comparable to the Mode Answer baseline (31.27% vs 30.52%). The overall accuracy of the audio only per-template models is slightly better than the Mode Answer P/T baseline (39.72% vs 43.59%). These results indicate that DAQA has minimal audio-conditional bias.

Question Only. We use several baselines that ignore the audio. We train logistic regression models on a one-hot encoding of each question’s template (Question Template) and on Bag of Words encodings of questions (Q-only Logit). We also train models that encode the questions with 128-dimensional word embeddings followed by a two-layer unidirectional Long Short-Term Memory (LSTM) network with 512 units per layer and predict answers with a linear projection of the final LSTM hidden state (Q-Only LSTM).

We train the latter two models both for the entire dataset as well as separate models per-template (P/T). Per-template LSTMs are not substantially better than the Mode Answer P/T baseline (44.93% vs 39.72%), suggesting that DAQA has minimal question-conditional bias.

FCN-LSTM. This model is composed of an FCN, identical to the FCN used for the audio only model, to encode the audio and a two-layered LSTM, with 512 units in each layer, to encode 256-dimensional word embeddings of each word in the question. Both representations are concatenated and fed to a fully connected neural network with a single hidden layer of 1024 units and ReLUs to predict an answer.

This model performs relatively poorly (53.65%) but outperforms prior audio only and question only baselines.

ConvLSTM-LSTM. This model is composed of five convolutional blocks identical to the VGG blocks in the audio only model, that encode the audio into a variable-sized

TABLE I: Audio question answering performance on the DAQA test set. We use baselines to divulge bias in the dataset and adapt recently proposed models for visual question answering [13], [14] to the AQA task. Our proposed MALiMo model outperforms prior methods due to improved temporal reasoning capabilities.

Model	# Params	Exist	Query	Count	Compare	Compare Int	All (%)
Random Answer	–	2.55	2.96	2.73	2.80	2.88	2.76
Mode Answer	–	51.50	0.00	0.00	51.33	50.69	30.52
Random Answer P/T	–	49.56	4.75	15.66	50.10	49.59	34.15
Mode Answer P/T	–	52.02	7.52	32.54	51.41	52.05	39.72
Audio Only (FCN)	7.65M	55.64	0.03	0.0	50.66	51.54	31.27
Audio Only P/T (FCN)	–	55.90	17.52	37.62	50.73	57.24	43.59
Question Template	–	52.02	7.48	32.53	51.41	52.05	39.72
Q-Only (Logit)	–	53.02	12.88	33.98	51.46	51.95	41.19
Q-Only P/T (Logit)	–	53.75	12.38	34.56	51.95	52.37	41.59
Q-Only (LSTM)	3.45M	54.20	13.09	34.22	61.23	51.04	44.53
Q-Only P/T (LSTM)	–	53.87	13.48	34.78	61.72	52.93	44.93
FCN-LSTM	9.79M	67.88	30.07	46.42	59.99	62.99	53.65
ConvLSTM-LSTM	14.78M	71.52	45.72	52.09	62.13	65.11	59.22
FCN-LSTM-SA	10.58M	75.52	45.68	57.1	61.67	70.5	61.48
FiLM-512 (2 FiLM layers)	5.49M	77.27	56.23	57.77	62.17	75.22	64.27
FiLM-512 (4 FiLM layers)	6.35M	77.63	56.69	59.11	62.23	76.53	64.84
FiLM-512 (6 FiLM layers)	7.21M	79.6	60.54	62.61	63.56	81.42	67.49
FiLM-512 (8 FiLM layers)	8.07M	80.08	72.91	77.22	65.93	89.23	74.39
FiLM-512 (10 FiLM layers)	8.93M	81.03	76.37	75.28	66.63	94.59	75.3
FiLM-512 (12 FiLM layers)	9.79M	82.92	80.13	79.74	69.83	93.68	78.33
FiLM-1024 (2 FiLM layers)	15.73M	78.33	56.57	59.06	62.61	76.7	65.09
FiLM-1024 (4 FiLM layers)	16.85M	79.29	61.04	63.54	62.92	84.87	67.76
FiLM-1024 (6 FiLM layers)	17.97M	79.76	61.22	64.77	63.76	88.17	68.66
FiLM-1024 (8 FiLM layers)	19.09M	79.69	75.86	70.15	63.93	93.53	72.79
FiLM-1024 (10 FiLM layers)	20.21M	79.98	61.33	64.35	63.4	85.48	68.34
FiLM-1024 (12 FiLM layers)	21.33M	78.77	76.46	75.28	64.96	88.13	73.87
MALiMo (1 Block; ours)	8.91M	78.34	46.73	64.18	62.53	83.45	65.08
MALiMo (2 Blocks; ours)	9.77M	80.86	78.21	71.52	67.34	90.75	74.65
MALiMo (3 Blocks; ours)	10.63M	82.62	81.09	77.22	68.95	91.16	77.39
MALiMo (4 Blocks; ours)	11.49M	85.69	86.33	80.49	74.93	95.64	81.87
MALiMo (5 Blocks; ours)	12.34M	89.10	88.80	85.05	79.63	95.84	85.59
MALiMo (6 Blocks; ours)	13.20M	91.08	90.60	86.35	86.53	97.04	88.86

representation, which is fed to a single-layered LSTM, with 512 units, that encodes the variable-sized representation into a fixed-sized representation of the audio; another two-layered LSTM, with 512 units in each layer, is used to encode 256-dimensional word embeddings of each word in the question into a fixed-sized representation of the question. Both representations are concatenated and fed to a fully connected neural network with a single hidden layer of 1024 units and ReLUs to predict an answer.

This model outperforms all baselines and the FCN-LSTM model (59.22% vs 53.65%).

FCN-LSTM-Stacked Attention. This model is similar to the FCN-LSTM model described above, albeit with two layers of Stacked Attention (SA) [13]. Concretely, instead of concatenating the audio representations from the FCN and the encoded question representations from the LSTM, the global average pooling layer in the FCN is replaced with an adaptive average pooling layer with a 2×8 output, and two SA layers are employed that ingest the representations from the FCN and LSTM to produce attention maps (see [13] for details), which are then added to the FCN representations, and fed to a fully connected neural network with a single hidden layer of 1024

units and ReLUs to predict an answer.

This model, augmented with SA, slightly outperforms the ConvLSTM-LSTM model (61.48% vs 59.22%) using less parameters (14.78M vs 10.58M).

FiLM. FiLM [14] adopts feature-wise linearly modulated layers conditioned on the question that manipulate the intermediate representations of a neural network to adaptively influence the output of the neural network by applying an affine transformation as follows. Let $\mathbf{H}^{(l)} \in \mathbb{R}^{n \times h \times w}$ be the output of a linear convolution layer l that has n filters, the modulation parameters are applied per feature map $\mathbf{H}_i^{(l)}$:

$$FiLM(\mathbf{H}_i^{(l)} | \gamma_i^{(l)}, \beta_i^{(l)}) = \gamma_i^{(l)} \mathbf{H}_i^{(l)} + \beta_i^{(l)}, \quad (1)$$

where $\mathbf{H}_i^{(l)} \in \mathbb{R}^{h \times w}$ is the feature map of filter i of convolution layer l and $\gamma^{(l)}, \beta^{(l)} \in \mathbb{R}^n$ are the modulation parameters from f_c , i.e., $(\gamma, \beta) = f_c(\mathbf{z})$, such that f_c is a neural network and \mathbf{z} is the question.

We adapt the FiLM model to the AQA task. Our FiLM model is composed of three convolutional blocks, identical to the VGG blocks in the audio only model, that are used to process the audio. A two-layered LSTM, with 512 or 1024 units per layer, encodes the question and produces the modulation

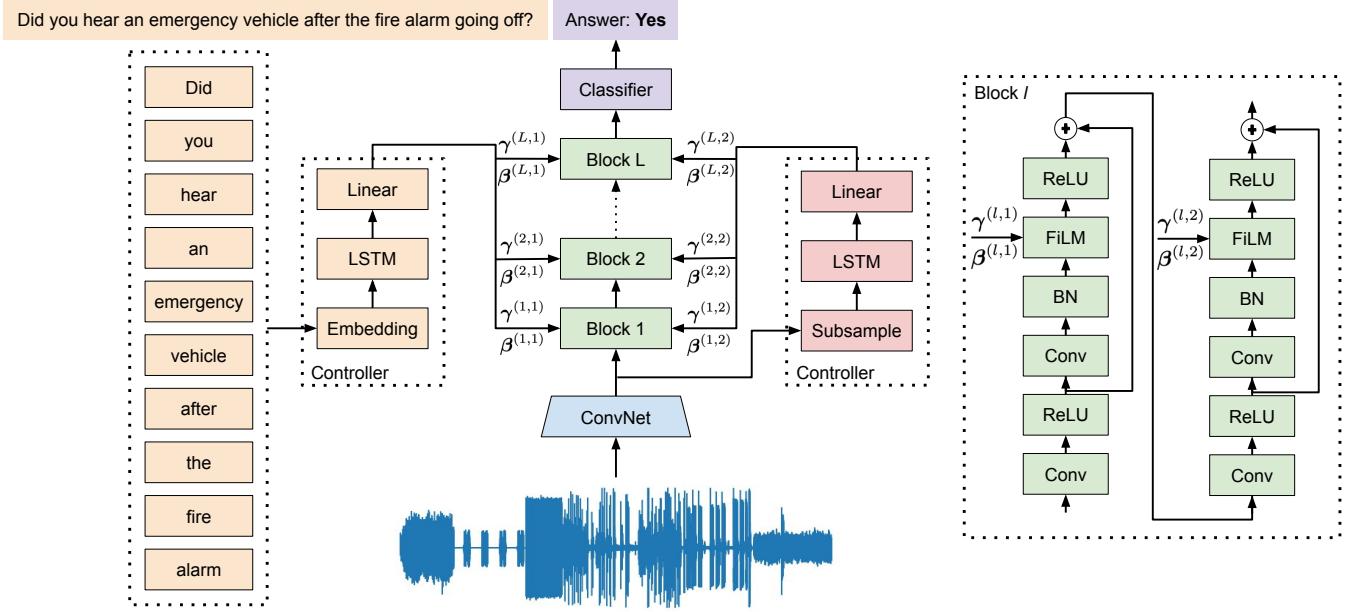


Fig. 6: Multi Auxiliary Controllers for Linear Modulation (MALiMo) for AQA. Two *auxiliary controllers*: one ingests the question text, and another operates on subsampled audio features, are used to modulate intermediate representations of a convolutional network that ingests the audio and predicts an answer.

parameters (γ, β) for residual convolutional blocks that ingest the processed representations from the preceding block. Each modulated block comprises a convolution layer with 128 3×3 filters and ReLUs, followed by another convolution layer with 128 3×3 filters, BatchNorm (with no affine transformation), feature-wise linear modulation, and ReLUs, and a residual connection from the output of the first ReLUs to output of the second ReLUs. Two coordinate feature maps indicating relative time and frequency positions respectively, scaled from -1 to 1 , were concatenated to the input in each modulated block. A convolution layer with 512 1×1 filters followed by global average pooling across each channel is applied to the output of the final modulated block and the output is fed to a fully connected neural network with a single hidden layer of 1024 units with ReLUs to predict an answer.

We evaluate FiLM exhaustively by varying the number of units in network f_c and the number of modulated blocks as shown in Table I. FiLM outperforms all previous baselines and models. The best FiLM model achieves an overall accuracy of 78.33%. Nevertheless, most of the classification error in the best performing FiLM model stems from questions and templates that require temporal reasoning as illustrated in Figure 9 and discussed later in Section VI.

V. MULTI AUXILIARY CONTROLLERS FOR LINEAR MODULATION (MALiMo)

MALiMo comprises a main module f_x and auxiliary controller(s) f_c . These can all be implemented as end-to-end differentiable neural networks. The main module f_x maps the principal input x to the output y . Each auxiliary controller $f_c^{(k)}$ ingests either the principal input x or another supplementary input $z^{(k)}$ and produces a set of parameters $(\gamma^{(k)}, \beta^{(k)})$ that modulate the intermediate representations in the main module

f_x , where k denotes the k^{th} controller. This allows computation in the main module f_x to be conditioned on the outputs (γ, β) of the auxiliary controller(s) f_c . Crucially, at least one auxiliary controller acts on the principal input x , or a processed version of the principal input, which enables the auxiliary controller to alter the intermediate representations in the main module acting on the principal input as a function of the principal input itself to facilitate relational and temporal reasoning, emulating the role of self-attention [33] and non-local operators [34]. The modulation parameters (γ, β) from the auxiliary controller(s) f_c are injected into the main module f_x as feature-wise linear transformations on the intermediate representations in a stacked manner in each modulated layer as follows.

Let $\mathbf{H}^{(l)} \in \mathbb{R}^{n \times h \times w}$ be the output of a linear convolution layer l that has n filters, the modulation parameters are applied per feature map $\mathbf{H}_i^{(l)}$ [14]:

$$MALiMo(\mathbf{H}_i^{(l)} | \gamma_i^{(l,k)}, \beta_i^{(l,k)}) = \gamma_i^{(l,k)} \mathbf{H}_i^{(l)} + \beta_i^{(l,k)}, \quad (2)$$

where $\mathbf{H}_i^{(l)} \in \mathbb{R}^{h \times w}$ is the feature map of filter i of convolution layer l and $\gamma^{(l,k)}, \beta^{(l,k)} \in \mathbb{R}^n$ are the modulation parameters from the auxiliary controller $f_c^{(k)}$, i.e., $(\gamma^{(l,k)}, \beta^{(l,k)}) = f_c^{(k)}(\mathbf{z})$, such that $f_c^{(k)}$ is a neural network and \mathbf{z} is the input to the controller $f_c^{(k)}$. Note that both Equation 1 and Equation 2 linearly modulate intermediate representations, but whereas FiLM uses a single controller, MALiMo uses multiple controllers.

MALiMo for Audio Question Answering. We instantiate MALiMo for AQA as follows. The model comprises a convolutional network f_x and two auxiliary controllers, as in Figure 6. The first auxiliary controller $f_c^{(1)}$ ingests the question to produce parameters $(\gamma^{(1,1)}, \beta^{(1,1)})$ that modulate the convolutional network. The second auxiliary controller

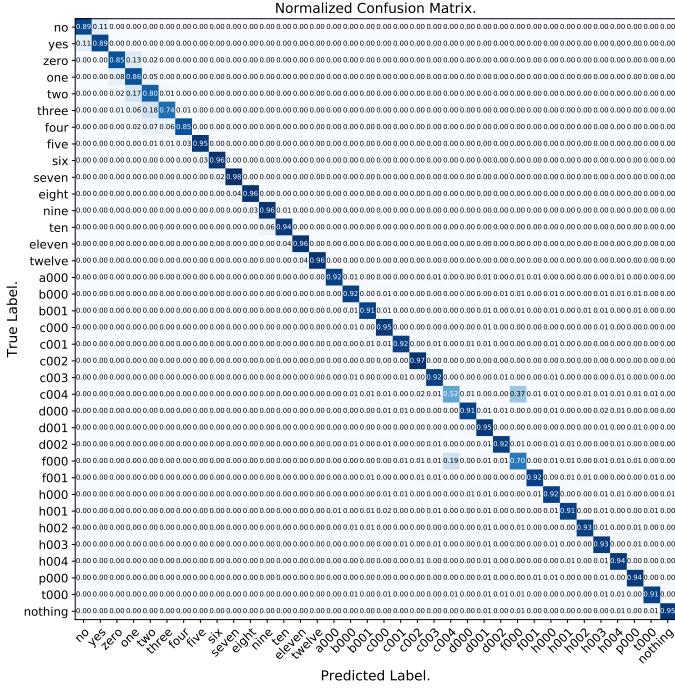


Fig. 7: Normalized confusion matrix for the MALiMo (6 Blocks) model on the DAQA test set. Errors are predominantly due to confusion between yes and no; 0, 1, and 2; and c004 (car passing by) and f000 (fire engine passing).

$f_c^{(2)}$ ingests a subsampled version of the processed audio to produce another set of parameters $(\gamma^{(2)}, \beta^{(2)})$ that modulate the convolutional network as well. Both controllers are implemented as LSTMs. The convolutional network ingests the audio to predict an answer.

Implementation Details. Audio is converted to MFSCs, which are processed via three convolutional blocks similar to those in VGG [39], such that each block comprises two convolution layers with BatchNorm and ReLUs, and a max pooling layer after the second convolution layer. The convolution layers in the first convolutional block have 32 filters and the number of filters in the convolution layers double in each subsequent convolutional block. The first convolution layer has filters of size 3×12 with stride 1×9 , whereas all remaining convolution layers have filters of size 3×3 with stride 1. The max pooling layers have a 2×2 window with stride 2.

The first auxiliary controller $f_c^{(1)}$ comprises a two-layered LSTM, with 512 units in each layer, that encodes 256-dimensional word embeddings of each word in the question into a fixed-sized representation of the question. This representation is mapped to the first set of modulation parameters $(\gamma^{(1)}, \beta^{(1)})$ via a linear layer.

The second auxiliary controller $f_c^{(2)}$ comprises a two-layered LSTM, with 512 units in each layer, that encodes the representation produced by the three convolutional blocks described above, subsampled using a mean pooling layer with a 8×8 window with stride 8, into a fixed-sized representation. This representation is mapped to the second set of modulation parameters $(\gamma^{(2)}, \beta^{(2)})$ via a linear layer.

A number of modulated convolutional MALiMo blocks ingest the output of the three convolutional blocks described above to produce a representation modulated by both auxiliary controllers $f_c^{(1)}$ and $f_c^{(2)}$. Each MALiMo block comprises a stack of FiLM layers, where each FiLM layer is composed of a convolution layer with 128 3×3 filters and ReLUs, followed by another convolution layer with 128 3×3 filters, BatchNorm (with no affine transformation), feature-wise linear modulation, and ReLUs, and a residual connection from the output of the first ReLUs to output of the second ReLUs, as illustrated in Figure 6. Two coordinate feature maps indicating relative time and frequency positions, scaled from -1 to 1, were concatenated to the input in each modulated block.

A convolution layer with 512 1×1 filters followed by global average pooling across each channel is applied to the output of the final modulated block and the output is fed to a fully connected neural network with a single hidden layer of 1024 units with ReLUs to predict an answer.

The experimental setup and all training details are identical to those described in Section IV.

Results. MALiMo is evaluated with varying numbers of modulated blocks as shown in Table I. MALiMo significantly outperforms all variants of FiLM and all other baselines.

Particularly, in comparison to the best FiLM model (FiLM-512; 12 layers), the best MALiMo model (6 Blocks) contains a larger number of parameters (13.20M vs 9.79M) due to the additional auxiliary controller but a similar number of layers⁴ and approximately equivalent training time, with a significant improvement in classification accuracy (88.86% vs 78.33%). Doubling the size of the auxiliary controller in FiLM (FiLM-1024) to account for the capacity of two auxiliary controllers in MALiMo does not result in an improvement in performance affirming that the improvement in classification accuracy is not simply due to the additional capacity in the model.

The confusion matrix for the MALiMo (6 Blocks) model on the DAQA test set is depicted in Figure 7, where we observe that most errors are predominantly due to confusion between yes and no; integers: 0, 1, and 2; and events: c004 (car passing by) and f000 (fire engine passing by).

VI. DISCUSSION AND FUTURE WORK

Saliency Visualization. Saliency visualizations in Figure 8 are used to provide insight into where MALiMo attends to when answering a question about the audio. Precisely, the first spectrogram is the time frequency representation of an audio input from the validation set annotated for clarity, whereas the following spectrograms are the same spectrogram superimposed with the norms of the gradient of the predicted answer scores with respect to the final convolution layer in the main module before the classifier, upsampled to match the spectrogram size, for four different questions and answers on the same audio input. The saliency visualizations show that the model learned to attend to the times and events of interest when performing

⁴A MALiMo model contains roughly the same number of layers (subtracting the additional auxiliary controller) as a FiLM model with double the number of layers (e.g. FiLM (6 layers) and MALiMo (3 Blocks)), since a MALiMo block is composed of a stack of two FiLM layers.

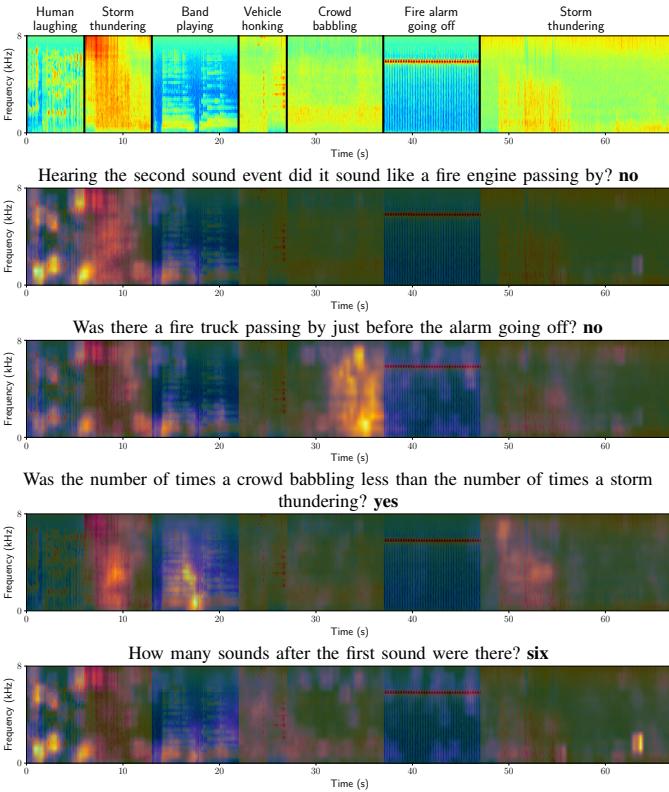


Fig. 8: Saliency visualizations to highlight where MALiMo (6 blocks) attends to when answering a question about the audio. The first spectrogram is the time frequency representation of the audio input annotated for clarity. The following spectrograms are the same spectrogram superimposed with the norms of the gradient of the predicted answer scores with respect to the final convolution layer in the main module before the classifier, upsampled to match the spectrogram size, for four different questions and answers on the same audio input. The saliency visualizations show that the model learned to attend to the times and events of interest when performing temporal reasoning. Examples are from the validation set.

temporal reasoning. For instance, in the second question in Figure 8 *Was there a fire truck passing by just before the alarm going off?*, the model attends mostly to the segment *just before the fire alarm going off* in the spectrogram to predict the correct answer *no*.

Temporal Reasoning. The convolutional network in MALiMo ingests the audio and is modulated using two auxiliary controllers: one ingests the question text, and another operates on subsampled audio features, to predict an answer. The latter controller is crucial for relational and temporal reasoning skills as it conditions the processing in the convolutional network on the audio, which facilitates dealing with variable-sized audio and a variable number of sound events. Figure 9 plots the classification accuracy per template for FiLM (FiLM-512; 12 layers) and MALiMo (6 blocks). It can be seen that the majority of improvement in MALiMo over FiLM is in the templates that require temporal reasoning, e.g. contain prepositions: before / after, or ordinals, highlighted in gray in Figure 9.

Low Resource Audio Question Answering. All models

TABLE II: Audio question answering performance on the DAQA test set using the low resource training set. # Blocks indicates the number of MALiMo blocks. The corresponding # Params is listed in Table I. A significant drop in classification accuracy is evident compared with the same models trained using the full training set.

# Blocks	Exist	Query	Count	Comp	CompInt	All (%)
1	58.12	23.63	36.14	52.31	52.70	44.93
2	55.94	13.26	37.40	51.17	52.88	42.67
3	52.53	11.93	34.60	53.97	50.94	41.79
4	55.53	13.38	36.87	50.65	52.29	42.28
5	55.09	12.82	34.16	51.10	52.73	41.61
6	54.35	11.97	35.06	51.74	51.25	41.63

presented so far were trained using all 80,000 audio clips and 399,924 questions and answers in the DAQA training set, which roughly corresponds to 5 (question, answer) pairs per audio clip. A more challenging setting is to reason and generalize using a smaller training set. To this end, we propose a new smaller training set, called low resource training set, that comprises 80,000 audio clips and 80,000 questions and answers, which corresponds to a (question, answer) pair per audio clip. In both cases, the validation and test sets are identical to those described in Section III.

We train MALiMo using the low resource training set and evaluate the models using the DAQA test set. The experimental setup and all training details are identical to those described in Section IV. Table II lists the classification accuracy on the DAQA test set for a number of MALiMo models. A significant drop in classification accuracy is evident compared with the same models trained using the full training set listed in Table I. These results indicate that developing models that can reason and generalize from a relatively small dataset remains a challenging task for future work.

Selective Computation. The models studied herein are capable of dealing with variable-length audio up to approximately 3 mins; however, the models must process the entire audio clip to produce an answer, which may not be efficient for cases such as, *What was the first sound?* or *What was the last sound?*. Thus, models that can adaptively navigate the audio clip, without necessarily processing the entire clip, would be a desirable avenue for future work [40], [41].

VII. CONCLUSION

We demonstrate that the AQA task can be used to study the temporal reasoning abilities of machine learning models. We introduce the DAQA dataset that comprises audio sequences of natural sound events and programmatically generated questions and answers that probe various aspects of temporal reasoning. We show that several recent state-of-the-art methods for VQA adapted to the AQA task perform poorly, particularly on questions that require in-depth temporal reasoning. We propose a new model, MALiMo, that utilizes auxiliary controllers to modulate the intermediate representations of a convolutional network processing the audio conditioned on the question and the audio itself. Experiments on DAQA show that MALiMo significantly outperforms other competing approaches, and that

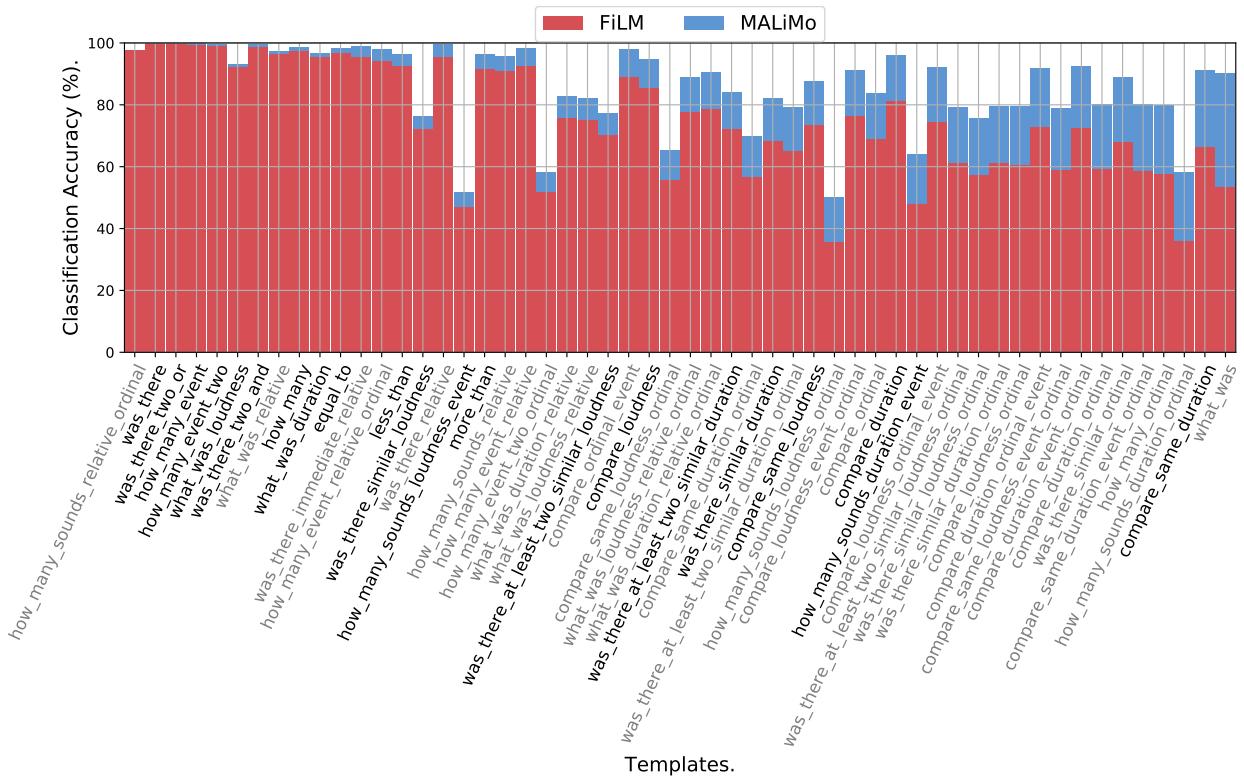


Fig. 9: Audio question answering performance per template on the DAQA test set for FiLM (FiLM-512; 12 layers) and MALiMo (6 blocks). Templates highlighted in gray require temporal reasoning, e.g. contain prepositions: before / after, or ordinals.

it leads to the largest improvements on questions that require in-depth temporal reasoning. Avenues for future work include learning to reason and generalize from small datasets and developing models that can process only relevant segments of the audio as opposed to the entire audio clip. We envisage DAQA to foster research on AQA and temporal reasoning and MALiMo a step towards models for AQA.

REFERENCES

- it leads to the largest improvements on questions that require in-depth temporal reasoning. Avenues for future work include learning to reason and generalize from small datasets and developing models that can process only relevant segments of the audio as opposed to the entire audio clip. We envisage DAQA to foster research on AQA and temporal reasoning and MALiMo a step towards models for AQA.

REFERENCES

 - [1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh, “VQA: Visual question answering,” in *ICCV*, 2015.
 - [2] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, “CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning,” in *CVPR*, 2017.
 - [3] Y. Zhu, O. Groth, M. Bernstein, and L. Fei-Fei, “Visual7W: Grounded question answering in images,” in *CVPR*, 2016.
 - [4] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, “Squad: 100,000+ questions for machine comprehension of text,” in *EMNLP*, 2016.
 - [5] J. Weston, A. Bordes, S. Chopra, A. M. Rush, B. van Merriënboer, A. Joulin, and T. Mikolov, “Towards ai-complete question answering: A set of prerequisite toy tasks,” in *ICLR*, 2016.
 - [6] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, “Embodied question answering,” in *CVPR*, 2018.
 - [7] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler, “MovieQA: Understanding stories in movies through question-answering,” in *CVPR*, 2016.
 - [8] J. P. Bigham, C. Jayant, H. Ji, G. Little, A. Miller, R. C. Miller, R. Miller, A. Tatarowicz, B. White, S. White, and T. Yeh, “VizWiz: nearly real-time answers to visual questions,” in *UIST*, 2010.
 - [9] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham, “VizWiz grand challenge: Answering visual questions from blind people,” in *CVPR*, 2018.
 - [10] W. S. Lasecki, P. Thihia, Y. Zhong, E. Brady, and J. P. Bigham, “Answering visual questions with conversational crowd assistants,” in *SIGACCESS*, 2013.
 - [11] D. G. Barrett, F. Hill, A. Santoro, A. S. Morcos, and T. Lillicrap, “Measuring abstract reasoning in neural networks,” in *ICML*, 2018.
 - [12] G. R. Yang, I. Ganichev, X.-J. Wang, J. Shlens, and D. Sussillo, “A dataset and architecture for visual reasoning with a working memory,” in *ECCV*, 2018.
 - [13] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, “Stacked attention networks for image question answering,” in *CVPR*, 2016.
 - [14] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, “FiLM: Visual reasoning with a general conditioning layer,” in *AAAI*, 2018.
 - [15] P. Rajpurkar, R. Jia, and P. Liang, “Know what you don’t know: Unanswerable questions for squad,” in *ACL*, 2018.
 - [16] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh, “Making the v in vqa matter: Elevating the role of image understanding in visual question answering,” in *CVPR*, 2017.
 - [17] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei, “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” *IJCV*, 2017.
 - [18] M. Malinowski and M. Fritz, “A multi-world approach to question answering about real-world scenes based on uncertain input,” in *NeurIPS*, 2014.
 - [19] J. Lei, L. Yu, M. Bansal, and T. L. Berg, “TVQA: Localized, compositional video question answering,” in *EMNLP*, 2018.
 - [20] K.-H. Zeng, T.-H. Chen, C.-Y. Chuang, Y.-H. Liao, J. C. Niebles, and M. Sun, “Leveraging video descriptions to learn video question answering,” in *AAAI*, 2017.
 - [21] L. Zhu, Z. Xu, Y. Yang, and A. G. Hauptmann, “Uncovering the temporal context for video question answering,” *IJCV*, 2017.
 - [22] D. A. Hudson and C. D. Manning, “Gqa: A new dataset for real-world visual reasoning and compositional question answering,” *CVPR*, 2019.
 - [23] T. Zhang, D. Dai, T. Tuytelaars, M.-F. Moens, and L. Van Gool, “Speech-based visual question answering,” *arXiv preprint arXiv:1705.00464*, 2017.
 - [24] J. Abdelnour, G. Salvi, and J. Rouat, “CLEAR: A dataset for compositional language and elementary acoustic reasoning,” *arXiv preprint arXiv:1811.10561*, 2018.
 - [25] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang, “Bottom-up and top-down attention for image captioning and visual question answering,” in *CVPR*, 2018.

- [26] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach, “Multimodal compact bilinear pooling for visual question answering and visual grounding,” in *EMNLP*, 2016.
- [27] M. Malinowski, M. Rohrbach, and M. Fritz, “Ask your neurons: A neural-based approach to answering questions about images,” in *ICCV*, 2015.
- [28] A. Singh, V. Natarajan, Y. Jiang, X. Chen, M. Shah, M. Rohrbach, D. Batra, and D. Parikh, “Pythia-a platform for vision & language research,” in *NeurIPS SysML Workshop*, 2019.
- [29] D. A. Hudson and C. D. Manning, “Compositional attention networks for machine reasoning,” in *ICLR*, 2018.
- [30] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick, “Inferring and executing programs for visual reasoning,” in *ICCV*, 2017.
- [31] A. Santoro, D. Raposo, D. G. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, “A simple neural network module for relational reasoning,” in *NeurIPS*, 2017.
- [32] H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville, “Modulating early visual processing by language,” in *NeurIPS*, 2017.
- [33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *NeurIPS*, 2017.
- [34] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *CVPR*, 2018.
- [35] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *CVPR*, 2018.
- [36] 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.
- [37] B. R. Glasberg and B. C. Moore, “A model of loudness applicable to time-varying sounds,” *Journal of the Audio Engineering Society*, vol. 50, no. 5, pp. 331–342, 2002.
- [38] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *ICLR*, 2015.
- [39] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *ICLR*, 2015.
- [40] V. Mnih, N. Heess, and A. Graves, “Recurrent models of visual attention,” in *NeurIPS*, 2014.
- [41] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, “End-to-end learning of action detection from frame glimpses in videos,” in *CVPR*, 2016.

APPENDIX A DIAGNOSTIC AUDIO QUESTION ANSWERING (DAQA) EXAMPLES

Figure 10 is a number of random examples from the training set of DAQA.

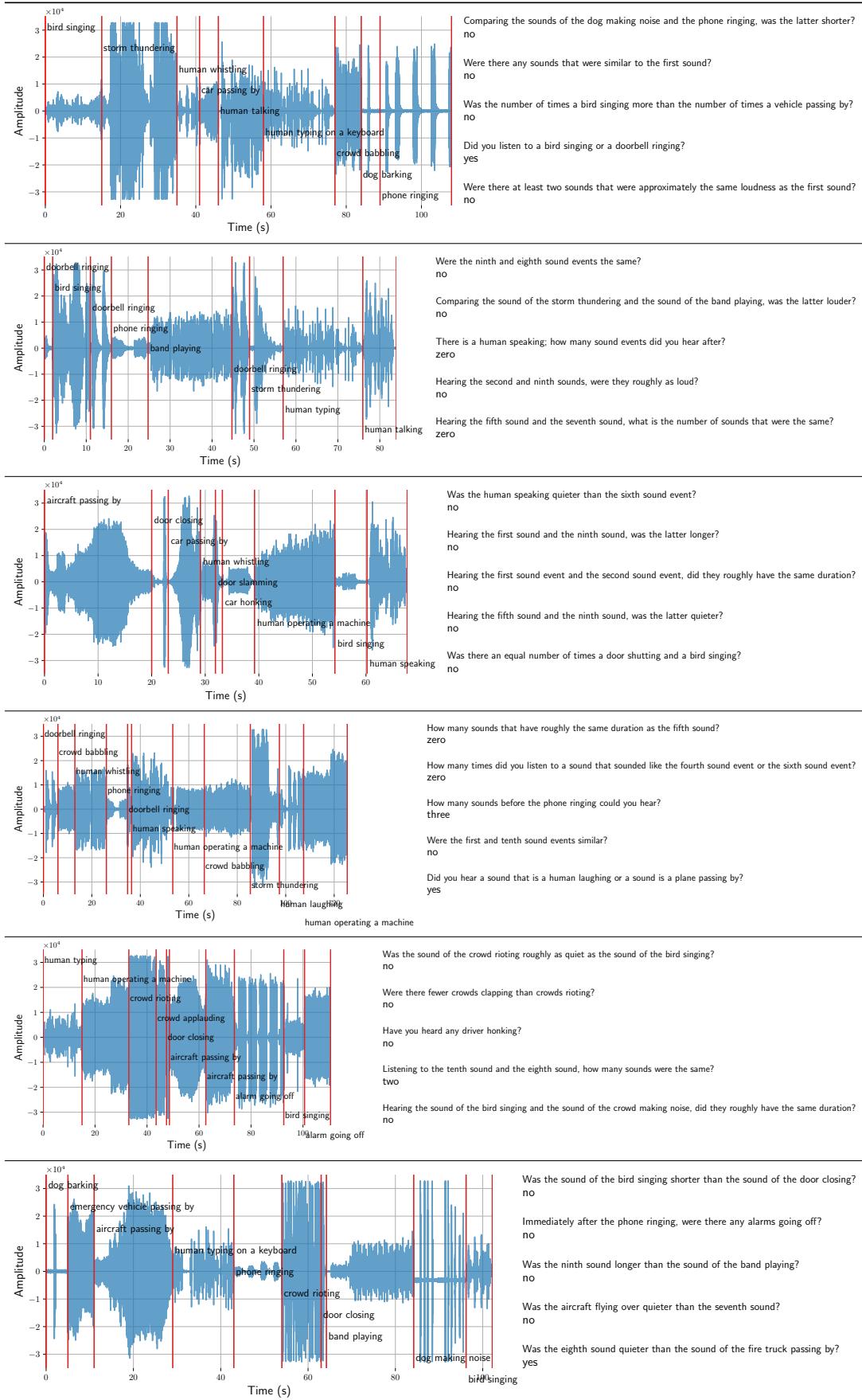


Fig. 10: Audio clips composed of natural sound events (left) and programmatically generated questions and answers about those audio clips that test various aspects of temporal reasoning (right).