NAAQA: A Neural Architecture for Acoustic Question Answering

Jérôme Abdelnour, Jean Rouat, *Senior Member, IEEE*, and Giampiero Salvi, *Member, IEEE*

Abstract—The goal of the Acoustic Question Answering (AQA) task is to answer a free-form text question about the content of an acoustic scene. It was inspired by the Visual Question Answering (VQA) task. In this paper, based on the previously introduced CLEAR dataset, we propose a new benchmark for AQA, namely CLEAR2, that emphasizes the specific challenges of acoustic inputs. These include handling of variable duration scenes, and scenes built with elementary sounds that differ between training and test set. We also introduce NAAQA, a neural architecture that leverages specific properties of acoustic inputs. The use of 1D convolutions in time and frequency to process 2D spectro-temporal representations of acoustic content shows promising results and enables reductions in model complexity. We show that time coordinate maps augment temporal localization capabilities which enhance performance of the network by ∼17 percentage points. On the other hand, frequency coordinate maps have little influence on this task. NAAQA achieves 79.5% of accuracy on the AQA task with ∼four times fewer parameters than the previously explored VQA model. We evaluate the performance of NAAQA on an independent data set reconstructed from DAQA. We also test the addition of a MALiMo module in our model on both CLEAR2 and DAQA. We provide a detailed analysis of the results for the different question types. We release the code to produce CLEAR2 as well as NAAQA to foster research in this newly emerging machine learning task.

Index Terms—Audio, question answering, reasoning, temporal reasoning, CLEAR, coordconv, auditory scene analysis

1 Introduction

Question answering (QA) tasks are examples of constrained and limited scenarios for research in reasoning. The agent's task in QA is to answer questions based on context. Text-based QA uses text corpora as context [1], [2], [3], [4], [5], [6]. In visual question answering (VQA) the questions are related to a scene depicted in still images [7], [8], [9], [10], [11], [12], [13]. Finally, video question answering attempts to use both the visual and acoustic information in video material as context [14], [15], [16], [17], [18], [19]. The use of the acoustic channel is usually limited to linguistic information that is expressed in text form, either with manual transcriptions (e.g., subtitles) or by automatic speech recognition [20].

In most studies, reasoning is supported by spatial and symbolic representations in the visual domain [21], [22]. However, reasoning and logic relationships can also be studied via representations of sounds [23]. Including the auditory modality in studies on reasoning is of particular interest for research in artificial intelligence [24], but also has implications in real world applications [25]. In [26], audio was used in combination with video and depth information to recognize human activities. It was shown that sound can be more discriminative than the corresponding visual cues. As an example, imagine using an espresso machine. Besides possibly a display, all information about the different phases of producing coffee, from grinding the beans, to pressing the powder into the holder and brewing the coffee with high pressure hot water are conveyed by the sounds. Detection of abnormalities in machinery where the moving parts are hidden, or the detection of threatening or hazardous events are other examples of the importance of the audio information for cognitive systems.

The audio modality provides important information that can be leveraged in the context of QA reasoning. Audio allows QA systems to answer relevant questions more accurately, or even to answer questions that are not approachable from the visual domain alone. In [27], we introduced the AQA task and proposed a new database (CLEAR) to promote research in AQA. The agent's goal, in the proposed task, was to answer questions related to *acoustic scenes* composed by a sequence of *elementary musical sounds*. The questions foster reasoning on the properties of the elementary sounds and their relative and absolute position in the scene. To build CLEAR, we were inspired by the work of Johnson *et al.* [7] for VQA. Similarly, we tested an architecture built for VQA and based on *FiLM layers* [28] on the newly proposed AQA task. Fayek and Johnson [29] later proposed to extend the questions to more acoustically realistic situations by developing a new database called DAQA. To evaluate the results, they proposed the MALiMo network which relies on several FiLM layers.

- *Jérôme Abdelnour and Jean Rouat are with the NECOTIS, Department of Electrical and Computer Engineering, Sherbrooke University, Sherbrooke, QC J1K 2R1, Canada. E-mail: {Jerome.Abdelnour, Jean.Rouat}@usherbrooke.ca.*
- *Giampiero Salvi is with the Department of Electronic Systems, Norwegian University of Science and Technology, 7034 Trondheim, Norway, and also with the Department of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 114 28 Stockholm, Sweden. E-mail: giampiero.salvi@ntnu.no.*

Manuscript received 28 April 2021; revised 20 May 2022; accepted 21 July 2022. Date of publication 19 September 2022; date of current version 6 March 2023.
This work was supported by the CHIST-ERA IGLU Project, by CRSNG, by the Michael-Smith scholarships, and by the Universities of Sherbrooke and NTNU.
(Corresponding author: Jérôme Abdelnour.)
Recommended for acceptance by J. Glass.
Digital Object Identifier no. 10.1109/TPAMI.2022.3194311

The works cited above use neural network architectures that are largely inspired by image processing research. However, the structure of acoustic data is fundamentally different from that of visual data. This is illustrated for example in [30] where two standard data sets in computer vision (MNIST) and speech technology (Google Speech Commands) are compared via T-SNE [31]. A legitimate question is whether it is possible to obtain better results (in terms of accuracy and network complexity) by adapting the first layers of the architectures to take into account intrinsic characteristics of acoustic signals. Even within the AQA domain, the properties of acoustic data may vary significantly depending on the nature of the auditory scenes (e.g., CLEAR versus DAQA). It is, therefore interesting to evaluate the impact of the dataset on system performance.

To answer the above questions, we present a study that evaluates the impact of audio pre-processing, of acoustic feature extraction and of dataset characteristics on the performance neural architectures for AQA. When considering performance, we focus both on accuracy and complexity of the models. We provide a detailed analysis of our results based on question type to improve interpretability. The main contributions can be summarized as follows:

- We introduce CLEAR2 a more challenging version of the CLEAR dataset, which comprises scenes of variable duration and different elementary sounds for the training and test sets.
- We propose a highly optimized FiLM-based architecture (NAAQA) inspired by VQA tasks containing new feature extraction modules that are tailored to acoustic inputs.
- We study the effect of time and frequency coordinate maps for acoustic data at different levels in the architecture.
- We evaluate the generality of the methods by testing NAAQA on a regenerated a version of the DAQA dataset (DAQA′) and by adding a MALiMo module (from [29]) into our NAAQA architecture.
- We provide a detailed analysis of our experimental results that helps interpretability of the model.

On the CLEAR2 dataset NAAQA outperforms the VQA baseline (which is 4 times more complex in terms of number of parameters) by 17.2 percent points in the accuracy score.

The rest of the paper is organized as follows: Section 2 reports on recent related work, Section 3 describes both our CLEAR2 dataset and the DAQA′ dataset, Section 4 presents the QA models we have tested, Section 5 gives details on the experimental settings, Section 6 presents and discusses the results and, finally, Section 7 concludes the paper. Some extra information can be found in the supplementary material, on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TPAMI.2022.3194311.

2 Related Work

This section presents previous research in QA systems including data generation and modeling.

2.1 Text-Based Question Answering

The question answering task was introduced as part of the Text Retrieval Conference [1]. In text-based question answering, both the questions and the context are expressed in text form. Answering these questions can often be approached as a pattern matching problem in the sense that the information can be retrieved almost verbatim in the text (e.g., [3], [4], [5], [6]).

2.2 Visual Question Answering (VQA)

Visual Question Answering aims to answer questions based on a visual scene. Several VQA datasets are available to the scientific community [7], [8], [9], [10], [32], [33], [34], [35], [36], [37]. However, designing an unbiased dataset is non-trivial. Agrawal *et al.* [11] observed that the type of questions has a strong impact on the results of neural network based systems which motivated research to reduce the bias in VQA datasets [7], [12], [13], [38], [39], [40]. Gathering good labeled data is also non-trivial which induced Zhang *et al.* and Geman *et al.* [12], [13] to constrain their work to yes/no questions. To alleviate this problem, Johnson *et al.* [7] proposed the use of synthetic data for both questions and visual scenes. The resulting CLEVR dataset has been extensively used to evaluate neural networks for VQA applications [28], [41], [42], [43], [44], [45] which helped foster research on VQA. To create visual scenes, the authors automated a 3D modelling software. This allows for an unlimited supply of labeled data eliminating the time and effort needed for manual annotations. For the questions, they first manually designed semantic representations for each type of question. These representations describe the reasoning steps needed to answer a question (i.e., "find all cubes | that are red | and metallic"). The semantic representations are then instantiated based on the visual scene composition thus creating a question and an answer for a given scene. This gives complete control over the labelling process.

2.3 Dababases for AQA

As in VQA, using generated data in the design of AQA datasets has substantial advantages. Data can be automatically annotated which saves time and complexity. The number of training examples that can be generated is only limited by the available computational resources. Controlling the generation process gives a complete understanding of the properties and relations of the objects in a scene. This understanding can be leveraged to reduce bias in the dataset and to generate complex questions and their corresponding answers. The CLEAR dataset [27] has been initially generated using semi-synthetic data. The elementary sounds were real recordings of musical notes played by various instruments and players. The auditory scenes were obtained by concatenating these elementary sounds in different combinations. The data set had two main limitations: scenes had fixed duration, and the same elementary sounds were used to generate the test and training scenes (although test and training scenes were different). The DAQA dataset [29] comprises more complex and less stationary elementary natural sounds coming for example from aircrafts, cars, doors, human speaking, bird singing, dog barking, etc. Although more complex and varied than CLEAR, the evaluation also uses the same elementary sound recordings for training and testing.

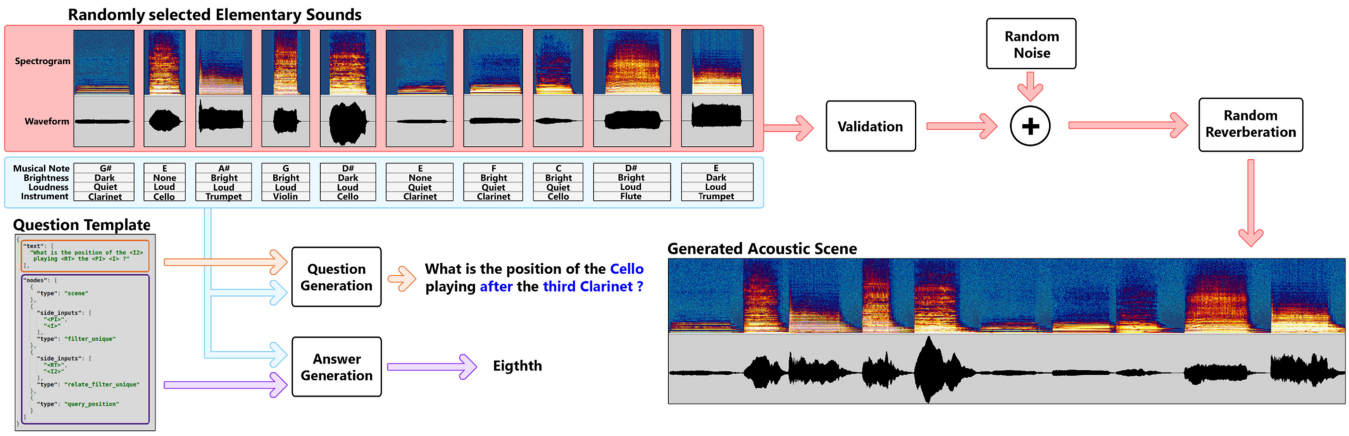


Fig. 1. *Overview of the CLEAR dataset generation process.* Highlighted in red: ten randomly sampled sounds from the elementary sounds bank, are assembled to create an acoustic scene. The attributes of each elementary sound are depicted in blue. The question template (orange) and the elementary sounds attributes are combined to instantiate a question. The answer is generated by applying each steps of the question functional program (purple) on the acoustic scene definition (blue). The impact of the reverberations can be seen in the changes of the signals envelops.

In this paper, we propose a more challenging version of CLEAR which uses different elementary sound recordings for the training and test sets and generates variable duration auditory scenes.

2.4 Convolutional Neural Network on Audio

Convolutional neural networks (CNN) have dominated the visual domain in recent years. More recently, they have also been applied to a number of problems in the acoustic domains such as acoustic scene classification [46], [47], [48], [49], music genre classification [48], [50], instrument classification [51], [52], sound event classification and localization [53] and speech recognition [48]. Some authors [49], [50], [51], [52] use intermediate representations such as STFT [54], MFCC [55] or CQT [56] spectrograms while others work directly with the raw audio signal [46], [48].

Square convolutional and pooling kernels are often used to solve visual task such as VQA, visual scene classification and object recognition [57], [58], [59], [60]. oddapati *et al.*, Hershey *et al.*, Kumar and Raj [47], [61], [62] have successfully used visually motivated CNN with square filters to solve audio related tasks. Time-frequency representations of audio signals are however structured very differently than visual representations. Pons *et al.* [63] explore the performance of different structures of convolutive kernels when working with music signals classification. They propose the use of 1D convolution kernels to capture time-specific or frequency-specific features. They demonstrate that similar accuracy can be reached using a combination of 1D convolutions instead of 2D convolutions by combining 1D time and 1D frequency filters while using much fewer parameters. They also explore rectangular kernels which capture both time and frequency features at different scales. The impact of such strategies for music classification is still an open question in the context of auditory scene analysis.

Coordinate maps initially proposed in [64] by Liu *et al.* have proven successful for processing visual data in the context of VQA. The method consists in augmenting the visual input with matrices containing numbers in the range -1 to 1 which vary either in the x or in the y-dimension. With MALiMo [29], the same strategy is used to indicate the simultaneous relative positions of features in frequency and time. Koutini *et al.* [65] proposed *Frequency-Aware convolutions* which are equivalent to concatenating coordinate maps only in the *frequency* axis. The effectiveness of coordinate maps on the time dimension for audio signals have not been studied to the best of our knowledge.

In this study we first evaluate the performance of a network initially designed for the VQA task (Visual FiLM) [28] on the AQA task, using the CLEAR2 data set. Then we introduce the NAAQA architecture to leverage specific properties of acoustic inputs. For this architecture, we analyze the influence of separate time and frequency coordinate maps. We then study the impact of adding a MALiMo block into our architecture. Finally, we evaluate our model on the DAQA′ dataset.

3 Data

We use two very different datasets in our experiments in order to study the effect of the AQA task characteristics on the model performance. The first set, that is also a contribution of this paper, comprises musical sounds (CLEAR2); the second includes short environmental sounds (DAQA′).

3.1 CLEAR2

CLEAR2 is an updated version of CLEAR [27]. A graphical overview of the generation process is depicted in Fig. 1. Each record in the dataset is a unique combination of a scene, a question and an answer.

To build acoustic scenes, we prepared a bank of elementary sounds composed of real musical instrument recordings extracted from the Good-Sounds [66] dataset[1]. Differently from CLEAR, in CLEAR2 we make sure that the

1. Each elementary sound in a scene is characterized by an n-tuple: [*Instrument*, *Brightness*, *Loudness*, *Musical Note*, *Duration*, *Absolute position in scene*, *Relative position in scene*, *Global position*]. The *Brightness* property is computed by using the `timbralmodels` [67] library. A threshold is used to define the label of the sound (*Dark* or *Bright*). The *Loudness* labels are assigned based on the perceptual loudness as defined by the ITU-R BS.1770-4 international normalization standard [68]. Again, a threshold is used to determine if the sound is *Quiet* or *Loud*. All attribute values are listed in Table 1 as possible answers to the questions explained below.

TABLE 1
Types of Questions with Examples and Possible Answers

Question type	Example	Possible Answers	#
Note	What is the note played by the *flute* that is *after* the *loud bright D* note?	A, A#, B, C, C#, D, D#, E, F, F#, G, G#	12
Instrument	What instrument plays a *dark quiet* sound in the *end* of the scene?	bass, cello, clarinet, flute, trumpet, violin	5
Brightness	What is the brightness of the *first clarinet* sound?	bright, dark	2
Loudness	What is the loudness of the *violin* playing *after* the *third trumpet*?	quiet, loud	2
Absolute Position	What is the position of the *A#* note playing *before* the *bright B* note?	first, second ... fifteenth	15
Relative Position	Among the *trumpet* sounds which one is a *F*?		
Global Position	In what part of the scene is the *clarinet* playing a *loud G* note ?	beginning, middle, end (of the scene)	3
Counting	How many other sounds have the same brightness as the *third violin*?	0, 1 ... 15	16
Counting Instruments	How many different instruments are playing *before* the *second trumpet*?		
Exist	Is there a *bass* playing a *bright C#* note?	yes, no	2
Counting comparison	Is there an *equal* number of *loud cello* sounds and *quiet clarinet* sounds?		
Total			57

The variable parts of each question is emphasized in bold italics. The number of possible answer per question type is reported in the last column. Certain questions have the same possible answers, the meaning of which depends on the type of question.

recordings (players, instruments, microphones) of the elementary sounds are different for the training and test sets. For the training set, the bank comprises 135 unique recordings (compared to 56 in CLEAR) sampled at 48KHz including 6 different instruments (bass, cello, clarinet, flute, trumpet and violin), 12 notes (chromatic scale) and 3 octaves. A different set of 135 recordings of the same instruments recorded using different microphones and players is used to create the test set. The acoustic scenes are built by concatenating between 5 to 15 randomly chosen sounds from the elementary sound bank into a sequence (as opposed to CLEAR which comprised fixed duration scenes). Silence segments of random duration are added in-between elementary sounds. The acoustic scenes are then corrupted by filtering to simulate room reverberation and by adding a white uncorrelated uniform noise. Both the amount of noise and reverberation vary from scene to scene with the goal of increasing variability in the data.

For each scene, a number of questions is generated using CLEVR-like [7] templates. A template defines the reasoning steps required to answer a question based on the composition of the scene (i.e., "find all instances of violin | that plays before trumpet | that is the loudest"). 942 templates where designed for this AQA task. Not all template instantiations results in a valid question. The generated questions are filtered to remove ill posed questions similarly to [7]. Table 1 shows examples of questions with their answers.

The a priori probability of answering correctly with no information about the question or the scene, and assuming a uniform distribution of classes, is $\frac{1}{57} = 1.75\%$. These probabilities are higher, on average, if we introduce information about the question. For example, if we know that the question is of the type *Exist* or *Counting comparison*, there are only two possible answers (yes or no) and the probability of answering correctly by chance is 0.5. The majority class accuracy (always answering the most common answer: Yes) is 7.5%. Statistics on the CLEAR2 dataset are presented in Table 2.

The generation process was built with extensibility in mind. Different versions of the dataset with fewer or more objects per scene can be generated by using different parameters for the generation script. It is also possible to modify the elementary sounds bank to generate datasets for AQA in other domains, speech or environmental sounds, for example. The code for generating the dataset is available on GitHub.[2] Pre-generated version of the dataset is available both on IEEE Dataport[3] and HuggingFace.[4]

3.2 Reproducing DAQA

We were not able to fully recreate the DAQA dataset because it relies on some AudioSet [69] YouTube videos that have since been deleted. We were able to retrieve 358 sounds out of the 400 sounds that were used to generate the original dataset. We used these sounds to generate the dataset. Changing the number of elementary sounds also impacts the whole generation process. This dataset is therefore different from the original DAQA and will be referred to as DAQA′ from now on. Our results are therefore not fully comparable to the ones reported in [29]. A list of all the missing sounds is available in Supplementary Materials.

3.3 Comparing CLEAR2 and DAQA′

Table 2 report statistics on both CLEAR2 and DAQA′. The major difference between both datasets is the type of elementary sounds used to generate the acoustic scenes, that is, sustained musical notes (CLEAR2) versus possibily transient environmental sounds (DAQA′).

Acoustic scenes in DAQA′ are much longer on average than the ones in CLEAR2. This results in much bigger input spectrograms, and, in turns, much higher computational requirements and longer training time.

Finally, the original DAQA [29] and, consequently, our reconstruction (DAQA′) suffer from the same problem as the original CLEAR. The same elementary sounds are used in the training and test scenes. Although scenes are still different between training and test set, this may cause the models to "remember" the elementary sounds rather than extracting their properties. In CLEAR2, this problem was mitigated by using different elementary sounds for the training and test set.

2. https://github.com/NECOTIS/CLEAR-AQA-Dataset-Generator
3. https://dx.doi.org/10.21227/7x26-a025
4. https://huggingface.co/datasets/J3romee/CLEAR

TABLE 2
Datasets Statistics

Datasets global statistics		
	Dataset	
	CLEAR2	DAQA′
# of questions	200 000	599 441
# of scenes	50 000	100 000
# of answers	57	52
# of elementary sounds	135 (+ 135 for test)	358
# of types of question	11	5
# of unique vocabulary words	91	158

CLEAR2 Dataset detailed statistics			
	Mean	Min	Max
# of sounds per scene	10	5	15
Elementary sound duration	0.85s	0.69s	1.11s
Scene duration	10.69s	4.46s	17.82s
# of words per question	17	6	28
# of unique words per question	12	5	19

DAQA′ Dataset detailed statistics			
	Mean	Min	Max
# of sounds per scene	9	5	12
Elementary sound duration	9.35s	0.6s	20s
Scene duration	1min 19s	9s	3min 4s
# of words per question	13	5	27
# of unique words per question	11	5	22

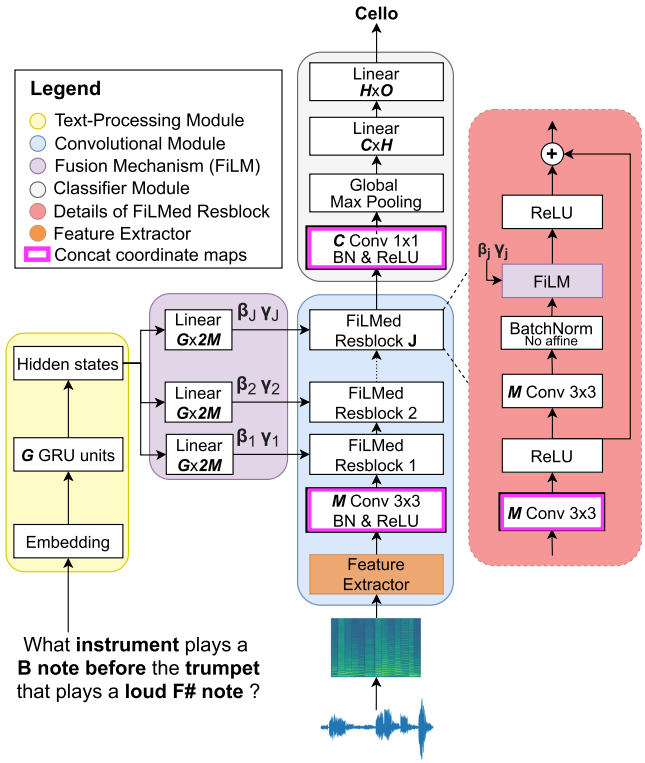


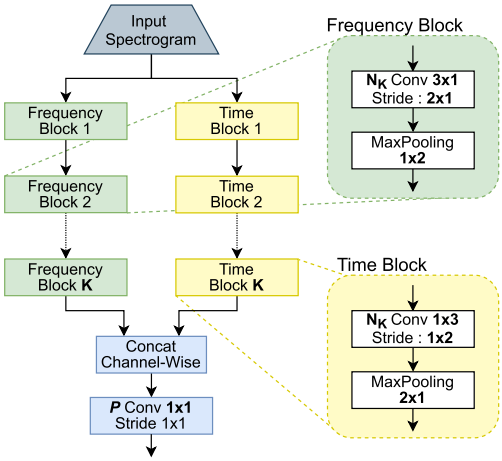
Fig. 2. *Common Architecture.* Two inputs: a spectro-temporal representation of an acoustic scene and a textual question. The spectro-temporal representation goes through a feature extractor (*Parallel* and *Interleaved* feature extractor detailed in Section 4.2.1 for NAAQA and Resnet101 pretrained on ImageNet for Visual FiLM) and then a serie of J Resblocks that are linearly modulated by $\boldsymbol{\beta}_j$ and $\boldsymbol{\gamma}_j$ (learned from the question input) via FiLM layers. Coordinate maps are inserted before convolution blocks that are illustrated with a pink border. The output is a probability distribution of the possible answers.

4 Method

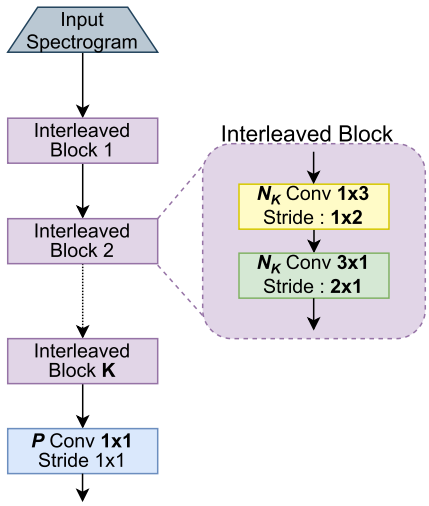
We first describe the original Visual FiLM architecture [28] that we use as baseline model, then the proposed modifications that lead to our NAAQA architecture and, finally, NAAQA with a MALiMo module.

4.1 Baseline Model: Visual FiLM

Both the proposed NAAQA and Visual FiLM [28] share an overall common architecture which is depicted in Fig. 2. Visual FiLM, that we will use as baseline model, is inspired by Conditional Batch Normalization architectures [70] and achieved state of the art results on the CLEVR VQA task [7]. The network takes a visual scene and a text-based question as inputs and predicts an answer to the question for the given scene. The text-processing module uses G unidirectional gated recurrent units (GRUs) to extract context from the text input (yellow area in Fig. 2). The visual scene is processed by the convolutional module (blue area in the figure). The first step of this module is feature extraction (orange box), performed by a Resnet101 model [59] pre-trained on ImageNet [71]. The extracted features are processed by a convolutional layer with batch normalization [70] and ReLU [72] activation followed by J Resblocks illustrated in details in the red area in the figure. Unless otherwise specified, batch normalization and ReLU activation functions are applied to all convolutional layers. Each Resblock j comprises convolutional layers with M filters that are linearly modulated by *FiLM layers* through the two $M \times 1$ vectors $\boldsymbol{\beta}_j$ (additive) and $\boldsymbol{\gamma}_j$ (multiplicative). This modulation emphasizes the most important feature maps and inhibits the irrelevant maps given the context of the question. $\boldsymbol{\beta}_j$ and $\boldsymbol{\gamma}_j$ are learned via fully connected layers using the text embeddings extracted by the text processing module as inputs (purple area in the figure). The affine transformation in the batch normalization before the FiLM layer is deactivated. The FiLM layer applies its own affine transformation using the learned $\boldsymbol{\beta}_j$ and $\boldsymbol{\gamma}_j$ to modulate features. Several Resblocks can be stacked to increase the depth of the model, as illustrated in Fig. 2. Finally, the classifier module is composed of a 1×1 convolutional layer [73] with C filters followed by max pooling and a fully connected layer with H hidden units and an output size O equal to the number of possible answers (Gray in Fig. 2). A softmax layer predicts the probabilities of the answers. In order to use the Visual FiLM as a baseline for our experiments, we extract a 2D spectro-temporal representation of the acoustic scenes as depicted at the bottom of Fig. 2. The Resnet101 pre-trained extractor expects a 3 channels visual input but the spectro-temporal representation comprises only 1 channel. To work around this constraint, the spectro-temporal information is simply repeated 3 times thus creating a 3 channels input (only when using Resnet101 as feature extractor). This modified spectro-temporal representation is then fed to the model as if it was an image which is the simplest way to adapt the



(a) **Parallel feature extraction.** The input spectrogram is processed by 2 parallel pipelines. The first pipeline (in green) captures *frequency* features using a serie of K 1D convolutions with N_k filters and a stride of 2×1. Since the stride is larger than 1×1, each convolution downsample the frequency axis. The 1×2 maxpooling then downsamples the time axis. The second pipeline (in yellow) captures *time* features using the same structure with transposed filter size. Features from both pipelines are concatenated and fused using a 1×1 convolution with P filters to create.



(b) **Interleaved feature extraction.** 1D time (in yellow) and frequency (in green) convolutions are applied alternately on the input spectrogram building a **time-frequency** representation after each block. The order of the convolution in each block can be reversed. The extractor is composed of K blocks where each convolution has N_k filters followed by a 1×1 convolution with P filters.

Fig. 3. Acoustic feature extraction.

unmodified visual architecture to acoustic data. We call this architecture **`Visual FiLM Resnet101`**.

4.2 The Proposed NAAQA Architecture

To create the NAAQA architecture, we made modifications to the baseline architecture that will be described in the following sections. The code is available on GitHub.[5]

4.2.1 Feature Extraction

As in Visual FiLM, the first step in the NAAQA model is feature extraction (orange box in Fig. 2). The most obvious adaptation of Visual FiLM to acoustic data is to retrain the feature extraction module on the scenes from CLEAR2. To do this, we used three 2D convolutional layers, with 3×3 kernels, stride 2×2 and N_1, N_2, and N_3 filters respectively followed by a 1×1 convolutions with N_4 filters. We refer to this model as **`NAAQA 2D Conv`**.

However, as acoustic signals present unique properties, we introduce two feature extraction modules that are specifically tailored to sounds: the *Parallel* feature extractor (Fig. 3a) processes time and frequency features independently in parallel pipelines; the *Interleaved* feature extractor (Fig. 3b) captures time and frequency features in a single convolutional pipeline. In both cases, the feature extractor is trained end-to-end with the rest of the network and uses a combination of 1D convolutional filters to process a 2D spectro-temporal representation. The 1D filters process the time and frequency axis independently as opposed to the 2D filters typically used in image processing.

5. https://github.com/NECOTIS/NAAQA-Acoustic-Question-Answering

The design of the *Parallel* feature extractor (Fig. 3a) is inspired by the work of Pons *et al.* [63] where 1D filters are used to capture time and frequency features separately. While Pons *et al.* time-frequency model includes only 1 time and 1 frequency convolution which are then concatenated together, our extractor stacks multiple 1D convolutions in two parallel pipelines. The time and frequency features are only fused at the end of both pipelines. This yields more complex features. The *frequency pipeline* (green in the figure) comprises a serie of K frequency blocks. Each block is composed of a 1D convolution with N_K 3×1 kernels and 2×1 strides followed by a 1×2 maxpooling. With a stride larger than 1×1, the convolution operation downsamples the frequency axis and the pooling operation downsamples the time axis. This downsampling strategy allows features in both parallel pipelines to be of the same dimensions. The *time pipeline* (yellow in the figure) is the same as the frequency pipeline except that the convolutional kernel operates along the time dimension and the pooling along the frequency dimension. The convolution kernel is 1×3 and the pooling kernel 2×1. The activation maps of both pipelines are concatenated channel-wise and a representation combining both the time and frequency features is created using a 1×1 convolution [73] with P filters and a stride of one. The feature maps dimensionality is either compressed or expanded depending on the number of filters P in the 1×1 convolution. We name the corresponding model as **`NAAQA Parallel`**. The 1×1 convolution can also be removed thus leaving it up to the next 3×3 convolution to fuse the time and frequency features.

The *Interleaved* feature extractor (Fig. 3b) processes the input spectrogram in a single pipeline composed of a serie of K interleaved blocks (purple in the figure). Each block

comprises a 1×3 *time* convolution with N_K filters and stride 1×2 followed by a 3×1 *frequency* convolution with N_K filters and stride 2×1. A 1×1 convolution with P filters processes the output of the last block to either compress or expand its dimensionality. We name the corresponding model as **`NAAQA Interleaved Time`**.

As an alternative configuration, the order of the convolution operation in each block can be reversed so that it first operates along the frequency axis and then the time axis. The model is called **`NAAQA Interleaved Freq`**, in this case. Compared to the *Parallel* feature extractor, time-frequency representations are created after each block instead of only at the end of the pipeline.

For all extractors, the convolutions in the first block comprise N_1 convolutional filters and the number of filters is doubled after each block ($N_i = 2N_{i-1}$). More blocks (higher K) gives a larger downsampling of the feature maps which brings down the computational cost of the model.

4.2.2 Coordinate Maps for Acoustic Inputs

When tackling the VQA task, the Visual FiLM model concatenates coordinate maps (`CoordConv` [64]) to the input of convolutional layers (pink border boxes in Fig. 2). In the visual domain both axis of an image encode spatial information. Coordinate maps have, therefore, the same meaning in the x or y-axis.

In spectro-temporal representations for audio, however, the y-axis corresponds to frequency and the x-axis to time. We, therefore, call the maps *frequency* and *time* coordinate map, respectively. All spectro-temporal representations in CLEAR2 have the same range for the frequency axis but the range for the time axis varies depending on the duration of the acoustic scenes. We hypothesize that *time* coordinate maps might have a stronger impact on performance because they provide a relative time scale that the model can use to enhance its temporal localization capabilities.

4.2.3 Complexity Optimization

We performed optimization of the most important hyperparameters in the NAAQA architecture with the goal of reducing model complexity. These include number of *GRU* text-processing units G; the number of Resblock J that dictates the number of FiLM layers and, therefore, the number of modulation coefficients to compute; the number of convolutional filters M in each block; the number of filters C and the number of hidden units H in the classifier module. We refer to the resulting model by prepending **`Optimized`** to the model name.

4.2.4 NAAQA with a MALiMo Module

In MALiMo [29], Fayek and Johnson add a second set of FiLM layers that acts has an auxiliary controller. The controller uses the extracted acoustic features to further modulate the intermediate Resblocks. To evaluate the impact of MALiMo on CLEAR2, a MALiMo module was added to NAAQA. We refer to this configuration by appending **`MALiMo ctrl`** to the names introduced above. In our implementation of the module we replaced LSTMs with GRUs and adapted the inputs to the acoustic features that we study.

5 Experiments

We perform experiments to compare the effect on performance of our modification to the baseline model. Most experiments are conducted on the proposed CLEAR2 dataset. We first investigate different feature extraction methods and compare them to the **`Visual FiLM Resnet101`** feature extractor. Then, we show the effect of time and frequency coordinate maps at different levels of the model. Moreover, we perform an hyper-parameters ablation study to reduce the complexity of the model. We finally test the addition of a MALiMo module to our model. To demonstrate the generality of the results, we compare the performance of our model on CLEAR2 and DAQA′ datasets.

5.1 Acoustic Pre-Processing

The raw acoustic signal (sampled at 48 kHz for CLEAR2 and 16kHz for DAQA′) is processed to create a 2D time-frequency representation with Mel scale [74] spectrograms. After preliminary tests it was decided to extract 64 Mel coefficients for both CLEAR2 and DAQA′ computed over samples weighted by a Hanning window. The window size was of 512 samples (∼10.6 msec) for CLEAR2 whereas for DAQA′ it was of 400 samples (∼25ms) as in [29]. The time shift between consecutive windows (stride) was also optimized depending on the characteristics of audio data. We found that the best results for CLEAR2 was a time shift of 2048 samples (∼42.7 msec). This is feasible because of the sustained notes which vary slowly in time. Using such a long time shift allowed us to reduce more than ten folds the computational costs. As DAQA′ contains sounds that are shorter and less stable, the same optimization is not feasible. In fact, with a time shift of 1600 samples (100ms) a 5% drop in accuracy is observed in comparison with 160 samples (10ms) shifts. All results based on CLEAR2 are reported with long window shifts (long stride), with the exception of the comparison between short and long strides on both CLEAR2 and DAQA′ in Supplementary Materials.

As duration of scenes are not constant in CLEAR2, spectrograms are zero padded along the time axis so that they all have the same dimension ($1 \times 64 \times 418$) which corresponds to a maximum length of ∼17.9 sec. The power spectrum is normalized to the mean and standard deviation of the training data with the goal of speeding up convergence [75].

5.2 Experimental Conditions

Unless specified otherwise, the models presented in subsequent sections are trained on the CLEAR2 dataset which comprises 50 000 scenes and 4 questions per scene for a total of 200 000 records from which 140 000 (70%) are used for training, 30 000 (15%) for validation and 30 000 (15%) for test. The test set is generated using a different set of elementary sounds which ensures that the network could not memorize them and can therefore acts has a better generalization benchmark. The optimization techniques and other training settings are further described in Supplementary Materials.

TABLE 3
Results on CLEAR2

Configuration	Number of Parameters	Overall Acc.	Accuracy by question type (%)										
			Instrument	Note	Brightness	Loudness	Exist	Abs. Pos.	Glob. Pos.	Rel. Pos.	Count	Count Comp.	Count Inst.
Baselines													
Random Answer	—	1.75	1.75	1.75	1.75	1.75	1.75	1.75	1.75	1.75	1.75	1.75	1.75
Most common Answer (Yes)	—	7.3	0	0	0	0	55.62	0	0	0	0	47.27	0
Visual FiLM Resnet101	6.71 M	62.3 ±0.78	61.9 ±0.85	37.8 ±1.30	83.9 ±0.58	81.9 ±0.61	73.2 ±0.70	51.7 ±1.40	73.1 ±2.29	48.4 ±1.82	41.5 ±0.42	59.8 ±0.57	39.9 ±3.03
NAAQA													
NAAQA 2D Conv	5.61 M	77.6 ±0.72	80.6 ±0.66	73.4 ±1.28	90.5 ±1.44	86.5 ±0.76	81.3 ±0.57	74.7 ±2.50	87.5 ±0.42	54.0 ±1.43	53.8 ±1.01	60.8 ±1.34	49.0 ±3.53
NAAQA Interleaved Freq	5.61 M	67.2 ±0.98	63.0 ±1.93	48.0 ±2.14	84.5 ±1.07	81.1 ±0.52	72.1 ±1.24	65.1 ±1.04	80.1 ±0.66	49.3 ±1.59	42.5 ±1.51	56.6 ±2.45	46.5 ±1.57
NAAQA Interleaved Time	5.61 M	78.0 ±0.51	**81.8 ±1.06**	70.6 ±1.24	90.9 ±0.78	**87.9 ±0.94**	**81.6 ±0.66**	75.9 ±1.08	87.4 ±0.25	**60.0 ±4.07**	53.6 ±0.63	**61.2 ±0.73**	50.0 ±2.23
NAAQA Parallel	5.61 M	78.5 ±0.45	80.4 ±0.83	72.6 ±1.19	91.2 ±0.74	87.1 ±0.28	80.4 ±0.16	78.7 ±1.26	88.8 ±0.53	55.4 ±1.63	52.6 ±0.43	59.8 ±2.13	50.2 ±3.49
Optimized NAAQA Parallel	1.68 M	**79.5 ±0.05**	81.7 ±0.20	**74.2 ±0.58**	**91.9 ±0.17**	87.4 ±0.31	81.2 ±0.10	**79.3 ±0.77**	**90.0 ±0.15**	58.0 ±1.32	**53.8 ±1.40**	60.6 ±0.89	**50.4 ±1.10**
NAAQA + MALiMo													
Visual FiLM Resnet101 + MALiMo ctrl	**6.82 M**	63.6 ±1.20	61.8 ±1.40	36.3 ±1.41	83.5 ±0.86	82.3 ±0.59	73.4 ±1.06	57.1 ±3.58	76.1 ±1.94	47.4 ±2.98	41.6 ±0.80	56.7 ±4.23	43.2 ±3.64
NAAQA 2D Conv + MALiMo ctrl	6.71 M	77.1 ±1.12	79.2 ±1.01	71.3 ±1.54	90.5 ±0.81	86.1 ±0.73	80.3 ±0.77	75.7 ±2.08	87.8 ±0.74	52.7 ±1.94	51.9 ±2.73	59.0 ±3.00	50.2 ±2.48
NAAQA Interleaved Freq + MALiMo ctrl	6.71 M	64.8 ±3.91	55.3 ±8.33	43.1 ±8.77	83.2 ±1.33	80.0 ±1.75	70.3 ±2.74	63.6 ±4.72	78.7 ±3.27	45.8 ±3.53	41.4 ±1.96	55.1 ±2.75	48.7 ±2.05
NAAQA Interleaved Time + MALiMo ctrl	6.71 M	77.1 ±0.63	79.9 ±0.86	70.8 ±1.10	90.5 ±1.00	86.7 ±0.89	80.3 ±0.34	74.5 ±1.06	87.6 ±0.35	55.2 ±2.39	53.8 ±0.43	59.9 ±2.32	50.0 ±0.93
NAAQA Parallel + MALiMo ctrl	6.71 M	77.3 ±0.93	78.2 ±1.50	71.3 ±1.00	89.9 ±0.66	85.8 ±0.67	79.7 ±0.45	77.7 ±2.02	87.6 ±0.53	53.6 ±1.84	51.9 ±1.82	59.0 ±2.29	48.1 ±1.65
Optimized NAAQA Parallel + MALiMo ctrl	2.78 M	78.2 ±0.06	80.8 ±0.41	72.4 ±0.33	89.6 ±0.62	86.2 ±0.04	79.7 ±0.10	79.3 ±0.22	88.4 ±0.36	54.0 ±0.33	52.2 ±0.70	58.8 ±0.45	45.6 ±0.00

Table gives the number of parameters, average accuracy (%), and standard deviation over five repetitions of the training. Overall accuracy as well as question-kind dependent accuracy are reported. Different configurations are reported in the same order as they are discussed in the paper. The most common answer is "Yes".

Results are reported in terms of accuracy, that is in percentage of correct answers over the total. Since initialization of deep architectures has a profound impact on training convergence, we developed a python library `torch-reproducible-block`[6] to control the model initial conditions and design reproducible experiments. To ensure the robustness of the results, each model is trained 5 times with 5 different random seeds.

5.3 Initial Model Configuration

The initial configuration for the proposed model comprises $G = 4096$ GRU units, $J = 4$ Resblocks with $M = 128$ filters each and a classifier composed of a 1×1 convolution with $C = 512$ filters and $H = 1024$ hidden units in the fully connected layer. This configuration includes both *time* and *frequency* coordinate maps in each location highlighted in pink in Fig. 2.

6 Results and Discussion

Main results on the CLEAR2 data set are presented in Table 3. The complexity of the models in terms of number of parameters, the overall accuracy and accuracy dependent on question's type are reported. Results from two theoretical baselines - random chance and majority class answers - are first given. Then we report results from the **`Visual FiLM Resnet101`** baseline model with the initial configuration described in Section 5.3. This architecture achieves the lowest accuracy of 62.3% in comparison with all tested models. As expected, the pre-learned knowledge gathered in a visual context does not transfer directly to the acoustic context. Mel spectrograms have a very different structure than visual scenes features.

6.1 NAAQA Modifications

Unless specified otherwise, the initial configuration described in section 5.3 is used for all models in this section.

6.1.1 Feature Extraction

The first improvement to the baseline is given by introducing a specific audio feature extraction module based on 2D convolutions. The **`NAAQA 2D Conv`** model has slightly fewer parameters than **`Visual FiLM Resnet101`** because of the simpler feature extraction module and a much higher overall accuracy of 77.6%.

We then tested two versions of the *Interleaved* feature extractor (Fig. 3b). The computation order of the 1D convolutions in each block has a significant impact on performance. When the first 1D convolution in each block is computed along the frequency axis (**`NAAQA Interleaved Freq`**), the network reaches an overall accuracy of 67.2%. It performs especially poorly with questions related to *count* (42.5%), *count instruments* (46.5%) and *notes* (48.0%). The performance on *position* questions is also the lowest among all extractors. When the computation order of the convolution is reversed (**`NAAQA Interleaved Time`**), information is better captured and the network reaches 78.0% of overall accuracy. A possible explanation relates to the nature of the sounds in the CLEAR2 dataset which mainly consists of sustained musical notes. The time dimension at short scales does not contain much information that helps identifying the individual sounds. At larger scales, however, the time axis contains information relative to the scene as a whole which is exploited by higher level layers (Resblocks) to take into account the connections between different sounds. Because its stride is greater than 1x1, each 1D convolution downsamples the axis on which it is applied. When the first is a frequency convolution, the frequency axis of the resulting features is downsampled which reduces the information that can be captured by the time convolution that follows.

The *Parallel* feature extractor (**`NAAQA Parallel`**, Fig. 3a) reaches an overall accuracy of 78.5%. It performs well on all question's types except *relative position*, *count* and *count instrument*. Refer to Section 6.2 for further analysis. These results show that building complex time and frequency feature separately and fusing them at a later stage is a good strategy to learn acoustic features for this task. This claim is further strengthen by the analysis of Section 6.3.2.

Out of all extractors, **`NAAQA Parallel`** is the one that performs the best and constitutes the basis of NAAQA in all subsequent experiments.

6.1.2 Coordinate Maps

Coordinate maps can be inserted before any convolution operation (Fig. 2). We therefore analyzed the impact of the

6. https://github.com/NECOTIS/torch-reproducible-block

placement of *Time* and *Frequency* coordinate maps at different depths in the network. All possible locations were evaluated via grid-search. For each location, we inserted either a *Time* coordinate map, a *Frequency* coordinate map or both. Results are detailed in Table 5. *Time* coordinate maps have the biggest impact on performance, especially when inserted in the first convolution after the feature extractor or in the Resblocks. This could be because the fusion of the textual and acoustic features, and therefore most of the reasoning, is performed in the Resblocks. The network might be using the additional localization information to inform the modulation of the convolutional feature maps based on the context of the question. Surprisingly, the *Frequency* coordinate maps have a minimal impact on performance. We further compare the impact of *Time* versus *Frequency* coordinate maps in Supplementary Materials.

6.1.3 Complexity Optimization

As described in Section 4.2.3, we optimized the most important hyper-parameters (G, J, M) in the NAAQA model to reduce its complexity. The baseline **`Visual FiLM Resnet101`** configuration comprises 6.71 M parameters and achieves only 62.3%. **`NAAQA Parallel`** comprises 5.61 M parameters and performs significantly better with 78.5%. With this model as a starting point, we performed an ablation study to find which hyper-parameters can be reduced without impacting accuracy. The **`Optimized NAAQA Parallel`** configuration is the best trade-off between model complexity and performance. It comprises 1.68 M parameters and achieves the best overall accuracy with 79.5%. The most notable complexity reduction comes from the reduction of the number of GRU units G. Reducing G from 4096 to 512 increased accuracy while reducing the number of parameters by a factor of 3 (6.61 M versus 1.68 M). The **`Optimized NAAQA Parallel`** is composed of a *Parallel* extractor with $K = 3$ blocks and $P = 64$, $G = 512$ GRU units, $J = 4$ Resblocks with $M = 128$ filters, a classifier module with $C = 512$ filters and $H = 1024$ units. Results for this configuration can be found in Table 3 and Fig. 4. Further results related to the ablation study can be found in Supplementary Materials.

6.1.4 Adding a MALiMo Controller

The bottom rows of Table 3 show results where the configurations described in previous sections are augmented with a MALiMo controller. Although the model complexity is significantly increased (∼1M parameters), this addition does not bring any improvement in the model performance on CLEAR2. Almost all the tested configurations with a MALiMo controller perform slightly worse than the same configuration without the module, as can be seen in Table 3. This may be again explained by the characteristics of the sounds in CLEAR2. A more in-depth discussion is given when we evaluate the models on DAQA′.

6.2 Summary of Results on CLEAR2

NAAQA performs well on the CLEAR2 AQA task with 79.5% overall accuracy. It does however struggle with certain types of question as shown in Table 3 and Fig. 4.

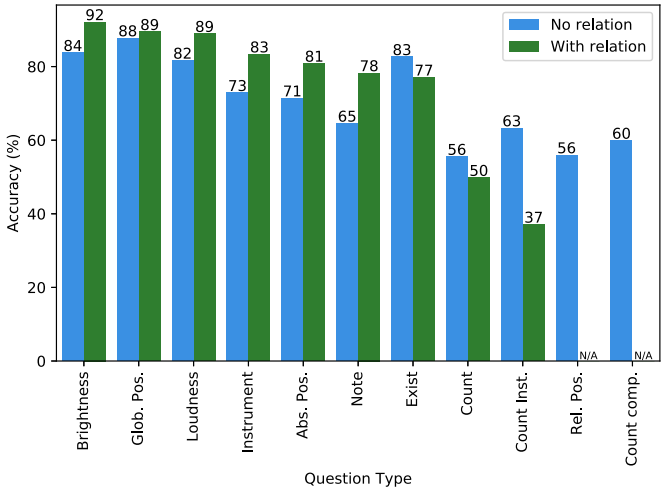


Fig. 4. *Test accuracy by question type and the number of relation in the question for* **`Optimized NAAQA Parallel`**. The overall accuracy for this configuration is 79.1%. The presence of *before* or *after* in a question constitutes a temporal relation. The accuracy is *N/A* for *relative position* and *count compare* since these types of question do no include relations. The hyper-parameters are described in the end of Section 6.1.3.

When asked to *count* the number of sounds with specific attributes, NAAQA reaches only 53.8% accuracy. This limitation is more severe if the question is to count the *different instruments* playing in a given part of the scene (50.4%). It attains slightly higher accuracy when asked to compare the number of instances of acoustic objects (more, fewer or equal number) with specific attributes (60.6%). In contrast, the network can successfully recognize individual instruments in the scene (81.7%). This suggests, that the problem lies in the logical complexity of the question rather than in the pattern matching from the acoustic scene. As an example, the question (count instrument): "How many different instruments are playing after the third cello playing a C# note?" requires to first identify the cello playing the C# note, then identify all acoustic objects that are playing after this sound, determine which instruments are of the same family and finally count the number of different families. The model struggles when it must focus on a large number of acoustic objects which explains the low accuracy for this type of question.

A similar argument could explain why models also have difficulties with questions related to the *relative position* of the instruments (58.0%). For example, to answer the question "Among the flute sounds, which one plays an F note?", the model must find all flutes playing in the scene, determine which one plays an F note, counts the number of flute playing before and translates the count to a position.[7] This also requires the network to focus on multiple objects.

Certain questions include temporal relations between sounds (*before* and *after*) as exemplified in Table 1. Questions that include relations require focusing on several sounds to be answered. Fig. 4 shows the accuracy for each question type depending on the presence of

7. This is one possible strategy to answer the question. There may be other ways.

TABLE 4
Results on DAQA′ the Table Presents Number of Parameters, Average Training, Validation and Test Accuracy (%) With Standard Deviation Over Five Repetitions of the Training as Well as Average Training Time

Configuration	# Parameters	Train Acc.	Val Acc.	Test Acc.	Trainig time
`Optimized NAAQA 2D Conv`	1.68 M	65.2 ±0.64	58.0 ±0.82	58.3 ±0.98	0 days 06:48:12
`Optimized NAAQA Parallel`	1.68 M	66.6 ±0.51	60.4 ±0.08	60.4 ±0.21	0 days 06:49:52
`Optimized NAAQA 2D Conv` + **`MALiMo ctrl`**	**2.78 M**	58.5 ±3.03	54.2 ±2.42	54.4 ±2.35	0 days 07:31:37
`Optimized NAAQA Parallel` + **`MALiMo ctrl`**	**2.78 M**	**67.3 ±1.22**	**64.1 ±0.54**	**64.3 ±0.72**	**1 days 04:59:46**

Results are reported for four configurations, with and without the MALiMo module in the same order as they are presented in the paper.

TABLE 5
Impact of the Placement of *Time* and *Frequency Coordinate Maps*

Coordinate maps				Accuracy (%)		
Extractor	1st Conv	Resblocks	Classifier	Train	val	Test
–	–	Time	–	95.0 ±0.79	**90.2 ±0.62**	**79.0 ±0.44**
–	Time	–	–	**95.1** ±0.90	90.0 ±0.28	**79.0 ±0.43**
–	–	Both	–	94.9 ±1.08	90.1 ±0.83	78.8 ±0.52
–	Both	–	–	95.1 ±1.23	90.0 ±0.46	78.7 ±0.70
Time	–	–	–	94.7 ±1.06	88.3 ±1.24	76.7 ±0.83
Both	–	–	–	94.1 ±1.47	88.4 ±1.06	76.6 ±0.36
–	–	–	Freq	84.0 ±2.17	73.5 ±1.40	64.8 ±1.51
–	–	–	–	85.2 ±0.71	72.9 ±1.36	63.8 ±0.70
–	–	–	Both	83.5 ±3.46	72.3 ±2.71	63.5 ±1.93
–	–	Freq	–	84.5 ±0.67	71.7 ±2.85	62.6 ±2.36
–	Freq	–	–	83.4 ±1.53	70.7 ±1.74	61.7 ±1.16
–	–	–	Time	81.9 ±0.88	70.1 ±1.99	61.7 ±0.76
Freq	–	–	–	79.7 ±3.70	67.1 ±3.36	59.3 ±2.20

All possible positions are illustrated by the pink border boxes in Fig. 2. The value Both *indicate that both* Time *and* Frequency *coordinate maps were inserted at the given position. The* **`NAAQA Parallel`** *is used with hyper-parameters from the initial configuration (defined in section 5.2. The rows are ordered by test accuracy).*

temporal relations. Questions that require the network to focus on a single acoustic object (*brightness*, *loudness*, *instrument*, *note*, *global position* and *absolute position*) benefit from the presence of a relation in the question. This might be explained by the fact that the question contains more information about the scene which helps to focus on the right acoustic object. However, the presence of relations in questions that already require the network to focus on multiple objects (*exist*, *count* and *count comparison*) is detrimental. This again supports the idea that having to focus on too many objects in the scene hinders the network performance.

6.3 Evaluation on DAQA′

To compare our results to those of Fayek and Johnson [29], we evaluated our models on a version of the DAQA data set. As mentioned in Section 3.2, we were not able to reproduce the original DAQA dataset which means that results presented in this section are not fully comparable with [29]. Results for different configurations of NAAQA tested on our modified DAQA′ are reported in Table 4.

6.3.1 NAAQA on DAQA′

The models explored in this section matches the performance of previous efforts [29]. The smallest model they evaluated had 5.49M parameters, the biggest model had 21.33M parameters and the best performing model had 13.20M parameters. The **`Optimized NAAQA 2D Conv`** model only has 1.68 M parameters and reaches an accuracy of 58.3% on DAQA′. The **`Optimized NAAQA Parallel`** has the same number of parameters and performs slightly better with and accuracy of 60.4%. When we analyzed both of these models on CLEAR2 dataset in Section 6.1.1, we found a much smaller difference between the performance of the **`NAAQA 2D Conv`** and the **`NAAQA Parallel`**. This difference suggests that the parallel extractor is more effective in the context of complex acoustic sounds (DAQA′) than with sustained musical notes (CLEAR2).

Even though these results are not 100% comparable with [29] because of the difference in the dataset composition, we want to emphasize that **`Optimized NAAQA Parallel`** reaches a somewhat similar accuracy than the smallest FiLM in [29] (60.4% versus 64.3%) with significantly smaller number of parameters (1.68 M versus 5.49 M).

6.3.2 NAAQA with a MALiMo Module on DAQA′

In Section 6.1.4, we found that adding a MALiMo controller to our NAAQA models did not improve the accuracy on CLEAR2. On the other hand, the MALiMo controller has a significant positive impact when the model is evaluated on DAQA′ dataset (Table 4). We see an increase of almost 4% when using **`Optimized NAAQA Parallel`** + **`MALiMo ctrl`** compared to **`Optimized NAAQA Parallel`** alone. These results are consistent

with Fayek and Johnson findings and with the hypothesis that MALiMo increases performance when working with complex sounds.

The **`Optimized NAAQA Parallel + MALiMo ctrl`** configuration performs about the same as the smallest MALiMo model evaluated in [29] (64.3% versus 65.1%) with significantly fewer parameters (2.78 M versus 8.91 M).

7 Conclusion

Acoustic Question Answering (AQA) is a newly emerging task in the area of machine learning research. As performance is strongly dependent on the acoustical environments and types of questions, it is important to understand the relationship between the application and the chosen neural architecture. We propose a benchmark for AQA based on musical sounds (CLEAR2) and a neural architecture that is tailored to interpreting acoustic scenes (NAAQA). NAAQA introduces a number of modifications to a FiLM based architecture to optimize acoustic scenes analysis. These includes several strategies for neural feature extraction, an ablation study of the hyper-parameters and the optimization of coordinate maps. We confirm that FiLM layers are very effective to modulate activation maps in the AQA application. We are able to optimize our NAAQA neural network so to obtain competitive results with a fraction of the model complexity. These results are confirmed on a different AQA task (DAQA′) comprising more complex sounds with the addition of a MALiMo controller in the model. We release all code openly in the hope that these resources may foster increased research activity in solving the AQA task.

Acknowledgments

The authors would like to thank the reviewers for their constructive comments that helped us improve the paper. The NVIDIA Corporation for the donation of GPUs.

References

[1] E. Voorhees, "The TREC-8 question answering track report," in *Proc. Text Retrieval Conf.*, 1999, pp. 77–82.
[2] E. M. Voorhees and D. M. Tice, "Building a question answering test collection," in *Proc. 23rd Annu. Int. ACM SIGIR Conf. Res. Develop. Informat. Retrieval*, 2000, pp. 200–207.
[3] M. M. Soubbotin and S. M. Soubbotin, "Patterns of potential answer expressions as clues to the right answers," in *Proc. Text Retrieval Conf.*, 2001, pp. 293–302.
[4] E. H. Hovy, L. Gerber, U. Hermjakob, M. Junk, and C.-Y. Lin, "Question answering in webclopedia," in *Proc. Text Retrieval Conf.*, 2000, pp. 53–56.
[5] M. Iyyer, J. Boyd Graber, L. Claudino, R. Socher, and H. Daumé, III, "A neural network for factoid question answering over paragraphs," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 633–644.
[6] D. Ravichandran and E. Hovy, "Learning surface text patterns for a question answering system," in *Proc. 40th Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 41–47.
[7] J. Johnson, B. Hariharan, L. van der Maaten, L. F. Fei, C. L. Zitnick, and R. Girshick, "CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1988–1997.
[8] S. Antol *et al.*, "VQA: Visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2425–2433.
[9] Y. Zhu, O. Groth, M. Bernstein, and L. F. Fei, "Visual7W: Grounded question answering in images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4995–5004.
[10] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu, "Are you talking to a machine? Dataset and methods for multilingual image question," in *Proc. 28th Int. Conf. Neural Informat. Process. Syst.*, 2015, pp. 2296–2304.
[11] A. Agrawal, D. Batra, and D. Parikh, "Analyzing the behavior of visual question answering models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1955–1960.
[12] P. Zhang, Y. Goyal, D. Summers Stay, D. Batra, and D. Parikh, "Yin and yang: Balancing and answering binary visual questions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5014–5022.
[13] D. Geman, S. Geman, N. Hallonquist, and L. Younes, "Visual turing test for computer vision systems," in *Proc. Nat. Acad. Sci. USA*, vol. 12, pp. 3618–3623, 2015.
[14] J. Cao, J. A. Robles Flores, D. Roussinov, and J. F. Nunamaker, "Automated question answering from lecture videos: NLP versus. pattern matching," in *Proc. Annu. Hawaii Int. Conf. System Sci.*, 2005, Art. no. 43b.
[15] T.-S. Chua, "Question answering on large news video archive," in *Proc. Int. Symp. Image Signal Process. Anal.*, 2003, pp. 289–294.
[16] H. Yang, L. Chaisorn, Y. Zhao, S.-Y. Neo, and T.-S. Chua, "VideoQA: Question answering on news video," in *Proc. 11th ACM Int. Conf. Multimedia*, 2003, pp. 632–641.
[17] K.-M. Kim, M.-O. Heo, S.-H. Choi, and B.-T. Zhang, "DeepStory: Video story QA by deep embedded memory networks," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, 2017, pp. 2016–2022, doi: 10.24963/ijcai.2017/280.
[18] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler, "MovieQA: Understanding stories in movies through question-answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4631–4640.
[19] Y.-C. Wu and J.-C. Yang, "A robust passage retrieval algorithm for video question answering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 10, pp. 1411–1421, Oct. 2008.
[20] T. Zhang, D. Dai, T. Tuytelaars, M.-F. Moens, and L. Van Gool, "Speech-based visual question answering," 2017, *arXiv:1705.00464*.
[21] S. Chang and E. Jungert, *Symbolic Projection for Image Information Retrieval and Spatial Reasoning*. Amsterdam, The Netherlands: Elsevier, 1996, doi: 10.1016/b978–0-12-168030-5.x5000-1.
[22] A. Moktefi and S.-J. Shin, *Vis. Reasoning With Diagrams*. Berlin, Germany: Springer, 2013, doi: 10.1007/978–3-0348-0600-8.
[23] M. Champagne, "Sound reasoning (literally): Prospects and challenges of current acoustic logics," *Logica Universalis*, vol. 9, pp. 331–343, 2015.
[24] A. S. Koepke, A.-M. Oncescu, J. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries: A benchmark study," *IEEE Trans. Multimedia*, early access, Feb. 8, 2022, doi: 10.1109/TMM.2022.3149712.
[25] M. Champagne, "Teaching argument diagrams to a student who is blind," in *Proc. Int. Conf. Theory Appl. Diagrams*, 2018, pp. 783–786.
[26] A. Pieropan, G. Salvi, K. Pauwels, and H. Kjellström, "Audio-visual classification and detection of human manipulation actions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2014, pp. 3045–3052.
[27] J. Abdelnour, G. Salvi, and J. Rouat, "CLEAR: A dataset for compositional language and elementary acoustic reasoning," 2018, *arXiv:1811.10561*.
[28] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3942–3951.
[29] H. M. Fayek and J. Johnson, "Temporal reasoning via audio question answering," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 2283–2294, 2020.
[30] C. Zhang, C. Öztireli, S. Mandt, and G. Salvi, "Active mini-batch sampling using repulsive point processes," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 5741–5748.
[31] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.
[32] D. Hudson and C. Manning, "GQA: A new dataset for real-world visual reasoning and compositional question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6700–6709.
[33] P. Wang, Q. Wu, C. Shen, A. Dick, and A. Van Den Hengel, "FVQA: Fact-based visual question answering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol 40, no. 10, pp. 2413–2427, Oct. 2018.

[34] K. Marino, M. Rastegari, A. Farhadi, and R. Mottaghi, "OK-VQA: A visual question answering benchmark requiring external knowledge," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3190–3199.
[35] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi, "From recognition to cognition: Visual commonsense reasoning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 6713–6724.
[36] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi, "IQA: Visual question answering in interactive environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4089–4098.
[37] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6325–6334.
[38] V. Manjunatha, N. Saini, and L. Davis, "Explicit bias discovery in visual question answering models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9562–9571.
[39] A. Das, S. Anjum, and D. Gurari, "Dataset bias: A case study for visual question answering," *Proc. Assoc. Informat. Sci. Technol.*, vol. 56, pp. 58–67, 2019.
[40] A. Agrawal, D. Batra, D. Parikh, and A. Kembhavi, "Don't just assume; Look and answer: Overcoming priors for visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4971–4980.
[41] D. Arad Hudson and C. D. Manning, "Compositional attention networks for machine Reasoning," in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=S1Euwz-Rb
[42] R. Vedantam, K. Desai, S. Lee, M. Rohrbach, D. Batra, and D. Parikh, "Probabilistic neural symbolic models for interpretable visual question answering," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6428–6437.
[43] R. Hu, A. Rohrbach, T. Darrell, and K. Saenko, "Language-conditioned graph networks for relational reasoning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 10 293–10 302.
[44] R. Hu, J. Andreas, T. Darrell, and K. Saenko, "Explainable neural computation via stack neural module networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 55–71.
[45] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum, "Neural-symbolic VQA: Disentangling reasoning from vision and language understanding," in *Proc. 28th Int. Conf. Neural Informat. Process. Syst.*, 2018, pp. 1039–1050.
[46] S. Abdoli, P. Cardinal, and A. L. Koerich, "End-to-end environmental sound classification using a 1D convolutional neural network," *Expert Syst. Appl.*, pp. 252–263, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417419304403
[47] V. Boddapati, A. Petef, J. Rasmusson, and L. Lundberg, "Classifying environmental sounds using image recognition networks," *Procedia Comput. Sci.*, vol. 12, pp. 2048–2056, 2017.
[48] J. Lee, T. Kim, J. Park, and J. Nam, "Raw waveform-based audio classification using sample-level CNN architectures," 2017, *arXiv:1712.00866*.
[49] Y. Han and K. Lee, "Acoustic scene classification using convolutional neural network and multiple-width frequency-delta data augmentation," 2016, *arXiv:1607.02383*.
[50] J. Nam, K. Choi, J. Lee, S.-Y. Chou, and Y.-H. Yang, "Deep learning for audio-based music classification and tagging: Teaching computers to distinguish rock from bach," *IEEE Signal Process. Mag.*, vol. 36, no. 1, pp. 41–51, Jan. 2019.
[51] W. Zheng, Z. Mo, X. Xing, and G. Zhao, "CNNs-based acoustic scene classification using multi-spectrogram fusion and label expansions," 2018, *arXiv:1809.01543*.
[52] J. Pons, O. Slizovskaia, R. Gong, E. Gómez, and X. Serra, "Timbre analysis of music audio signals with convolutional neural networks," in *Proc. Eur. Signal Process. Conf.*, 2017, pp. 2744–2748.
[53] M. Brousmiche, J. Rouat, and S. Dupont, "SECL-UMons database for sound event classification and localization," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 756–760.
[54] S. H. Nawab and T. F. Quatieri, *Short-Time Fourier Transform*. Hoboken, NJ, USA: Prentice-Hall, 1987, pp. 289–337.
[55] B. Logan, "Mel frequency cepstral coefficients for music modeling," in *Proc. Int. Symp. Music Informat. Retrieval*, 2000.
[56] J. Brown and M. Puckette, "An efficient algorithm for the calculation of a constant Q transform," *J. Acoustical Soc. Amer.*, vol. 92, 1992, Art. no. 2698.
[57] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 28th Int. Conf. Neural Informat. Process. Syst.*, 2012, pp. 1097–1105.
[58] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.
[59] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[60] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
[61] S. Hershey *et al.*, "CNN architectures for large-scale audio classification," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2017, pp. 131–135.
[62] A. Kumar and B. Raj, "Deep CNN framework for audio event recognition using weakly labeled web data," 2017, *arXiv:1707.02530*.
[63] J. Pons, T. Lidy, and X. Serra, "Experimenting with musically motivated convolutional neural networks," in *Proc. 14th Int. Workshop Content-Based Multimedia Indexing*, 2016, pp. 1–6.
[64] R. Liu *et al.*, "An intriguing failing of convolutional neural networks and the coordconv solution," in *Proc. 28th Int. Conf. Neural Informat. Process. Syst.*, 2018, pp. 9605–9616.
[65] K. Koutini, H. Eghbal zadeh, and G. Widmer, "Receptive-field-regularized CNN variants for acoustic scene classification," in *Proc. Workshop Detection Classification Acoustic Scenes Events*, 2019, pp. 124–128.
[66] O. R. Picas *et al.*, "A real-time system for measuring sound goodness in instrumental sounds," in *Proc. AES Conv.*, 2015, p. 9350.
[67] A. Pearce, T. Brookes, and R. Mason, "Timbral models, audiocommons project, deliverable D5.7," 2018. [Online]. Available: http://www.audiocommons.org/materials/
[68] I. Telecommunication Union, "Algorithms to measure audio programme loudness and true-peak audio level (ITU-R BS.1770–4)," 2015. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/bs/R-REC-BS.1770–4-201510-I!!PDF-E.pdf
[69] J. F. Gemmeke *et al.*, "Audio set: An ontology and human-labeled dataset for audio events," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2017, pp. 776–780.
[70] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456. [Online]. Available: http://dl.acm.org/citation.cfm?id=3045118.3045167
[71] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, pp. 211–252, 2015.
[72] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2009, pp. 2146–2153.
[73] M. Lin, Q. Chen, and S. Yan, "Network in networks," in *Proc. Int. Conf. Learn. Representations*, 2014, *arxiv.org/abs/1312.4400*.
[74] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *J. Acoustical Soc. Amer.*, vol. 8, no. 3, pp. 185–190, 1937.
[75] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 9–48.

Jérôme Abdelnour received the degree in computer engineering from the Department of Electrical and Software engineering, University of Sherbrooke, and the master's degree in machine learning from the University of Sherbrooke. He is currently working on electrifying the powersport industry with Taiga Motors as a backend/devops specialist. His research interest include machine learning, acoustic processing, software engineering, and large-scale distributed-systems.

Jean Rouat (Senior Member, IEEE) received the PhD degree from the University de Sherbrooke, He is currently a full professor with the University de Sherbrooke where he founded the Computational Neuroscience and Intelligent Signal Processing Research group (NECOTIS). His translational research links neuroscience and engineering for the creation of new technologies and a better understanding of learning multimodal representations. Development of hardware low power consumption Neural Processing Units for a sustainable development, interactions with artists for multimedia and musical creations are examples of transfers that he leads based on the knowledge he gains from neuroscience. He is leading funded projects to develop sensory substitution and intelligent systems based on neuromorphic computing and implementations.

Giampiero Salvi (Member, IEEE) received the MSc degree in electronic engineering from Università la Sapienza, Rome, Italy, and the PhD degree in computer science from KTH. He is currently a professor with the Department of Electronic Systems, Norwegian University of Science and Technology (NTNU), Trondheim, Norway, and associate professor with the KTH Royal Institute of Technology, Department of Electrical Engineering and Computer Science, Stockholm, Sweden. He was a postdoctoral fellow with the Institute of Systems and Robotics, Lisbon, Portugal. He was a co-founder of the company SynFace AB, active between 2006 and 2016. His main interests include machine learning, speech technology, and cognitive systems.