(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
12 December 2024 (12.12.2024)

WIPO | PCT

(10) International Publication Number
WO 2024/254467 A2

(51) International Patent Classification:
Not classified

(21) International Application Number:
PCT/US2024/033033

(22) International Filing Date:
07 June 2024 (07.06.2024)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
63/507,360 09 June 2023 (09.06.2023) US
63/593,933 27 October 2023 (27.10.2023) US

(71) Applicant: **UNIVERSITY OF WASHINGTON** [US/US]; 1100 NE Campus Pkwy, Suite 200, Seattle, Washington 98105 (US).

(72) Inventors: **VELURI, Bandhav**; c/o University of Washington, 1100 NE Campus Pkwy, Suite 3600, Seattle, Washington 98105 (US). **ITANI, Malek**; c/o University of Washington, 1100 NE Campus Pkwy, Suite 200, Seattle, Washington 98105 (US). **GOLLAKOTA, Shyamnath**; c/o University of Washington, 1100 NE Campus Pkwy, Suite 200, Seattle, Washington 98105 (US).

(74) Agent: **SPAITH, Jennifer** et al.; Stoel Rives LLP, 600 University Street, Suite 3600, Seattle, Washington 98101 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST,

(54) Title: NOISE CANCELLATION AND TARGET SIGNAL EXTRACTION SYSTEMS AND METHODS

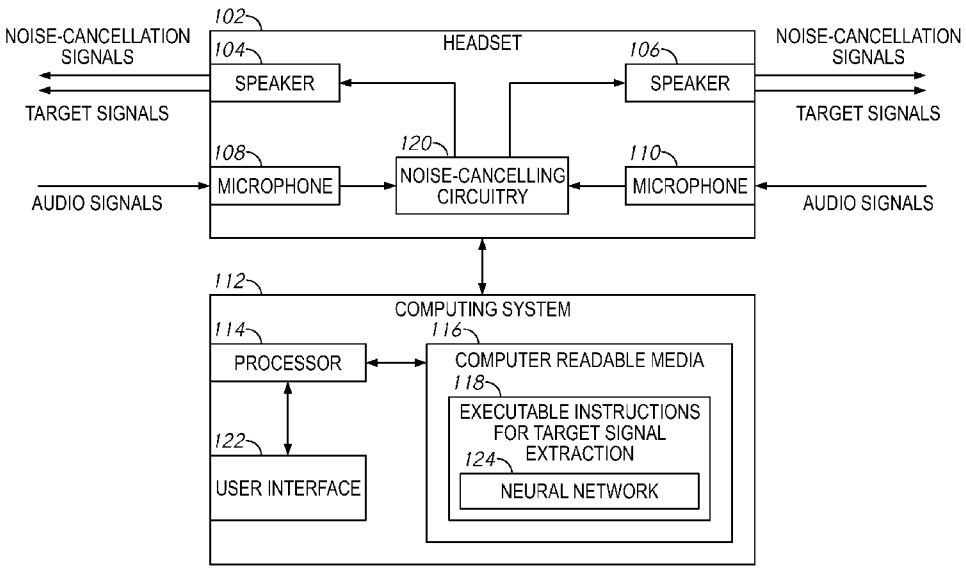

FIG. 1

(57) Abstract: Examples of systems and methods described herein may receive audio signals from at least one microphone. Example systems and methods may generate noise cancellation signals based at least in part on the audio signals and extract target signals based at least in part on the audio signals using digital neural network processing. Example systems and methods may provide the noise cancellation signals and the target signals from at least one speaker.

[Continued on next page]

WO 2024/254467 A2

SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *without international search report and to be republished upon receipt of that report (Rule 48.2(g))*

NOISE CANCELLATION AND TARGET SIGNAL EXTRACTION SYSTEMS AND METHODS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims the benefit under 35 U.S.C. § 119 of the earlier filing date of U.S. provisional applications 63/507,360, filed June 9, 2023 and 63/593,933 filed October 27, 2023, both of which provisional applications are hereby incorporated by reference in their entirety for any purpose.

TECHNICAL FIELD

[0002] Examples described herein relate generally to audio systems that provide noise cancellation and extraction of target signals from an environment.

BACKGROUND

[0003] Noise cancellation systems (e.g., earbud systems like AirPods Pro and AirPods Max) may achieve reasonable noise cancellation in practical scenarios. Using these systems, generally, all external sounds and noise may be cancelled.

[0004] Speech systems have predominantly focused on improving the performance of speech-related tasks for in-ear devices (e.g., Airpods), telephony (e.g., Microsoft Teams), and voice assistants (e.g., Google Home). Oftentimes, these systems collectively regard all non-speech sounds just as noise.

[0005] Acoustic transparency mode for in-ear devices tries to imitate the sound response of an open-ear system by transmitting the appropriate signals into the ear canal. Like active noise cancellation, this is agnostic to the sounds. Adaptive transparency on Apple Airpods is designed to automatically reduce the amplitude of loud sounds. However, this does not allow the user to pick and choose which sounds to hear.

[0006] Animals have evolved over millions of years to focus on target sounds and the associated directions. In- or over-ear devices such as headsets or earphones may make it difficult for users to hear target sounds.

SUMMARY

[0007] Methods are described herein. An example method includes receive audio signals from at least one microphone, generate noise cancellation signals based at least in part on the audio signals, extract target signals based at least in part on the audio signals using digital

neural network processing, and provide the noise cancellation signals and the target signals from at least one speaker.

[0008] Generating noise cancellation signals may include using signal processing. In some examples, the generation of noise cancellation signals is performed independent of extracting target signals.

[0009] Example methods may also include receive an indication of a sound class from a user. The digital neural network processing may be configured to extract the target signals based at least in part on the indication of the sound class.

[0010] Extracting the target signals may include extracting signals originating from one or more human speakers.

[0011] In some examples, the target signals are binaural signals, and the methods may include preserving a directionality of the binaural signals.

[0012] In example methods, a time to generate a portion of said noise cancellation signals based on a portion of the audio signals is a shorter time than a time to extract a portion of said target signals based on the portion of the audio signals.

[0013] In some examples, using digital neural network processing includes using an encoder-decoder architecture.

[0014] Examples of systems are described herein. An example system includes a plurality of microphones, the plurality of microphones configured to generate binaural audio signals from an environment. The system also includes at least one processor. The system also includes at least one non-transitory computer readable medium encoded with instructions which, when executed by the at least one processor cause the system to perform operations including utilize a trained neural network to extract target signals from the binaural audio signals based on a sound class for the target signals, where the target signals comprise binaural target signals, and a plurality of speakers, the plurality of speakers configured to playback the binaural target signals.

[0015] The operations may also include receive an indication of the sound class from a user.

[0016] The plurality of microphones and the plurality of speakers may be disposed in a headset. The plurality of microphones and the plurality of speakers may be disposed in a hearing aid.

[0017] Examples of systems may also include signal processing circuitry, the signal processing circuitry configured to generate noise cancelling signals based on the binaural audio signals.

[0018] In some examples, the sound class may include human speakers. In some examples, the sound class includes a class of sounds to eliminate from the binaural audio signals.

[0019] In some examples, the trained neural network includes an encoder-decoder architecture including an encoder configured to encode the binaural audio signals independent of the target signals to generate encoded data, and a decoder configured to condition the encoded data with an embedding to provide conditioned data and extract the target signals based on the conditioned data.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] To easily identify the discussion of any particular element or act, the most significant digit or digits in a reference number refer to the figure number in which that element is first introduced.

[0021] FIG. 1 is a schematic illustration of a system arranged in accordance with examples described herein.

[0022] FIG. 2 is a schematic timing diagram illustrating components contributing to end-to-end latency in binaural acoustic processing systems.

[0023] FIG. 3 is a schematic illustration of a framework for a neural network arranged in accordance with examples described herein.

[0024] FIG. 4 is a schematic illustration of an example encoder for use in neural networks described herein.

[0025] FIG. 5 is a schematic illustration of an example decoder for use in neural networks described herein.

[0026] FIG. 6 is a sheet of Equations referred to herein.

DETAILED DESCRIPTION

[0027] Examples described herein provide functionality for hearable devices, which allow users to program acoustic scenes in real-time and choose the sounds they want to hear from real-world environments, while also preserving the spatial cues in the target sounds. For example, users may listen to the birds chirping in a park without hearing the chatter from other hikers. In another example, users may block out traffic noise on a busy street while still being able to hear emergency sirens and car honks.

[0028] Examples described herein include examples of neural networks that can achieve binaural target sound extraction in the presence of interfering sounds and background noise.

Example training methodologies are described that allow systems to generalize to real-world use.

[0029] Semantic hearing systems and methods described herein generally utilize an understanding of the semantics of various natural and artificial sounds in real-time, in the presence of interfering sounds, and determine which sounds to allow and which to block, based on user input. Speech may be one amongst many other sound classes in systems described herein.

[0030] Examples described herein may advantageously address challenges relating to focusing on sounds using in- or over-ear devices (e.g., headsets). One set of challenges may be real-time use of such systems. Generally, the sound output should be generally synced with the user's visual senses. This may involve real-time processing that satisfies stringent latency requirements. Generally, a latency of less than 20-50 ms may be preferred in some examples. This may involve identifying the target sounds using 10 ms or less of audio blocks, separating them from interfering sounds, and then playing them back, all on a computationally-constrained device like a smartphone.

[0031] Examples described herein include semantic hearing systems and methods which may program a binaural acoustic scene based on semantic sound descriptions (e.g., sound classes). Examples of neural networks are described that may achieve binaural target sound separation and demonstrate that the network can run in real-time on smartphones. Examples of training methodologies are described to generalize a system to unseen real-world environments, and users. An example system is implemented using certain off-the-shelf hardware to show that an example system may achieve goals described herein in real-world environments.

[0032] In some examples, binaural processing is provided. Sounds arrive at the two ears with different delays and attenuations. The physical separation between the two ears and the reflections/diffraction from the wearer's head, e.g., the head-related transfer function, provide cues for spatial perception. To preserve these cues, a binaural output may be used to preserve or recover this spatial information for the target sounds across the two ears.

[0033] In some examples, neural network models described herein may have real-world generalization. Training and testing a neural network on synthetic data may be used. Examples of binaural target sound extraction networks may generalize to real-world hearable applications. The complexity of real-world reverberations and head-related transfer

functions (HRTFs) may be addressed in simulations. Generalization to in-the-wild use in unseen acoustic environments across different users may be used.

[0034] Some implemented results show that an example system can operate with 20 sound classes and that an example transformer-based network has a runtime of 6.56 ms on a connected smartphone. In-the-wild evaluation with participants in previously unseen indoor and outdoor scenarios shows that an example implemented system can extract the target sounds and generalize to preserve the spatial cues in its binaural output.

[0035] Certain details are set forth herein to provide an understanding of described embodiments of technology. However, other examples may be practiced without various of these particular details. In some instances, well-known circuits, control signals, timing protocols, computing system components, audio components, and/or software operations have not been shown in detail in order to avoid unnecessarily obscuring the described embodiments. Other embodiments may be utilized, and other changes may be made, without departing from the spirit or scope of the subject matter and/or claims presented here.

[0036] Examples of systems and methods described herein may receive audio signals from at least one microphone. Example systems and methods may generate noise cancellation signals based at least in part on the audio signals and extract target signals based at least in part on the audio signals using digital neural network processing. Example systems and methods may provide the noise cancellation signals and the target signals from at least one speaker.

[0037] FIG. 1 is a schematic illustration of a system arranged in accordance with examples described herein. The system of FIG. 1 includes headset 102 and computing system 112. The headset 102 may include speaker 104, speaker 106, microphone 108, microphone 110 and noise-cancelling circuitry 120. The noise-cancelling circuitry 120 may be coupled to speaker 104, speaker 106, microphone 108, and microphone 110. The headset 102 may be coupled to (e.g., in communication with) computing system 112. The computing system 112 may include processor 114, computer readable media 116, and user interface 122. The computer readable media 116 may include executable instructions for target signal extraction 118 including neural network 124.

[0038] The components shown in FIG. 1 are exemplary only. Additional, fewer, and/or different components may be used in other examples.

[0039] Examples of systems described herein may include one or more microphones and one or more speakers, such as speaker 104, speaker 106, microphone 108, and microphone 110 of

FIG. 1. The microphones and/or speakers may be provided in any of a variety of form factors. As shown in FIG. 1, the speakers and microphones may form all or part of a headset, such as headset 102. For example, microphones and/or speakers may be provided in one or more ear buds. For example, the speaker 104 and microphone 108 may be provided in an enclosure formed as an ear bud. The microphone speaker 106 and microphone 110 may be provided in an enclosure formed as an ear bud. In some examples, the microphones and/or speakers may be provided in one or more ear cups or other structures which may be shaped to sit on and/or over the ears of a user. In some examples, the microphones and/or speakers may be provided in a hearing aid which may sit in and/or proximate the ears of a user. While two microphones and two speakers are shown, any number may be used. Additional components may be provided in headset 102 which may be coupled to the components shown, such as speaker 104, speaker 106, microphone 108, microphone 110, and/or noise-cancelling circuitry 120. For example, drivers may be provided for one or more speakers. A headband may be provided to generally be located above a user's head and may hold ear cups in place. One or more interfaces for wired or wireless electrical connectivity may be provided including, but not limited to, one or more 3.5mm jacks, USB interfaces, Bluetooth, and/or WiFi interfaces.

[0040] In some examples, two microphones may be used to receive audio signals from an environment, such as microphone 108 and microphone 110. The two microphones may be referred to as binaural microphones. The microphones may be used to receive audio signals, which may also be referred to as acoustic signals. The microphones may receive audio signals which may be referred to as binaural signals. The speakers may provide (e.g., play back) audio signals which may be referred to as binaural signals. Binaural signals generally refer to signals associated with a same audio source received and/or transmitted at different locations. For example, binaural signals may refer to audio signals received by and/or associated with both ears of a listener. Binaural signals typically preserve a directionality of an audio signal. For example, receipt and/or playback of binaural signals (e.g., through speaker 104 and speaker 106) may preserve a direction of one or more sound sources.

[0041] Headsets described herein, such as the headset 102 of FIG. 1, may be noise-cancelling headsets. Noise-cancelling headsets may generally generate noise cancellation signals intended to cancel audio signals that were generated in an environment and received at the noise-cancelling headset. By playing back the noise cancellation signals, a user may generally experience a quiet and/or silent environment (e.g., because the user's ear receives both the signals generated in the environment and the noise-cancelling signals). The noise

cancellation signals may be based on the audio signals received at microphones described herein, such as microphone 108 and microphone 110.

[0042] Accordingly, examples described herein may utilize signal processing techniques to generate noise cancellation signals. For example, the noise cancellation signals may have amplitudes and/or frequencies calculated to cancel the received audio signals. Analog signal processing may be used to generate the noise cancellation signals. Analog signal processing refers to the processing of continuous-time signals. Analog signals have a value that may vary over time, and the value may be representative of a physical measurement (e.g., an audio frequency and/or amplitude). Analog signal processing generally uses analog components (e.g., circuitry including one or more resistors, capacitors, inductors, diodes, and/or transistors). Analog signal processing may accordingly occur generally real-time, which may be advantageous to reflect real-time changes in audio sources in the environment. Real-time refers to a latency with which it is possible to generate effective noise cancellation in an environment. In some examples, the analog signal processing may generate noise cancellation signals within less than a millisecond, on the order of microseconds in some examples. Other time scales may be used.

[0043] Examples of systems described herein may accordingly include noise-cancelling circuitry, such as noise-cancelling circuitry 120 of FIG. 1. The noise-cancelling circuitry 120 is coupled to microphone 108 and microphone 110. The noise-cancelling circuitry 120 may receive analog audio signals from an environment, which may be binaural signals. The noise-cancelling circuitry 120 may generate noise cancellation signals using signal processing techniques, such as analog signal processing. The noise-cancelling circuitry 120 may provide the noise cancellation signals to speakers, such as speaker 104 and speaker 106, which may play back the noise cancellation signals. Playback of the noise cancellation signals may result in a generally quiet and/or silent listening environment for a user as the noise cancellation signals may cancel audio signals received at the headset 102. The noise-cancelling circuitry 120 may be implemented using any number of analog components in some examples, including one or more resistors, capacitors, inductors, diodes, and/or transistors.

[0044] Examples of systems described herein may include one or more computing systems, such as computing system 112 of FIG. 1. The computing system 112 may be implemented using, for example, one or more computers, tablets, laptops, desktops, servers, smartphones, smartspeakers, cellular phones, wearable devices, appliances, and/or vehicles. The computing system 112 may be in communication with headset 102 in some examples. For example, the computing system 112 may communicate with the headset 102 using Bluetooth,

ZigBee, USB, 3.5mm wired connection, WiFi, and/or other communication interface. Generally, audio signals received at microphone 108 and/or microphone 110 may be communicated to the computing system 112. Target signals generated by the computing system 112 may be communicated to the headset 102 for playback by speaker 104 and/or speaker 106. In some examples, the computing system 112 may be wholly and/or partially integrated with the headset 102. For example, a noise-cancelling headset may be used that provides user access to the microphone data, such as the Sennheiser AMBEO Smart Headset. Examples of systems described herein may be implemented on such a device using fewer wires and may directly connect to the smartphone at a single point, without the need for an additional pair of binaural earphones.

[0045] Examples of computing systems described herein may include one or more processors and computer readable media, such as processor 114 and computer readable media 116 of FIG. 1. Processors described herein may generally be implemented using any processor circuitry, including one or more processors, one or more processor cores, field programmable gate arrays (FPGAs), central processing units (CPUs), graphical processing units (GPUs), application specific integrated circuits (ASICs), microcontrollers, and/or embedded processors. Computer readable media may generally be implemented using memory, random access memory (RAM), solid state drives (SSDs), read only memory (ROM), SD cards, and/or disk drives. While a single processor 114 and computer readable media 116 are shown in FIG. 1, any number may be used.

[0046] Software may be used to implement operations described herein. For example, the computer readable media 116 of FIG. 1 may be encoded with instructions which, when executed, cause the processor 114 to perform operations described herein. For example, the computer readable media 116 may include executable instructions for target signal extraction 118. Computer readable media 116 may store data used in operations described herein, such as indications of sound classes, sound sources, and/or embeddings described herein. While a computer readable media 116 is shown as including executable instructions for target signal extraction 118, it is to be understood that multiple computer readable media may be used.

[0047] Examples of software described herein may utilize neural network processing to extract target signals from audio signals received from an environment. Accordingly, the executable instructions for target signal extraction 118 may include neural network 124. The neural network 124 may be used to extract target signals from the audio signals, such as the audio signals received at microphone 110 and/or microphone 108. Accordingly, audio signals may be provided from headset 102 (e.g., from microphone 108 and/or microphone 110) to the

computing system 112. The audio signals may be converted into digital signals either before and/or after being provided to the computing system 112. Accordingly, in some examples, the headset 102 may convert the audio signals to digital signals. In some examples, the computing system 112 may convert the audio signals to digital signals. Digital processing, such as digital neural network processing, may be used to extract target signals from the audio signals. In some examples, the audio signals provided to the computing system 112 may be binaural audio signals. Similarly, the target signals extracted may be binaural target signals. The neural network 124 may be a neural network trained to extract target signals from the audio signals. The neural network 124 may be trained using supervised and/or unsupervised learning techniques or other learning techniques. The neural network 124 may be trained to extract a particular kind or type of target signals. For example, the neural network 124 may be trained to extract signals originating from a particular class of sound sources in an environment. In some examples, the neural network 124 may be trained to extract signals originating from one or more human speakers in an environment. In some examples, the neural network 124 may be trained to extract signals from audio signals such that sounds made by sources in an environment that are undesirable to hear are wholly and/or partially removed.

[0048] Examples of neural networks described herein may be capable of achieving binaural target sound extraction. An example network (e.g., an example of neural network 124) takes two audio signals from microphones at the two ears (e.g., microphone 108 and microphone 110 of FIG. 1) as binaural input and outputs two audio signals as binaural output, while preserving the directionality of the target sounds in the acoustic scene. To do this, the neural network 124 may start with a single-channel (e.g., not binaural) transformer model for target signal extraction. The network may be optimized for real-time operations on computing systems, such as smartphones. Then, the neural network 124 may include a network that jointly processes the binaural input signals, allowing the network to preserve the spatial information about the target sounds and output binaural audio. This joint processing may be more effective at binaural target sound extraction and may have less (e.g., half) the computational cost of processing the binaural input signals separately.

[0049] Examples of a training methodology are described that may allow a binaural network to generalize to real-world situations, such as reverberations, multipath, and HRTFs. Obtaining training data in fully natural environments can be difficult because mixtures may be captured but lack access to the ground truth sounds needed for supervised learning. Moreover, training a network that can generalize to in-the-wild use with hearables generally

involves training data that captures reverberations, multipath, and HRTFs across a large number of users. To achieve this, examples described herein synthesize training data using multiple datasets. In one example, first, an HRTF dataset is used, which includes measurements from users in non-reverberant environments. The room impulse responses may be convolved with thousands of examples from different audio classes (e.g., 20 different audio classes) to generate both mixtures and ground truth binaural audio. However, this may not capture the reverb and multipath in realistic environments. Therefore, these synthesized mixtures may be augmented with training data synthesized from three different datasets that provide binaural room impulse responses captured in real rooms. This facilitates example networks to generalize to users and real-world environments that are not in the training dataset.

[0050] In some examples, the neural network 124 may be implemented using an encoder-decoder architecture. The neural network 124 may include an encoder. The encoder may encode audio signals. The audio signals may be encoded in a manner independent of the target signals. For example, an indication of sound class, sound source, or other enrollment for target signal extraction may not be used to encode the audio signals by the encoder. The neural network 124 may include a decoder. The decoder may condition the encoded data with an embedding to provide conditioned data and extract the target signals based on the conditioned data. The embedding may be indicative of a sound class and/or sound source as described herein.

[0051] Accordingly, digital neural network processing may be used to extract target signals from audio signals described herein. The neural network processing may operate on digital signals. Utilizing digital signal processing and/or utilizing software executed by one or more processors may generally cause the process of extracting target signals to be slower than the process of generating noise cancellation signals described herein. Note that the generation of noise cancellation signals by systems described herein may be independent of the extraction of target signals from the audio signals. One process (e.g., signal processing) is used to receive the audio signals and generate noise cancellation signals. Another process (e.g., digital neural network processing) is independently used to receive the audio signals and extract target signals. For example, referring to FIG. 1, the noise-cancelling circuitry 120 may generate noise cancellation signals using signal processing techniques (e.g., analog signal processing). This process may be real-time (e.g., near real-time). For example, the generation of noise cancellation signals may occur within less than a millisecond in some examples. While the noise-cancelling circuitry 120 is generating noise cancellation signals

based on received audio signals, the headset 102 is also providing the audio signals to the computing system 112. The computing system 112 may process the audio signals using digital neural network processing to extract target signals. The digital neural network processing may take a longer time than the generation of the noise cancellation signals. In some examples, the digital neural network processing may take milliseconds of time.

[0052] The target signals may be extracted as digital signals. The target signals may be converted into audio signals at either computing system 112 and/or headset 102 for playback by speaker 104 and/or speaker 106.

[0053] During operation, speakers described herein may produce noise-cancelling signals and target signals. While each of speaker 104 and speaker 106 is shown in FIG. 1 as generating both noise-cancelling signals and target signals, in other examples one speaker may generate noise-cancelling signals and another speaker may generate target signals. Accordingly, a user listening to the output of speakers described herein, such as speaker 104 and speaker 106, may hear target signals clearly, with other portions of the audio signals suppressed. Note that the noise cancellation signals will be played by the speakers at an earlier time than the target signals, due to the longer time taken by the digital neural network processing to generate the target signals. Moreover, the communication between the headset 102 and computing system 112 may further delay the target signals. However, this delay is likely acceptable to users because the noise cancellation generated through signal processing is adequate to cancel incoming audio sounds, creating a cancelled sound environment into which the target signals can be played back, even if they are played back at a delay.

[0054] Recall that the audio signals provided by microphones described herein may be binaural audio signals. Accordingly, binaural audio signals may be used to generate noise cancellation signals. The noise cancellation signals may be binaural noise cancellation signals. The binaural audio signals may also be used to extract target signals using digital neural network processing. Accordingly, the target signals may be binaural target signals. In this manner, the target signals output from the speakers may preserve a directionality present in the binaural audio signals (e.g., a directionality of one or more sound sources).

[0055] Examples of systems described herein may include a user interface, such as user interface 122 of FIG. 1. While user interface 122 is shown in computing system 112, in some examples, some or all of user interface 122 may be implemented in headset 102. The user interface 122 may be implemented, for example, as a button, a touchscreen, a speaker for

receipt of audio input commands (which may be implemented using another speaker described herein), a display, a keyboard, and/or a mouse. Other input devices may also be used.

[0056] The user interface 122 may be used for a user to input information used by the executable instructions for target signal extraction 118 to extract target signals. For example, a sound class and/or sound source may be input by a user using the user interface 122. A sound class refers to sounds which may be specified by semantic description (e.g., a semantic name for a particular type or source of sound). Thus, a sound class may be in contrast to directional hearing, which may refer to the ability to hear sounds from a particular direction. Examples of sound classes include, but are not limited to, mechanical sounds, animal sounds, human speech sounds, vehicle sounds, etc. Examples of sound classes include particular audio sources. Examples of sound classes include emergency sirens, ocean sounds, human speech, alarm clock sounds, baby sounds, street noise, and/or birds chirping. The user may use the user interface 122 to provide an indication of a sound class. The indication of the sound class may be used (e.g., by processor 114 and computer readable media 116) to extract the target signals. For example, the neural network 124 may be trained to extract target signals belonging to a selected sound class.

[0057] In some examples, a sound class and/or sound source may be identified for suppression. For example, a user may use user interface 122 to provide an indication of a sound class to be suppressed. The executable instructions for target signal extraction 118 may accordingly extract target signals from the audio signals where the target signals represent the audio signals without the audio generated by the sound class. Accordingly, in some examples, the user interface 122 may display multiple sound classes. A user may select one or more sound classes to hear and one or more sound classes to suppress.

[0058] In other examples, automatic speech recognition may be used to provide an indication of sound classes. For example, a user could say, "Only allow sounds from birds." Speech-to-text and intent classification systems can be used to transcribe these speech commands and identify the target sound class as "bird chirps," and then use a neural network to perform the task. Finally, a described example herein utilized 20 sound classes in an implementation to push the system to its limits and demonstrate that a real-time binaural neural network has the capability to operate with a decent number of classes. However, it is likely that from a user interface perspective, one may want to limit the number of classes for a better user experience. Accordingly, a fewer number of classes may be used in other examples.

[0059] The indication of sound classes to hear and sound classes to suppress may be used by neural networks described herein to extract target signals for playback. Accordingly, examples of binaural target sound extraction can also be used to subtract the identified sounds and play the residual sounds (which would be the target signals in those examples) into the ear. For example, computer typing and/or hammer sounds may be removed from audio signals to focus on human speech. This can be beneficial when the user knows the specific type of environmental noise that they feel is annoying (e.g., computer typing in an office room) as this approach would remove only the specified noise and thus allow the user to focus on the speech and the other sounds in the environment.

[0060] During operation, users of systems described herein may provide an indication of one or more audio sources for target signal extraction. The indication of audio sources may be, for example, an indication of one or more sound classes and/or one or more human speakers in an environment. Processors described herein, such as processor 114, may enroll the audio sources and/or may facilitate enrollment of the audio source(s). Enrollment generally refers to the process of generating an enrollment signal (e.g., an embedding, also referred to as an embedding vector in some examples) that may be indicative of characteristics of the target audio source(s) and may subsequently be used to extract target signals from received audio signals.

[0061] In some examples, the processor 114 may perform the enrollment (e.g., the computer readable media 116 may include executable instructions for generating enrollment signals). In some examples, another computing system may perform the enrollment. For example, the processor 114 may provide audio signals received to another computing system (e.g., a cloud computing system). The processor 114 may transmit the audio signals through a wired or wireless connection. The other computing system may provide the enrollment signals back to the system of FIG. 1, and the processor 114 may store the enrollment signals in the computer readable media 116 and/or other computer readable media.

[0062] Audio signals that are received and and/or provided by microphones described herein may be used to provide both noise cancellation signals and target signals which may be extracted from the audio signals. In this manner, users of systems described herein may hear target audio sources (e.g., target human speakers and/or sound classes) with improved clarity, even in the presence of background noise, and even as the user and the target audio source(s) move freely around the environment.

[0063] In this manner, speakers described herein may provide noise-cancelling signals and target signals. The noise-cancelling signals generally may be intended to create a silent environment in the ear of a hearer, and then the target signals may be intended to provide the audio generated by particular audio source(s), or by sources other than particular audio source(s) in some examples. In this manner, users of systems described herein may hear desired audio sources with increased clarity. Because enrollment signals are used to extract the target signals, the extraction may be generally robust as the user and/or the target audio source(s) move through the environment in the presence of other interfering signals.

[0064] Examples of systems described herein, such as the example system of FIG. 1, may be implemented using a variety of form factors including one or more noise-cancelling headsets, smartphones, ear buds, headbands, hearing aids and/or other wearable devices.

[0065] Examples of systems described herein may be used in a variety of use cases. For example, systems described herein may be used to more clearly hear a speaker in a crowded or noisy area. Examples include waiting rooms, classrooms, performances, outdoor environments, and/or indoor environments having mechanical or other noise. Examples of systems described herein may be used to provide target speaker signals in hearing aids. Note that noise-cancelling may be of less benefit and may not be used in hearing aids. This may be due to the poor hearing of noise sounds by the user in any event.

[0066] In some examples, a user may wear ear-worn devices on a beach and input an indication to listen to the calming sounds of the ocean (e.g., a sound class of ocean sounds) while suppressing any human speech nearby (e.g., a sound class of human speech). In an example, while walking on a busy street, a user may provide an indication to listen only to a sound class of emergency sirens. In an example, while sleeping, a user may provide an indication to listen to a class of alarm clock sounds or baby sounds but to suppress noise from the street (e.g., a sound class of street noise). In an example, a user may be on a plane and input to hear human speech and announcements but to suppress the sound of a crying baby. Or while hiking, in some examples a user may provide an indication to listen to the birds chirping but not the chatter from other hikers. These and other potential use cases utilize noise-cancelling earphones described herein for cancelling most and/or all the sounds and then techniques described herein for introducing back the target sounds into the earphones. Accordingly, examples described herein may program the output acoustic scene in real-time at least in part by semantically associating the individual incoming sounds with user input to determine which sounds to allow in and which sounds to block.

[0067] Examples of systems described herein may program the acoustic environment with imperceptible latency such that the target sounds of interest are present but interfering sounds are suppressed. Given the stringent latency constraints, the computation may not be desirably performed in the cloud, but may operate in real-time using computationally-constrained devices like smartphones (e.g., on computing system 112 of FIG. 1). Further, the target signals extracted by the neural network should originate from the same spatial directions as the real-world target sounds. Thus, examples described herein may advantageously have: 1) real-time low-latency operation, and 2) binaural real-world generalization.

[0068] FIG. 2 is a schematic timing diagram illustrating components contributing to end-to-end latency in binaural acoustic processing systems. FIG. 2 depicts acoustic signals, such as the audio signals which may be received by microphone 108 and microphone 110 of FIG. 1. Blocks for processing time are shown. A resulting output of speakers is depicted, such as speaker 104 and speaker 106 of FIG. 1.

[0069] Received audio signals (e.g., acoustic) signals may be stored into two memory buffers of the binaural microphones, e.g. microphone 108 and microphone 110. The acoustic data from the two microphones in each block may then be fed into a neural network (e.g., neural network 124 of FIG. 1) that outputs a block-length worth of binaural target sound data. This binaural output may then be played back through the two speakers on the headset (e.g., speaker 104 and speaker 106).

[0070] To aid in ensuring that the audio played through the headset is synced with the user's visual senses, this end-to-end latency should preferably be less than 50 ms in some examples, or less than 20-50 ms in some examples. To achieve this, the buffer duration, the lookahead duration and the processing time may all be reduced. Note that 1) a small buffer duration of say 10 ms means that the target signal extraction technique has only a 10 ms block of data to not only understand the semantics of the acoustic scene but also to separate the target sound from other interfering sounds.

[0071] While the acoustic signals that arrived prior to the current block may be used, many target sounds (e.g., door knocks) are not continuous. Reducing the buffer size even further to say 2 ms may increase the number of system calls and load the operating system of a device used to buffer the sounds. While a large lookahead can provide more context for the neural network to extract the target sounds, meeting end-to-end latency requirements described herein generally reduces the leeway available in terms of the available lookahead to a few milliseconds.

[0072] Real-time operation generally utilizes processing each acoustic block within the duration of the block itself. This means that it should take less than 10 ms to process a 10 ms buffer. Note that neural networks are not known for their lightweight computation. Further, since the data is preferably not sent to the cloud (e.g., over a network or Internet), the processing may preferably be performed on-device on computationally-constrained devices like smartphones. In addition to all the above constraints, the operating system used to buffer the sound and/or run the neural network may also have I/O delays which for audio on iOS is on the order of 4 ms, depending on the buffer size.

[0073] In real life, the target sounds experience reverberations and multipath propagation due to reflections from walls and other objects in the environment. Further, the human head and torso reflect and obstruct sounds. As a result the target sound arrives at systems described herein, such as the microphones of headset 102 of FIG. 1, with different amplitudes and delays at the two ears. The differences in the received sounds across the two ears provide spatial awareness to humans. Thus, it may be desirable to preserve these differences and play the target sounds with different amplitudes and delays through the two speakers of the headset. Note that the target and interfering sounds can be at different positions and experience different reverberations and reflections from the HRTFs. Further, the multipath effects and reverberations may be complex in real-world environments, and the HRTFs can change across wearers.

[0074] Examples of binaural target sound extraction networks are described herein. The binaural target sound extraction networks may, for example, be used to implement and/or may be implemented by the neural network 124 of FIG. 1. The networks generally receive binaural input sounds (e.g., data representing binaural input sounds generated by microphones). The networks may be implemented using one or more computing devices (e.g., one or more smartphones, tablets, computers, servers, desktops, wearable devices), such as the computing system 112 of FIG. 1. The networks may be implemented using one or more neural networks, which may include circuitry and/or executable instructions which may be executed by one or more processors for performing the neural network operations described herein. The networks may output binaural sounds that may be extracted from the input binaural sounds. The networks may be trained networks (e.g., data for implementing the networks may be obtained through training of the network on training data). The network may be trained to output binaural sounds from one or more sound classes (e.g., semantic sound classes). The output binaural sounds may be provided to one or more speakers and played for a user.

[0075] Examples of neural networks using an encoder-decoder network, e.g., binaural target sound extraction networks, are described with reference to FIGS. 3 through 5. The described neural networks may be used to implement the neural network 124 of FIG. 1, for example. FIG. 3 provides an example high-level binaural extraction framework. A mask estimation network may be an encoder-decoder architecture operating on latent space representation of binaural signals to extract the mask for target sound based on the query vector q. FIGS. 4 and 5 show the encoder and decoder architectures used in the mask estimation network. The encoder processes the previous input context and does not consider the label embedding. The decoder first conditions the encoded representation with the label embedding, 1, and then generates the mask corresponding to the target sound using the conditioned representation. These encoders and decoders may be implemented using computing devices described herein, such as one or more smartphones.

[0076] An example of a high-level framework for a binaural target sound extraction network is described. FIG. 3 is a schematic illustration of a framework for a neural network arranged in accordance with examples described herein. The framework shown in FIG. 3 may be used to implement the neural network 124 of FIG. 1, for example. The neural network framework of FIG. 3 includes a convolution block 302. An output of convolution block 302 may be provided to the mask estimation network 304. An output of the mask estimation network 304 may be combined with an output of the convolution block 302 at operator 306. An output of operator 306 may be provided to transposed convolution block 308.

[0077] Consider $s \in R^{2\times T}$ to be the input binaural signal provided to the target sound extraction network. The input binaural signal may be, for example, the audio signals received at and/or generated by the microphone 108 and microphone 110 of FIG. 1.

[0078] As shown in FIG. 3 the input binaural signal is first mapped to a representation in a latent space, $x \in R^{D\times(T/L)}$, by using a 1D convolution layer with a kernel size $\geq L$ and a stride equal to L. D and L are tuneable hyperparameters of the model. D is the dimensionality of the model, having a significant effect on the parameter count, and consequently the computational and memory complexities. L determines the duration of the smallest audio chunk that can be processed with the model. The latent space representation, x, is then passed to a mask generator, M, which estimates an element-wise mask m, given as Equation 1 in FIG. 6.

[0079] N_c in Equation 1 is the total number of sound classes the model is trained for. The representation (y) corresponding to the target sound is obtained by element-wise

multiplication of the input representation, x, and the mask, m, as given in Equation 2 in FIG. 6. The output audio signal $\hat{s} \in R^{2\times T}$ is then obtained by applying a 1D transposed convolution on y, with a stride of L.

[0080] In contrast to some example binaural extraction frameworks proposed specifically for speech, where each channel is separately and parallelly processed, examples of neural networks described herein may jointly process the two channels of binaural signals for computational efficiency. In experiments, examples of this jointly processing framework performed competitively with the parallel processing frameworks in terms of target sound extraction accuracy, even with a 50% lower runtime cost.

[0081] For real-time on-device operation, the model generally should output the audio corresponding to the target sound (e.g., target signals as described herein) as soon as the input audio is received, e.g., within the latency requirements described herein. Since the audio is fed to the model from the device buffers, the buffer size generally determines and/or influences the duration of the audio chunk the model receives at each time step. Assuming the buffer size to be divisible by the stride size L, the audio chunk size can be represented as the number of strides, K. That is, the buffer size of an audio chunk of size K is equal to KL samples. Such a real-time setup means that example models may only have access to the current and previous chunks, but not future chunks.

[0082] This means examples of neural networks provided herein, such as neural network 124 of FIG. 1, are preferably causal with a time resolution of the buffer size, e.g., KL audio samples. As a result, in examples of the high-level framework described herein and with reference to FIG. 3, the input convolution, the mask estimation block, the element-wise multiplication, and the output transposed convolution may operate on one audio chunk at each time step.

[0083] The binaural target sound extraction framework described herein may be adapted to chunk-wise streaming inference in some examples as follows. Consider the input audio signal corresponding to the kth chunk to be $s_k \in R^{2\times KL}$, as shown in FIG. 3. The input 1D convolution (e.g., convolution block 302) maps this audio chunk to its latent space representation, $x_k \in R^{D\times K}$. The mask estimation block (e.g., mask estimation network 304) is then used to estimate the mask corresponding to the target sound, based on the current chunk, as well as a finite number of the previous chunks, as given by Equation 3 of FIG. 6.

[0084] The previous chunks act as the audio context for the neural network, referred to as the receptive field of the model. An example receptive field of 1-1.5s is shown to result in

good performance. The output representation of the current chunk corresponding to the target sound, $y_k \in R^{D \times K}$, can then be obtained at the operator 306 as given by Equation 4 of FIG. 6.

[0085] The resulting output representation is then converted to the output signal $\hat{s}_k \in R^{2 \times KL}$ by applying the 1D transposed convolution in transposed convolution block 308.

[0086] Examples of architectures that may be used for mask estimation include Conv-TasNet, U-Net, SepFormer, ReSepFormer, and Waveformer. Waveformer is an efficient streaming architecture implementing chunk-based processing, which may make it advantageous in examples described herein. Examples of Waveformer are described in Bandhav Veluri, Justin Chan, Malek Itani, Tuochao Chen, Takuya Yoshioka, and Shyamnath Gollakota, "Real-time target sound extraction," in IEEE ICASSP, 2023, arXiv:2211.02250v3 [cs.SD], which publication is hereby incorporated by reference in its entirety for any purpose. Examples described herein may utilize a modified version of Waveformer to further increase efficiency without an overall loss in performance. The mask estimation network (e.g., mask estimation network 304) may be implemented as an encoder-decoder neural network architecture, where the encoder may be purely convolution-based and the decoder is a transformer decoder.

[0087] In examples described herein, the same dimensionality may be used for both the encoder and the decoder. This may allow for use of a standard transformer decoder, instead of a modified one as may be used in the Waveformer. For examples of binaural applications described herein, however, different dimensionality is not necessarily providing gains that warrant the complexity of the projection layers and the long residual connection.

[0088] FIG. 4 is a schematic illustration of an example encoder for use in neural networks described herein. The encoder of FIG. 4 may be used, for example, to implement an encoder of neural network 124 of FIG. 1. Recall mask estimation in Equation 3 involves processing many previous chunks in addition to the current chunk to obtain the mask corresponding to the current chunk. Repeated processing of the entire receptive field for each iteration could become intractable for a real-time on-device application. To mitigate this inefficiency, while achieving a large receptive field, examples of mask estimation networks described herein may implement Wavenet style dilated causal convolutions for processing the input and previous chunks. Examples of Wavenet are described in Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and

Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016, arXiv:1609.03499v2 [cs.SD], which publication is hereby incorporated by reference in its entirety for any purpose. In examples described herein, for efficient on-device inference, the dynamic programming algorithm proposed in Fast Wavenet may be implemented. Examples of Fast Wavenet are described in Tom Le Paine, Pooya Khorrami, Shiyu Chang, Yang Zhang, Prajit Ramachandran, Mark A. Hasegawa-Johnson, and Thomas S. Huang, "Fast wavenet generation algorithm," 2016, arXiv:1611.09482v1 [cs.SD], which publication is hereby incorporated by reference in its entirety for any purpose.

[0089] As shown in FIG. 4, higher efficiency may be achieved by reusing the intermediate results computed in the previous iterations. The encoder function $\mathscr{E}$ processes the input chunk x_k and an encoder context ξ_k to generate the encoded representation of the input chunk as given by Equation 5 of FIG. 6.

[0090] The size of the context ξ_k depends on the hyperparameters of the encoder. In example implementations described herein, the encoder may include a stack of 10 dilated causal convolution layers. The kernel size of all layers is equal to 3, and the dilation factor is progressively doubled after each layer starting with 1, resulting in dilation factors $\{2^0, 2^1, \ldots, 2^9\}$. Since the kernel size is equal to 3, the context needed for each dilated convolution layer is twice the layer's dilation factor. As long as this context is saved after each iteration, and padded with the input chunk in the next iteration, the intermediate results corresponding to the previous chunks do not have to be recomputed. Thus the size of the context ξ_k is equal to $2 \times \sum_{i=0}^{9} 2^i = 2046$.

[0091] FIG. 5 is a schematic illustration of an example decoder for use in neural networks described herein. The decoder of FIG. 5 may be used, for example, to implement a decoder of neural network 124 of FIG. 1. The decoder of FIG. 5 includes a combination of the encoded data from an encoder (e.g., from the encoder of FIG. 4) with an embedding, l. The combination is then provided to multiple multi-head attention layers, each followed by an add and normalization block. Lastly, a feed-forward block is provided followed by a final add and normalization block to provide an output of the decoder. The embedding l may correspond with one or more sound classes described herein and/or other indication of target signals.

[0092] Referring to FIG. 5, the query vector q is first embedded into the embedding space using a linear layer to generate a label embedding $l \in R^D$. The mask corresponding to the target sound m_k is estimated using a transformer decoder layer, represented here as $\mathscr{D}$. The

encoded representation is first conditioned with the label embedding l by an element-wise multiplication. The encoded representation and the conditioned encoded representation are first concatenated in the time dimension, with those from the previous time step, before processing with the transformer decoder layer $\mathscr{D}$. The encoded representation from the previous time step, e_{k-1}, acts as the decoder context. The mask estimation can be written as Equation 6 in FIG. 6, where {} represents concatenation in the time dimension. As shown in FIG. 6, the transformer decoder $\mathscr{D}$ first computes the self-attention result of the conditioned encoded representation $\{l\ e_{k-1}, l\ e_k\}$ using the first multi-head attention block, followed by cross-attention between the self-attention result and the unconditioned encoded representation $\{e_{k-1}, e_k\}$ using the second multi-head attention block. A feed-forward block along with residual connection generates the final mask corresponding to the target signals.

[0093] Examples of neural networks described herein, such as neural network 124 of FIG. 1, may be trained to extract target signals based on an indication of audio sources. For example, neural networks may be trained to extract target signals based on an indication of sound class. Examples of audio class dataset curation are described and then training methodologies are described to generalize to real-world scenarios.

[0094] Example systems should preferably efficiently handle target sounds encountered in real-world situations. By focusing on practical applications, a manageable set of target sound classes may be identified to extract. However, in reality, a wide range of background sounds may be presented, many of which may not be part of the system's target sound classes. To curate a dataset of sound classes, an ontology (e.g., an AudioSet ontology) may be followed, which provides a comprehensive and structured representation of the relationships between various sound classes. The ontology arranges the sound classes as nodes in a graph and groups them into seven main sound categories. Each sound class node has a unique AudioSet ID and may contain one or more child nodes that represent more specific sound classes. For example, the "Hands" sound class has two children, namely "Finger Snapping" and "Clapping." Examples of target sound classes are described as well as the interfering classes.

[0095] Various indoor and outdoor scenarios where the system is likely to operate are considered, such as beaches, parks, streets, living rooms, offices, and cafes. Based on these scenarios, potential sound sources may be identified that are prevalent in such locations, such as human speech, dogs, cats, birds, sea waves, and music. Sound classes associated with these selected target sounds may be developed and mapped to a label, such as in the AudioSet ontology. In some examples, 20 sound classes were selected that human listeners could distinguish with reasonably high accuracy.

[0096] In the real world, the interfering sounds and noise often do not belong to the example target sound classes identified, such as the 20 target sound classes mentioned. To create a neural network that can generalize to interference from these sounds, a diverse set of interfering sound classes may be preferable in a dataset. Note also that the target sound classes may be interfering with each other. Note that these sounds can come from a very large variety of sources, which may make it infeasible to exhaustively enumerate all of them. Secondly, since they may be used as interfering signals, it may be preferable to ensure that these sound classes do not overlap with a set of target classes. To overcome these constraints, examples may use the AudioSet hierarchical structure and a set of target classes (e.g., 20 target classes) to generate a large set of other sound classes (e.g., 141 other sound classes). For example, this set may be defined as the nodes that are neither a more specific nor a more general instance of any target (or known) class, according to the AudioSet hierarchy. Accordingly, by considering the AudioSet ontology as a directed acyclic graph with edges from each sound class node towards its child nodes, unknown sound classes may be defined as the set of AudioSet nodes that are disconnected from all target sound class nodes.

[0097] Given the sets of the target and other sound classes, labeled audio recordings may be obtained for each of the sound classes. Note that it may not be preferable to rely only single general-purpose audio-tagging datasets. This is because such datasets may not contain audio samples of all target sound classes, and may contain a limited number of audio samples from the other sound classes. So, in some examples, combined audio samples from multiple (e.g., four) different datasets may be used: in some examples those may be FSD5OK (general-purpose), ESC-50 (environmental sounds), MUSDB18 (music and vocals) and noise files for the DISCO dataset (noise sounds). The datasets are described in the following publications, for example, all of which are incorporated by reference in their entirety for any purpose: Eduardo Fonseca, Xavier Favory, Jordi Pons, Frederic Font, and Xavier Serra, "Fsd50k: An open dataset of human-labeled sound events," 2022, arXiv:2010.00475v2 [cs.SD]; Karol J. Piczak, "ESC: Dataset for environmental sound classification," MM '15: Proceedings of the 23rd ACM International Conference on Multimedia, pp. 1015–1018, New York, NY, USA, 2015; Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner, "Musdb18 – a corpus for music separation," 2017, 10.5281/zenodo.1117371. hal-02190845; Nicolas Furnon, Noise files for the disco dataset, September 2020, 10.5281/zenodo.4019030. A script to download these files is available on https://github.com/nfurnon/disco.

[0098] Since each dataset uses different class names, the class labels may be standardized into the AudioSet labels by mapping every class in each dataset to the semantically closest label in the AudioSet ontology, if any. For FSD5OK and MUSDB18, additional dataset-specific pre-processing procedures may be performed. For example, to create binaural mixtures of individual sources from multiple directions, audio samples may be excluded from FSD5OK that were already mixtures of multiple distinct sound sources. For MUSDB18, examples may extract and split audio into vocal and instrumental streams and assign them the AudioSet labels "Singing" and "Melody" respectively.

[0099] In some examples a set of sound classes may be used which includes: alarm clock, baby cry, birds chirping, car horn, cat, rooster crow, typing, cricket, dog, door knock, glass breaking, gunshot, hammer, music, ocean, singing, siren, speech, thunderstorm, toilet flush. Other classes may be used in other examples.

[0100] The resulting audio samples may be divided into segments and silent ones discarded. Each dataset may be split into mutually exclusive training, testing, and validation sets and then combined into a final dataset. In implemented examples, for both the FSD5OK and MUSDB18, the training and validation audio files were sampled from the development split (90-10 split), and the testing samples from the evaluation split. For the ESC-50 dataset, the first three folds were used for training, the fourth fold for validation and the fifth for testing. For the DISCO noise dataset, the audio samples for each sound class were split into train, test and validation sets (60-33-7) before combining with the rest of the datasets. The final combined dataset includes 20 target sound classes and 141 other sound classes.

[0101] These examples include how single-channel sound classes may be sampled from audio datasets. In some examples, binaural data sets may be created and/or used to train neural networks described herein. It may be preferable to create binaural mixtures that (1) are representative of spatial sounds perceived by a diverse set of listeners, and (2) capture the idiosyncrasies of real-world reverberant environments. For this purpose, examples described herein may use a dataset including human head-related transfer function (HRTF) measurements. In one example, a pre-existing dataset of 43 human (HRTF) measurements were used. This may be augmented with additional (e.g., three) datasets of measured and simulated reverberant binaural room impulse responses (BRIRs).

[0102] Each dataset may be split across rooms and listeners into train, test and validation (70-20-10) sets. BRIR subjects or rooms may not be sampled across different sets. For each sample during training, one of the datasets may be randomly chosen and may sample a single

room and participant from its training set. Then, to create a binaural mixture with K sources, a source direction was independently selected for each of the K sources, out of all source directions available for this room and this participant subject in the dataset. Note that since the source directions are independently picked, two different sound sources might end up being at the same direction from the wearer. A set of $2K$ room impulse responses $h_{1,L}, h_{1,R}, h_{2,L}, \ldots, h_{K,R} \in \boldsymbol{R}^N$, where N is the length of the room impulse response, may be obtained. Hence, for a training sample with input audio signals of length T samples long, denoted by $x_1, x_2, \ldots, x_n \in \boldsymbol{R}^T$, we can compute the sound received at the left and right ears, s_L and s_R as, $s_L = \sum_{k=1}^{K} x_k * h_{k,L}$ and $s_R = \sum_{k=1}^{K} x_k * h_{k,R}$. Here "*" represents the convolution operation.

[0103] For training, a toolkit, such as the Scaper toolkit, examples of which are described at Justin Salamon, Duncan MacConnell, Mark Cartwright, Peter Li, and Juan Pablo Bello, Scaper: A library for soundscape synthesis and augmentation, in WASPAA, 2017, may be used to synthesize binaural mixtures dynamically on the fly during training. For training and validation, binaural mixtures may include two randomly picked target classes, each with a 5-15 dB SNR relative to the background sounds, and 1-2 other classes that each have a 0-5 dB SNR relative to the background sounds. Background sounds sourced from the TAU Urban Acoustic Scenes 2019 dataset were also used in mixtures described herein. See Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen, "A multi-device dataset for urban acoustic scene classification," in DCASE2018, November 2018, arXiv:1807.09840v2 [eess.AS], which is hereby incorporated by reference in its entirety for any purpose. Each mixture was 6 seconds long in some examples. Sounds from the target and other background classes were between 3 and 5 seconds long in some examples, while background urban sounds persist for the entire duration of the mixture. In addition to the mixture, ground truth signals y_L and y_R were also synthesized at the left and right channels, respectively, for each chosen target sound source t as, $y_L = x_t * h_{t,L}$ and $y_R = x_t * h_{t,R}$.

[0104] A network was then trained to produce a pair of left and right channel target sound estimates $\hat{y}_L$ and $\hat{y}_R$. To preserve the spatial cues, such as interaural time differences (ITD) and interaural level differences (ILD), the sample-sensitive and scale-sensitive signal-to-noise ratio (SNR) loss function was used and applied independently, and then the left and right SNRs were averaged to obtain the loss function as shown in Equations 7 and 8 in FIG. 6.

[0105] A transformer model was trained for 80 epochs, with an initial learning rate of 5e-4. After completing 40 epochs, the learning rate was halved if there was no improvement in the validation SNR for more than five epochs. Note that in some examples the training data do not include any measurements with example binaural hardware described herein, and the results reported may accordingly be used to evaluate generalization to example hardware, unseen users and environments.

[0106] Generally, a variety of binaural target signal extraction frameworks may be used. In some examples, a dual-channel architecture may be used for efficient binaural target sound extraction. In this framework, the binaural signal is converted into a combined latent space representation before the mask estimation. Since both left and right channels are combined into a common representation, a single instance of the mask estimation network is used for estimating the mask corresponding to the target sound. We consider our mask estimation architecture with both $D = 128$ and $D = 256$.

[0107] In some examples, a parallel may be used that implements parallel processing of the left and right channels, along with some cross-communication between channels. This framework may be implemented for both a mask estimation network with $D = 128$ and Conv-TasNet, for example. Other networks may be used in other examples.

IMPLEMENTED EXAMPLES

[0108] An implemented example arranged in accordance with techniques described herein includes an augmented off-the-shelf noise-cancelling headset with commercial wired binaural earphones that provide access to data from both microphones. An example neural network was implemented on a connected smartphone and trained with 20 different sound classes, including sirens, baby cries, speech, vacuum cleaners, alarm clocks, and bird chirps.

[0109] Example results included an average signal improvement of 7.17 dB across 20 target sounds, in the presence of interfering sounds and urban background noise. An example implemented real-time network had a 6.56 ms runtime on an iPhone 11 for processing a 10 ms chunk of binaural audio.

[0110] In-the-wild evaluation was performed with participants in various indoor and outdoor scenarios with our hardware and showed that the example system can extract the target sounds and generalize to previously unseen participants and environments, without using any training data collection with the implemented example hearable hardware.

[0111] In a spatial hearing study where sounds were played from different directions in five previously unseen rooms, participants were able to predict the direction of the target sounds

output by the example system with 50th and 90th percentile errors of 22.5° and 45° respectively. These errors were similar for noise-free clean sounds.

[0112] In a user study with 22 participants who spent over 330 minutes rating binaural data from real-world indoor and outdoor environments, an example implemented system achieved a higher mean opinion score and interference removal for the target sounds than the binaural input.

[0113] An example setup for real-world evaluations is described and then binaural network benchmarks are presented.

[0114] An example hardware setup includes a pair of SonicPresence SP15C binaural microphones that are wired to capture high-quality recordings. An iPhone 12 was used to process the recorded data and output the audio through noise-cancelling headphones like JBL Live 65OBTNC and the NUBWO gaming headsets.

[0115] A lightning-to-aux adapter was used to connect the headphones to the iPhone over a wire. A USB hub was used to connect both the microphones and the headphones to the smartphone.

[0116] Nine individuals were recruited (three female, six male) across in-the-wild and spatial cues evaluations. Twenty-two participants (six female, 16 male) were also invited for an online hearing study.

[0117] To evaluate the implemented system in real-life scenarios, in-the-wild experiments were conducted to assess the effectiveness of example systems described herein.

[0118] Five individuals (three female and two male) were recruited to wear example hardware and collect sounds in the real world. These experiments were conducted in typical application settings: offices, living rooms, streets, rooftops, parks, and restrooms. Since some of the sound classes were relatively less common, the in-the-wild experiments had a subset of classes which most commonly appeared in our recordings: alarm clock, car horn, door knock, speech, computer typing, hammer, birds chirping, and music. The position and movement of the sound sources were uncontrolled and reflective of real-world scenarios, where the sound sources could be mobile. Furthermore, in all experiments, participants had complete freedom to move their heads, causing the sound source positions relative to the microphones to vary over time. Thus, an in-the-wild evaluation captured both mobile wearers as well as mobile sound sources that naturally occurred in real-world scenarios (e.g., cars moving or birds flying).

[0119] Unlike with examples of simulated training data, clean, sample-aligned ground truth signals may not be available to objectively compare the binaural outputs of the system with. Hence, a listening study was conducted to compute a mean opinion score (MOS) regarding the sound extraction accuracy. This metric may be used to evaluate the perceptual quality of algorithms described herein for end users. In an implemented study, 22 participants (six female, 16 male, mean age 34.6) were invited to the listening study. The study included 16 sections. In each section, the participants evaluated the quality of three or four 5.0-8.5 second audio samples. The audio samples played at each section were in-the-wild recordings processed in the following three ways for the same target label: (1) the original recording, (2) the output of a 128-dimensional binaural network arranged in accordance with examples described herein, and (3) the output of a 256-dimensional binaural network arranged in accordance with examples described herein. For the subset of the evaluations that involved speech as the target sound, an additional fourth audio sample was also included that was obtained by extracting the interfering class (e.g., door knocks) and then subtracting it from the input recording to estimate the target speech.

[0120] Results of user evaluations for the interference sound suppression and overall quality improvement of an example system for different target sound labels were obtained. The results demonstrated the system's capability to significantly reduce unwanted background sounds, as indicated by an increase in the overall noise suppression score. A similar trend was also observed in the overall MOS improvement, with an improvement from 2.63 for the input signal to 3.54 and 3.80 after processing with the 128-dimensional and 256-dimensional models, respectively. Results also showed that examples of networks described herein preserve the timing of the target sounds and can silence noise outside the target sound duration.

[0121] From the foregoing it will be appreciated that, although specific embodiments have been described herein for purposes of illustration, various modifications may be made while remaining with the scope of the claimed technology.

[0122] Examples described herein may refer to various components as "coupled" or signals as being "provided to" or "received from" certain components. It is to be understood that in some examples the components are directly coupled one to another, while in other examples the components are coupled with intervening components disposed between them. Similarly, signal may be provided directly to and/or received directly from the recited components without intervening components, but also may be provided to and/or received from the certain components through intervening components.

CLAIMS

What is claimed is:

1. A method comprising:

receive audio signals from at least one microphone;

generate noise cancellation signals based at least in part on the audio signals;

extract target signals based at least in part on the audio signals using digital neural network processing; and

provide the noise cancellation signals and the target signals from at least one speaker.

2. The method of claim 1, wherein said generate noise cancellation signals comprises using signal processing.

3. The method of claim 2, wherein said signal processing comprises analog signal processing.

4. The method of claim 1, wherein said generate noise cancellation signals is performed independent of said extract target signals.

5. The method of claim 1, further comprising receive an indication of a sound class from a user and wherein said digital neural network processing is configured to extract the target signals based at least in part on the indication of the sound class.

6. The method of claim 1, wherein said extract the target signals comprises extract signals originating from one or more human speakers.

7. The method of claim 1, wherein said target signals are binaural signals, and wherein said provide the target signals from the at least one speaker comprises preserving a directionality of the binaural signals.

8. The method of claim 1, wherein a time to generate a portion of said noise cancellation signals based on a portion of the audio signals is a shorter time than a time to extract a portion of said target signals based on the portion of the audio signals.

9. The method of claim 1, wherein said using digital neural network processing comprises using an encoder-decoder architecture.

10. The method of claim 9, wherein said encoder-decoder architecture comprises:

an encoder configured to encode the audio signals independent of the target signals to generate encoded data; and a decoder configured to condition the encoded data with an embedding to provide conditioned data and extract the target signals based on the conditioned data.

11. The method of claim 10, wherein the embedding is indicative of a sound class or sound source.

12. A system comprising:

a plurality of microphones, the plurality of microphones configured to generate binaural audio signals from an environment;

at least one processor;

at least one non-transitory computer readable medium encoded with instructions which, when executed by the at least one processor, cause the system to perform operations comprising:

utilize a trained neural network to extract target signals from the binaural audio signals based on a sound class for the target signals, and wherein the target signals comprise binaural target signals; and

a plurality of speakers, the plurality of speakers configured to play back the binaural target signals.

13. The system of claim 12, wherein the operations further comprise:

receive an indication of the sound class from a user.

14. The system of claim 12, wherein the plurality of microphones and the plurality of speakers are disposed in a headset.

15. The system of claim 14, wherein the at least one processor and the at least one non-transitory computer readable medium are disposed in a computing device in communication with the headset.

16. The system of claim 12, wherein the plurality of microphones and the plurality of speakers are disposed in a hearing aid.

17. The system of claim 12 further comprising signal processing circuitry, the signal processing circuitry configured to generate noise-cancelling signals based on the binaural audio signals.

18. The system of claim 12, wherein the sound class comprises human speakers.

19. The system of claim 12, wherein the sound class comprises a class of sounds to eliminate from the binaural audio signals.

20. The system of claim 12, wherein the trained neural network comprises an encoder-decoder architecture comprising:

an encoder configured to encode the binaural audio signals independent of the target signals to generate encoded data; and

a decoder configured to condition the encoded data with an embedding to provide conditioned data and extract the target signals based on the conditioned data.

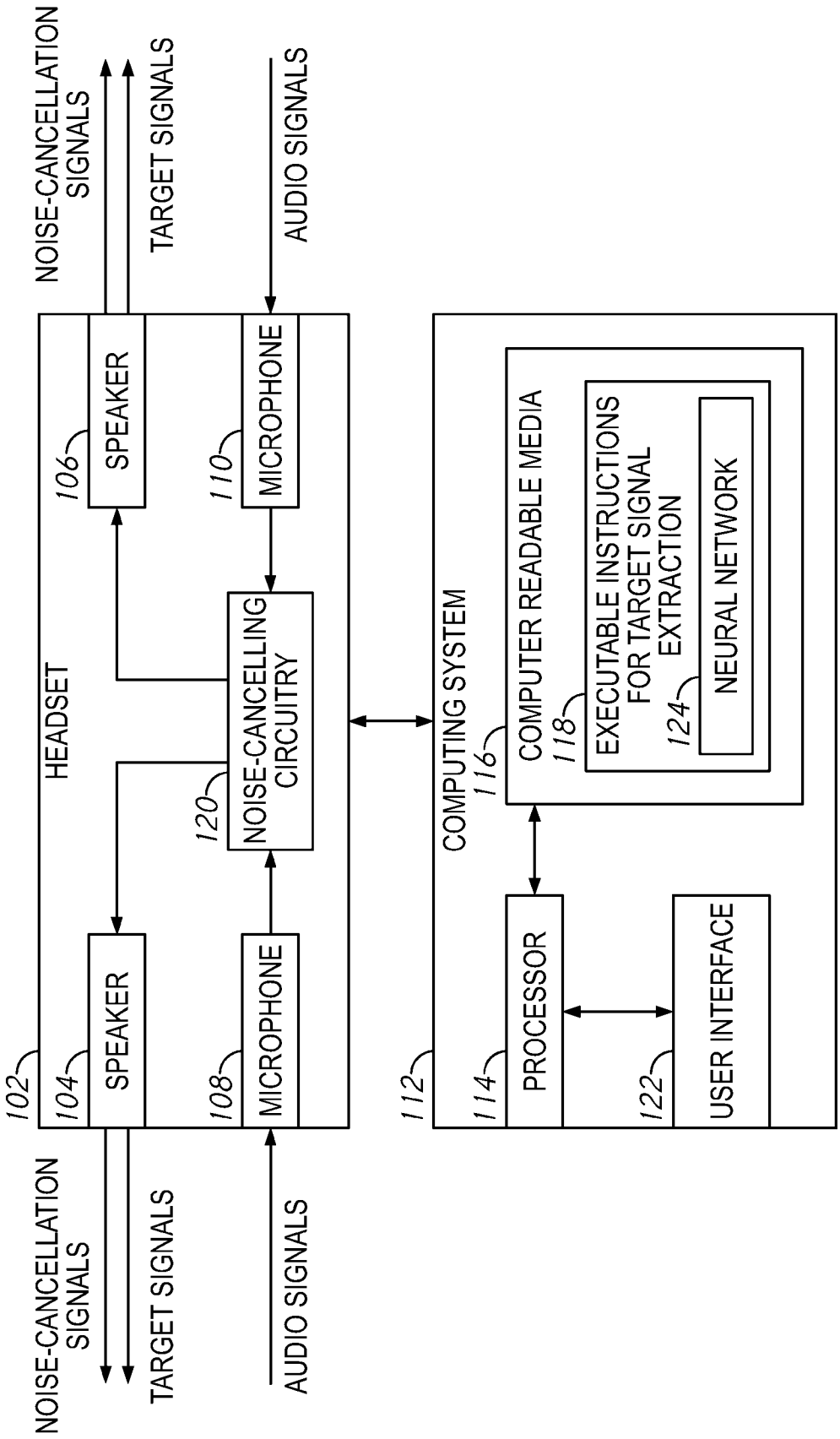
NOISE-CANCELLATION SIGNALS
TARGET SIGNALS
AUDIO SIGNALS
HEADSET
102
SPEAKER
104
SPEAKER
106
MICROPHONE
108
MICROPHONE
110
NOISE-CANCELLING CIRCUITRY
120
COMPUTING SYSTEM
112
PROCESSOR
114
USER INTERFACE
122
COMPUTER READABLE MEDIA
116
EXECUTABLE INSTRUCTIONS FOR TARGET SIGNAL EXTRACTION
118
NEURAL NETWORK
124
NOISE-CANCELLATION SIGNALS
TARGET SIGNALS
AUDIO SIGNALS

FIG. 1

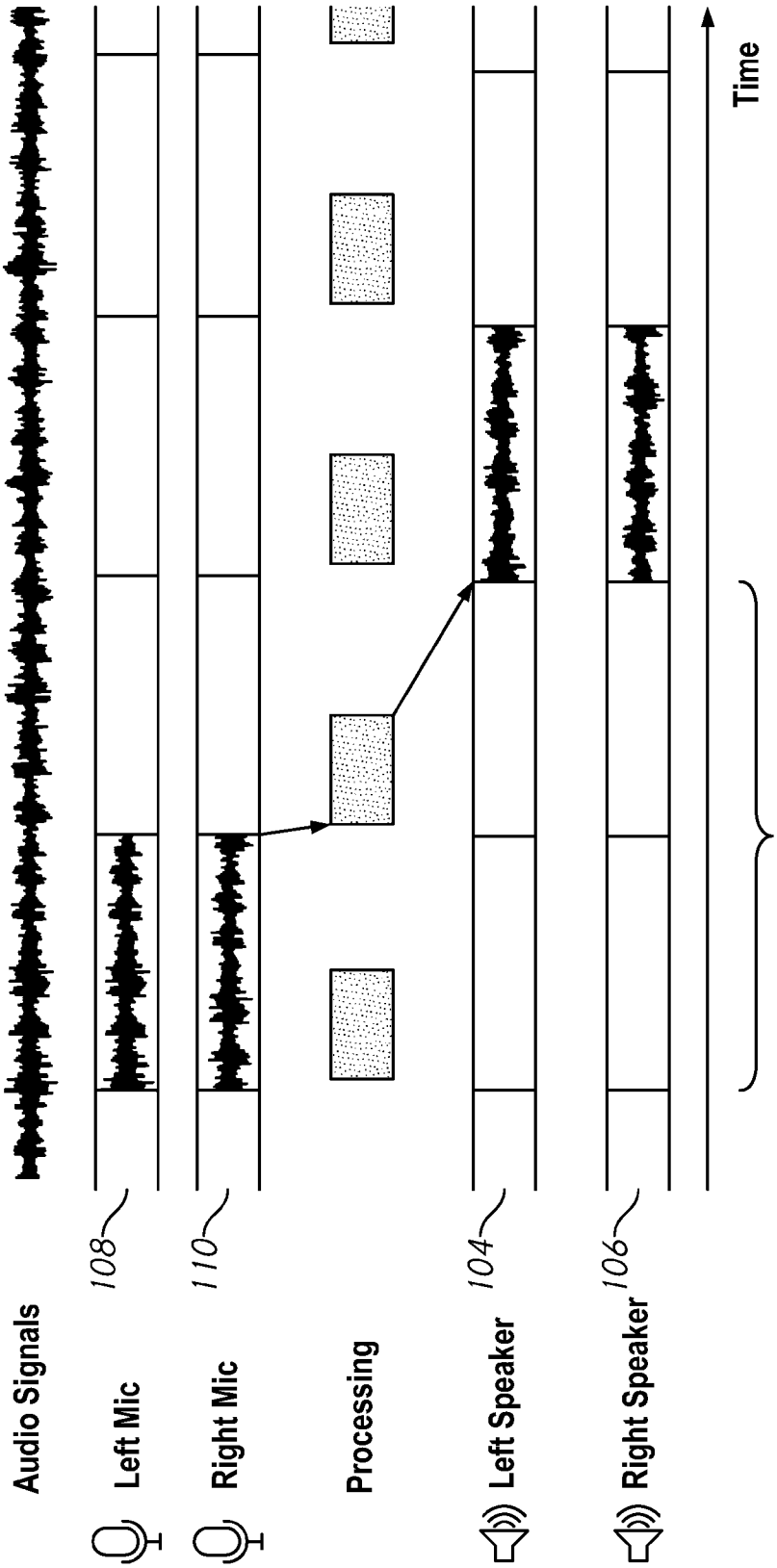
Audio Signals
Left Mic
Right Mic
Processing
Left Speaker
Right Speaker
108
110
104
106
Time
Latency

FIG. 2

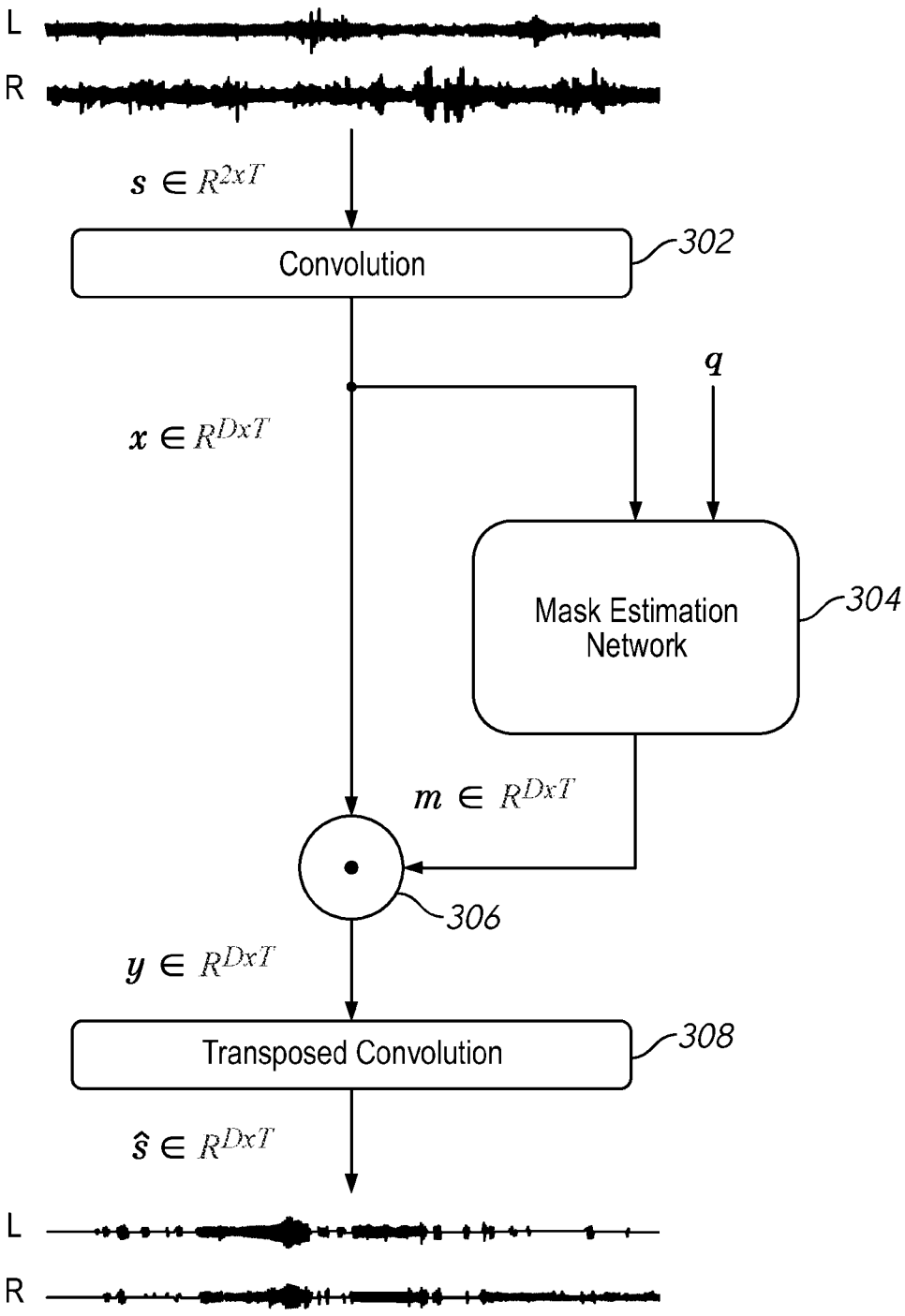
L
R
$s \in R^{2xT}$
Convolution
302
q
$x \in R^{DxT}$
Mask Estimation Network
304
$m \in R^{DxT}$
306
$y \in R^{DxT}$
Transposed Convolution
308
$\hat{s} \in R^{DxT}$
L
R

FIG. 3

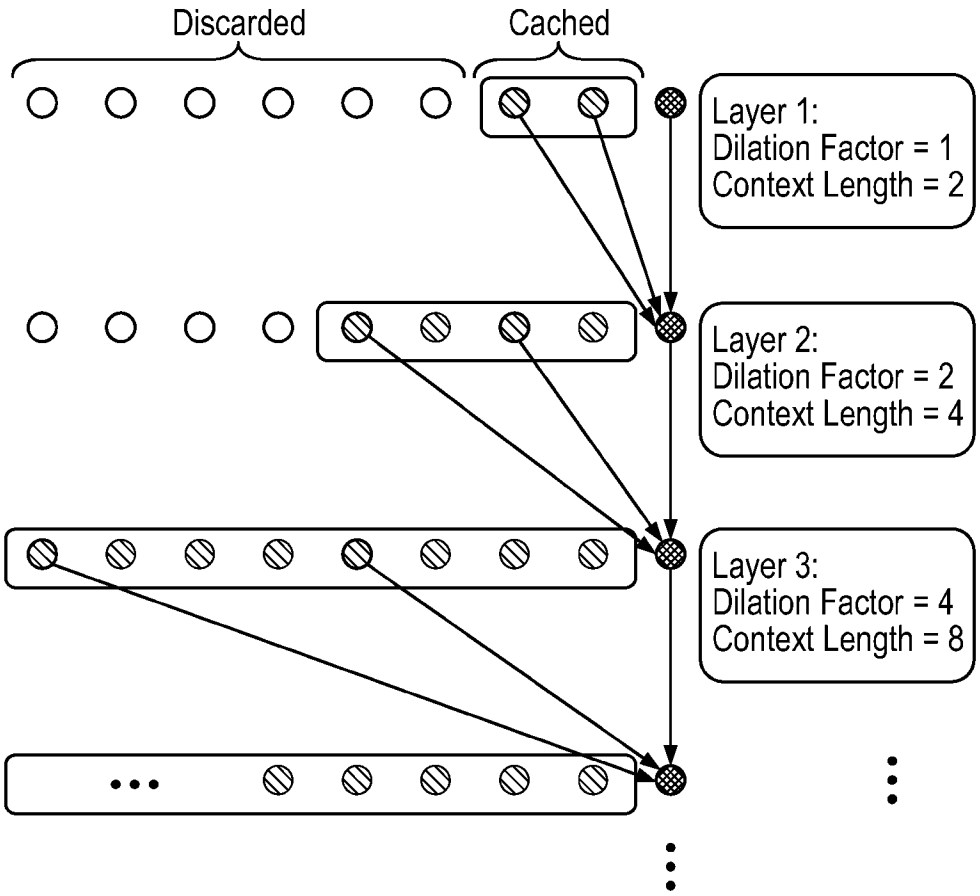
Discarded
Cached
Layer 1:
Dilation Factor = 1
Context Length = 2
Layer 2:
Dilation Factor = 2
Context Length = 4
Layer 3:
Dilation Factor = 4
Context Length = 8

FIG. 4

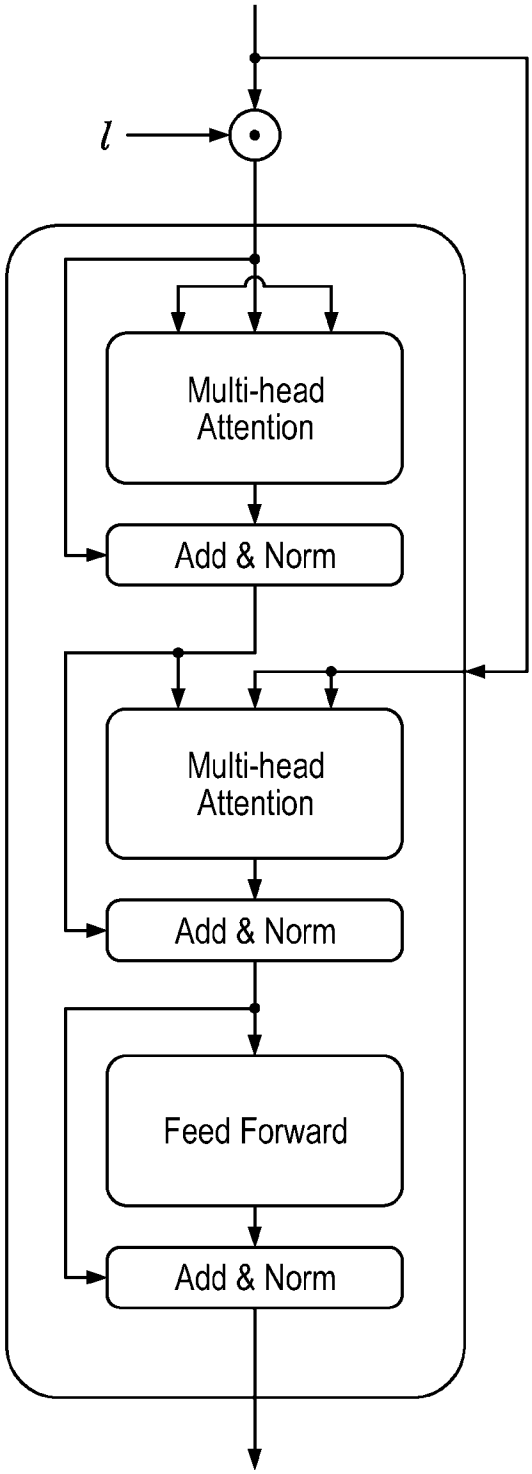
l
Multi-head Attention
Add & Norm
Multi-head Attention
Add & Norm
Feed Forward
Add & Norm

FIG. 5

EQUATION 1. $$m = \mathcal{M}(x, q) \mid m \in R^{D\times(T/L)};\ q \in \{0,1\}^{N_c}$$

EQUATION 2. $$y = x \odot m \mid y \in R^{D\times(T/L)}$$

EQUATION 3. $$m_k = \mathcal{M}(x_k, q, x_{k-1}, x_{k-2}, \cdots) \mid m_k \in R^{D\times K}$$

EQUATION 4. $$y_k = x_k \odot m_k$$

EQUATION 5. $$e_k, \xi_{k+1} = \varepsilon(x_k, \xi_k) \mid e_k \in R^{D\times K}$$

EQUATION 6. $$m_k = \mathcal{D}(\{l \cdot e_{k-1}, l \cdot e_k\}, \{e_{k-1}, e_k\},)$$

EQUATION 7. $$SNR(\hat{x}, x) = 10\log\left(\frac{\|x\|^2}{\|x-\hat{x}\|^2}\right)$$

EQUATION 8. $$L = -\left(\frac{1}{2}SNR(\hat{y}L, yL) + \frac{1}{2}SNR(\hat{y}R, yR)\right)$$

FIG. 6