
AUTOMATIC CONTEXTUAL AUDIO DENOISING

Diep Luong^{1,2}, Konstantinos Drossos², Mikko Heikkinen², and Tuomas Virtanen¹

¹Tampere University, Finland

²Nokia, Finland

ABSTRACT

Audio context determines which sound components and sources are relevant and which can be perceived as irrelevant (noise) by listeners. For example, traffic noise is informative in urban surveillance but noise for a phone call at the same location. Most current audio denoising systems apply fixed target–noise definitions, often removing useful components in one context while failing to suppress irrelevant components. To address this, we introduce the concept *automatic contextual audio denoising* (ACAD) which defines target and noise based on the inferred context. In this work, we restrict context to be associated with an acoustic scene class. We label sound events outside the event distribution of a scene class (noise) as out-of-context (OC) and events typical for that scene as in-context (IC). We implement a deep learning method that automatically infers the context of the audio signal and removes OC components, and benchmark it against variants: without context inference, with oracle context, and with separately provided uninformative context. On paired clean/noisy data across diverse contexts, where OC components in one context may be IC in another, our proposed method outperforms other approaches across standard objective metrics, indicating that the model can infer context and context-dependent processing can enhance denoising.

Keywords automatic contextual audio denoising, audio denoising, source separation, context-dependent, audio understanding, neural network.

1 Introduction

The distinction between relevant and irrelevant (noise) sound components and sources inherently depends on different audio contexts and the listener’s goals [1]. For example, traffic noise is an informative source in urban surveillance or traffic monitoring systems but noise for a phone call on the street at the same location. Since denoising aims at suppressing components that interfere with the source of interest, the objective of audio denoising systems should extend beyond merely estimating a fixed, predefined target source from a noisy audio mixture. To optimize the overall perceptual listening experience, it is beneficial to define target and noise components depending on the audio context.

Despite impressive advancements of deep learning approaches in audio denoising [2, 3, 4, 5, 6], the vast majority of current models falls short of considering the audio context. Typically, models are trained to learn a mapping from noisy to target signal, where the targets are specific classes of sound (e.g. speech). Under reconstruction-based

objectives, models learn global, fixed inference rules that distinguish target and noise components based on the statistical characteristics learned during training. A rigid definition of target components and the lack of context dependence most likely strips away from the audio components and sources that are useful in some context, which contribute to an immersive and natural audio experience, while fails to suppress irrelevant components.

Related work on source separation offers mechanisms that enable more adaptive definition of target sources. Instead of relying on a single global definition of the target across all mixtures, target source separation methods condition the model on information about the target source to determine the target distribution. Enrollment audio is often used for conditioning in target speaker extraction [7, 8]. Beyond audio guidance, several studies leverage visual information about the target speaker to improve separation [9, 10, 11]. Multiple cross-modality cues (e.g., sound event tags, textual descriptions, and video clips) have been employed for target source extraction [12]. Text-derived semantic information of the target sources has also been used in extraction task [13]. Besides multimodal information, human-input guidance via text-based input prompts has been utilized to condition the separation [14]. In all these studies, auxiliary information of the target source is required to guide the separation process.

In other audio processing tasks, context information has been used to enhance the performance. For example, predefined scene contexts have been used to facilitate sound event detection in seen and unseen scenes [15]. Taking into account that what a person says depends on the current particular context they are in, a contextualized automatic speech recognition model [16] has been proposed.

To address the lack of context dependence in current audio denoising and motivated by the use of context in other audio processing tasks, we propose the concept of **automatic contextual audio denoising** (ACAD). In ACAD, target and noise are context-dependent, i.e. ACAD defines the target and noise distributions under the learned context, and different contexts can give different definitions of target and noise. ACAD hence avoids static, context-independent definitions of target and noise, and operates without explicit guidance about which sound components constitute the target.

Under the above general definition of ACAD, in this study, we focus on removing out-of-context (OC) components from a noisy mixture based on the learned audio context. We define OC components that can detract from the primary audio context as noise, while such OC components in one context can be in-context (IC) in another. To start and foster further work on ACAD, we present a first baseline method and a corresponding dataset [17] consisting of pairs of clean and noisy audio across different contexts, which we release publicly. The baseline is a DNN-based model that infers the context and performs denoising using only the noisy audio signal as input. We evaluate the performance of the baseline method against three other approaches, with no context inference, with oracle context, and with separately provided uninformative context. Obtained results show that the proposed ACAD method outperforms the other approaches, suggesting that informative context can enhance the ACAD process and that the learned context (compared to oracle-based approach) can further enhance the ACAD. The rest of the paper is organized as follows: Section 2 presents the problem setup, the dataset, and our proposed method. In Section 3 we present the evaluation setup of the ACAD method. Results are discussed in Section 4, and Section 5 concludes this paper.

2 Problem setup, dataset, & proposed method

In ACAD, a method takes a single input audio signal, \tilde{x} , infers its context, e , and outputs x in which all components and sources considered noise with respect to e are removed.

In this first study, we consider each audio context to be associated with one acoustics scene class, and we use a fixed set of acoustic scene classes. We define context as the statistical properties of a audio signal, that can consistently

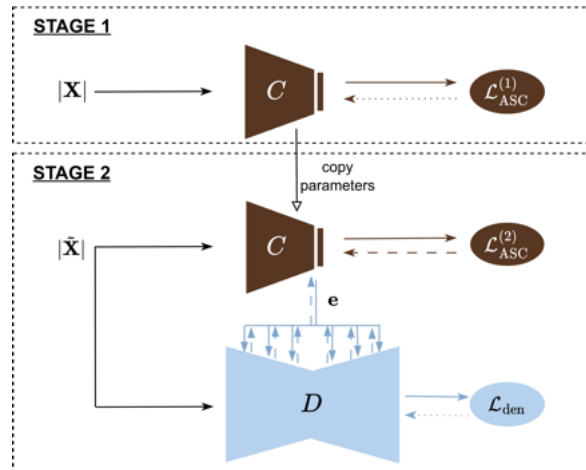


Figure 1: Overview of our ACAD method. Context extractor C is pretrained for ASC on clean audio spectrogram. Denoiser D removes OC components from noisy spectrogram under learned context e from C . In the frozen setting, C is fixed; in the finetuned setting, C is trained jointly with D . Backpropagation is shown with dashed or dotted arrows; backpropagation along dashed arrows is only available in the finetuned setting.

be classified with a specific acoustic scene class. Such statistical properties can include, but are not limited to, sound event distribution, spectral and temporal patterns, loudness, dynamic range, and other related and high-level factors that can be used to define a scene. OC component (or noise) denotes outlier sound events that do not belong to the current scene’s sound event distribution.

Consequently, we consider \tilde{x} as the noisy audio signal and x as the corresponding clean one without any OC components under the current context. Thus, the ACAD method takes as an input \tilde{x} , infers e , and performs contextual denoising on \tilde{x} . The output of the method is an estimate of x , \hat{x} .

2.1 Dataset construction

To create our ACAD dataset, we employ a dataset of real-life recordings from different acoustic scene classes and a labeled sound event dataset. The former serves as the clean (target) signals x and the latter as the pool for the OC components used to construct noisy scenes \tilde{x} by mixing them with the clean signals. The acoustic scene audio from CochScene dataset [18] and the OC components from FSD50K [19] are used to create the dataset in this study. We focus on six acoustic scene classes from CochScene: Kitchen, Park, Restaurant, Restroom, Street, and Subway, which are distinct from each other considering their audio content. We construct the noisy audio by mixing clean audio and OC events as noise.

To identify OC components for each acoustic scene class, we first define preliminary IC and OC sets of event classes.

<https://doi.org/10.5281/zenodo.20287453>

A set of event classes within each scene class is identified using pre-trained PANNs for sound event detection [20]. Within each time step in the PANNs output, if an event label and any of its child labels in the AudioSet ontology [21] are present together, the parent label is removed while retaining all child labels (e.g., if “Animal”, “Bird”, and “Dog” are present, only “Bird” and “Dog” are kept). Event classes within one scene class are sorted in a decreasing order according to their activity, which is quantified as the total active duration. The 20 most active event classes and their hierarchical descendants are considered as the preliminary IC set for each scene class. Conversely, the direct siblings (and their descendants) of the 20 most active event classes form the preliminary OC. We augment the preliminary OC set with the 20 most active events (with descendants) and their siblings (with descendants) from other scene classes, but excluding those already in the preliminary IC set.

A manual refinement step is performed to ensure a clear distinction between IC and OC events. This involved informal human judgement on the semantic similarity between each preliminary IC event class, k , and any of its ancestors, p , found in the preliminary set of OC event classes, following three rules: 1) *Highly similar parent*: if p and its descendants are very similar to k , p and descendants are discarded from OC, 2) *Partially similar parent*: if p has diverse descendants distinct from k but p itself could sound similar to k , only p is removed from OC, and 3) *Distinct parent*: if p and its descendants are diverse and acoustically distinct from k , p remains in OC.

The dataset construction process ensures that the OC to be removed in a specific scene class can appear to be the IC within other acoustic scene classes. This is an important point for the ACAD task, since it enforces context-dependent learning of OC components and impede the model to rigidly learn components to be removed globally across all scene classes.

Specifically, following the steps mentioned above, we employ Scaper library [22] to generate $N_{train} = 10000$ noisy-clean pairs of 10 sec audio for training and $N_{val/test} = 3000$ for validation/testing for each of the ($C = 6$) acoustic scene classes. We follow the splitting of the employed datasets, mixing scenes and events from corresponding splits (e.g., training with training, etc). Each clean audio file, scaled to a random LUFS level in [-10,-15] LUFS, is mixed with 1 to 3 OC event classes selected from the refined OC set. OC events, with 1 to 2 instances per event class (each of 0.5s-3s long), are added to the clean audio at a level of [-5,10] dB compared to the background scene.

Finally, we obtain

$$\mathbb{D} = \{ \{ (\tilde{\mathbf{x}}^{n,i}, \mathbf{x}^{n,i}) \}_{n=1}^N, c^i \}_{i=1}^C,$$

comprising of N pairs of $\tilde{\mathbf{x}}$ and corresponding \mathbf{x} for each of C distinct acoustic scene classes c^i .

2.2 Proposed method

Our method for ACAD, illustrated in Figure 1, consists of a context extractor C and denoising model D . C takes audio $\tilde{\mathbf{x}}$ as an input and outputs a latent embedding \mathbf{e} that represents the estimated denoising context. D takes $\tilde{\mathbf{x}}$ as an input, conditioned on \mathbf{e} , and outputs $\hat{\mathbf{x}}$. We employ a two-stage training for our method, including first the pretraining of C followed by the training of D . In the second stage, we integrate C in two settings; i) frozen, where C is kept fixed during the training of D , and ii) finetuned, where C is finetuned jointly with D .

In the first stage and in both settings, C is pretrained to perform acoustic scene classification (ASC) on clean audio \mathbf{x} . Specifically, C computes the log Mel spectrogram $|\mathbf{X}|_{\text{mel}}$ by applying Mel filter banks to the short-time Fourier transform (STFT) magnitude spectrogram $|\mathbf{X}|$ of \mathbf{x} , and outputs the probability distribution of acoustic scene classes. For the pretraining of C , we minimize the cross-entropy loss for ASC

$$\mathcal{L}_{\text{ASC}} = \mathbb{E}_{(\mathbf{x},c) \sim \mathbb{D}} [-\log p_c] \quad (1)$$

where p_c is the predicted probability corresponding to the ground truth acoustic scene class c .

In the second stage, D is trained to do denoising of $\tilde{\mathbf{x}}$, given the inferred context. To condition D on the context, C computes $|\tilde{\mathbf{X}}|_{\text{mel}}$ and outputs a latent representation, \mathbf{e} , which is the inferred context from $\tilde{\mathbf{x}}$. Employing similar STFT parameters to what is used for C , we yield the magnitude spectrum $|\tilde{\mathbf{X}}|$ and corresponding phase $\angle \tilde{\mathbf{X}}$ from $\tilde{\mathbf{x}}$. Then, D takes $|\tilde{\mathbf{X}}|$ and \mathbf{e} as inputs and outputs a contextual denoising mask with values in $[0, 1]$. The estimated clean audio spectrogram $|\hat{\mathbf{X}}|$ is obtained by masking as

$$|\hat{\mathbf{X}}| = |\tilde{\mathbf{X}}| \odot D(|\tilde{\mathbf{X}}|, \mathbf{e}). \quad (2)$$

where \odot is the element-wise product. We optimize D by minimizing the loss

$$\mathcal{L}_{\text{den}} = \mathbb{E}_{(\mathbf{x},\tilde{\mathbf{x}}) \sim \mathbb{D}} [l(\mathbf{x}, \hat{\mathbf{x}})] \quad (3)$$

where $\hat{\mathbf{x}}$ is the estimated clean audio reconstructed from $|\hat{\mathbf{X}}|$ and $\angle \tilde{\mathbf{X}}$, $l(\cdot)$ is a reconstruction loss (e.g., L_p -based loss or typical source separation loss like [SI-]SDR/SNR).

In the second setting, where C is finetuned during the training of D , the optimization minimizes the joint loss

$$\mathcal{L}_{\text{tot}} = \lambda_{\text{ASC}} \mathcal{L}_{\text{ASC}} + \lambda_{\text{den}} \mathcal{L}_{\text{den}}, \quad (4)$$

where λ_{ASC} and λ_{den} are weights for each corresponding loss.

3 Evaluation

3.1 Model architectures

In our model, C is a CRNN-based architecture consisting of three convolutional blocks with residual connections, followed by a recurrent neural network (RNN) and temporal attention pooling. Each convolutional block contains a

convolutional neural network (CNN) with a square kernel of 3×3 with 2×2 stride, doubling the channel dimension, and a 3×3 CNN with unit stride. The later CNN preserves the feature’s dimension, and a residual connection is employed between its input and output. The first block outputs 8 channels. The final CNN’s output is processed by an RNN with 128 hidden units, followed by temporal attention pooling and two fully connected layers of sizes 64 and the number of scene classes. An intermediate embedding is extracted after the first fully connected layer of C , serving as context e .

Our D is a UNet with skip-connections, following a 3-block deep encoder-decoder structure. Each encoder block consists of a CNN with a 3×3 kernel, doubling the number of channels at each successive layer. The initial encoder layer outputs 16 channels. The bottleneck layer employs a CNN with a 3×3 kernel and a transposed CNN (TrCNN) layer with the same kernel size. In the decoder path, following the concatenation of features from the skip connections, each decoder block utilizes a CNN with a 3×3 kernel and unit stride to halve the channel dimension and a TrCNN for upsampling. All CNNs and TrCNNs, apart the CNNs in the decoder, use a stride of 2×2 . The final output of the UNet is a mask, applied to $|\hat{X}|$ to obtain $|\hat{X}|$. To incorporate the context during the training of D , we leverage feature-wise linear modulation (FiLM) layers [23] with e at every layer within the encoder and decoder of D to allow modulating the feature maps at multiple scales.

3.2 Training setup

Audio signals are resampled to 22050 kHz to balance computational cost and retained signal information. We employ a STFT with window size of 1024 samples, 50% overlap, and a 64-band Mel filter bank over the 10-sec audio to extract the input features into model. For training, we used a batch size of 64 and Adam optimizer, with default hyperparameters and learning rate of 10^{-3} . SI-SNR loss is used to for the reconstruction loss. We use $\lambda_{ASC} = \lambda_{den} = 1$.

3.3 Baselines

We use a common UNet backbone for the denoising model D and perform an ablation of context utilization strategies. Specifically, we compare our our method, UNet_{ASC}, against three variants: i) without any context inference indicated as UNet, ii) with oracle context indicated as UNet_{oracle}, and iii) with separately provided uninformative context indicated as UNet_{const}. UNet model is a standard UNet architecture without FiLM layers and with no context inference from the context extractor C ; this serves as a naive baseline. To investigate the effect of conditioning on the oracle context (scene class), we employ UNet_{oracle} and feed the FiLM layers with the oracle one-hot encoded scene class. At UNet_{const}, uninformative context is provided through a constant vector of all ones as the conditioning vector. For UNet_{ASC}, there are two variants, UNet_{Fr-ASC} and UNet_{Tu-ASC}, depending on whether C is frozen or finetuned during denoising. To ensure a robust

Table 1: Summary of different experiment setups

Model	Context utilization
UNet	None
UNet _{ASC}	ASC embedding
UNet _{oracle}	Oracle scene class
UNet _{const}	Uninformative constant vector

Table 2: Mean/STD of evaluation metrics for the proposed ACAD method. *Embedding size* refers to the conditioning vector’s shape. *Embedding size I* is (batch, ASC emb size) and *Embedding size II* is (batch, num scene class)

Model	SI-SDR (dB)	SDR (dB)
Noisy input	4.27/0.00	4.26/0.00
UNet	10.16/0.02	10.56/0.02
<i>Embedding size I</i>		
UNet _{Tu-ASC}	12.12/0.04	12.56/0.04
UNet _{Fr-ASC}	11.04/0.07	11.47/0.09
UNet _{const}	10.02/0.01	10.41/0.03
<i>Embedding size II</i>		
UNet _{oracle}	10.82/0.02	11.23/0.03
UNet _{const}	10.13/0.03	10.53/0.05

comparison, we experiment with the UNet_{const} using different embedding sizes for the conditioning vector, matching those used for UNet_{ASC} (*Embedding size I*) and UNet_{oracle} (*Embedding size II*). Different experiment setups along with their conditioning inputs are summarized in Table 1.

4 Results and discussion

Table 2 presents the results of all experiments. We report the mean and standard deviation (STD) of the scale-invariant signal-to-distortion ratio (SI-SDR) and signal-to-distortion ratio (SDR) between the estimated signal and the clean ground truth obtained through five runs with identical experimental settings for each experiment. The result is calculated on the testing split of our dataset.

As shown in Table 2 UNet, serving as the lower bound reference for evaluation, yields 10.16 dB SI-SDR and 10.56 dB SDR. With oracle context, UNet_{oracle} achieves a 0.66 dB and 0.67 dB gain in SI-SDR and SDR over UNet. Learned context offers even larger improvements. A possible explanation is that the oracle context is constrained to scene-class information, whereas the vector-based representation learned from the noisy audio provides richer contextual information. UNet_{Fr-ASC} and UNet_{Tu-ASC} employs the same context extractor C from the first training stage, which obtains an accuracy of 84.18% in ASC on the testing set. Using the learned context, UNet_{Fr-ASC} observes increases of 0.88 dB in SI-SDR and 0.91 dB in SDR over UNet. Further finetuning C jointly with the denoising model D in UNet_{Tu-ASC} yields additional gains, outperforming UNet by 1.96 dB and 2 dB in SI-SDR and SDR, respectively. This suggests finetuning allows better alignment between the context extractor C and the denoiser D for denoising. In contrast, both UNet_{const} variants, with

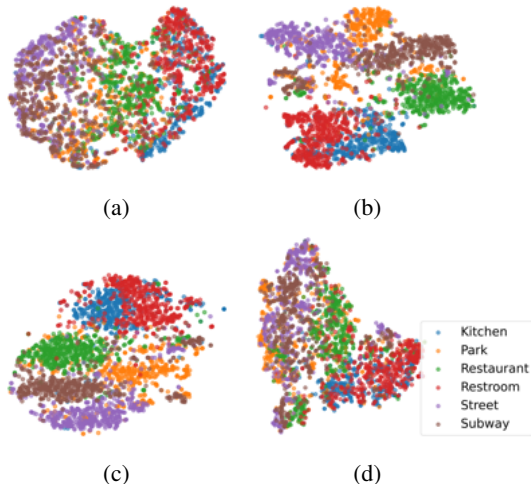


Figure 2: t-SNE representation of the bottleneck features from D of (a) UNet, (b) UNet_{Fr-ASC}, (c) UNet_{Tu-ASC}, and (d) UNet_{const} (*Embedding size I*)

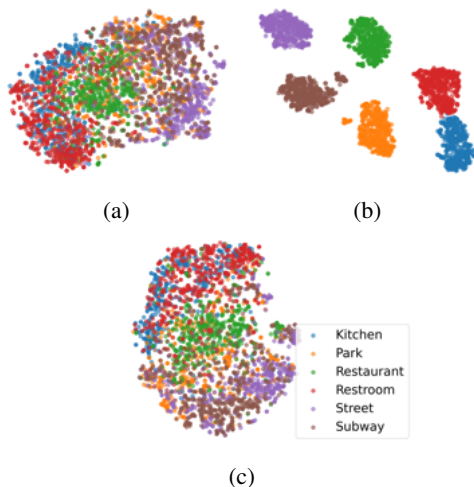


Figure 3: t-SNE representation of the bottleneck features from D of (a) UNet, (b) UNet_{oracle}, and (c) UNet_{const} (*Embedding size II*)

uninformative context, show inferior performance compared UNet. This indicates that uninformative information does not help or might even distract the denoising task. Although the unconditioned UNet in principle can internally infer the context during denoising, incorporating informative context, either via learned context (UNet_{ASC}) or through oracle context (UNet_{oracle}), shows consistent improvement in the denoising performance compared to the model without context inference.

As can be seen from Table 2, unconditioned UNet model obtained a decent performance, gaining 5.89 dB in SI-SDR compared to the noisy input. To explain this observation, we speculate that with the dataset where OC components are synthetically added to clean audio, the model may ex-

plot the acoustic statistical mismatch between the two sources, and preliminary experiments have verified this possibility. Given that this is observed in UNet, we acknowledge that UNet_{ASC} and UNet_{oracle} could likewise have the tendency to capture the acoustic mismatch rather than primarily relying on the context for denoising. However, the improvement in the performance of the denoising models when being explicitly introduced to informative context demonstrates that conditioning on informative context enhances the denoising process. Addressing this problem of acoustic statistical mismatch would require new data collection method and is a topic for further study.

The effect of context conditioning on the denoising model’s latent representations is observed in the t-SNE [24] visualizations of the bottleneck features from the denoiser D in all experiments. Figures 2 and 3 illustrate that, at comparable hierarchical levels, models conditioned with informative context (ASC context and Oracle context) achieve distinct clustering following the scene classes, showing that context information is retained in the model’s latent representations.

5 Conclusion

In this paper, we introduce the concept of **automatic audio contextual denoising** where target (what to be kept) and noise (what to be removed) depends on the learned context. For this first work, we define the context as information associated with an acoustic scene class. We propose a deep learning based method that automatically infers the context and suppresses out-of-context components relative to the learned context. Experiments are conducted on paired clean and noisy scene audio across different contexts, with out-of-context components as the noise to be removed. The proposed method is compared against three variants: without any context inference, with oracle context, and with separately provided uninformative context. Results show that learned context consistently achieves better result than the other variants, suggesting that the model can learn the context and context-dependent processing improve denoising performance. At the same time, we acknowledge that the current dataset, with synthetically added out-of-context components to real-life recorded clean audio signal, could guide the model to learn statistical mismatch cues instead of focusing on the context to do denoising. To mitigate this confound and isolate true contextual gains, further study and experiment are needed to investigate and address this issue.

References

- [1] William W. Gaver. What in the world do we hear? An ecological approach to auditory event perception. *Ecological Psychology*, 5(1):1–29, 1993.
- [2] Konstantinos Drossos et al. MaD TwinNet: Masker-denoiser architecture with twin networks for monaural sound source separation. In *International Joint Conference on Neural Networks (IJCNN)*, 2018.

- [3] Marc Aubeville et al. Deep denoising for hearing aid applications. In *International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 361–365. IEEE, 2018.
- [4] Ritwik Giri, Umut Isik, and Arvinth Krishnaswamy. Attention wave-u-net for speech enhancement. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2019.
- [5] Eloi Moliner and Vesa Välimäki. A two-stage u-net for high-fidelity denoising of historical recordings. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 841–845. IEEE, 2022.
- [6] Konstantinos Drossos, Mikko Heikkinen, and Paschalis Tsiaflakis. Lightweight DNN for full-band speech denoising on mobile devices: Exploiting long and short temporal patterns. In *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2025.
- [7] Kateřina Žmolíková et al. Speakerbeam: Speaker aware neural network for target speaker extraction in speech mixtures. *IEEE Journal of Selected Topics in Signal Processing*, 13(4):800–814, 2019.
- [8] Ke Zhang et al. Multi-level speaker representation for target speaker extraction. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.
- [9] Tsubasa Ochiai et al. Multimodal SpeakerBeam: Single channel target speech extraction with audio-visual speaker clues. In *INTERSPEECH*, 2019.
- [10] Triantafyllos Afouras, Joon Son Chung, and Andrew Zisserman. My lips are concealed: Audio-visual speech enhancement through obstructions. In *INTERSPEECH*, 2019.
- [11] Chenda Li and Yanmin Qian. Deep audio-visual speech separation with attention mechanism. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.
- [12] Chenda Li et al. Target sound extraction with variable cross-modality clues. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [13] Tuochao Chen et al. SoundSculpt: Direction and Semantics Driven Ambisonic Target Sound Extraction. In *INTERSPEECH*, 2025.
- [14] Xiang Hao et al. Typing to listen at the cocktail party: Text-guided target speaker extraction. *IEEE Transactions on Cognitive and Developmental Systems*, pages 1–12, 2025.
- [15] Noriyuki Tonami et al. Sound event detection guided by semantic contexts of scenes. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 801–805, 2022.
- [16] Golan Pundak et al. Deep context: End-to-end contextual speech recognition. In *IEEE Spoken Language Technology Workshop (SLT)*, pages 418–425, 2018.
- [17] Diep Luong et al. Automatic contextual audio denoising dataset. *Zenodo*, 2026.
- [18] Il-Young Jeong and Jeongsoo Park. CochScene: Acquisition of acoustic scene data using crowdsourcing. In *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2022.
- [19] Eduardo Fonseca et al. FSD50K: an open dataset of human-labeled sound events. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:829–852, 2021.
- [20] Qiuqiang Kong et al. PANNs: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894, 2020.
- [21] Jort F. Gemmeke et al. Audio Set: An ontology and human-labeled dataset for audio events. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.
- [22] Justin Salamon et al. Scaper: A library for soundscape synthesis and augmentation. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2017.
- [23] Ethan Perez et al. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.
- [24] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research (JMLR)*, 9(11), 2008.