The Title

Author

December 18, 2025

Abstract

In the acoustic event classification (AEC) framework that employs Blinkies, audio signals are converted into LED light emissions and subsequently captured by a single video camera. However, the 30 fps optical transmission channel conveys only about 0.2% of the normal audio bandwidth and is highly susceptible to noise. We propose a novel sound-to-light conversion method that leverages the encoder of a pre-trained autoencoder (AE) to distill compact, discriminative features from the recorded audio. To pre-train the AE, we adopt a noise-robust learning strategy in which artificial noise is injected into the encoder's latent representations during training, thereby enhancing the model's robustness against channel noise. The encoder architecture is specifically designed for the memory footprint of contemporary edge devices such as the Raspberry Pi 4. In a simulation experiment on the ESC-50 dataset under a stringent 15 Hz bandwidth constraint, the proposed method achieved higher macro-F_1 scores than conventional sound-to-light conversion approaches.

1 Introduction

1.1 Acoustic Event Classification

Acoustic event classification (AEC) is the task of estimating the posterior probabilities of multiple predefined event categories from an observed acoustic signal and outputs the most probable category. This technology is widely adopted in applications such as environmental monitoring, intelligent surveillance, and home automation. Significant progress in single-channel AEC has been achieved through the development of deep neural network (DNN)-based methods. In such methods, acoustic features like time-frequency representations are first extracted from raw audio signals and then provided as input to DNN models (e.g., CNNs and Transformers) to obtain class-posterior probabilities [1–3]. In addition, considerable efforts have been devoted to implementing AEC methods on edge devices [4, 5].

Besides spectral information from a single microphone, spatial information from a distributed array [6, 7] improves AEC. Placing several microphones near sound sources raises the signal-to-noise ratio, and the distinct inter-microphone delays preserve clues needed to separate simultaneously occurring events. Fully exploiting these spatial information demands sample-level synchronization across channels. Because wireless microphones are inherently asynchronous, recent work has focused on blind time alignment [8]. Wireless transmission also restrict throughput: at 16 kHz/16-bit resolution each sensor yields approximately 32 kB of data per second, so many microphones can conflict with the available communication bandwidth and storage resources.

1.2 Blinky, a sound-to-light converter

Addressing these issues, a sound-to-light conversion device named a Blinky has been developed [9–12]. These Blinkies can convert acoustic signals into varying intensities of light via an inbuilt light-emitting diode (LED). A video camera is employed to synchronously record LED brightness from multiple Blinkies spread over a wide region. Aggregating the Blinky signals from the recorded video, the fusion center, which is a high-performance server, performs AEC by integrating and analyzing acoustic information. Previous studies have validated its effectiveness in sound localization, speech enhancement, [10] and AEC tasks [13–15], demonstrating the feasibility of optical transmission of acoustic features.

However, the blinky remains constrained by severe bandwidth limitations imposed by the camera's frame rate, which is typically 30 frames per second (FPS), as well as by noise introduced during light-signal transmission through the air. This frame rate is considerably lower than the standard sampling rate of microphones, which is typically 16 kHz, and the bandwidth available for LED signals is approximately 1/533 that of audio signals. In an initial attempt to transmit sound signals via LED light signals, sound power was simply used as the basis for sound-to-light conversion; however, this approach is suboptimal for AEC due to restricted bandwidth and noise susceptibility. For this reason, Kinoshita et al. have proposed training a DNN for sound-to-light conversion in an end-to-end manner. Nevertheless, Kinoshita's method requires backpropagation of the loss of AEC through the unknown, non-differentiable physical light transmission channel, rendering it impractical for training the DNN on real-world environments. Therefore, a critical challenge arises: how to transmit semantically discriminative features through severely bandwidth-limited channels while preserving Blinky's practical deployment advantages.

1.2.1 Contributions

To overcome this challenge, in this paper, we propose a novel sound-to-light conversion method that does not require backpropagation through the physical transmission channel. For the proposed method, we first pre-train a lightweight autoencoder (AE) to encode audio

signals into compact latent vectors while preserving their essential information, where artificial noise is injected into the latent vectors during training to enhance robustness against noise. The encoder network of the pre-trained AE is then deployed on each Blinky, enabling the extraction of latent vectors from the recorded audio signals. The latent vectors are subsequently transmitted as light signals using Blinky's four LEDs. AEC is performed at a centralized fusion center, where a DNN classifier analyzes the LED signals captured by a camera.

We evaluated the efficacy of our AE-based sound-to-light conversion for acoustic event classification (AEC) through simulation studies utilizing the ESC-50 dataset. The experimental results demonstrate that our method outperforms both sound-power-based and end-to-end training approaches in terms of macro-F1 score, improving performance from 0.34 (sound-power-based) and 0.31 (end-to-end training) to 0.54.

Our primary contributions are as follows:

- We introduce a novel pre-trained AE-based sound-to-light conversion method, along with a noise-robust training strategy for AE pre-training, for acoustic sensing with Blinkies, enabling the transmission of discriminative features over noisy channels under the stringent bandwidth constraint of 15 Hz.
- We present a novel AE architecture whose encoder has a total inference-time memory footprint of approximately 3.40 MB, which is well within the capabilities of contemporary edge platforms such as the Raspberry Pi 4.
- We provide an open-source implementation of our simulation experiments, including both the proposed and conventional sound-to-light conversion methods.

2 Problem Setting

We address the challenge of implementing the AEC framework using Blinkies. This section formalizes the AEC pipeline and defines the core research problem.

2.1 Formalization of AEC Framework using Blinkies

The AEC procedure using N Blinkies consists of three sequential stages: on-device embedding (Embed$(\cdot)$), optical signal transmission (Transmit$(\cdot)$), and downstream classification (Classify$(\cdot)$). This pipeline is expressed as:

$$z_i = \text{Embed}(\boldsymbol{x}_i), \tag{1}$$

$$z'_i = \text{Transmit}(z_i), \tag{2}$$

$$\hat{y} = \text{Classify}(z'_1, \cdots, z_N). \tag{3}$$

As illustrated in Fig. 4, the i-th Blinky first acquires an acoustic signal $\boldsymbol{x}_i$. The embedding function Embed$(\cdot)$ on the Blinky maps the

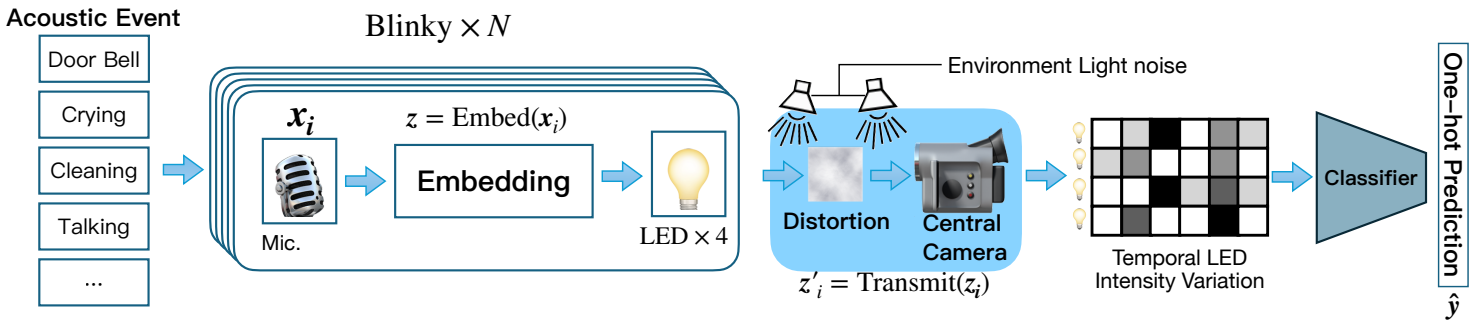

Figure 1: Diagram of AEC framework via blinky array.

captured signal to a feature vector $z_i \in \mathbb{R}^L$, which subsequently modulates the device's LEDs. The resulting light signal propagates through the environment and is captured by a central camera; the composite process is abstracted by the function Transmit$(\cdot)$. Finally, the classifier Classify$(\cdot)$ operates on the collection of received signals $\{z'_i\}$ to infer the acoustic event label $\hat{y}$.

Within the transmission channel, Blinkies' LED signals are perturbed by ambient illumination and noise before being temporally resampled by the camera. Accordingly, we model the channel as the composition of a distortion operator followed by camera resampling, i.e., Transmit$(\cdot)$ = (Resample $\circ$ Distort$(\cdot)$). The distortion is parameterized as follows:

$$\text{Distort}(z) = az + b\mathbf{1} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I}), \tag{4}$$

where a denotes the attenuation coefficient—inversely proportional to the squared distance between the camera and a Blinky—and b captures the bias introduced by ambient light, whereas $\boldsymbol{\epsilon}$ represents additive white Gaussian noise with covariance $\sigma^2\boldsymbol{I}$. The vector $\mathbf{1}$ is the L-dimensional all-ones vector and the matrix $\boldsymbol{I}$ is the identity matrix with a size of $L \times L$. The resampling operator Resample : $\mathbb{R}^L \to \mathbb{R}^{N_{\text{LED}} \times \lfloor F_s \times T \rfloor}$ encapsulates the camera's temporal sampling, governed by its frame rate F_s, the number of LEDs per device N_{LED}, and the audio segment duration T. To respect the sensor's dynamic range, resampled values are clipped to the interval $[0, 1]$.

2.2 Challenge: Designing Embedding Function

Within this framework, the central challenge is to design an embedding function Embed$(\cdot)$ that retains as much acoustic information as possible despite stringent bandwidth constraints and pronounced channel noise. To date, two principal strategies have been explored.

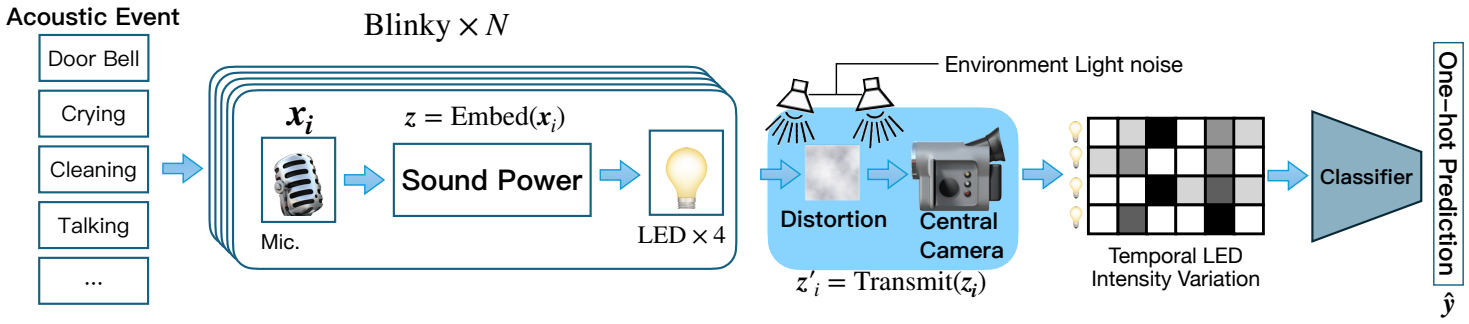

Figure 2: Diagram of sound power approach.

Sound Power As shown in 2, the simplest method defines Embed$(\cdot)$ by mapping the raw waveform $\boldsymbol{x}(t)$ to its power: $z = (x(1)^2, x(2)^2, \cdots, x(T)^2)^\top$ Although computationally negligible, the downsampling process in the camera discards most of the semantic information (e.g., timbre), resulting in insufficient performance on AEC tasks.

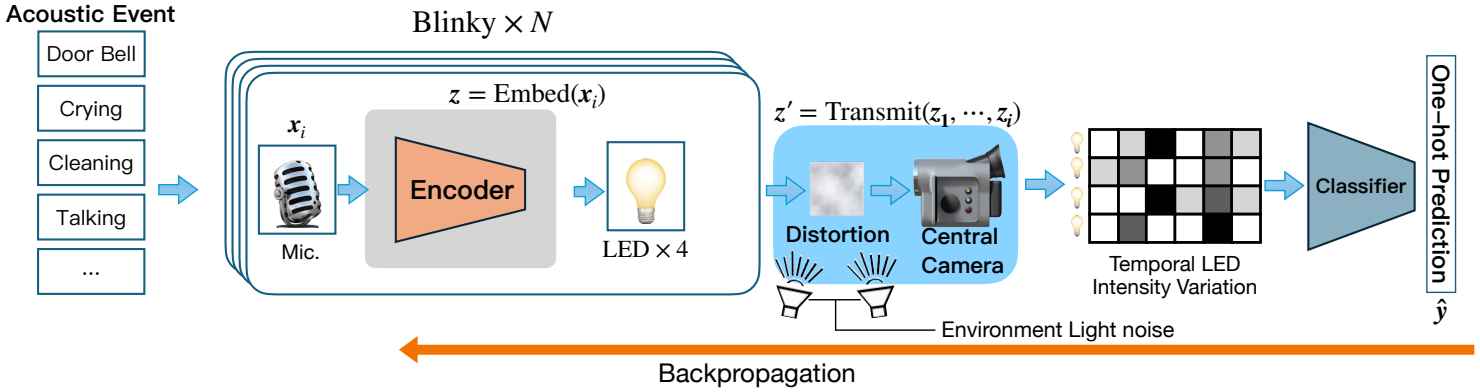

Figure 3: Diagram of End-to-End approach proposed by Kinoshita et al.

End-to-End Training Approach As shown in 3, to convey more discriminative features, Kinoshita et al. [13] proposed using a DNN encoder for Embed$(\cdot)$ and optimized both Embed$(\cdot)$ and Classify$(\cdot)$ (Eqs. 1 and 3) in an end-to-end fashion. However, this strategy is impractical in real-world deployments: the transmit channel (Eq. 2) constitutes an unknown, non-differentiable physical process, thereby blocking gradient flow back to the on-device encoder.

Both approaches therefore exhibit critical shortcomings: the sound-power approach is too rudimentary for high-accuracy classification, whereas the end-to-end training approach is physically unrealizable. Consequently, a practical solution is required—one that permits the transmission of expressive, learned features without necessitating differentiation through the physical channel.

3 Methodology

To overcome the limitations described in the previous section, we propose a practical solution that decouples the feature embedding from the non-differentiable physical channel. Our approach, illustrated in Fig. 4, realizes the embedding function Embed$(\cdot)$ via the encoder of a pre-trained autoencoder (AE), denoted as $E_{\phi^*}(\cdot)$.

The procedure begins by transforming the raw input waveform $\boldsymbol{x}$ into a log-Mel spectrogram $\boldsymbol{X} = \text{Logmel}(\boldsymbol{x})$, a widely adopted time-frequency representation of audio. The embedding function then maps the waveform to a latent vector z through the log-Mel representation:

$$z = \text{Embed}(\boldsymbol{x}) = E_{\phi^*}(\boldsymbol{X}).$$

By pre-training the encoder, we obtain a compact yet discriminative representation that is suited for transmission, thereby obviating the need for end-to-end optimization through the physical channel. This section details the AE's learning objective, its network architecture,

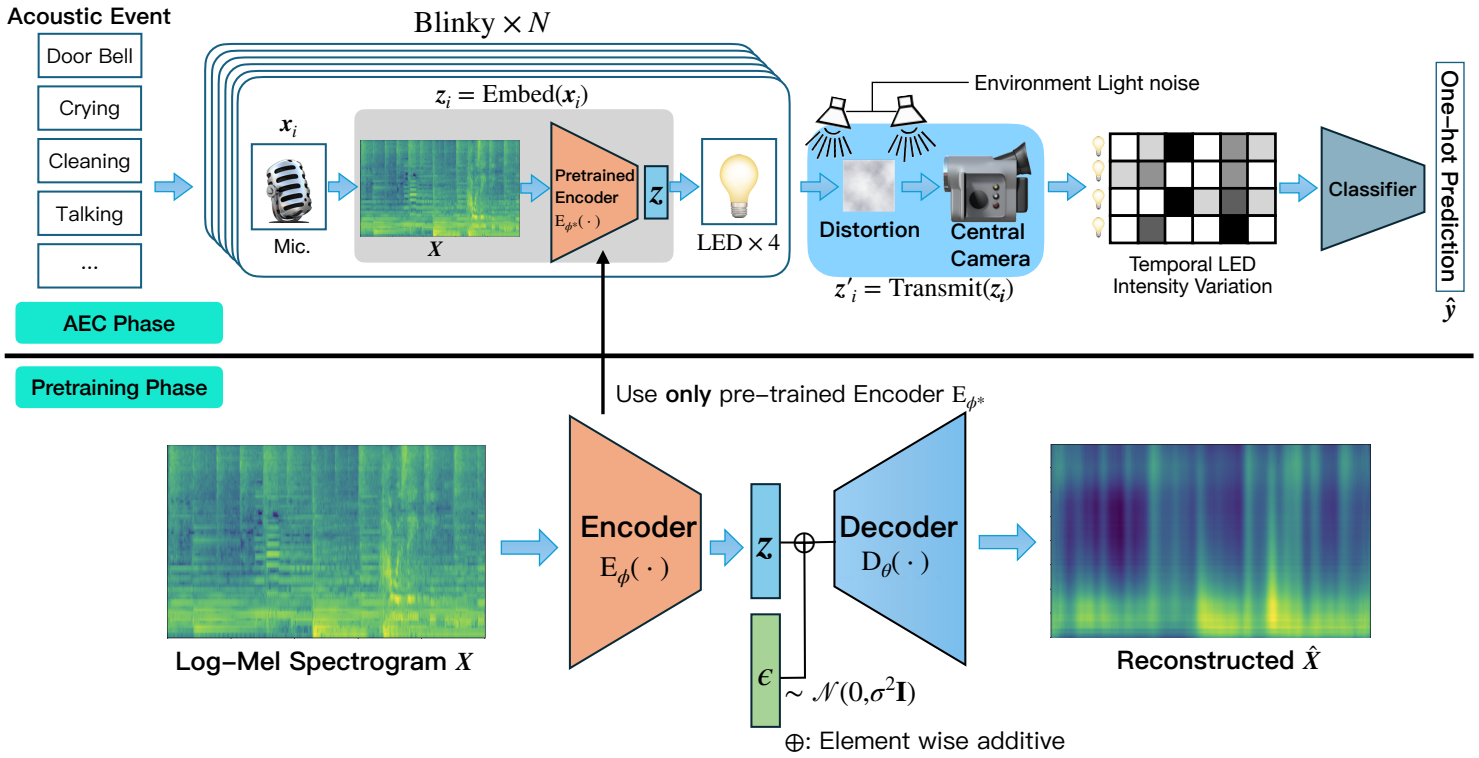

Figure 4: Overview of the proposed framework. **Pre-training phase** (bottom): An autoencoder (AE) is pre-trained to learn noise-robust representations; Gaussian noise ($\boldsymbol{\epsilon}$) is injected into the latent vector (z) and the decoder (D_θ) reconstructs the original log-Mel spectrogram ($\boldsymbol{X}$). **AEC phase** (top): At inference, the frozen encoder (E^*_ϕ) produces a compact latent representation that modulates four LEDs. A central camera records the signal as *temporal LED intensity variations*, and the downstream classifier predicts the event label ($\hat{\boldsymbol{y}}$).

pre-training procedure, and training procedure for the downstream classifier. [1]

3.1 Learning Objective

The primary objective of an AE is to distill a compact representation from a large, unlabeled audio dataset (e.g. Google AudioSet [16]) by means of a self-supervised reconstruction task. A standard AE minimizes the reconstruction error between an input log-Mel spectrogram $\boldsymbol{X}$ and its reconstruction $\hat{\boldsymbol{X}}$ via

$$\mathcal{L}_{\text{rec}} = \mathbb{E}\left[||\boldsymbol{X} - D_\theta(E_\phi(\boldsymbol{X}))||_F^2\right], \tag{5}$$

where $E_\phi(\cdot)$ and $D_\theta(\cdot)$ represent the encoder and decoder, respectively; the optimized parameters after pre-training are (ϕ^*, θ^*).

However, this objective does not consider the perturbation of the latent vector $E_\phi(\boldsymbol{X})$ by channel noise. To address this issue, we adopt a noise-robust training strategy: Gaussian noise $\boldsymbol{\epsilon}$ is injected into the latent vectors during training to mimic channel distortion and compel the model to produce a more resilient representation. The revised objective is

$$\mathcal{L}_{\text{rec-robust}} = \mathbb{E}\left[||\boldsymbol{X} - D_\theta(E_\phi(\boldsymbol{X}) + \boldsymbol{\epsilon})||_F^2\right], \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, \sigma^2\mathbf{I}), \tag{6}$$

thereby encouraging $E_\phi(\cdot)$ to produce channel-invariant and discriminative features.

3.2 AE Architecture

The Encoder architecture, summarized in Table 1, is a hybrid design that combines convolutional neural networks (CNNs) and self-attention. A 1-D convolutional layer first extracts local temporal patterns, after which a multi-head self-attention layer captures long-range dependencies, achieving a global receptive field without excessive network depth. A series of strided convolutional layers performs temporal downsampling before projecting the signal into an L-dimensional latent space. We choose L to match the physical channel capacity, i.e., $L = 15\,\text{symbol/s} \times 4\,\text{LEDs} \times 5\text{s} = 300$, where 15 Hz is the per-LED Nyquist rate imposed by the 30 FPS camera. Unlike audio transformer models [3], which prioritize accuracy without reducing bitrate, our AE explicitly compresses the representation to meet the optical-channel capacity.

A crucial deployment constraint is the encoder's computational footprint on resource-constrained hardware. Our encoder contains 517,508 trainable parameters, occupying roughly 2.04 MB of memory, and its total inference-time footprint is about 3.40 MB. This is comfortably within the capacity of modern edge platforms such as the Raspberry Pi 4.

[1] All training scripts, including Autoencoder pre-training and downstream evaluation, are available at `https://github.com/ykinolab-tokai/multi-ae-ace`.

Table 1: Autoencoder architecture. Output shapes are listed as frequency $F \times$ time T. Self-Attention block consists of multi-head attention layer with two heads, layer normalization, feed-forward network (FFN) that expands feature dimension from 40 to 128 and then projects it back, and layer normalization with residuals. k, s, and c denote kernel size, stride, and channel, respectively.

Layer	**Output Shape** $(F \times T)$
Input: Log-Mel Spectrogram	[80, 501]
Encoder (E_ϕ)	
Conv1d (c=40, k=5, s=1), ReLU	[40, 497]
Self-Attention (embed=40, heads=2)*	[40, 497]
ReLU	[40, 497]
Conv1d (c=20, k=7, s=2), ReLU	[20, 246]
Conv1d (c=20, k=11, s=3), ReLU	[20, 79]
Flatten	[1, 1580]
Linear (in_features=1580, out_features=300)	[1, 300]
Decoder (D_θ)	
Linear (in_features=300, out_features=1580)	[1, 1580]
Reshape	[20, 79]
ConvTranspose1d (c=20, k=11, s=3), ReLU	[20, 245]
ConvTranspose1d (c=40, k=7, s=2), ReLU	[40, 495]
ConvTranspose1d (c=80, k=5, s=1)	[80, 499]
Interpolate (target_time_dim=501)	[80, 501]
Output: Reconstructed Log-Mel	[80, 501]

3.3 Pre-training

We aim to infer a single acoustic-event label y from a set of 5-second audio waveform $\{\boldsymbol{x}_i\}$ sampled at 16 kHz. Each 10-second audio clip from the Google AudioSet [16] is first down-mixed to monaural and resampled to 16 kHz. For data augmentation, a 5-second segment is then randomly cropped from every clip. A log-Mel spectrogram $\boldsymbol{X}$ is then computed from this segment, using a Hann window with a length of 400, a 160-sample hop length, and 80 Mel filters.

The AE is pre-trained for 300 epochs on the balanced subset of AudioSet with the Adam optimizer [17], an initial learning rate of 1×10^{-3} and a batch size of 1280. A `ReduceLROnPlateau` learning rate scheduler lowers the learning rate by a factor of 0.1 whenever the validation loss fails to improve for 10 consecutive epochs.

3.4 Training Downstream Classifier

To train the downstream classifier that predicts acoustic event labels from the transmitted data, we first record the Blinky signals corresponding to each input audio instance in the training set with the central camera, thereby obtaining a set of latent vectors $\{\boldsymbol{z}'_i\}$ (see Eqs. 1 and 2). The classifier is subsequently trained to infer the target labels y from these latent representations, whereas the parameters of the pre-trained encoder E_{ϕ^*} are kept fixed throughout this phase.

4 Performance Evaluation

We evaluated the efficacy of our AE − based sound-to-light conversion for AEC through a simulation study. The experimental design rests on a straightforward premise: embedding functions that generate more discriminative features should yield latent representations $\boldsymbol{z}$ that result in a superior classification performance (e.g. macro F1-score).

4.1 Experimental Setup

The experiments were conducted in a simulated rectangular room measuring $8 \times 6 \times 4\,\mathrm{m}^3$, where five Blinkies were deployed. The process of estimating the class label $\hat{y}$ from the waveform $\boldsymbol{x}_i$ received by the i-th Blinky follows the procedure outlined in Eqs. 1 to 3. AEC performance was measured using the F_1 score and model selection relied on the highest F_1 score obtained on a validation set.

To assess the contribution of our AE-based sound-to-light conversion, we compared the following configurations for Embed$(\boldsymbol{x}_i)$:

- **Log-Mel spectrogram without embedding**: Converts the raw waveform $\boldsymbol{x}$ into a log-Mel spectrogram $\boldsymbol{X}$, with no further embedding, i.e., Embed$(\boldsymbol{x}) = \boldsymbol{X}$. This evaluation is provided as a reference for the ideal classification performance on the same data, without the Blinky channel constraints.
- **Sound Power**: Computes $\boldsymbol{z}$ as the signal power, i.e., Embed$(\boldsymbol{x}) = (x(1)^2, \cdots, x(T)^2)^\top$, following the approach in [9].

- **End-to-End**: Computes $\boldsymbol{z}$ by using a DNN encoder $E_\phi(\cdot)$, i.e., $\mathrm{Embed}(\boldsymbol{x}) = E_\phi(\boldsymbol{X})$. In this method, both the encoder and classifier are jointly optimized in an end-to-end fashion using the classification loss. The encoder architecture is identical to that of our pre-trained encoder E_ϕ.
- **Autoencoder**: Computes $\boldsymbol{z}$ as the latent representation produced by an autoencoder pre-trained with Eq. 5, i.e., $\mathrm{Embed}(\boldsymbol{x}) = E_{\phi^*}(\mathrm{Logmel}(\boldsymbol{x}))$.
- **Noise-robust autoencoder (Ours)**: Computes $\boldsymbol{z}$ as the latent representation from our noise-aware autoencoder pre-trained with Eq. 6, i.e., $\mathrm{Embed}(\boldsymbol{x}) = E_{\phi^*}(\mathrm{Logmel}(\boldsymbol{x}))$.

We utilized the ESC-50 dataset [18], partitioned into training, validation, and test sets with an 8:1:1 ratio on a per-class basis to ensure class-wise balance across all subsets. The acoustic waveform $\boldsymbol{x}_i$ received at the i-th Blinky was generated by convolving a source signal s with the corresponding room impulse response (RIR) between that source and the Blinky. To simulate class-dependent spatial characteristics, a single sound source was randomly positioned for each class, with the k-th source assigned to the k-th class. Source and Blinky positions remained fixed throughout the experiment. RIRs were generated using the image-source method implemented in the *pyroomacoustics* [19], with a wall-absorption coefficient of 0.4 and a maximum image source order of 10, then zero-padded to equal length. Parameter a, b, and σ of Eq. 4 were 1, 0.1, and 0.05, respectively.

The classifier $\mathrm{Classify}(\cdot)$ was implemented as a ResNet-18 [20]. During training the classifier, we applied BC-learning [21] as a data augmentation, wherein two samples from different classes and their one-hot labels were linearly combined to produce mixed inputs and soft targets. The classifier was trained using the Kullback − Leibler (KL) divergence as the loss function and Adam optimizer [17]. As in the pre-training phase, we used the `ReduceLROnPlateau` scheduler in the classifier training phase, initializing the learning rate of 1×10^{-2}. The scheduler monitored the validation loss and reduced the learning rate by a factor of 0.5 whenever the validation loss plateaued.

This simulation experiment was implemented using the *PyTorch* framework [22].

4.2 Results

Table 2: Macro-averaged F_1 score (mean ± std. across four independent trials) on test set under three transmission conditions: no degradation $\mathrm{Transmit}(\boldsymbol{z}) = \boldsymbol{z}$ (*Ideal channel*), resampling $\mathrm{Transmit}(\boldsymbol{z}) = \mathrm{Resample}(\boldsymbol{z})$ (*Resample*), and resampling with distortions $\mathrm{Transmit}(\boldsymbol{z}) = (\mathrm{Resample} \circ \mathrm{Distort})(\boldsymbol{z})$ (*Resample + Distort*). Definitions of each embedding method are provided in Section 4.1.

Embedding method	**Ideal channel**	**Resample**	**Resample + Distort**
Log-Mel spectrogram without embedding	0.9949	—	—
Sound Power	—	0.3976 ± 0.0189	0.3359 ± 0.0238
End-to-End	—	0.4965 ± 0.0731	0.3077 ± 0.0705
Autoencoder	0.6842 ± 0.0448	0.6696 ± 0.0109	0.5227 ± 0.0206
Noise-Robust Autoencoder (Ours)	**0.7248 ± 0.0405**	**0.7279 ± 0.0201**	**0.5357 ± 0.0178**
ESC-50 Human Benchmark		0.8130	

Table 2 presents macro-averaged F_1 scores on the test set under three transmission scenarios: (i) no degradation, $\mathrm{Transmit}(\boldsymbol{z}) = \boldsymbol{z}$; (ii) resampling only, $\mathrm{Transmit}(\boldsymbol{z}) = \mathrm{Resample}(\boldsymbol{z})$; and (iii) resampling followed by optical distortions, $\mathrm{Transmit}(\boldsymbol{z}) = (\mathrm{Resample} \circ \mathrm{Distort})(\boldsymbol{z})$. Except for the log-Mel spectrogram without embedding, each method was evaluated four times with different random seeds, and the mean and standard deviation across these four runs are reported in Table 2.

Across all conditions, our AE-based method substantially outperforms both the sound-power and the end-to-end approaches. Moreover, our noise-robust autoencoder achieves higher F_1 scores than the standard autoencoder, highlighting the effectiveness of the noise-aware training objective described in Eq. 6.

4.3 Discussion and Limitations

Through simulation experiments, we demonstrated the efficacy of the proposed noise-robust autoencoder for sound-to-light conversion. Nevertheless, because these experiments were purely virtual, its performance under real-world conditions remains unverified. Our immediate objective is therefore to implement the framework in a physical environment and conduct a comprehensive evaluation of its AEC accuracy. This assessment will also quantify the computational overhead and determine the feasibility of real-time processing on embedded hardware.

Moreover, the present study does not incorporate the spatial characteristics inherent in audio signals during AE training. Consequently, the latent vectors may lack room-impulse response (RIR) information contained in the original signal, potentially diminishing the spatial cues available for AEC. Designing a loss function that explicitly preserves spatial information could yield richer embeddings and further improve performance.

Another limitation of this study is that our evaluation focuses exclusively on single-event clips and does not address scenarios in which two or more acoustic events occur simultaneously. This work primarily emphasizes validating whether the proposed method functions as intended, rather than exhaustively delineating its operational boundaries; therefore, multi-event classification is deferred to future investigations. Nevertheless, since the downstream classifier is trained with BC learning—thereby being exposed to mixtures and soft labels —the resulting decision function is capable of handling mixed inputs to some extent, indicating a promising potential for the recognition of concurrent events in more complex environments.

It is further assumed that each Blinky transforms its recorded audio into a low-dimensional vector within a shared 5-second time window, and it is not assumed that recordings occur over differing temporal intervals across devices. However, this assumption may not hold in practical deployments, where device-specific asynchrony can arise. Consequently, the design of robust synchronization mechanisms and the development of training or inference strategies resilient to temporal misalignment represent important avenues for future research.

In addition to temporal considerations, there is a limitation regarding the spatial dimension of the input data. Currently, the training data is down-mixed to mono-channel by calculating the sample mean of the channels. However, given that each Blinky device is equipped with two fixed-position microphones designed to capture spatial cues, this down-mixing process discards potentially valuable spatial information. Future iterations will aim to fully leverage these on-device spatial characteristics.

Finally, although we employed an autoencoder for unsupervised representation learning, alternative methods—such as generative approaches (e.g., variational autoencoders) and contrastive learning objectives—warrant investigation.

5 Visualization of Latent Representations

While in 4 demonstrates the superior classification performance of the proposed method under limited bandwidth, the internal mechanism of the autoencoder remains a "black box." Specifically, it is unclear what features the encoder prioritizes when compressing audio into such a compact latent space (15 Hz) without semantic supervision. This section investigates the intrinsic structure of the latent space through visualization and quantitative acoustic verification.

5.1 Global Latent Geometry Visualization

To intuitively understand the topological structure of the learned representations, we visualized the latent spaces generated by both the normal and noise-robust encoders. We performed inference on the complete ESC-50 dataset ($N = 2000$) to extract high-dimensional latent variables, resulting in a data matrix $\mathbf{Z} \in \mathbb{R}^{2000\times300}$.

Dimensionality reduction techniques, specifically Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE), were employed to project the 300-dimensional latent vectors onto a 2D plane. The visualization results are presented in Fig. 5, Fig. 6, Fig. 7 and Fig. 8.

5.2 Symmetry and Structural Isomorphism

A notable observation from the low-dimensional projections (both PCA and t-SNE) is the apparent symmetry between the manifolds of the normal and noise-robust encoders. Although dimensionality reduction algorithms often possess inherent sign ambiguities (e.g., the arbitrary sign of eigenvectors in PCA), the consistent symmetric patterns motivated us to investigate whether this geometric relationship holds in the native 300-dimensional space.

To quantitatively verify this hypothesis, we conducted a geometric alignment analysis between the latent variables produced by the normal encoder ($\mathbf{Z}_{norm}$) and the noise-robust encoder ($\mathbf{Z}_{rob}$).

First, we calculated the sample-peer cosine similarity between the paired latent vectors. The average cosine similarity reached **0.97**, indicating that the two encoders, despite being trained with different objectives, have learned highly similar semantic manifolds.

Furthermore, to explicitly determine the geometric transformation between the two spaces, we formulated the alignment as an Orthogonal Procrustes [23] problem. We sought an orthogonal matrix $\mathbf{Q}$ that aligns $\mathbf{Z}_{norm}$ to $\mathbf{Z}_{rob}$ by minimizing the Frobenius norm:

$$\min_{\mathbf{Q}\in\mathbb{R}^{300\times300}} ||\mathbf{Z}_{rob}\mathbf{Q} - \mathbf{Z}_{norm}||_F, \quad \text{s.t. } \mathbf{Q}^T\mathbf{Q} = \mathbf{I}$$

The determinant of the optimal rotation matrix was calculated to be $\det(\mathbf{Q}) \approx 1$.

5.3 Discussion

This result ($\det(\mathbf{Q}) \approx 1$) implied that $\mathbf{Z}_{norm}$ and $\mathbf{Z}_{rob}$ can be aligned almost perfectly through a simple proper rotation. This suggests that the injection of Gaussian noise into the latent variables during training does not fundamentally alter the topology of the learned manifold. The average sample-wise cosine similarity between paired samples, however, is not exactly 1, and the residual reconstruction error (Frobenius norm) is non-zero (about 0.23), which we interpret as small but systematic geometric deviations between the two manifolds.

Given the superior classification performance demonstrated in Section 4, we **hypothesize** that while the global geometric structure remains largely invariant, these minute deviations represent a refinement of the feature space. It is precisely these subtle structural adjustments that equip the Autoencoder with its observed resilience to channel noise.

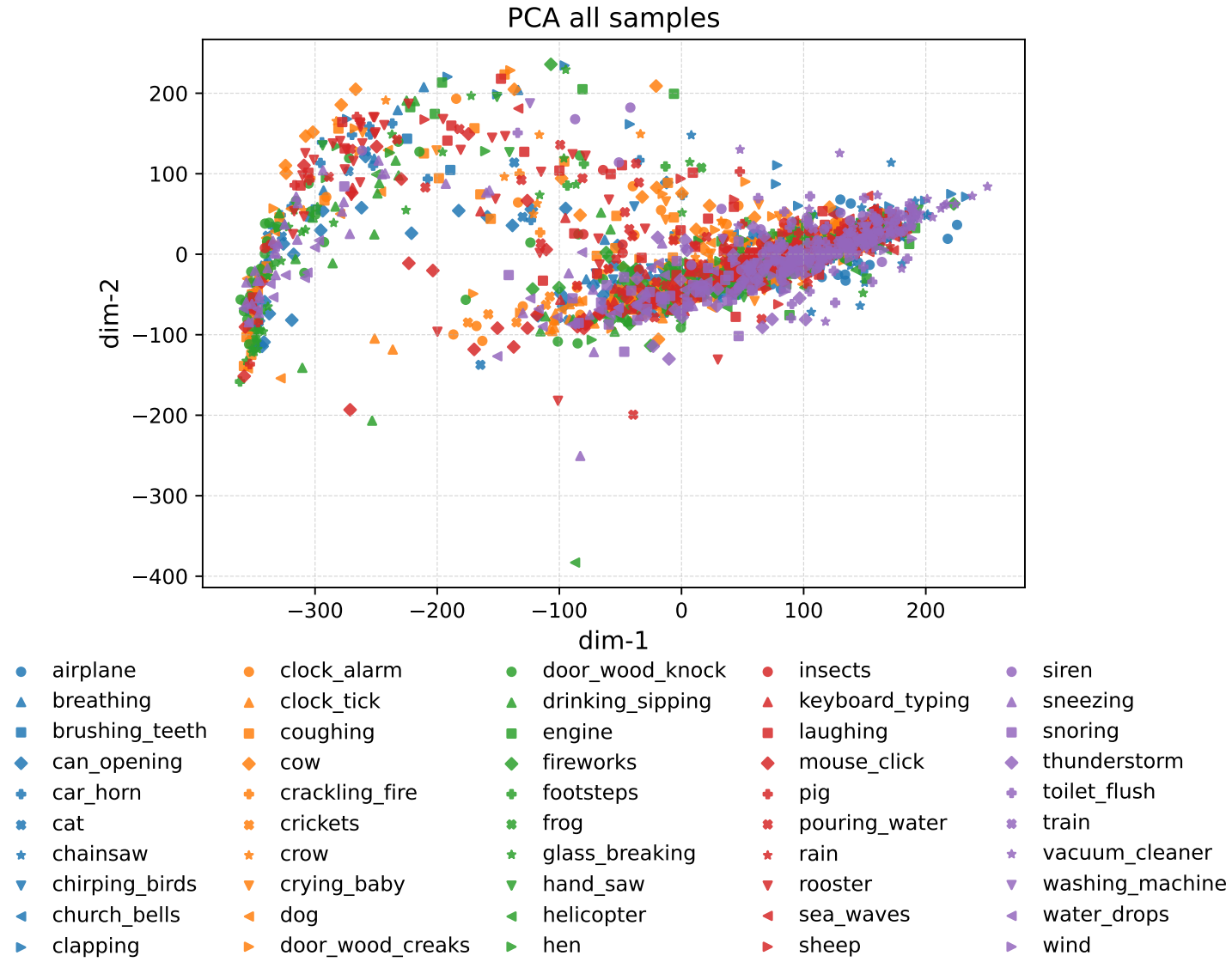

Figure 5: Visualization of the latent space distributions of noise-robust autoencoder. Dimensionality reduction algorithms: PCA.

6 Conclusion

In this paper, we propose a practical sound-to-light conversion method for AEC using Blinkies. Conventional methods either suffer from low accuracy or face difficulties in real-world deployment due to non-differentiable physical channels. To overcome this, we separate the training of the feature embedding function from the physical channel and use a pre-trained autoencoder (AE) encoder for the embedding. For pre-training AE, a noise-robust training strategy is introduced to enhance robustness against distortion in the transmission channel. Simulation results show that the proposed AE-based method consistently outperforms conventional approaches across various conditions, achieving higher F_1 scores.

Despite promising results in simulations, real-world evaluation remains a future task. Additionally, incorporating spatial information into the training objective and exploring other self-supervised methods like VAE or contrastive learning are potential future directions.

Additionally, incorporating spatial information into the training objective and exploring contrastive learning [24–26] are potential future directions, as contrastive objectives could enforce explicit class-wise clustering to enhance the semantic discriminability beyond the physical temporal structures captured by the current reconstruction loss.

References

[1] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition," 2020.

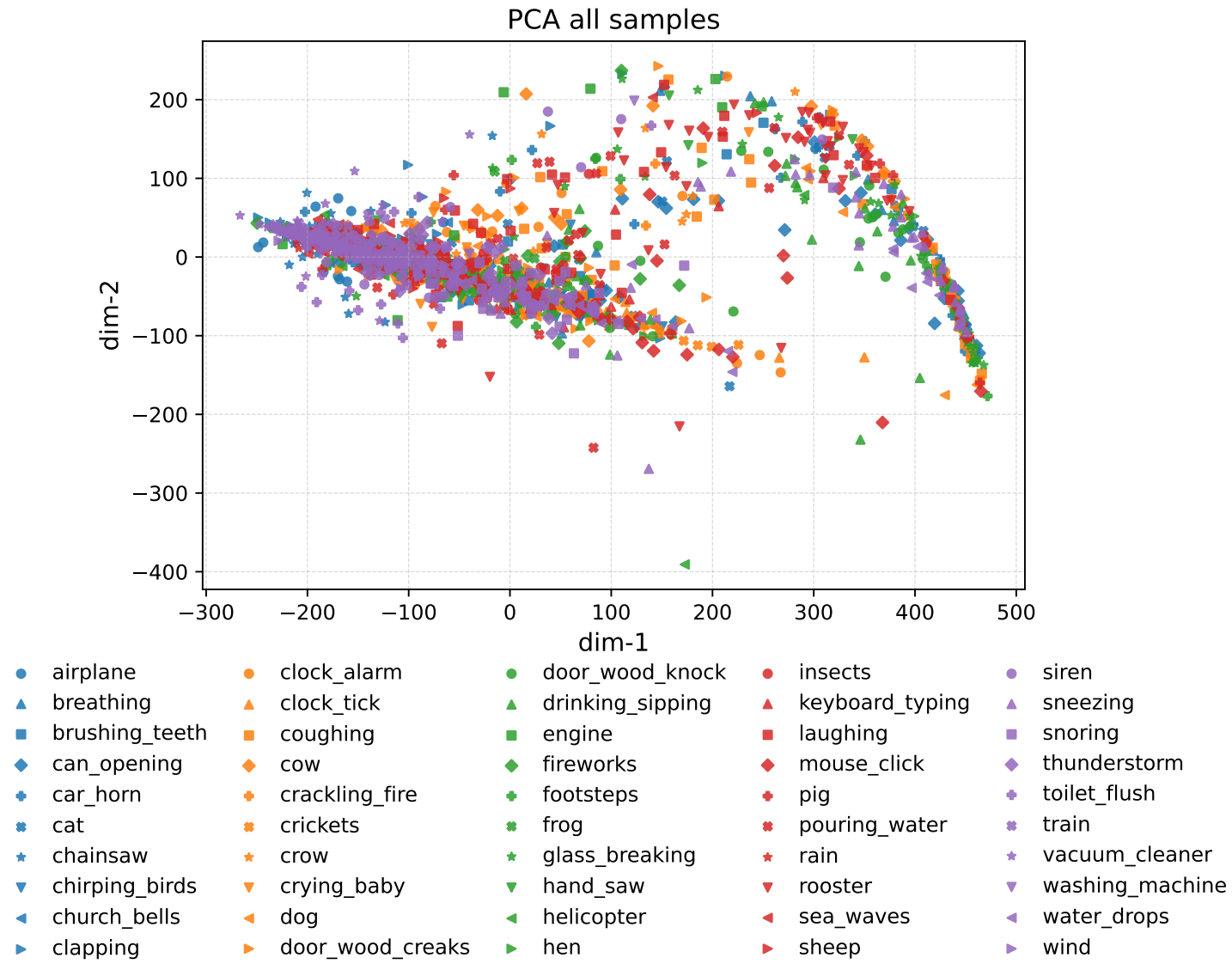

Figure 6: Visualization of the latent space distributions of normal autoencoder. Dimensionality reduction algorithms: PCA.

[2] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "HTS-AT: A Hierarchical Token-Semantic Audio Transformer for Sound Classification and Detection," 2022.

[3] Y. Gong, Y.-A. Chung, and J. Glass, "Ast: Audio spectrogram transformer," 2021.

[4] Z. Huang, A. Tousnakhoff, P. Kozyr, R. Rehausen, F. Bießmann, R. Lachlan, C. Adjih, and E. Baccelli, "TinyChirp: Bird Song Recognition Using TinyML Models on Low-power Wireless Acoustic Sensors," 2024.

[5] M. Mohaimenuzzaman, C. Bergmeir, I. West, and B. Meyer, "Environmental Sound Classi fi cation on the Edge: A Pipeline for Deep Acoustic Networks on Extremely Resource-Constrained Devices," *Pattern Recognition*, vol. 133, p. 109025, 2023.

[6] D. Zhang, J. Chen, J. Bai, and M. Wang, "Sound event localization and classification using WASN in Outdoor Environment," 2024.

[7] D. Stowell, D. Giannoulis, E. Benetos, M. Lagrange, and M. D. Plumbley, "Detection and Classification of Acoustic Scenes and Events," *IEEE Transactions on Multimedia*, vol. 17, no. 10, pp. 1733–1746, 2015.

[8] D. Cherkassky and S. Gannot, "Blind synchronization in wireless acoustic sensor networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 3, pp. 651–661, 2017.

[9] R. Scheibler and N. Ono, "Blinkies: Open Source Sound-to-Light Conversion Sensors for Large-Scale Acoustic Sensing and Applications," *IEEE Access*, vol. 8, pp. 67 603–67 616, 2020.

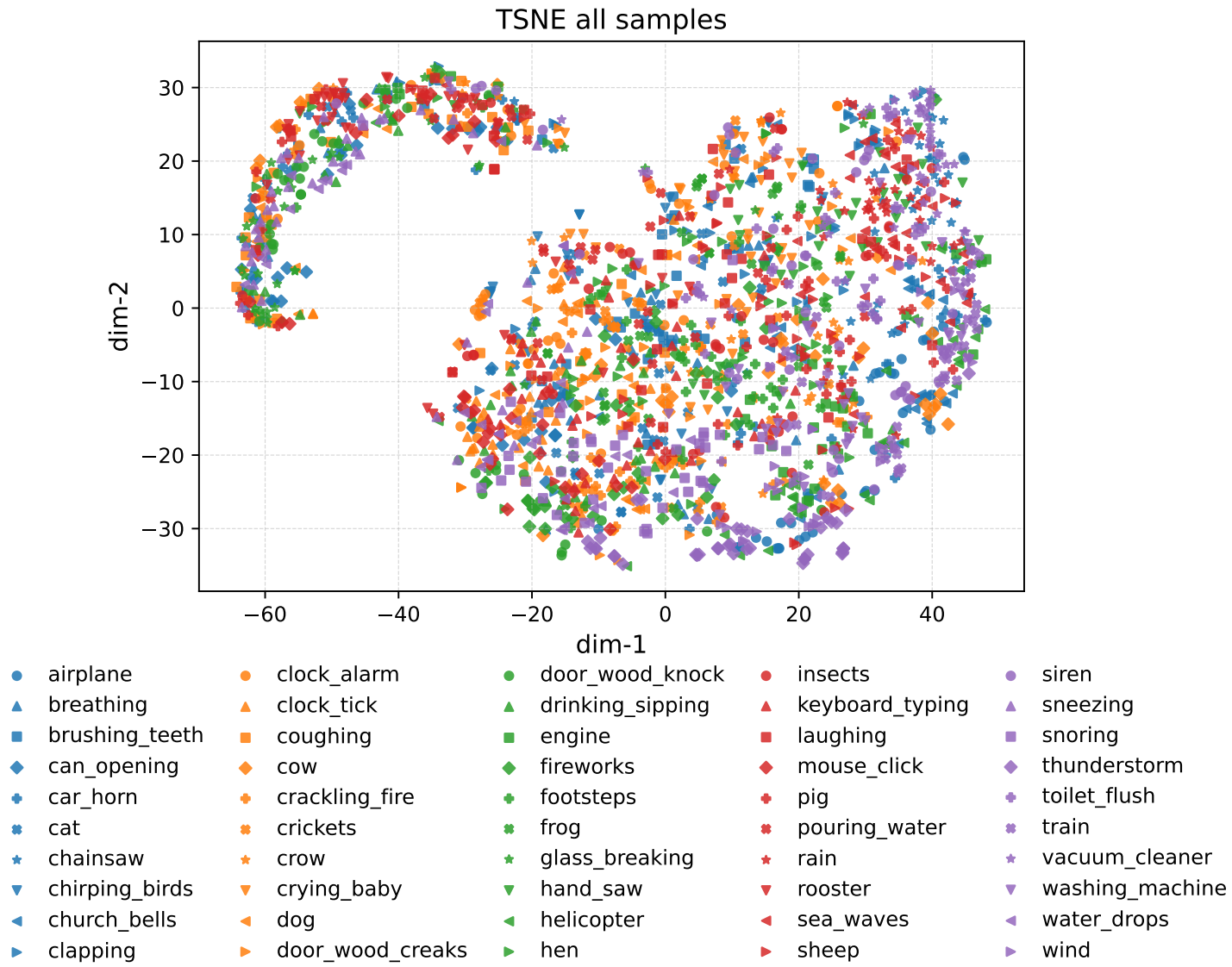

Figure 7: Visualization of the latent space distributions of noise-robust autoencoder. Dimensionality reduction algorithms: T-SNE.

[10] R. Scheibler, D. Horiike, and N. Ono, "Blinkies: Sound-to-light conversion sensors and their application to speech enhancement and sound source localization," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2018, pp. 1899–1904.

[11] K. Ishii, Y. Kinoshita, Y. Wakabayashi, and N. Ono, "Real-Time Pitch Visualization with "Blinky" Sound-to-Light Conversion Device," *Journal of Signal Processing*, vol. 25, no. 6, pp. 213–220, 2021.

[12] K. Nishida, N. Ueno, Y. Kinoshita, and N. Ono, "Estimation of Transfer Coefficients and Signals of Sound-to-Light Conversion Device Blinky Under Saturation," in *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. Chiang Mai, Thailand: IEEE, Nov. 2022, pp. 717–722.

[13] Y. Kinoshita and N. Ono, "End-to-End Training for Acoustic Scene Analysis with Distributed Sound-to-Light Conversion Devices," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 1010–1014.

[14] ——, "End-to-end training of acoustic scene classification using distributed sound-to-light conversion devices: verification through simulation experiments," *EURASIP J. Audio Speech Music Process.*, vol. 2024, no. 1, Sep. 2024. [Online]. Available: https://doi.org/10.1186/s13636-024-00369-z

[15] ——, "Analysis on roles of dnns in end-to-end acoustic scene analysis framework with distributed sound-to-light conversion devices," in *APSIPA Annual Summit and Conference*, Tokyo, Japan, Dec. 2021, pp. 1167–1172.

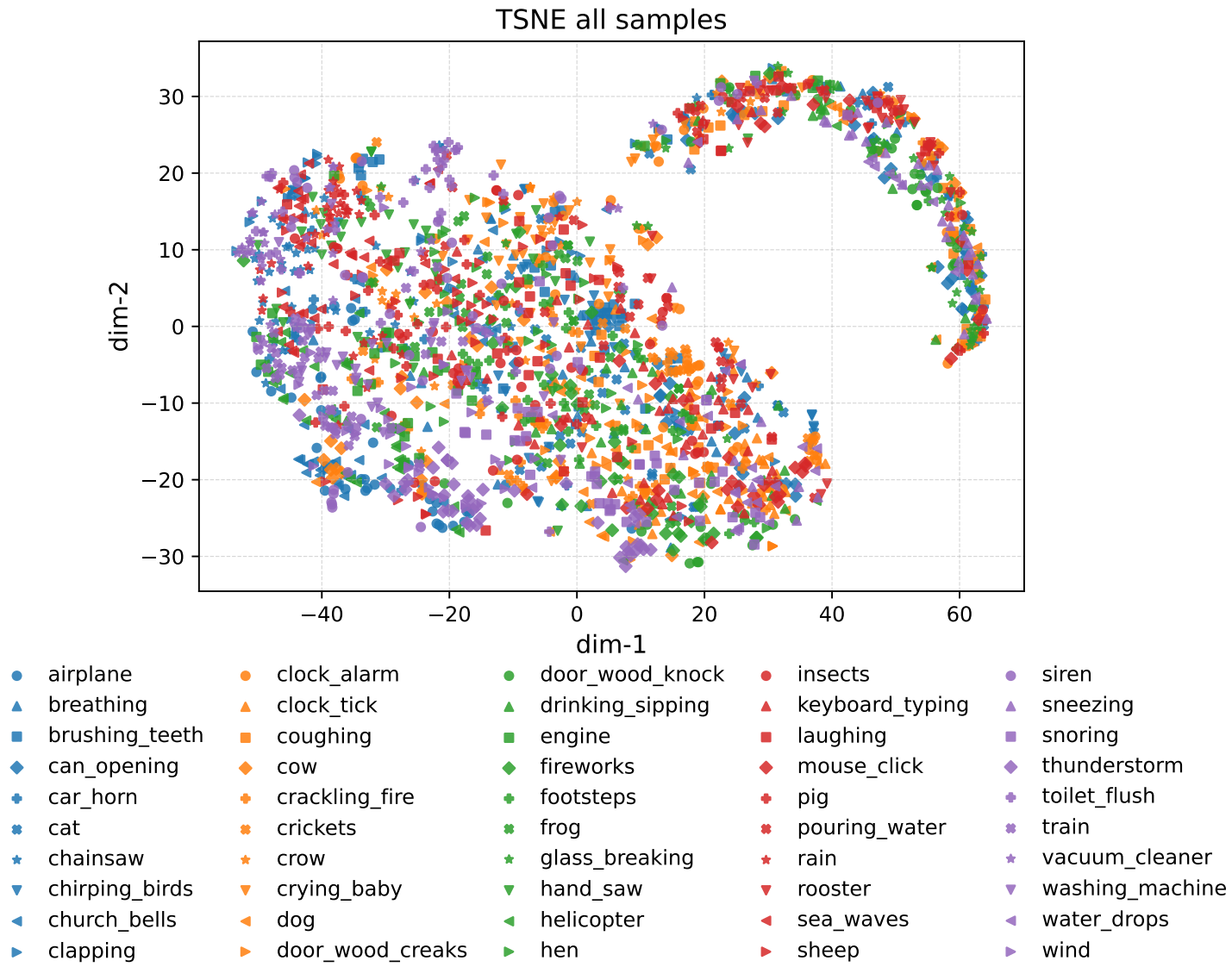

Figure 8: Visualization of the latent space distributions of normal autoencoder. Dimensionality reduction algorithms: T-SNE.

[16] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio Set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[17] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," 2017.

[18] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM Press, 2015, pp. 1015–1018.

[19] R. Scheibler, E. Bezzam, and I. Dokmanic, "Pyroomacoustics: A Python Package for Audio Room Simulation and Array Processing Algorithms," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 351–355.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2015.

[21] Y. Tokozume, Y. Ushiku, and T. Harada, "Learning from Between-class Examples for Deep Sound Recognition," 2018.

[22] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," 2019.

[23] P. H. Schönemann, "A generalized solution of the orthogonal procrustes problem," *Psychometrika*, vol. 31, no. 1, p. 1 – 10, 1966.

[24] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," 2020. [Online]. Available: https://arxiv.org/abs/2002.05709

[25] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," 2020. [Online]. Available: https://arxiv.org/abs/1911.05722

[26] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," 2021. [Online]. Available: https://arxiv.org/abs/2004.11362