APSIPA Transactions on Signal and Information Processing, 2025, 14, e34

Original Paper

Audio Difference Learning Framework for Audio Captioning

Tatsuya Komatsu[1,2*], Kazuya Takeda[2] and Tomoki Toda[2]

[1]*LY Corporation, Japan*
[2]*Nagoya University, Japan*

ABSTRACT

This paper proposes a novel learning method for audio captioning, which we call Audio Difference Learning. The core idea is to construct a feature space where differences between two audio inputs are explicitly represented as feature differences. This method has two main components. First, we introduce a diff block, which is placed between the audio encoder and text decoder. The diff block computes the difference between the features of an input audio clip and an additional reference audio clip. The text decoder then generates text descriptions based on the difference features. Second, we use a mixture of the original input audio and reference audio as a new input to eliminate the need for explicit difference annotations. The diff block then calculates the difference between the mixed audios embeddings and those of the reference audio. This difference embedding effectively cancels out the reference audio, leaving only information from the original audio input. Consequently, the model can learn to caption this difference using the original input audios caption, thus removing the need for additional difference annotations. In experiments conducted using the Clotho and ESC50 datasets, the proposed method achieved an 8% improvement in the SPIDEr score compared to conventional methods.

*Corresponding author: komatsu.tatsuya@lycorp.co.jp

Received 02 April 2025; revised 28 June 2025; accepted 04 August 2025
ISSN 2048-7703; DOI 10.1561/116.20250021

Keywords: Audio captioning, audio difference captioning, audio difference learning

1 Introduction

Audio captioning is a task of describing the content of input audio in natural language, and it has become an important technology in the field of audio processing [4, 32, 17, 18]. This technology supports a wide range of applications, such as providing accessibility services for the hearing impaired, enabling efficient audio content search, and analyzing audio environments for surveillance systems.

General audio captioning models employ an encoder-decoder framework [25]. These models consist of an audio encoder that extracts feature representations and a text decoder that generates captions from these representations. Techniques, including RNNs (Recurrent Neural Networks) [4, 32], CNNs (Convolutional Neural Networks), and Transformers [30] are often utilized in the audio encoder [17], while the decoder frequently employs Transformers.

A major challenge in audio captioning is the limited availability of paired audio-caption data. For instance, the widely used Clotho dataset [5] comprises roughly 5,000 audio clips ranging from 15 to 30 seconds with a total duration of approximately 30 hours. Similarly, AudioCaps [9] contains approximately 140 hours of data from about 46,000 clips, each around 10 seconds long. In contrast to Automatic Speech Recognition (ASR), which benefits from data spanning hundreds or even thousands of hours, audio captioning is limited by relatively small datasets. Consequently, many researchers resort to leveraging pre-trained models, such as PANNs [14] and BEATs [2], and employ data augmentation techniques.

However, data augmentation for audio captioning is not straightforward. While audio input alone can be augmented using various techniques such as speed perturbation and SpecAugment [20], caption augmentation poses a greater challenge. Several methods involve rephrasing captions through techniques like synonym substitution [3], adversarial training [19], and leveraging language models [23, 33]. Recently, approaches inspired by MixGen [7], initially proposed in the vision-language domain, have been explored [10, 3, 12]. These methods involve mixing two audio clips and concatenating their corresponding captions using conjunctions such as “and”. Other techniques propose using temporal connectors like ’followed by’ or ’after’ to capture the temporal dependencies of audio content [31, 35]. However, these rule-based strategies often show limited effectiveness, sometimes only enhancing performance on specific metrics. The augmented captions risk deviating from the actual content description, leading to potential performance degradation. Thus, developing a more effective learning methodology is essential.

In this paper, we propose a novel learning approach termed “audio difference learning.” This approach introduces a reference audio as an additional input during training. Captions are generated by the differences between the original input audio and the reference audio, computed in the feature space as a difference representation. A critical component of this method is the introduction of a "diff block," which computes audio differences. Initial experiments [13] demonstrate the effectiveness of a simple subtraction-based diff block. However, this approach necessitates the temporal alignment of mixed audio and reference audio, which can be limiting. To address these constraints and enhance flexibility, we also propose a masking-based diff block utilizing cross-attention mechanisms. This advanced block design allows for more robust handling of audio differences without strict alignment requirements. Additionally, we implement a learning strategy that circumvents the need for manual annotations and heuristic-rule-based text processing typically required for audio difference learning. Specifically, we create a mixed audio input by combining the reference audio with the original input audio. By computing the difference between the mixed input and the reference audio in the feature space, we effectively recreate the feature representation of the original input. Consequently, the target caption remains consistent with the original, allowing training without additional annotations or text processing. This method also enables new applications in which the differences between two audio clips are articulated in captions.

2 Related Works

Data augmentation for captioning.
Captioning tasks across images, video, and audio increasingly benefit from dataaugmentation techniques. In computer vision and video, vocabulary diversification via backtranslation [29], synonym substitution with BERT [1], languagemodelbased caption expansion for video [16], and the synthesis of imagecaption pairs through diffusion models [34] have all proven effective.

Audiospecific augmentation.
For audio captioning, MixGeninspired methods [10, 3, 12] concatenate the captions of two mixed audio inputs, whereas rulebased strategies attempt to reflect temporal order using connectors such as followed by [31, 35]. Although such rules capture temporal structure, their simplistic nature often degrades performance on standard metrics.

Leveraging large language models.
More recently, large language models (LLMs) like ChatGPT have been used to combine two captions [33] or to rephrase existing ones [23]. While LLMs

increase caption diversity, they operate without direct access to the audio signal and can therefore generate captions misaligned with the underlying content.

Differenceaware captioning in vision.
In vision, differenceaware learning explicitly models semantic differences between paired inputs. Early work–including SpottheDifference [8] and Neural Naturalist [6]–introduced datasets for finegrained comparison. Later models such as DUDA [21], viewpointadapted encoders [24], cycleconsistent training [11], and contrastive pretraining [36] improved robustness to viewpoint changes and irrelevant scene variations, confirming that attention, alignment, and contrastive objectives yield contextsensitive, discriminative captions.

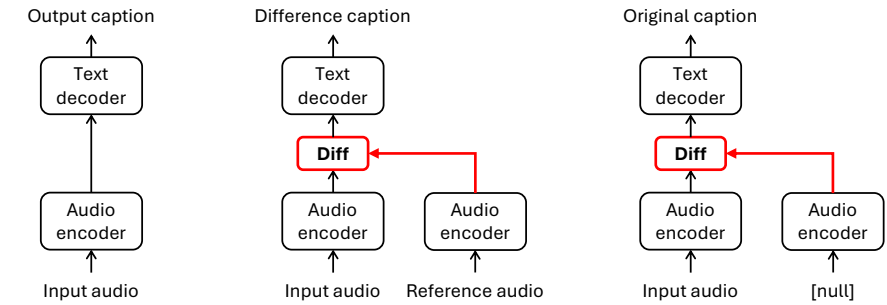
Difference modeling in audio.
For audio, recent studies [28, 26] generate captions that describe differences between two clips. These approaches target *difference captioning* itself, rely on specially crafted differenceoriented captions, and do not improve general audiocaption performance. Generalpurpose audio representation work [27] explores subtractionbased difference learning, yet our preliminary experiments indicate that pure subtraction is limiting.

Our contribution.
In contrast to these prior studies, our proposed method, audio difference learning, aims to enhance data augmentation by explicitly learning audio differences. Our masking-based diff block, leveraging cross-attention, offers greater flexibility and robustness in handling audio differences without requiring precise temporal alignment. Our proposed method not only augments data diversity but also removes the need for additional human annotations, offering a cost-effective and scalable solution for data augmentation in audio captioning.

3 Audio Captioning

Audio captioning is a technique for generating natural language descriptions of audio content. Audio captioning generally uses an encoder-decoder architecture consisting of an audio encoder that extracts features from the input audio and a text decoder that generates captions from the extracted features.

Figure 1-(a) shows the block diagram of audio captioning with an encoder-decoder structure. Let the spectral feature of the input audio be denoted as $\mathbf{X}_{\mathsf{in}} \in \mathbb{R}^{T \times F}$, and let the target text sequence be $\mathbf{y} \in \mathcal{V}^{L}$, where T is the length of the audio sequence, F is the number of frequency bins, and L is the length of the text sequence. First, the input audio $\mathbf{X}_{\mathsf{in}}$ is fed into the audio



(a) Audio captioning (b) The proposed method (c) The proposed method w/o reference audio

Figure 1: (a) A conventional audio captioning system, (b) The proposed method generates the difference between the input and reference audio based on the difference of the encoded audio representation. (c) The proposed method behaves as the general audio captioning system when the reference audio is null.

encoder, transforming it into a feature representation $\mathbf{Z} \in \mathbb{R}^{T \times D}$, where D is the feature dimension. This transformation can be expressed as:

$$\mathbf{Z} = \mathsf{AudioEncoder}(\mathbf{X}_{\mathsf{in}}). \tag{1}$$

The feature representation $\mathbf{Z}$ can be considered as a representation of the semantic content within $\mathbf{X}_{\mathsf{in}}$. Feeding $\mathbf{Z}$ into the decoder yields an estimation of $\mathbf{y}$, denoted as $\hat{\mathbf{y}}$, as:

$$\hat{\mathbf{y}} = \mathsf{TextDecoder}(\mathbf{Z}). \tag{2}$$

Here, $\hat{\mathbf{y}} \in [0, 1]^{L \times |\mathcal{V}|}$ represents the predicted probability distribution over the vocabulary corresponding to the content of the input audio $\mathbf{X}_{\mathsf{in}}$.

Cross-entropy between the predicted text sequence $\hat{\mathbf{y}}$ and the target text sequence $\mathbf{y}$ is commonly used as a loss function in the training process:

$$\mathcal{L} = \mathsf{CrossEntropy}(\mathbf{y}, \hat{\mathbf{y}}). \tag{3}$$

4 Proposed Method: Audio Difference Learning

4.1 Overview

In this paper, we propose audio difference learning which incorporates an additional reference audio input $\mathbf{X}_{\mathsf{ref}} \in \mathbb{R}^{T \times F}$. The proposed method is trained to describe the difference between the input $\mathbf{X}_{\mathsf{in}}$ and the reference $\mathbf{X}_{\mathsf{ref}}$, enabling

the model to capture subtle distinctions between audio inputs for caption generation.

The key aspect of our method is to construct a network that can perform semantic manipulation of audio content in the feature space by training a model based on differences in the representations. The structure is shown in Figure 1-(b). We begin by encoding $\mathbf{X}_{\mathsf{ref}}$ into a reference feature representation $\mathbf{Z}_{\mathsf{ref}}$ as in Equation 1:

$$\mathbf{Z}_{\mathsf{ref}} = \mathsf{AudioEncoder}(\mathbf{X}_{\mathsf{ref}}). \tag{4}$$

We then derive a difference representation $\mathbf{Z}_{\mathsf{diff}}$ by calculating the difference between the input $\mathbf{Z}_{\mathsf{in}}$ and the reference $\mathbf{Z}_{\mathsf{ref}}$:

$$\mathbf{Z}_{\mathsf{diff}} = \mathsf{diff}(\mathbf{Z}_{\mathsf{in}}, \mathbf{Z}_{\mathsf{ref}}). \tag{5}$$

Here, $\mathsf{diff}()$ is a critical function that must be carefully designed to represent the difference between two feature representations. The goal is to design a feature space where semantic addition and subtraction of audio can be performed. In this paper, we propose two approaches: one based on subtraction and another based on masking. The details of each will be explained in Sections 4.3.1 and 4.3.2.

The obtained difference representation $\mathbf{Z}_{\mathsf{diff}}$ is fed into the decoder,

$$\hat{\mathbf{y}}_{\mathsf{diff}} = \mathsf{TextDecoder}(\mathbf{Z}_{\mathsf{diff}}). \tag{6}$$

This results in $\hat{\mathbf{y}}_{\mathrm{diff}}$, which is a caption of the difference between the input and the reference audio. Note that if the reference audio $\mathbf{X}_{\mathsf{ref}}$ is set to null, the system behaves identically to a conventional audio captioning system as shown in Figure 1-(c). This highlights the versatility of the proposed method, making it applicable to various scenarios and inputs.

A major challenge is the difficulty of obtaining ground-truth labels for $\mathbf{y}_{\mathsf{diff}}$ to calculate the cross-entropy loss,

$$\mathcal{L}_{\mathsf{diff}} = \mathsf{CrossEntropy}(\mathbf{y}_{\mathsf{diff}}, \hat{\mathbf{y}}_{\mathsf{diff}}). \tag{7}$$

To train using Equation 7, it is necessary to annotate the text $\mathbf{y}_{\mathsf{diff}}$ that represents the difference between $\mathbf{X}_{\mathsf{in}}$ and $\mathbf{X}_{\mathsf{ref}}$. In this study, we propose a training strategy that enables training using difference representations without the need for annotation, making the process more efficient and scalable. The details of this strategy will be discussed in the following Section 4.2.

4.2 Training Strategy

We present a training strategy that eliminates the need for explicitly annotated differences. First, we construct a new input $\mathbf{X}_{\mathsf{in}}^{+}$ by adding the reference

audio to the original input in the time domain. Let the corresponding waveforms be denoted as $\mathbf{X}_{\mathsf{in}}^{+}$, $\mathbf{X}_{\mathsf{in}}$, and $\mathbf{X}_{\mathsf{ref}}$ $\mathbf{x}_{\mathsf{in}}^{+}$, $\mathbf{x}_{\mathsf{in}}$, and $\mathbf{x}_{\mathsf{ref}}$, respectively. Thus,

$$\mathbf{x}_{\mathsf{in}}^{+} = \mathbf{x}_{\mathsf{in}} + \mathbf{x}_{\mathsf{ref}}. \tag{8}$$

We then convert it into the spectral representation $\mathbf{X}_{\mathsf{in}}^{+}$. Next, we obtain feature representations of the new input and the reference as follows (see also Equations 1 and 4):

$$\mathbf{Z}_{\mathsf{in}}^{+} = \mathsf{AudioEncoder}(\mathbf{X}_{\mathsf{in}}^{+}), \tag{9}$$

$$\mathbf{Z}_{\mathsf{ref}} = \mathsf{AudioEncoder}(\mathbf{X}_{\mathsf{ref}}). \tag{10}$$

We compute the difference representation and its corresponding caption, as in Equations 5 and 6:

$$\mathbf{Z}_{\mathsf{diff}} = \mathsf{diff}(\mathbf{Z}_{\mathsf{in}}^{+}, \mathbf{Z}_{\mathsf{ref}}), \tag{11}$$

$$\hat{\mathbf{y}}_{\mathsf{diff}} = \mathsf{TextDecoder}(\mathbf{Z}_{\mathsf{diff}}). \tag{12}$$

Because $\mathbf{Z}_{\mathsf{in}}^{+}$ contains information from both the original input $\mathbf{X}_{\mathsf{in}}$ and the reference audio $\mathbf{X}_{\mathsf{ref}}$, the difference $\mathbf{Z}_{\mathsf{diff}} = \mathsf{diff}(\mathbf{Z}_{\mathsf{in}}^{+}, \mathbf{Z}_{\mathsf{ref}})$ aims to isolate and retain only the content of the *original* input $\mathbf{X}_{\mathsf{in}}$. Hence, the difference representation $\mathbf{Z}_{\mathsf{diff}}$ should produce the same caption $\mathbf{y}$ that originally describes $\mathbf{X}_{\mathsf{in}}$. We therefore compute a cross-entropy loss against the original caption $\mathbf{y}$:

$$\mathcal{L}_{\mathsf{diff}}^{+} = \mathsf{CrossEntropy}(\mathbf{y}, \hat{\mathbf{y}}_{\mathsf{diff}}^{+}). \tag{13}$$

This approach enables learning from audio differences without extra annotation costs.

4.3 Design of diff *Block*

In this section, we describe two approaches to the diff function: a subtraction-based approach (Figure 2) and a masking-based approach (Figure 3).

4.3.1 Subtraction-based Difference Calculation

In this section, we describe a subtraction-based approach to implementing the diff function, as shown in Figure 2. In this block, the difference embedding is obtained by simply subtracting the two embeddings $\mathbf{Z}_{\mathsf{in}}$ and $\mathbf{Z}_{\mathsf{ref}}$:

$$\begin{aligned} \mathbf{Z}_{\mathsf{diff}} &= \mathsf{diff}(\mathbf{Z}_{\mathsf{in}}, \mathbf{Z}_{\mathsf{ref}}) \\ &= \mathbf{Z}_{\mathsf{in}} - \mathbf{Z}_{\mathsf{ref}}. \end{aligned} \tag{14}$$

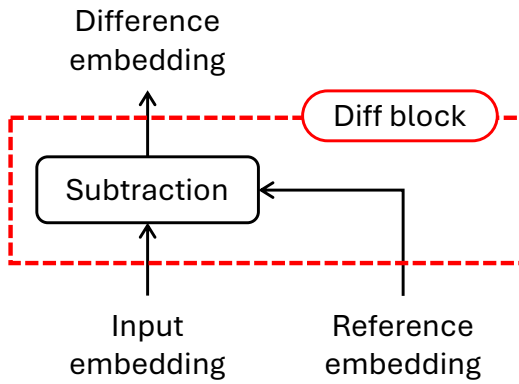


Figure 2: Diagram of the Diff block using the Subtraction method. The input embedding and reference embedding are subtracted to produce the difference embedding.

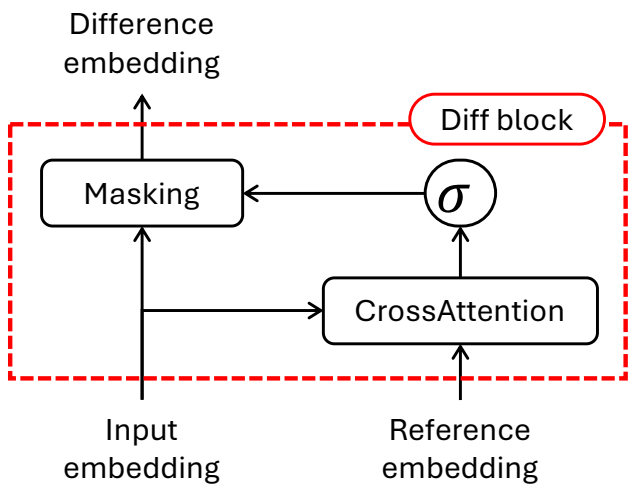


Figure 3: Diagram of the Diff block using the Masking method. Cross-attention calculates the similarity between the input and reference embeddings. The resulting weights are scaled by a sigmoid function and applied in the masking process to produce the difference embedding.

Because this method does not require additional parameters, it enables a simple and parameter-free implementation.

Note, however, that both the input audio and the reference audio must be temporally aligned. If the audio signals are misaligned in time, element-wise

subtraction may fail to capture meaningful differences. Temporal alignment ensures that corresponding elements in the embeddings are subtracted accurately, thus reflecting valid differences between the audio inputs.

4.3.2 Masking-based Difference Calculation

We next describe a masking-based approach to compute differences, illustrated in Figure 3. This approach first computes the similarity of $\mathbf{Z}_{\mathsf{in}}$ to $\mathbf{Z}_{\mathsf{ref}}$ using cross-attention, and then derives a mask that suppresses features in $\mathbf{Z}_{\mathsf{in}}$ that resemble those in $\mathbf{Z}_{\mathsf{ref}}$.

The process begins by calculating a cross-attention matrix between $\mathbf{Z}_{\mathsf{in}}$ and $\mathbf{Z}_{\mathsf{ref}}$:

$$\begin{aligned}\mathbf{Z}_{\mathsf{sim}} &= \mathsf{CrossAttention}(\mathbf{Z}_{\mathsf{in}}, \mathbf{Z}_{\mathsf{ref}}) \\ &= \mathsf{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^{\mathrm{T}}}{\sqrt{d}}\right)\mathbf{V},\end{aligned} \tag{15}$$

where $\mathbf{Z}_{\mathsf{sim}}$ has the same temporal length and feature dimension as $\mathbf{Z}_{\mathsf{in}}$. Here, $\mathbf{Q}, \mathbf{K}$, and $\mathbf{V}$ are computed by linear transformations of the respective embeddings:

$$\mathbf{Q} = \mathsf{Linear}(\mathbf{Z}_{\mathsf{in}}), \quad \mathbf{K} = \mathsf{Linear}(\mathbf{Z}_{\mathsf{ref}}), \quad \mathbf{V} = \mathsf{Linear}(\mathbf{Z}_{\mathsf{ref}}). \tag{16}$$

Each element of $\mathbf{Z}_{\mathsf{sim}}$ is passed through a sigmoid function (to map values into $[0, 1]$) and subtracted from 1 to create the mask $\mathbf{M}$:

$$\mathbf{M} = 1 - \sigma(\mathbf{Z}_{\mathsf{sim}}). \tag{17}$$

This mask emphasizes parts of $\mathbf{Z}_{\mathsf{in}}$ that are less related to $\mathbf{Z}_{\mathsf{ref}}$. Finally, the mask $\mathbf{M}$ is applied to $\mathbf{Z}_{\mathsf{in}}$ by element-wise multiplication:

$$\mathbf{Z}_{\mathsf{masked}} = \mathbf{M} \odot \mathbf{Z}_{\mathsf{in}}. \tag{18}$$

Hence, the function $\mathsf{diff}(\mathbf{Z}_{\mathsf{in}}, \mathbf{Z}_{\mathsf{ref}})$ preserves features in $\mathbf{Z}_{\mathsf{in}}$ that are distinct from those in $\mathbf{Z}_{\mathsf{ref}}$. Since this method leverages cross-attention rather than simple subtraction, it can handle scenarios in which $\mathbf{Z}_{\mathsf{in}}$ and $\mathbf{Z}_{\mathsf{ref}}$ are not aligned in time, unlike the subtraction-based approach.

5 Experimental Evaluations

5.1 Experimental Settings

We employed the baseline system[1] from DCASE2023 Task6 for the captioning model. We kept all hyperparameters consistent with the baseline. The input

[1] https://github.com/felixgontier/dcase-2023-baseline.

audio features are 64-dimensional mel-spectrograms with a sampling rate of 44.1 kHz, a window length of 40 ms and a hop size of 20 ms. The audio encoder uses a pre-trained 12-layer CNN, followed by an adapter composed of linear layers. The feature representation has a dimensionality of 768. The text decoder uses BART [15]. In the masking-based approach, we insert a cross-attention module that linearly projects $\mathbf{Z}_{\mathsf{in}}$ and $\mathbf{Z}_{\mathsf{ref}}$ into 768-dimensional query, key, and value vectors (see Equation 16). The attention output is passed through a sigmoid, inverted, and then element-wise multiplied with $\mathbf{Z}_{\mathsf{in}}$ to produce the masked representation (Equation 18). The training was conducted for 40 epochs, with a batch size of 32. We used the model parameters from the 40th epoch for evaluation.

In the experiment, we compared the baseline trained with standard cross-entropy to our proposed audio difference learning method. We also conducted a comparison with Al-MixGen, also referred to as PairMix [10], a mix-up-like augmentation that involves mixing two audio files and generating a target caption by concatenating the individual captions.

We evaluated captioning performance using the coco caption toolkit,[2] which computes standard metrics [18] such as BLEU_n, METEOR, and ROUGE_L, that assess n-gram precision, lexical overlap, and other linguistic factors. We also used CIDEr, SPICE, and SPIDEr, which place emphasis on term frequency-inverse document frequency weighting and scene graph alignment.

5.2 Design of Training Dataset

In our experiments, we utilized two datasets: the Clotho [5] and ESC-50 [22] datasets. The Clotho dataset is commonly used in audio captioning tasks and consists of 4981 audio clips, each 15-30 seconds long. These clips comprising 2,893 training, 1,045 validation, and 1,043 test samples. The ESC-50 dataset is a collection of 2000 environmental sound clips evenly distributed across 50 different classes. Each class represents a specific sound event, offering a diverse set of reference sounds for this study.

For the generation of $\mathbf{X}^{+}_{\mathsf{in}}$, we superimposed Clotho and ESC-50 audio clips in the time domain with matched power levels. For the reference audio $\mathbf{X}_{\mathsf{ref}}$, we used three conditions based on whether the same ESC-50 clip is used and whether the clips are temporally aligned. These patterns are described as (Sound source / Time alignment):

- **(Same / Same)**: The audio clip for $\mathbf{X}^{+}_{\mathsf{in}}$ generation and $\mathbf{X}_{\mathsf{ref}}$ is the same, and are temporally aligned.
- **(Same / Diff)**: The audio clip for $\mathbf{X}^{+}_{\mathsf{in}}$ generation and $\mathbf{X}_{\mathsf{ref}}$ is the same, are not temporally aligned.

[2] https://github.com/tylin/coco-caption.

- **(Diff / Diff)**: The audio clips for $\mathbf{X}_{\mathsf{in}}^{+}$ generation and $\mathbf{X}_{\mathsf{ref}}$ are different but belong to the same acoustic class, and have different time alignments.

Additionally, we also considered conditions in which multiple ESC-50 clips were superimposed: **(Multi / Same)** and **(Multi / Diff)**. The detailed procedure for loading audio clips is described in the following algorithm:

Algorithm 1 Audio loading process.

Require: Clotho dataset, ESC-50 dataset
Ensure: Generated audio $\mathbf{X}_{\mathsf{in}}^{+}$ and reference audio $\mathbf{X}_{\mathsf{ref}}$
Sample a Clotho audio clip $\mathbf{X}_{\mathsf{clotho}}$ and an ESC-50 audio clip $\mathbf{X}_{\mathsf{esc}}$
if source condition is **(Same / *) then**
    Set reference ESC-50 audio $\mathbf{X}_{\mathsf{ref}} = \mathbf{X}_{\mathsf{esc}}$
else
    Sample a different ESC-50 clip $\mathbf{X}_{\mathsf{ref}}$ from the same class
end if
Randomly choose t_{add} within the duration of $\mathbf{X}_{\mathsf{clotho}}$
Superimpose $\mathbf{X}_{\mathsf{esc}}$ onto $\mathbf{X}_{\mathsf{clotho}}$ at t_{add} to create $\mathbf{X}_{\mathsf{in}}^{+}$
if scenario is **(* / Same) then**
    Zero-pad $\mathbf{X}_{\mathsf{ref}}$ to ensure temporal alignment with t_{add}
end if
if using pattern **(Multi / *) then**
    Repeat the sampling and superimposition steps for additional ESC-50 clips
end if
return the pair $\mathbf{X}_{\mathsf{in}}^{+}$ and $\mathbf{X}_{\mathsf{ref}}$

The subtraction-based difference calculation is limited to the (Same / Same) scenario, whereas the masking-based approach is applicable to all other scenarios. Performance variations across these patterns are analyzed in Section 5.4.

5.3 Experimental Results

Table 1 presents the experimental results for the audio captioning task. All metrics were evaluated on the test split of the Clotho dataset. The proposed method used the reference audio only during training and did not use it during evaluation. These results highlight the effectiveness of our proposed audio difference learning for general audio captioning. In the SPIDEr metric, we achieved an 8% improvement. These results confirm the effectiveness of the proposed audio difference learning.

Table 1: Experimental results of the general audio captioning task setting: The proposed method employed the reference audio only during the training phase, and it was not used during the evaluation. These results highlight the impact of our proposed audio difference learning on the general audio captioning.

Model	Bleu_1	Bleu_2	Bleu_3	Bleu_4	METEOR	ROUGE_L	CIDEr	SPICE	SPIDEr
Baseline	0.585	0.379	0.251	0.161	0.179	0.386	0.399	0.120	0.259
AL-MixGen [10]	0.590	0.384	0.254	0.164	0.180	0.392	0.404	0.122	0.263
Proposed (Subtraction)	0.593	0.386	0.257	0.164	0.181	0.392	0.403	0.122	0.264
Proposed (Mask)	**0.606**	**0.395**	**0.266**	**0.173**	**0.187**	**0.405**	**0.431**	**0.129**	**0.280**

Both the subtraction- and masking-based approaches outperform the baseline and the conventional AL-MixGen method. The proposed method with the masking-based approach exhibits the best performance. It achieves the highest scores across all metrics, particularly excelling in CIDEr and SPIDEr.

The superiority of the masking-based approach can be attributed to the increased expressive capacity enabled by the use of cross-attention. This allows for a more effective modeling of subtle differences between different audio inputs. In contrast, the Subtraction approach is less flexible because it relies on temporal alignment. These results suggest that the masking-based method is more adaptable and better suited to handle a wider variety of audio inputs, making it a robust, versatile solution for audio captioning.

5.4 Comparison of diff *Blocks*

Table 2 also summarizes the results obtained with the two diffblock variants, *Subtraction* and *Masking*, under five mixing conditions defined by the origin of the ESC50 clip and its temporal alignment with the reference. The key findings are as follows.

Table 2: Performance comparison of the proposed methods with various combinations of audio clips. Evaluations were conducted using Subtraction and Mask approaches across different source and time alignment patterns. The results indicate that the Mask approach consistently outperforms others, maintaining high performance across various conditions.

Model (Source / Time)	Bleu_1	Bleu_2	Bleu_3	Bleu_4	METEOR	ROUGE_L	CIDEr	SPICE	SPIDEr
Subtraction (Same / Same)	0.590	0.383	0.253	0.162	0.180	0.390	0.403	0.121	0.262
Mask (Same / Same)	**0.606**	**0.395**	**0.266**	**0.173**	**0.187**	0.405	**0.431**	**0.129**	**0.280**
Mask (Same / Diff)	0.605	0.394	0.264	0.171	**0.187**	0.405	0.427	**0.129**	0.278
Mask (Diff / Diff)	0.605	0.392	0.261	0.168	0.186	**0.406**	0.419	0.126	0.273
Mask (Multi / Same)	**0.606**	0.394	0.264	0.171	**0.187**	0.404	0.419	0.127	0.273
Mask (Multi / Diff)	0.603	0.389	0.259	0.167	0.184	0.401	0.414	0.127	0.271

The **Subtraction** block can be trained only in the (Same / Same) scenario, because it requires the reference audio to be identical and perfectly timealigned with the audio mixed into $\mathbf{X}_{\mathsf{in}}^{+}$. All other scenarios violate this assumption, making elementwise subtraction meaningless. In contrast, the

Masking block, implemented with crossattention and a learned mask, successfully trains under all five conditions, demonstrating far broader coverage.

When the ideal (Same / Same) condition is met, Masking attains the highest scores across all metrics, exceeding Subtraction despite the latters perfect temporal alignment. This margin reflects the greater generalization power of the crossattention mechanism, which can model nonlinear differences better than a simple vector subtraction.

Moving from (Same / Same) to (Same / Diff), where the mixed segment is timeshifted, the Masking block loses only 0.002 absolute SPIDEr (0.7% relative), confirming that it can internally compensate for timing discrepancies. Even when the reference is a different audio recording of the same class (Diff/Diff), the decline is merely 0.007 SPIDEr (2.5%), showing that the learned attention mask still isolates the original Clotho content effectively.

Superimposing multiple ESC50 clips (Multi/Same) and (Multi/Diff) produces no additional gains and even marginally lowers SPIDEr. A likely reason is that dense overlapped events dilute the semantic cues originating from the target clip.

Because realworld deployments rarely provide perfectly aligned {input, reference} pairs, the Masking blocks resilience to misalignment and source variation makes it the preferred choice. However, at present there are no publicly available corpora of real-world {input, reference} pairs; our evaluation therefore relied on synthetic mixtures. Collecting genuine paired recordings remains an open challenge for future research and would allow the true upper bound of differenceaware captioning to be measured in practical settings.

5.5 Inference Examples of Audio Difference

Table 3 presents examples of captions generated by our proposed method and the baseline in four different ways: (1) Original input from the Clotho dataset (in), (2) Mixed sound of in with sounds from the ESC-50 dataset (in_+), (3) Captions produced from the difference representation between the mixed audio and the ESC50 clip; this caption should match that of (1). (4) Captions produced from the difference representation between the mixed audio and the original input (the inverse of (3)), leaving the ESC50 component to be captioned.

Both the proposed method and the baseline perform well in captioning the input. However, the baseline struggles with the input_+ when an additional event is superimposed. The proposed audio difference learning method successfully separates and encodes the individual audio contents in the representation space.

In the examples where captions are generated based on the difference representation, it can be seen that the proposed method is able to caption only the semantic difference from the original audio in the input sound as highlighted

Table 3: Examples of difference captioning results generated using difference-representation between two audio. The proposed method can handle mixed sounds and differences.

(1) Input Audio: *kids are playing as one child shrieks while birds are chirping*	
Baseline	birds are chirping and children are talking in the background
Proposed	birds are chirping and children are talking and playing in the background
(2) Caption for mixed audio: $\mathsf{in}_+ = \mathsf{in} + \mathsf{Car\ Horn\ sound}$	
Baseline	birds are chirping and children are talking in the background
Proposed	birds are chirping and children are talking in the background **as a car drives by**
(3) Caption with difference representation: $\mathsf{in}_+ - \mathsf{Car\ Horn\ sound} \Rightarrow (1)$	
Baseline	birds are chirping and children are talking to each other
Propose	birds are chirping and children are talking in the background
(4) Caption with difference representation: $\mathsf{in}_+ - \mathsf{in} \Rightarrow \mathsf{Car\ Horn\ sound}$	
Baseline	a person is using a hard object to make a few seconds
Proposed	**an engine is whirring** and then it gets louder and louder
(1) Input Audio: *a distorted drum or similar instrument is played*	
Baseline	a synthesizer is playing a musical instrument
Proposed	a synthesizer is playing a synthesizer with a musical instrument
(2) Caption for mixed audio: $\mathsf{in}_+ = \mathsf{in} + \mathsf{Laughing\ sound}$	
Baseline	a person is playing a synthesizer with a musical instrument in the background
Proposed	a person is playing a synthesizer **with a man talks in the background**
(3) Caption with difference representation: $\mathsf{in}_+ - \mathsf{Laughing\ sound} \Rightarrow (1)$	
Baseline	a synthesizer is playing a musical instrument
Propose	a synthesizer is playing a musical instrument
(4) Caption with difference representation: $\mathsf{in}_+ - \mathsf{in} \Rightarrow \mathsf{Laughing\ sound}$	
Baseline	a person is playing a synthesizer while another person is speaking in the background
Proposed	a person is speaking and then **a child laughs**

in **bold**. The baseline, on the other hand, is unable to model the difference representation effectively, leading to captions that blend events from both superimposed sounds, particularly for $\mathsf{input}_+ - \mathsf{input} = \mathsf{input}_{\mathsf{esc}}$.

The proposed method not only improves performance but also suggests the potential for additional new applications, such as captioning differences between two audio recordings.

6 Conclusion

In this paper, we introduced a novel learning method for audio captioning, termed Audio Difference Learning. This method trains a model to generate captions based on feature representations of differences between audio inputs, establishing a representational space specifically attuned to these differences. By designing the input and reference audio so that the difference representation can reconstruct the original data, we enable learning without the need for human-annotated differences. Our experiments demonstrated that the

proposed method achieves superior captioning performance compared to traditional approaches. Additionally, the results indicate potential new applications, such as generating captions that specifically articulate the nuances of differences between audio recordings.

Acknowledgments

This work was partly supported by a project, JPNP20006, commissioned by NEDO, Japan.

References

[1] V. Atliha and D. eok, "Text Augmentation Using BERT for Image Captioning", *Applied Sciences*, 10(17), 2020, ISSN: 2076-3417, DOI: 10.3390/app10175978, https://www.mdpi.com/2076-3417/10/17/5978.

[2] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, "BEATs: Audio Pre-Training with Acoustic Tokenizers", in *Proc. ICML2023*, ed. A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Vol. 202, *Proceedings of Machine Learning Research*, PMLR, July 2023, 5178–93, https://proceedings.mlr.press/v202/chen23ag.html.

[3] J.-H. Cho, Y.-A. Park, J. Kim, and J.-H. Chang, "HYU submission for the DCASE 2023 task 6a: automated audio captioning model using ALMixGen and synonyms substitution", *tech. rep.*, DCASE2023 Challenge, May 2023.

[4] K. Drossos, S. Adavanne, and T. Virtanen, "Automated audio captioning with recurrent neural networks", in *Proc. WASPAA2017*, 2017, 374–8, DOI: 10.1109/WASPAA.2017.8170058.

[5] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset", in *Proc. ICASSP2020*, IEEE, 2020, 736–40.

[6] M. Forbes, C. Kaeser-Chen, P. Sharma, and S. Belongie, "Neural Naturalist: Generating Fine-Grained Image Comparisons", in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019, 708–17.

[7] X. Hao, Y. Zhu, S. Appalaraju, A. Zhang, W. Zhang, B. Li, and M. Li, "MixGen: A new multi-modal data augmentation", in *Proc. WACV 2023*, 2023, https://www.amazon.science/publications/mixgen-a-new-multi-modal-data-augmentation.

[8] H. Jhamtani and T. Berg-Kirkpatrick, "Learning to Describe Differences Between Pairs of Similar Images", in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018, 4024–34.

[9] C. D. Kim, B. Kim, H. Lee, and G. Kim, "Audiocaps: Generating captions for audios in the wild", in *Proc. NAACL-HLT2019*, 2019, 119–32.

[10] E. Kim, J. Kim, Y. Oh, K. Kim, M. Park, J. Sim, J. Lee, and K. Lee, "Exploring Train and Test-Time Augmentations for Audio-Language Learning", *arXiv preprint arXiv:2210.17143*, 2023.

[11] H. Kim, J. Kim, H. Lee, H. Park, and G. Kim, "Viewpoint-Agnostic Change Captioning with Cycle Consistency", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, 1621–30.

[12] J. Kim, Y.-A. Park, J.-H. Cho, and J.-H. Chang, "Improving Automated Audio Captioning Fluency Through Data Augmentation and Ensemble Selection", in *Proc. DCASE2023*, Tampere, Finland, September 2023, 86–90.

[13] T. Komatsu, Y. Fujita, K. Takeda, and T. Toda, "Audio Difference Learning for Audio Captioning", in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2024, 1456–60.

[14] 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", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28, 2020, 2880–94.

[15] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension", in *Proc. ACL202*, Association for Computational Linguistics, 2020.

[16] S. Li, B. Yang, and Y. Zou, "Utilizing Text-based Augmentation to Enhance Video Captioning", in *2022 5th International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 2022, 287–93, DOI: 10.1109/ICAIBD55127.2022.9820499.

[17] X. Mei, X. Liu, Q. Huang, M. D. Plumbley, and W. Wang, "Audio Captioning Transformer", in *DCASE2021*, Barcelona, Spain, November 2021, 211–5, ISBN: 978-84-09-36072-7.

[18] X. Mei, X. Liu, M. D. Plumbley, and W. Wang, "Automated Audio Captioning: An Overview of Recent Progress and New Challenges", *EURASIP J. Audio Speech Music Process.*, 2022(1), October 2022, ISSN: 1687-4714, DOI: 10.1186/s13636-022-00259-2, https://doi.org/10.1186/s13636-022-00259-2.

[19] X. Mei, X. Liu, J. Sun, M. D. Plumbley, and W. Wang, "Diverse Audio Captioning via Adversarial Training", in *ICASSP*, 2022, 8882–6.

[20] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition", *Proc. Interspeech 2019*, 2019, 2613–7.

[21] D. H. Park, T. Darrell, and A. Rohrbach, "Robust Change Captioning", in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, 4623–32.

[22] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification", in *Proc. ACMMM2015*, Brisbane, Australia: ACM Press, October 13, 2015, 1015–8, ISBN: 978-1-4503-3459-4, DOI: 10.1145/2733373.2806390, http://dl.acm.org/citation.cfm?doid=2733373.2806390.

[23] P. Primus, K. Koutini, and G. Widmer, "Advancing Natural-Language Based Audio Retrieval with Passt and Large Audio-Caption Data Sets", in *Proc. DCASE2023*, Tampere, Finland, September 2023, 161–5.

[24] X. Shi, X. Yang, J. Guan, S. Joty, and J. Cai, "Finding it at Another Side: A Viewpoint-Adapted Matching Encoder for Change Captioning", in *European Conference on Computer Vision (ECCV)*, 2020, 574–90.

[25] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks", *Advances in neural information processing systems*, 27, 2014.

[26] D. Takeuchi, Y. Ohishi, D. Niizumi, N. Harada, and K. Kashino, "Audio Difference Captioning Utilizing Similarity-Discrepancy Disentanglement", in *Proc. DCASE2023*, Tampere, Finland, September 2023, 191–5.

[27] D. Takeuchi, M. Yasuda, D. Niizumi, and N. Harada, "Towards Learning a Difference-Aware General-Purpose Audio Representation", in *Proceedings of the Detection and Classification of Acoustic Scenes and Events 2024 Workshop (DCASE2024)*, Tokyo, Japan, October 2024, 176–80.

[28] S. Tsubaki, Y. Kawaguchi, T. Nishida, K. Imoto, Y. Okamoto, K. Dohi, and T. Endo, "Audio-Change Captioning to Explain Machine-Sound Anomalies", in *Proc. DCASE2023*, Tampere, Finland, September 2023, 201–5.

[29] I. R. Turkerud and O. J. Mengshoel, "Image Captioning using Deep Learning: Text Augmentation by Paraphrasing via Backtranslation", in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2021, 1–10, DOI: 10.1109/SSCI50451.2021.9659834.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez,. Kaiser, and I. Polosukhin, "Attention is all you need", in *Proc NIPS*, 2017, 6000–10.

[31] H.-H. Wu, O. Nieto, J. P. Bello, and J. Salamon, "Audio-Text Models Do Not Yet Leverage Natural Language", in *Proc. ICASSP 2023*, 2023, 1–5, DOI: 10.1109/ICASSP49357.2023.10097117.

[32] M. Wu, H. Dinkel, and K. Yu, "Audio Caption: Listen and Tell", in *Proc. ICASSP2019*, 2019, 830–4, DOI: 10.1109/ICASSP.2019.8682377.

[33] S.-L. Wu, X. Chang, G. Wichern, J.-w. Jung, F. Germain, J. L. Roux, and S. Watanabe, "BEATs-based audio captioning model with INSTRUCTOR embedding supervision and ChatGPT mix-up", *tech. rep.*, DCASE2023 Challenge, May 2023.

[34] C. Xiao, S. X. Xu, and K. Zhang, "Multimodal Data Augmentation for Image Captioning Using Diffusion Models", in, *LGM3A '23*, Ottawa ON, Canada: Association for Computing Machinery, 2023, 23–33, ISBN: 9798400702839, DOI: 10.1145/3607827.3616839, https://doi.org/10.1145/3607827.3616839.

[35] Z. Xie, X. Xu, M. Wu, and K. Yu, "Enhance Temporal Relations in Audio Captioning with Sound Event Detection", in *Proc. INTERSPEECH 2023*, 2023, 4179–83, DOI: 10.21437/Interspeech.2023-1614.

[36] L. Yao, W. Wang, and Q. Jin, "Image Difference Captioning with Pre-training and Contrastive Learning", in *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, 2022, 3108–16.