DistillCaps: Enhancing Audio-Language Alignment in Captioning via Retrieval-Augmented Knowledge Distillation

Thinh Pham*
thinhcm2003@gmail.com
University of Science,
Vietnam National University
Ho Chi Minh City, Vietnam

Nghiem Diep*
nghiemdt2013@gmail.com
University of Science,
Vietnam National University
Ho Chi Minh City, Vietnam

Lizi Liao
lzliao@smu.edu.sg
School of Computing and Information Systems,
Singapore Management University
Singapore

Binh T. Nguyen†
ngtbinh@hcmus.edu.vn
AISIA Lab, University of Science,
Vietnam National University
Ho Chi Minh City, Vietnam

Abstract

Automated audio captioning (AAC) benefits from incorporating external context to interpret complex sounds, but doing so with retrieval-augmented generation (RAG) at inference is sometimes infeasible due to data availability or incurs significant latency and complexity. We propose **DistillCaps**, a novel training-time framework that leverages RAG to guide knowledge distillation for improved audio-language alignment, while lessening the reliance on retrieval during inference. In our framework, a RAG-equipped teacher model retrieves relevant textual information (*e.g.*, similar captions) for each audio clip and uses it for training to generate context-enriched captions. Simultaneously, a student model is trained to imitate this teacher, learning to produce high-quality captions from audio alone. We further introduce a Fast Fourier Transform (FFT) adapter in the audio encoder to inject frequency-domain features, enhancing the quality of audio representations before feeding them into the language model. The result is an efficient captioning model that retains RAG's contextual benefits without its deployment overhead. On standard AAC benchmarks (AudioCaps and Clotho), DistillCaps achieves performance competitive with or exceeding prior RAG-based systems despite using no retrieval at test time. Notably, our distilled model matches state-of-the-art captioning results under real-time settings, and when optionally allowing retrieval, it even outperforms previous models by up to 4% on the Clotho benchmark on the in-distribution setting, demonstrating the effectiveness of RAG-guided distillation for audio-language alignment. Code and dataset are available here[1].

CCS Concepts

• **Computing methodologies** → **Natural language generation**;
• **Information systems** → *Retrieval models and ranking*.

Keywords

Audio Captioning; Knowledge Distillation; RAG

ACM Reference Format:
Thinh Pham, Nghiem Diep, Lizi Liao, and Binh T. Nguyen. 2025. DistillCaps: Enhancing Audio-Language Alignment in Captioning via Retrieval-Augmented Knowledge Distillation. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management (CIKM '25), November 10–14, 2025, Seoul, Republic of Korea.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3746252.3761269

1 Introduction

Automated audio captioning (AAC) is the task of automatically generating textual descriptions for audio clips, aiming to interpret complex auditory scenes (*e.g.*, speech, music, environmental sounds) and produce coherent, semantically accurate sentences that capture key sound events and context [36, 53]. By converting audio into descriptive text, AAC enables important applications: it can provide accessibility for hearing-impaired users by transcribing sound into captions, improve multimedia search by indexing audio content as text for retrieval, and help intelligent systems better understand their environment by describing surrounding sounds.

A typical AAC model consists of two main components: an audio encoder and a text decoder, as shown in Figure 2a. The audio encoder, often based on convolutional or transformer architectures such as PANNs [26], HTS-AT [5], BEATs [7], or CED [10], extracts high-level acoustic features from the raw audio and maps them into a semantic embedding space aligned with text. The text decoder, commonly a transformer-based language model like GPT-2 [39], BART [28], or LLaMA [46], then generates a natural language caption from these audio-informed embeddings. Despite this architecture, key challenges remain in bridging the semantic gap between audio and language, handling limited training data, and ensuring the captions generalize to diverse sounds.

Over the years, various strategies have been explored to address these challenges. Fully transformer-based encoder–decoder architectures (*e.g.*, ACT [35]) have improved modeling capacity for

*Equal contributions.
†Corrresponding Author: Binh T. Nguyen (ngtbinh@hcmus.edu.vn)
[1]https://github.com/pgthinh/DistillCaps

CIKM '25, Seoul, Republic of Korea.

ACM ISBN 979-8-4007-2040-6/2025/11
https://doi.org/10.1145/3746252.3761269

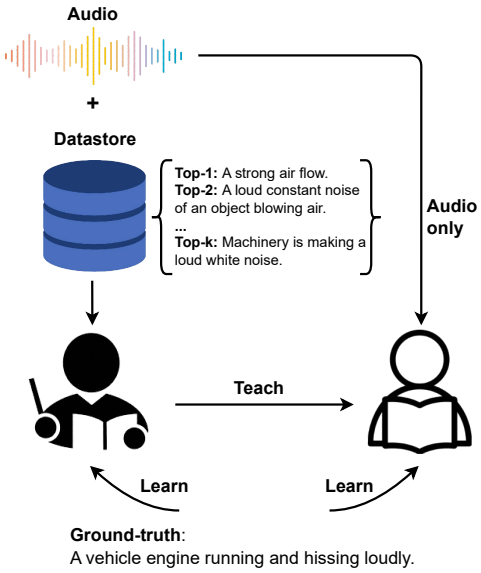

Figure 1: We propose DistillCaps, a RAG-aware, distillation-based framework for audio captioning. DistillCaps can help model retains RAG's contextual benefits without its deployment during inference thanks to distillation mechanism.

sequential audio-text data. Incorporating high-level semantic cues has also proven effective: BART-tags [18] injects AudioSet tags into the decoder, and CNN10-AT [50] leverages pretrained audio tagging features to enrich the encoder representations, reducing reliance on large caption corpora. To exploit large pre-trained language models, approaches like Prefix AAC [24] and Pengi [9] use prefix-tuning to adapt a frozen GPT-2, improving sample efficiency for AAC. Multi-scale feature fusion methods (LHDFF [42], PFCA-Net [43]) further enhance audio representations by capturing both coarse and fine-grained acoustic patterns. More recently, advanced audio-language models like EnCLAP [23] use a discretized audio codec (EnCodec [14]) to provide a better audio embedding for a BART-based captioner. To enrich the decoder's context, retrieval-based methods such as RECAP [17] and DRCap [30] follow the retrieval-augmented generation (RAG) paradigm (an architecture example is shown in Figure 2b): they employ a CLAP [15] model to retrieve semantically relevant captions from a reference database and feed these as additional context to the captioning model. By injecting external information in this way, these RAG-based approaches significantly improve caption quality and audio–text alignment compared to using the audio alone.

However, deploying RAG-based systems in practice comes with notable drawbacks. Relying on external retrieval at inference time introduces extra latency (due to database lookup) and system complexity (additional retrieval parts and memory for storing knowledge), which can hinder real-time performance and complicate maintenance. In resource-constrained or privacy-sensitive scenarios, maintaining a large up-to-date caption datastore is impractical. These limitations motivate a key question: ***can we reap the benefits of retrieval-augmented training without the overhead of retrieval during deployment***? We hypothesize that a captioning model can indeed learn from RAG's enriched context during training and yet operate independently at inference, thus enjoying improved audio-language alignment without runtime retrieval.

In this paper, we propose a framework to realize that idea. We introduce **DistillCaps** (as shown in Figure 1), which leverages RAG during training via a knowledge distillation approach, thereby mitigating retrieval at inference. Specifically, we first equip a **teacher** captioning model with retrieval-augmented generation: for each input audio, the teacher retrieves relevant textual context (such as similar audio captions or other external descriptions) and feeds this information into a language model for training to produce an enriched caption. At the same time, we train a **student** model to mimic the teacher's output using only the audio as input. Through this RAG-guided distillation process, the student model learns to generate captions that are as informative as the teacher's, despite not having access to external knowledge at runtime. Additionally, we incorporate an FFT-based adapter into the audio encoder to inject frequency-domain information before the audio features are mapped into the language model's embedding space. This adapter helps capture global temporal patterns and salient frequency characteristics of the audio, making the learned audio representation more compatible with the language model and improving robustness to noise. The overall design distills the alignment-strengthening benefits of the RAG-enhanced teacher into a streamlined student model, enabling it to benefit from external knowledge during training while maintaining a simple, efficient architecture for inference.

In summary, our main contributions are as follows:

- We propose a distillation module leveraging RAG at training time to enhance audio-language alignment, enabling effective captioning without retrieval during inference.
- We introduce an FFT-based adapter that captures frequency-domain features, improving temporal robustness and quality of audio representations.
- Our distilled model achieves competitive or superior performance on standard benchmarks without using retrieval during inference, and further outperforms existing models when retrieval is optionally retained.

2 Related Works

In this section, we will provide a brief overview of research areas related to this paper, including: Automated Audio Captioning, Fast Fourier Transform, Retrieval-Augmented Generation, and Knowledge Distillation.

2.1 Automated Audio Captioning

Automated Audio Captioning (AAC) has attracted increasing attention since its introduction in 2017 [12]. Initially, research focused on employing recurrent neural networks (RNNs) to generate natural language descriptions directly from audio signals. Subsequently, deep learning approaches, especially those utilizing the encoder-decoder framework, have become the standard methodology for this cross-modal task. Within this framework, the encoder is responsible for extracting meaningful audio features from raw audio, while the decoder generates the corresponding textual descriptions.

In early AAC studies, RNNs were predominantly used as encoders [41, 48, 56]. However, these models frequently encountered

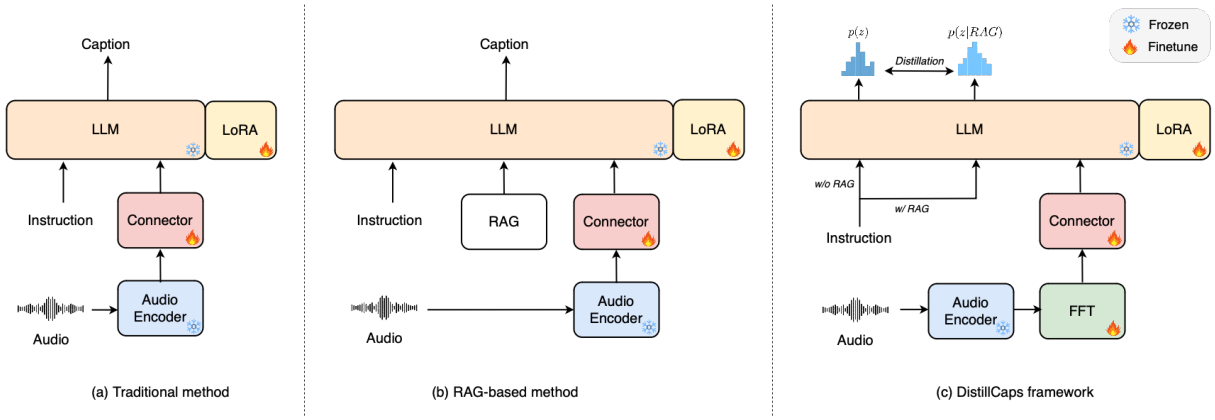

Figure 2: A Comparison of DistillCaps framework vs. previous methods. (a) Traditional automated audio captioning architecture. (b) RAG is used to add additional information related to audio input, thereby enhancing audio understanding ability of LLM. (c) Our framework that integrate Distillation-based Distribution Alignment (DDA) module to align RAG-based and RAG-free responses to retains RAG's contextual benefits without its deployment overhead. FFT adapter is also used to inject frequency-domain patterns to audio features.

challenges when handling long audio sequences, resulting in sub-optimal performance. To mitigate these issues, Convolutional Neural Networks (CNNs) were introduced as effective alternatives [6, 19, 26, 50]. CNNs excel at capturing position-invariant features, making them particularly suitable for tasks involving image processing. Given that audio spectrograms resemble single-channel images, CNNs have naturally adapted well to audio-based tasks, maintaining popularity as encoders in AAC systems to this day. More recently, Transformer-based architectures have gained prominence [27, 35]. Originally developed for natural language processing, Transformers are adept at modeling long-range relationships within sequences. Their capability to effectively manage long audio sequences and discern complex temporal patterns has established Transformers as a powerful option for AAC tasks.

On the decoding side, initial AAC research typically employed RNN-based sequence-to-sequence models [41, 48]. However, RNN decoders often struggled with capturing long-range dependencies between words in the generated captions. To overcome this limitation, attention mechanisms were introduced [51, 54], enhancing the decoder's ability to model long-term and global relationships more effectively. Furthermore, decoding strategies have been identified as critical factors influencing AAC performance [44, 49]. Specifically, beam search decoding was shown to consistently outperform greedy decoding [44], while recent studies [49] proved the superior performance of nucleus sampling under the SPIDEr-FL evaluation metric. These findings underscore the importance of selecting appropriate decoding techniques to optimize AAC performance.

2.2 Fast Fourier Transform

Fast Fourier Transform (FFT) has long been a fundamental tool in digital signal processing, particularly for analyzing the frequency components of sampled waveforms [3]. Since audio signals are inherently waveforms, the FFT is highly effective for audio processing tasks. It is used for key operations such as spectrogram computation, convolution for filtering, and correlation, all of which are essential in audio analysis [3]. Due to its efficiency in computing the Discrete Fourier Transform, FFT has become widely adopted in various audio applications, including speech recognition [34, 37, 40], audio captioning [6, 25], and audio classification [8, 55]. These studies show that FFT is a powerful and commonly used tool for extracting frequency features from audio signals. In this paper, we utilize the FFT to inject useful frequency-based properties into audio features.

2.3 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) enhances language models by incorporating relevant context from an external knowledge base during the generation process. In audio captioning, recent works such as DRCAP [30] and RECAP [17] utilize RAG to improve caption quality by retrieving semantically similar audio-text pairs. P2PCAP [4] further advances this by introducing Generative Pair-to-Pair Retrieval, which uses both audio and its generated caption as retrieval queries, and Refined Knowledge Base filtering to ensure high-quality, context-aligned retrieval. These methods address challenges like audio ambiguity and demonstrate the importance of retrieval quality in audio-language alignment. However, while RAG-based methods have shown strong performance, their reliance on external datastores or knowledge bases can be impractical in real-world applications, especially in scenarios with limited storage, restricted latency budgets, or privacy concerns. Therefore, reducing this reliance remains an important challenge in advancing robust and efficient audio captioning systems.

2.4 Knowledge Distillation

Knowledge distillation (KD) is a model compression technique that transfers knowledge from a large, high-capacity "teacher" model to a more compact "student" model [20]. Widely adopted in deep learning, KD enables the deployment of efficient models on resource-constrained devices while maintaining comparative performance. In recent AAC research, large-scale models have demonstrated superior captioning capabilities but face deployment challenges due to their size and computational demands. To address this, prior work [52] has utilized knowledge distillation to train a compact model, employing EfficientNet-B2 [45] as the audio encoder and a shallow 2-layer Transformer as the text decoder, achieving performance comparable to the teacher model with HTSAT [5] and

BART [28]. While [52] has explored distillation using standard encoder-decoder architectures, our work introduces a distinct strategy: a RAG-based teacher model with retrieval guides the training of a retrieval-free student model. Unlike the prior KD method, our framework shares the same model for both teacher and student branches in training. Our teacher injects external knowledge during training while simultaneously learning from its own outputs, and the student learns to produce high-quality captions guided by the teacher without ever accessing the external datastore. By doing so, our model achieves deeper audio understanding, enabling it to better capture the semantic nuances of audio inputs.

3 Methodology

In this section, we first describe a popular framework that is often used in AAC problems. Next, we present our proposed method, including Fast Fourier Transform (FFT) adapter, RAG-based training strategy, and Distillation-based Distribution Alignment (DDA) module. Finally, we propose the final training objective of the DistillCaps framework and demonstrate the advantages of our framework. An overview of the proposed method is provided in Figure 3.

3.1 AAC's Framework Recap

A commonly adopted framework for AAC problem comprises three main components, as illustrated in Figure 2a: (a) Audio Feature Extraction: An audio encoder—such as PANNs [26], HTS-AT [5], BEATs [7], or CED [10]—is employed to extract acoustic features from raw audio inputs, which may include spoken language, environmental sounds, or other audio events. (b) Audio-Language Projection: The extracted features are then passed through a projection layer—typically implemented as either a Multi-Layer Perceptron (MLP) or a Q-former [29] module, followed by an MLP layer, which maps them into a representation space interpretable by an LLM. (c) Text Decoder: The projected features after projection into the word embedding space are subsequently fed into an LLM and use autoregressive loss to optimize (in training) or generate descriptive captions (in inference), utilizing the capabilities of language models such as BART [28], GPT-2 [39], or LLaMA [46].

Specifically, given an audio captioning dataset containing N audio-instruction-caption triples $D = \{audio_i, instruct_i, cap_i\}_{i=1}^{N}$, where instruction can be: "Describe the detail of this audio", the framework utilizes the combination of audio encoder and LLM to minimize the following objective:

$$\min_{\theta} \mathcal{L}^{auto} = \frac{1}{N} \sum_{i=1}^{N} -\log p_\theta(cap_i | audio_i, instruct_i). \quad (1)$$

In this framework, the audio encoder remains frozen, while the projection layer is learnable, and the LLM is fine-tuned using Low-Rank Adaptation (LoRA) [22].

3.2 DistillCaps

3.2.1 Fast Fourier Transform Adapter. After extraction by the audio encoder, the audio features are fed into an FFT adapter before being forwarded to the projection layer by utilizing the time order in the embeddings. This FFT adapter allows the model to learn and manipulate meaningful frequency patterns such as pitch, tone, and rhythm. In general, the adapter first transforms the audio feature sequences into the frequency domain using the Fast Fourier Transform (FFT). A learnable linear transformation is then applied as an adaptive filter that emphasizes or suppresses specific frequency components relevant to the task by adjusting both amplitude and phase through learning the real and imaginary components. Finally, the filtered frequency signal is converted back to the time domain using inverse FFT (IFFT).

Table 1: Retrieved vs. Ground Truth Captions.

Retrieved Captions	1: A car revving its engine while stopped. 2: A car is accelerating, then throttles down smoothly. 3: An engine accelerating is making room sounds.
Ground Truth	An engine hums as it idles.
Retrieved Captions	1: A loud pop followed by hissing and spraying. 2: Loud hissing then a burst. 3: There is hissing and then a loud pop.
Ground Truth	Short spray followed by louder, longer spray.

Specifically, we denote by $\mathcal{F}(\cdot)$ a FFT and $\mathcal{F}^{-1}(\cdot)$ as the inverse of it. The audio feature sequences, denoted as $f \in \mathbb{R}^{n \times d}$ where n is the sequence length and d is the feature dimension, are first transformed into the frequency domain as follows:

$$z = \mathfrak{R}_a + i \cdot \mathfrak{I}_a = \mathcal{F}(f), \quad (2)$$

where $\mathfrak{R}_a$ and $\mathfrak{I}_a$ are real and imaginary components, respectively. Then, a learnable linear transformation is applied to both real and imaginary parts, independently:

$$\hat{\mathfrak{R}_a} = \mathfrak{R}_a W + b, \quad \hat{\mathfrak{I}_a} = \mathfrak{I}_a W + b. \quad (3)$$

This transformation functions as an adaptive filter, emphasizing or suppressing specific frequency components related to the task. Finally, an inverse FFT is applied to recover the time-domain feature from the filtered frequency signal:

$$\hat{z} = \hat{\mathfrak{R}_a} + i \cdot \hat{\mathfrak{I}_a}, \quad (4)$$

$$\hat{f} = \mathcal{F}^{-1}(\hat{z}) \in \mathbb{R}^{n \times d} \quad (5)$$

This proposed module allows the model to capture global temporal patterns of audio features effectively while enhancing the robustness to time-domain noise through adaptive filtering in the frequency domain. Overall, the FFT adapter provides an interpretable way to inject frequency-awareness into audio features before projecting these filtered feature sequences into the word embedding space of the LLM.

3.2.2 RAG-based Training Strategy. Instead of feeding only audio features into the LLM for captioning, we first define a datastore before training, then retrieve top-k corresponding captions similar to the input audio from the datastore, and attach them to the instruction prompt in training and inference. Since this paper focuses on the training strategy to retain RAG's contextual benefit without its deployment overhead, we follow the retrieval module from the previous work [17].

Specifically, before training an AAC framework, we construct a datastore $DS = \{cap_i\}_{i=1}^{M}$ comprising a large set of example audio captions, where M is the size of the datastore. In our experiments, we use the training set as the datastore. Next, we leverage the CLAP encoder to map audio and text into a shared embedding space, as it

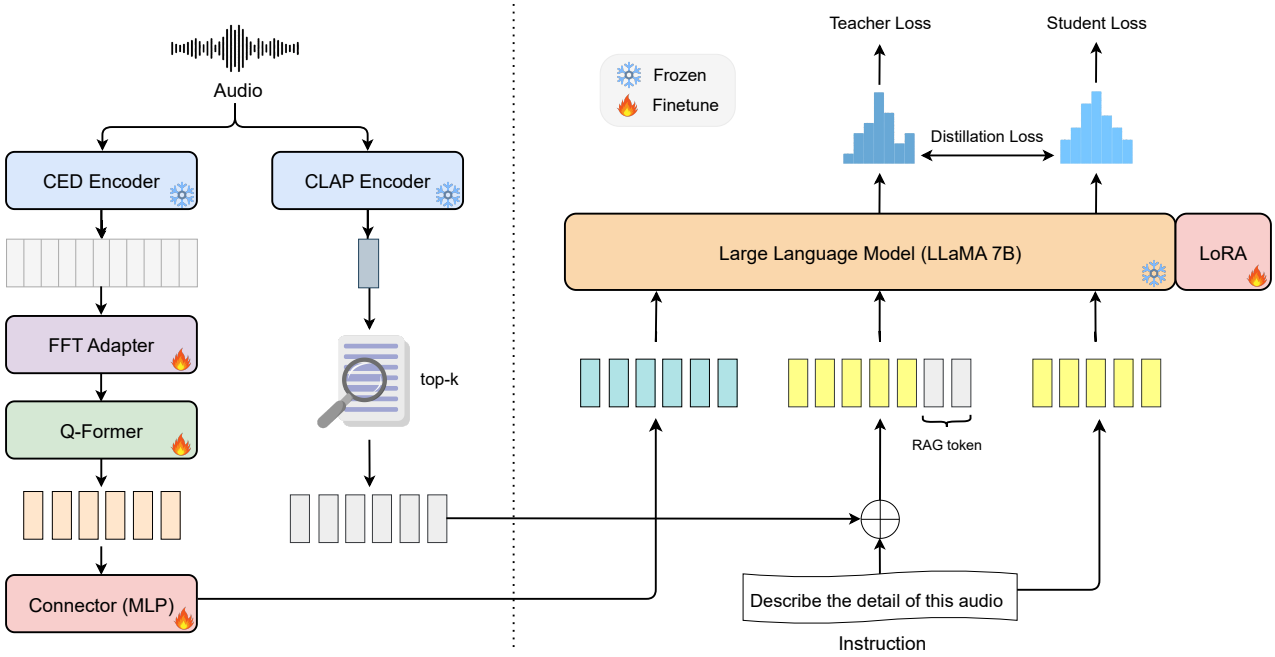

Figure 3: The proposed DistillCaps framework. *(Left)* Given an audio input, the CLAP encoder retrieves the top-k similar captions from a datastore. The audio is then encoded by the CED encoder, followed by an FFT adapter that enhances feature quality. The refined features are compressed by a Q-Former into a fixed number of tokens, then projected into the LLM's word embedding space via an MLP connector. *(Right)* The LLM component processes two branches: one with RAG input (teacher) and one without (student). The outputs are supervised by a distillation loss—where the RAG-based branch guides the RAG-free branch—alongside individual auto-regressive losses for both.

outperforms most prior models in audio-text retrieval tasks, making it well-suited for our framework. All embeddings of captions in the datastore and the given audio input a_i are encoded by CLAP's text encoder $\mathcal{TE}(.)$ and audio encoder $\mathcal{AE}(.)$ as follows:

$$z_{c,i} = \mathcal{TE}(cap_i), \quad z_{a,i} = \mathcal{AE}(a_i), \tag{6}$$

resulting in a vector datastore $DS_z = \{z_{c,1}, z_{c,2}, ..., z_{c,M}\}$ and the audio feature sequence $z_{a,i} \in R^{n \times d}$ where n is the sequence length and d is the feature dimension.

Based on this, the top-k retrieval captions TopK(a, DS, k) for audio input a_i are:

$$topk_i = \text{TopK}(a_i, DS, k) = \underset{\substack{S \subset DS \\ |S|=k}}{\arg\max} \sum_{c_j \in S} \text{sim}(a_i, c_j), \tag{7}$$

$$sim(a_i, c_j) = cosine(z_{a,i}, z_{c,j}). \tag{8}$$

These top-k captions are then attached to the instruction prompt with the template as shown in Figure 4. For efficiency, the RAG-based branch is co-trained rather than trained separately for response distribution alignment. With this setup, the minimization objective is as follows:

$$\min_{\theta} \mathcal{L}^{rag} = \frac{1}{N} \sum_{i=1}^{N} -\log p_\theta(cap_i | audio_i, instruct_i, topk_i). \tag{9}$$

The rationale behind RAG's benefit in this setting is that it provides additional information related to the audio input. This helps bridge the semantic gap between audio signals and language representations, allowing LLM to understand the audio input better

RAG-based Instruction Prompt Template

```
Similar audios sound like:
 - caption 1
 - caption 2
 - caption 3
 - ...
 - caption k
Describe the details of this audio:
<AcousticTokens>
- - -
Detailed:
```

Figure 4: Instruction prompt template after using RAG

even when the alignment between audio and language is not strong. Some examples are shown in Table 1.

3.2.3 Distillation-based Distribution Alignment. A key intuition behind employing RAG in the AAC task is that it provides additional audio-relevant information, thereby assisting the LLM in better understanding the input and generating more accurate captions. While effective, integrating RAG directly during training necessitates the presence of a retrieval datastore during inference. Without RAG at inference time, the model may exhibit performance degradation or bias, as it has been conditioned during training to rely on the additional information provided by RAG (Table 5 in Ablation study). However, deploying RAG-based systems introduces some challenges, such as a suitable datastore may not always be available during inference, and frequent access to the datastore (retrieval)

can potentially slow down the AAC system. A natural question arises: Can we improve the audio understanding of LLMs without relying on RAG in inference time? The answer is **Yes**. Instead of only explicitly learning RAG-based Audio Captioning, we propose a new additional module named Distillation-based Distribution Alignment (DDA) that implicitly enhances the alignment from audio features to language space and learns the desired captions for a RAG-free model.

The idea is that the framework comprises two input-output pathways, as illustrated in detail in Figure 3: a teacher branch that incorporates RAG in the input and a student branch that operates without RAG. The teacher branch guides the student branch in the training process through a distillation-based mechanism. Specifically, we minimize the Kullback-Leibler (KL) divergence between the response distributions of the RAG-free branch and the RAG-based branch:

$$\min_{\theta} \mathcal{L}^{DDA} = \frac{1}{N} \sum_{i=1}^{N} D_{KL}(p_\theta(cap_i|topk_i)||p_\theta(cap_i)), \tag{10}$$

where $p_\theta(cap_i) = p_\theta(cap_i|audio_i, instruct_i)$ and $p_\theta(cap_i|topk_i) = p_\theta(cap_i|audio_i, instruct_i, topk_i)$ for simplicity. The DDA module enables the model to align the output distribution of the RAG-free branch with that of the RAG-based branch, thereby enhancing the alignment between audio and language. This is achieved by encouraging the RAG-free model to mimic the behavior of the RAG-based counterpart.

The proposed DDA module in (10) is optimized along with the objectives in (1) and (9). As we aim to learn a RAG-free model from a RAG-based model rather than the other way around, we detach the gradient of the distribution logits in the RAG-based branch $p_\theta(cap_i|topk_i)$ when computing the optimization problem mentioned in (10).

3.3 Training Objective

Given local objectives from proposed strategies, we present the final DistillCaps framework objective to enhance audio understanding of LLM via Distilled Retrieval-Augmented Generation as follows:

$$\mathcal{L}^{DistillCaps} = \mathcal{L}^{auto} + \mathcal{L}^{rag} + \mathcal{L}^{DDA}. \tag{11}$$

The DistillCaps framework offers several advantages for the Audio Captioning (AAC) task. One major benefit is that it implicitly enhances the alignment between the audio and language spaces through a distillation-based mechanism, enabling the model to comprehend audio inputs better. This yields notable performance gains in AAC systems, even when RAG is not used during inference.

Furthermore, when RAG is present at inference time, DistillCaps substantially outperforms not only conventional RAG-based AAC systems but also other strong audio captioning models, due to its deeper audio understanding. Additionally, the framework can be integrated into a wide range of AAC architectures to strengthen the alignment between audio and language, thereby helping the LLM to better understand the semantics of audio inputs and learn the distribution of desired responses.

4 Experiments

To highlight the advantages of our proposed framework, we conduct a series of audio captioning experiments on standard AAC benchmarks. It is important to note that the DistillCaps framework can be compatible with various AAC architectures. To demonstrate its effectiveness, we integrate it into the LOAE [32] baseline for our experiments. Sections 4.1, 4.2, and 4.3 provide a detailed description of our experimental setup, including the datasets, baseline methods, and metrics used for comparison, respectively. In Section 4.4, we provide the implementation details for our proposed framework, while the main results are presented in Section 4.5. Additionally, we conduct an ablation study to investigate the contributions of individual components within our model in Section 4.6.

4.1 Datasets

We evaluate our method on two widely used audio captioning benchmarks: AudioCaps [11] and Clotho [13].

AudioCaps is a large-scale audio captioning dataset constructed from AudioSet [16], comprising audio clips paired with human-written captions. Each clip is approximately 10 seconds long. While each training clip is annotated with a single caption, the validation and test clips are annotated with five captions each. The dataset includes 49838 audio-caption pairs for training, 495 for validation, and 975 for testing, totaling 51308 samples.

Clotho is an audio captioning dataset sourced from the Freesound platform. Each audio sample is 15-30 seconds long and annotated with five captions, each containing 8-20 words. The dataset contains 3839 audio-caption pairs for training, 1045 for validation, and 1045 for testing, totaling 5929 samples.

4.2 Baselines

To evaluate our model's effectiveness, we compare it against several baseline models in both in-domain and out-of-domain settings.

In-domain baselines include models trained and evaluated on the same dataset. We compare our model with recent competitive audio captioning systems on the AudioCaps and Clotho datasets. BART-tags [18] conditions a BART decoder on predefined AudioSet tags. CNN10-AT [50] applies transfer learning by leveraging features from Audio Tagging. Both Prefix AAC [24] and Pengi [9] utilize prefix tuning on a frozen GPT-2. LHDFF [42] and PFCA-Net [43] fuse multi-scale audio features via a Residual PANNs encoder and a pyramid encoder, respectively. EnCLAP [23] uses EnCodec for acoustic representation and uses masked codec modeling to enhance BART's audio awareness; its contrastive learning (CL) extension further improves audio-text alignment. LOAE [32] uses LoRA [21] to fine-tune the audio encoder and text decoder. And, LOAE with RAG incorporates retrieval-augmented generation to enhance decoding with external information (RAG-based LOAE). Finally, RECAP [17] and DRCap [30] retrieve semantically similar captions using CLAP [15] to enrich the decoder with additional context. Note that DistillCaps with RAG, LOAE with RAG, and RECAP all employ a datastore constructed from the training split of the dataset.

Out-of-domain baselines involve training on one dataset and evaluating on another to assess cross-domain generalization. Specifically, we consider models trained on AudioCaps and tested on Clotho, and vice versa. This setting reflects real-world scenarios

Table 2: In-domain evaluation results on the Clotho and AudioCaps datasets.

Model	Clotho Evaluation							AudioCaps Evaluation						
	BL_1	BL_4	MT	RG_L	CD	SC	SD	BL_1	BL_4	MT	RG_L	CD	SC	SD
BART-tags [18]	50.6	13.4	14.8	33.8	27.8	9.2	18.5	69.9	26.6	24.1	49.3	75.3	17.6	46.5
CNN10-AT [50]	55.6	15.9	16.9	36.8	37.7	11.5	-	65.5	23.1	22.9	46.7	66	16.8	-
Prefix AAC [24]	56	16	17	37.8	39.2	11.8	25.5	71.3	30.9	24	50.3	73.3	17.7	45.5
Pengi [9]	57	15	17.2	37.5	41.6	12.6	27.1	69.1	25.3	23.2	48.2	75.2	18.2	46.7
LHDFF [42]	57	15.9	17.5	37.8	40.8	12.2	26.5	67.4	26.7	23.2	48.3	68	17.1	42.6
RECAP [17]	56.3	16.5	17.9	38.3	39.8	12.2	21.4	72.1	**31.6**	25.2	**52.1**	75	18.3	47.2
PFCA-Net [43]	56.4	16	17.4	37.5	40.1	12.3	26.2	67.8	26.8	23.4	48.6	69.8	17.3	43.6
EnCLAP [23]	-	-	18.2	38	41.7	13	27.3	-	-	25.4	50	77	18.6	48
EnCLAP + CL	-	-	18.5	37.6	40.5	13.1	27.1	-	-	25.7	49.6	76.8	**19**	48.1
DRCap [30]	-	-	18.2	-	43.8	13.3	28.5	-	-	25.3	-	70.5	18	44.2
LOAE [32]	54.3	13.1	17.1	36.4	34.3	11.6	22.9	71.7	27.1	25.2	49.1	75.3	18.2	46.9
LOAE w/ RAG	57.3	15.5	17.7	37.6	41.2	12.2	26.8	71.8	28.1	25.2	49.9	76.8	18.4	47.5
DistillCaps w/o RAG	58.1	15.7	17.8	38.3	41.7	12.7	27.2	72.1	28.6	25.4	50	76	18.6	47.3
DistillCaps w/ RAG	**58.6**	**16.7**	**18.7**	**39.4**	**45.6**	**13.4**	**29.4**	**73.4**	27.3	**25.8**	50.9	**78.2**	18.9	**48.5**

Metrics: BLEU-1 (BL_1), BLEU-4 (BL_4), METEOR (MT), ROUGE-L (RG_L), CIDEr (CD), SPICE (SC), SPIDEr (SD).

where domain shift affects model performance. Compared models include ACT [35], a fully Transformer-based encoder-decoder model, along with BART-tags [18], Prefix-AAC [24], EnCLAP-large [23], and RECAP [17]. In this setting, our model with RAG, LOAE with RAG, and RECAP still employs a datastore built from the training split of the dataset, similar to the in-domain setting.

4.3 Evaluation Metrics

To evaluate the quality of generated audio captions, we adopt standard captioning metrics including BLEU [38], METEOR [2], ROUGE-L [31], CIDEr [47], SPICE [1], and SPIDEr [33]. BLEU-n measures n-gram precision by calculating the overlap between generated and reference captions. METEOR measures unigram matches and also considers word stems, synonyms, and paraphrases, while ROUGE-L evaluates the longest common subsequence between captions. CIDEr scores based on how often their n-grams appear in reference captions. SPICE analyzes captions through scene graphs to assess semantic content. Lastly, SPIDEr is computed by taking the average of CIDEr and SPICE scores, balancing consensus and semantics for a comprehensive evaluation. These metrics are computed using the publicly available aac-metrics library. Higher scores across these metrics indicate better captioning performance.

4.4 Implementation Details

4.4.1 Model Architecture Configuration. Similar to LOAE [32], we use the LLaMA-7B large language model (LLM) as the text decoder component. To extract audio feature sequences and to bridge the modality gap between audio and language, we leverage CED (base) as an audio encoder and Q-Former as a pooling module, followed by an MLP projection, respectively. Our model is trained for 50 epochs. The training configuration includes a total batch size of 64, and the AdamW optimizer is used with a learning rate of 5e-6 and weight decay of 1e-6. In both datasets, we use the sample rate of 16000. The maximum lengths for AudioCaps and Clotho are 10 and 30 seconds for each dataset. The temperature for distillation loss is 1.0 for both datasets. In the model, the audio encoder is frozen, while the FFT adapter and projection layer are learnable. The LLM is fine-tuned using LoRA.

4.4.2 Running-time. We train DistillCaps using 4 NVIDIA A100 PCIe GPUs with 80 GB memory each, and complete the training process for AudioCaps in 22 hours and for Clotho in 11.5 hours.

4.5 Main Results

4.5.1 Impact of DistillCaps in In-Distribution Setting. Applying the DistillCaps framework significantly enhances AAC performance in the in-distribution setting. All models are trained on the AudioCaps (or Clotho) training set and tested on the AudioCaps (or Clotho) testing set, respectively. Table 2 summarizes the detailed evaluation results where all models are trained and tested on either the AudioCaps or the Clotho datasets. Without utilizing RAG during inference, our DistillCaps framework achieves notable improvements over the baseline, with gains ranging from 0.7% to 7.4% on Clotho and 0.2% to 1.5% on AudioCaps across various metrics. Specifically, on the Clotho dataset, our method achieves CIDEr and SPIDEr scores of 41.7% and 27.2%, respectively, compared to baseline methods, clearly demonstrating the substantial performance benefits of our approach. Moreover, our framework is competitive with, and in many cases outperforms, methods that rely on RAG-based training strategies on the Clotho dataset, while remaining highly competitive on AudioCaps. For example, it surpasses the RAG-based LOAE variant by 0.7% on ROUGE-L and 0.4% on SPIDEr on Clotho. Additionally, it achieves a substantial improvement of 5% on SPIDEr compared to RECAP on the same dataset.

4.5.2 Enhanced Performance with RAG During Inference. When we further integrate RAG during inference of our method, it further enhances the advantages of our DistillCaps framework, significantly outperforming baselines that also use RAG strategies (see Table 2). Notably, on the Clotho dataset, DistillCaps surpasses the baseline

Table 3: Out-domain evaluation results on the Clotho and AudioCaps datasets.

Model	AudioCaps → Clotho							Clotho → AudioCaps						
	BL_1	BL_4	MT	RG_L	CD	SC	SD	BL_1	BL_4	MT	RG_L	CD	SC	SD
ACT [35]	29.4	4.3	9.6	23.9	11.7	5	8.4	41.5	6.3	13.4	30.3	14.9	6.6	10.7
BART-tags [18]	30.9	3.4	9.8	23.3	11.2	4.6	7.9	42.5	6.1	12.8	29.8	14.7	6	10.4
Prefix AAC [24]	34.2	6.5	11.2	27.6	19.2	7.4	13.3	44.9	8.4	14.4	33	21.1	8.3	14.7
EnCLAP-large [23]	-	-	11.1	-	13.8	5.9	9.9	-	-	13.3	-	17.4	8.0	12.6
RECAP [17]	33.9	6.8	11.0	27.6	19.5	8.4	13.7	42.7	6.5	11.2	28.1	19.1	7.8	13.6
LOAE w/ RAG	37.1	6.5	12.3	28.2	18.9	7.6	13.2	48.9	10.3	17.6	36.2	31.9	11.3	21.7
DistillCaps w/ RAG	**38.5**	**8.3**	**12.9**	**29.7**	**23.4**	**8.9**	**16.1**	**52.3**	**11.2**	**18.2**	**37.6**	**32.9**	**11.4**	**22.1**

Table 4: Qualitative comparison between original LOAE and DistillCaps-enhanced versions (w/ and w/o RAG).

Ground Truth	1: A wind chime is making noise while people are talking in the background. 2: Ducks quack, and a faint tapping noise occurs as water runs in the background. 3: A man is speaking, and a crowd applauds. 4: An engine running.
LOAE	1: Wind chimes tinkle as birds chirp in the background. 2: Birds are chirping and a duck is quacking. 3: A man speaks followed by applause. 4: Loud continuous spraying.
LOAE w/ RAG	1: Wind chimes are ringing in the background as people are talking. 2: A duck is quacking in a pond while other ducks are quacking in the background. 3: A man speaks and a crowd applauds. 4: Loud hissing and vibrating.
DistillCaps w/o RAG	1: A wind chime is blowing in the wind while people are talking in the background. 2: A duck is quacking while water is flowing in the background. 3: A man is giving a speech and a crowd applauds. 4: An engine works nearby.
DistillCaps w/ RAG	1: Wind chimes are clanging in the background while people are talking. 2: A duck is quacking and water is flowing in the background. 3: A man is giving a speech and a crowd applauds. 4: An engine running consistently.

LOAE with RAG implementations by substantial margins, achieving improvements of 4.4% on CIDEr and 2.6% on SPIDEr metrics, with additional enhancements ranging from 1.0% to 1.8% on other evaluation metrics. Additionally, our method demonstrates state-of-the-art performance compared to prominent AAC models. For instance, DistillCaps achieves superior CIDEr, SPICE, and SPIDEr scores on Clotho by 1.8%, 0.1%, and 0.9%, respectively, compared to DRCap, a recent RAG-based AAC method. On AudioCaps, our method achieves higher METEOR and SPIDEr scores (25.8% and 48.5%, respectively) compared to the RECAP method, validating the effectiveness of DistillCaps.

4.5.3 Qualitative Improvements. Qualitative comparisons between baseline and DistillCaps-enhanced versions (with and without RAG) are detailed in Table 4, further illustrating the practical improvements offered by our proposed framework. For example, we can observe that in caption 1, LOAE fails to capture the background sound of people talking, instead incorrectly describing bird chirping. In caption 2, LOAE and LOAE with RAG only capture the duck's quacking, and they misinterpret the faint tapping sound as bird chirping and additional duck quacking, respectively. We can see that even with retrieval, LOAE struggles to capture these subtle audio details. In contrast, both DistillCaps with and without RAG successfully identify the duck sound and water flow. Additionally, in caption 4, while LOAE and LOAE with RAG describe only the sound, DistillCaps models go further by capturing the subject and conveying the full semantic meaning of the scene.

4.5.4 Robustness of DistillCaps in Out-of-Distribution Setting. The robustness and domain-transfer capabilities of the DistillCaps framework are clearly demonstrated in out-of-distribution scenarios (see Table 3). Note that with methods that use RAG in inference, the datastore is collected from the training set of the training data. In evaluations involving models trained on AudioCaps and tested on Clotho (and vice versa), our DistillCaps method consistently outperforms baseline implementations utilizing a RAG-based training strategy. Notably, as compared to the method RAG-based LOAE, our method achieves improvements of 4.5% and 2.9% on CIDEr and SPIDEr metrics, respectively, on the Clotho dataset, and further gains of 1.0% and 0.4% on AudioCaps. Moreover, our method sets new state-of-the-art benchmarks across all evaluation metrics, surpassing existing methods such as RECAP and Prefix AAC by significant margins. Specifically, on Clotho, our method surpasses RECAP by 2.1%, 3.9%, and 2.4% in ROUGE-L, CIDEr, and SPIDEr metrics, respectively, while outperforming Prefix AAC on AudioCaps by 4.6%, 11.8%, and 7.4% on the same metrics. These results underline the strong generalization and effectiveness of our DistillCaps framework.

4.6 Ablation Study

We conducted an extensive ablation study to evaluate the contribution of each component, including the benefits of the FFT adapter, the DDA component, and inference time analysis. For fairness, we compare models with identical parameters and complexity.

4.6.1 Distillation-based Distribution Alignment. Table 5 highlights the effectiveness of the DDA component in improving the LLM's understanding of audio for AAC tasks. It enhances audio-language alignment by guiding the RAG-free branch distribution (student) to

Table 5: Ablation study on Distillation and RAG on the Clotho and AudioCaps datasets.

Distillation	RAG		Clotho Evaluation							AudioCaps Evaluation						
	Train	Test	BL_1	BL_4	MT	RG_L	CD	SC	SD	BL_1	BL_4	MT	RG_L	CD	SC	SD
✗	✗	✗	54.3	14.1	17.3	36.6	35	11.5	23.3	72.3	27.3	25.2	49.2	75.3	18.3	46.8
✗	✓	✗	49.2	10.3	15	33.3	23.6	9.3	16.4	68.5	23.6	23.9	47.4	66.6	17.5	42.1
✗	✓	✓	57.4	15.7	17.8	38.2	41.8	12.5	27.3	72.3	28.1	25.3	50.1	76.7	18.5	47.5
✓	✓	✗	58.1	15.7	17.8	38.3	41.7	12.7	27.2	72.1	**28.6**	25.4	50	76	18.6	47.3
✓	✓	✓	**58.6**	**16.7**	**18.7**	**39.4**	**45.6**	**13.4**	**29.4**	**73.4**	27.3	**25.8**	**50.9**	**78.2**	**18.9**	**48.5**

align with the distribution of the RAG-based branch (teacher). This guidance leads to better performance on AAC tasks. For instance, after incorporating the DDA module, our method without using RAG in inference outperforms the setting that omits both DDA and the RAG-based training strategy (rows 1 and 4). It also achieves performance comparable to the setting that uses the RAG-based training strategy without DDA (rows 3 and 4). Furthermore, when RAG is used during inference with a datastore built from the training set, the DDA component yields the best overall performance (row 5). Additionally, we find that using RAG-based training strategy without RAG during inference (row 2)—which can occur in real applications due to the absence of an appropriate datastore—results in a significant performance drop. This setting performs worse than both the RAG-inference setup (row 3) and even models trained entirely without RAG (row 1). These results demonstrate the DDA component's effectiveness in aligning the distribution of the RAG-free model with that of the RAG-based model, thereby enhancing the alignment between the audio and language spaces, improving audio understanding of LLM, and lessening the dependency on RAG during inference.

Table 6: Ablation of FFT on Clotho.

Method	BL_1	BL_4	MT	RG_L	CD	SC	SD
w/o FFT	**58.7**	16.3	18.2	38.8	44.6	13.2	28.9
w/ FFT	58.6	**16.7**	**18.7**	**39.4**	**45.6**	**13.4**	**29.4**

Table 7: Ablation of FFT on AudioCaps.

Method	BL_1	BL_4	MT	RG_L	CD	SC	SD
w/o FFT	72.5	27	25.3	49.9	76.1	18.7	47.4
w/ FFT	**73.4**	**27.3**	**25.8**	**50.9**	**78.2**	**18.9**	**48.5**

4.6.2 Fast Fourier Transform Adapter. To evaluate the effectiveness of the FFT adapter in our framework, we compare the performance of DistillCaps with and without the FFT adapter. As shown in Tables 6 and 7, incorporating the FFT adapter consistently improves performance across both benchmarks. This demonstrates that injecting frequency-awareness into audio features—prior to projecting these filtered feature sequences into the LLM's word embedding space—enables the model to better capture global temporal patterns. Additionally, the FFT adapter enhances robustness to time-domain noise through adaptive filtering in the frequency domain, which helps to improve the performance of the model on AAC tasks.

4.6.3 Analysis on Inference Time. The bar chart in Figure 5 presents the inference times for the Clotho and AudioCaps benchmarks, comparing the processing speeds of DistillCaps (without RAG in inference) and the RAG-based LOAE. For the Clotho dataset, DistillCaps achieves an inference time of 7.1 minutes, significantly

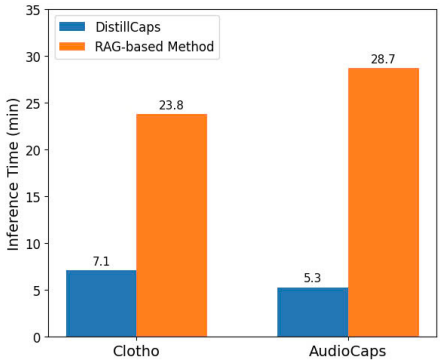

Figure 5: Inference time comparison on the Clotho and AudioCaps datasets.

lower than the 23.8 minutes required by the RAG-based method, indicating a speed improvement of over three times. Similarly, on the AudioCaps dataset, DistillCaps completes the process in 5.3 minutes, while the RAG-based method takes 28.7 minutes—more than five times longer. This consistent trend across both datasets suggests that DistillCaps is a more time-efficient solution, which could be particularly beneficial in scenarios where rapid processing is essential.

5 Conclusion

In this work, we introduced DistillCaps, a novel framework that enhances automated audio captioning (AAC) by leveraging the benefits of retrieval-augmented generation (RAG) during training while minimizing its deployment overhead at inference. By distilling knowledge from a RAG-based teacher into a retrieval-free student model, DistillCaps effectively improves audio-language alignment and overall captioning performance without the latency and complexity of retrieval at test time. Our use of an FFT adapter further boosts representation quality, contributing to the model's strong results. Extensive experiments on AudioCaps and Clotho benchmarks demonstrate that DistillCaps matches or surpasses prior systems, and outperforms when retrieval is enabled at inference time, showing the effectiveness of RAG-guided distillation for efficient and accurate AAC.

Acknowledgements

This research was supported by the Lee Kong Chian Fellowship and research funding from the Faculty of Information Technology, University of Science, Vietnam National University, Ho Chi Minh City.

GenAI Usage Disclosure

We used Grammarly and ChatGPT to improve the spelling, grammar, punctuation, and clarity of our paper. While generative AI tools were utilized during the writing process, the authors are fully accountable for the content.

References

[1] Peter Anderson, Basura Fernando, Mark Johnson, and Stephen Gould. 2016. Spice: Semantic propositional image caption evaluation. In *European Conference on Computer Vision (ECCV)*. Springer, 382–398.

[2] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 65–72.

[3] Glenn D. Bergland. 1969. A guided tour of the fast Fourier transform. *IEEE Spectrum* 6, 7 (1969), 41–52. doi:10.1109/MSPEC.1969.5213896

[4] Choi Changin, Lim Sungjun, and Rhee Wonjong. 2024. Audio Captioning via Generative Pair-to-Pair Retrieval with Refined Knowledge Base. *arXiv preprint arXiv:2410.10913* (2024).

[5] Ke Chen, Xingjian Dum, Bilei Zhu, Zejun Ma, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. 2022. HTS-AT: A hierarchical token-semantic audio transformer for sound classification and detection. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 646–650.

[6] Kun Chen, Yusong Wu, Ziyue Wang, Xuan Zhang, Fudong Nian, Shengchen Li, and Xi Shao. 2020. Audio Captioning Based on Transformer and Pre-Trained CNN. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*. 21–25.

[7] Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, and Furu Wei. 2022. Beats: Audio pre-training with acoustic tokenizers. *arXiv preprint arXiv:2212.09058* (2022).

[8] Wei Chu and Benoît Champagne. 2008. A Noise-Robust FFT-Based Auditory Spectrum With Application in Audio Classification. *IEEE Transactions on Audio, Speech, and Language Processing* 16, 1 (2008), 137–150. doi:10.1109/TASL.2007.907569

[9] Soham Deshmukh, Benjamin Elizalde, Rita Singh, and Huaming Wang. 2023. Pengi: An Audio Language Model for Audio Tasks. In *Advances in Neural Information Processing Systems*, Vol. 36. Curran Associates, Inc., 18090–18108. https://proceedings.neurips.cc/paper_files/paper/2023/file/3a2e5889b4bbef997ddb13b55d5acf77-Paper-Conference.pdf

[10] Heinrich Dinkel, Yongqing Wang, Zhiyong Yan, Junbo Zhang, and Yujun Wang. 2024. CED: Consistent Ensemble Distillation for Audio Tagging. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 3. 291–295. doi:10.1109/ICASSP48485.2024.10446348

[11] Kim Chris Dongjoo, Kim Byeongchang, Lee Hyunmin, and Kim Gunhe. 2019. AudioCaps: Generating Captions for Audios in The Wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Burstein Jill, Doran Christy, and Solorio Thamar (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 119–132. doi:10.18653/v1/N19-1011

[12] Konstantinos Drossos and Sharath Adavanne andTuomas Virtanen. 2017. Automated audio captioning with recurrent neural networks. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. doi:10.1109/WASPAA.2017.8170058

[13] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. 2020. Clotho: an Audio Captioning Dataset. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 736–740. doi:10.1109/ICASSP40776.2020.9052990

[14] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. 2022. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438* (2022).

[15] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. 2023. CLAP Learning Audio Concepts from Natural Language Supervision. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 1–5. doi:10.1109/ICASSP49357.2023.10095889

[16] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, and Channing Moore. 2017. Audio Set: An ontology and human-labeled dataset for audio events. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 776–780. doi:10.1109/ICASSP.2017.7952261

[17] Sreyan Ghosh, Sonal Kumar, Chandra K. R. Evuru, Ramani Duraiswami, and Dinesh Manocha. 2024. Recap: Retrieval-Augmented Audio Captioning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 1161–1165. doi:10.1109/ICASSP48485.2024.10448030

[18] Félix Gontier, Romain Serizel, and Christophe Cerisara. 2021. Automated audio captioning by fine-tuning bart with audioset tags. In *Workshop on Detection and Classification of Acoustic Scenes and Events*.

[19] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, and Channing Moore. 2017. CNN architectures for large-scale audio classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 131–135. doi:10.1109/ICASSP.2017.7952132

[20] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[21] Edward Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. *ICLR* 1, 2 (2022), 3.

[22] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR* 1, 2 (2022), 3.

[23] Jaeyeon Kim, Jaeyoon Jung, Jinjoo Lee, and Sang Hoon Woo. 2024. EnCLAP: Combining Neural Audio Codec and Audio-Text Joint Embedding for Automated Audio Captioning. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 6735–6739. doi:10.1109/ICASSP48485.2024.10446672

[24] Minkyu Kim, Kim Sung-Bin, and Tae-Hyun Oh. 2023. Prefix Tuning for Automated Audio Captioning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. doi:10.1109/ICASSP49357.2023.10096877

[25] Andrew Koh, Xue Fuzhao, and Chng Eng Siong. 2022. Automated Audio Captioning Using Transfer Learning and Reconstruction Latent Space Similarity Regularization. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 7722–7726. doi:10.1109/ICASSP43922.2022.9747676

[26] Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D. Plumbley. 2020. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020), 2880–2894.

[27] Thodoris Kouzelis, Grigoris Bastas, Athanasios Katsamanis, and Alexandros Potamianos. 2023. Efficient audio captioning transformer with patchout and text guidance. *arXiv preprint arXiv:2304.02916* (2023).

[28] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461* (2019).

[29] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*. PMLR, 19730–19742.

[30] Xiquan Li, Wenxi Chen, Ziyang Ma, Xuenan Xu, Yuzhe Liang, and Zhisheng Zheng. 2025. DRCap: Decoding CLAP Latents with Retrieval-Augmented Generation for Zero-shot Audio Captioning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 1–5. doi:10.1109/ICASSP49660.2025.10890325

[31] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[32] Jizhong Liu, Gang Li, Junbo Zhang, Heinrich Dinkel, Yongqing Wang, Zhiyong Yan, Yujun Wang, and Bin Wang. 2024. Enhancing automated audio captioning via large language models with optimized audio encoding. *arXiv preprint arXiv:2406.13275* (2024).

[33] Siqi Liu, Zhenhai Zhu, Ning Ye, Sergio Guadarrama, and Kevin Murphy. 2017. Improved image captioning via policy gradient optimization of spider. In *Proceedings of the IEEE international conference on computer vision*. 873–881.

[34] Weiqiang Liu, Qicong Liao, Fei Qiao, Weijie Xia, Chenghua Wang, and Fabrizio Lombard. 2019. Approximate Designs for Fast Fourier Transform (FFT) With Application to Speech Recognition. *IEEE Transactions on Circuits and Systems I: Regular Papers* 66, 12 (2019), 4727–4739. doi:10.1109/TCSI.2019.2933321

[35] Xinhao Mei, Xubo Liu, Qiushi Huang, Mark D. Plumbley, and Wenwu Wang. 2021. Audio captioning transformer. *arXiv preprint arXiv:2107.09817* (2021).

[36] Xinhao Mei, Xubo Liu, Mark D. Plumbley, and Wenwu Wang. 2022. Automated audio captioning: An overview of recent progress and new challenges. *EURASIP journal on audio, speech, and music processing* 2022, 1 (2022), 26.

[37] Alan V. Oppenheim. 1970. Speech spectrograms using the fast Fourier transform. *IEEE Spectrum* 7, 8 (1970), 57–62. doi:10.1109/MSPEC.1970.5213512

[38] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 311–318.

[39] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[40] Bhiksha Raj, Lorenzo Turicchia, Bent Schmidt-Nielsen, , and Rahul Sarpeshkar. 2007. An FFT-based companding front end for noise-robust automatic speech recognition. *EURASIP Journal on Audio, Speech, and Music Processing* 2007 (2007), 1–13.

[41] Ikawa Shota and Kashino Kunio. 2019. Neural Audio Captioning Based on Conditional Sequence-to-Sequence Model. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events Workshop (DCASE)*. 99–103. doi:10.33682/7bay-bj41

[42] Jianyuan Sun, Xubo Liu, Xinhao Mei, Volkan Kılıç, Mark D. Plumbley, and Wenwu Wang. 2023. Dual Transformer Decoder based Features Fusion Network for Automated Audio Captioning. In *Interspeech*. 4164–4168. doi:10.21437/Interspeech.2023-943

[43] Jianyuan Sun, Wenwu Wang, and Mark D. Plumbley. 2024. PFCA-Net: Pyramid Feature Fusion and Cross Content Attention Network for Automated Audio Captioning. In *Interspeech*.

[44] Daiki Takeuchi, Yuma Koizumi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino. 2020. Effects of word-frequency based pre-and post-processings for audio captioning. *arXiv preprint arXiv:2009.11436* (2020).

[45] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*. PMLR, 6105–6114.

[46] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[47] Ramakrishna Vedantam, Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 4566–4575.

[48] Mengyue Wu, Heinrich Dinkel, and Kai Yu. 2019. Audio Caption: Listen and Tell. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 830–834. doi:10.1109/ICASSP.2019.8682377

[49] Shih-Lun Wu, Xuankai Chang, Gordon Wichern, Jee weon Jung, François Germain, Jonathan L. Roux, and Shinji Watanabe. 2023. BEATs-based audio captioning model with INSTRUCTOR embedding supervision and ChatGPT mix-up. In *Proceedings of the Detection and Classification of Acoustic Scenes and Events Challenge (DCASE)*. 1–5.

[50] Xuenan Xu, Heinrich Dinkel, Mengyue Wu, Zeyu Xie, and Kai Yu. 2021. Investigating Local and Global Information for Automated Audio Captioning with Transfer Learning. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 905–909. doi:10.1109/ICASSP39728.2021.9413982

[51] Xuenan Xu, Heinrich Dinkel, Mengyue Wu, Zeyu Xie, and Kai Yu. 2021. Investigating Local and Global Information for Automated Audio Captioning with Transfer Learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 905–909. doi:10.1109/ICASSP39728.2021.9413982

[52] Xuenan Xu, Haohe Liu, Mengyue Wu, Wenwu Wang, and Mark D. Plumbley. 2024. Efficient Audio Captioning with Encoder-Level Knowledge Distillation. *arXiv preprint arXiv:2407.14329* (2024).

[53] Xuenan Xu, Zeyu Xie, Mengyue Wu, and Kai Yu. 2024. Beyond the Status Quo: A Contemporary Survey of Advances and Challenges in Audio Captioning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 32 (2024), 95–112. doi:10.1109/TASLP.2023.3321968

[54] Zhongjie Ye, Helin Wang, Dongchao Yang, and Yuexian Zou. 2021. Improving the performance of automated audio captioning via integrating the acoustic and semantic information. *arXiv preprint arXiv:2110.06100* (2021).

[55] Yixiao Zhang, Baihua Li, Hui Fang, and Qinggang Meng. 2022. Spectrogram Transformers for Audio Classification. In *IEEE International Conference on Imaging Systems and Techniques (IST)*. 1–6. doi:10.1109/IST55454.2022.9827729

[56] Ayşegül Özkaya Eren and Mustafa Sert. 2020. Audio Captioning Based on Combined Audio and Semantic Embeddings. In *IEEE International Symposium on Multimedia (ISM)*. 41–48. doi:10.1109/ISM.2020.00014