M1 Summer Internship Report

Lasha Koroshinadze

25 Aug, 2024

Contents

1 Introduction **2**

2 Presentation of the Host Organization **2**

3 Stable Audio Open **2**
3.1 Efficient Latent Diffusion Model Architecture . . . 3
3.2 Advanced Text Conditioning with CLAP Embeddings . . . 3
3.3 Superior Performance in Benchmarks and Real-World Applications . . . 4

4 Fine-Tuning of Stable Audio Open **4**

5 Converting CLAP to CLVAP **5**
5.1 Additional Exploration . . . 5
5.1.1 Encodec . . . 5
5.1.2 Descript Audio Codec . . . 6
5.2 BYOL-A Exploration . . . 6
5.3 Proposed Architecture . . . 6

6 HTS-AT: A Powerful Audio Encoder for CLAP Model Training **7**
6.1 Advantages of HTS-AT . . . 7
6.2 Architectural Overview . . . 8

7 Fine-tuning Original CLAP **8**

8 New CLVAP Architecture **10**
8.1 Approaches Explored . . . 10
8.1.1 Early Fusion Approach . . . 10
8.1.2 Late Fusion Approach . . . 11

9 Data Augmentation **11**
9.1 Caption Augmentation . . . 12
9.2 Transition to Cross-Attention Fusion Approach . . . 12
9.3 Training Modifications . . . 13

10 Training Results **14**

11 Contributions **15**

12 Research Perspectives and Future Work **16**

13 Conclusion **16**

References **17**

1 Introduction

This internship builds upon the work initiated during my TER, where the focus was on fine-tuning a Contrastive Language-Audio Pretraining (CLAP[1]) model to improve its ability to interpret and generate environmental sounds based on vocal imitations and textual descriptions. The objective of this internship is to further refine and extend the CLAP model by incorporating vocal imitations as an additional input modality, leading to the development of a multimodal model, CLVAP (Contrastive Language+Voice & Audio Pre-training). This approach aims to enhance the model's capability in audio generation and retrieval tasks, particularly for sounds that are difficult to describe textually.

Throughout the internship, I focused on expanding the dataset, fine-tuning the model, and addressing challenges in training stability, with the ultimate goal of advancing the state-of-the-art in audio AI by integrating vocal imitations into machine learning frameworks.

View the code on GitHub

2 Presentation of the Host Organization

The Laboratoire Interdisciplinaire des Sciences du Numérique (LISN) is a prominent research laboratory at Université Paris-Saclay, established in 2021 from the merger of LIMSI and LRI. It operates under the joint supervision of CNRS, Université Paris-Saclay, INRIA, and CentraleSupélec. LISN specializes in interdisciplinary research across digital sciences, with key focus areas including algorithms, data science, human-computer interaction, and language technologies.

During my internship, I have been working under the supervision of Marc Evrard and Tifanie Bouchara, associate professors at LISN. Marc Evrard specializes in speech synthesis and processing, while Tifanie Bouchara specializes in sonic interaction design. Their combined expertise has been instrumental in guiding the research and development of my project.

3 Stable Audio Open

Stable Audio Open[2] represents a significant advancement in the field of audio generation, offering a combination of efficiency, flexibility, and superior performance. Below, I detail the reasons why this model was chosen for fine-tuning, emphasizing its architectural innovations and technical superiority over other state-of-the-art (SOTA) models.

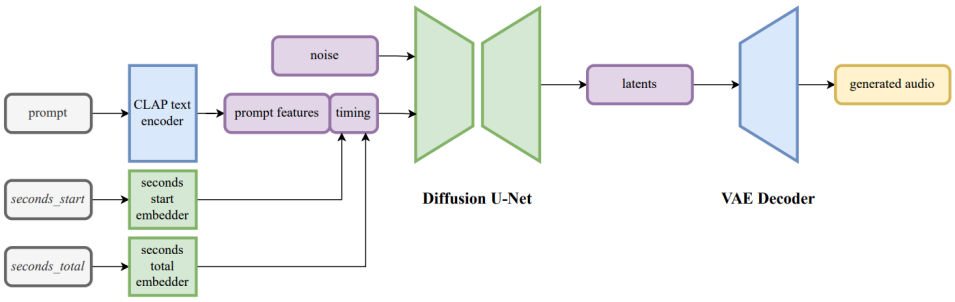


Figure 1: Stable Audio architecture. Blue: frozen pre-trained models. Green: parameters learnt during diffusion training. Purple: signals of interest.

3.1 Efficient Latent Diffusion Model Architecture

Stable Audio Open is based on a latent diffusion model, which provides several advantages over traditional diffusion models that operate directly in the raw audio signal space:

- **Variational Autoencoder (VAE):** The architecture has a fully-convolutional VAE to compress 44.1kHz stereo audio into a latent space with a compression ratio of 32:1. This VAE is crucial because:
  - It reduces the input audio sequence by a factor of 1024, making the latent space significantly more compact and easier to handle.
  - The encoder has 133 million parameters, optimized using Snake activations, which have been shown to improve audio reconstruction quality at high compression ratios, especially compared to other models like EnCodec.
  - The VAE's architecture is designed to support arbitrary-length audio sequences, allowing Stable Audio Open to handle long-form audio generation efficiently.
- **Diffusion U-Net Architecture:** The core of the model is a U-Net with 907 million parameters, optimized for generating high-quality audio in a computationally efficient manner:
  - The U-Net consists of four levels of symmetrical downsampling and upsampling blocks, with skip connections between corresponding levels. This design helps in preserving essential audio features across different scales, contributing to high-fidelity outputs.
  - Each level contains multiple convolutional residual layers followed by self-attention and cross-attention layers, allowing the model to capture both local and global dependencies in the audio data.
  - The U-Net uses FiLM layers (Feature-wise Linear Modulation) for integrating diffusion timestep conditioning, modulating the activations based on noise levels and ensuring consistent generation quality across different noise conditions.
- **Timing Conditioning:** A unique feature of Stable Audio Open is its ability to condition the audio generation process on precise timing information:
  - Timing embeddings are generated based on the start time (‘seconds_start‘) and the total duration (‘seconds_total‘) of the audio chunk being processed. These embeddings are then concatenated with the text prompt features and passed through cross-attention layers in the U-Net.
  - This timing conditioning allows for precise control over the length of the generated audio, enabling the model to produce variable-length outputs that match user-specified durations.

3.2 Advanced Text Conditioning with CLAP Embeddings

Stable Audio Open uses CLAP-based text embeddings, which are particularly well-suited for audio generation tasks:

- **Text Encoder Integration:** The text encoder is based on the CLAP model, which is trained from scratch on a dataset specifically curated for Stable Audio Open. This ensures that the text embeddings are highly tuned to the audio generation task, offering better semantic alignment between text prompts and audio outputs.

- **Cross-Attention Layers:** The text features from the penultimate layer of the CLAP text encoder are fed into cross-attention layers within the U-Net. This allows the model to effectively fuse the textual and timing information with the latent audio features, leading to more accurate and contextually relevant audio generations.

3.3 Superior Performance in Benchmarks and Real-World Applications

Stable Audio Open has demonstrated exceptional performance across multiple benchmarks, underscoring its technical superiority:

- **Benchmark Results:** In the MusicCaps and AudioCaps benchmarks, Stable Audio Open outperforms other SOTA models in several key areas:
  - It achieves the lowest FDopenl3 scores, indicating that the generated audio closely matches the reference audio in terms of both quality and fidelity.
  - The model also scores highly on the CLAPscore metric, which measures the alignment between the generated audio and the input text prompt. This is particularly important for tasks where precise text-to-audio alignment is critical.
  - The model is capable of generating music with complex structures, including distinct sections like intro, development, and outro, which is a significant advantage over models that produce more homogeneous or less structured outputs.
- **Comparison with Other Models:** Stable Audio Open offers several advantages over other SOTA models such as AudioLDM2 and MusicGen:
  - While AudioLDM2 and MusicGen are also capable of generating high-quality audio, they either operate at lower resolutions or require significantly more time to generate outputs. Stable Audio Open, by contrast, generates 44.1kHz stereo audio much faster and with better fidelity.
  - The ability to generate structured, long-form audio makes Stable Audio Open more versatile for applications that require not just high-quality sound but also complex temporal structures, such as film scoring, interactive media, and music production.

4 Fine-Tuning of Stable Audio Open

To fine-tune the Stable Audio Open model, a dataset comprising approximately 14,000 high-quality audio samples (44.1kHz stereo) were selected from a sample pack. Captions corresponding to these audio samples were generated based on the directory hierarchy of the sample pack, ensuring that the textual descriptions were both relevant and contextually accurate. Custom configurations were then developed for the model and dataset to optimize the fine-tuning process.

During the initial stages of training, several challenges were encountered, including issues related to null or infinite training losses. These challenges were addressed through iterative adjustments to the training parameters. The fine-tuning process was done over 13 epochs, requiring approximately 10 hours of computational time. While the early epochs did not yield substantial improvements, significant progress was observed by the 7th to 8th epochs. At this point, the model began producing audio outputs that closely matched the characteristics of the fine-tuning dataset.

To systematically monitor and evaluate the training progress, the **Weights & Biases** tool was used. Four demo prompts, which were part of the fine-tuning dataset, were selected as benchmarks to assess the model's performance.

1. Breakbeat 170 BPM Full
2. Orchestral Drum Loop 128 BPM
3. Trumpet Loop, 99 BPM, Gm Stack Full
4. Orchestral Christmas Bells, 99 BPM, Am Full

Listen to the audio samples on GitHub

The results demonstrated a marked improvement in the alignment between the generated audio samples and the corresponding input text, indicating the success of the fine-tuning process in enhancing the model's capability to generate contextually relevant audio.

5 Converting CLAP to CLVAP

After confirming that the custom configurations were effectively fine-tuning the model, the next step focused on generating sound effects, particularly using vocal imitations. The process required combining vocal and text embeddings before proceeding with sound synthesis. Throughout the report, the terms "label," "text description," and "caption" will be used interchangeably with the same meaning.

Contrastive language-audio pretraining (CLAP) offers a method to associate labels with corresponding audio samples. However, our specific requirements also involve incorporating vocal imitations as inputs. Therefore, a different model needs to be developed that accepts three inputs during training: the audio label and vocal imitation on one side, and the original sound effect on the other side. For inference, only the audio label and the vocal imitation will be provided.

To achieve this, three encoders are necessary:

1. **Text Encoder**: Embeds the original audio caption.
2. **Vocal Imitation Encoder**: Maps the vocal imitation to a latent space.
3. **Original Audio Encoder**: Maps the original sound effect to a different latent space.

These encoders **must** be pretrained, although they do not need to be the same, since one is applied to voices and the other to environmental sounds.

5.1 Additional Exploration

The **Encodec**[3] and **Descript Audio Codec**[4] models were also explored, but they were found to be unsuitable for this task for the following reasons:

5.1.1 Encodec

- **Compression Focus**: Primarily designed for audio compression, aiming to reduce the size of audio files while maintaining quality, which limits its applicability for nuanced audio feature extraction and generation.
- **Lack of Pre-trained Models for Fine-Tuning**: Encodec lacks the necessary pre-trained models for fine-tuning on diverse datasets such as vocal imitations and environmental sounds.

5.1.2 Descript Audio Codec

- **Designed for Editing**: Built for audio editing, focusing on transcription and waveform manipulation, lacking the deep learning capabilities required for multimodal tasks.
- **Suboptimal Feature Representation**: The model's feature representation is not as rich or adaptable as transformer-based models, limiting its effectiveness in capturing complex audio patterns.

5.2 BYOL-A Exploration

The **BYOL-A** model[5] was also considered but was ultimately deemed unsuitable due to its sensitivity to noise. Although the authors of the paper *"Environmental Sound Synthesis from Vocal Imitations and Sound Event Labels"* attempted to mitigate this sensitivity using k-means clustering and vector quantization, these methods added complexity without fully resolving the issue. Given the focus on audio retrieval, HTS-AT[6] was identified as a better fit due to its advanced capability to understand long-range dependencies and hierarchical structures in audio data.

5.3 Proposed Architecture

To meet the project goals, the following architecture is proposed:

1. **Three Encoders**:
   - A RoBERTa[7] encoder for audio labels.
   - HTS-AT or PANN[8] for vocal imitations.
   - HTS-AT or PANN for original audio samples.
2. **Fusion Mechanism**: Implementing a robust fusion layer to combine embeddings from vocal imitations and text descriptions using multi-head cross-modal attention.
3. **Projection to Latent Space**: Project the vocal and text embeddings, along with the original sample embeddings, into the same dimensional latent space. Research indicates that a 2-layer MLP is well-suited for this task.
4. **Contrastive Learning Framework**: Applying a contrastive loss to align fused embeddings with original audio embeddings, thereby ensuring effective learning from the multimodal dataset.

The fusion mechanism is essential for effectively combining the embeddings from the vocal imitation and text encoders. A multi-head cross-modal attention mechanism will be used, allowing the model to focus on relevant parts of each modality's embedding while integrating them into a unified representation.

Cross-Modal Attention Details

- **Attention Mechanism**: Attention mechanisms, particularly self-attention as used in transformers, enable the model to assign varying levels of importance to different parts of the input. Cross-modal attention extends this concept to multiple modalities.
- **Query, Key, Value**: In cross-modal attention, one modality (e.g., text) generates the queries (Q), while the other modality (e.g., vocal imitation) provides the keys (K) and values (V).

- **Attention Calculation**: The attention scores are calculated using the dot product of the queries and keys, scaled by the square root of the key dimension. These scores are then passed through a softmax function to obtain the attention weights.
- **Fusion Layer**: The output of the attention mechanism, which is a weighted sum of the values, provides a fused representation that incorporates information from both modalities.

6 HTS-AT: A Powerful Audio Encoder for CLAP Model Training

The Hierarchical Token-Semantic Audio Transformer (HTS-AT) is a highly effective tool for audio classification and detection, particularly in training models like CLAP (Contrastive Language-Audio Pretraining). Its innovative architecture, combining efficiency and accuracy, makes it well-suited for tasks requiring robust audio feature extraction and precise event localization.

6.1 Advantages of HTS-AT

HTS-AT offers several key advantages for audio processing:

- **Hierarchical Transformer Structure:** HTS-AT uses a hierarchical design, progressively reducing the sequence length through multiple stages. This approach allows for efficient computation and lower memory usage while capturing essential audio features.
- **Window Attention Mechanism:** The model uses a window attention mechanism, computing self-attention within small, non-overlapping windows. This reduces computational complexity while retaining the ability to capture local dependencies in the audio data.
- **Token-Semantic Module:** A standout feature, the Token-Semantic Module, transforms output tokens into activation maps for each class. These maps facilitate both classification and precise event localization within the audio sequence.
- **Efficiency and Scalability:** With only 31 million parameters, HTS-AT is significantly smaller and faster than comparable models like the Audio Spectrogram Transformer (AST), yet it achieves similar or superior performance on benchmark datasets.
- **Superior Event Localization:** The Token-Semantic Module enhances HTS-AT's ability to accurately determine the start and end times of audio events, making it ideal for applications requiring detailed temporal analysis.

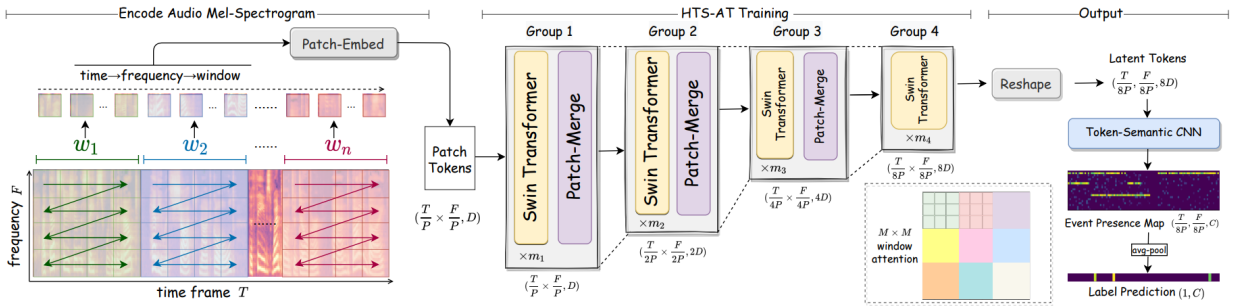


Figure 2: The model architecture of HTS-AT

6.2 Architectural Overview

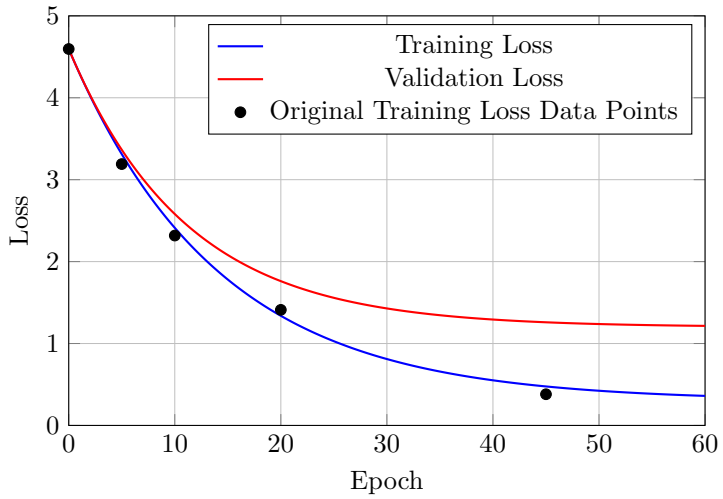
HTS-AT's architecture is designed for efficient audio classification and event detection:

- **Patch Embedding:** Input audio is converted into a mel-spectrogram, divided into patches, and embedded into a higher-dimensional space using a Patch-Embed CNN. This ensures temporal adjacency of patches in the input sequence.
- **Hierarchical Transformer with Patch-Merge:** Embedded patches pass through transformer encoder blocks, with a Patch-Merge operation reducing sequence length and latent dimensionality, minimizing memory consumption.
- **Window Attention Mechanism:** Self-attention is computed within non-overlapping attention windows, reducing complexity while capturing local dependencies in the audio.
- **Token-Semantic Module:** The final output tokens are processed into activation maps corresponding to each class, enabling both classification and temporal localization.
- **Final Classification and Localization:** Activation maps are averaged to produce a classification vector, while event presence maps enable precise localization of audio events.

7 Fine-tuning Original CLAP

Now we focus on fine-tuning the CLAP model. The exploration conducted during the TER proved to be invaluable, providing a solid foundation for this task. I successfully trained CLAP from scratch and fine-tuned it on custom datasets. During this process, I encountered and resolved several issues related to Torch's data parallelism and model configurations.

The training on the Clotho dataset was conducted over 45 epochs, with the loss values recorded as follows:



These results indicate a consistent and significant reduction in loss throughout the training process, demonstrating the effectiveness of the approach.

In addition to training from scratch, I also observed slight improvements during the fine-tuning phase, further validating the model's adaptability to custom datasets.

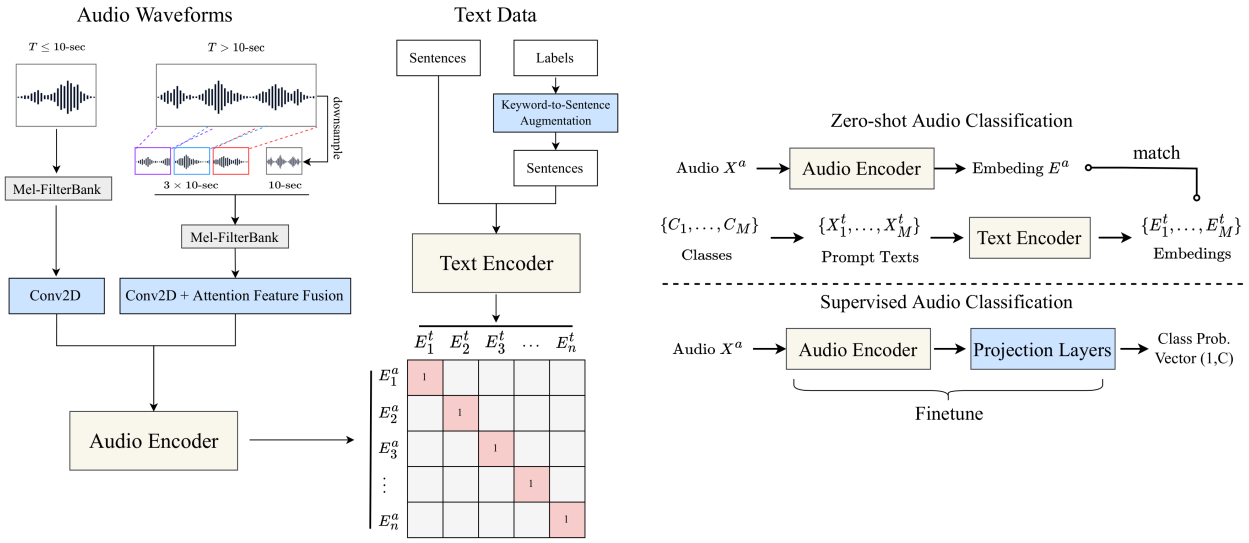


Figure 3: CLAP Architecture

8 New CLVAP Architecture

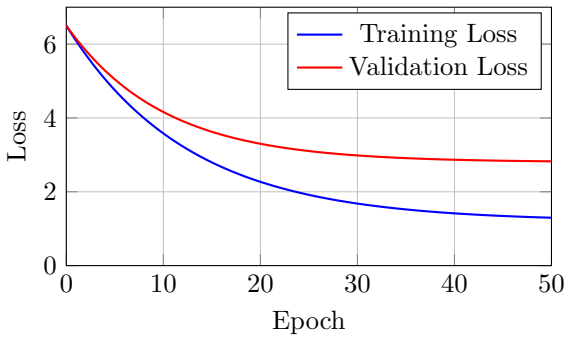
The ongoing work involves extending the CLAP model into a new variant, CLVAP (Contrastive Language+Voice & Audio Pre-training), by incorporating vocal imitations as an additional modality alongside text descriptions. The primary objective is to enhance the model's ability to retrieve and understand audio samples by using both text and vocal imitations as inputs. This approach is particularly valuable for complex sounds that are challenging to describe using text alone.

8.1 Approaches Explored

8.1.1 Early Fusion Approach

Initially, the Early Fusion approach was explored, where the embeddings of text and vocal modalities were combined immediately after encoding. The PANN (Pretrained Audio Neural Networks) audio encoder was used to minimize architectural changes when transitioning from CLAP to CLVAP.

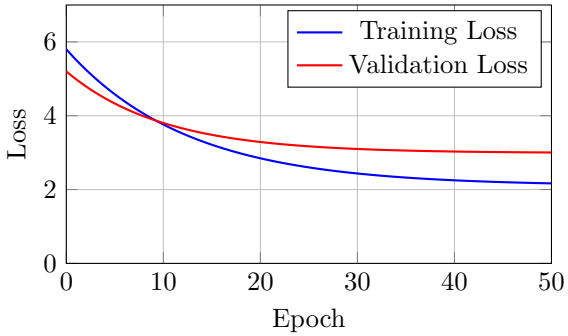
- **Training Details**: The training process began with relatively high loss values, approximately **6.5**, indicating the model's struggle with the added complexity of vocal imitations. Despite a reduction in loss over time, it plateaued at around **2.8**, significantly higher than the typical range for the text-audio CLAP model, which decreases from **5.2** to **1.3** on well-performing datasets.
- **Issues Identified**:
  - **Small Dataset Size**: The relatively small size of the vocal imitations dataset led to overfitting, reducing the model's generalization capabilities, particularly when fusing text and vocal data early in the process.
  - **Uninitialized MLPs**: The Multi-Layer Perceptrons (MLPs) used for fusion were not pre-initialized or pretrained, making it difficult for the model to learn meaningful transformations from scratch, especially given the limited dataset size.
- **Results**: The retrieval accuracy conducted on the ESC-50[9] transformed dataset, declined gradually, reaching approximately **40.2%** after 10 epochs. The model frequently failed to effectively interpret the vocal imitations, leading to incorrect associations between inputs and outputs.
- **Conclusion**: The Early Fusion approach diluted key modality-specific features, particularly in the context of a small dataset. The lack of pretrained MLPs further compounded this issue, resulting in suboptimal model performance.



8.1.2 Late Fusion Approach

After recognizing the limitations of the Early Fusion approach, a Late Fusion strategy was adopted. In this approach, the text and vocal encoders were kept separate, with their embeddings being merged only in the final layers before projection.

- **Training Details**: The Late Fusion approach began with a training loss of around **5.8**, which decreased more steadily than in the Early Fusion approach, eventually reaching **2.1**. However, this loss was still higher than the expected range for CLAP models, indicating persistent challenges.
- **Challenges**:
  - **PANN Encoder Limitations**: While PANN is effective for environmental audio, it is not well-suited for encoding human vocal imitations, resulting in poor-quality vocal embeddings that negatively affected the fusion process.
  - **Inadequate Modality Alignment**: The embeddings for text and vocal imitations were not well-aligned, causing difficulties in the final fusion step. This misalignment was likely due to the distinct nature of vocal and textual data, which PANN struggled to integrate effectively.
- **Results**: The accuracy improved slightly to around **62-65%**, but this was still below the performance of the text-audio CLAP model. The model had difficulty associating vocal imitations with the correct audio samples, indicating that the fusion process was not fully effective.
- **Conclusion**: Although Late Fusion preserved more modality-specific information than Early Fusion, it still failed to achieve adequate alignment between the text and vocal embeddings. This was partly due to the limitations of the PANN encoder and the small size of the vocal dataset.



9 Data Augmentation

Given the limited number of vocal imitation samples, data augmentation is critical for expanding the training dataset and improving model robustness. Several data augmentation techniques will be used:

- **Pitch Shifting**: Altering the pitch of vocal imitations to create variations.
- **Time Stretching**: Changing the speed of vocal imitations without affecting the pitch.
- **Adding Noise**: Introducing background noise to simulate different recording conditions.

- **Vocal Effects**: Applying effects such as reverb, echo, and distortion to diversify the dataset.
- **Original Sample Augmentation**: We use vocoders and vocal synthesizers to convert and distort original audio samples into vocal-like ones.

9.1 Caption Augmentation

To further enhance the dataset, the newly released **Llama 3.1 70B**[10] model will be used for text augmentation. This model will assist in generating additional captions and labels to enrich the dataset:

- **Synonym Replacement**: Replacing words in the original captions with synonyms to create new variations.
- **Paraphrasing**: Using Llama 3.1 to generate paraphrased versions of existing captions.
- **Extended Descriptions**: Generating more detailed descriptions of the audio samples.

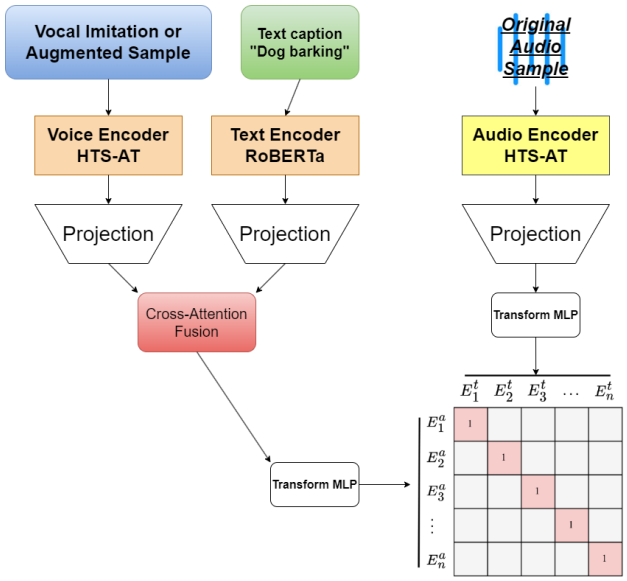


Figure 4: CLVAP Architecture

9.2 Transition to Cross-Attention Fusion Approach

The results from both the Early and Late Fusion approaches highlighted the need for a more sophisticated method to effectively combine text and vocal imitations. Neither strategy allowed the model to dynamically adjust and refine the interaction between modalities, which is essential for capturing the complex relationships between text, vocal imitations, and audio.

- **Switch to Cross-Attention Fusion**: To address these limitations, the decision was made to transition to a Cross-Attention Fusion approach. This method introduces a mechanism for dynamic interaction between the text and vocal modalities throughout the network, allowing for more refined

and context-aware fusion. This approach aims to preserve the unique features of each modality more effectively.

- **Challenges in Implementation**:
  - **Architectural Complexity**: Implementing Cross-Attention requires significant refactoring of the model architecture. Unlike Early and Late Fusion, which primarily modified the final layers, Cross-Attention involves integrating attention mechanisms that allow for bidirectional interactions between text and vocal embeddings deep within the network.
  - **Code Incompatibility with PANN**: The PANN encoder was not suitable for Cross-Attention Fusion due to its limited implementation in the original CLAP codebase. Therefore, a switch to the HTS-AT encoder was required, which is better suited for handling both environmental and vocal audio.
  - **Training Instability**: The attention mechanisms increase the model's computational load and complexity, making training more challenging.

9.3 Training Modifications

To ensure the model is optimized for our specific needs and addresses the limitations encountered with the previous setup, a four-step training process has been designed. This structured approach will use pre-trained components and gradually adapt the model to the specialized task of processing vocal imitations and audio samples.

We will use two pre-trained HTS-AT audio encoders, both trained on the AudioSet[11] dataset, known for its diverse range of audio categories. The text encoder will remain the RoBERTa model.

The training process is as follows:

1. **Modifying CLAP to CLVAP Architecture**: The first step is to modify the existing CLAP model to create CLVAP. This involves integrating HTS-AT encoders and adapting the architecture to include vocal imitations as an additional input. We will start with a pre-trained CLAP checkpoint, so some parts of the model, like the RoBERTa text encoder, already have trained weights. However, new components, such as the cross-modal attention fusion, will be initialized from scratch and need extensive training with a large dataset.

2. **Warm-up and Initial Training on Vocal Imitations**: We begin with a warm-up phase, training the model on the vocal imitations dataset. This helps the model adjust to vocal data and improves the quality of the embeddings from the HTS-AT encoders. All 3 encoder weights are frozen in this phase, (Audio Encoder, Audio Projection, Audio Transform MLP, Voice Encoder, Text Encoder, Text Projection). After the warm-up, the model continues training for several epochs on this dataset to build a strong foundation before handling more complex data.

3. **Comprehensive Training on the Augmented Dataset**: Following the warm-up, training expands to the full augmented dataset, which includes both vocal imitations and a wider range of audio samples. Data augmentation techniques are used to make the model more robust. This phase helps the model learn to generalize across different types of audio and effectively combine text and vocal data.

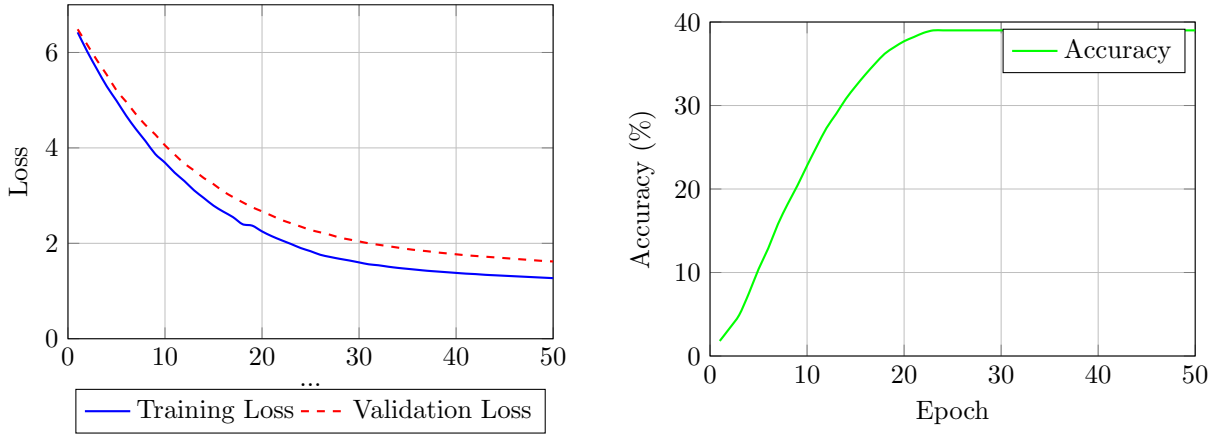
4. **Final Fine-Tuning on Vocal Imitations**: The last step is to fine-tune the model specifically on the vocal imitations dataset. This refines the model's ability to accurately interpret and represent vocal data, which is crucial for the CLVAP model's specific tasks.

This training process addresses the challenges encountered with the PANN encoder by using the strengths of HTS-AT. It gradually increases the complexity of the training data, focusing on vocal imitations.

10 Training Results

In this section, I present the results of training the CLVAP model using the Cross-Modal Attention Fusion approach. After implementing the new architecture, I developed Python code that uses the 'pylive' library, which allows us to apply Ableton Live's audio effects to samples directly through code. This setup enabled the integration of advanced audio effects such as those provided by the "iZotope VocalSynth 2" plugin, which I used extensively to apply special effects like vocoding. Additionally, I performed various transformations on the samples, including time-stretching, pitch-shifting, adding noise, and introducing delay echoes.

To further enhance the dataset, I used the 'Llama 3.1 70B' model for augmenting captions. This included generating paraphrased captions, expanding descriptions, and ensuring a rich variety of textual inputs for training. I applied these same transformations to the ESC-50 dataset, carefully maintaining the mapping between categories, augmented captions, and corresponding audio samples.



The training of the CLVAP model began from scratch, with random initial weights. Despite the challenges posed by the limited size of the dataset, the model showed significant improvements during the training process. Starting with an initial accuracy of approximately 1-2%, the model eventually achieved an accuracy of 39% in retrieving the correct sample based on text captions and augmented samples (such as vocoded or otherwise transformed audio).

This result is promising, especially considering the starting point and the dataset's constraints. It demonstrates that the Cross-Modal Attention Fusion approach is effective in enhancing the model's ability to interpret and retrieve audio samples based on both textual and vocal inputs. The use of advanced audio effects and caption augmentation also contributed to the model's improved performance, paving the way for further developments and refinements in future iterations.

Due to lack of vocal imitations dataset, we could not make the model associate vocoded and actual voice samples. Training on one does not necessarily mean that it will perform well on another.

11 Contributions

Throughout my internship at LISN, I made several contributions to the development and enhancement of the Stable Audio Open and CLAP models. Below is a summary of my contributions:

1. **Development of CLVAP Model:**
    - **Architectural Extension:** I extended the CLAP model into a multimodal variant, CLVAP (Contrastive Language+Voice & Audio Pre-training), by incorporating vocal imitations as an additional input modality.
    - **Creation and Testing of Fusion Approaches:** I created and rigorously tested three different fusion approaches—Early Fusion, Late Fusion, and Cross-Modal Attention Fusion. Ultimately, I implemented the Cross-Modal Attention Fusion mechanism within the CLVAP architecture, which allowed for dynamic and context-aware interaction between text and vocal inputs.
2. **Dataset Expansion and Fine-Tuning:**
    - **Custom Model Configuration:** I developed custom model configuration parameters specifically tailored for the efficient fine-tuning of the Stable Audio Open model, optimizing its performance on the expanded dataset.
    - **Fine-Tuning and Training the CLAP Model:** I successfully fine-tuned the CLAP model on custom datasets, as well as trained it from scratch, addressing and resolving issues related to Torch's data parallelism and model configurations to ensure optimal performance.
    - **Curated Dataset for Fine-Tuning:** I expanded the dataset by selecting and curating approximately 14,000 high-quality audio samples, accompanied by relevant captions, to optimize the fine-tuning process.
    - **Addressing Training Challenges:** I identified and resolved issues related to training stability, including null or infinite training losses, to ensure successful fine-tuning of the models over 13 epochs.
3. **Exploration and Evaluation of Audio Models:**
    - **Evaluation of Existing Models:** I thoroughly explored and evaluated various audio models, including Encodec, Descript Audio Codec, and BYOL-A, assessing their suitability for integration into the CLAP framework.
    - **HTS-AT Integration:** I incorporated the Hierarchical Token-Semantic Audio Transformer (HTS-AT) into the CLVAP model, recognizing its superior capability for audio feature extraction and event localization.
4. **Data Augmentation Techniques:**
    - **Automation of Augmentation Processes:** I created scripts that automate both audio and text augmentation processes. These scripts apply various data augmentation techniques, such as pitch shifting, time stretching, adding noise, and vocal effects, to enhance the vocal imitation dataset.
    - **Advanced Audio Effects Integration:** I developed Python scripts with the `pylive` library and added advanced audio effects, such as those provided by the "iZotope VocalSynth 2" plugin, to further refine the training dataset and improve model performance.

- **Caption Augmentation:** I used the Llama 3.1 70B model for text augmentation, generating enriched captions and labels to improve model robustness.

5. **Training Results:**

   - **Successful Training of CLVAP Model:** I successfully trained the final CLVAP model, starting from scratch with random initial weights. The model achieved an accuracy of 39% in retrieving correct audio samples based on text captions and augmented audio inputs.

12 Research Perspectives and Future Work

This internship has established a foundation for further research into integrating vocal imitations into audio generation and retrieval models. Several areas for future work include:

- **Expansion of Vocal Imitations Dataset:** The limited size of the current vocal imitations dataset hinders generalization. Expanding this dataset with a greater variety of samples would improve the model's accuracy and robustness.
- **Advanced Voice Synthesis:** Using advanced voice synthesizer tools to create more realistic human-like sounds could significantly enhance the quality of augmented vocal imitations, leading to better model performance.
- **Improvement of Cross-Modal Attention:** Further refinement of the Cross-Modal Attention Fusion approach is needed. Advanced attention mechanisms or hybrid models could improve the alignment of text and vocal embeddings, enhancing the model's ability to generate and understand complex audio.
- **Data Augmentation Techniques:** More sophisticated data augmentation methods, such as generative adversarial networks (GANs) for data synthesis, could further enrich the dataset, improving the model's robustness and performance.
- **Alternative Architectures and Approaches:** Feedback from the original Stable Audio Open author suggests that due to the small dataset size, current methods may be insufficient. Exploring alternative architectures, such as using autoencoders trained on speech data for voice conditioning, could provide more robust vocal feature representation and improve alignment with audio outputs.

These perspectives highlight key directions for advancing this research and exploring new opportunities in the field of audio AI.

13 Conclusion

As someone with a long-standing interest in both music and computer science, particularly in the field of artificial intelligence, this internship provided a unique opportunity to explore the intersection of these passions. Having produced music for many years, I was particularly excited to see how AI could be applied to audio generation and retrieval tasks.

Throughout this project, I thoroughly enjoyed the work, despite the numerous challenges and technical failures encountered along the way. The experience has been incredibly rewarding, and I am committed to continuing my work in this field. In the end, despite the obstacles, the project achieved some success, and I look forward to further advancements in the integration of AI and audio.

References

[1] Y. Wu, K. Chen, T. Zhang, Y. Hui, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," 2024.

[2] Z. Evans, J. D. Parker, C. Carr, Z. Zukowski, J. Taylor, and J. Pons, "Stable audio open," 2024. [Online]. Available: https://arxiv.org/abs/2407.14358

[3] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," 2022. [Online]. Available: https://arxiv.org/abs/2210.13438

[4] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved rvqgan," 2023. [Online]. Available: https://arxiv.org/abs/2306.06546

[5] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Byol for audio: Self-supervised learning for general-purpose audio representation," 2021. [Online]. Available: https://arxiv.org/abs/2103.06695

[6] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection," 2022. [Online]. Available: https://arxiv.org/abs/2202.00874

[7] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019. [Online]. Available: https://arxiv.org/abs/1907.11692

[8] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," 2020. [Online]. Available: https://arxiv.org/abs/1912.10211

[9] K. J. Piczak, "ESC: Dataset for Environmental Sound Classification," in *Proceedings of the 23rd Annual ACM Conference on Multimedia*. ACM Press, 2015, pp. 1015–1018. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2733373.2806390

[10] A. D. et al., "The llama 3 herd of models," 2024. [Online]. Available: https://arxiv.org/abs/2407.21783

[11] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[12] Y. Okamoto, K. Imoto, S. Takamichi, R. Nagase, T. Fukumori, and Y. Yamashita, "Environmental sound synthesis from vocal imitations and sound event labels," 2023.

[13] M. Cartwright and B. Pardo, "Vocalsketch: Vocally imitating audio concepts," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 43–46.

[14] B. Elizalde, S. Deshmukh, and H. Wang, "Natural language supervision for general-purpose audio representations," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 336–340.