AN ABSTRACT OF THE THESIS OF

Peter J. McCabe for the degree of Master of Science in Artificial Intelligence presented on March 13, 2026.

Title: Audio-Free Timbre Classification With Text-Only Training

Abstract approved: _____

Patrick Donnelly

Text-only Training is a promising alternative machine learning paradigm for training multimodal models that does not require data from every modality. However, despite the potential of Text-only Training, its use as an approximation of, and replacement for, missing data for supervised learning in data-scarce environments remains under-explored. In this work I define the kinds of problems well suited to Text-only Training and examine techniques to label, generate, and design text-based training data. I address the Modality Gap's effect on the performance of Text-only Training by extending and contrasting theoretically-justified methods with those commonly used in practice. I provide a case study on classifying subjective audio timbre descriptions based on three kinds of text-only training data, four models, and fourteen augmentation methods on eight audio-timbre datasets. Finally, my work shows that the Text-only Training paradigm successfully allows for the training of audio classifiers without real audio data, and lays the founda-

tion for assessing Text-only Training in data-scarce supervised learning problems in general.

©Copyright by Peter J. McCabe
March 13, 2026
All Rights Reserved

Audio-Free Timbre Classification With Text-Only Training

by

Peter J. McCabe

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented March 13, 2026
Commencement June 2026

Master of Science thesis of Peter J. McCabe presented on March 13, 2026.

APPROVED:

Major Professor, representing Artificial Intelligence

Head of the School of Electrical Engineering and Computer Science

---

Dean of Graduate Education

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

Peter J. McCabe, Author

ACKNOWLEDGEMENTS

I would like to acknowledge the massive amount of support and encouragement I have received from my friends, family, and loved ones. Without you, not only could I have never completed this, I would have never thought to try. I love you all. Additionally, I affirm that this research received no specific funding from any particular source in public, not-for-profit, or for-profit sectors.

TABLE OF CONTENTS

Page

1 Introduction 1
1.1 Introduction 1
1.2 Text-Only Training 5

2 Background 9
2.1 Timbre 9
2.2 Audio-Language Modeling 10
2.2.1 Contrastive Multimodal Models 11
2.3 The Modality Gap 13
2.3.1 Definition 13

3 Related Work 18
3.1 Text-only Training 18
3.2 Closing the Modality Gap 20
3.3 Timbre Classification 25

4 Datasets 27
4.1 Timbre Classification Datasets 27
4.1.1 Audio Commons 28
4.2 Audio Captioning 30
4.2.1 Clotho 30
4.2.2 SoundBible and WavCaps 32
4.3 Synthetic Text-Timbre Data 33
4.3.1 Pseudo-Labeling Real Captions 34
4.3.2 Pseudo-labeled LLM-Generated Captions 36
4.3.3 Label-Centered Gaussian Noise 40

5 Methodology 42
5.1 Bridging the Modality Gap 42
5.1.1 Method Details 44
5.2 Metrics 47
5.2.1 Modality Gap Metrics 48

TABLE OF CONTENTS (Continued)

Page

5.2.2 Representation Alignment Metrics 51
5.3 Experiments 56
5.3.1 Models 56
5.3.2 Experimental Design 59

6 Modality Gap Augmentations 61
6.1 Representational Alignment 61
6.2 Modality Gap Metrics 72

7 Timbre Classification Results 86
7.1 Augmentations 86
7.2 Training Datasets 90
7.3 Model Performance 95
7.4 Ablations 98
7.5 Baseline Comparisons 101

8 Conclusion 105
8.1 Limitations and Future Work 105
8.2 Conclusion 108

Bibliography 110

LIST OF FIGURES

Figure | Page

1.1 An overview of the classification task to be performed in the case study. Rather than classify the timbral descriptors of audio data, Text-only Training allows one to learn a classifier using language data instead. 5

1.2 A simplified audio classification pipeline using Text-only Training. During training the missing data is described with words, and encoded into a multimodal vector space. The resulting vectors approximate the missing audio data and are used to train a classifier. At inference, the real audio data is encoded the same way, and is classified with the trained model. 6

2.1 An general overview of CLIP's structure and use. Left: Batches of images and text embeddings are compared and CLIP predicts the matches. Right: At inference, CLIP can be used for zero-shot classification by providing a general description of the image class. Image credit to [78]. 11

2.2 An overview of CLAP's architecture. CLIP-style training (Top) and zero-shot classification (Bottom). Image credit to [111]. 14

2.3 An example of the modality gap effect in CLAP. 400 random samples are drawn from the Clotho+ dataset (see Section 4.2.1) half from audio, and half from text, and projected from 512 down to two dimensions using multidimensional scaling. The two modalities are mostly separate with scattered outliers even in only two dimensions. 15

4.1 Class label distributions of WavCaps [68] as determined by text-text similarity between captions and a label prompt. I compare CLAP [111], an audio-language model, and SBERT [82], a language semantic similarity model. 35

4.2 The process for generating text data using an LLM. First, a label is sampled from the desired class label distribution. Then, the LLM is prompted to describe an audio event that corresponds to that descriptor. The outputs and sampled label are then used as training data. 37

LIST OF FIGURES (Continued)

Figure — Page

5.1 A simplified overview of the case study. **Top:** the desired audio data is described by natural language, such as a label or Anchor Prompt. Next, the training data is generated by one of the three dataset creation methods discussed in Section 4.3 and is embedded in CLAP. **Bottom:** an augmentation method is selected to combat the affect of the modality gap. The resulting approximation of audio embeddings is used as the data for the classification experiments. . 43

5.2 The empirical measurements of the conditions required for C2 to remove the modality gap taken from CLIP embeddings of the MS-COCO dataset. Image Credit [119]. 57

5.3 My empirical measurements of the C2 conditions using CLAP embeddings from the Clotho+ dataset. I find very similar results to those of Zhang *et al.* [119] indicating the C2 conditions are met. . . 58

6.1 A comparison of the impact clean and noisy versions of each augmentation method have on the pair alignment of Clotho+. The standard deviations are negligible and not displayed. 62

6.2 Distribution estimations of percent change when choosing to augment the data over performing no augmentation. Supervised methods tend to create the most impactful changes, followed by Mean Shifting. The other representational metrics not shown have less significant ranges of $< 5\%$. 67

6.3 Consistency metrics split by text and audio datasets. Cross-Consistency shows slightly lower values on the Cloth+ Audio combinations. This is likely due to the bias of the augmentation methods towards the Clotho+ Audio distributions leading to more change. Self and Pseudo-Consistency are constant across the audio datasets as they only depend on text data. 69

6.4 NDCG metrics split by text and audio datasets. Dataset combination has a strong effect on Cross-NDCG. This is likely due to the bias of the augmentation methods towards the Clotho+ Audio distributions leading to more change. Again, Self and Pseudo-NDCG are constant across audio datasets. 70

LIST OF FIGURES (Continued)

Figure Page

6.5 Consistency metrics split by text and augmentation. A noticeable split between supervised and unsupervised methods can be seen, especially in the LLM and Label-Centered Noise datasets. These likely indicate distributional differences between the Clotho+ dataset on which the supervised models were trained. 73

6.6 NDCG metrics split by text and augmentation. The same split between supervised and unsupervised methods is present here. Moreover, one can see the effect of training the supervised methods with added noise in the Cross-NDCG (Right). The added noise reduces neighborhood change in the cross-modal setting when compared with supervised methods without added noise. 74

6.7 Mean and standard deviations of Linear Separability for each augmentation method. I average across all text and audio datasets, and over each of the ten noisy training runs. Almost all methods–except C2—fail to merge the two subspaces such that a linear classifier is unable to nearly perfectly separate them. 75

6.8 Mean and standard deviations of Mean Distance for each augmentation method. C2 is once again surpasses other options. Mean Shifting does next best, but suffers from a biased construction: it will always shift towards the Clotho+ Audio distribution, meaning if the test audio follows a different distribution, the gap persists. . . 77

6.9 Mean and standard deviations of Approximate Hausdorff for each augmentation method. The methods are mostly interchangeable, with the supervised methods lowering the distance slightly more than the unsupervised. 78

6.10 **Top:** Mean and standard deviation of the Sliced Wasserstein Distance for each augmentation.
Bottom: A boxplot of the SWD. This reveals the mean and median SWD values for each metric besides C2 are approximately equal, share a small spread of values, and are relatively comparable. Additionally, it shows that the median of the C2 method is the lowest of all the methods, and that the distribution is bimodal. 79

6.11 Mean Gap metric heatmaps organized by text and audio datasets. 80

LIST OF FIGURES (Continued)

Figure Page

6.12 A Pearson's r-correlation matrix heatmap for both distance and alignment metrics. Linear Separability and Mean Distance are both correlated, along with Self- and Pseudo-NDCG. There are mild correlations between the other representational metrics, and a negative correlation between Sliced Wasserstein Distance and Linear Separability. 82

6.13 Heatmap of Pearson's r correlation coefficient matrix between all metrics and downstream classification F_1-scores. The Pseudo-NDCG, Cross-NDCG, and Pseudo-Consistency are moderately correlated with training F_1-score, while none of the metrics are strongly correlated with test F_1-score. 83

7.1 Mean test scores using unsupervised augmentation methods. Applying on augmentation typically leads to lower scores, with the exception of the Hard dataset. The C2 method outperforms the Mean Shifting method in every case. 88

7.2 Mean test scores using supervised augmentation methods. The supervised methods are more varied in their scores than the unsupervised methods, but NonlinearN is the only method among them to never be worse on average than applying no augmentation technique. 89

7.3 Comparison of the baseline, best unsupervised and supervised methods: Nothing, C2, and NonlinearN. C2 returns the best overall F_1-score. 90

7.4 Distributions of test audio F_1-scores for each augmentation. C2 and the nonlinear methods have the best overall score curves. 91

7.5 Mean and standard deviation of test scores averaged across data augmentation methods. Label-Centered Noise and WavCaps_CLAP are the best on average. Some exceptions include the Hard and Sharp audio datasets, in which WavCaps_SBERT does best or better than its CLAP version. 92

LIST OF FIGURES (Continued)

Figure | Page

7.6 Mean and standard deviation of train and test scores across all augmentations and targets. The LLM Captions dataset has the greatest difference in train and test scores and the widest variance. Despite having some of the highest trainging scores, it has the lowest average test scores. 93

7.7 Mean and standard deviation of test scores across for each text dataset separated by augmentations. C2 remains both the highest-scoring method on average, but also has one of the most stable score distributions across datasets. 94

7.8 The average train and test scores for models and text training datasets. The text datasets are broadly consistent across model types. 97

7.9 The average train and test F_1-scores for each model type and audio dataset. The Booming and Warm datasets challenged all model types, but the Hard dataset was difficult for only cross entropy models. 97

7.10 The average test scores for each model and augmentation. Unsupervised methods tend to lead to high train scores for all model types, but only the MSE models score highly on test data. Supervised models tend to score lower train scores for all models, but the cross entropy models tend to score higher on test data with supervised models than with unsupervised. 99

7.11 The average and standard deviation of train F_1-scores for each large and unbalanced text datasets compared with the sampled datasets used in the main experiments. The full datasets score lower and have a wider variance, despite consisting of significantly more data. 100

7.12 The average and standard deviation of test F_1-scores for each large and unbalanced text datasets compared with the sampled datasets used in the main experiments. The large and unbalanced datasets fail to overcome their biased label distributions and score lower than the sampled datasets. This indicates the undersampling method used in the main experiments was justified. 100

LIST OF TABLES

Table — Page

2.1 L2 norms between Clotho (C) and SoundBible (Sb) datasets. Audio and text embeddings are noted by subscripts a and t, respectively. The distance between modalities within datasets is approximately equal, while those between datasets are not. The distances across datasets are also greater than those within, supporting the assumption that the text embeddings that match the audio (within dataset) are closer in the CLAP embedding space than those embeddings that do not (between datasets). 16

4.1 Sizes and class balance of each timbral descriptor audio dataset. †I use half of the reverb dataset ($n \approx 400$) for testing and training the audio baseline, and the other for development. 30

4.2 A set of captions from the Clotho dataset in which all captions corresponds to the same source audio [24]. 31

4.3 Examples of raw data and ChatpGPT outputs used in WavCaps. Taken from [68]. 33

4.4 The prompts used for semantic similarity labeling with CLAP and SBERT. Each prompt is embedded into the model's embedding space and the mean is calculated and re-normalized for use as the comparison pseudo-prompt. 34

4.5 Total sizes of the three text datasets used in the classification experiments. 41

6.1 The Pearson's r-correlation coefficients between Pair Alignment and distance metrics (Top) and representation metrics (Bottom). Most are either uncorrelated or negatively correlated, indicating pair alignment is harmed by higher distances between clusters and rigid augmentations that do not allow for the supervised pairs to become closer. 64

6.2 The average percent change in using noisy or clean augmentation methods. The Paired row represents the Clotho+ dataset's values, while the Unpaired row consists of all other combinations of text and audio datasets which do not contain paired data. In both cases, adding noise has almost no effect. 65

LIST OF TABLES (Continued)

Table		Page
6.3	The mean and standard deviation of percent increase in metric values across all datasets and experiments. Note that C2 is omitted from the L2-Distance columns as a percent increase from a value of 0 is undefined.	81
7.1	The average change in F_1-score when applying noisy augmentations instead of the non-noisy versions. All methods improve when noise is applied. Supervised methods improve more than unsupervised methods.	87
7.2	The average and standard deviation of F_1-scores per level and loss function, and the average change when trained with noisy augmentation. MLP1 refers to one-layer linear networks, while MLP2 refers to two-layer networks.	95
7.3	Test F_1 scores of the mean, median, and best models with data augmentation, CLAP zero-shot, and 10-fold cross validation on real audio. The best text-only results are in bold.	102

Chapter 1: Introduction

1.1 Introduction

In the modern AI era, automated tasks are increasingly being performed by machine-learning based systems. As these tasks and corresponding machine learning systems continue to get more and more sophisticated, the demand for data has grown exponentially. However, as machine learning problems grow more complex, so too do the data required to train the machine learning systems. For example, if one wants to automatically caption an audio recording, they need a system capable of handling both audio and language data. These different sources of data are often called "modalities", and when working with more than one, it generally becomes harder to perform machine learning tasks. That difficulty stems from the fact that the data take different forms depending on the modality. For example, an audio-video recording has both audio and video data, as well as various forms of metadata. The audio data takes the form of a series of numbers, the number and range of which depend on the sampling rate of the recording equipment. The video, on the other hand, takes the form of a series of images, themselves are a set of pixels, which are a collection of color and sometimes brightness values. The exact nature of the number of images or number of pixels is once again dependent on the recording equipment. Finally, the metadata consist of a small set of numbers

and words, such as the time and place of recording, the recording equipment make and model, and other details related to the data. For example, in ten seconds of footage recorded with a typical modern smartphone, one can reasonably expect 441,000 audio samples and 300 1080x1920 images. The vast majority of machine learning algorithms require all data samples to be the same size and shape, so the fundamental challenge becomes: how does one combine these various forms of data in a reasonable manner?

One method that has gained traction over the last few years is called representation learning. Representation learning seeks to find an alternative"representation" of the data that allows a practitioner to work with it more easily. Commonly, practitioners will sacrifice some of the information in the original format in exchange for a smaller and consistently-sized alternative. That way, rather than having a ten-second recording with 441,000 samples and an eight-second recording with 352,800 samples, one can learn a representation of the audio with a consistent size, for example, a vector of size 512 for each recording. Then, once all of the recordings are the same size, one can use machine learning for the desired downstream task.

One particular method for representation learning is called *contrastive learning*. Contrastive learning seeks to find a representation where for each datum in the training set, all data that is *dissimilar* to any given datum is also dissimilar in the learned representation, and all data that is *similar* before transformation remains similar afterwards. As an example, if two audio recordings were of the same event, such as two people recording the same dog barking, then they should still somehow be *similar* following the transformation to the new representation. Likewise, if two

recordings are of drastically different things, like a quiet library and a loud sporting event, then they should be *dissimilar* in the new representation. Precisely how one defines similarity and enforces that constraint on the contrastive learning process is an area of active research, but one of the most common ways is by having a set of "similar" examples, and defining everything else as "dissimilar". While this is a simplistic approach, it allows one to easily understand what makes something similar or dissimilar, which would otherwise be a difficult thing to define. For example, which pair is more similar, a recording of a single honeybee in flight and a recording of a wasp under similar circumstances, or a recording of a single honeybee and a recording of an active hive? Depending on one's definition of similarity, either pair could be more so than the other. Happily, this method allows one to easily work with data with different modalities, reffered to as multimodal data. By assigning data that correspond to the same event in both modalities to each others' similarity set, the contrastive learning model learns that the two examples "mean" the same thing, despite coming from different sources. This kind of technique has led to improvements in several different applications, including captioning and retrieval [78, 18]. One of the benefits of representation learning is that the representation can be re-used for new applications such as classifying sounds.

Because learned representations can be used for so many downstream applications, improving the quality of the representations is an efficient way to improve many systems and research areas. One particular issue that is present in contrastive learning-based representations is the so-called "Modality Gap". The modality gap

is an empirically observed gap between the vectors of each modality within the embedding spaces of visual-language, audio-language, and other multimodal models [53, 91]. Despite being trained to keep similar items close and dissimilar items far, most or all of the vectors sourced from language are grouped together entirely separately from the group of audio or image vectors. While not necessarily an inherent issue with contrastive learning, this gap does suggest inefficiencies within the contrastive learning process that is also an active research area. The gap does, however, pose an issue when trying to use the learned representations for certain downstream applications. In an ideal world where there is no gap, for any given data the learned representation from both modalities would be nearly or exactly identical. That could allow one to use the data from only one modality *in place of the other modality.*

Interest in this idea has been growing in the research community. Specifically, researchers have shown that using text as the only data modality during model training is effective for multimodal or cross-modal applications at inference time [46]. This methodology, called Text-only Training (ToT) in this work, is promising because it allows practitioners to avoid collecting expensive multimodal datasets. Acquiring text data is often far easier to obtain at scale and less expensive than other modalities like images, video, or audio. However, most research in this area has focused on problems like captioning [70, 17], information retrieval [40], and zero-shot classification [36]. These problems are well-suited to Text-only Training because they naturally allow one to use several similar samples of text data, such as multiple captions of the same image or metadata, as both labels and input in a

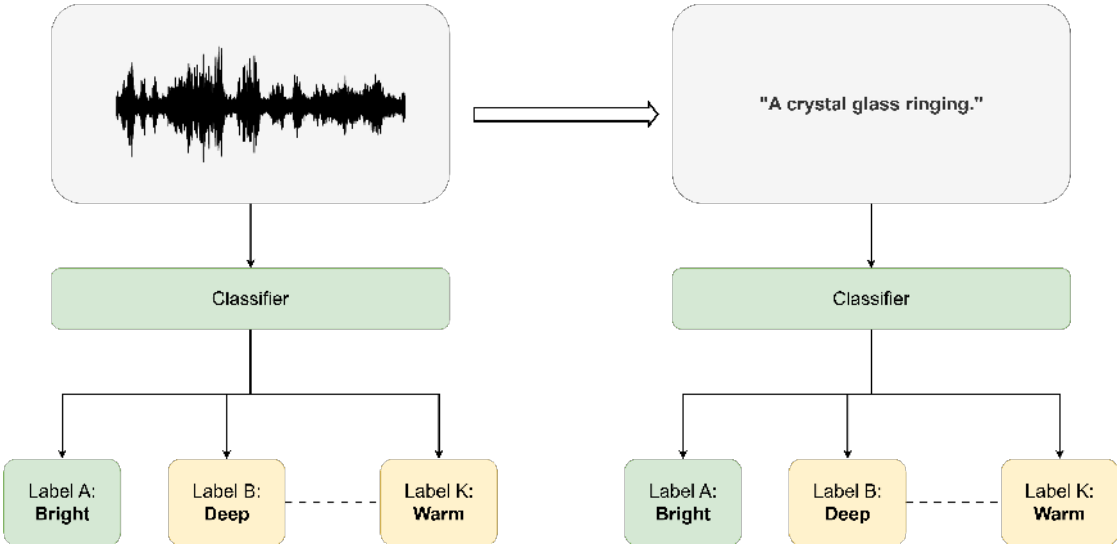


Figure 1.1: An overview of the classification task to be performed in the case study. Rather than classify the timbral descriptors of audio data, Text-only Training allows one to learn a classifier using language data instead.

self-supervised fashion [70, 17]. Therefore, few works have explored ToT for explicit use in downstream supervised learning. In other words, directly training machine learning models on cross-modal language data as a substitute for supervised data in the desired modality is an under-explored use for ToT. This provides the motivation for this work. To explore this, I present a case study on the task of non-musical audio timbre classification.

1.2 Text-Only Training

First, I define Text-only Training and the problems for which it is well-suited in more detail. An overview is provided in Figure 1.2. ToT refers to using only

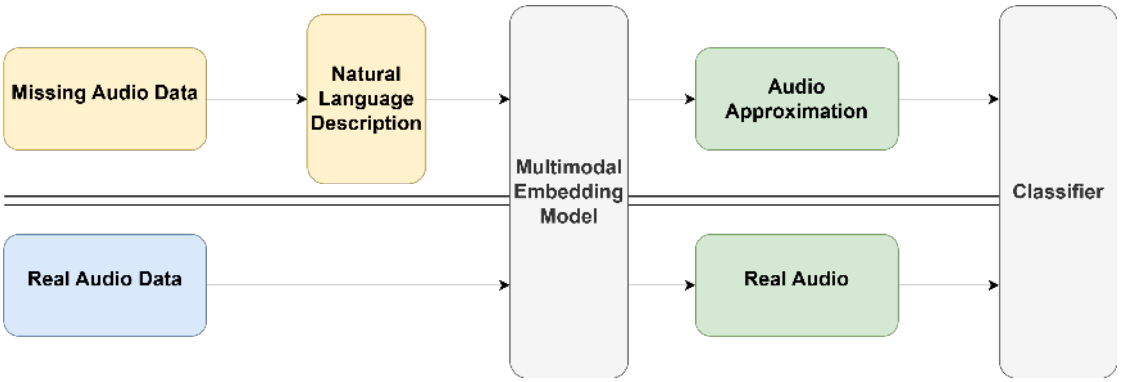


Figure 1.2: A simplified audio classification pipeline using Text-only Training. During training the missing data is described with words, and encoded into a multimodal vector space. The resulting vectors approximate the missing audio data and are used to train a classifier. At inference, the real audio data is encoded the same way, and is classified with the trained model.

text data to train machine learning models by leveraging a pretrained multimodal model able to compare data from different modalities. By finding the most similar text data to the desired data, one may approximate it in the desired modality. In other words, instead of training on real image data, one could use the text caption that is most similar to it according to the pretrained multimodal model. For these reasons, the supervised learning problems well-suited to ToT have three properties:

1. There is a lack of genuine labeled data in the desired modality.

2. A practitioner must have access to a pretrained multimodal model able to measure semantic similarity between text and the desired modality.

3. The labels or characteristics of the desired data should be well-described by language.

First, as evidenced by the modality gap, the ToT process is inexact and noisy. Therefore, one should prefer to use clean data from the desired modality and only use Text-only Training as a supplement to other methods or as a last resort. Second, ToT requires a method for determining the most semantically similar data across modalities. The current state of the art in multimodal semantic similarity modeling is usually a pretrained contrastive multimodal model trained with deep learning. Third, to best approximate the desired data, its key characteristics or properties—such as, label, structure, or value—must be fully describable with natural language. In other words, the information lost by representing non-language data with language is minimized.

In the case study I explore classifying descriptive labels of non-musical timbre. This task consists of predicting how a person might describe non-musical sounds with words, that is, what a sound "sounds like." For example, people often consider wind chimes to sound "bright". This problem meets the above criteria, so it is well-suited to ToT: there are no large audio and timbral descriptor datasets, there are many suitable multimodal contrastive audio-language models [111, 26, 37], and there is a plethora of ways to describe the timbre of a sound [11, 48]. In this work I purport that audio timbre classification represents a typical use-case for Text-only Training and will show the efficacy, issues, and decisions required to make the paradigm successful. In doing so, I will address the modality gap, and present ten metrics for measuring its representation, distribution, and extent. I provide more than a dozen methods for closing or combating it, including theory-based and commonly-used methods in the literature. To sum up, I present a case study

that examines the promising Text-only Training paradigm for machine learning as a method for overcoming a lack of training data via the task of timbre description classification.

Chapter 2: Background

I assume the reader has basic knowledge of machine learning and deep learning concepts including but not limited to supervised, unsupervised, and semi-supervised scenarios, loss functions, classification and regression, neural networks and common deep learning practice. However, more specific models, techniques, and concepts will be explained as needed.

2.1 Timbre

Along with pitch, rhythm, and volume, timbre, sometimes referred to as tone color, is a fundamental component of sound. Timbre allows humans to distinguish between multiple sounds even when all other aspects are identical. In other words, a sound's timbre is often described as what a sound "sounds" like, as opposed to the subject, source, or content of the sound. Timbre is usually associated with the individual characteristics of instruments [42], musical moods, themes, or contexts [32], and musical genre [101]. However, all sounds have their own timbres. For example, one could describe the timbre of keys jingling in someone's pocket as "metallic", "noisy", or "percussive" [89]. The timbre of non-musical sounds is an often overlooked area of study in the machine listening community; without enough data to support machine learning algorithms, there is little point in its

use. Typically, researchers have favored studying the timbre of musical excerpts [9], instruments [38], indirectly as parts of caption labels [20, 68], or simply using timbre as features for other machine learning tasks [6, 96].

2.2 Audio-Language Modeling

This work falls under the umbrella of Audio-Language Modeling: a field that seeks to find relationships between audio and language. At the most basic level, a simple Audio-Language Model (ALM) might describe sounds using words or sounds. For example, the familiar musical letter notation that assigns a letter from A to G to each pitch in a musical scale could be considered an ALM as it models the relationship of octaves to how humans perceive pitch. ALMs are becoming increasingly sophisticated in the modern age of machine learning, especially through the development of effective deep learning techniques. More and more, ALMs seek to find higher-level relationships between audio and language. These might include long-form descriptions of audio or music, similarities between musical and linguistic grammars, speech recognition, and so on. ALMs have broad applications including diverse subjects such as speech recognition [30], computational musicology [66], and medicine [95]. Commonly, ALMs are employed for tasks such as automated captioning [112], generation [2], information retrieval [21], and as feature extractors for other downstream tasks [99].

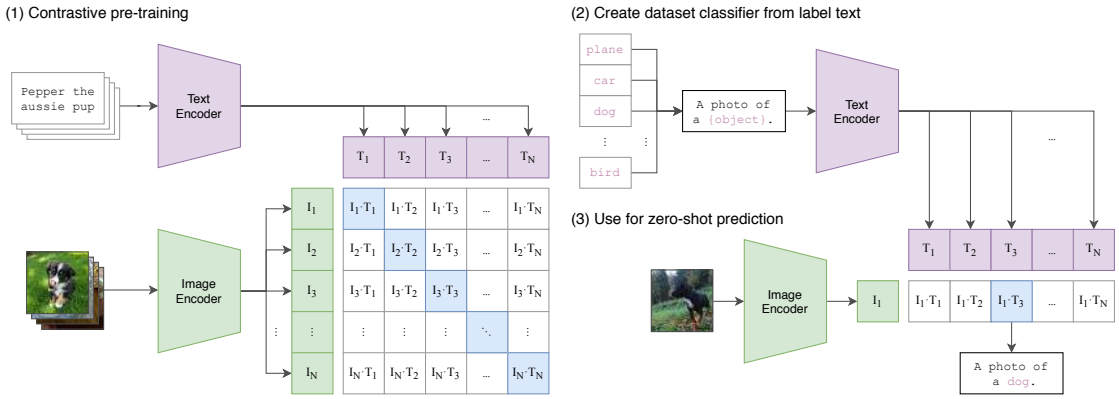


Figure 2.1: An general overview of CLIP's structure and use. Left: Batches of images and text embeddings are compared and CLIP predicts the matches. Right: At inference, CLIP can be used for zero-shot classification by providing a general description of the image class. Image credit to [78].

2.2.1 Contrastive Multimodal Models

Some of the most popular and powerful multimodal models in both academia and industry are based on contrastive pretraining. In this work I focus on CLAP [25, 111], a popular contrastive ALM based on its widely-used computer vision predecessor CLIP [78]. Because much of the literature on Text-only Training and the modality gap (see Section 2.3) focuses on the CLIP latent space, I briefly expand on its architecture and design.

2.2.1.1 CLIP: Contrastive Language-Image Pretraining

CLIP, or Contrastive Language-Image Pretraining, is a highly influential multimodal model in the computer vision community. At a high level, CLIP is trained to encode paired image and text data as vectors and predict matching pairs within

a given training batch. At inference time, CLIP is used to measure how well given images and texts match semantically. A simple use case is often providing the simple sentences with the names of certain classes, such as "A photo of a [class]", and seeing which class CLIP predicts to match each image [78]. Mathematically, given a batch of N paired image x_i, and text $y_i, i \in \{1, 2, \dots N\}$, CLIP seeks to minimize the symmetric InfoNCE [72] loss:

$$\begin{aligned} L_{\text{CLIP}} = & -\frac{1}{2N}\sum_{i=1}^{N}\log\left(\frac{\exp(f(x_i)^T g(y_i)/\tau)}{\sum_{j=1}^{N}\exp(f(x_i)^T g(y_j)/\tau)}\right) \\ & -\frac{1}{2N}\sum_{j=1}^{N}\log\left(\frac{\exp(f(x_j)^T g(y_j)/\tau)}{\sum_{i=1}^{N}\exp(f(x_i)^T g(y_j)/\tau)}\right) \end{aligned} \tag{2.1}$$

where τ is a temperature parameter, and $f(\cdot)$, $g(\cdot)$ are the normalized vector output of image and text encoders, respectively. The left term contrasts images and texts by penalizing incorrectly predicting the j_{th} text as matching the i_{th} image, and vice versa for the right-hand term.

Overall, the effect of this loss encourages the vector representations of paired or *semantically similar* image and text vectors to be near each other in the CLIP embedding space, and to encourage *dissimilar* pairs to be far apart. This fundamental property of contrastive learning models plays a crucial role throughout this work.

2.2.1.2 CLAP

CLAP, or Contrastive Language-Audio Pretraining, is mathematically and conceptually identical to CLIP, but with audio data substituted for images. Accordingly, the architecture for CLAP is built with an audio encoder instead of an image encoder, but the loss function and usage remain the same. Just like with CLIP, the most fundamental task one might perform using CLAP is to associate given audio with a set of example classes. In other words, seeing which audio is most like the sentence(s) "The sound of a [class]". Because the two models are so similar, much of the research done on CLIP and its variants are directly or conceptually extensible to CLAP.

2.3 The Modality Gap

Unfortunately, models like CLIP and CLAP suffer from the effects of the modality gap. In this section, I define the modality gap and discuss its origin and properties.

2.3.1 Definition

The modality gap refers to the phenomenon observed in deep, usually contrastive learning-based, multimodal models in which the embeddings from each modality lie in separate subspaces within the overall embedding space [53]. These subspaces are often entirely disjoint, and the distance between these spaces leads to the name "modality gap". For a more precise definition I turn to Liang *et al.* [53] who first

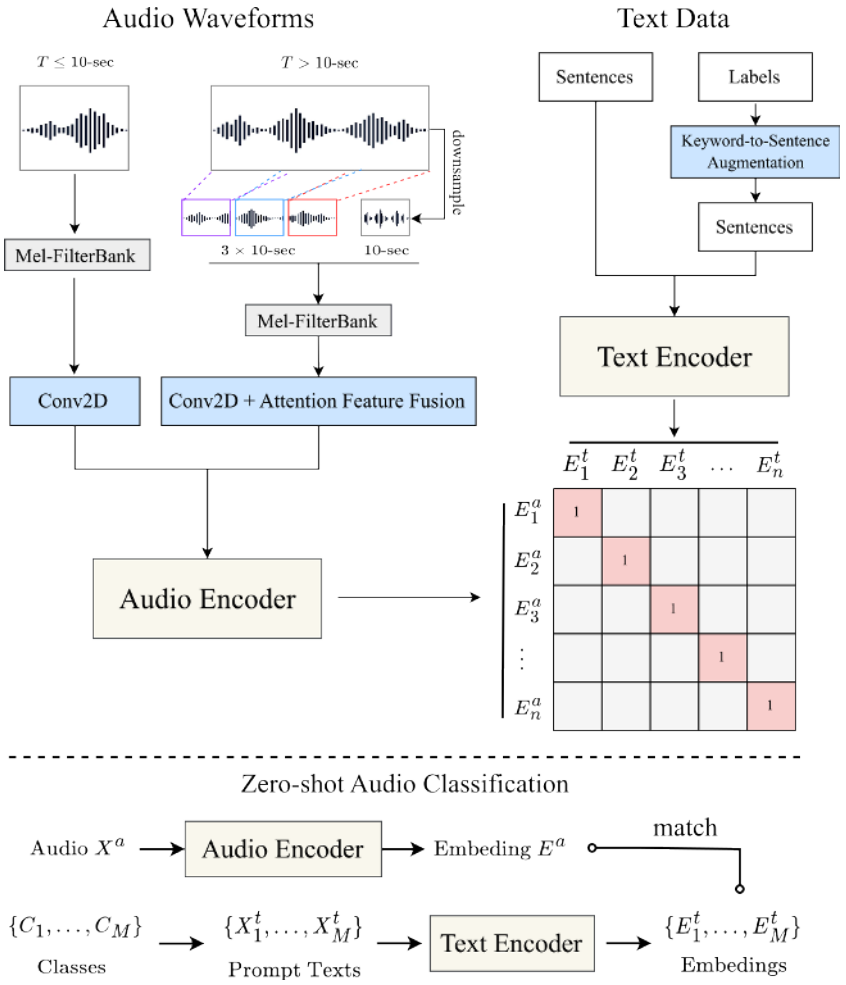


Figure 2.2: An overview of CLAP's architecture. CLIP-style training (Top) and zero-shot classification (Bottom). Image credit to [111].

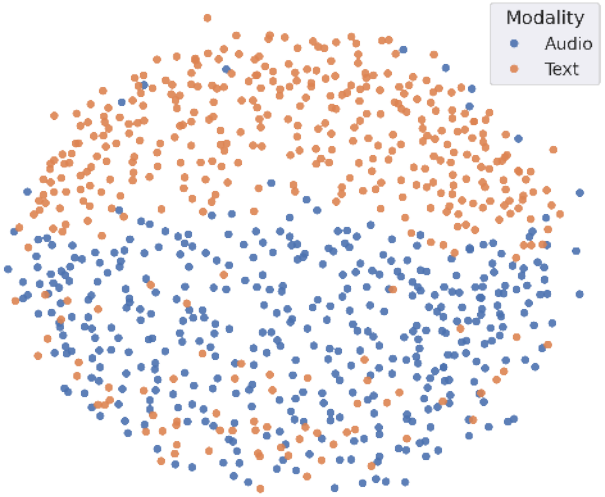


Figure 2.3: An example of the modality gap effect in CLAP. 400 random samples are drawn from the Clotho+ dataset (see Section 4.2.1) half from audio, and half from text, and projected from 512 down to two dimensions using multidimensional scaling. The two modalities are mostly separate with scattered outliers even in only two dimensions.

described the phenomenon in the CLIP embedding space. They define it as the difference between the centroids of each embedding subspace. Empirically, for N L2-normalized image embeddings x_i, and M text embeddings y_j this becomes:

$$\Delta_{gap} = \frac{1}{N}\sum_{i=1}^{N} x_i - \frac{1}{M}\sum_{j=1}^{M} y_j \tag{2.2}$$

The size, direction, and statistical properties of the resulting Δ_{gap} vector can then be used to characterize and measure the modality gap. A natural metric used by Liang *et al.* and others is the L2-norm of this vector. Examples are shown in Figure 2.3 and Table 2.1.

Datasets	$C_t \leftrightarrow C_a$	$Sb_t \leftrightarrow Sb_a$	$C_t \leftrightarrow Sb_a$	$Sb_a \leftrightarrow C_t$
$\|\Delta_{\text{gap}}\|_2$	0.439	0.440	0.596	0.464

Table 2.1: L2 norms between Clotho (C) and SoundBible (Sb) datasets. Audio and text embeddings are noted by subscripts a and t, respectively. The distance between modalities within datasets is approximately equal, while those between datasets are not. The distances across datasets are also greater than those within, supporting the assumption that the text embeddings that match the audio (within dataset) are closer in the CLAP embedding space than those embeddings that do not (between datasets).

While the exact origin and cause of the modality gap remains an active research area, Liang *et al.* theorize that the modality gap is an artifact of the deep learning process or contrastive learning objective [53]. Several other authors explore these ideas, and find that in the multimodal setting, the temperature parameter found in the loss function of CLIP and CLAP is negatively correlated with embedding alignment, and embedding distributional uniformity increases loss in the presence of mismatched data [103, 3]. Shi *et al.* hypothesize that the modality gap is simply a local minimum in the contrastive loss landscape, and show that through many training steps it can be overcome, and that increasing sensitivity via the temperature parameter extends the amount of training required [94]. Similarly, Fahim *et al.* suggest that the modality gap is a misnomer, and unrelated to multimodal settings. Instead, it is simply a property of the contrastive loss function [28]. Others have suggested the gap's origins are tied to the inherent difference in information contained in each modality in multimodal data and its effect on the behavior of contrastive learning [41, 91, 5, 4].

In addition to searching for its origin, much of the modality gap literature deals

with methods for overcoming or reducing the modality gap. In the context of this work, addressing the modality gap is essentially a form of domain adaptation. Because the CLAP embeddings of each modality are mostly disjoint, training a machine learning model on only one modality would not necessarily perform well on the other. In order to approximate audio data with text suitable for ToT, one needs to address the differences in their distributions. At a high level, domain adaptation is a set of techniques for solving machine learning problems in which the test or inference domain is different from the training domain. The principles behind domain adaptation fuel much of the modality gap literature and contribute to this work's experimental design. For an extended definition and overview of domain adaptation refer to [29].

Chapter 3: Related Work

In this chapter I provide a broad overview of the literature regarding single-modality training in multimodal settings, the modality gap, and machine listening. Additionally, I summarize several pertinent studies on each topic in greater detail, focusing on closing the modality gap, Text-only Training, and automatic timbre classification.

3.1 Text-only Training

Text-only Training, or more generally cross-modal or uni-modal training, and its predecessors have mostly been explored in the computer vision community. Uni-modal training has been applied to a variety of CV tasks, including: image captioning [50, 51, 70, 109, 31, 100, 108, 77, 116], generation [81, 123], zero-shot classification [36, 114], multimodal retrieval [40], visual question answering [98], category discovery [107], and improving multimodal embedding model alignment [115, 58].

Some of the most relevant studies to this work includes DECAP [51], an image captioning model trained with only text data. First, a memory bank of text embeddings are computed and stored. Second, image embeddings are approximated as a weighted sum of the text embedding, where each weight is proportional to the

similarity between each text embedding and the image embedding. While an important early ToT method, this method suffers from expensive data requirements, and is therefore not suitable for data-scarce environments like audio-timbre classification. Another relevant study was performed by Nukrai *et al.* in which they train an image captioning model without images [70]. They measure the distribution of distances between an image and several of its captions, and use this empirical sphere of influence to model modality alignment noise. Then by adding that noise to text embeddings, they mimic training on real image embeddings. Qiu *et al.* use a similar approach: they combine both noise injection and a nearest-neighbor strategy for zero-shot image captioning [77]. They bridge the modality gap with noise-injected text embeddings, and provide additional filtering at inference time by generating a set of potential captions and selecting the most similar result to the input image according to CLIP. Another important study was completed by Tam *et al.* who compare alternative methods for closing the modality gap [100]. They train to predict one caption given another augmented caption. Rather than applying stochastic data augmentation, Tam *et al.* rely on supervised and unsupervised alignment techniques, ranging from Mean Shifting (see Section 5.1) to optimal transport.

In the audio domain, ToT has been used in targeted sound extraction [61, 60, 87], generation [22], captioning [46, 117], and retrieval [39]. I take major inspiration from Deshmukh *et al.* [17] who train an audio captioning model with only text. Similarly to Nukrai *et al.* [70], they train their model with noise-augmented captions as input instead of audio, however they also compare using a supervised

linear mapping as an alternative to adding Gaussian noise.

3.2 Closing the Modality Gap

Without addressing the modality gap, ToT risks facing significant performance degradation. Broadly, these methods tend to be supervised [81, 22, 23] or unsupervised [70, 123, 36] mappings that act directly on one modality's embeddings to approximate embeddings in the other. Other work examines architecture or optimization changes [103, 3, 71, 94, 28], or changes to the geometry or properties of the underlying embedding space [12, 80, 120]. As my work focuses on utilizing pre-trained embedding models, I focus on the first category.

In one of the first works on approximating cross-modal embeddings, Ramesh *et al.* compare an autoregressive and diffusion model for generating CLIP image embeddings from text embeddings, then using a diffusion decoder to generate images conditioned on the pseudo-image embeddings [81]. Lee *et al.* [49] train a diffusion model on text embeddings, but during inference introduce image embeddings as a condition at an intermediate step in the reverse-diffusion process to guide text-embedding generation. Zhou *et al.* [122] similarly guide the generation of image embeddings from text embeddings by using "shifted" diffusion, a novel diffusion technique that incorporates prior knowledge to shift from a standard Gaussian to more desirable distribution, namely, the distribution of image embeddings in CLIP.

After introducing the modality gap, Liang *et al.* mitigate the gap by vector

addition [53] (see Mean Shifting in Section 5.1). Nukrai *et al.* compare that vector addition method with adding Gaussian noise and prefer the latter approach [70], while Tam *et al.* find that it is preferred over using optimal transport or PCA features [100]. Saijo *et al.* compare several gap-closing methods, including Gaussian noise and PCA features, and find that adding noise is the superior method [87].

Several other authors use random noise and stochastic methods to bridge the gap [36, 61]. Gu *et al.* employ Gaussian noise to bridge from text to image embeddings on several tasks, and compare a standard normal distribution with a variance hyperparameter to an empirically measured covariance matrix, finding some improvement with the latter [36]. Ma *et al.* randomly linearly interpolate between audio and text embeddings in their sound extraction method CLAPSep [61], but change to a retrieval and Gaussian noise-based method in more recent work [59]. When given both text and audio modalities as a query, they replace the separate embeddings with a new stochastic linear interpolation of the audio and text embeddings, linearly closing the modality gap between the two.

In [103], Udandarao explores two point-cloud alignment methods: Orthogonal Procrustes [92] and Relaxed Procrustes, and they find that for both methods the modality gap is significantly reduced and model semantic consistency increases. A study by D'Orazio *et al.* bridges the gap by solving a small orthogonal Procrustes problem for each element in the training dataset [23]. To transform CLIP text embeddings to image embeddings, they retrieve the top k most similar text and corresponding image embeddings from an auxiliary dataset. Then, they solve the

Orthogonal Procrustes problem to find the matrix that best aligns them. By applying that transformation matrix to the input text, it is projected to the CLIP image subspace. Similarly, a study by Role *et al.* finds both optimal transport and PCA features inferior to a spectral embedding method [86].

A method to improve zero-shot classification with CLAP-like models is proposed in [47] and used by Acevedo *et al.* to better sound classification with background noise present [1]. Rather than using class names or text prompt embeddings for comparison with test audio, as is typically done in CLAP and other CLIP-style classification pipelines, one can learn "prototype" embeddings that better capture the underlying idea or class. The prototypes are collected by finding the centroid of the top k most-similar audio embeddings for a given prompt or class name. Both studies found that using the prototypes led to higher classification metrics than any set of text prompts. Additionally, Acevedo *et al.* take a further post-processing step after computing similarities between class prototypes and test audio. In order to account for the effect of background noise, they compute the similarities between the audio embedding of the background and the class prototypes, as well as embeddings of text descriptions of the background noise and the prototypes. By subtracting the similarity scores of the test audio (foreground sounds with background noise) and either the background audio or text description embedding similarities, they change the overall class similarity distribution, suppressing the affect of background noise and amplifying that of the foreground sound.

A similar prototype and logit-based approach is taken by Barbier *et al.* and

applied to few-shot remote sensing and object detection [8]. They adapt a Vision-Language Model for few-shot object detection by training a bounding-box regressor by comparing similarity scores of image region embeddings and class name embeddings. Then, for better performance downstream, they create prototype image embeddings similar to [47], but with a variety of selection criteria. However, to account for the difference in similarity-score distributions between the prototypes and class names when compared with region embeddings (the modality gap), at test time they directly alter the logits by standardizing them to match the training distribution.

Alternatively, An *et al.* propose both post-processing and peri-training methods to close the modality gap in CLIP [5]. They empirically show that some dimensions of CLIP embeddings in both modalities show strong activations regardless of input, and then prove theoretically that activations of this kind limit cosine similarities between modalities and contribute to the modality gap. They further hypothesize these activations are caused by the encoders learning modality specific information. To mitigate large activations they investigate clipping and standardization and show that standardizing the embedding distributions (subtracting their means and renormalizing with the Frobenius Norm) greatly reduces the modality gap and improves retrieval. However, because this methods requires access to the entire dataset of embeddings, they introduce an algorithm for tracking a moving mean and norm through iterative batch normalization [5].

In [118], Zhang *et al.* adopt the definition of the modality gap used by Liang *et al.* and empirically show both that in CLIP the modality gap can be approx-

imated by a constant vector (such as the Δ_{gap} vector used by Liang *et al.*), and that the gap vector is orthogonal to the span of text and image embeddings. Assuming these to be true, and using additional empirical evidence, they prove that a linear classifier that minimizes a regularized quadratic loss is equivalent in both modalities. In other words, under the appropriate conditions, a simple linear classifier trained on just one modality would be *identical* to one trained on data from the other modality. They test this experimentally on the multi-label classification dataset MS-COCO [56] by training a classifier on image embeddings and testing on text embeddings (and vice versa) with and without closing the gap. In contrast to Liang *et al.*, to close the gap they subtract the mean of the input modality's embeddings from each example rather than adding the mean difference vector (see "C2" in Section 5.1). They argue that by re-centering the embeddings to zero they better match the theoretical assumptions and that there are no inconsistencies between modalities. Ultimately, they find that closing the gap improves classification metrics.

Zhang *et al.* extend their analysis in [119], in which they find that in contrastive models like CLIP (and therefore by extension CLAP) there are near-constant dimensions in the embeddings of both modalities—dimensions that do not contain any information. Zhang *et al.* argue that this is a source of the modality gap and provides further evidence that the orthogonality condition assumed in [118] is true. Finally, they experiment with a modified version of the method in [118]. This time in addition to the modality gap, they assume there is some "alignment noise" that can be modeled by a Gaussian distribution. To simulate the noise, after re-

centering the embeddings to zero they add a random Gaussian noise vector. These important theoretical works lay the foundation for the experimental methodology in this work and are explored further in Section 5.3.1.

3.3 Timbre Classification

Machine listening, or computer audition, is a subfield of machine learning that focuses on automatic and learned methods for understanding sound. Some of the major applications in machine listening are generation [15, 69, 2], representation or feature learning [7, 52], captioning [27, 67], speech recognition [113], retrieval and/or recommendation systems [120], musical understanding [57, 16], and composition, performance, and other tools [54, 93].

The study of human description and perception of timbre is most often found in the musicology [48, 84] and psycho-acoustic [96, 11, 106] literature. Relatively few studies in the machine learning community have focused on the prediction or linguistic description of timbre, choosing to use the timbre of sounds as features rather than labels [38, 6, 96]. Mannone *et al.* cluster listener associations with timbre and color [63], and Reymore *et al.* differentiate between fine-grained timbral descriptors of noisy instruments [83]. Mapping timbre has been investigated in the context of digital music production [90, 104, 55] and audio effects [13]. Audealize [93] uses a crowd-sourcing approach to learn connections between words and digital audio effects.

With the rise of LLMs, interest in how well LLMs "understand" concepts like

timbre has grown in popularity [64]. Saitis *et al.* [88] and Siedenburg *et al.* [97] compare human and LLM timbre ratings of instruments and find similar intra-rater agreements as well as moderate rating correlations. Czedik-Eysenberg *et al.* measure the similarities between human perception of three timbral descriptors and popular audio feature extraction software libraries and five variants of CLAP [14]. Tian *et al.* perform a similar analysis between 18 representations [102]. Finally, Velissaridis *et al.* compare language and audio-language models equipped with timbral descriptors in instrument classification and emotion recognition [105]. Most relevant to this work, directly predicting timbral descriptors has been performed in K-Pop vocal styles [43], musical instruments [42, 85], and general audio [73].

Chapter 4: Datasets

In this chapter I describe the main datasets I employ in my experiments. First, I introduce the datasets on which I evaluate my timbre classifiers: the AudioCommons Timbral Characterisation Tool Development Datasets [75]. Next, I review the two audio captioning datasets I combine to learn methods for closing the modality gap: Clotho and SoundBible. Lastly, I describe my methodology behind creating synthetic caption-timbre datasets for use in text-only training of timbre classifiers.

4.1 Timbre Classification Datasets

One of the biggest barriers to multimodal machine learning is a lack of data, especially when trying to solve niche problems or those about which data is particularly difficulty to obtain. Partially motivating this work, the lack of large and high-quality audio-timbre datasets hinders researchers' ability to improve machine understanding of music and general audio, specifically in using natural language to describe human perception of audio. To overcome the lack of data in the non-music audio timbre space I employ three methods for data labeling, curation, and creation. First, I re-purpose existing audio captions as raw text data and label it with an pretrained model. Second, I invoke an LLM, `Llama-3.1-8B-Instruct`, to generate large quantities of pseudo-captions featuring timbral descriptors and label

them based on the provided prompt. Third, I create a "text" dataset entirely from noise centered around class names in the CLAP embedding space. The names and sizes of these datasets are summarized at the end of this chapter in Table 4.5.

Because I investigate the scenario in which there is no available data whatsoever, arguably the most important decision a practitioner of ToT must make is how to source the text data. Depending on the exact nature of the desired dataset, there are numerous ways to create text-based synthetic data. Wang *et al.* compare methods for closing the modality gap using four datasets with different levels of multimodality [110]. Qi *et al.* scrape text from the web and use filter models to remove unwanted data [76]. A similar approach uses a captioning model and filter to generate synthetic captions for both images [50, 120] and audio [45]. Zhou *et al.* create pseudo-captions by comparing images to a vocabulary of word embeddings and constructing sentences with the most similar words [121]. As I do in this study, LOVM [114] directly generates text data via LLM by prompting for descriptions based on class labels.

4.1.1 Audio Commons

The principle audio datasets used in my work are the AudioCommons (AC) Timbral Characterisation Tool v0.2 Development Dataset(s) [74, 73] developed to create the `Timbral Models` python package [75]. This tool is designed to measure how well seven different timbral descriptors match a given audio recording, as well as detect the presence or absence of reverb. The dataset is comprised of eight

separate subsets corresponding to each descriptor. See Table 4.1 for the list of timbral descriptors. For each descriptor a collection of sounds were sourced from FreeSound[1]. Then, for all words except *Reverb*, human annotators were presented with a set of eight sounds and rated each on a continuous scale from 0-100 on how well that word described the sound relative to the others. *Reverb* was handled differently as it was treated as binary: either absent or present, and simply annotated by the dataset creators.

To simplify creating new text data, I binarize the continuous-valued annotations into two classes. Not only does this unify all eight subsets into one data format, it reduces the difficulty of the downstream task and data-creation processes. That way, one can more clearly see the effects of ToT on an otherwise simple problem, while still allowing for many synthetic data generation options. First, I calculate the mean of all continuous-valued annotations, then set all values less than 50 as the negative class (e.g., *Not Bright*) and those 50 or greater as the positive class (e.g., *Bright*). Finally, I compute the CLAP embeddings for all audio files, and use the pairs of embedding and binary label as the testing data. Table 4.1 provides the specific details on dataset sizes and label distributions of the resulting seven binary datasets, along with the *Reverb* subset.

[1] `https://freesound.org/`

	Booming	**Bright**	**Deep**	**Hard**	**Reverb†**	**Rough**	**Sharp**	**Warm**
Count	102	210	218	206	220	212	165	184
% Positive	30	62	61	67	53	55	60	42

Table 4.1: Sizes and class balance of each timbral descriptor audio dataset.
†I use half of the reverb dataset ($n \approx 400$) for testing and training the audio baseline, and the other for development.

4.2 Audio Captioning

Several of the methods I explore for bridging the modality gap require supervised learning (see Section 5.1). I train these models to close the modality gap with the concatenation of the Clotho and SoundBible audio captioning datasets.

4.2.1 Clotho

Clotho [24] is an audio captioning dataset that consists of 4981 audio samples taken from the Freesound platform. All recordings range from 15 to 30 seconds in duration and have five corresponding captions for a total of 24,905 captions. The captions range from 8-20 words, and were crowd-sourced using Amazon Mechanical Turk[2] by annotators from english-speaking countries.

Each recording uploaded to Freesound is assigned tags by the creator to help users find that recording when searching within the site. The authors designed Clotho to span as wide a range of topics and sources as possible, and use these tags to measure how broad a range the recordings cover. Consistent with the

[2]https://www.mturk.com/

literature, all recordings with tags that might indicate music or speech were automatically discarded. The authors prioritized natural sounds, as speech and music are challenges in their own right and deserve separate datasets. Additionally, the authors chose to remove recordings with tags that were related to sound effects. Additional details and information on post-processing methods can be found in the original paper [24]. An example set of captions is provided below in Table 4.2. As Clotho and the AudioCommons datasets are both sourced from Freesound, any duplicate files are removed from the Clotho set to preserve as much labeled timbre data as possible.

File Name	**Caption 1**	**Caption 2**	**Caption 3**	**Caption 4**	**Caption 5**
Distorted AM Radio noise.wav	A muddled noise of broken channel of the TV	A television blares the rhythm of a static TV.	Loud television static dips in and out of focus	The loud buzz of static constantly changes pitch and volume.	Heavy static and the beginnings of a signal on a transistor radio

Table 4.2: A set of captions from the Clotho dataset in which all captions corresponds to the same source audio [24].

4.2.2 SoundBible and WavCaps

The SoundBible[3] dataset is only a small subset of the much larger audio captioning dataset WavCaps [68]. WavCaps consists of approximately 400,000 audio recordings and paired captions taken from four existing datasets and web sources, with synthetic captions generated by ChatGPT. WavCaps combines recordings from Freesound (262,300), the Strongly-Labelled Subset of the audio classification dataset AudioSet (108,317) [33], BBC Sound Effects[4] (31,201), and SoundBible (1232)—a free sound effects sharing site. Since each source has its own set of tags, descriptions, and/or metadata, Mei *et al.* [68] task ChatGPT to re-write the given information as full sentences. While this process can introduce noise, the large scale of the dataset usually compensates for any mistakes. An example of the outputs is shown in Table 4.3.

I rely on the SoundBible subset for two reasons. First, because SoundBible consists of sound effects, it naturally extends upon the subjects in Clotho, from which sound effects are notably excluded. Second, because SoundBible is relatively small, including it with Clotho would not significantly alter the overall distribution of sound sources in a combined dataset. Together with Clotho, the combined audio captioning dataset consists of 6213 audio recordings and 26,137 text captions. I refer to this combined set as Clotho+.

Additionally, as I utilize the captions of WavCaps in one of the methods for sourcing synthetic data (see Section 4.3.1). To avoid data leakage, I remove all

[3]https://soundbible.com/

[4]https://sound-effects.bbcrewind.co.uk/

Table 4.3: Examples of raw data and ChatpGPT outputs used in WavCaps. Taken from [68].

Data sources	Raw descriptions	ChatGPT's output
FreeSound	This sound is of a book falling down the staircase in the Stanford University library west stacks.	A book is falling down a staircase.
	After 2h and 15min. Excerpt.	Failure.
BBC Sound Effects	Timber & Wood - Rip saw, carpenters' workshop.	Someone is using a rip saw in a carpenter's workshop.
	Motor Car: Ford Ecsort 1300 (Automatic) - Ford Escort 1300, exterior, passing with horn. (1300cc engine, automatic transmission.)	A car is passing with its horn.
SoundBible	Tasmanian Devil growling screaming hissing. Warning sounds from a Tasmanian Devil in Zoo.	An animal is growling, screaming, and hissing.
	Large Tibetan Bells ringing in a temple. Could also use for Monastery or Monks.	Bells are ringing.
AudioSet SL	['Accelerating, revving, vroom', 'Race car, auto racing']	A race car is accelerating and revving.
	['Female speech, woman speaking', 'Whoosh, swoosh, swish']	A woman is speaking while something whooshes.

duplicate files and captions from the Freesound subset of WavCaps that are also found in the Clotho and AudioCommons datasets.

4.3 Synthetic Text-Timbre Data

This section contains the methodologies for constructing or sourcing the synthetic text-timbral descriptor data used in my ToT experiments. These data serve as the training data and are tested on the real audio-timbre data from the AudioCommons datasets.

A {label} sound.
A sound that could be described as: {label}
An audio clip with a {label} quality.
An audio clip that sounds {label}.
A sound/audio clip that has or contains {label}.

Table 4.4: The prompts used for semantic similarity labeling with CLAP and SBERT. Each prompt is embedded into the model's embedding space and the mean is calculated and re-normalized for use as the comparison pseudo-prompt.

4.3.1 Pseudo-Labeling Real Captions

The first of the three dataset sourcing methods I explore in this work is labeling real captions with text-based semantic similarity models. The process consists of labeling extant captions by comparing each caption with a predetermined sentence describing the label. In this work I refer to these as "Anchor Prompts". I write five versions of the anchor prompt to reduce bias and average their CLAP embeddings to create a central pseudo-prompt. The prompts are inspired by the literature [79, 18, 10] and are listed in Table 4.4.

As in works like [79], each caption is first embedded using a pretrained multimodal encoder. Then, each class label's mean anchor prompt embedding is compared with the caption's embedding and the class with the highest cosine similarity is selected. I use two large models that capture semantic relationships between text pairs: LAION-CLAP [111] and the SBERT [82] model all-MiniLM-L12-v2. CLAP is the natural choice for audio-related text-text similarity, since it is explicitly trained to match text and audio. This also represents the natural choice for a

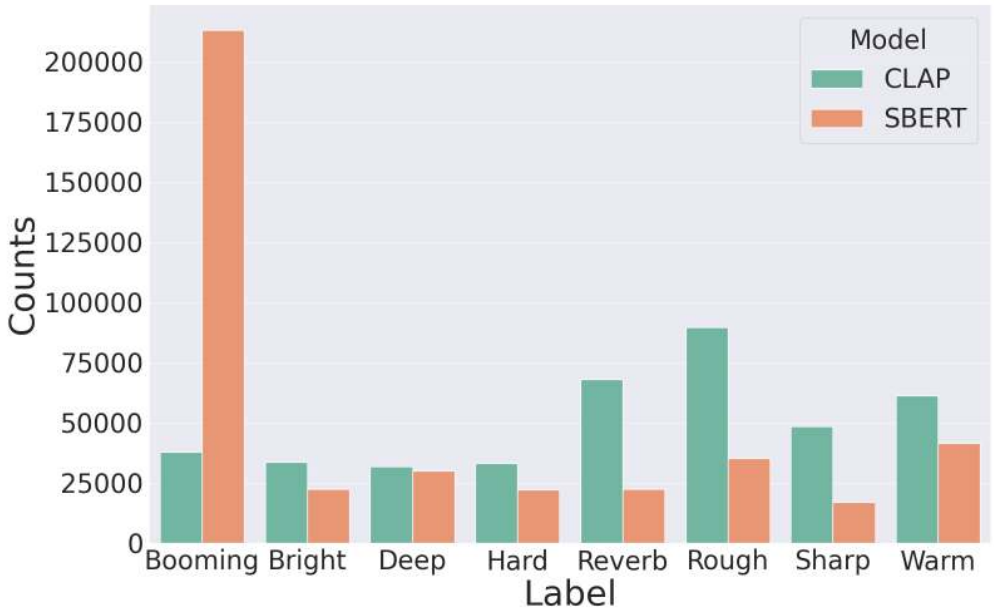


Figure 4.1: Class label distributions of WavCaps [68] as determined by text-text similarity between captions and a label prompt. I compare CLAP [111], an audio-language model, and SBERT [82], a language semantic similarity model.

practitioner already using CLAP as an encoder to also use it as the labeling tool in their workflow. SBERT juxtaposes CLAP as a text-only semantic similarity model trained on general text data. While similarly designed to correctly identify semantically-similar text pairs, SBERT was not necessarily intended to be used on audio domain-specific tasks, and is selected to compare strategies of using a domain-specific similarity versus a general language similarity model.

I choose WavCaps [68] as the caption dataset because it is both large and widely used in training multimodal models for tasks like audio captioning. Because much of the data used in WavCaps was also used to train LAION-CLAP, I can be more confident that the concepts that the captions describe are in CLAP's

distribution, and the label predictions are likely more accurate than otherwise [111]. The final distributions of labeled data for each model are shown in Figure 4.1.

The advantage to this method is that it uses real text data and is often cheaper and easier than sourcing new data altogether. However, this method relies on the performance of the model(s) chosen to label the data. As can be seen in Figure 4.1, the bias of the chosen model can dramatically alter the final dataset's distribution. Additionally, because label noise is particularly harmful to machine learning models, one must rely on finding large enough quantities of real data to overcome the noise present in the labeling process. This is of particular concern since this work investigates the scenario in which there is no labeled data for the downstream task whatsoever, meaning there is no guarantee suitable real text data is available as a substitute.

4.3.2 Pseudo-labeled LLM-Generated Captions

To contrast with labeling existing captions, another natural choice is to use one of the widely available LLMs to generate captions based on a desired class [114]. To test this method, I employ the `Llama-3.1-8B-Instruct`[35] LLM. For each label in the AudioCommons datasets I prompt the LLM to generate descriptions of audio events that might be described using that label. For each label I generate approximately 15k samples after removing duplicates, and assign a class label based on the prompted timbral descriptor.

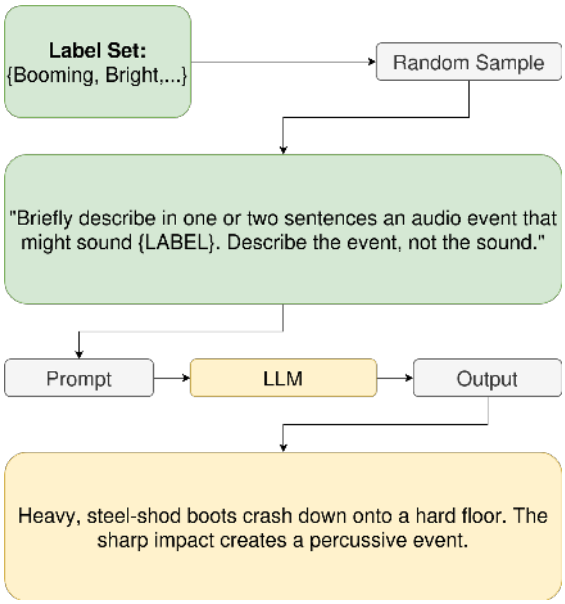


Figure 4.2: The process for generating text data using an LLM. First, a label is sampled from the desired class label distribution. Then, the LLM is prompted to describe an audio event that corresponds to that descriptor. The outputs and sampled label are then used as training data.

The overall process is summarized in Figure 4.2. First, I select a set of timbral descriptors to use in the LLM's prompt and as labels. This process is extensible to any set of descriptors, but I limit the set to the labels in the AudioCommons datasets. For each desired caption, I randomly sample a descriptor that acts as the timbre label and seed for `Llama`'s caption. For the binary classification experiments conducted in this work I uniformly sample one descriptor from the set of timbre words, but one could choose any distribution to match should the desired dataset require it. One of the advantages of this technique is the flexibility it provides the practitioner: the ability to tailor the training dataset to one's needs.

Once the label(s) have been sampled, I prompt `Llama` to generate a caption for

a description of a sound that might fit that word. The full prompt is in two parts. First, the system message is a constant message used in all prompts and reads:

> "You estimate events that create sound given a set of characteristics of the sound. You may use the given words in the description but you are not required to do so. Be succinct and as realistic as possible."

The second half of the prompt is unique to each prompt instance and is composed of a template that is modified to include the randomly sampled tag(s) from the previous step.

> "Briefly describe in one or two sentences an audio event that might sound `{sampled adjective(s)}`, and has a(n)/some `{sampled noun(s)}`. Describe the event, not the sound."

in which the grammar is corrected to account for any number and combination of adjectives and/or nouns. An example of the outputs is shown in the last box in Figure 4.2. Again this method is extensible to arbitrary numbers of descriptors with varying parts of speech, however, recall that for the body of this work, I only sample one descriptor to simplify this process and the downstream task to binary classification.

One of the major drawbacks of this method, however, is that it requires tuning hyperparameters. `Llama` requires setting several hyperparameters that can drastically affect the outputs it produces, namely the *temperature* and *top_p* values. Each of these parameters affect the model's randomness. The *temperature* hyperparameter alters the generation steps by altering the underlying token probability

distribution. If set to 0, the model becomes deterministic, and the model is unaffected when set to 1[5]. The *top_p* hyperparameter changes the pool of tokens from which the model is allowed to sample. Only the smallest set of tokens whose probabilities sum to the value of *top_p* are available for sampling. With possible values found in $[0 - 1]$, lower values encourage consistency as only the most likely set of tokens are chosen, while larger values allow for a wider range of tokens to be sampled.

To encourage diversity in captions I compared values for both *temperature* and *top_p* by manually inspecting generated outputs for coherence and diversity. For low and moderate values of *temperature* and *top_p* the outputs almost always shared the same sentence structure and were not diverse in subject. However for greater values the captions would often lose syntactical and semantic coherence, for example, the first half of a sentence might not match the second. I increased the value of both hyperparameters until 5 of the first 20 outputs were no longer grammatically correct or consistent in subject matter, then I slowly backtracked until all of the first 50 were coherent. The final values of *temperature* and *top_p* were set to 0.7 and 1, respectively.

One might prefer this method over labeling real data because one could theoretically generate an arbitrarily large amount of data. However, generating high-quality text data using LLMs is expensive and ultimately not as reliable as real data. This method suffers from the same possible label noise mentioned above, because the generated text might not actually correspond to the provided label

[5]Note that these values may be swapped depending on the implementation.

as desired. For example, I empirically found that despite selecting hyperparameters that promoted varied responses, many of the generated captions described the same scene in slightly different words. This limits the overall distribution of captions and corresponds to an effective dataset size smaller than it seems. And just as seen above, the exact output distribution depends on the model chosen, but with the added burden of tuning hyperparameters.

4.3.3 Label-Centered Gaussian Noise

For my final method, I explore training purely on Gaussian noise centered around the anchor prompts. This way, a practitioner need not bother with the potentially expensive processes of pseudo-labeling or generating captions from scratch. Inspired by [119], who model discrepancies in modality alignment as Gaussian noise, I utilize the anchor prompts and their mean embedding as the centers of six separate Gaussian distributions with a shared variance I measure following the procedure in [70, 17]. In other words, each label's distribution is constructed as a mixture of six Gaussians, each with the same variance, but centered around different anchor prompts. Each training example is "drawn from" one particular Gaussian distribution centered at the chosen anchor prompt in the CLAP text embedding space. Then, the overall dataset distribution would be a collection of these Gaussian mixtures, one for each label. I generate approximately 25k samples for each label.

This method is the least labor-intensive of the three I present and is easily

	WavCaps	LLM-Generated	Label-Centered Noise
Count	401772	119906	200016

Table 4.5: Total sizes of the three text datasets used in the classification experiments.

scalable to an arbitrary size. However, it assumes that the embedding model has a general "understanding" of the classes in both modalities, and that the anchor prompts act as a good measure of the mean of each class's distribution in the text embedding subspace. Luckily, we already assume the embedding model has at least some correlation between the desired data and descriptive text labels, otherwise the contrastive-learning objective underpinning the CLAP model would have failed. Since I can empirically see the effectiveness of the CLAP model and its variants [61, 37], I am able to implicitly verify this assumption.

To reiterate, I borrow the labels found in the real audio-timbre dataset AudioCommons to create four text datasets using three methods. First, I label existing audio captions with the CLAP and SBERT models via semantic similarity with the label names. The former takes advantage of domain knowledge, while the latter is designed for text-text semantic similarity comparisons. Second, I use an LLM to generate captions based on a provided prompt. Given a label from the AudioCommons set, I prompt the LLM to describe an audio event that matches the timbral descriptor. Lastly, I simply add noise to the CLAP embeddings of the AudioCommons labels, mitigating the need for real text data.

Chapter 5: Methodology

In this chapter, I discuss the methods with which I address the modality gap, my experimental design, evaluation metrics, classification models, and the comparative baselines I select. An overview of the case study methodology is provided in Figure 5.1

5.1 Bridging the Modality Gap

Bridging the modality gap is a specific form of domain adaptation, meaning applying machine learning to one's out-of-domain dataset will likely result in poor performance. Therefore, it is critical that a practitioner of ToT address the modality gap problem. While I have shown there are many methods used in practice, I narrow my focus to common baselines, the methods with the most theoretical backing, and extensions thereof. I compare three unsupervised and four supervised methods, and noisy versions of each, for a total of fourteen methods. First, I describe the unsupervised methods:

1. Nothing: applying no augmentation.
2. Mean Shifting: the method first explored by Liang *et al.* in their work that introduced the modality gap [53].

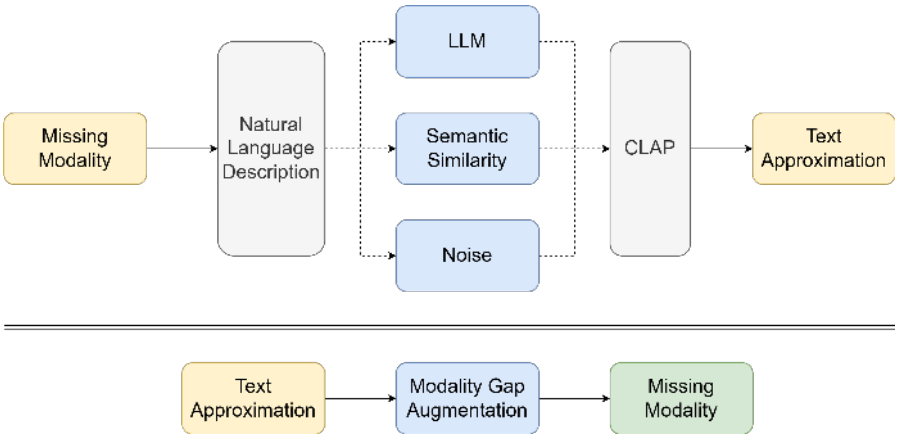


Figure 5.1: A simplified overview of the case study. **Top:** the desired audio data is described by natural language, such as a label or Anchor Prompt. Next, the training data is generated by one of the three dataset creation methods discussed in Section 4.3 and is embedded in CLAP. **Bottom:** an augmentation method is selected to combat the affect of the modality gap. The resulting approximation of audio embeddings is used as the data for the classification experiments.

3. Connect & Collapse (C2): the method described by Zhang *et al.* [118, 119].

These unsupervised methods are attractive because they do not require any external paired multimodal training data, which means they are perfect for the kinds of problems for which ToT should be applied. Recall that problems well-suited to ToT are those with little to no real training data, those with access to a multimodal embedding model, and those with characteristics easily describable with natural language (see Section 1.2). Under those circumstances, unsupervised methods like those listed above may be the only options available. Next, I describe the supervised methods:

1. Linear: a one-layer linear neural network trained to directly map from text to audio embeddings.

2. Nonlinear: a two-layer version of the shallow network described above.

3. LinearN: the same linear model, but trained with Gaussian noise added to the training data.

4. NonlinearN: similar to the above method, but using the two-layer neural network.

While supervised methods may require external data, they are both powerful and a natural choice when paired data is available. Finally, I also examine adding Gaussian noise as a secondary augmentation step in addition to those listed above.

5.1.1 Method Details

In more detail, Mean Shifting involves measuring the mean difference vector between each modality, then using that mean vector to shift samples from one modality towards the other. The mean difference vector is defined as:

$$\Delta_{gap} = \frac{1}{n}\sum_{i=1}^{n} a_i - \frac{1}{m}\sum_{j=1}^{m} t_j \tag{5.1}$$

in which a_i and t_j refer to the i_{th} audio and j_{th} text embedding respectively. Then, by adding the mean vector to the text embeddings, one can approximate the paired audio vectors:

$$\hat{a}_i = t_i + \Delta_{gap} \tag{5.2}$$

I store the mean audio embedding of the Clotho+ dataset (see Section 4.2.1) and calculate the mean of each text dataset or batch during training.

C2, short for Connect & Collapse, is one of the most important methods I explore in this work. Zhang *et al.* proved under certain empirically verified assumptions that C2 removes the modality gap [119]. C2 involves centering each modality at zero by subtracting the expectation, or mean embedding vector, from each modality during both training and inference. Similar to Mean Shifting, C2 shifts one modality towards the other, but while Mean Shifting pushes one modality towards the center of the other, C2 shifts both modalities towards the origin. Mathematically, for audio embeddings dataset $A \in R^{N \times d}$ and paired[1] text embeddings dataset $T \in R^{M \times d}$,

$$\begin{aligned} A_{\text{C2}} &= A - E[A] \\ T_{\text{C2}} &= T - E[T] \end{aligned} \tag{5.3}$$

Notably, C2 requires augmenting the data at inference time in addition to during training to also center incoming data at the origin. Unfortunately while the two modalities are nominally aligned at zero, when augmenting real data Zhang *et al.* claim the embedding vectors suffer from alignment noise that prevents paired text and audio from being perfectly equal [119]. To account for alignment noise, the Zhang *et al.* introduce a noisy version of C2 called "C3" (Connect, Collapse, Corrupt). C3 expands on C2 by injecting Gaussian noise into the training data

[1]Ideally, each audio and text example would be a unique pair, but in practice—as is the case with the Clotho dataset—there are often many text examples for each example of audio.

to mimic additional noisy samples centered near the real data and to help train models robust to noisy data. This also falls well within the pattern of the literature, in which many authors use added Gaussian noise or other random methods for bridging the modality gap [70, 17, 87].

The supervised methods involve learning a transformation from text embeddings to audio embeddings. I compare both a linear transformation in the form of a shallow one-layer neural network, and a non-linear transformation via a two-layer neural network with hidden dimension size equal to the embedding dimension, and ReLU activation. Both are trained to minimize the mean squared error between the paired text and audio embeddings in the Clotho+ dataset with batched gradient descent and the Adam [44] optimization algorithm. Next, I train a second version of these two regression models and inject Gaussian noise into the text embeddings during each training batch to encourage models robust to noisy data and to improve generalization. I select the Gaussian noise variance parameter based on the method in [17] by measuring the mean infinity norm between random samples of the Clotho+ dataset.

These methods may be suitable when a practitioner has access to paired audio and language data unrelated to the downstream task. By leveraging this data, they can consider a version of this augmentation technique for addressing the modality gap while still training the main model entirely on text. For example, my case study is on timbre classification, but the Clotho+ dataset consists of audio and captions and has no direct relationship to my downstream task.

Lastly, because I want to examine both C2 and C3, for a fair comparison I

also add noise to each of the other methods listed above. I add Gaussian noise after the application of the augmentation technique as a second step. When I add additional Gaussian noise the process becomes:

$$\begin{aligned} \hat{a_i} &= g(t_i) + \epsilon \\ \epsilon &\sim \mathcal{N}(0, \sigma^2) \end{aligned} \tag{5.4}$$

for some augmentation function $g(\cdot)$ from the set above, and text embeddings t_i, $i \in \{1, 2, 3 \ldots, n\}$. I use the same variance parameter described above.

In summary, I compare three unsupervised methods: a baseline of no augmentation (Nothing), Mean Shifting, and C2, with four supervised methods: linear regression (Linear), nonlinear regression (Nonlinear), and versions trained with added noise, LinearN and NonlinearN. Furthermore, I present both clean versions, and the same augmentation methods with additional added Gaussian noise.

5.2 Metrics

Before describing the experiments I perform, I first must establish the metrics by which I evaluate them. I track eleven metrics to measure the modality gap and change in representations. For classification experiments, I choose the F_1-score as my metric because it is preferred for unbalanced binary classification tasks. First, I describe the metrics that track the modality gap

5.2.1 Modality Gap Metrics

Following Liang *et al.*, one of the key metrics I use to measure the extent of the modality gap is the Mean L2-Distance, or simply Mean Distance [53]. This metric measures the L2-distance between the modality centroids, or in other words, the average Euclidean distance between the embeddings of each modality. For N text t and audio a embeddings, the definition of Mean Distance is:

$$\text{MD} = \left\| \frac{1}{N} \sum_{i=1}^{N} (a_i - t_i) \right\|_2 \tag{5.5}$$

This metric is the origin of the Mean Shifting augmentation, which intends to minimize this metric [53].

Next, Linear Separability is used to both describe the modality gap and measure to what extent the modalities are mixed [94]. This metric measures how well a linear classifier, typically an SVM or Perceptron, separates the modalities. Given data from two modalities $\{x_1, x_2, \ldots, x_n\} \in X$ with class label $y = 0$ corresponding to data from the first modality, and class label $y = 1$ corresponding to data from the other, Linear Separability is defined as:

$$\text{Sep} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(f^*(x_i) = y_i) \tag{5.6}$$

in which $\mathbb{1}(\cdot)$ is the indicator function, and $f^*(\cdot)$ is the ideal linear classifier. In other words, Linear Separability is the accuracy of the best possible linear classifier. Of course, one cannot access the ideal classifier, so I approximate this by

calculating the accuracy of a linear classifier trained with gradient descent. If the accuracy is high, this means that the classifier can easily separate the two modalities with a simple hyperplane. Conversely, if accuracy is low, then data from the two modalities are highly interspersed, and separating them would require higher-order methods. By definition, the modality gap implies at least some, but usually large amounts of, separability.

Next, the Approximate Hausdorff Distance refers to a modified Hausdorff distance that is more robust to outliers. Normally, this point-cloud metric measures the maximum distance between all possible nearest neighbors in each cloud. In other words, it finds the sample in each cloud that is as far as possible from the other cloud, and takes the maximum between those distances. For some distance function $d(\cdot)$

$$\text{Haussdorf} = \max\left[\max_{a\in A} d(A,T), \max_{t\in T} d(T,A)\right] \tag{5.7}$$

This metric is useful as it measures the modality gap at its most extreme as opposed to its average. However, this means that Hausdorff is quite sensitive to the presence of outliers, and can be uninformative if transformations that affect the point clouds on average have little effect on the outliers. For example, consider reflecting a point cloud across an axis: if the farthest point from the second cluster was very close to the axis of symmetry, its distance would be relatively unchanged compared to those points farther away from the axis. This lack of relative change would be reflected in little change in the Hausdorff distance, even if the rest of

the points in the cloud were perfectly aligned. The Approximate Hausdorff is a modified version that uses the 90^{th} percentile of distances instead of the true maximum cross-modal nearest neighbor distances. This way, it is more robust to extreme outliers, but less accurate in terms of its original intent, as it will generally reflect slightly smaller values than the real Hausdorff metric.

The final modality gap-based metric is the Sliced Wasserstein Distance. This metric is a variant on the Wasserstein distance commonly used in point-cloud analysis for its computational efficiency on high-dimensional data like the embedding vectors found in CLIP or CLAP. The distance is calculated by making K random projections u_k onto one-dimensional lines, and measuring the average Wasserstein Distance between the two clusters on that projected line. In this work I calculate the Wasserstein 2-distance, or in one dimension, the squared difference between the nearest point from the other cluster on the line:

$$\text{SWD} = \frac{1}{K} \sum_{i}^{K} \left[\text{SORT}(A^T u_k) - \text{SORT}(T^T u_k)\right]^2 \tag{5.8}$$

The projection values are sorted for efficient nearest-neighbor calculation. In this work I use $K = 512$ projections. Spatially, this metric measure how well the two clusters line up from any given direction, but it also acts as a distributional similarity metric, meaning if the distance is low, the two point clouds have a similar probability distribution, and vice versa when its value is high.

5.2.2 Representation Alignment Metrics

Next, I introduce the seven representation alignment metrics. Contrastive embedding models like CLAP are trained to align data such that those with similar meaning are placed nearby within the embedding space. Therefore, augmentations that alter the distance between or relative orderings of embeddings have altered the implied semantic relationship between data as imposed by the CLAP embedding space. The following metrics measure how much each augmentation changes that alignment. Importantly, altering the alignment is not necessarily harmful, nor is maintaining it required: consider the highly-likely event that some vectors might be misplaced in the CLAP embedding space. In this situation, altering the relative similarities between the misplaced vector and all others to "fix" its placement would actually increase overall model fidelity. Likewise, consider that maintaining the structure of the original space might be worse than the resulting augmented representation for some downstream task. That is, if the downstream task does not require semantic similarity to be encoded as cosine similarity within the vector space, it could be beneficial to lose that constraint.

First is a metric that specifically cares about the alignment between paired data. Pair Alignment measures the average cosine similarity between paired multimodal data. For paired normalized audio and text embeddings a_i, t_i:

$$\text{Pair Alignment} = \frac{1}{N} \sum_{i}^{N} a_i^T t_i \tag{5.9}$$

In the CLAP embedding space vectors with greater cosine similarity are in-

tended to have greater semantic similarity as well. From this perspective, even if the modality gap were to be entirely mitigated, if the Pair Alignment is decreased, the embedding space would have lost its original spatio-semantic relationship. However, a change in Pair Alignment does not necessarily correlate with the performance of a downstream task. For example, if one particular augmentation technique misaligns paired data, but successfully orders them in some useful way, then a practitioner might still consider it an overall success.

Unfortunately, Pair Alignment can only be measured on paired data. This means I cannot use it in a typical ToT scenario in which I have access to only unlabeled multimodal data or data from only one modality—text. To address this issue, I measure six alternatives to Pair Alignment that may be used on unsupervised or text-only data. Similar to Pair Alignment, the next three metrics Self-Consistency, Pseudo-Consistency, and Cross-Consistency, relate to the alignment of embeddings within the embedding space. First, Self-Consistency measures how the similarity values between the training data change after augmentation. It is calculated by subtracting the Normalized Frobenius Norm of the difference between the self-similarity graph of the data before and after augmentation from 1. For two row-normalized data matrices A and T of size $N \times d$:

$$C_{\text{Self}} = 1 - \frac{1}{N}||TT^T - \tilde{T}\tilde{T}^T||_F \tag{5.10}$$

Self-Consistency measures the average squared difference between the similarities of each graph. In other words, if an augmentation only slightly changes the

similarity between each example, then the graph could be considered consistent, and Self-Consistency would be high. Conversely, if there are dramatic changes and examples that were very similar before augmentation are moved far apart, then the similarity graph would no longer be consistent before and after augmentation and Self-Consistency reflects that with a low value. It is named Self-Consistency to reflect that the metric tracks if the self-similarity graph is altered by augmentation.

Second, Pseudo-Consistency is defined the same as Self-Consistency, but measures the difference between the original self-similarity graph and that between the original and augmented data:

$$C_{\text{Pseudo}} = 1 - \frac{1}{N}||TT^T - T\tilde{T}^T||_F \tag{5.11}$$

This is another way to measure how the augmentation affects the representational consistency: whereas Self-Consistency checks if the graph has changed relative to itself with augmentation, Pseudo-Consistency checks if augmented data is still consistent with the original data. For example, if one augmented the data by translation, (as is done in Mean Shifting), one could expect the Self-Consistency metric to be high. However, the relative positions of the augmented and original data could be changed enough such that the similarity values between is changed as well, which would be reflected in a lowered Pseudo-Consistency.

Next, Cross-Consistency is also defined in a similar manner as Self-Consistency but measures the change in similarities between the two modalities:

$$C_{\text{Cross}} = 1 - \frac{1}{N}||TA^T - \tilde{T}A^T||_F \tag{5.12}$$

Just like Self-Consistency, Cross-Consistency tracks if the relationships between clusters has been changed post-augmentation. If the similarities between data clusters remains unchanged, then Cross-Consistency will be greater than if the cross-cluster similarities were significantly altered. Importantly all of these metrics allow us to see how an augmentation strategy impacts the overall similarity graph without relying on supervised pairs as with Pair Alignment.

Next, I also measure three more Self, Pseudo, and Cross-representation alignment metrics based on Normalized Discounted Cumulative Gain (NDCG). In this work NDCG is defined as the ratio of measured discounted cumulative gain and the perfect or ideal discounted cumulative gain:

$$NDCG = \frac{DCG}{IDCG} \tag{5.13}$$

in which DCG is the discounted cumulative gain calculated with:

$$DCG_p = \sum_i \frac{S(\cdot,\cdot)}{\log(i+1)} \tag{5.14}$$

and $S(\cdot,\cdot)$ is the relevance score function. For each metric, Self-NDCG, Pseudo-NDCG, and Cross-NDCG the score function is the relative position or ranking of each point's neighbors, where the ranking is determined by the post-augmentation self-similarity graph, pre- and post-augmentation similarity graph, and pre- and

post-augmentation cross-cluster similarity graphs, as in the consistency metrics. These metrics also measure changes in the similarity graph, but they are affected most by *relative* change in rankings in the similarity graph, instead of actual similarity values. This means that if the order of the the nearest neighbors (within its own cluster, the text cluster or the audio cluster) for any given sample of data changes due to augmentation, these metrics will be lowered as a result.

Note that for large datasets, calculating full similarity graphs could be intractable, and it is common practice to calculate metrics like NDCG on smaller subsets of the full dataset, typically the top K scores. For this reason I calculate all Consistency and NDCG metrics using the top $K = 100$ scores. This means that for small datasets like the AudioCommons test data, almost all of the dataset is used in calculation, but for the large text datasets like WavCaps only a small fraction of the graph with the highest similarity is considered.

Finally, I select F_1-Score as my classification metric. Because my testing datasets are generally imbalanced and there is no particular cost to either false positives or false negatives, the F_1-Score is the metric of choice over other options like accuracy or weighted F-Score. To review, I track four metrics that measure the extent of the modality gap, six methods that measure change in the representational alignment, and one metric for measure the impact of augmentations on supervised data for a total of eleven metrics.

5.3 Experiments

Here I explain the classification experiment methodology, the models I use for classification, and the ablation study I perform.

5.3.1 Models

I select several classification models for comparison in this work. As with my augmentation methods, I restrict my model selection set to those with theoretical justification, common baselines, and extensions thereof.

I select my first classification model based on the theoretical guarantees provided by Zhang *et al.* [118]. Under the same conditions that guarantee C2 removes the modality gap, they prove that any regularized linear classifier minimizing a quadratic loss will be identical when trained on either modality. This means that under perfect conditions the model trained on text would *exactly match one trained on real audio*. These conditions are:

1. The modality gap is a constant vector
2. The modality gap is orthogonal to the span of embeddings from both modalities
3. Alignment noise can be approximated by a Gaussian distribution centered at the origin

The measurements taken by Zhang *et al.* to support these assumptions are found in Figure 5.2. I follow the same process described in [119] on the Clotho+

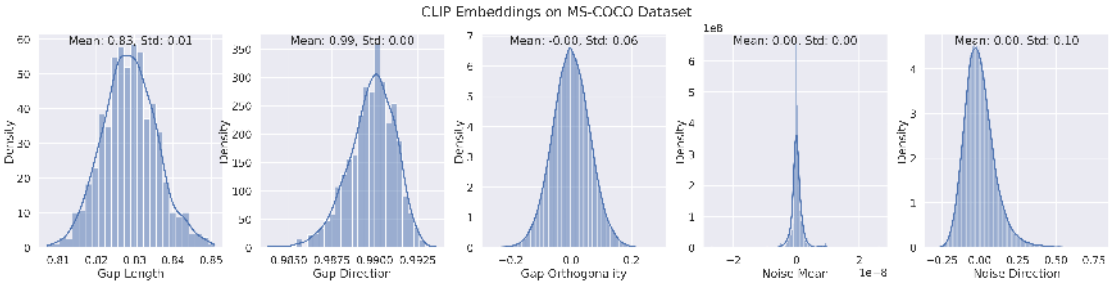


Figure 5.2: The empirical measurements of the conditions required for C2 to remove the modality gap taken from CLIP embeddings of the MS-COCO dataset. Image Credit [119].

dataset to estimate how well CLAP's embedding space follows the assumptions (see Figure 5.3), and one can clearly see the assumptions hold. This suggests that not only will C2 likely perform well as an augmentation, but that the model described by Zhang *et al.* would not suffer from the affect of the modality gap. Specifically, I train a L2-regularized one-layer linear neural network to minimize the mean squared error between class labels. This model matches the requirements set by Zhang *et al.*, and based on their theoretical results one can expect it to perform well in the classification experiments.

However, to expand on their results, I also compare three more models with the following variations:

1. A nonlinear, two-layer version of the neural network[2].

2. An identical linear model instead trained to minimize the cross entropy between classes.

3. A version of the nonlinear network that minimizes cross entropy.

[2]The architecture matches that of the Nonlinear augmentation model

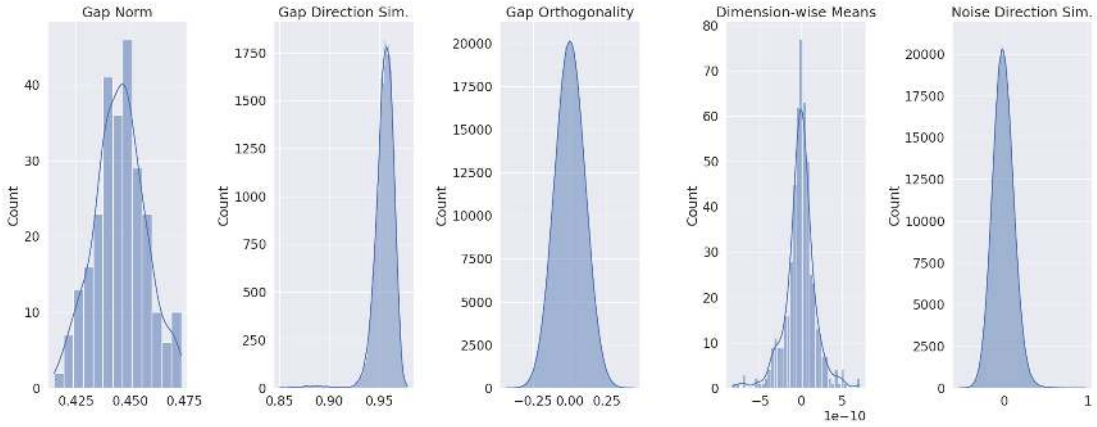


Figure 5.3: My empirical measurements of the C2 conditions using CLAP embeddings from the Clotho+ dataset. I find very similar results to those of Zhang *et al.* [119] indicating the C2 conditions are met.

These alternative formulations act to either extend or contrast with the model suggested by Zhang *et al.* They provide no theoretical analysis on either nonlinear model behavior, or models optimizing alternative loss functions. Therefore, I think it prudent to compare these reasonable variations on Zhang's relatively simple model, despite its theoretical justification.

As additional comparison, I utilize CLAP's zero-shot similarity scores between the test audio datasets and the mean embedding of the anchor prompts as a common baseline. CLAP and other CLIP-style models are commonly used in the literature as performance benchmarks. For each dataset, I calculate the mean of the seven negative class anchor prompt embeddings and select the larger CLAP scores between the positive and mean negative prompt embedding as the predicted class. In other words, I take the mean of the anchor prompts of the seven labels that do not correspond to the positive class for that experiment. Then, I mea-

sure the CLAP similarity score between the mean of the negative classes and the positive class embedding. Finally, I train directly on the small AudioCommons audio-timbre datasets. I perform 10-fold cross validation and report the average performance. This helps orient the performance of ToT models against those produced by simply using small real audio datasets.

5.3.2 Experimental Design

For each combination of training dataset, data augmentation strategy, classification model, and target dataset I perform the following experiment:

1. Randomly sample a balanced training dataset by undersampling to equal size.
2. Augment the data with the selected augmentation strategy.
3. Train the classification model, or perform a gradient descent step.
4. Calculate the eleven modality gap and representational metrics as applicable.
5. Determine the resulting F_1-score on the target audio dataset.

First, recall that there are eight total classes but I perform a series of binary classifications. This means that for any given class, such as "Bright", there seven other class's negative examples to train with. Usually then, there are vastly more negative examples than positive ones. However, as I have shown in Chapter 4, the class-wise distributions may vary drastically depending on the method used to

acquire the training data. To mitigate these differences, I randomly undersample data from both classes to an equal number of examples—10000—regardless of data source. This also helps mitigate potential bias from the differently sized training datasets.

Second, I augment the data according to the augmentation strategy. Usually this process is entirely completed at this stage, but C2 requires special treatment. Since C2 requires altering the testing data as well, I additionally modify the incoming test data just before model inference. However, for noisy augmentations I add noise during training, but not during testing.

Third, I calculate the modality gap and representation metrics discussed above. Pair Alignment requires paired multi-modal data, and is therefore not measured between un-paired training and test sets.

Next I train my model. As my models are trained iteratively by gradient descent, I loop the previous steps of augmenting data and updating weights until training is complete. I train for 100 epochs, and do not perform early stopping.

Finally, I record the F_1-score the classification model receives on the target audio-timbre dataset. I run every experiment 10 times and report the mean and standard deviation of test F_1-scores and modality gap and representational similarity metrics.

Chapter 6: Modality Gap Augmentations

In this chapter I analyze how the augmentation strategies affect each metric, and interact with each text and audio source. With the exception of Pair Alignment, which requires supervised data, I measure each metric in a similar fashion to the overall classification case study: for every combination of text and audio data, I perform an augmentation and measure each metric. For noisy metrics, I collect 10 samples. Note that because I measure the entire dataset, I only take one measurement for each clean data augmentation.

6.1 Representational Alignment

First I discuss the representational alignment metrics, and take advantage of the paired-data exclusive Pair Alignment metric on the Clotho+ dataset. The representational metrics seek measure how the augmentations impact the spatio-semantic assumption of the CLAP embedding space: by altering how the embeddings are aligned in space, a practitioner may reduce or entirely remove the distance-as-meaning property of the CLAP embeddings. Figure 6.1 shows the Pair Alignment of both noisy and clean augmentations. We see that adding noise lowers the Pair Alignment by ≈ 0.075 when compared to clean augmentations. This is unsurprising as Pair Alignment measures the normalized average cosine similarity

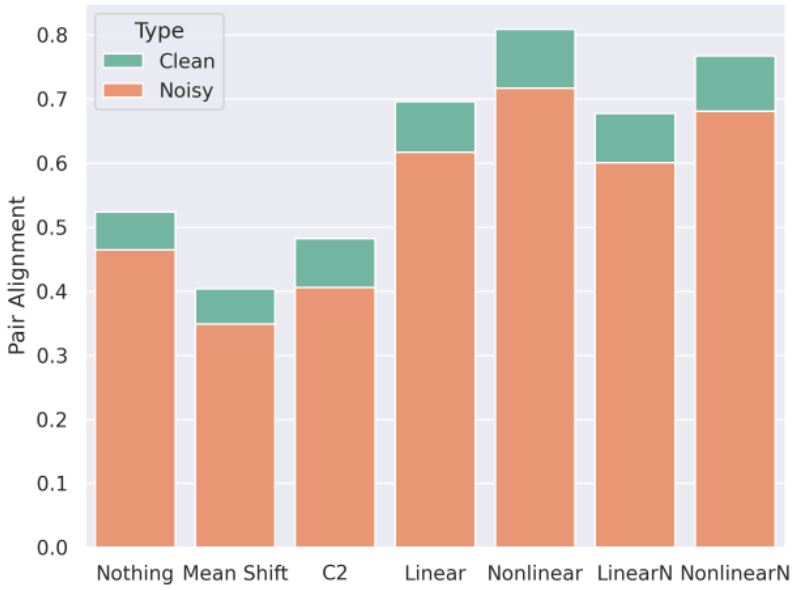


Figure 6.1: A comparison of the impact clean and noisy versions of each augmentation method have on the pair alignment of Clotho+. The standard deviations are negligible and not displayed.

of paired data. When applying additional random noise to the augmentation, the relative directions of the data will naturally be changed somewhat. This implies that anytime a noisy method is selected, a practitioner is choosing to somewhat diminish the fidelity of the representation. Therefore, the noisy methods may not be suitable for downstream tasks that require maintaining a consistent semantic structure, such as information retrieval.

Figure 6.1 also reveals a split in behavior by augmentation type. The unsupervised methods—Mean Shift and C2—both diminish Pair Alignment, while the supervised methods increase it. The supervised methods were explicitly trained to minimize the Euclidean distance between the paired data, and because cosine

similarity is directly proportional to Euclidean distance for normalized vectors, by extension the supervised metrics also optimized for Pair Alignment. Conversely, the unsupervised methods are not influenced by the presence of supervised paired data, and seemingly disrupt the alignment between known pairs.

Table 6.1 explores Pair Alignment's correlations with the other distance-based and representational alignment metrics on the Clotho+ dataset. Almost all metrics are either uncorrelated or negatively correlated with Pair Alignment. Unsurprisingly, higher distances between modality data tends to worse Pair Alignment, which is itself a distance based metric. The Approximate Hausdorff distance reveals itself to be nearly perfectly (up to rounding) negatively correlated with Pair Alignment. Given these metrics are calculated on supervised data, the farthest possible cross-modal neighbor implies some paired-data that is also at least that far apart, meaning as the maximum distance increases, the alignment of the paired data decreases. The representation methods show a different story, as the Consistency metrics contain the most positive correlations with Pair Alignment, while the NDCG metrics contain some of the most negative. The Consistency metrics show that little to no change in neighbor distances is either unrelated to or slightly encouraging of higher Pair Alignment. However, the NDCG metrics show that little or no change in the relative ordering of neighborhoods suggests lower Pair Alignment. This is likely because the original order of neighbors is affected by the modality gap. Once paired embeddings are better aligned (high Pair Alignment), then the order of the neighborhoods may have changed dramatically (low NDCG), even if the actual distances between them was only marginally changed

Metric	Lin. Sep.	Mean Dist.	Hausdorff	SWD
Correlation	0.05	-0.56	-1.0	-0.66

Metric	Self-C	Pseudo-C	Cross-C	Self-NDCG	Pseudo-NDCG	Cross-NDCG
Correlation	0.54	-0.04	0.23	-0.87	-0.84	-0.60

Table 6.1: The Pearson's r-correlation coefficients between Pair Alignment and distance metrics (Top) and representation metrics (Bottom). Most are either uncorrelated or negatively correlated, indicating pair alignment is harmed by higher distances between clusters and rigid augmentations that do not allow for the supervised pairs to become closer.

(high Consistency).

Next, I look at the three variants of NDCG and Consistency, Self, Pseudo, and Cross. Recall that these metrics measure the relative and real changes in neighborhood orders and distances, or lack thereof, between the original, augmented, and audio embeddings. Counter to Pair Alignment, Table 6.2 shows that the noisy augmentations have very little impact on the other six representational metrics. Not only do we see that adding noise barely changes these metrics, we see that only Cross-Consistency increases on average. Unlike the Pair Alignment results, this shows that adding noise does not significantly change the neighborhoods around each datum on average. However, these metrics are unsupervised, meaning that they treat every neighbor as important, even if in reality they are irrelevant. Therefore I still conclude that adding noise may not be appropriate for alignment-sensitive tasks such as information retrieval, because if the overall neighborhood is mostly unchanged, but the most important data is at risk of being misaligned, there is little benefit. However, the same argument can be made in favor of adding noise as a cheap data augmentation technique. Since it has a rel-

Metric	Self-C	Pseudo-C	Cross-C	Self-NDCG	Pseudo-NDCG	Cross-NDCG
Paired	-1.01	-0.77	0.20	-0.20	-0.07	-0.06
Unpaired	-0.65	-0.49	0.03	-0.19	-0.07	-0.76

Table 6.2: The average percent change in using noisy or clean augmentation methods. The Paired row represents the Clotho+ dataset's values, while the Unpaired row consists of all other combinations of text and audio datasets which do not contain paired data. In both cases, adding noise has almost no effect.

atively small impact, a practitioner may be more interested in adding noise than either another more expensive method or one that drastically changes the alignment graph. On the other hand, augmentations that do significantly change the neighborhood structure of the data are not necessarily a bad thing. For example, in their landmark study, Liang *et al*. find that for zero-shot image classification, in almost all cases, not changing or *increasing* the modality gap led to superior accuracy [53]. If the downstream task is not sensitive to perturbations in the neighbors graph, as mine is, then a practitioner would likely choose a more powerful but disruptive method, such as a supervised augmentation method.

On the same theme, I find that choosing to perform any augmentation, either with or without adding noise, has a small impact on these metrics. The values of all metrics but Pseudo-Consistency and Cross-NDCG changed on average less than 5% when any augmentation was performed. Figure 6.2 shows the two exceptions. In both cases, the supervised augmentations cause the greatest changes. This is likely due to some form of overfitting or bias. Recall that the supervised methods I use are pre-trained on the Clotho+ dataset, and therefore are biased towards its distribution. If a new set of text or audio data is particularly noisy relative to

Clotho+ or out of its distribution, then the supervised augmentations are likely to dramatically alter the representational alignment of the data. A large change in Pseudo-Consistency but no other consistency metrics indicates that the augmented text embeddings are moved relative to the original text embedding space, but remain consistent among themselves, and with the audio embeddings. The former is the desired behavior from an augmentation: my goal is to move embeddings away from text and towards audio. Additionally, it seems that the lack of change in the Cross-Consistency metric implies the supervised augmentations do not dramatically change the distances between neighbors when doing so.

However, when looking at the Cross-NDCG distributions, it is clear that the supervised metrics are capable of significantly altering the order of the nearest neighbors between the augmented embeddings and the audio embeddings. This means that even if the distances themselves were only changed slightly, the changes shuffled the order of the nearest neighbors. This would indicate the fidelity of the alignment may have been reduced. A change in Cross-NDCG but no other NDCG metrics may suggest that the augmentations move the embeddings in such as way as to maintain the order of the original text space neighborhoods, but disrupt the cross-modal neighbors graph.

The same is true to a lesser extent of the Mean Shifting method, as it is semi-supervised. I again rely on the Clotho+ audio data in my calculation of the gap vector used to shift the embeddings, and am again subject to its bias. This can be seen reflected in Figure 6.2 as Mean Shifting is the next most impactful approach after the supervised methods. However, this is less biased that the fully supervised

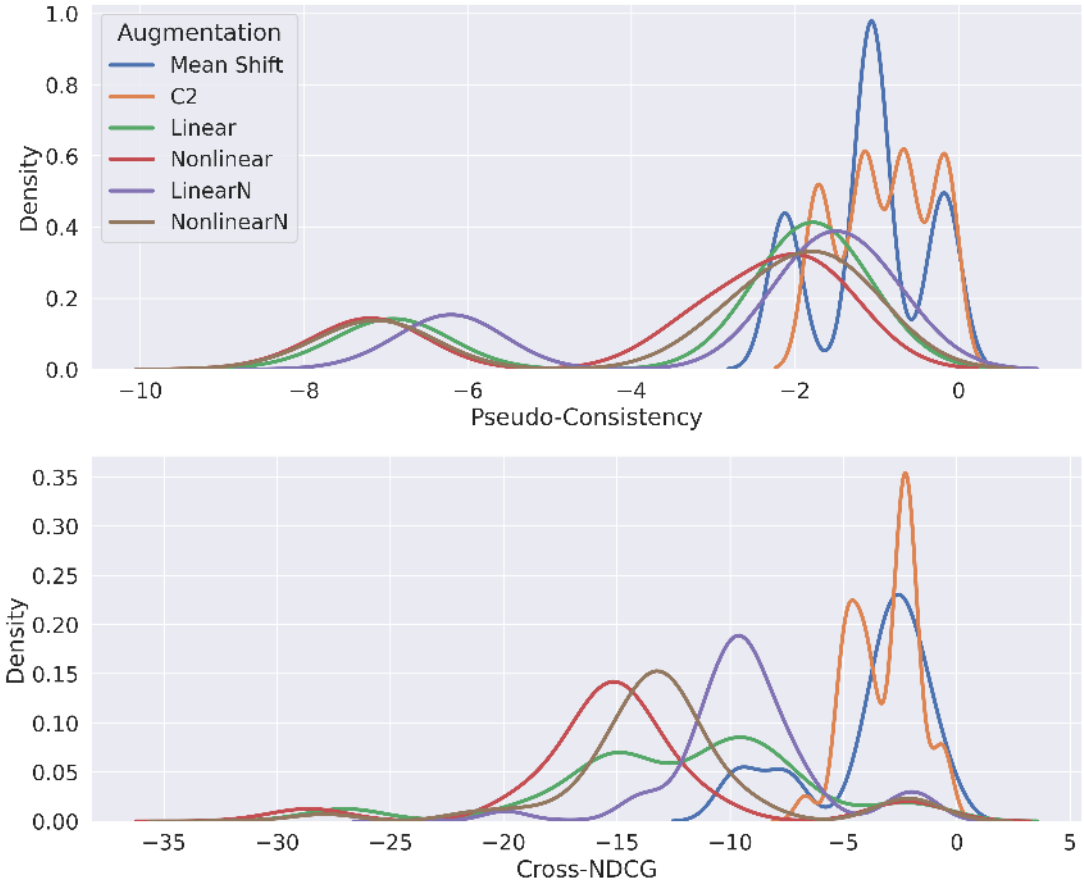


Figure 6.2: Distribution estimations of percent change when choosing to augment the data over performing no augmentation. Supervised methods tend to create the most impactful changes, followed by Mean Shifting. The other representational metrics not shown have less significant ranges of $< 5\%$.

methods, as it does not rely on any assumptions of shape or distribution beyond those already assumed by the CLAP embedding space. Namely, that embeddings near each other should be semantically similar, which implies that the text subspace is more or less the same shape as the audio space, and so moving a text embedding into its "matching" location in the audio space is a meaning-preserving action. In other words, Mean Shifting simply assumes the text data distribution should be approximately equal to that of the Clotho+ audio distribution, and places it in that area of space without altering the distribution in any other way[1]. If that assumption fails, then the method may be too biased.

Next, Figure 6.3 and Figure 6.4 show the text and audio dataset-wise Consistency and NDCG scores, respectively. Note that Self and Pseudo-Consistency do not depend on audio data, and are constant across this dimension. However, the audio-dependent Cross-Consistency scores still vary little across audio datasets for every text dataset. The Clotho+ audio data displays a slight deviation because it has slightly lower values compared to the other audio-text dataset pairs in each of its rows. This is also likely because the augmentations are biased towards that particular dataset, leading to a more significant change from the text embedding neighbor graph towards the audio embedding neighbor graph. Recall that Cross-Consistency measures the squared differences between the nearest-neighbors similarity graphs of the original text and audio embeddings and the altered text and audio embeddings. Therefore, if the embeddings are changed to match a

[1]Technically, because I re-normalize the data after augmentation, the distribution is altered. Mean Shifting along the manifold should be considered for future work.

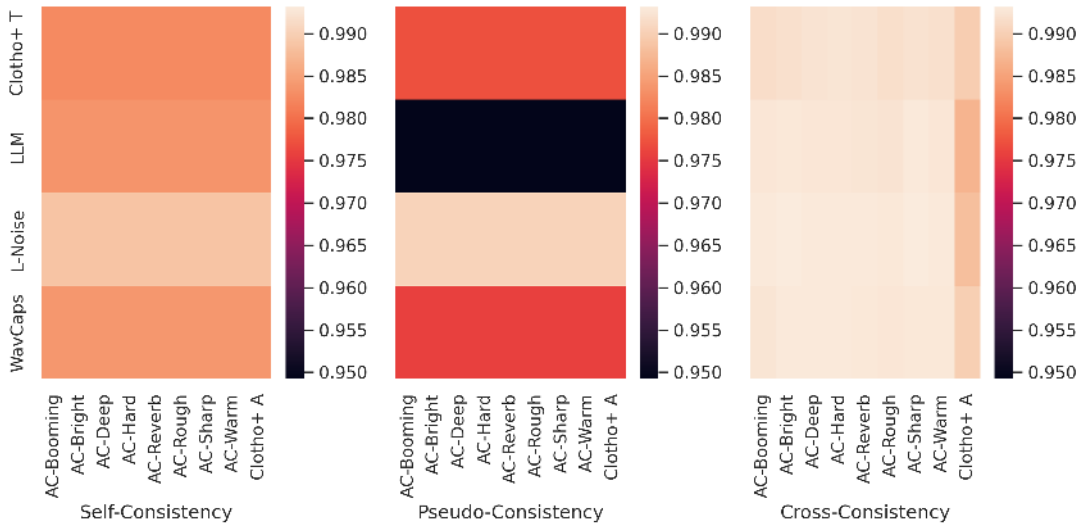


Figure 6.3: Consistency metrics split by text and audio datasets. Cross-Consistency shows slightly lower values on the Cloth+ Audio combinations. This is likely due to the bias of the augmentation methods towards the Clotho+ Audio distributions leading to more change. Self and Pseudo-Consistency are constant across the audio datasets as they only depend on text data.

new distribution—the Clotho+ audio distribution—the Cross-Consistency will decrease.

Moving along the y-axis, one can see that the text datasets have different values of Self-Consistency and Pseudo-Consistency, while Cross-Consistency generally remains consistent. The Label-Centered Noise dataset is the least affected across all three metrics, while the LLM Captions are one of the most, particularly for Pseudo-Consistency. WavCaps and the Clotho+ text embeddings fall in the middle, but also see a dip in Pseudo-Consistency. However, it is important to note that the theoretical range of values for all three Consistency metrics is $[0 - 1]$, while the measured range of these experiments remains greater than 94% of that

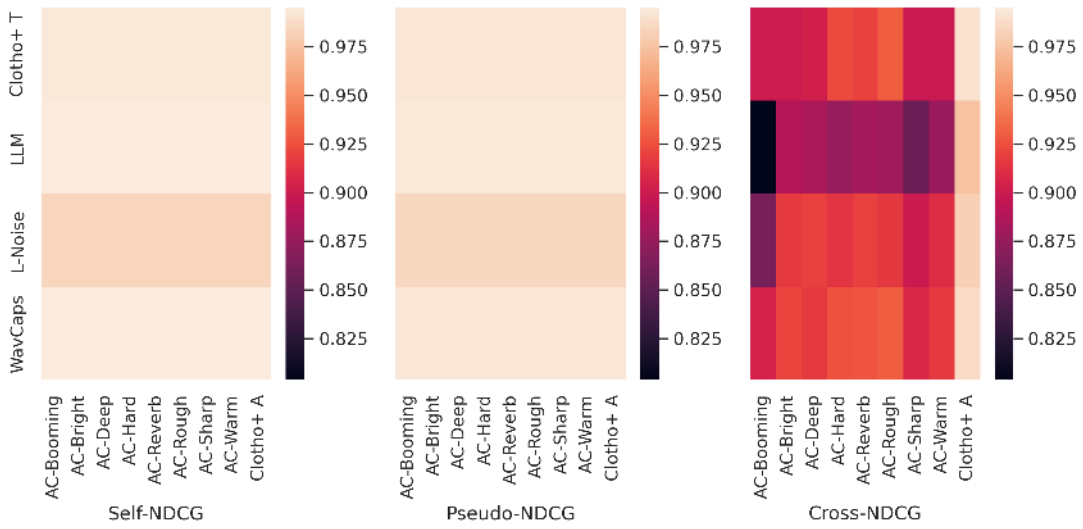


Figure 6.4: NDCG metrics split by text and audio datasets. Dataset combination has a strong effect on Cross-NDCG. This is likely due to the bias of the augmentation methods towards the Clotho+ Audio distributions leading to more change. Again, Self and Pseudo-NDCG are constant across audio datasets.

range, meaning these augmentations are only slightly modifying the distances in the similarity graphs. Continuing on to Figure 6.4, we see a different picture. Here, the Self-NDCG and Pseudo-NDCG are almost indistinguishable save for the Label-Centered Noise row, and the Cross-NDCG has unique values for each dataset combination. This shows that the relative ordering of the neighbors graph changed very little relative to the itself or text data, but changed more significantly with respect to the audio data. As above, the contrast with the Consistency metrics suggests that the order of neighbors changed but the overall distances between the neighbors remained mostly the same, for the audio case, and vice-versa with respect to the text embeddings.

Drilling further into these metrics, Figure 6.5 and Figure 6.6 show the interaction between text dataset and augmentation method on each Consistency and NDCG metric, respectively. Immediately one can see the unsupervised augmentations do not tend to alter the metrics as much as the supervised methods. This is consistent with the previous findings shown in Figure 6.2. More specifically, one can see that some text datasets are more affected than others. Notably, the Pseudo-Consistency of the LLM-Captions dataset augmented by supervised methods is lowered farther than any other combination of text data or augmentation. Because the supervised methods are trained to predict the Clotho+ audio embedding given the Clotho+ text embedding, this likely means that the distribution of the LLM data was more different from the Clotho+ text embedding distribution than either WavCaps or Label-Centered Noise. The Label-Centered Noise dataset also displays an interesting interaction with the methods. It has the great-

est difference between supervised and unsupervised NDCG scores. Meaning that of all the text datasets, the Label-Centered Noise had the most change in neighbor order when supervised methods were applied. This provides a clue to the nature of the CLAP embedding space. By definition th Label-Centered Noise dataset is composed of a several clusters of spherical Gaussians centered at the various class label anchor prompts. However, the supervised methods were trained on the true or noisy Clotho+ text distributions. Because the orderings were changed relative to the Gaussian clusters, one can infer that Clotho+ text distribution does not nicely match that shape or is out of distribution. Otherwise, one would expect to see very little relative change in the nearest neighbors order as entire neighborhoods get moved to the same place. One can also see that the supervised augmentations trained on the noisy Clotho+ text data had higher Cross-NDCG than those trained on the clean Clotho+ data. Because these models were trained on partial Gaussian Noise, they were better able to retain the Gaussian neighborhoods. However, it remains to be seen if that is to the benefit or detriment of downstream classification.

6.2 Modality Gap Metrics

These metrics are designed to measure the extent to which the embedding spaces are separate. Figure 6.7 depicts the average Linear Separability by augmentation. There one can see that almost all augmentation methods maintain a very high degree of Separability, save C2. Indeed, all methods except C2 approach the

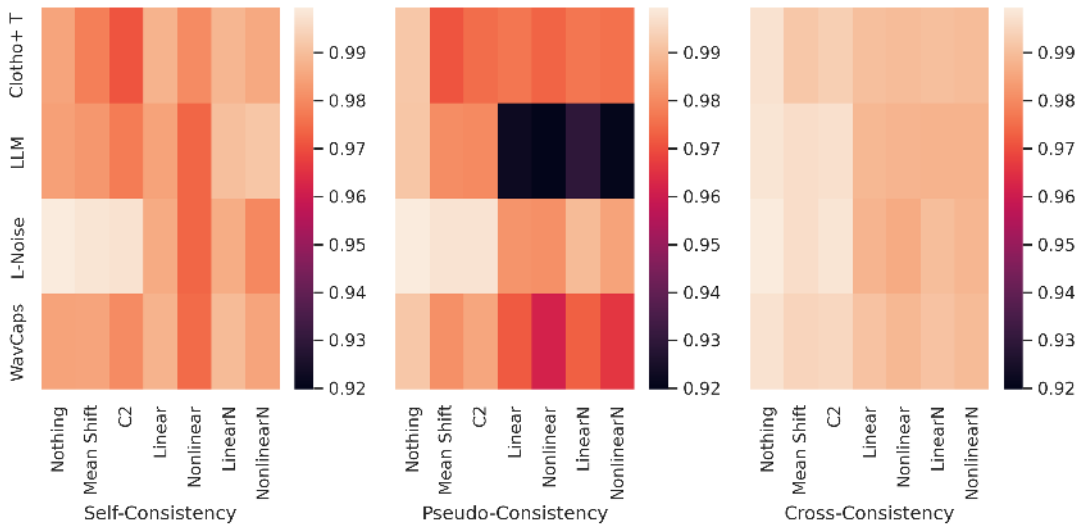


Figure 6.5: Consistency metrics split by text and augmentation. A noticeable split between supervised and unsupervised methods can be seen, especially in the LLM and Label-Centered Noise datasets. These likely indicate distributional differences between the Clotho+ dataset on which the supervised models were trained.

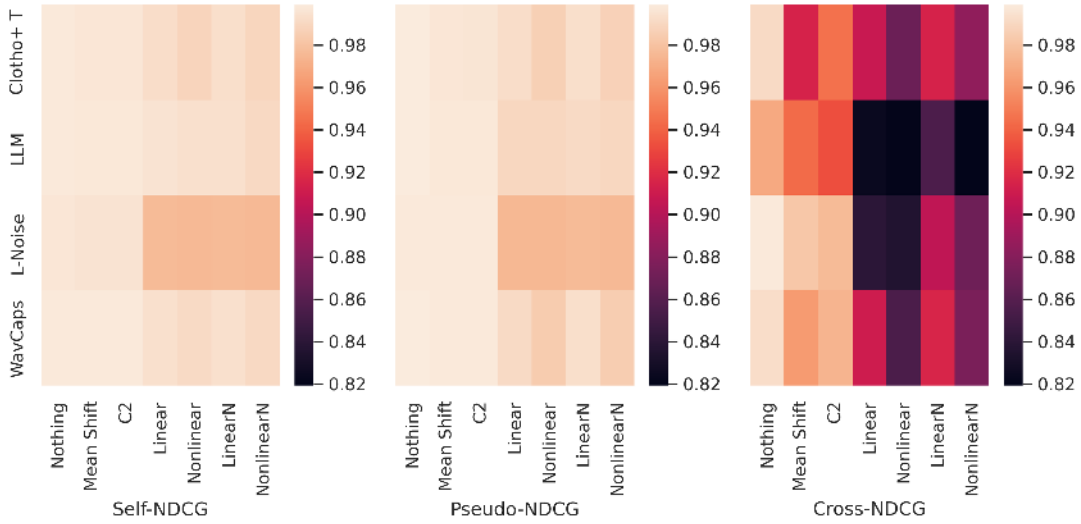


Figure 6.6: NDCG metrics split by text and augmentation. The same split between supervised and unsupervised methods is present here. Moreover, one can see the effect of training the supervised methods with added noise in the Cross-NDCG (Right). The added noise reduces neighborhood change in the cross-modal setting when compared with supervised methods without added noise.

maximum of perfectly separate. This is especially true of applying simple Gaussian Noise or performing Mean Shifting. Clearly, these methods do not adequately merge text and audio embedding spaces. That being said, all of the supervised methods are only marginally better, with comparable performances just below entirely separate. The only exception is C2. This is not particularly surprising, as it centers both modalities at 0, and my problem matches the assumptions Zhang *et al.* [119] made when designing the technique (see Section 5.3.1).

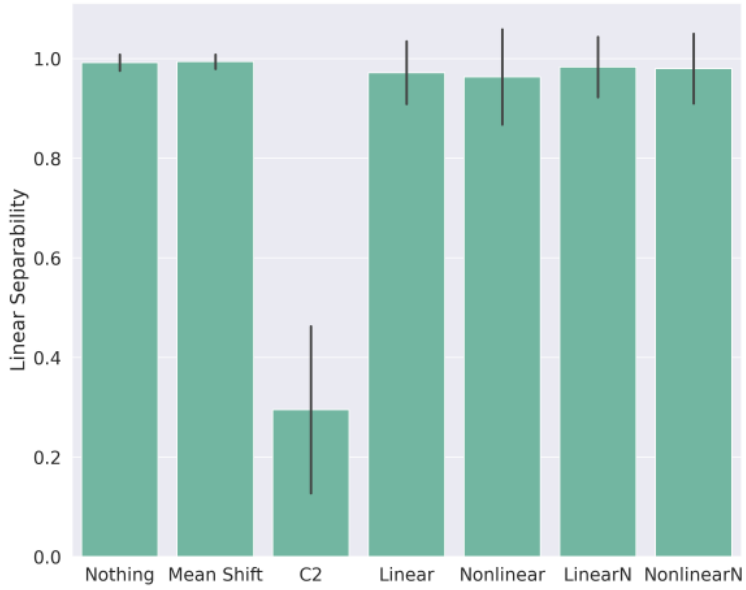


Figure 6.7: Mean and standard deviations of Linear Separability for each augmentation method. I average across all text and audio datasets, and over each of the ten noisy training runs. Almost all methods–except C2—fail to merge the two subspaces such that a linear classifier is unable to nearly perfectly separate them.

Next, in Figure 6.8, I compare the augmentation's affect on the Mean L2-Distance between each modality centroid. Similar to the Linear Separability re-

sults, most of the methods are interchangeable on average. Again, C2 outperforms the other augmentations as the best approach to reduce this metric. Once again, this is an expected result based on the definition of the method: if both modalities are centered at 0, then distance between them should be 0 as well. Perhaps more surprising, is that Mean Shifting did not particularly stand out. While it is second best at reducing the Mean L2-Distance, it is still mostly interchangeable with the other methods. While on paper Mean Shifting minimizes this metric, it only does so for supervised problems. Because Mean Shifting requires an estimate of the target modality's centroid, if the estimate is noisy, incorrectly biased, or out-of-distribution, then Mean Shifting no longer eliminates the gap. In practice, a Text-only Training practitioner would be to performing this augmentation for explicitly unsupervised problems, and the best they can do is use a biased estimation from an alternate data source.

Next, the results using the two point cloud metrics are shown in Figure 6.9 and Figure 6.10. First, one can see that all of the methods are comparable at reducing Approximate Hausdorff Distance, but notably the supervised methods are slightly better than the unsupervised methods. This could be because the unsupervised methods mostly leave the distributions in tact, meaning outliers remain far from the mean. Conversely, the supervised models, especially the nonlinear versions, may be able to better condense the point clouds. Similarly, the supervised methods achieve most of the lowest median Sliced Wasserstein Distances, although Gaussian Noise and Mean Shifting are also return similar results. C2 is a significant outlier in this metric, as it has more than twice the mean distance than the next highest

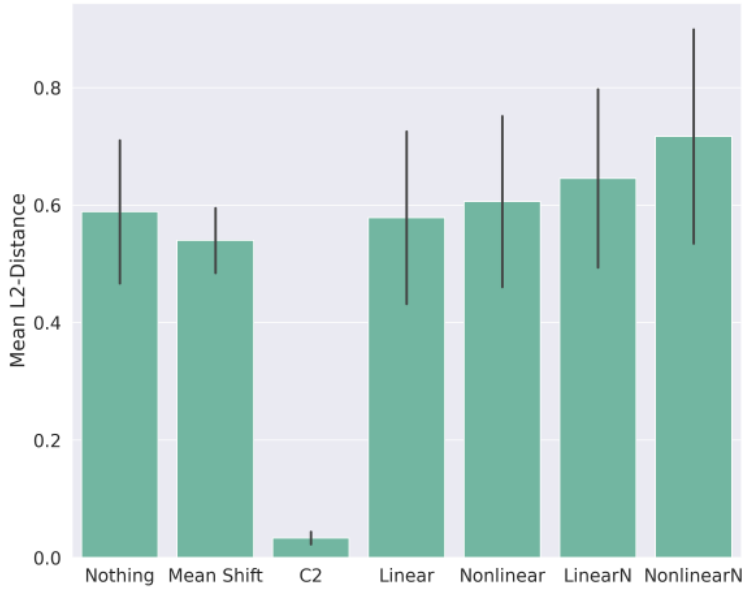


Figure 6.8: Mean and standard deviations of Mean Distance for each augmentation method. C2 is once again surpasses other options. Mean Shifting does next best, but suffers from a biased construction: it will always shift towards the Clotho+ Audio distribution, meaning if the test audio follows a different distribution, the gap persists.

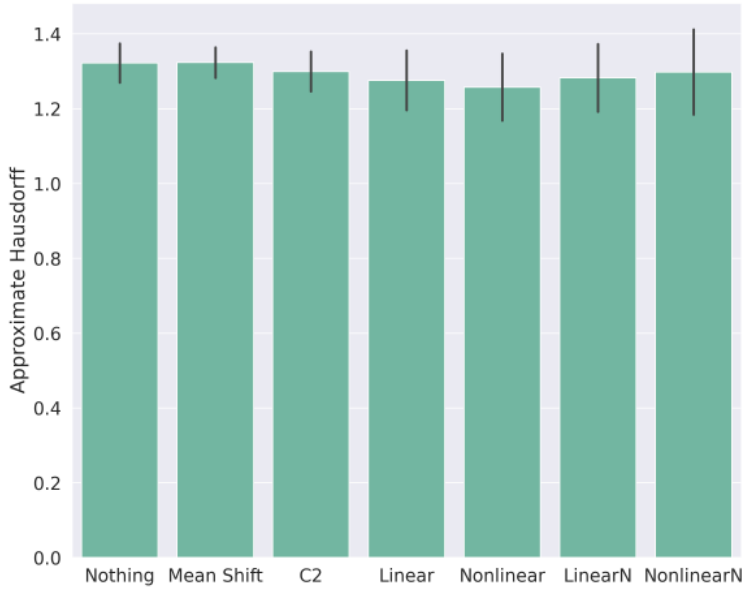


Figure 6.9: Mean and standard deviations of Approximate Hausdorff for each augmentation method. The methods are mostly interchangeable, with the supervised methods lowering the distance slightly more than the unsupervised.

scorer, and it has a very large spread of scores. However, looking at the median, one can see it is actually one of the best-performing augmentations. This is due to the interaction C2 has with the Sliced Wasserstein Distance and the Label-Centered Noise text dataset.

Figure 6.11 shows this interaction clearly: the rightmost heatmap indicates that the Label-Centered Noise text dataset has a high average Sliced-Wasserstein Distance to each audio dataset, while the other three sets have much lower values. C2 inflates the average, because it is designed to match the text and audio distributions together, assuming there are shared statistics and behaviors between them [119]. However, the Label-Centered Noise is explicitly a Gaussian distribution,

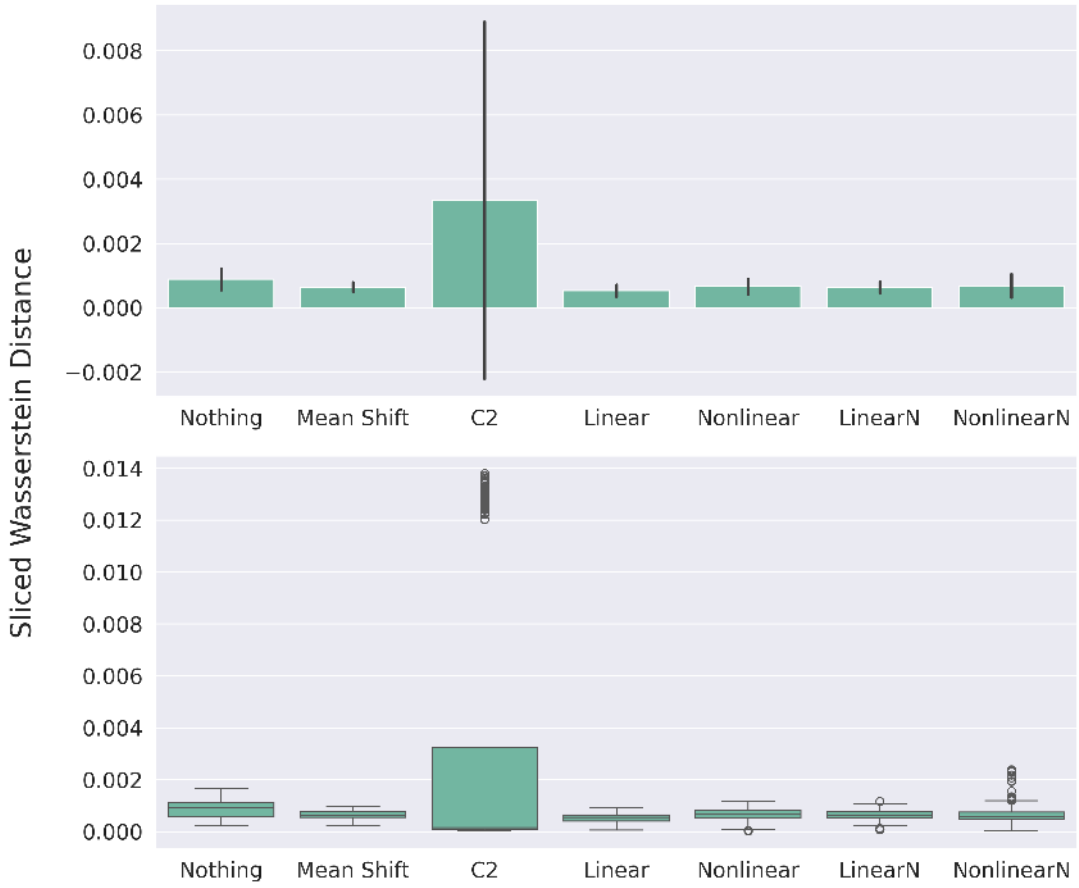


Figure 6.10: **Top:** Mean and standard deviation of the Sliced Wasserstein Distance for each augmentation.
Bottom: A boxplot of the SWD. This reveals the mean and median SWD values for each metric besides C2 are approximately equal, share a small spread of values, and are relatively comparable. Additionally, it shows that the median of the C2 method is the lowest of all the methods, and that the distribution is bimodal.

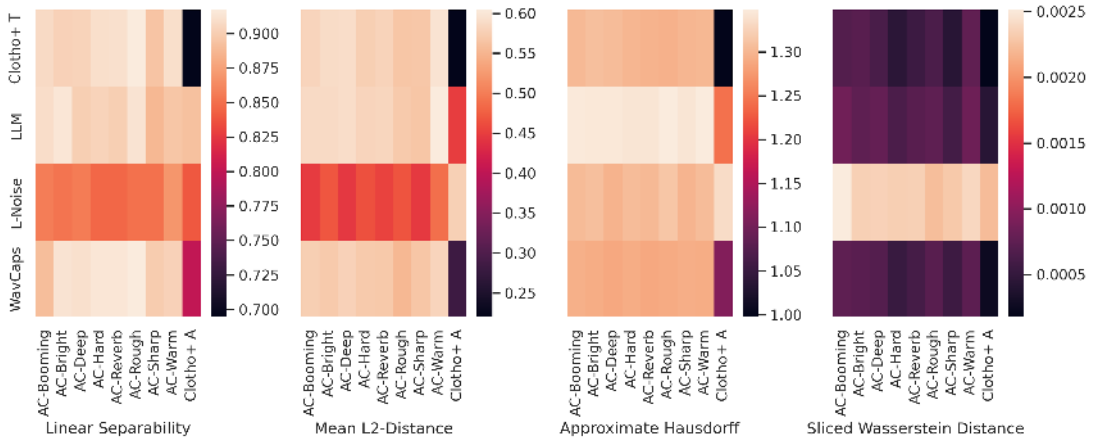


Figure 6.11: Mean Gap metric heatmaps organized by text and audio datasets.

which breaks the assumptions required for C2 to function correctly. The rest of Figure 6.11 shows the Label-Centered Noise dataset generally being closer to the audio datasets than the other datasets, with the exception of the Clotho+ text and audio sets, which once again exhibit biased behavior.

As shown in Table 6.3, I investigate the extent to which adding noise affects the outcomes associated with each method. Specifically, it shows the mean percent change in the metric value between noisy and clean versions of the augmentations. Generally, adding noise in addition to performing the augmentation improves performance, with several exceptions. The Approximate Hausdorff distance increases when including noise, which could be caused by the introduction of additional outliers far from the opposing cluster. Both Linear Separability and L2-Distance see a large increase when adding noise with C2, but the actual values C2 acheives on these metrics are so low, even a small increase leads to large relative change. Finally, the Sliced Wasserstein Distance of C2 also increases when Gaussian noise

Method	Hausdorff	Separability	L2-Distance	Wasserstein
Nothing	0.5 ± 0.9	0.0 ± 1.8	-4.4 ± 2.5	-8.7 ± 5.7
Mean Shift	0.6 ± 0.7	0.0 ± 1.4	-4.3 ± 2.5	-8.3 ± 5.9
C2	1.0 ± 1.2	13.2 ± 95.0	N/A	34.6 ± 32.2
Linear	1.2 ± 1.6	-1.3 ± 6.1	-6.1 ± 5.0	-9.3 ± 8.0
LinearN	1.0 ± 1.7	-1.0 ± 4.0	-9.0 ± 3.6	-17.2 ± 7.2
Nonlinear	1.6 ± 2.3	-0.6 ± 3.6	-7.6 ± 5.1	-10.9 ± 6.5
NonlinearN	0.7 ± 2.4	-1.2 ± 5.7	-9.7 ± 4.6	-22.8 ± 17.7

Table 6.3: The mean and standard deviation of percent increase in metric values across all datasets and experiments. Note that C2 is omitted from the L2-Distance columns as a percent increase from a value of 0 is undefined.

is added. While C2's extension C3 [119] is intended to have Gaussian noise added, this could be an indicator that the Gaussian noise is overwhelming the signal. Overall, these gap metrics are much more sensitive to adding noise than the representation metrics, and adding noise is a much more promising approach.

Lastly, I measure the Pearson correlation coefficients for each gap and representation alignment metric, as well as the downstream classification F_1-scores. As seen in Figure 6.12, the metrics display a wide range of correlations both negative and positive. For the most part, the alignment and gap metrics appear uncorrelated across metric types, but show more correlation within metric types. Cross-NDCG and Cross-Consistency are mildly correlated with each other and the Pseudo versions of each metric, while the Self variants appear uncorrelated with almost every other metric. The notable exception is that Self-NDCG and Pseudo-NDCG are almost perfectly correlated. This can also be seen in Figure 6.4 in which they produce nearly identical heatmaps. Next, we see that L2-Distance is highly correlated with Linear Separability. Naturally, data clusters that lie far from each other in

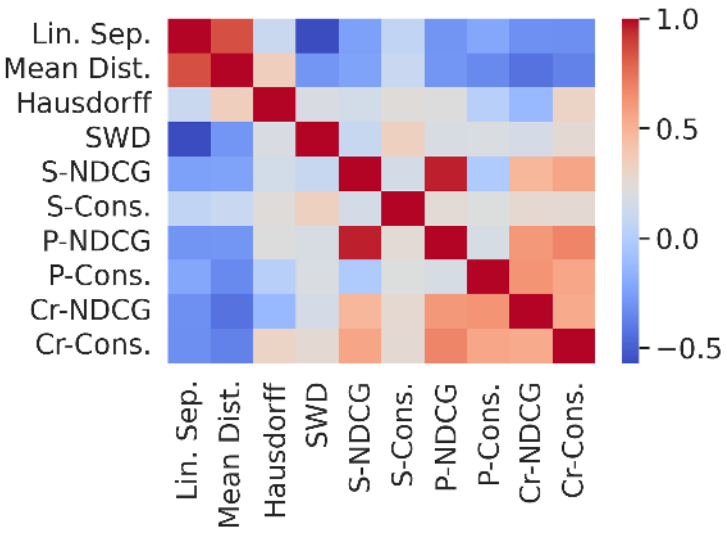


Figure 6.12: A Pearson's r-correlation matrix heatmap for both distance and alignment metrics. Linear Separability and Mean Distance are both correlated, along with Self- and Pseudo-NDCG. There are mild correlations between the other representational metrics, and a negative correlation between Sliced Wasserstein Distance and Linear Separability.

space are likely to have less overlap. A similar explanation could be posited for the mild correlation between Approximate Hausdorff and L2-Distance: if the clusters are farther apart, then their extrema are likely farther apart as well. Finally, we see that Sliced Wasserstein Distance has a moderate negative correlation with Linear Separability, meaning that as the Sliced Wasserstein Distance decreases, the Linear Separability increases, and vice-versa. Because Sliced Wasserstein Distance measures distributional similarity, one can see that the more similar two distributions are, the harder it would be to easily separate them.

Finally, one can see the correlation between the gap and alignment metrics

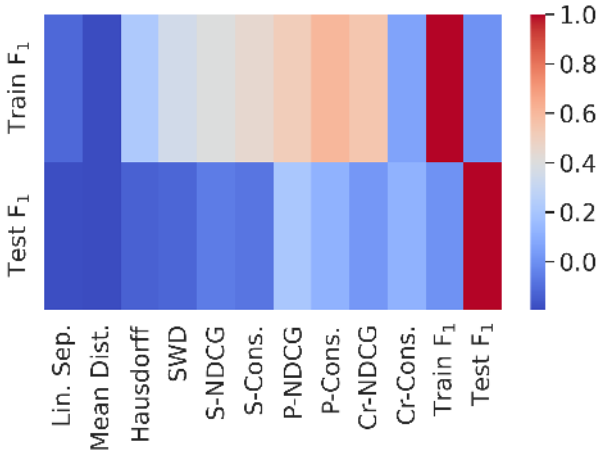


Figure 6.13: Heatmap of Pearson's r correlation coefficient matrix between all metrics and downstream classification F_1-scores. The Pseudo-NDCG, Cross-NDCG, and Pseudo-Consistency are moderately correlated with training F_1-score, while none of the metrics are strongly correlated with test F_1-score.

in Figure 6.13. None of the metrics are strongly correlated with the train or test F_1-scores in either direction. The only noteworthy moderate correlations are between the training scores and the Pseudo-Consistency, Cross-NDCG, and Cross-Consistency. This correlation implies that when the augmentation does not dramatically alter the relative or absolute positions of the augmented data with respect to either the text or the audio spaces, training on the text data will be easier. This could be because those augmentations that do alter the similarity graph structure more than others could also be creating a harder binary classification problem as well, while those augmentations that do not particularly affect the shape of the data may not increase difficulty beyond the baseline difficulty of the classification problem.

In summary, I showed that supervised augmentation methods improve the Pair Alignment metric, while unsupervised and noisy methods tend to decrease it. Additionally, some of the best negative correlates with Pair Alignment were Approximate Hausdorff and the NDCG metrics. These may be of use as a proxy estimate for Pair Alignment in future work on addressing the modality gap. Next, in the unsupervised setting I showed that all of the data augmentation techniques had little impact on the representational metrics, save for Pseudo-Consistency and Cross-NDCG. These two metrics were often lowered substantially by the supervised metrics, indicating change in distances between text and augmented neighbors, and the relative orderings of the nearest neighbors graphs between the augmented and audio embeddings. Continuing, I found that the supervised methods affect some text datasets more than others; when augmented with supervised methods, the LLM-Captions had the lowest Pseudo-Consistency, and the NDCG scores of the Label-Centered Noise dataset were lowered the most. These likely indicate greater distributional differences between them and the Clotho+ text distribution than the WavCaps dataset.

Moving on the modality gap metrics, I found that C2 was the best at reducing nearly all the metrics. Only the Approximate Hausdorff had slightly better results from the supervised methods, and C2 had a relatively high Sliced Wasserstein Distance from the Label-Centered Noise dataset, due to that dataset violating C2's assumptions. For both distance and representational metrics I found that adding noise generally does not have a significant impact on the values of any metric save Pair Alignment, and conclude that if a ToT practitioner's downstream

task is sensitive to Pair Alignment, such as information retrieval, they should consider avoiding added noise. Finally, most of the metrics were uncorrelated with themselves or the downstream classification F_1-scores with some intuitive exceptions.

Chapter 7: Timbre Classification Results

In this chapter I similarly examine the impact of training datasets and augmentation techniques, but on the binary F_1-score for each audio dataset. I also compare the behaviors of the four classification models. Then, I provide an ablation study on the undersampling methodology used in the classification experiments. Finally, I compare the performance of ToT models with CLAP zero-classification and training on real audio.

7.1 Augmentations

I again analyze the impact of each gap-closing method. First, Table 7.1 shows the average increase in score when using noisy versions of each method instead of the non-noisy version. Applying additional Gaussian noise increases the downstream classification F_1-score on the real audio test sets for every augmentation technique. The supervised methods have higher gain than the unsupervised methods, for which the gain is close to zero, however the amounts increased are mostly negligible. This is likely due to the noise acting as a moderately-successful data augmentation technique in the traditional sense. By adding more noisy samples to the training data, regardless of the method, the model is less sensitive to the real data distribution and is less likely to overfit. That being said, an explanation

Method	Nothing	Mean Shift	C2	Linear	LinearN	Nonlinear	NonlinearN
Train	-0.005	-0.008	0.002	0.017	0.041	0.022	0.022
Test	0.002	0.008	0.004	0.018	0.035	0.017	0.028

Table 7.1: The average change in F_1-score when applying noisy augmentations instead of the non-noisy versions. All methods improve when noise is applied. Supervised methods improve more than unsupervised methods.

for the differences in gain for supervised and unsupervised methods could be that because the supervised methods are already "overfit" to the Clotho+ audio distribution, adding noise is more beneficial to countering that bias that it is in the unsupervised cases.

Continuing on to the unsupervised methods, applying no augmentation generally harms performance. This can been seen in Figure 7.1 which shows the average F_1-score on the test audio data by each unsupervised method. The Deep and Reverb datasets' performance suffers relative to the other methods when no augmentation is applied. However, the opposite can be true, as is the case with the Hard dataset. Applying no augmentation and Mean Shifting exhibit approximately comparable performance for the rest of the audio datasets, and both exhibit a wide distribution of scores. Conversely, C2 emerges as the dominant method in all real audio test sets, and has the tightest distribution of scores. This indicates that it is the most consistent and effective of the unsupervised methods.

Next, I present the results of the supervised methods. In Figure 7.2, I compare the average and standard deviation of the F_1-score across methods and test sets. There is no clear best performer this time, and each of the supervised models performs well in some contexts. However, the less effective methods are, once

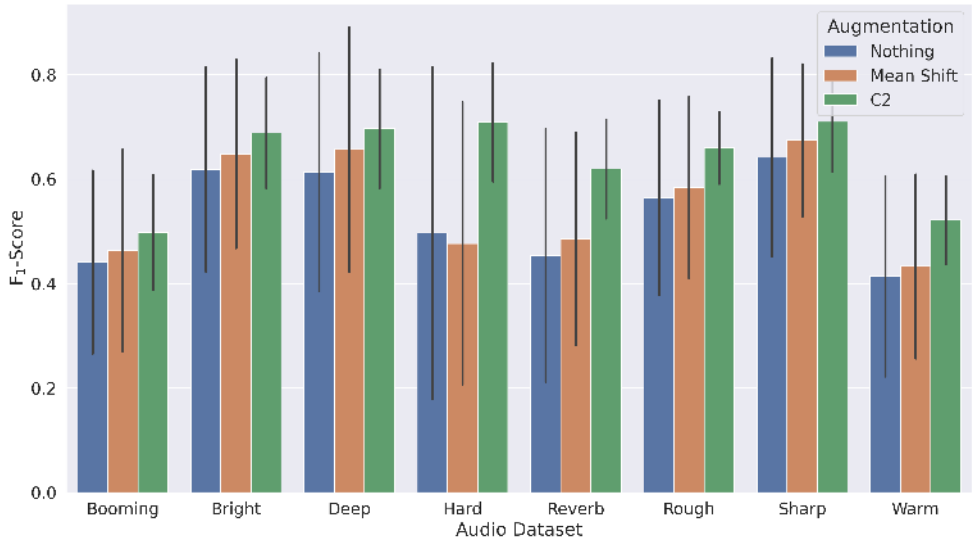


Figure 7.1: Mean test scores using unsupervised augmentation methods. Applying on augmentation typically leads to lower scores, with the exception of the Hard dataset. The C2 method outperforms the Mean Shifting method in every case.

again, those performing no data augmentation and the linear model trained with noise augmentation, LinearN. The other linear method and both nonlinear methods are competitive, but due to its lower variance and frequent high scores, I conclude that NonlinearN outperforms the other supervised methods.

I compare my baseline of no augmentation, the best unsupervised method, C2, and the best supervised method, NonlinearN, as shown in Figure 7.3. Overall, C2 outperforms NonlinearN on all test datasets, and has a superior overall score distribution, as can be seen in Figure 7.4. This empirically supports the theoretical results from Zhang *et al.* [118, 119] that guarantee the same classifier can be learned from either data modality using C2 under the conditions I verified in Section 5.3.1.

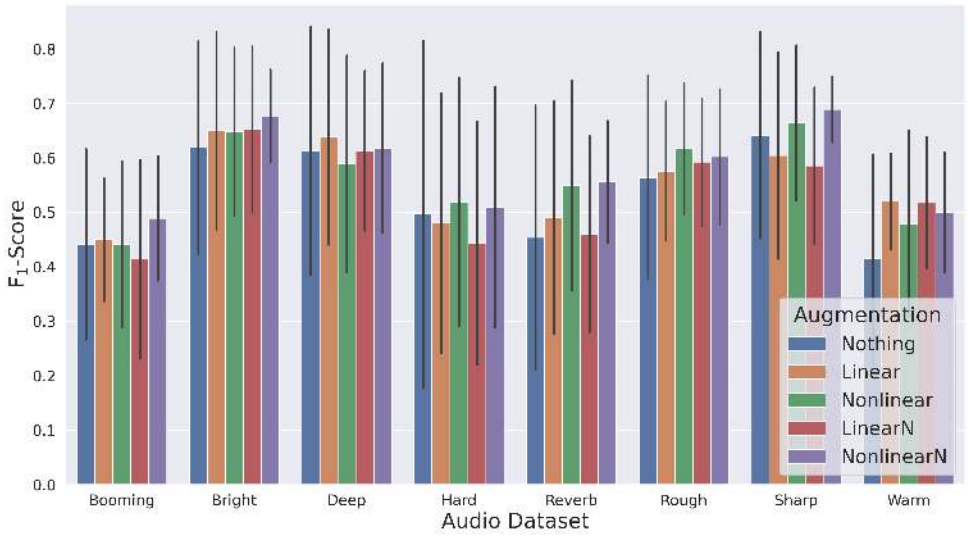


Figure 7.2: Mean test scores using supervised augmentation methods. The supervised methods are more varied in their scores than the unsupervised methods, but NonlinearN is the only method among them to never be worse on average than applying no augmentation technique.

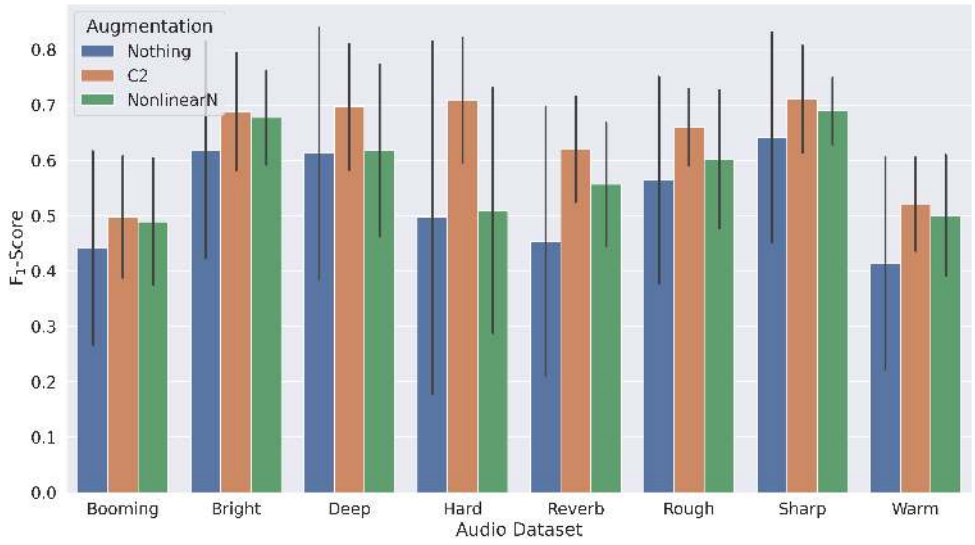


Figure 7.3: Comparison of the baseline, best unsupervised and supervised methods: Nothing, C2, and NonlinearN. C2 returns the best overall F_1-score.

7.2 Training Datasets

To explore the effectiveness of each text data creation method, I examine the differences in performance between each dataset. In Figure 7.5, one can see that the CLAP labeling of WavCaps and Label-Centered Noise tend to outperform both the LLM-generated captions and the SBERT labeling. Unsurprisingly, the two best methods are based around the CLAP embedding space itself. The Label-Centered Noise dataset assumes that the audio clips that have the described timbral characteristic are embedded in the CLAP space near the anchor prompt, while the WavCaps-CLAP dataset relies on the captions that display semantic similarity to the anchor prompts to be embedded nearby. The former approach relies on cross-modal audio-text similarity, while the latter relies on within-modality

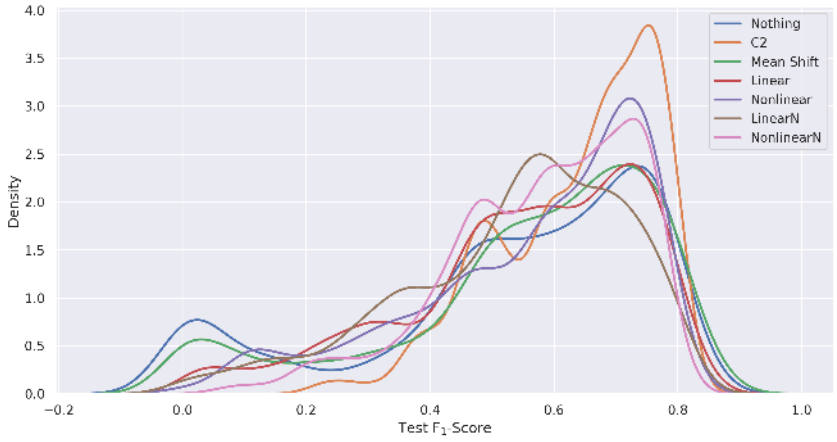


Figure 7.4: Distributions of test audio F_1-scores for each augmentation. C2 and the nonlinear methods have the best overall score curves.

text-text similarity. Viewed through that lens, the better performance on the Label-Centered Noise dataset could be explained by the effects of the modality gap itself, because within-modality similarities are on average higher than cross-modal similarities [53, 91]. This means that there is a greater chance of CLAP mislabeling data due to captions with similar meanings in the CLAP embedding space being confused with each other.

As shown in Figure 7.6, the LLM-generated dataset appears to perform particularly poorly in the audio modality, despite occasionally scoring highly during training. It exhibits both the widest score distribution, and the greatest deviation between train and test scores, indicating that its training examples were easily separable, but failed to emulate real audio data. This could be explained by the kind of language model I chose. By following existing work [114] and using a general language model for dataset creation, I sacrifice domain-specific knowledge

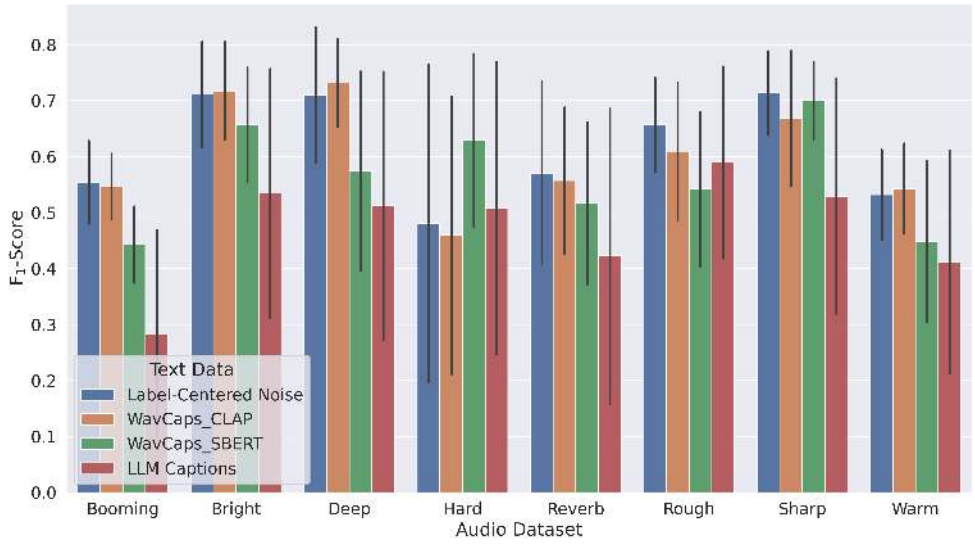


Figure 7.5: Mean and standard deviation of test scores averaged across data augmentation methods. Label-Centered Noise and WavCaps_CLAP are the best on average. Some exceptions include the Hard and Sharp audio datasets, in which WavCaps_SBERT does best or better than its CLAP version.

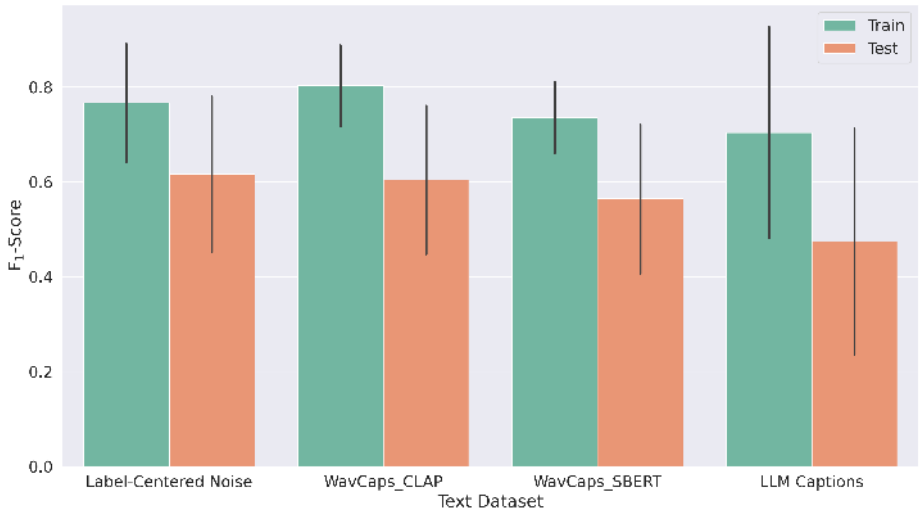


Figure 7.6: Mean and standard deviation of train and test scores across all augmentations and targets. The LLM Captions dataset has the greatest difference in train and test scores and the widest variance. Despite having some of the highest trainging scores, it has the lowest average test scores.

that might be available in a specialized LLM. The SBERT labeling exhibits the opposite behavior: a small distribution of train and test scores, and consistently second or third highest. Notably, the dataset's heavy bias (see Figure 4.1) towards the Booming class does not help its performance on that audio dataset. Additionally, since Booming is a minority class in the audio modality, one might expect SBERT's labeling to fail to achieve success on real audio in general. A breakdown by augmentation is provided in Figure 7.7, and continues to show C2 as the method of choice, followed by NonlinearN.

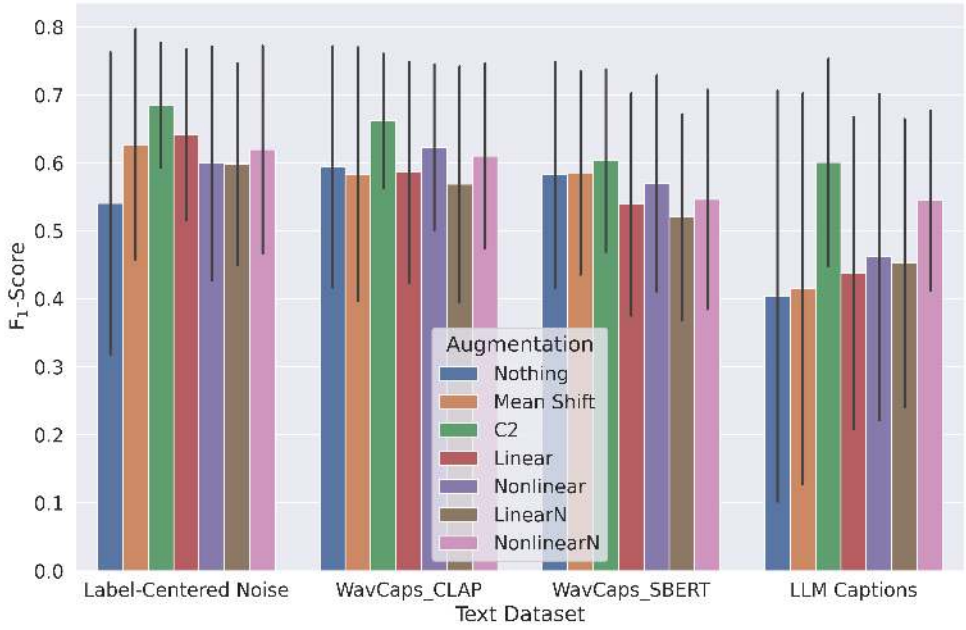


Figure 7.7: Mean and standard deviation of test scores across for each text dataset separated by augmentations. C2 remains both the highest-scoring method on average, but also has one of the most stable score distributions across datasets.

Models	**MLP1-CE**	**MLP1-MSE**	**MLP2-CE**	**MLP2-MSE**
Train F_1	0.775 ± 0.180	0.713 ± 0.048	0.795 ± 0.184	0.727 ± 0.077
Test F_1	0.482 ± 0.197	0.667 ± 0.057	0.458 ± 0.172	0.651 ± 0.073
Noise Gain	-0.003	0.018	0.007	0.042

Table 7.2: The average and standard deviation of F_1-scores per level and loss function, and the average change when trained with noisy augmentation. MLP1 refers to one-layer linear networks, while MLP2 refers to two-layer networks.

7.3 Model Performance

Lastly, I show how the classification model choice interacted with training data generation, augmentation methods, and test data. The first rows of Table 7.2 shows the average and standard deviation of the F_1-scores for each model type. Unsurprisingly, the linear model suggested by Zhang *et al.* [119] received the highest test scores with the tightest spread. The nonlinear version performed only marginally worse, but the two models minimizing cross entropy scored significantly lower. Interestingly, choosing a nonlinear model for either loss function led to a lower average test score. This is likely a sign of overfitting, which is further evidenced by the negative correlation between train and test scores. Despite having the lowest train scores, the linear model minimizing mean squared error (MLP1-MSE) returned the highest test scores.

The bottom row displays the average increase in test F_1-score when applying noisy augmentation. This shows that adding noise in addition to the base methods did not particularly help the cross entropy models, but did somewhat improve the mean squared error (MSE) models. The nonlinear MLP2-MSE model in particular

improved its F_1-scores by an average of 0.042. Again, I suspect the added noise is helping combat overfitting in both cases. In general, nonlinear models are high variance and require more data to overcome their sensitivity. By adding Gaussian noise in addition to the basic augmentations, I essentially create additional training data in each batch. The same is true for the linear model, but to a much lesser extent. The large difference in test scores between the two loss functions could also be caused by the cross entropy loss function matching the distribution of the training data better than MSE, and since the distribution is not exactly correct, the less-precise MSE models end up doing well in practice. Supporting this, the cross entropy models have a greater average train score than both the MSE models, but fail to generalize to the test sets.

Figure 7.8 and Figure 7.9 compare the train and test scores between audio test sets and text training sets, respectively. Most audio datasets follow the general trend of cross entropy models doing well in training but failing during testing, and vice versa for MSE. The Booming and Warm datasets were the most difficult for all models, and the Hard dataset was difficult for the cross entropy models, but not the MSE models. This could be because these datasets are some of the least balanced among the AudioCommons dataset, or because the labeling process is particularly noisy for these timbral characteristics (see Figure 4.1). Moving on to the text-wise results, most scores were relatively consistent between train and test; the average relative ranking of a text dataset is approximately the same when comparing train and test scores.

Finally, Figure 7.10 shows the average test scores for each model and augmen-

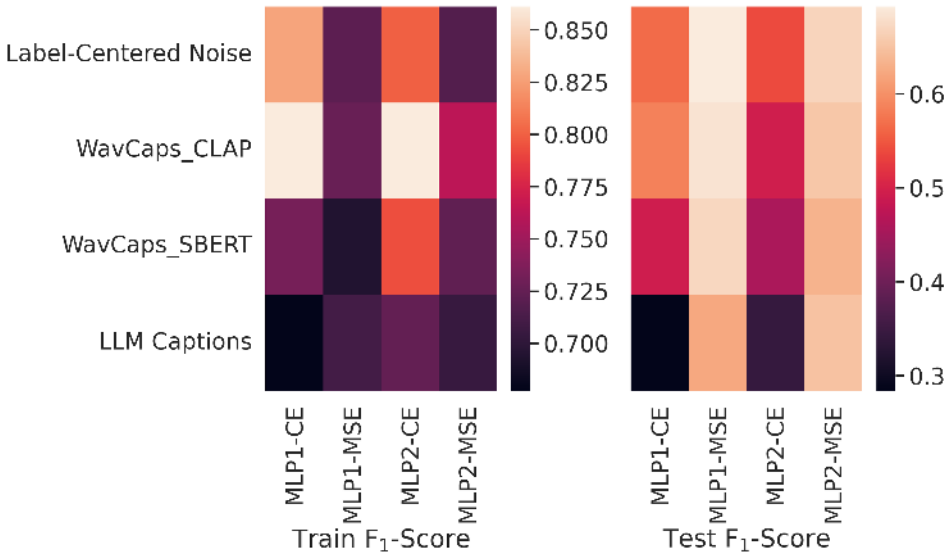


Figure 7.8: The average train and test scores for models and text training datasets. The text datasets are broadly consistent across model types.

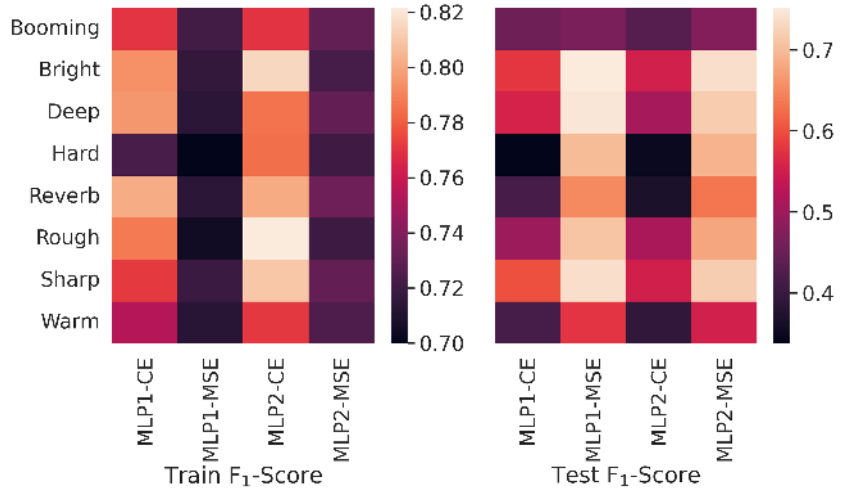


Figure 7.9: The average train and test F_1-scores for each model type and audio dataset. The Booming and Warm datasets challenged all model types, but the Hard dataset was difficult for only cross entropy models.

tation. Here we see that the baseline of no augmentation and the unsupervised methods led to the highest train scores for all four models, while the supervised methods led to much lower train scores. This could be because the supervised models encountered data outside their training distributions which made the problem more difficult than the otherwise simple unsupervised methods. Supporting this claim, the supervised models trained on noisy data led to a higher average train score than those without. On the test side, the unsupervised methods still led to the best scores, but the supervised models have reversed outcomes from the train scores: supervised methods trained with additional noise led to a worse test score. This could indicate that the noise added to Clotho+ dataset on which these models were trained does not match the alignment noise found in the real audio data.

7.4 Ablations

Before I discuss my final comparisons, I briefly explore the effects of the size of the dataset in an ablation study. In Figure 7.11 and Figure 7.12 I show additional experiments performed on the full and unbalanced datasets. Recall that each constructed dataset has approximately equal numbers of each class before binarization, which means each binary dataset has seven times the number of negative examples as positive examples. Additionally, the class distributions of the pseudo-labeled datasets WavCaps-CLAP and WavCaps-SBERT are dependent on the models' outputs and vary widely between them (see Figure 4.1).

I find that the average train F_1-score for all large unbalanced datasets is lower

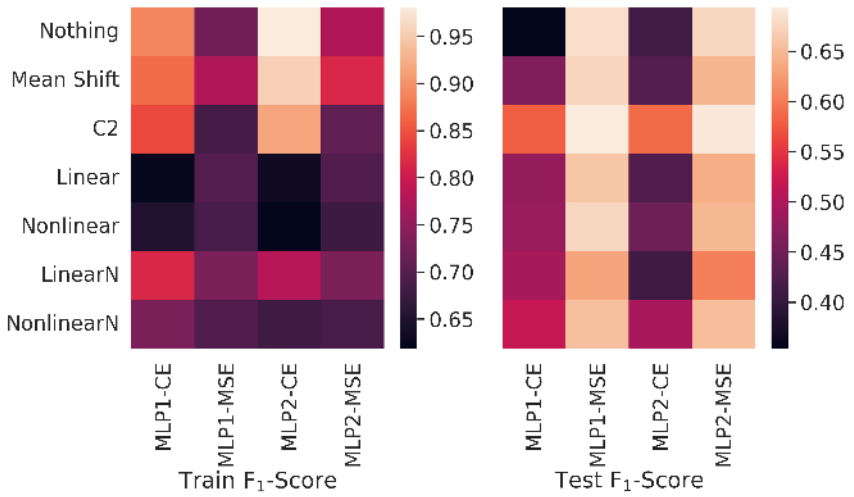


Figure 7.10: The average test scores for each model and augmentation. Unsupervised methods tend to lead to high train scores for all model types, but only the MSE models score highly on test data. Supervised models tend to score lower train scores for all models, but the cross entropy models tend to score higher on test data with supervised models than with unsupervised.



Figure 7.11: The average and standard deviation of train F_1-scores for each large and unbalanced text datasets compared with the sampled datasets used in the main experiments. The full datasets score lower and have a wider variance, despite consisting of significantly more data.

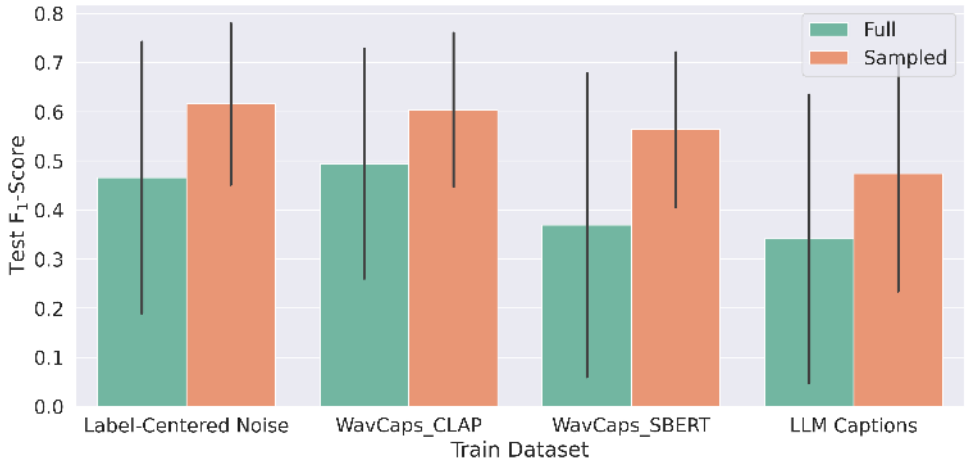


Figure 7.12: The average and standard deviation of test F_1-scores for each large and unbalanced text datasets compared with the sampled datasets used in the main experiments. The large and unbalanced datasets fail to overcome their biased label distributions and score lower than the sampled datasets. This indicates the undersampling method used in the main experiments was justified.

than than the same average scores of the balanced datasets. Additionally, the variance of the scores are much higher. The unbalanced datasets have a different ranking than the sampled versions: rather than Label-Centered Noise being the best and LLM Captions the worst, their positions have been swapped with the CLAP and SBERT labelings of WavCaps, respectively. This is intuitive, as Label-Centered Noise and WavCaps-CLAP had similar scores in the sampled experiments, and both have fairly even class distributions. Meanwhile, WavCaps-SBERT had an extremely biased label distribution, which caused it to perform poorly. As for test scores, one can see that again the full datasets return a lower performance on the downstream classification task. The score rankings return closer to that seen in the sampled datasets, but with WavCaps-CLAP performing slightly better than Label-Centered Noise. All in all, this justifies my undersampling methodology. and indicates that naïvely creating large datasets may not be effective for ToT, as the size does not overcome the biased label distributions or class imbalances.

7.5 Baseline Comparisons

Finally, Table 7.3 lists the average and best performing model scores for experiments with data augmentation. For comparison, I show CLAP zero-shot scores and the mean 10-fold cross-validation score from training on the AudioCommons audio datasets. While the average model scores are generally less than CLAP's zero-shot performance, one can see that the mean score is reduced by the low-

Dataset	Booming	Bright	Deep	Hard	Reverb	Rough	Sharp	Warm
Mean Model	0.46	0.66	0.64	0.52	0.53	0.61	0.65	0.50
Median Model	0.48	0.72	0.71	0.58	0.59	0.65	0.71	0.55
CLAP Zero-Shot	0.58	0.76	0.85	0.25	0.57	0.62	0.75	0.47
Best Model	**0.72**	**0.82**	**0.86**	**0.82**	**0.74**	**0.80**	**0.82**	**0.66**
Audio Datasets	0.72	0.79	0.89	0.85	0.82	0.81	0.78	0.77

Table 7.3: Test F_1 scores of the mean, median, and best models with data augmentation, CLAP zero-shot, and 10-fold cross validation on real audio. The best text-only results are in bold.

est outlier scores. The median score reveals an even tighter competition between the average ToT model and CLAP zero-shot, including superior performance on two additional datasets in *Reverb* and *Rough*. Additionally I show that the best performing model is always better than CLAP. Notably, for a majority of audio datasets, the best text-only models were within 0.03 F_1-score of the average model trained on real audio. More impressively, ToT meets or exceeds the performance of training on real audio on the *Booming*, *Bright*, and *Sharp* datasets, respectively. This is an improvement over my prior work, in which the average model was not always better than CLAP, and the best text-only models were only as good or better than two out of eight audio datasets [65]. However, the mean model scores decreased, and the median scores increase, indicating that the score distribution widened in both directions. Overall this shows that the true center of the test score distribution lies closer to the top end, and is superior to CLAP and competitive with training on real audio.

In all brevity, I showed that applying Gaussian noise leads to slight improvement to downstream classification test scores, especially when applying supervised

augmentation methods. Next, I discussed the performance of each augmentation method with respect to classification, and concluded that C2 and NonlinearN were the best respective unsupervised and supervised methods, and that C2 was the best augmentation overall. This supported the theoretical results provided by Zhang *et al.* [118, 119] and empirically indicated that their method was superior to other methods without theoretical justification.

Next, I juxtaposed each text dataset creation method, and found that the Label-Centered Noise and CLAP labeling of WavCaps were the best in terms of average test scores. I suggested that this could be because these two methods best take advantage of the domain knowledge and contrastive properties inherent in CLAP that the other two methods lack. Then, I compared the per-model performances along train, test, and augmentation axes. First I found that models optimizing mean squared error outperformed those optimizing cross entropy, and that the linear model outperformed the nonlinear variant. This result is also in line with the theoretical results from Zhang *et al.* I additionally saw that training with added noise helps the best performing models while having little impact on the worse models.

Finally, I performed an ablation study on the undersampling component of my experiment methodology and compared the classification scores of my Text-only Training models with those of CLAP zero-shot and training on the real audio data from the AudioCommons datasets. The ablation study revealed that training on large and imbalanced datasets is worse than training on smaller balanced ones. This justified my experimental methodology, as it corrects for biased distributions

and class imbalance. Lastly, I found that the median ToT model always outperforms CLAP zero-shot, and the best-scoring ToT models are competitive with training on real audio, and even matched or outperformed training on real audio in three out of eight datasets.

Chapter 8: Conclusion

In this final chapter I consider the limitations of my work and suggest areas for further improvement and future study. Furthermore, I summarize my research contribution.

8.1 Limitations and Future Work

Although I demonstrated that Text-only Training can be effective under the right circumstances, there are still many avenues of future research to explore. In this work I limited myself to binary classification to simplify and align the downstream classification task with Text-only Training. However, timbre is inherently multi-label and continuous-valued, and my in future work investigating both timbre classification and Text-only Training will explore these more difficult versions of the problem. Additionally, Text-only Training can be used for downstream tasks beyond simple classification: information retrieval [40], captioning [46], domain adaptation [19], and representation learning [80] are among the many tasks currently being studied. These machine learning problems are not typically solved without any training data whatsoever, and through future work I may consider applying similar techniques to those explored in this work to explore these more challenging problems.

In this investigation, I explored three methods for creating text data, but there are several potential improvements and extensions to my approach, and many more approaches are possible. For example, I did not exhaustively explore methods for adding noise, such as alternative noise distributions, a tuned variance parameter, or Gaussian noise with non-diagonal covariance matrices. Furthermore, in this work, I did not compare using different LLMs to generate training data, or different embedding models to encode them. As mentioned earlier, each LLM could have a different biased output distribution with respect to desired properties, and each embedding model has its own biases that might affect performance on downstream tasks. Yet, even when using the same LLM, there are more variations of dataset creation methods to be explored, such as prompt tuning or learning, swapping to multimodal LLMs, or using weights as data. For simplicity I restricted my generated datasets to have labels that matched the test sets, but in the real world many different descriptors might correctly be used to describe the same sound. For that reason, I will consider future work that investigates using multiple labels as prompts, out-of-domain labels as negative examples, synthetic data from multiple modalities, and other techniques to better mimic real data. Additionally, while it was beyond the scope of my focus on Text-only Training, I am particularly interested in exploring how to best enhance real data with supplemental text-based synthetic data, and I recommend future researchers explore this potentially fruitful area.

An important limitation to note is that I do not perform hyperparameter tuning on models or noise, nor claim to have found the best-performing paradigm or

framework for Text-only Training. This work serves as a proof of concept, empirical support to theoretical results, and a case study in an under-researched field, but not as a definitive source for the state of the art on this task. While that leaves the option open for future investigators to further optimize my solution to this problem, I recommend they continue to further develop the practice and theory behind it. For example, I only explored common-practice and theoretically justified augmentation techniques and models, and basic extensions thereof. However, Text-only Training is most often used in conjunction with far more complicated deep learning architectures such as multimodal transformers [8] and diffusion U-nets [81]. Additionally, I define three criteria for problems that best suit Text-only Training, but provide no empirical or theoretical justification. As future work, I intend to explore how much mutual information is shared between the missing modality and the language used to describe it. From there I can empirically measure how that parameter affects performance on downstream tasks, or work towards proving generalization bounds or information-theoretic relationships between the modalities [62].

My modality gap and representational alignment metrics, while insightful, could not fully capture the relationships between embedding subspaces nor could they predict downstream performance. Going forward I am interested in applying more sophisticated analysis techniques to this task, such as manifold learning, metric learning, and low-rank approximation. Additionally, there is interesting work being done in shaping the properties of the contrastive embedding spaces that underpin Text-only Training that could prove beneficial [34, 80]. My future

work will focus on creating separate modality-aware subspaces that have distinct information and properties to directly support cross-modal alignment.

Ultimately, many design choices made in this work can be changed, critiqued, or extended: there are far too many variations on this one specific case study to list—which underscores how much more research there is to be done in this field and on this specific problem. Text-only Training is only one facet of AI research at the intersection of broad and important research areas like domain adaptation, multimodal representation learning, explainable AI, and efficient machine learning. More than iterating on this design, future researchers should seek to use this work and Text-only Training to make insightful and far-reaching connections between these fields.

8.2 Conclusion

In this work I introduce a text-only supervised learning paradigm named Text-only Training, discuss the challenges of dataset curation and creation, address the modality gap, and expand on theoretical results with new empirical evidence. I present a case study on audio timbre classification and compare three training data generation techniques, fourteen augmentation strategies, and four classification models on eight timbre datasets. I analyze the effect of both clean and noisy data augmentations on the modality gap and representational alignment, and find that C2 is an effective data augmentation method for removing the modality gap. Furthermore I find that adding noise is generally beneficial to downstream tasks as

it improves classification performance without dramatically altering the representational alignment. However, I also show that supervised augmentation methods improve the Pair Alignment metric, while unsupervised and noisy methods tend to decrease it, meaning if a ToT practitioner's downstream task is sensitive to Pair Alignment, they should consider avoiding noisy unsupervised augmentation. Next, I discuss the performance of each augmentation method with respect to classification, and conclude that C2 is the best overall augmentation. Similarly, I empirically verify the theoretical result that linear classifiers optimizing mean squared error are ideal for cross-modal binary classification. Then, I compare methods for creating data suitable for ToT and find that leveraging the embedding model's inherent knowledge is superior to using dedicated text-text similarity methods for labeling raw text data or generating data via an LLM. I conduct an ablation study that both justifies my methodology and shows that naïvely creating large datasets is less effective that hand-crafting them. Finally, I show that Text-only Training is a viable machine learning paradigm by meeting or exceeding the performance of CLAP zero-shot and models trained on real audio.

Bibliography

[1] Emiliano Acevedo, Martín Rocamora, and Magdalena Fuentes. Domain Adaptation Method and Modality Gap Impact in Audio-Text Models for Prototypical Sound Classification, June 2025. arXiv:2506.04376 [cs].

[2] Andrea Agostinelli, Timo I. Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, Matt Sharifi, Neil Zeghidour, and Christian Frank. MusicLM: Generating Music From Text. 2023.

[3] Mohammad Al-Jaff. *Messing With The Gap: On The Modality Gap Phenomenon In Multimodal Contrastive Representation Learning*. PhD thesis, Uppsala University, 2023.

[4] Antonio Almudévar, José Miguel Hernández-Lobato, Sameer Khurana, Ricard Marxer, and Alfonso Ortega. Aligning Multimodal Representations through an Information Bottleneck. June 2025. ICML 2025.

[5] Na Min An, Eunki Kim, James Thorne, and Hyunjung Shim. I0T: Embedding Standardization Method Towards Zero Modality Gap, December 2024. arXiv:2412.14384 [cs].

[6] J. Aucouturier and F. Pachet. Improving Timbre Similarity : How high's the sky? 2004.

[7] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations, October 2020. arXiv:2006.11477 [cs].

[8] Clément Barbier, Baptiste Abeloss, and Stéphane Herbin. Bridging the modality gap: Training-free adaptation of vision-language models for remote sensing via visual prototypes. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR) Workshops*, pages 3057–3066, June 2025.

[9] Dmitry Bogdanov, Won Minz, Philip Tovstogan, and Alastair Porter. MTG-Jamendo Dataset, June 2019.

[10] Sebastian Braun and Hannes Gamper. Multi-label audio classification with a noisy zero-shot teacher. In *International Workshop on Acoustic Signal Enhancement (IWAENC)*, September 2024.

[11] Maxime Carron, Thomas Rotureau, Françoise Dubois, Nicolas Misdariis, and Patrick Susini. Speaking about sounds: a tool for communication on sound features. *J. of Design Research*, 15(2):85, 2017.

[12] Hongzhou Chen, Lianghua He, Yihang Liu, and Longzhen Yang. Visual Neural Decoding via Improved Visual-EEG Semantic Consistency, August 2024. arXiv:2408.06788 [cs].

[13] Annie Chu, Patrick O'Reilly, Julia Barnett, and Bryan Pardo. Text2FX: Harnessing CLAP Embeddings for Text-Guided Audio Effects. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, April 2025.

[14] Isabella Czedik-Eysenberg and Christoph Reuter. Evaluating Timbre-Related Audio Descriptors Across Different Libraries and Multimodal Embeddings. March 2025.

[15] Hannah Davis and Saif M. Mohammad. Generating Music from Literature. 2014.

[16] Zihao Deng, Yinghao Ma, Yudong Liu, Rongchen Guo, Ge Zhang, Wenhu Chen, Wenhao Huang, and Emmanouil Benetos. MusiLingo: Bridging music and text with pre-trained language models for music captioning and query response. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3643–3655, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[17] Soham Deshmukh, Benjamin Elizalde, Dimitra Emmanouilidou, Bhiksha Raj, Rita Singh, and Huaming Wang. Training Audio Captioning Models without Audio. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 371–375, Seoul, Korea, Republic of, April 2024. IEEE.

[18] Soham Deshmukh, Benjamin Elizalde, and Huaming Wang. Audio Retrieval with WavText5K and CLAP Training. pages 2948–2952, 2023.

[19] Soham Deshmukh, Rita Singh, and Bhiksha Raj. Domain Adaptation for Contrastive Audio-Language Models. pages 1680–1684, 2024.

[20] SeungHeon Doh, Keunwoo Choi, Jongpil Lee, and Juhan Nam. LP-MusicCaps: LLM-Based Pseudo Music Captioning. 2023.

[21] SeungHeon Doh, Minz Won, Keunwoo Choi, and Juhan Nam. Toward universal text-to-music retrieval. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023.

[22] Hao-Wen Dong, Xiaoyu Liu, Jordi Pons, Gautam Bhattacharya, Santiago Pascual, Joan Serrà, Taylor Berg-Kirkpatrick, and Julian McAuley. CLIPSonic: Text-to-Audio Synthesis with Unlabeled Videos and Pretrained Language-Vision Models. In *2023 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 1–5, New Paltz, NY, USA, October 2023. IEEE.

[23] Antonio D'Orazio, Maria Rosaria Briglia, Donato Crisostomi, Dario Loi, Emanuele Rodolà, and Iacopo Masi. Implicit Inversion turns CLIP into a Decoder, June 2025. arXiv:2505.23161 [cs].

[24] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: an Audio Captioning Dataset. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 736–740, Barcelona, Spain, May 2020. IEEE.

[25] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. CLAP Learning Audio Concepts from Natural Language Supervision. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, June 2023. ISSN: 2379-190X.

[26] Benjamin Elizalde, Soham Deshmukh, and Huaming Wang. Natural Language Supervision For General-Purpose Audio Representations. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 336–340, Seoul, Korea, Republic of, April 2024. IEEE.

[27] Ayşegül Özkaya Eren and Mustafa Sert. Audio Captioning using Gated Recurrent Units. 2020.

[28] Abrar Fahim, Alex Murphy, and Alona Fyshe. It's Not a Modality Gap: Characterizing and Addressing the Contrastive Gap, June 2024. arXiv:2405.18570.

[29] Abolfazl Farahani, Sahar Voghoei, Khaled Rasheed, and Hamid R. Arabnia. A Brief Review of Domain Adaptation. In Robert Stahlbock, Gary M. Weiss, Mahmoud Abou-Nasr, Cheng-Ying Yang, Hamid R. Arabnia, and Leonidas Deligiannidis, editors, *Advances in Data Science and Information Engineering*, pages 877–894. Springer International Publishing, Cham, 2021.

[30] Yassir Fathullah, Chunyang Wu, Egor Lakomkin, Junteng Jia, Yuan Shangguan, Ke Li, Jinxi Guo, Wenhan Xiong, Jay Mahadeokar, Ozlem Kalinli, Christian Fuegen, and Mike Seltzer. Prompting large language models with speech recognition abilities. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 13351–13355, 2024.

[31] Junjie Fei, Teng Wang, Jinrui Zhang, Zhenyu He, Chengjie Wang, and Feng Zheng. Transferable Decoding with Visual Entities for Zero-Shot Image Captioning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3113–3123, Paris, France, October 2023. IEEE.

[32] Rafael Ferrer and Tuomas Eerola. Semantic structures of timbre emerging from social and acoustic descriptions of music. *EURASIP Journal on Audio, Speech, and Music Processing*, 2011(1):11, December 2011.

[33] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, 2017.

[34] Shashank Goel, Hritik Bansal, Sumit Bhatia, Ryan A. Rossi, Vishwa Vinay, and Aditya Grover. CYCLIP: cyclic contrastive language-image pretraining. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pages 6704–6719, Red Hook, NY, USA, November 2022. Curran Associates Inc.

[35] Aaron Grattafiori and Others. The Llama 3 herd of models, 2024. Last accessed 2025/10/25.

[36] Sophia Gu, Christopher Clark, and Aniruddha Kembhavi. I can't believe there's no images! : Learning Visual Tasks Using Only Language Supervision. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2672–2683, October 2023. ISSN: 2380-7504.

[37] Julien Guinot, Alain Riou, Elio Quinton, and György Fazekas. SLAP: Siamese Language-Audio Pretraining Without Negative Samples for Music Understanding, June 2025. arXiv:2506.17815 [cs].

[38] Perfecto Herrera-Boyer, Geoffroy Peeters, and Shlomo Dubnov. Automatic Classification of Musical Instrument Sounds. *Journal of New Music Research*, 32(1):3–21, March 2003.

[39] Ruofan Hu, Yan Xia, Minjie Hong, Jieming Zhu, Bo Chen, Xiaoda Yang, Minghui Fang, and Tao Jin. Vela: Scalable Embeddings with Voice Large Language Models for Multimodal Retrieval, June 2025. arXiv:2506.14445 [cs].

[40] Young Kyun Jang and Ser-nam Lim. Towards Cross-modal Backward-compatible Representation Learning for Vision-Language Models, May 2024. arXiv:2405.14715 [cs].

[41] Qian Jiang, Changyou Chen, Han Zhao, Liqun Chen, Qing Ping, Son Dinh Tran, Yi Xu, Belinda Zeng, and Trishul Chilimbi. Understanding and Constructing Latent Modality Structures in Multi-Modal Representation Learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7661–7671, June 2023. ISSN: 2575-7075.

[42] Wei Jiang, Jingyu Liu, Zijin Li, Jiaxing Zhu, Xiaoyi Zhang, and Shuang Wang. Analysis and Modeling of Timbre Perception Features of Chinese Musical Instruments. In *2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS)*, pages 191–195, Beijing, China, June 2019. IEEE.

[43] Keunhyoung Luke Kim, Jongpil Lee, Sangeun Kum, Chae Lin Park, and Juhan Nam. Semantic Tagging of Singing Voices in Popular Music Recordings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:1656–1668, 2020.

[44] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. San Diego, CA, USA, May 2015. arXiv. arXiv:1412.6980 [cs].

[45] Zhifeng Kong, Sang-gil Lee, Deepanway Ghosal, Navonil Majumder, Ambuj Mehrish, Rafael Valle, Soujanya Poria, and Bryan Catanzaro. Improving Text-To-Audio Models with Synthetic Captions, 2024.

[46] Theodoros Kouzelis and Vassilis Katsouros. Weakly-supervised Automated Audio Captioning via text only training, 2023.

[47] Saksham Singh Kushwaha and Magdalena Fuentes. A multimodal prototypical approach for unsupervised sound classification. In *INTERSPEECH 2023*, pages 266–270, ISCA, August 2023. ISCA.

[48] Megan Lavengood. *A New Approach to the Analysis of Timbre*. PhD thesis, CUNY Academic Works, 2017.

[49] Jeong Ryong Lee, Yejee Shin, Geonhui Son, and Dosik Hwang. Diffusion Bridge: Leveraging Diffusion Model to Reduce the Modality Gap Between Text and Vision for Zero-Shot Image Captioning. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 4050–4059, June 2025.

[50] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 12888–12900. PMLR, July 2022.

[51] Wei Li, Linchao Zhu, Longyin Wen, and Yi Yang. DeCap: Decoding CLIP Latents for Zero-Shot Captioning via Text-Only Training. September 2022.

[52] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghao Xiao, Chenghua Lin, Anton Ragni, Emmanouil Benetos, Norbert Gyenge, Roger Dannenberg, Ruibo Liu, Wenhu Chen, Gus Xia, Yemin Shi, Wenhao Huang, Zili Wang, Yike Guo, and Jie Fu. MERT: Acoustic Music Understanding Model with Large-Scale Self-supervised Training. 2023.

[53] Weixin Liang, Yuhui Zhang, Yongchan Kwon, Serena Yeung, and James Zou. Mind the gap: understanding the modality gap in multi-modal contrastive representation learning. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, pages 17612–17625, Red Hook, NY, USA, November 2022. Curran Associates Inc.

[54] Cynthia C. S. Liem, Martha Larson, and Alan Hanjalic. When music makes a scene: Characterizing music in multimedia contexts via user scene descriptions. *International Journal of Multimedia Information Retrieval*, 2(1):15–30, March 2013.

[55] Christian Limberg, Fares Schulz, Zhe Zhang, and Stefan Weinzierl. Pitch-conditioned instrument sound synthesis from an interactive timbre latent space. In L. Gabrielli and S. Cecchi, editors, *Proceedings of the 28-th Int. Conf. on Digital Audio Effects (DAFx25)*, pages 433–440, Ancona, Italy, September 2025.

[56] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context, 2014. Version Number: 3.

[57] Shansong Liu, Atin Sakkeer Hussain, Chenshuo Sun, and Ying Shan. Music Understanding LLaMA: Advancing Text-to-Music Generation with Question Answering and Captioning. 2023.

[58] Zhiyue Liu, Jinyuan Liu, and Fanrong Ma. Improving cross-modal alignment with synthetic pairs for text-only image captioning. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, volume 38 of *AAAI'24/IAAI'24/EAAI'24*, pages 3864–3872. AAAI Press, February 2024.

[59] Hao Ma, Zhiyuan Peng, Xu Li, Yukai Li, Mingjie Shao, Qiuqiang Kong, and Ju Liu. Language-Queried Target Sound Extraction Without Parallel Training Data, 2024.

[60] Hao Ma, Zhiyuan Peng, Xu Li, Yukai Li, Mingjie Shao, Qiuqiang Kong, and Ju Liu. Language-queried target sound extraction without parallel training data. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, Hyderabad, India, April 2025. IEEE.

[61] Hao Ma, Zhiyuan Peng, Xu Li, Mingjie Shao, Xixin Wu, and Ju Liu. CLAPSep: Leveraging Contrastive Pre-Trained Model for Multi-Modal

Query-Conditioned Target Sound Extraction. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:4945–4960, 2024. Publisher: Institute of Electrical and Electronics Engineers (IEEE).

[62] Sijie Mai, https://orcid.org/0000-0001-9763-375X, View Profile, Ying Zeng, View Profile, Haifeng Hu, https://orcid.org/0000-0002-4884-323X, and View Profile. Multimodal Information Bottleneck: Learning Minimal Sufficient Unimodal and Multimodal Representations. *IEEE Transactions on Multimedia*, 25:4121–4134, January 2023. Publisher: IEEE Press.

[63] Maria Mannone, Veronica Distefano, and Giovanni Santini. Classes of Colors and Timbres: A Clustering Approach. *Electronic Journal of Applied Statistical Analysis*, 15(3):588–605, December 2022. Number: 3.

[64] Raja Marjieh, Ilia Sucholutsky, Pol Van Rijn, Nori Jacoby, and Thomas L. Griffiths. Large language models predict human sensory judgments across six modalities. *Scientific Reports*, 14(1):21445, September 2024.

[65] Peter McCabe and Patrick Donnelly. Classifying Audio Timbre Without Audio Using Text-only Training. *15th International Conference, EvoMUSART 2026, Held as Part of EvoStar 2026 [UNPUBLISHED]*, April 2026. UNPUBLISHED.

[66] Ashman Mehra, Aryan Mehra, and Pratik Narang. Classification and study of music genres with multimodal spectro-lyrical embeddings for music (slem). *Multimedia Tools and Applications*, 84(7):3701–3721, February 2025.

[67] Xinhao Mei, Xubo Liu, Mark D. Plumbley, and Wenwu Wang. Automated audio captioning: an overview of recent progress and new challenges. *EURASIP Journal on Audio, Speech, and Music Processing*, 2022(1):26, October 2022.

[68] Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D. Plumbley, Yuexian Zou, and Wenwu Wang. WavCaps: A ChatGPT-Assisted Weakly-Labelled Audio Captioning Dataset for Audio-Language Multimodal Research. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:3339–3354, 2024. Publisher: Institute of Electrical and Electronics Engineers (IEEE).

[69] Jan Melechovsky, Zixun Guo, Deepanway Ghosal, Navonil Majumder, Dorien Herremans, and Soujanya Poria. Mustango: Toward Controllable Text-to-Music Generation. 2023.

[70] David Nukrai, Ron Mokady, and Amir Globerson. Text-Only Training for Image Captioning using Noise-Injected CLIP. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4055–4063, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics.

[71] Changdae Oh, Junhyuk So, Hoyoon Byun, YongTaek Lim, Minchul Shin, Jong-June Jeon, and Kyungwoo Song. Geodesic Multi-Modal Mixup for Robust Fine-Tuning. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 52326–52341. Curran Associates, Inc., 2023.

[72] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding, January 2019. arXiv:1807.03748 [cs].

[73] Andy Pearce, Tim Brookes, and Russell Mason. Modelling Timbral Hardness. *Applied Sciences*, 9(3):466, January 2019.

[74] Andy Pearce, Tim Brookes, and Russell Mason. Timbral Characterisation Tool v0.2 Development Dataset, January 2019.

[75] Andy Pearce, Saeid Safavi, Brookes Tim, Wenwu Mason, Russell Wang, and Mark Plumbley. D5.8: Release of timbral characterisation tools for semantically annotating non-musical content, 2019. last accessed 2026/02/1.

[76] Di Qi, Lin Su, Jia Song, Edward Cui, Taroon Bharti, and Arun Sacheti. ImageBERT: Cross-modal Pre-training with Large-scale Weak-supervised Image-Text Data. 2020.

[77] Longtian Qiu, Shan Ning, and Xuming He. Mining fine-grained image-text alignment for zero-shot captioning via text-only training. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, volume 38 of *AAAI'24/IAAI'24/EAAI'24*, pages 4605–4613. AAAI Press, February 2024.

[78] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. (arXiv:2103.00020), February 2021. arXiv:2103.00020 [cs].

[79] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, February 2021. last accessed 2025/09/30.

[80] Sameera Ramasinghe, Violetta Shevchenko, Gil Avraham, and Ajanthan Thalaiyasingam. Accept the Modality Gap: An Exploration in the Hyperbolic Space. pages 27263–27272, 2024.

[81] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical Text-Conditional Image Generation with CLIP Latents, April 2022. arXiv:2204.06125 [cs].

[82] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, 2019.

[83] Lindsey Reymore, Emmanuelle Beauvais-Lacasse, Bennett K. Smith, and Stephen McAdams. Modeling Noise-Related Timbre Semantic Categories of Orchestral Instrument Sounds With Audio Features, Pitch Register, and Instrument Family. *Frontiers in Psychology*, 13:796422, April 2022.

[84] Lindsey Reymore and David Huron. Using auditory imagery tasks to map the cognitive linguistic dimensions of musical instrument timbre qualia. *Psychomusicology: Music, Mind, and Brain*, 30(3):124–144, 2020. Place: US.

[85] Lindsey Reymore, Jason Noble, Charalampos Saitis, Caroline Traube, and Zachary Wallmark. Timbre semantic associations vary both between and within instrumentsan empirical study incorporating register and pitch height. *Music Perception*, 40(3):253–274, February 2023. Publisher: University of California Press.

[86] François Role, Sébastien Meyer, and Victor Amblard. Fill the Gap: Quantifying and Reducing the Modality Gap in Image-Text Representation Learning, 2025.

[87] Kohei Saijo, Janek Ebbers, François G. Germain, Sameer Khurana, Gordon Wichern, and Jonathan Le Roux. Leveraging Audio-Only Data for Text-Queried Target Sound Extraction. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, Hyderabad, India, April 2025. IEEE.

[88] Charalampos Saitis and Kai Siedenburg. When ChatGPT talks timbre. In *Proceedings of the Third International Conference on Timbre*, pages 73–77, Aristotle University of Thessaloniki, 2023. The School of Music Studies Aristotle University of Thessaloniki.

[89] Charalampos Saitis and Stefan Weinzierl. The Semantics of Timbre. In Kai Siedenburg, Charalampos Saitis, Stephen McAdams, Arthur N. Popper, and Richard R. Fay, editors, *Timbre: Acoustics, Perception, and Cognition*, volume 69, pages 119–149. Springer International Publishing, Cham, 2019.

[90] Jessie Salas. Generating Music from Literature Using Topic Extraction and Sentiment Analysis. *IEEE Potentials*, 37(1):15–18, January 2018.

[91] Simon Schrodi, David T. Hoffmann, Max Argus, Volker Fischer, and Thomas Brox. Two Effects, One Trigger: On the Modality Gap, Object Bias, and Information Imbalance in Contrastive Vision-Language Models. October 2024.

[92] Peter H. Schönemann. A Generalized Solution of the Orthogonal Procrustes Problem. *Psychometrika*, 31(1):1–10, March 1966.

[93] Prem Seetharaman and Bryan Pardo. Audealize: Crowdsourced Audio Production Tools. *Journal of the Audio Engineering Society*, 64(9):683–695, September 2016.

[94] Peiyang Shi, Michael C. Welle, Mårten Björkman, and Danica Kragic. Towards understanding the modality gap in CLIP. May 2023.

[95] Myranda Uselton Shirk, Catherine Dang, Jaewoo Cho, Hanlin Chen, Lily Hofstetter, Jack Bijur, Claiborne Lucas, Andrew James, Ricardo-Torres Guzman, Andrea Hiller, Noah Alter, Amy Stone, Maria Powell, and Matthew E. Pontell. Leveraging large language models for automated detection of velopharyngeal dysfunction in patients with cleft palate. *Frontiers in Digital Health*, Volume 7-2025, 2025.

[96] Kai Siedenburg, Ichiro Fujinaga, and Stephen McAdams. A Comparison of Approaches to Timbre Descriptors in Music Information Retrieval and Music Psychology. *Journal of New Music Research*, 45(1):27–41, January 2016.

[97] Kai Siedenburg and Charalampos Saitis. The language of sounds unheard: Exploring musical timbre semantics of large language models, May 2023. arXiv:2304.07830 [cs].

[98] Haoyu Song, Li Dong, Weinan Zhang, Ting Liu, and Furu Wei. CLIP models are few-shot learners: Empirical studies on VQA and visual entailment. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Dublin, Ireland, 2022. Association for Computational Linguistics.

[99] Yi Su, Jisheng Bai, Qisheng Xu, Kele Xu, and Yong Dou. Audio-language models for audio-centric tasks: A survey, 2025.

[100] Derek Tam, Colin Raffel, and Mohit Bansal. Simple Weakly-Supervised Image Captioning via CLIP's Multimodal Embeddings. In *The AAAI-23 Workshop on Creative AI Across Modalities*, January 2023.

[101] Abbey E. Thompson. Playing Tag: An Analysis of Vocabulary Patterns and Relationships Within a Popular Music Folksonomy. page 33 p., April 2008.

[102] Haokun Tian, Stefan Lattner, and Charalampos Saitis. Assessing the alignment of audio representations with timbre similarity ratings, July 2025. ISMIR 2025, 2025/09/24.

[103] Vishaal Udandarao. Understanding and Fixing the Modality Gap in Vision-Language Models. Master's thesis, University of Cambridge, January 2022.

[104] Gwendal Le Vaillant and Yannick Molle. Contrastive timbre representations for musical instrument and synthesizer retrieval, September 2025. last accessed 2025/09/24.

[105] Georgios Velissaridis, Rajpreet Athwal, Muneeb Musharaf, George Fazekas, and Charalampos Saitis. Evaluating foundation models on timbre-related cognitive tasks. In *1st Workshop on Large Language Models for Music & Audio (LLM4MA)*, 2025. last accessed 2025/10/27.

[106] Zachary Wallmark. Semantic Crosstalk in Timbre Perception. *Music & Science*, 2:205920431984661, January 2019.

[107] Enguang Wang, Zhimao Peng, Zhengyuan Xie, Fei Yang, Xialei Liu, and Ming-Ming Cheng. GET: Unlocking the Multi-modal Potential of CLIP for Generalized Category Discovery. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pages 20296–20306. arXiv, March 2025. arXiv:2403.09974 [cs].

[108] Junyang Wang, Ming Yan, Yi Zhang, and Jitao Sang. From Association to Generation: Text-only Captioning by Unsupervised Cross-modal Mapping. volume 4, pages 4326–4334, August 2023. ISSN: 1045-0823.

[109] Junyang Wang, Yi Zhang, Ming Yan, Ji Zhang, and Jitao Sang. Zero-shot Image Captioning by Anchor-augmented Vision-Language Space Alignment, November 2022. arXiv:2211.07275 [cs].

[110] Yiyu Wang, Hao Luo, Jungang Xu, Yingfei Sun, and Fan Wang. Text Data-Centric Image Captioning with Interactive Prompts, March 2024. arXiv:2403.19193 [cs].

[111] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-Scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, Rhodes Island, Greece, June 2023. IEEE.

[112] Xuenan Xu, Zeyu Xie, Mengyue Wu, and Kai Yu. Beyond the status quo: A contemporary survey of advances and challenges in audio captioning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:95–112, 2024.

[113] Mu Yang, Szu-Jui Chen, Jiamin Xie, and John Hansen. Bridging the Modality Gap: Softly Discretizing Audio Representation for LLM-based Automatic Speech Recognition, June 2025. arXiv:2506.05706 [eess].

[114] Chao Yi, Yuhang He, De-Chuan Zhan, and Han-Jia Ye. Bridge the Modality and Capability Gaps in Vision-Language Model Selection. November 2024.

[115] Youngjae Yu, Jiwan Chung, Heeseung Yun, Jack Hessel, JaeSung Park, Ximing Lu, Rowan Zellers, Prithviraj Ammanabrolu, Ronan Le Bras, Gunhee Kim, and Yejin Choi. Multimodal knowledge alignment with reinforcement learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. last accessed 2025/06/30.

[116] Delong Zeng, Ying Shen, Man Lin, Zihao Yi, and Jiarui Ouyang. Zero-Shot Image Captioning with Multi-type Entity Representations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(21):22308–22316, April 2025. Number: 21.

[117] Yiming Zhang, Xuenan Xu, Ruoyi Du, Haohe Liu, Yuan Dong, Zheng-Hua Tan, Wenwu Wang, and Zhanyu Ma. Zero-shot audio captioning using soft and hard prompts. *IEEE Transactions on Audio, Speech and Language Processing*, 33:2045–2058, 2025.

[118] Yuhui Zhang, Jeff Z. HaoChen, Shih-Cheng Huang, Kuan-Chieh Wang, James Zou, and Serena Yeung. Diagnosing and Rectifying Vision Models using Language. September 2022.

[119] Yuhui Zhang, Elaine Sui, and Serena Yeung. Connect, Collapse, Corrupt: Learning Cross-Modal Tasks with Uni-Modal Data. October 2023.

[120] Pengfei Zhao, Rongbo Luan, Wei Zhang, Peng Wu, and Sifeng He. Guiding Cross-Modal Representations with MLLM Priors via Preference Alignment, June 2025. arXiv:2506.06970 [cs].

[121] Yufan Zhou, Chunyuan Li, Changyou Chen, Jianfeng Gao, and Jinhui Xu. Lafite2: Few-shot Text-to-Image Generation, October 2022. arXiv:2210.14124 [cs].

[122] Yufan Zhou, Bingchen Liu, Yizhe Zhu, Xiao Yang, Changyou Chen, and Jinhui Xu. Shifted Diffusion for Text-to-image Generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10157–10166, June 2023. ISSN: 2575-7075.

[123] Yufan Zhou, Ruiyi Zhang, Changyou Chen, Chunyuan Li, Chris Tensmeyer, Tong Yu, Jiuxiang Gu, Jinhui Xu, and Tong Sun. Towards Language-Free Training for Text-to-Image Generation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17886–17896, New Orleans, LA, USA, June 2022. IEEE.