Differential Multi-Track Gain Analysis for AI Mixing Assistance

Pratham Vadhulas

September 18, 2025

1 Research Statement

Current Audio-Language Models (ALMs) show promise for descriptive audio tasks, but effective music mixing requires a relational understanding of multi-track audio that goes beyond simple analysis. This research investigates a novel framework where an ALM is conditioned on a designated anchor track (e.g., drums) to learn the relative levels of other stems and generate actionable gain-balancing advice. To achieve this, the study will leverage the MUSDB18 dataset to train and evaluate the model. The evaluation will be comprehensive, combining automated metrics with qualitative assessments from professional audio engineers to gauge the musicality, effectiveness, and real-world usefulness of the generated advice.

1.1 Research Questions

1.1.1 Primary Research Question

To what extent can an Audio Language Model, conditioned on an anchor track (e.g., drums), learn about relative levels of multi-track audio to generate effective gain-balancing advice for music mixing?

1.1.2 Secondary Research Questions

1. **Architecture** What is an effective model architecture for representing multi-track stems and a designated anchor track, enabling an Audio Language Model to learn and reason about their relative levels for music mixing?

2. **Learned Conventions** Beyond balancing the immediate tracks, does the model's generated advice demonstrate an understanding of established mixing conventions and genre-specific expectations (e.g., the typical vocal-to-instrument balance in pop versus jazz)?

3. **Evaluation & Usefulness** How do audio engineers and producers rate the effectiveness, musicality, and actionability of the generated gain-balancing advice, and what qualitative feedback do they provide on its integration into their workflow?

4. **Metric Correlation** For the specific task of evaluating mixing advice, what is the correlation between subjective human preference judgments and automated evaluation metrics (e.g., LLM-as-a-Judge, BERTscore)?

1.2 Scope and Limitations

This research is specifically focused on generating textual advice for gain parameter adjustments in multi-track audio by conditioning an ALM on an anchor track. The model's output will be advisory text that may include gain predictions, but evaluation will focus on the overall quality and usefulness of the advice rather than the accuracy of specific numerical predictions. The investigation deliberately excludes other mixing parameters such as equalization (EQ), dynamic range compression, and spatial effects to maintain a focused scope. The study will primarily use the MUSDB18 dataset for training and evaluation. Furthermore, the proposed system is designed for offline analysis and is not intended for real-time, interactive applications. The validity of this work relies on the assumption that the professionally mixed versions of the tracks in the MUSDB18 dataset represent a perceptually valid ground truth for a well-balanced mix and that the dataset is of sufficient quality and diversity for the task.

2 Motivation

2.1 Research Gap

Current models often operate as "black boxes," analyzing audio in isolation and generating mixed outputs without a clear, musically relevant context. This limits their usefulness for audio engineers, who need explainable, actionable guidance rather than final results. Engineers want to understand the reasoning behind mixing decisions and maintain control over the creative process. No existing systems provide anchor-based mixing advice, leaving a gap in the ability to learn and reason about the nuanced, context-dependent gain relationships that are fundamental to professional mixing.

2.2 Research Impact

This research aims to bridge this gap by introducing an anchor-conditioned framework for training ALMs. By teaching the model to reason about mix balance relative to a stable reference point, we can better align its "understanding" of multi-track audio with the cognitive workflows of human engineers. This approach enables effective human-AI collaboration, where an engineer could provide multi-track stems and receive contextually relevant, actionable advice. This paradigm shifts the focus from purely automated, black-box solutions to co-creative systems that empower musicians and producers with tools that understand and adapt to their creative process, fostering a more interactive and transparent mixing workflow. This research aims to build a foundation for AI mixing assistants that can seamlessly integrate into professional workflows.

3 Related Work

Addressing the gaps in current automatic mixing systems namely the lack of explainability, interactivity, and user control requires a conceptual shift away from fully autonomous "black box" solutions. The emerging field of "Co-Creative AI" provides a compelling framework for this shift, envisioning AI not as a replacement for human creativity, but as an intelligent partner that augments the artistic workflow [1, 2]. This collaborative paradigm is particularly resonant in a

field as nuanced and subjective as audio mixing, where artistic intent is paramount. Recent work has demonstrated the potential of co-creative systems in music production, showing positive user adoption and acceptance when AI tools provide appropriate levels of control and collaboration [3, 4]. This section reviews the evolution of co-creative systems and the underlying technologies that make a language-driven, interactive mixing agent feasible.

3.1 Paradigms of Co-Creative AI in Audio

Within the landscape of co-creative systems for audio engineering, early paradigms focused on automating the mixing process. These automated mixers often functioned as **black box systems**, which take raw multitrack audio as input to produce a fully mixed output with minimal user intervention [5, 6]. While some of these systems provide a degree of user control through high-level parameters, they fundamentally abstract the underlying process, which limits the fine-grained control required in professional workflows. In contrast, a more recent and promising paradigm is the **Language Bridge**, where natural language serves as the primary interface for interaction. This approach allows users to articulate creative intent in descriptive terms, and our research is situated within this paradigm. Recent work has begun exploring this direction across multiple dimensions: systems like MixAssist demonstrate the potential for audio-language models to provide contextual mixing advice through natural language dialogue [7]; Text2FX leverages CLAP embeddings to control audio effects through open-vocabulary natural language prompts [8]; and LLM2Fx shows that Large Language Models can predict audio effect parameters directly from textual descriptions in a zero-shot manner [9]. Additionally, research has explored speech recognition for mixer control [10], natural language interfaces for audio production tools [11], and word embeddings for automatic equalization [12].

3.2 The Evolution of Language-Driven Models

The journey towards sophisticated language-driven models has been incremental, progressing through several key phases that have transformed how we interact with and control various modalities through natural language. It began with **task-specific supervised models** for applications like audio tagging, which classified sounds into predefined categories [13]. A subsequent shift towards **representation learning** aimed to create more general-purpose audio embeddings to capture richer semantic information [14]. The emergence of **contrastive dual-encoder models** like CLAP marked a significant advancement, learning audio concepts from natural language supervision and enabling zero-shot audio classification and retrieval [15, 16]. The rise of deep **generative models** like GANs and VAEs enabled audio synthesis and transformation, though precise control remained a challenge [17, 18]. More recently, **generative and diffusion-based multimodal models** have emerged, leveraging the power of diffusion models and large language models for text-to-audio generation [19, 20]. Most significantly, the development of **Large Audio-Language Models (LALMs)** and instruction-following systems has been revolutionary, with models like AudioLM demonstrating language modeling approaches to audio generation [21]. The profound language understanding of these models, when combined with the ability to process audio, forms the technological cornerstone of the Language Bridge paradigm, making it possible to connect linguistic intent directly to audio manipulation [22].

3.3 Multimodal Large Language Models

The development of Multimodal Large Language Models (MM-LLMs) represents a significant advancement in AI capabilities, enabling models to process and understand multiple modalities simultaneously. Several architectural paradigms have emerged for creating these powerful systems:

Unified Multimodal Models are trained from the ground up on vast datasets spanning multiple modalities, allowing for deep integration of information across different data types [23]. These models, such as Unified-IO, demonstrate the potential for truly unified understanding across vision, language, and audio domains.

Modality Interface Architectures involve augmenting pre-trained, text-only LLMs with specialized encoders for other modalities. This approach allows LLMs to perceive and process new types of information without altering their core architecture. Notable examples include Qwen-Audio, which provides universal audio understanding capabilities across over 30 tasks [24], and SALMONN, which integrates speech and audio encoders with LLMs to achieve generic hearing abilities [25].

Any-to-Any Multimodal Systems represent the cutting edge, enabling models to both perceive and generate content across multiple modalities. NExT-GPT demonstrates this capability, connecting LLMs with multimodal adaptors and diffusion decoders to handle arbitrary combinations of text, image, video, and audio [26]. Similarly, Audiobox provides unified audio generation capabilities across speech, sound, and music through natural language prompts [27].

Instruction-Following Multimodal Models focus on following complex instructions across modalities. Macaw-LLM seamlessly integrates visual, audio, and textual information for multi-turn dialogue scenarios [28], while PandaGPT demonstrates emergent cross-modal behaviors through instruction-following capabilities [29].

3.4 Fine-Tuning Methods for Audio Tasks

The application of fine-tuning to MM-LLMs has unlocked a wide range of capabilities across various audio domains. This section provides an overview of the state-of-the-art methods for general audio sounds, music, and speech.

3.4.1 General Audio Sounds

Audio Understanding Models such as LTU [30], SALMONN [25], Qwen-Audio [24], and UNIFIED-IO 2 [31] leverage LLMs as their backbone for analyzing and interpreting diverse environmental sounds. Additionally, AudioGPT [32] and HuggingGPT [33] function as intelligent interfaces that coordinate various tools for audio understanding tasks. Furthermore, recent work has enhanced automated audio captioning by integrating pretrained models with LLMs [34].

Audio Generation Notable models in audio generation include TANGO [35], Make-an-Audio 2 [36], WavJourney [37], AudioLM [21], Audiobox [27], and UniAudio [38]. These approaches utilize a variety of techniques, such as text embedders (e.g., FLAN-T5 in TANGO), latent diffusion models, LLM agents for integrating audio models (WavJourney), discrete audio tokenization

(AudioLM), LLMs for data construction and flow-matching (Audiobox), and unified sequence tokenization for various audio types (UniAudio).

3.4.2 Music

Music Understanding In the music domain, models like Music Understanding LLaMA (MU-LLaMA) [39], LLARK [40], MusicAgent [41], LyricWhiz [42], and ChatMusician [43] are employed to analyze detailed music features, leverage refined annotations, automate tasks, and improve lyric transcription.

Music Generation Music generation methods include MusicLM [44], Jukebox [45], MusicGen [46], Music ControlNet [34], M2UGen [47], ChatMusician [43], and SongComposer [48]. These often use Transformer architectures for conditional music generation, compress raw audio into discrete codes (Jukebox), incorporate LLMs as text embedders (MusicGen, Music ControlNet), combine LLMs with other pretrained models (M2UGen), or intrinsically generate symbolic music (ChatMusician, SongComposer).

Music Editing Loop Copilot [49] combines LLMs with specialized AI music models to facilitate conversational, collaborative music loop creation and editing.

3.4.3 Speech

Speech Understanding Key contributions in speech understanding come from SpeechGPT [50], AudioPaLM [51], Speech-LLaMA [52], and recent works that utilize LLMs as structural backbones to process spoken language, support multimodal content, transfer inter-modal knowledge, and improve Automatic Speech Recognition (ASR) accuracy through in-context learning or specialized connector structures [53, 54].

Speech Generation VALL-E [55] uses a neural codec language model to reframe text-to-speech (TTS) as a conditional language modeling task. Other approaches integrate LLaMA/OPT with VALL-E [56], and LauraGPT [57] is a unified GPT model for speech recognition, translation, and TTS. Additionally, some research investigates word surprisal to improve speech synthesis prosody [58].

3.5 How LLMs are Utilized in Audio Tasks

The integration of LLMs in audio tasks can be categorized into several key approaches:

- **LLMs as Backbone:** Pre-trained LLMs (e.g., LLaMA) are used as the central architecture, either with modality-specific encoders/decoders (cascade approach) or by tokenizing raw audio into discrete tokens for direct LLM input (unified approach).
- **LLMs as Conditioner:** LLMs encode text prompts into embeddings that condition the audio generation process.

- **LLMs as Labeller:** LLMs are employed to convert class labels from large audio datasets into full-sentence audio descriptions or captions, often utilizing self-instruction techniques.
- **LLMs as Agent:** LLMs act as controllers, interfacing with and orchestrating various external tools to accomplish diverse audio tasks.
- **LLMs Inspired Backbone:** This approach discretizes audio into tokens for next-token prediction, aiming for LLM-like emergent capabilities in audio.

Furthermore, tool-augmented multimodal agents like ControlLLM [59], ModelScope-Agent [60], and HuggingGPT [33] can generate speech and music by invoking specialized audio tools. NExT-GPT also provides a framework that supports mixed inputs and outputs including audio, with diffusion models attached to the LLM [26].

4 Proposed Method

This research introduces and validates a novel, anchor-conditioned framework for fine-tuning an Audio-Language Model (ALM) as a specialized music mixing assistant. This approach is designed to emulate the cognitive workflow of human engineers, who often establish a mix foundation by balancing key elements against a stable anchor track (e.g., drums or vocals). By conditioning the model on an explicit anchor, we guide it to learn the relational and context-dependent principles of gain balancing. The methodology is executed in three distinct phases: anchor-based dataset creation, a multi-stem input formulation, and parameter-efficient fine-tuning.

4.1 Anchor-Based Dataset for Relational Mixing Instruction

The foundation of this work is the creation of a new, structured dataset designed specifically to teach the relational reasoning of gain balancing relative to a musical anchor.

Source Audio We will utilize the high-quality, professionally produced multi-track stems from the MUSDB18 dataset [61]. The use of this existing, permissively licensed multi-track dataset follows established ethical data sourcing practices.

Data Generation and Annotation For each song in the MUSDB18 dataset, we will programmatically generate training instances. A single instance will consist of a designated anchor track (e.g., the drum stem), a target track (e.g., the vocal stem), and a corresponding text annotation. The annotation will be a templated textual instruction describing the ideal gain relationship between the target and the anchor, derived from the professionally mixed reference. For example: "Relative to the drums, the vocal stem is well-balanced. It sits clearly on top of the mix without overpowering the rhythm section." To create varied training examples, we will also generate versions where the target track is intentionally made too loud or too soft, with corresponding instructional text, e.g., "Relative to the drums, the vocal stem is too loud. To achieve a better balance, its gain should be decreased." This strategy provides a scalable method

for creating a large, high-quality instruction-following dataset that teaches the model to assess and advise on gain relationships from a musically grounded perspective.

4.2 Model Architecture and Anchor-Conditioned Input Formulation

To process the relational audio data, we will employ a state-of-the-art, pre-trained ALM capable of handling multiple audio inputs.

Base Model We will use Qwen-Audio as the foundational model [24]. Its architecture has demonstrated the ability to process multiple audio streams within a single context, making it exceptionally well-suited for the anchor-based comparison task central to our methodology.

Input Structure The model will be presented with a targeted set of stems for comparison. A typical input will consist of two audio files: the anchor stem (e.g., drums) and a target stem (e.g., vocals). These audio files will be accompanied by a text prompt that instructs the model to provide gain-balancing advice for the target stem relative to the anchor stem. This focused input structure allows the model to learn the specific gain relationships between pairs of instruments, mirroring a fundamental aspect of the mixing process.

4.3 Parameter-Efficient Fine-Tuning

To adapt the pre-trained ALM to our specialized mixing task, we will employ a parameter-efficient fine-tuning (PEFT) methodology.

Methodology We will use Low-Rank Adaptation (LoRA) to fine-tune the LLM component of the Qwen-Audio model [62]. LoRA is a standard, computationally efficient method that has been proven highly effective for adapting large models to downstream tasks. By inserting small, trainable low-rank matrices into the model's architecture, LoRA allows for significant task-specific adaptation while keeping the vast majority of the base model's parameters frozen. This approach dramatically reduces the computational resources required for training and mitigates the risk of catastrophic forgetting.

QLoRA Variant In resource-constrained settings, we will employ QLoRA, which performs LoRA fine-tuning on a 4-bit quantized copy of the base model [63]. QLoRA preserves model quality while enabling the fine-tuning of large models on a single high-memory GPU, facilitating broader experimentation (e.g., ablations across different anchor-target pairings, prompt formats, and genre contexts) without sacrificing performance.

5 Proposed Evaluation

The evaluation of our anchor-conditioned Audio-Language Model (ALM) will be centered on human-centric assessments to directly address our research questions concerning the usefulness and musicality of the generated advice. We will employ a mixed-methods approach, combining qualitative feedback from audio professionals with quantitative ratings and automated metrics.

The primary goal is to determine not just the technical accuracy of the advice, but its real-world value in a professional mixing context.

5.1 Subjective Evaluation by Audio Professionals

The core of our evaluation will be a formal listening study with a cohort of experienced audio engineers and producers.

Procedure Participants will be presented with a series of mixing scenarios. Each scenario will consist of a set of multi-track stems, a designated anchor track, and a target track with a subtle gain imbalance. They will be shown the gain-balancing advice generated by our fine-tuned ALM for each scenario. Participants will then rate the advice on several key criteria.

Evaluation Criteria Using a 7-point Likert scale, participants will rate the following:

- **Effectiveness:** How well would this advice solve the given mixing problem?
- **Musicality:** Is the advice musically sensible and appropriate for the genre?
- **Actionability:** Is the advice clear, specific, and easy to implement in a DAW?
- **Helpfulness:** How helpful would this advice be in a real-world mixing session?

Qualitative Feedback In addition to quantitative ratings, we will collect open-ended qualitative feedback. Participants will be asked to provide written comments on the strengths and weaknesses of the generated advice, its potential impact on their workflow, and any suggestions for improvement. This qualitative data will be analyzed using thematic analysis to identify recurring themes and provide deeper insights into the model's performance.

5.2 Analysis of Learned Mixing Conventions

To address the research question on learned conventions, a subset of the evaluation will focus on genre-specific scenarios. We will create test cases from different genres (e.g., pop, jazz, rock) and analyze whether the model's advice reflects established mixing norms for those genres (e.g., vocal level in pop vs. jazz). The ratings and qualitative feedback from the audio professionals on these specific scenarios will be used to assess the model's understanding of genre-specific expectations.

5.3 Correlation of Subjective and Automated Metrics

A key goal of this research is to understand the relationship between human perception and automated evaluation.

Automated Metrics We will compute a suite of automated metrics for the generated advice, including:

- **LLM-as-a-Judge:** A separate, powerful LLM (e.g., GPT-4) will be prompted to score the quality of the generated advice based on a detailed rubric.
- **Semantic Similarity Metrics:** We will calculate BERTScore, ROUGE-L, and METEOR by comparing the model's output to a reference text annotation.

Correlation Analysis We will perform a statistical analysis to determine the correlation (e.g., using Pearson or Spearman correlation coefficients) between the subjective ratings from our human evaluation and the scores from the automated metrics. This analysis will help quantify how well current automated metrics align with human judgments of mixing advice, addressing a specific secondary research question.

5.4 Ablation Studies

To validate our architectural choices, we will conduct ablation studies comparing our proposed model to several baselines:

- **No-Anchor Baseline:** A model trained without an explicit anchor track to assess the impact of the anchor-conditioning framework.
- **Generic ALM Baseline:** A general-purpose, pre-trained ALM (without fine-tuning) to measure the effectiveness of our task-specific training.

These baselines will be evaluated using the same subjective and automated protocols to quantify the performance gains of our proposed method.

6 Novelty of Proposed Work

This research introduces a novel conceptual framework for generating gain-balancing advice by conditioning an Audio-Language Model (ALM) on an anchor track, representing a significant departure from current "black-box" or single-mix analysis approaches. The novelty of this work is articulated across three core areas: problem formulation, methodological innovation, and evaluation methodology.

6.1 Advancements in Problem Formulation

- **Anchor-Conditioned Relational Reasoning:** To our knowledge, this is the first work to explicitly model the common human workflow of mixing relative to an anchor element. This shifts the task from absolute audio analysis to relational reasoning, better aligning the AI's "understanding" with the cognitive process of audio engineers.
- **Learning Implicit Mixing Conventions:** By framing the problem around anchor-target relationships, our approach is designed to implicitly learn established, genre-specific mixing conventions (e.g., the typical balance between kick drum and bass, or vocals and rhythm section), moving beyond simple gain prediction to a more musically aware form of guidance.

6.2 Methodological Innovations

- **Anchor-Target Architectural Representation:** We propose and will investigate novel architectural approaches for representing the relationship between a designated anchor track and other stems within a multi-modal, multi-audio input ALM. This addresses a key architectural challenge in enabling models to reason about the relative properties of different audio streams.
- **Structured, Anchor-Based Instruction Dataset:** We introduce a new methodology for creating a structured, instruction-following dataset based on anchor-target pairs. This provides a scalable and targeted way to teach relational audio concepts to an ALM.

6.3 Evaluation Methodology Advances

- **Focus on Usefulness and Actionability:** Our evaluation protocol is uniquely centered on assessing the practical usefulness, musicality, and actionability of the generated advice through formal studies with professional audio engineers. This moves beyond traditional objective metrics to measure real-world value.
- **Systematic Correlation of Human and Automated Metrics:** This research will provide one of the first systematic analyses of the correlation between subjective human judgments of mixing advice and a suite of automated metrics (including LLM-as-a-Judge and semantic similarity scores). This will yield valuable insights into the validity of current automated evaluation techniques for this specific creative task.

6.4 Impact on State-of-the-Art

This work advances the state-of-the-art by:

- Proposing a new, cognitively aligned paradigm for human-AI collaboration in music mixing.
- Establishing a novel methodology for teaching relational audio concepts to ALMs.
- Providing a critical analysis of the alignment between human perception and automated metrics in a creative audio domain.
- Bridging the gap between the technical capabilities of ALMs and the practical needs of audio professionals.

7 Required Resources

7.1 Computational Resources

- **Local GPU Infrastructure**: High-memory GPUs (24GB+ VRAM) for fine-tuning Qwen2-Audio-7B model with LoRA/QLoRA
- **Storage**: Large-scale local storage for MUSDB18 dataset and generated training data estimated to be max 100GB.

- **HPC Access**: Access to PACE for larger-scale training when needed

7.2 Software and Tools

- **Deep Learning Stack**: PyTorch 2.0+, Transformers 4.30+, PEFT for LoRA implementation
- **Audio Processing**: LibROSA, SoundFile for audio analysis and preprocessing
- **Model Infrastructure**: Qwen2-Audio-7B-Instruct as base model, BitsAndBytes for quantization
- **Experiment Tracking**: Weights & Biases and MLflow for experiment management
- **Containerization**: Docker and Docker Compose for reproducible environments
- **Configuration Management**: Hydra for configuration management and experiment organization

7.3 Datasets and Data

- **MUSDB18 Dataset**: High-quality multi-track stems for training and evaluation
- **Generated Training Data**: Training instances consisting of prompts, audio tracks (with designated anchor), and textual advice responses created from MUSDB18
- **Evaluation Data**: Subset of MUSDB18 for human evaluation studies
- **Reference Mixes**: Professionally mixed versions from MUSDB18 as ground truth

7.4 Human Resources

- **Audio Engineers**: Professional mixing engineers for subjective evaluation and feedback
- **Research Collaborators**: Experts in machine learning, audio processing, and music cognition
- **User Study Participants**: Audio professionals for evaluation studies and qualitative feedback
- **Technical Support**: DevOps and MLOps expertise for infrastructure management

7.5 Infrastructure and Development Environment

- **Development Environment**: Docker-based development setup with Jupyter Lab integration
- **Version Control**: Git-based version control with DVC for data versioning
- **Monitoring**: Prometheus and Grafana for system monitoring during training
- **CI/CD Pipeline**: Automated testing and deployment pipelines for model validation

7.6 Budget Considerations

- **Hardware**: Local GPU hardware for model training and evaluation
- **Storage**: Local storage for datasets, checkpoints, and experiment artifacts
- **Software Licenses**: Professional audio software for reference and validation
- **Personnel**: Compensation for expert consultants and evaluation study participants
- **HPC Access**: PACE cluster usage fees for larger-scale experiments

8 Deliverables

8.1 Research Publications

- **Conference Papers**: 2-3 papers at top-tier conferences (ISMIR, ICASSP, AES)
- **Journal Articles**: 1-2 papers in high-impact journals (JASA, IEEE TASLP)
- **Workshop Presentations**: Presentations at relevant workshops and symposiums
- **Technical Reports**: Detailed technical documentation of methods and results

8.2 Software and Code

- **Open-Source Implementation**: Complete source code of the automatic mixing system
- **API and SDK**: Software development kit for integration with existing tools
- **Plug-in Development**: VST/AU plugins for popular digital audio workstations
- **Web Application**: Browser-based interface for automatic mixing

8.3 Datasets and Resources

- **Training Datasets**: Curated datasets of mixed tracks with annotations
- **Evaluation Benchmarks**: Standardized test sets for system comparison
- **Pre-trained Models**: Trained models ready for use and further development
- **Documentation**: Comprehensive documentation for datasets and models

8.4 Evaluation Tools

- **Evaluation Framework**: Software tools for objective and subjective evaluation
- **Benchmarking Suite**: Automated testing and comparison tools
- **Visualization Tools**: Software for analyzing and visualizing mixing decisions
- **User Study Materials**: Protocols and materials for human evaluation studies

8.5 Documentation and Tutorials

- **User Manuals**: Comprehensive guides for end users
- **Developer Documentation**: Technical documentation for researchers and developers
- **Tutorial Videos**: Educational content demonstrating system capabilities
- **Best Practices Guide**: Recommendations for optimal system usage

8.6 Intellectual Property

- **Patents**: Novel algorithms and methods (if applicable)
- **Open Source Licenses**: Appropriate licensing for public release
- **Commercial Licensing**: Options for commercial use and integration
- **Research Agreements**: Collaboration agreements with industry partners

8.7 Dissemination

- **Conference Presentations**: Oral and poster presentations
- **Demonstrations**: Live demonstrations at conferences and workshops
- **Media Coverage**: Press releases and media outreach
- **Community Engagement**: Participation in relevant online communities and forums

9 Timeline

9.1 Phase 1: Foundation and Dataset Creation (Weeks 1-8)

- **Weeks 1-2**: Literature review and finalization of anchor-conditioned model architecture.
- **Weeks 3-5**: Development of the anchor-based dataset generation pipeline.
- **Weeks 6-8**: Generation and validation of the anchor-based instructional dataset.

9.2 Phase 2: Model Fine-Tuning and Development (Weeks 9-20)

- **Weeks 9-12**: Implementation of the data loading and pre-processing for the anchor-target model.
- **Weeks 13-16**: Initial model fine-tuning (LoRA/QLoRA) on the new dataset.
- **Weeks 17-20**: Iterative refinement of the model based on initial results; development of baseline models for ablation.

9.3 Phase 3: Evaluation and Analysis (Weeks 21-36)

- **Weeks 21-24**: Development of the evaluation protocol, including surveys for the human study and automated metric scripts.
- **Weeks 25-30**: Recruitment of audio professionals and execution of the subjective evaluation study.
- **Weeks 31-36**: Analysis of quantitative and qualitative data; correlation analysis between subjective and automated metrics.

9.4 Phase 4: Dissemination and Finalization (Weeks 37-52)

- **Weeks 37-44**: Preparation of manuscripts for conference and journal submission based on research findings.
- **Weeks 45-48**: Refinement of open-source code, models, and dataset for public release.
- **Weeks 49-52**: Finalization of thesis/dissertation and dissemination of results through presentations.

9.5 Milestones and Deliverables

- **Week 8**: Finalized anchor-based dataset.
- **Week 20**: First fine-tuned prototype of the anchor-conditioned model.
- **Week 36**: Completed analysis of human evaluation study and metric correlation.
- **Week 44**: First draft of primary research paper.
- **Week 52**: Final deliverables, including open-source code and pre-trained models.

9.6 Risk Mitigation

- **Data Quality**: Validate the generated dataset with a small pilot study before full-scale training.
- **Recruitment Challenges**: Begin recruitment for the human evaluation study early and leverage professional networks.
- **Technical Challenges**: Maintain clear versioning of models and experiments; rely on proven PEFT techniques.
- **Timeline Delays**: Incorporate buffer time within each phase and maintain a clear focus on the primary research question.

References

[1] R. Louie, A. Coenen, C. Z. Huang, M. Terry, and C. J. Cai, "Novice-ai music co-creation via ai-steering tools for deep generative models," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, (Honolulu HI USA), p. 1–13, ACM, Apr. 2020.

[2] A. Tsiros and A. Palladini, "Towards a human-centric design framework for ai assisted music production," June 2020.

[3] R. Bougueng Tchemeube, J. Ens, C. Plut, P. Pasquier, M. Safi, Y. Grabit, and J.-B. Rolland, "Evaluating human-ai interaction via usability, user experience and acceptance measures for mmm-c: A creative ai system for music composition," in *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, (Macau, SAR China), p. 5769–5778, International Joint Conferences on Artificial Intelligence Organization, Aug. 2023.

[4] S. S. Vanka, M. Safi, J.-B. Rolland, and G. Fazekas, "Adoption of ai technology in the music mixing workflow: An investigation," Sept. 2023. arXiv:2304.03407 [cs].

[5] C. J. Steinmetz, J. Pons, S. Pascual, and J. Serrà, "Automatic multitrack mixing with a differentiable mixing console of neural audio effects," Oct. 2020. arXiv:2010.10291 [eess].

[6] M. A. Martínez-Ramírez, W.-H. Liao, G. Fabbro, S. Uhlich, C. Nagashima, and Y. Mitsufuji, "Automatic music mixing with deep learning and out-of-domain data," Aug. 2022. arXiv:2208.11428 [eess].

[7] M. Clemens and A. Marasović, "Mixassist: An audio-language dataset for co-creative ai assistance in music mixing," July 2025. arXiv:2507.06329 [cs].

[8] A. Chu, P. O'Reilly, J. Barnett, and B. Pardo, "Text2fx: Harnessing clap embeddings for text-guided audio effects," Feb. 2025. arXiv:2409.18847 [eess].

[9] S. Doh, J. Koo, M. A. Martínez-Ramírez, W.-H. Liao, J. Nam, and Y. Mitsufuji, "Can large language models predict audio effects parameters from natural language?," July 2025. arXiv:2505.20770 [cs].

[10] S.-C. Lai, Y.-H. Hung, Y.-C. Zhu, S.-T. Wang, M.-H. Sheu, and W.-H. Juang, "A low-cost smart digital mixer system based on speech recognition," *Electronics*, vol. 11, p. 604, Feb. 2022.

[11] B. Pardo, M. Cartwright, P. Seetharaman, and B. Kim, "Learning to build natural audio production interfaces," *Arts*, vol. 8, p. 110, Aug. 2019.

[12] S. Venkatesh, D. Moffat, and E. R. Miranda, "Word embeddings for automatic equalization in audio mixing," *Journal of the Audio Engineering Society*, vol. 70, p. 753–763, Nov. 2022.

[13] Q. Kong, Y. Xu, and M. D. Plumbley, "Attention-based deep multiple instance learning for weakly supervised audio tagging," in *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 111–115, IEEE, 2018.

[14] K. Choi, J. Lee, and J. Nam, "Content-based music similarity with deep representation learning," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR 2017)*, 2017.

[15] B. Elizalde, S. Deshmukh, M. A. Ismail, and H. Wang, "Clap learning audio concepts from natural language supervision," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 1–5, June 2023.

[16] A. S. Koepke, A.-M. Oncescu, J. F. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries: A benchmark study," *IEEE Transactions on Multimedia*, vol. 25, p. 2675–2685, 2023.

[17] C. Donahue, J. McAuley, and M. Puckette, "Wavegan: A gan for raw audio synthesis," in *International Conference on Learning Representations*, 2018.

[18] J. Engel, A. Roberts, S. Dieleman, D. Askew, S. Oore, and D. Eck, "Disentangled representations of musical timbre with gans and vaes," *arXiv preprint arXiv:1911.08323*, 2019.

[19] H. Liu, Y. Yuan, X. Liu, X. Mei, Q. Kong, Q. Tian, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, "Audioldm 2: Learning holistic audio generation with self-supervised pretraining," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, p. 2871–2883, 2024.

[20] J. Xue, Y. Deng, Y. Gao, and Y. Li, "Auffusion: Leveraging the power of diffusion and large language models for text-to-audio generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, p. 4700–4712, 2024.

[21] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi, and N. Zeghidour, "Audiolm: A language modeling approach to audio generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, p. 2523–2533, 2023.

[22] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[23] J. Lu, C. Clark, R. Zellers, R. Mottaghi, and A. Kembhavi, "Unified-io: A unified model for vision, language, and audio," in *European Conference on Computer Vision*, pp. 525–542, Springer, 2022.

[24] Y. Chu, J. Xu, X. Zhou, Q. Yang, S. Zhang, Z. Yan, C. Zhou, and J. Zhou, "Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models," Dec. 2023. arXiv:2311.07919 [eess].

[25] C. Tang, W. Yu, G. Sun, X. Chen, T. Tan, W. Li, L. Lu, Z. Ma, and C. Zhang, "Salmonn: Towards generic hearing abilities for large language models," Apr. 2024. arXiv:2310.13289 [cs].

[26] S. Wu, H. Fei, L. Qu, W. Ji, and T.-S. Chua, "Next-gpt: Any-to-any multimodal llm," June 2024.

[27] A. Vyas, B. Shi, M. Le, A. Tjandra, Y.-C. Wu, B. Guo, J. Zhang, X. Zhang, R. Adkins, W. Ngan, J. Wang, I. Cruz, B. Akula, A. Akinyemi, B. Ellis, R. Moritz, Y. Yungster, A. Rakotoarison, L. Tan, C. Summers, C. Wood, J. Lane, M. Williamson, and W.-N. Hsu, "Audiobox: Unified audio generation with natural language prompts," Dec. 2023. arXiv:2312.15821 [cs].

[28] C. Lyu, M. Wu, L. Wang, X. Huang, B. Liu, Z. Du, S. Shi, and Z. Tu, "Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration," June 2023. arXiv:2306.09093 [cs].

[29] Y. Su, T. Lan, H. Li, J. Xu, Y. Wang, and D. Cai, "Pandagpt: One model to instruction-follow them all," May 2023. arXiv:2305.16355 [cs].

[30] Y. Gong, H. Luo, A. H. Liu, L. Karlinsky, and J. Glass, "Listen, think, and understand," Feb. 2024. arXiv:2305.10790 [eess].

[31] J. Lu, C. Clark, S. Lee, Z. Zhang, S. Khosla, R. Marten, D. Hoiem, and A. Kembhavi, "Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action," p. 26439–26455, 2024.

[32] R. Huang, M. Li, D. Yang, J. Shi, X. Chang, Z. Ye, Y. Wu, Z. Hong, J. Huang, J. Liu, Y. Ren, Z. Zhao, and S. Watanabe, "Audiogpt: Understanding and generating speech, music, sound, and talking head," Apr. 2023. arXiv:2304.12995 [cs].

[33] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang, "Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face," Dec. 2023. arXiv:2303.17580 [cs].

[34] S.-L. Wu, C. Donahue, S. Watanabe, and N. J. Bryan, "Music controlnet: Multiple time-varying controls for music generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, p. 2692–2703, 2024.

[35] D. Ghosal, N. Majumder, A. Mehrish, and S. Poria, "Text-to-audio generation using instruction-tuned llm and latent diffusion model," May 2023. arXiv:2304.13731 [eess].

[36] J. Huang, Y. Ren, R. Huang, D. Yang, Z. Ye, C. Zhang, J. Liu, X. Yin, Z. Ma, and Z. Zhao, "Make-an-audio 2: Temporal-enhanced text-to-audio generation," May 2023. arXiv:2305.18474 [cs].

[37] X. Liu, Z. Zhu, H. Liu, Y. Yuan, Q. Huang, M. Cui, J. Liang, Y. Cao, Q. Kong, M. D. Plumbley, and W. Wang, "Wavjourney: Compositional audio creation with large language models," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 33, p. 2830–2844, 2025.

[38] D. Yang, J. Tian, X. Tan, R. Huang, S. Liu, X. Chang, J. Shi, S. Zhao, J. Bian, Z. Zhao, X. Wu, and H. Meng, "Uniaudio: An audio foundation model toward universal audio generation," Dec. 2024. arXiv:2310.00704 [cs].

[39] S. Liu, A. S. Hussain, C. Sun, and Y. Shan, "Music understanding llama: Advancing text-to-music generation with question answering and captioning," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 286–290, Apr. 2024.

[40] J. Gardner, S. Durand, D. Stoller, and R. M. Bittner, "Llark: A multimodal instruction-following language model for music," June 2024. arXiv:2310.07160 [cs].

[41] D. Yu, K. Song, P. Lu, T. He, X. Tan, W. Ye, S. Zhang, and J. Bian, "Musicagent: An ai agent for music understanding and generation with large language models," Oct. 2023. arXiv:2310.11954 [cs].

[42] L. Zhuo, R. Yuan, J. Pan, Y. Ma, Y. LI, G. Zhang, S. Liu, R. Dannenberg, J. Fu, C. Lin, E. Benetos, W. Xue, and Y. Guo, "Lyricwhiz: Robust multilingual zero-shot lyrics transcription by whispering to chatgpt," July 2024. arXiv:2306.17103 [cs].

[43] R. Yuan, H. Lin, Y. Wang, Z. Tian, S. Wu, T. Shen, G. Zhang, Y. Wu, C. Liu, Z. Zhou, Z. Ma, L. Xue, Z. Wang, Q. Liu, T. Zheng, Y. Li, Y. Ma, Y. Liang, X. Chi, R. Liu, Z. Wang, P. Li, J. Wu, C. Lin, Q. Liu, T. Jiang, W. Huang, W. Chen, E. Benetos, J. Fu, G. Xia, R. Dannenberg, W. Xue, S. Kang, and Y. Guo, "Chatmusician: Understanding and generating music intrinsically with llm," Feb. 2024. arXiv:2402.16153 [cs].

[44] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, "Musiclm: Generating music from text," Jan. 2023. arXiv:2301.11325 [cs].

[45] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," Apr. 2020. arXiv:2005.00341 [eess].

[46] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation,"

[47] S. Liu, A. S. Hussain, Q. Wu, C. Sun, and Y. Shan, "M^{2}ugen: Multi-modal music understanding and generation with the power of large language models," Dec. 2024. arXiv:2311.11255 [cs].

[48] S. Ding, Z. Liu, X. Dong, P. Zhang, R. Qian, J. Huang, C. He, D. Lin, and J. Wang, "Songcomposer: A large language model for lyric and melody generation in song composition," May 2025. arXiv:2402.17645 [cs].

[49] Y. Zhang, A. Maezawa, G. Xia, K. Yamamoto, and S. Dixon, "Loop copilot: Conducting ai ensembles for music generation and iterative editing," Aug. 2024. arXiv:2310.12404 [cs].

[50] D. Zhang, S. Li, X. Zhang, J. Zhan, P. Wang, Y. Zhou, and X. Qiu, "Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities," May 2023. arXiv:2305.11000 [cs].

[51] P. K. Rubenstein, C. Asawaroengchai, D. D. Nguyen, A. Bapna, Z. Borsos, F. d. C. Quitry, P. Chen, D. E. Badawy, W. Han, E. Kharitonov, H. Muckenhirn, D. Padfield, J. Qin, D. Rozenberg, T. Sainath, J. Schalkwyk, M. Sharifi, M. T. Ramanovich, M. Tagliasacchi, A. Tudor, M. Velimirović, D. Vincent, J. Yu, Y. Wang, V. Zayats, N. Zeghidour, Y. Zhang, Z. Zhang, L. Zilka, and C. Frank, "Audiopalm: A large language model that can speak and listen," June 2023. arXiv:2306.12925 [cs].

[52] J. Wu, Y. Gaur, Z. Chen, L. Zhou, Y. Zhu, T. Wang, J. Li, S. Liu, B. Ren, L. Liu, and Y. Wu, "On decoder-only architecture for speech-to-text and large language model integration," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, p. 1–8, Dec. 2023.

[53] S. Wang, C.-H. Yang, J. Wu, and C. Zhang, "Can whisper perform speech-based in-context learning?," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 13421–13425, Apr. 2024.

[54] W. Yu, C. Tang, G. Sun, X. Chen, T. Tan, W. Li, L. Lu, Z. Ma, and C. Zhang, "Connecting speech encoder and large language model for asr," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 12637–12641, Apr. 2024.

[55] C. Wang, S. Chen, Y. Wu, Z. Zhang, L. Zhou, S. Liu, Z. Chen, Y. Liu, H. Wang, J. Li, L. He, S. Zhao, and F. Wei, "Neural codec language models are zero-shot text to speech synthesizers," Jan. 2023. arXiv:2301.02111 [cs].

[56] H. Hao, L. Zhou, S. Liu, J. Li, S. Hu, R. Wang, and F. Wei, "Boosting large language model for speech synthesis: An empirical study," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, p. 1–5, Apr. 2025.

[57] Z. Du, J. Wang, Q. Chen, Y. Chu, Z. Gao, Z. Li, K. Hu, X. Zhou, J. Xu, Z. Ma, W. Wang, S. Zheng, C. Zhou, Z. Yan, and S. Zhang, "Lauragpt: Listen, attend, understand, and regenerate audio with gpt," July 2024. arXiv:2310.04673 [cs].

[58] S. Kakouros, J. Šimko, M. Vainio, and A. Suni, "Investigating the utility of surprisal from large language models for speech synthesis prosody," June 2023. arXiv:2306.09814 [eess].

[59] Z. Liu, Z. Lai, Z. Gao, E. Cui, Z. Li, X. Zhu, L. Lu, Q. Chen, Y. Qiao, J. Dai, and W. Wang, "Controlllm: Augment language models with tools by searching on graphs," in *Computer Vision – ECCV 2024* (A. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, and G. Varol, eds.), (Cham), p. 89–105, Springer Nature Switzerland, 2025.

[60] C. Li, H. Chen, M. Yan, W. Shen, H. Xu, Z. Wu, Z. Zhang, W. Zhou, Y. Chen, C. Cheng, H. Shi, J. Zhang, F. Huang, and J. Zhou, "Modelscope-agent: Building your customizable agent system with open-source large language models," Sept. 2023. arXiv:2309.00986 [cs].

[61] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "Musdb18-hq - an uncompressed version of musdb18," Aug. 2019.

[62] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," Oct. 2021. arXiv:2106.09685 [cs].

[63] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, "Qlora: Efficient finetuning of quantized llms," *arXiv preprint arXiv:2305.14314*, 2023. arXiv:2305.14314.