AbletonBuddy: An AI-Powered Assistant for Ableton Live Control

Pratham Vadhulas
Advisor: Dr. Alexander Lerch
Fall 2025 Project Proposal

Georgia Tech · College of Design
Center for
Music Technology

Introduction

Overview

Georgia Tech · College of Design
Center for Music Technology

- **AbletonBuddy** is an AI-powered assistant that enables natural language control of Ableton Live.
- Uses large language models (LLMs) to understand user commands and translate them into OSC (Open Sound Control) commands.
- Provides an intuitive interface for controlling complex DAW operations through conversational interaction.
- Supports both real-time control and simulation mode for testing and development.

[System Architecture Diagram]

Introduction

Overview

- **AbletonBuddy** is an AI-powered assistant that enables natural language control of Ableton Live.
- Uses large language models (LLMs) to understand user commands and translate them into OSC (Open Sound Control) commands.
- Provides an intuitive interface for controlling complex DAW operations through conversational interaction.
- Supports both real-time control and simulation mode for testing and development.

[OSC Communication Flow]

Introduction
Overview

- **AbletonBuddy** is an AI-powered assistant that enables natural language control of Ableton Live.
- Uses large language models (LLMs) to understand user commands and translate them into OSC (Open Sound Control) commands.
- Provides an intuitive interface for controlling complex DAW operations through conversational interaction.
- Supports both real-time control and simulation mode for testing and development.

[User Interface Screenshot]

Introduction

Overview

- **AbletonBuddy** is an AI-powered assistant that enables natural language control of Ableton Live.
- Uses large language models (LLMs) to understand user commands and translate them into OSC (Open Sound Control) commands.
- Provides an intuitive interface for controlling complex DAW operations through conversational interaction.
- Supports both real-time control and simulation mode for testing and development.

[Simulation Mode Example]

Introduction
Key Features

- **Natural Language Interaction**: Control Ableton Live with simple, conversational commands
- **Comprehensive Control**: Manage tracks, scenes, clips, devices, and playback
- **Conversation Persistence**: All interactions saved to SQLite database for continuity
- **Real-time Streaming**: Watch the assistant work in real-time via streaming API
- **Modern Web Interface**: React-based chat UI with real-time updates
- **Extensible Architecture**: Easy to add new tools and capabilities

Introduction
Key Features

- **Natural Language Interaction**: Control Ableton Live with simple, conversational commands
- **Comprehensive Control**: Manage tracks, scenes, clips, devices, and playback
- **Conversation Persistence**: All interactions saved to SQLite database for continuity
- **Real-time Streaming**: Watch the assistant work in real-time via streaming API
- **Modern Web Interface**: React-based chat UI with real-time updates
- **Extensible Architecture**: Easy to add new tools and capabilities

Introduction

Key Features

- **Natural Language Interaction**: Control Ableton Live with simple, conversational commands
- **Comprehensive Control**: Manage tracks, scenes, clips, devices, and playback
- **Conversation Persistence**: All interactions saved to SQLite database for continuity
- **Real-time Streaming**: Watch the assistant work in real-time via streaming API
- **Modern Web Interface**: React-based chat UI with real-time updates
- **Extensible Architecture**: Easy to add new tools and capabilities

Introduction
Key Features

- **Natural Language Interaction**: Control Ableton Live with simple, conversational commands
- **Comprehensive Control**: Manage tracks, scenes, clips, devices, and playback
- **Conversation Persistence**: All interactions saved to SQLite database for continuity
- **Real-time Streaming**: Watch the assistant work in real-time via streaming API
- **Modern Web Interface**: React-based chat UI with real-time updates
- **Extensible Architecture**: Easy to add new tools and capabilities

Introduction

Key Features

- **Natural Language Interaction**: Control Ableton Live with simple, conversational commands
- **Comprehensive Control**: Manage tracks, scenes, clips, devices, and playback
- **Conversation Persistence**: All interactions saved to SQLite database for continuity
- **Real-time Streaming**: Watch the assistant work in real-time via streaming API
- **Modern Web Interface**: React-based chat UI with real-time updates
- **Extensible Architecture**: Easy to add new tools and capabilities

Introduction

Key Features

- **Natural Language Interaction**: Control Ableton Live with simple, conversational commands
- **Comprehensive Control**: Manage tracks, scenes, clips, devices, and playback
- **Conversation Persistence**: All interactions saved to SQLite database for continuity
- **Real-time Streaming**: Watch the assistant work in real-time via streaming API
- **Modern Web Interface**: React-based chat UI with real-time updates
- **Extensible Architecture**: Easy to add new tools and capabilities

Motivation

The Problem

- **Complex DAW Interfaces**: Ableton Live has extensive functionality but steep learning curve
- **Repetitive Tasks**: Many operations require multiple clicks and menu navigation
- **Accessibility Barriers**: Traditional GUI-based control limits accessibility
- **Workflow Disruption**: Switching between creative flow and technical operations

[Complex DAW Interface]

Motivation

The Problem

Georgia Tech · College of Design
Center for Music Technology

- **Complex DAW Interfaces**: Ableton Live has extensive functionality but steep learning curve
- **Repetitive Tasks**: Many operations require multiple clicks and menu navigation
- **Accessibility Barriers**: Traditional GUI-based control limits accessibility
- **Workflow Disruption**: Switching between creative flow and technical operations

[Task Complexity Visualization]

Motivation
The Problem

- **Complex DAW Interfaces**: Ableton Live has extensive functionality but steep learning curve
- **Repetitive Tasks**: Many operations require multiple clicks and menu navigation
- **Accessibility Barriers**: Traditional GUI-based control limits accessibility
- **Workflow Disruption**: Switching between creative flow and technical operations

[Accessibility Challenges]

Motivation

The Problem

- **Complex DAW Interfaces**: Ableton Live has extensive functionality but steep learning curve
- **Repetitive Tasks**: Many operations require multiple clicks and menu navigation
- **Accessibility Barriers**: Traditional GUI-based control limits accessibility
- **Workflow Disruption**: Switching between creative flow and technical operations

[Workflow Interruption]

Motivation

Why Natural Language?

- **Intuitive Interaction**: Users can express intent naturally without learning specific commands
- **Contextual Understanding**: LLMs can infer user intent from ambiguous requests
- **Learning Aid**: Helps users discover Ableton Live capabilities through conversation
- **Future-Proof**: Natural language interface adapts as LLM capabilities improve
- **Multimodal Potential**: Foundation for future audio-visual understanding and control

Motivation

Why Natural Language?

- **Intuitive Interaction**: Users can express intent naturally without learning specific commands
- **Contextual Understanding**: LLMs can infer user intent from ambiguous requests
- **Learning Aid**: Helps users discover Ableton Live capabilities through conversation
- **Future-Proof**: Natural language interface adapts as LLM capabilities improve
- **Multimodal Potential**: Foundation for future audio-visual understanding and control

Motivation

Why Natural Language?

- **Intuitive Interaction**: Users can express intent naturally without learning specific commands
- **Contextual Understanding**: LLMs can infer user intent from ambiguous requests
- **Learning Aid**: Helps users discover Ableton Live capabilities through conversation
- **Future-Proof**: Natural language interface adapts as LLM capabilities improve
- **Multimodal Potential**: Foundation for future audio-visual understanding and control

Motivation

Why Natural Language?

- **Intuitive Interaction**: Users can express intent naturally without learning specific commands
- **Contextual Understanding**: LLMs can infer user intent from ambiguous requests
- **Learning Aid**: Helps users discover Ableton Live capabilities through conversation
- **Future-Proof**: Natural language interface adapts as LLM capabilities improve
- **Multimodal Potential**: Foundation for future audio-visual understanding and control

Motivation

Why Natural Language?

- **Intuitive Interaction**: Users can express intent naturally without learning specific commands
- **Contextual Understanding**: LLMs can infer user intent from ambiguous requests
- **Learning Aid**: Helps users discover Ableton Live capabilities through conversation
- **Future-Proof**: Natural language interface adapts as LLM capabilities improve
- **Multimodal Potential**: Foundation for future audio-visual understanding and control

Motivation

Related Work

- **Co-creative AI Systems**: Systems that assist rather than replace human creativity **Louie˙Coenen˙Huang˙Terry˙Cai˙2020b**, **Tsiros˙Palladini˙2020**
- **AI in Music Production**: Adoption studies and workflow integration **Vanka˙Safi˙Rolland˙Fazekas˙2023**
- **Multimodal LLMs**: Audio-language models for music understanding **Chu˙Xu˙Zhou˙Yang˙Zhang˙Yan˙Zhou˙Zhou˙2023**, **Gardner˙Durand˙Stoller˙Bittner˙2024**
- **Agent Frameworks**: LLM-based agent systems for tool use **Li˙Chen˙Yan˙Shen˙Xu˙Wu˙Zhang˙Zhou˙Chen˙Cheng˙etal˙2023**, **Shen˙Song˙Tan˙Li˙Lu˙Zhuang˙2023**

Motivation

Related Work

- **Co-creative AI Systems**: Systems that assist rather than replace human creativity **Louie˙Coenen˙Huang˙Terry˙Cai˙2020b**, **Tsiros˙Palladini˙2020**
- **AI in Music Production**: Adoption studies and workflow integration **Vanka˙Safi˙Rolland˙Fazekas˙2023**
- **Multimodal LLMs**: Audio-language models for music understanding **Chu˙Xu˙Zhou˙Yang˙Zhang˙Yan˙Zhou˙Zhou˙2023**, **Gardner˙Durand˙Stoller˙Bittner˙2024**
- **Agent Frameworks**: LLM-based agent systems for tool use **Li˙Chen˙Yan˙Shen˙Xu˙Wu˙Zhang˙Zhou˙Chen˙Cheng˙etal˙2023**, **Shen˙Song˙Tan˙Li˙Lu˙Zhuang˙2023**

Motivation

Related Work

- **Co-creative AI Systems**: Systems that assist rather than replace human creativity **Louie˙Coenen˙Huang˙Terry˙Cai˙2020b**, **Tsiros˙Palladini˙2020**
- **AI in Music Production**: Adoption studies and workflow integration **Vanka˙Safi˙Rolland˙Fazekas˙2023**
- **Multimodal LLMs**: Audio-language models for music understanding **Chu˙Xu˙Zhou˙Yang˙Zhang˙Yan˙Zhou˙Zhou˙2023**, **Gardner˙Durand˙Stoller˙Bittner˙2024**
- **Agent Frameworks**: LLM-based agent systems for tool use **Li˙Chen˙Yan˙Shen˙Xu˙Wu˙Zhang˙Zhou˙Chen˙Cheng˙etal˙2023**, **Shen˙Song˙Tan˙Li˙Lu˙Zhuang˙2023**

Motivation

Related Work

- **Co-creative AI Systems**: Systems that assist rather than replace human creativity **Louie˙Coenen˙Huang˙Terry˙Cai˙2020b**, **Tsiros˙Palladini˙2020**
- **AI in Music Production**: Adoption studies and workflow integration **Vanka˙Safi˙Rolland˙Fazekas˙2023**
- **Multimodal LLMs**: Audio-language models for music understanding **Chu˙Xu˙Zhou˙Yang˙Zhang˙Yan˙Zhou˙Zhou˙2023**, **Gardner˙Durand˙Stoller˙Bittner˙2024**
- **Agent Frameworks**: LLM-based agent systems for tool use **Li˙Chen˙Yan˙Shen˙Xu˙Wu˙Zhang˙Zhou˙Chen˙Cheng˙etal˙2023**, **Shen˙Song˙Tan˙Li˙Lu˙Zhuang˙2023**

Methodology
System Architecture

- **LLM Backbone**: Uses pre-trained language models (e.g., Claude, GPT-4) via Marvin framework
- **Agent-Based Design**: Multi-agent system with specialized agents for different tasks
- **OSC Communication**: Open Sound Control protocol for real-time DAW control
- **Modular Tools**: Extensible tool system for different Ableton Live operations
- **Conversation Management**: Thread-based persistence with SQLite database

[System Architecture Diagram]

Methodology
System Architecture

- **LLM Backbone**: Uses pre-trained language models (e.g., Claude, GPT-4) via Marvin framework
- **Agent-Based Design**: Multi-agent system with specialized agents for different tasks
- **OSC Communication**: Open Sound Control protocol for real-time DAW control
- **Modular Tools**: Extensible tool system for different Ableton Live operations
- **Conversation Management**: Thread-based persistence with SQLite database

[Agent Structure]

Methodology
System Architecture

- **LLM Backbone**: Uses pre-trained language models (e.g., Claude, GPT-4) via Marvin framework
- **Agent-Based Design**: Multi-agent system with specialized agents for different tasks
- **OSC Communication**: Open Sound Control protocol for real-time DAW control
- **Modular Tools**: Extensible tool system for different Ableton Live operations
- **Conversation Management**: Thread-based persistence with SQLite database

[OSC Layer]

Methodology
System Architecture

- **LLM Backbone**: Uses pre-trained language models (e.g., Claude, GPT-4) via Marvin framework
- **Agent-Based Design**: Multi-agent system with specialized agents for different tasks
- **OSC Communication**: Open Sound Control protocol for real-time DAW control
- **Modular Tools**: Extensible tool system for different Ableton Live operations
- **Conversation Management**: Thread-based persistence with SQLite database

[Tool System]

Methodology

System Architecture

- **LLM Backbone**: Uses pre-trained language models (e.g., Claude, GPT-4) via Marvin framework
- **Agent-Based Design**: Multi-agent system with specialized agents for different tasks
- **OSC Communication**: Open Sound Control protocol for real-time DAW control
- **Modular Tools**: Extensible tool system for different Ableton Live operations
- **Conversation Management**: Thread-based persistence with SQLite database

[Database Schema]

Methodology
Agent Pipeline

1. **Input Classification**: Categorize user request (track, clip, device, etc.)
2. **Disambiguation**: Resolve ambiguous references (e.g., "the first track")
3. **Extraction**: Extract structured parameters from natural language
4. **Task Creation**: Generate executable tasks with specific instructions
5. **Execution**: Execute tasks via OSC commands to Ableton Live

[Classification Agent]

Methodology
Agent Pipeline

1. **Input Classification**: Categorize user request (track, clip, device, etc.)
2. **Disambiguation**: Resolve ambiguous references (e.g., "the first track")
3. **Extraction**: Extract structured parameters from natural language
4. **Task Creation**: Generate executable tasks with specific instructions
5. **Execution**: Execute tasks via OSC commands to Ableton Live

[Disambiguation Agent]

Methodology
Agent Pipeline

1. **Input Classification**: Categorize user request (track, clip, device, etc.)
2. **Disambiguation**: Resolve ambiguous references (e.g., "the first track")
3. **Extraction**: Extract structured parameters from natural language
4. **Task Creation**: Generate executable tasks with specific instructions
5. **Execution**: Execute tasks via OSC commands to Ableton Live

[Extraction Agent]

Methodology
Agent Pipeline

1. **Input Classification**: Categorize user request (track, clip, device, etc.)
2. **Disambiguation**: Resolve ambiguous references (e.g., "the first track")
3. **Extraction**: Extract structured parameters from natural language
4. **Task Creation**: Generate executable tasks with specific instructions
5. **Execution**: Execute tasks via OSC commands to Ableton Live

[Task Agent]

Methodology
Agent Pipeline

1. **Input Classification**: Categorize user request (track, clip, device, etc.)
2. **Disambiguation**: Resolve ambiguous references (e.g., "the first track")
3. **Extraction**: Extract structured parameters from natural language
4. **Task Creation**: Generate executable tasks with specific instructions
5. **Execution**: Execute tasks via OSC commands to Ableton Live

[Execution Flow]

Methodology
Key Components

- **Classification Agent**: Determines request category (song, track, clip, device, etc.)
- **Disambiguation Agent**: Resolves ambiguous references using context
- **Extraction Agent**: Extracts structured data from natural language
- **Task Agent**: Creates and executes sequences of operations
- **OSC Tools**: Modular tools for each Ableton Live operation type
- **Message Formatting**: Filters and formats agent messages for display

Methodology
Key Components

- **Classification Agent**: Determines request category (song, track, clip, device, etc.)
- **Disambiguation Agent**: Resolves ambiguous references using context
- **Extraction Agent**: Extracts structured data from natural language
- **Task Agent**: Creates and executes sequences of operations
- **OSC Tools**: Modular tools for each Ableton Live operation type
- **Message Formatting**: Filters and formats agent messages for display

Methodology
Key Components

- **Classification Agent**: Determines request category (song, track, clip, device, etc.)
- **Disambiguation Agent**: Resolves ambiguous references using context
- **Extraction Agent**: Extracts structured data from natural language
- **Task Agent**: Creates and executes sequences of operations
- **OSC Tools**: Modular tools for each Ableton Live operation type
- **Message Formatting**: Filters and formats agent messages for display

Methodology
Key Components

- **Classification Agent**: Determines request category (song, track, clip, device, etc.)
- **Disambiguation Agent**: Resolves ambiguous references using context
- **Extraction Agent**: Extracts structured data from natural language
- **Task Agent**: Creates and executes sequences of operations
- **OSC Tools**: Modular tools for each Ableton Live operation type
- **Message Formatting**: Filters and formats agent messages for display

Methodology
Key Components

- **Classification Agent**: Determines request category (song, track, clip, device, etc.)
- **Disambiguation Agent**: Resolves ambiguous references using context
- **Extraction Agent**: Extracts structured data from natural language
- **Task Agent**: Creates and executes sequences of operations
- **OSC Tools**: Modular tools for each Ableton Live operation type
- **Message Formatting**: Filters and formats agent messages for display

Methodology
Key Components

- **Classification Agent**: Determines request category (song, track, clip, device, etc.)
- **Disambiguation Agent**: Resolves ambiguous references using context
- **Extraction Agent**: Extracts structured data from natural language
- **Task Agent**: Creates and executes sequences of operations
- **OSC Tools**: Modular tools for each Ableton Live operation type
- **Message Formatting**: Filters and formats agent messages for display

Methodology
Supported Operations

Georgia Tech · College of Design
Center for Music Technology

Song Control

- Playback (play, stop, pause)
- Tempo and time signature
- Metronome control
- Session recording

Track Management

- Create/delete tracks
- Track naming and routing
- Volume and pan control

Clip Operations

- Create/delete clips
- Clip slot management
- Launch and stop clips

Device Control

- Add/remove devices
- Parameter adjustment
- Device management

Evaluation

Evaluation Framework

- **Functional Testing**: Verify correct OSC command generation and execution
- **Accuracy Metrics**: Measure success rate of command interpretation
- **User Studies**: Evaluate usability and user satisfaction
- **Performance Benchmarks**: Response time and system latency measurements
- **Error Analysis**: Categorize and analyze failure modes

Evaluation

Evaluation Framework

- **Functional Testing**: Verify correct OSC command generation and execution
- **Accuracy Metrics**: Measure success rate of command interpretation
- **User Studies**: Evaluate usability and user satisfaction
- **Performance Benchmarks**: Response time and system latency measurements
- **Error Analysis**: Categorize and analyze failure modes

Evaluation

Evaluation Framework

- **Functional Testing**: Verify correct OSC command generation and execution
- **Accuracy Metrics**: Measure success rate of command interpretation
- **User Studies**: Evaluate usability and user satisfaction
- **Performance Benchmarks**: Response time and system latency measurements
- **Error Analysis**: Categorize and analyze failure modes

Evaluation

Evaluation Framework

- **Functional Testing**: Verify correct OSC command generation and execution
- **Accuracy Metrics**: Measure success rate of command interpretation
- **User Studies**: Evaluate usability and user satisfaction
- **Performance Benchmarks**: Response time and system latency measurements
- **Error Analysis**: Categorize and analyze failure modes

Evaluation

Evaluation Framework

- **Functional Testing**: Verify correct OSC command generation and execution
- **Accuracy Metrics**: Measure success rate of command interpretation
- **User Studies**: Evaluate usability and user satisfaction
- **Performance Benchmarks**: Response time and system latency measurements
- **Error Analysis**: Categorize and analyze failure modes

Evaluation
Evaluation Metrics

Command Accuracy

- Correct interpretation rate
- Parameter extraction accuracy
- Disambiguation success rate

User Experience

- Task completion time
- User satisfaction scores
- Error recovery effectiveness

[Accuracy Metrics]

Evaluation

Evaluation Metrics

Command Accuracy

- Correct interpretation rate
- Parameter extraction accuracy
- Disambiguation success rate

User Experience

- Task completion time
- User satisfaction scores
- Error recovery effectiveness

[User Study Results]

Evaluation

Evaluation Metrics

Command Accuracy

- Correct interpretation rate
- Parameter extraction accuracy
- Disambiguation success rate

User Experience

- Task completion time
- User satisfaction scores
- Error recovery effectiveness

[Performance Benchmarks]

Evaluation
Test Scenarios

- **Basic Operations**: Simple commands (play, stop, create track)
- **Complex Tasks**: Multi-step operations (create track, add device, adjust parameters)
- **Ambiguous Requests**: Commands requiring disambiguation
- **Error Handling**: Invalid commands and edge cases
- **Conversation Continuity**: Multi-turn interactions and context retention

Evaluation

Test Scenarios

- **Basic Operations**: Simple commands (play, stop, create track)
- **Complex Tasks**: Multi-step operations (create track, add device, adjust parameters)
- **Ambiguous Requests**: Commands requiring disambiguation
- **Error Handling**: Invalid commands and edge cases
- **Conversation Continuity**: Multi-turn interactions and context retention

Evaluation

Test Scenarios

- **Basic Operations**: Simple commands (play, stop, create track)
- **Complex Tasks**: Multi-step operations (create track, add device, adjust parameters)
- **Ambiguous Requests**: Commands requiring disambiguation
- **Error Handling**: Invalid commands and edge cases
- **Conversation Continuity**: Multi-turn interactions and context retention

Evaluation

Test Scenarios

- **Basic Operations**: Simple commands (play, stop, create track)
- **Complex Tasks**: Multi-step operations (create track, add device, adjust parameters)
- **Ambiguous Requests**: Commands requiring disambiguation
- **Error Handling**: Invalid commands and edge cases
- **Conversation Continuity**: Multi-turn interactions and context retention

Evaluation

Test Scenarios

- **Basic Operations**: Simple commands (play, stop, create track)
- **Complex Tasks**: Multi-step operations (create track, add device, adjust parameters)
- **Ambiguous Requests**: Commands requiring disambiguation
- **Error Handling**: Invalid commands and edge cases
- **Conversation Continuity**: Multi-turn interactions and context retention

Results
System Performance

- **Command Interpretation**: High accuracy on common operations
- **Response Time**: Real-time interaction with low latency
- **Reliability**: Stable operation across extended sessions
- **Extensibility**: Easy addition of new tools and capabilities

[Accuracy Results]

Results
System Performance

- **Command Interpretation**: High accuracy on common operations
- **Response Time**: Real-time interaction with low latency
- **Reliability**: Stable operation across extended sessions
- **Extensibility**: Easy addition of new tools and capabilities

[Latency Measurements]

Results
System Performance

- **Command Interpretation**: High accuracy on common operations
- **Response Time**: Real-time interaction with low latency
- **Reliability**: Stable operation across extended sessions
- **Extensibility**: Easy addition of new tools and capabilities

[Stability Metrics]

Results
System Performance

- **Command Interpretation**: High accuracy on common operations
- **Response Time**: Real-time interaction with low latency
- **Reliability**: Stable operation across extended sessions
- **Extensibility**: Easy addition of new tools and capabilities

[Extension Examples]

Results
Key Achievements

- **Natural Language Interface**: Successfully implemented conversational control of Ableton Live
- **Comprehensive Coverage**: Support for major Ableton Live operations
- **User-Friendly Design**: Intuitive web interface with real-time feedback
- **Production-Ready**: Robust error handling and simulation mode for development
- **Extensible Architecture**: Foundation for future enhancements

Results
Key Achievements

- **Natural Language Interface**: Successfully implemented conversational control of Ableton Live
- **Comprehensive Coverage**: Support for major Ableton Live operations
- **User-Friendly Design**: Intuitive web interface with real-time feedback
- **Production-Ready**: Robust error handling and simulation mode for development
- **Extensible Architecture**: Foundation for future enhancements

Results
Key Achievements

- **Natural Language Interface**: Successfully implemented conversational control of Ableton Live
- **Comprehensive Coverage**: Support for major Ableton Live operations
- **User-Friendly Design**: Intuitive web interface with real-time feedback
- **Production-Ready**: Robust error handling and simulation mode for development
- **Extensible Architecture**: Foundation for future enhancements

Results
Key Achievements

- **Natural Language Interface**: Successfully implemented conversational control of Ableton Live
- **Comprehensive Coverage**: Support for major Ableton Live operations
- **User-Friendly Design**: Intuitive web interface with real-time feedback
- **Production-Ready**: Robust error handling and simulation mode for development
- **Extensible Architecture**: Foundation for future enhancements

Results
Key Achievements

- **Natural Language Interface**: Successfully implemented conversational control of Ableton Live
- **Comprehensive Coverage**: Support for major Ableton Live operations
- **User-Friendly Design**: Intuitive web interface with real-time feedback
- **Production-Ready**: Robust error handling and simulation mode for development
- **Extensible Architecture**: Foundation for future enhancements

Results
Example Interactions

Georgia Tech · College of Design
Center for Music Technology

Simple Commands

- "Play the song"
- "Create a new MIDI track"
- "Set tempo to 120 BPM"

Complex Tasks

- "Create a track, add a reverb device, and set the mix to 50%"
- "Duplicate the first scene and rename it"
- "Stop all clips and reset the tempo"

Results
Future Directions

- **Enhanced Understanding**: Better handling of complex, multi-step requests
- **Audio Analysis**: Integration of audio understanding for mix critique
- **Learning Capabilities**: System learns from user preferences and patterns
- **Multimodal Input**: Support for audio and visual inputs
- **Collaborative Features**: Multi-user support and shared sessions

Results
Future Directions

- **Enhanced Understanding**: Better handling of complex, multi-step requests
- **Audio Analysis**: Integration of audio understanding for mix critique
- **Learning Capabilities**: System learns from user preferences and patterns
- **Multimodal Input**: Support for audio and visual inputs
- **Collaborative Features**: Multi-user support and shared sessions

Results
Future Directions

- **Enhanced Understanding**: Better handling of complex, multi-step requests
- **Audio Analysis**: Integration of audio understanding for mix critique
- **Learning Capabilities**: System learns from user preferences and patterns
- **Multimodal Input**: Support for audio and visual inputs
- **Collaborative Features**: Multi-user support and shared sessions

Results
Future Directions

- **Enhanced Understanding**: Better handling of complex, multi-step requests
- **Audio Analysis**: Integration of audio understanding for mix critique
- **Learning Capabilities**: System learns from user preferences and patterns
- **Multimodal Input**: Support for audio and visual inputs
- **Collaborative Features**: Multi-user support and shared sessions

Results
Future Directions

- **Enhanced Understanding**: Better handling of complex, multi-step requests
- **Audio Analysis**: Integration of audio understanding for mix critique
- **Learning Capabilities**: System learns from user preferences and patterns
- **Multimodal Input**: Support for audio and visual inputs
- **Collaborative Features**: Multi-user support and shared sessions

references