

Anhang B

Die eigene Audio-KI: Finetuning von ACE-Step 1.5

In der modernen Unternehmenskommunikation ist Audio weit mehr als nur Begleitwerk. Es ist ein integraler Bestandteil der Markenidentität. Während die visuelle KI mit Bild und Video bereits fest in Workflows integriert ist, bietet die generative Audio-KI nun die Möglichkeit, eine konsistente »Sonic Identity«, also die klangliche Identität Ihres Unternehmens, inhouse zu gestalten.

Dieses Kapitel adressiert den professionellen Bedarf an maßgeschneiderten Audio-Lösungen. Wir verlassen die Ebene der generischen Musik-Prompts und nutzen ACE-Step 1.5, um durch gezieltes Finetuning (also die Erstellung eines LoRA-Adapters) sicherzustellen, dass die KI nicht nur Musik produziert, sondern die spezifische Klang-DNA Ihres Unternehmens oder Projekts reproduziert.

Warum setzen Unternehmen auf eigene Audio-Modelle?

- **Konsistenz:** Ein einmal trainiertes LoRA garantiert, dass alle Audio-Assets – vom Trainingsvideo bis zum Messeauftritt – einer einheitlichen akustischen Handschrift folgen.
- **Rechtliche Autonomie:** Durch das Training auf eigenen oder entsprechend lizenzierten Audiodaten minimieren Sie das Risiko von Urheberrechtsverletzungen und entziehen sich der Abhängigkeit von restriktiven Stock-Plattformen für Musik.
- **Skalierbarkeit:** Die API-Anbindung ermöglicht die automatisierte Vertonung von Content Ihrer Fachabteilung in Echtzeit. Denken Sie z. B. an GenAI-Agenten, die zukünftig eigenständig Videoinhalte erstellen und diese so auch mit Musik unterlegen können.

Hardware-Anforderungen im professionellen Kontext

Das Finetuning von Audio-Modellen ist rechenintensiv. Für eine produktive Umgebung ist eine NVIDIA-GPU mit mindestens 24 GB VRAM (z. B. eine RTX 4090 oder RTX A6000) der De-facto-Standard. Dies sichert nicht nur die Lauffähigkeit, sondern reduziert die Trainingszeit auf ein wirtschaftlich vertretbares Maß.

B.1 Installation: Die Audio-Werkstatt einrichten

Bevor Sie mit dem Finetuning beginnen können, müssen Sie die Software-Umgebung aufbauen. ACE-Step 1.5 ist erfreulicherweise sehr flexibel: Während für das Finetuning eine NVIDIA-GPU (CUDA) dringend empfohlen wird, unterstützt die Software für das reine Inferencing sogar Apple-Silicon-(MPS-), AMD-(ROCm-) oder Intel-Beschleuniger.

B.1.1 Schritt 1: Das Fundament mit uv

Für den produktiven Einsatz setzen wir auf eine isolierte, performante Installation. Wir nutzen hierfür erneut den Paketmanager `uv`, um Konflikte in der Python-Umgebung zu vermeiden. Sie sollten ACE-Step 1.5 auf Ihrem GPU-Server einrichten, um ausreichend Rechenleistung zur Verfügung zu haben. Sollten Sie `uv` noch nicht auf diesem installiert haben, holen Sie dies mit folgendem Befehl nach:

Befehl: `curl -LsSf https://astral.sh/uv/install.sh | sh`

B.1.2 Schritt 2: Quellcode beziehen und Umgebung synchronisieren

Sie klonen nun das offizielle Repository von GitHub und lassen `uv` die gesamte Umgebung inklusive aller notwendigen Bibliotheken (wie PyTorch, Audio-Codecs etc.) automatisch aufbauen:

Befehl: `git clone https://github.com/ACE-Step/ACE-Step-1.5.git`

Befehl: `cd ACE-Step-1.5`

Jetzt kommt die Synchronisationslogik von `uv` zum Einsatz. Der Befehl `sync` liest die Projektkonfiguration und installiert alles exakt so, wie es die Entwickler vorgesehen haben:

Befehl: `uv sync`

Je nach Internetanbindung und Geschwindigkeit kann die Einrichtung der ACE-Step-Umgebung bis zu 20 Minuten dauern.

B.1.3 Schritt 3: Der erste Start und Modell-Download

ACE-Step ist sehr benutzerfreundlich gestaltet. Beim ersten Start erkennt das System, dass die Modellgewichte noch fehlen, und lädt diese automatisch von Hugging Face herunter. Sie haben zwei Möglichkeiten, das System zu betreiben:

- die *Gradio* –Weboberfläche für die manuelle Bedienung und erste Tests,
- die *REST-API* für die Integration in eigene Workflows und Unternehmenssoftware.

Beginnen wir mit der Gradio-Weboberfläche. Sie ist ideal für die erste Inspektion und manuelle Tests der Audio-Generierung. Denken Sie hier daran, einen freien Port mit dem Zu-

satz `--port` mit zu übergeben, damit nicht der Standard-Port 7860 von Gradio verwendet wird, der eventuell bereits belegt ist. Mit dem Zusatz `--server-name 0.0.0.0` schaffen Sie es, dass die Oberfläche aus Ihrem Intranet heraus erreichbar ist.

Befehl: `uv run acestep --port 8083 --server-name 0.0.0.0`

Jetzt sollten Sie den *ACE-Step V1.5 Playground* öffnen können, wenn Sie folgende Adresse im Browser aufrufen: `http://[Ihre IP-Adresse]:8083`

Dann sollten Sie von der ACE-Step-V1.5-Playground-Weboberfläche begrüßt werden, die Sie in Abbildung B.1 sehen.

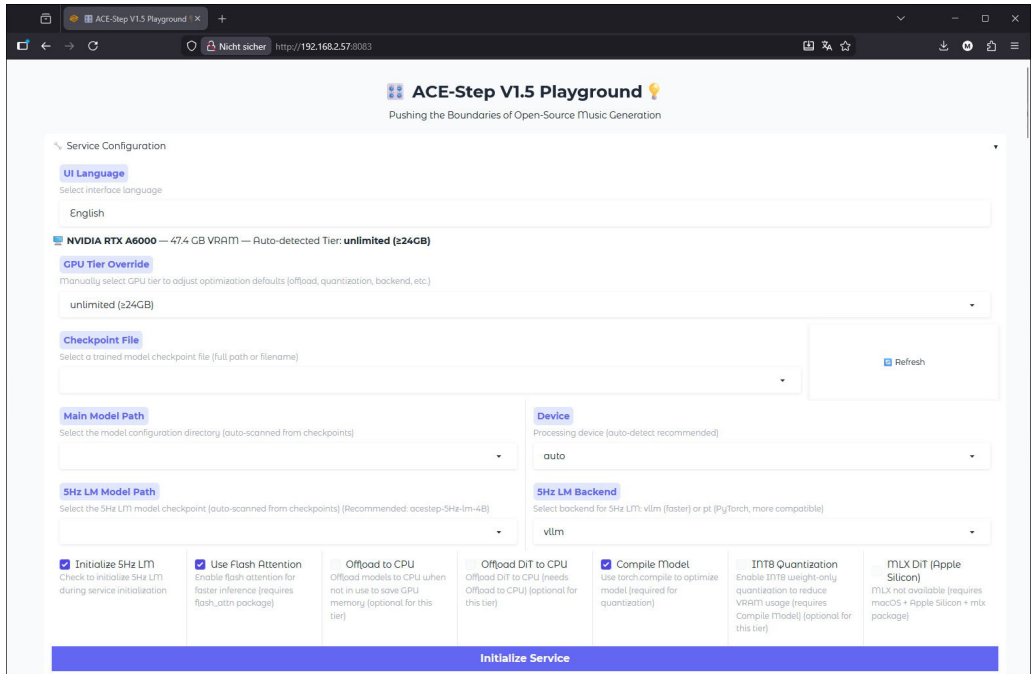


Abbildung B.1: ACE-Step V1.5 Playground – die Weboberfläche

Allerdings sehen Sie jetzt noch kein einziges ACE-Step-1.5-Modell, da noch keine Modelle heruntergeladen wurden. Um Modelle herunterzuladen, gibt es die folgenden Befehle, die Sie im Terminal-Fenster ausführen müssen.

Wenn Sie die Standardmodelle (*turbo* + *1.7B LM*) des ACE-Step-1.5-Projekts herunterladen wollen, dann können Sie das mit dem jetzt aufgeführten Befehl machen:

Befehl: `uv run acestep-download`

Wenn Sie ausreichend Platz auf Ihrem Rechner haben, empfehle ich Ihnen, einfach alle Modelle herunterzuladen. Dann führen Sie den folgenden Befehl aus:

Befehl: `uv run acestep-download --all`

Möchten Sie nicht alle ACE-Step-1.5-Modelle herunterladen, sondern lediglich bestimmte, dann können Sie die Auswahl der Modelle wie folgt festlegen und mit den gezeigten Beispielfehlen exakt diese Modelle herunterladen:

Befehl: `uv run acestep-download --model acestep-v15-base`

Befehl: `uv run acestep-download --model acestep-5Hz-lm-0.6B`

Wenn Sie sich alle Modelle anzeigen lassen wollen, die Sie bereits heruntergeladen haben, dann verwenden Sie den folgenden Befehl:

Befehl: `uv run acestep-download --list`

Sie sollten an diesem Punkt eine Auswahl oder eben alle ACE-Step-1.5-Modelle heruntergeladen haben.

Wenn Sie in den folgenden Abschnitten mit ACE-Step 1.5 arbeiten, werden Sie sich primär in diesem Dashboard als Anwender-Oberfläche bewegen. Dennoch möchte ich noch die REST-API-Schnittstelle erwähnen und nachfolgend kurz zeigen. Mit ihr können Sie die Musikproduktion in Ihre Workflows automatisch einbinden und starten.

Die Programmierschnittstelle (REST-API)

Wenn Sie ACE-Step in Ihre eigene Unternehmenssoftware oder in automatisierte Workflows integrieren möchten, dann starten Sie ACE-Step mit dem folgenden Aufruf:

Befehl: `uv run acestep-api`

Hinweis: Achten Sie beim ersten Start auf die Terminal-Ausgabe. Die Modelle werden einmalig geladen und stehen danach komplett offline zur Verfügung. Ihre Kompositionen verlassen somit nie Ihren Server.

B.2 ACE-Step-1.5-Finetuning: Technisches Verfahren zur klanglichen Adaption

Nachdem wir die Infrastruktur bereitgestellt haben, widmen wir uns dem Kernprozess: der projektspezifischen Anpassung, also dem Finetuning des Basismodells. Hierbei kommt die LoRA-Technik (*Low-Rank Adaptation*) zum Einsatz, die Sie bereits bei der Bildgenerierung kennengelernt haben. Anstatt das gesamte Modell ressourcenintensiv neu zu trainieren, applizieren wir eine hocheffiziente Wissensschicht. Diese Schicht extrahiert die spezifischen Nuancen Ihrer bereitgestellten Audiodaten und integriert sie in den Generierungsprozess.

B.2.1 Schritt 1: Kuratierung und Aufbereitung des Datasets

Die Qualität eines Finetunings korreliert direkt mit der Qualität der Trainingsdaten. In einem souveränen Unternehmensumfeld nutzen Sie hierfür eigene Audio-Assets, z. B. bestehende Sound-Identity-Elemente, lizenzierte Instrumentals oder spezifische atmo-

sphärische Aufnahmen. Ich empfehle Ihnen sehr, die folgende Anleitung initial einmal zu lesen bzw. diese offen zu haben, wenn Sie gleich der Erklärung folgen, die ich bewusst knapp gehalten habe, da die Entwicklung der KI-Werkzeuge rasant voranschreitet:

<https://github.com/ace-step/ACE-Step-1.5/blob/main/docs/en/Tutorial.md>

Jetzt folgt die knappe Anleitung zum Einstieg in das ACE-Step-Dashboard:

1. **Datenaggregation:** Erstellen Sie ein lokales Verzeichnis (z. B. `~/audio_training`) und hinterlegen Sie Ihre Audiodaten im WAV- oder MP3-Format. Achten Sie auf eine einheitliche Qualität der Quelldateien.
2. **Ingestion und Analyse:** Navigieren Sie im ACE-Step-Dashboard zum Reiter **Dataset Builder**. Hinterlegen Sie unter **Audio Directory Path** den Pfad zu Ihren Quelldateien und initiieren Sie den Scan-Vorgang mit **SCAN**. Das System erfasst nun die Metadaten Ihrer Dateien.
3. **Automatisierte Annotation (Auto-Labeling):** Ein erfolgreiches Training erfordert präzise Deskriptoren (*Tags*). ACE-Step nutzt hierfür eine automatisierte Audio-Analyse. Durch Klick auf **Auto-Label All** generiert das System Beschreibungen zu Rhythmik, Stimmung und Instrumentierung.

Professional-Tipp: Validieren Sie diese Tags manuell. Eine präzise Terminologie (wie z. B. *Cinematic*, *Minimalist*, *Tech-Focus*) ist entscheidend, um das Modell später über Prompts exakt steuern zu können.
4. **Datenaufbereitung:** Speichern Sie das Dataset über **Save Dataset** und führen Sie mittels **Preprocess** die Konvertierung in Tensoren durch. Dieser Schritt optimiert die Daten für den Zugriff durch die GPU während des Trainings.

B.2.2 Schritt 2: Konfiguration des Trainingslaufs

Wechseln Sie zum Reiter **Train Lora**. Hier definieren Sie die Parameter für die Modell-Adaption:

- **Model Selection:** Als Basis dient in der Regel das performante Modell *acestep-v15-turbo*.
- **Dataset Path:** Laden Sie das zuvor prozessierte Dataset.
- **Output Directory:** Definieren Sie einen Zielpfad (Standard: `./lora_output`).
- **Finetuning:** Starten Sie den Vorgang mit **Start Training**.

Während des Finetuning-Prozesses überwachen Sie im **Training Log** die *Loss-Kurve*. Technisch gesehen, minimiert die KI hierbei die Differenz zwischen ihrer eigenen Generierung und Ihren Trainingsvorgaben. Das Ziel ist ein stetig sinkender Loss-Wert. Ein Training mit ca. 20 Dateien und 1.000 Epochen beansprucht auf einer NVIDIA RTX A6000 etwa fünf Stunden. Sie sollten nach wenigen Minuten eine ähnliche Ansicht im ACE-Step 1.5 Dashboard sehen, wie sie hier in Abbildung B.2 gezeigt ist.

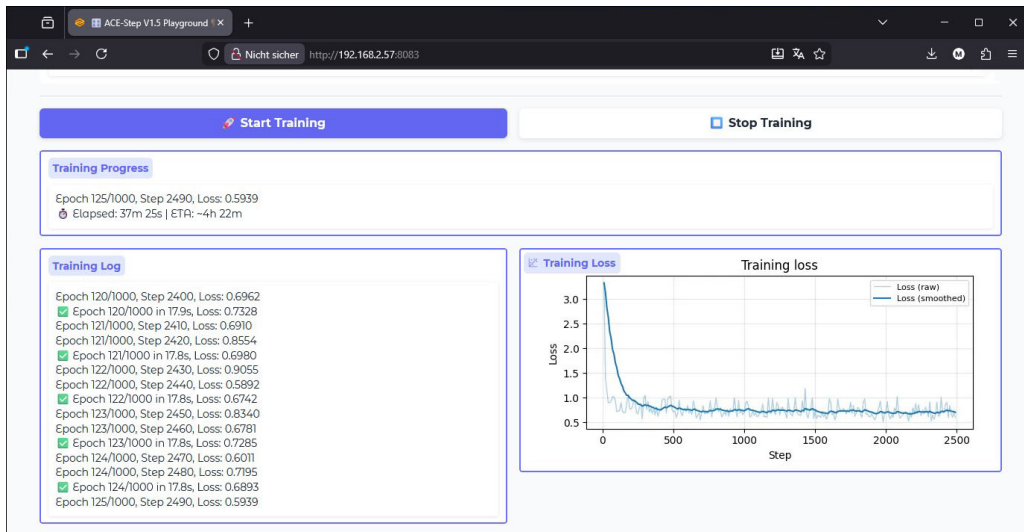


Abbildung B.2: ACE-Step V1.5 Playground – Trainingslauf

B.2.3 Schritt 3: Validierung und Deployment des LoRA-Adapters

Nach Abschluss des Trainings muss der Adapter für die Produktion exportiert werden über **Export LoRA**. Um das Ergebnis zu verifizieren, kehren wir in den *Playground* zurück:

1. **Adapter laden:** Hinterlegen Sie im Bereich **Lora Adapter** den Pfad zur generierten *.safetensors*-Datei und aktivieren Sie diese via **Load LoRA**.
2. **Prompt-Engineering:** Nutzen Sie die Deskriptoren aus Ihrem Trainingsdatensatz (einzusehen in der JSON-Datei im Verzeichnis */datasets/*), um eine **Song Description** zu erstellen.
3. **Inferenz:** Wenn Sie auf **Generate Music** klicken, erzeugt das System nun ein neues Werk, das die klanglichen Merkmale Ihrer Unternehmens-DNA (also Ihrer für das Fine-tuning bereitgestellten Audiodateien) trägt. Die Resultate stehen als MP3-Dateien für die weitere Verwendung zur Verfügung.

Das *ACE-Step 1.5 Dashboard* dient hierbei als Experten-Interface. Sobald ein LoRA-Adapter Ihre Qualitätsstandards erfüllt, sollte dieser in standardisierte Workflows (z. B. via ComfyUI oder API) überführt werden, um eine breitere Anwendung im Unternehmen zu ermöglichen.

In Abbildung B.3 sehen Sie das Ergebnis einer von mir generierten Musik unter Verwendung eines eigenen LoRA-Adapters, den ich zuvor trainiert habe. Sie können sich auch direkt im ACE-Step 1.5 Dashboard die generierten Musikstücke anhören und sie im Bereich **Batch Results & Generation Details** als MP3-Dateien lokal herunterladen.

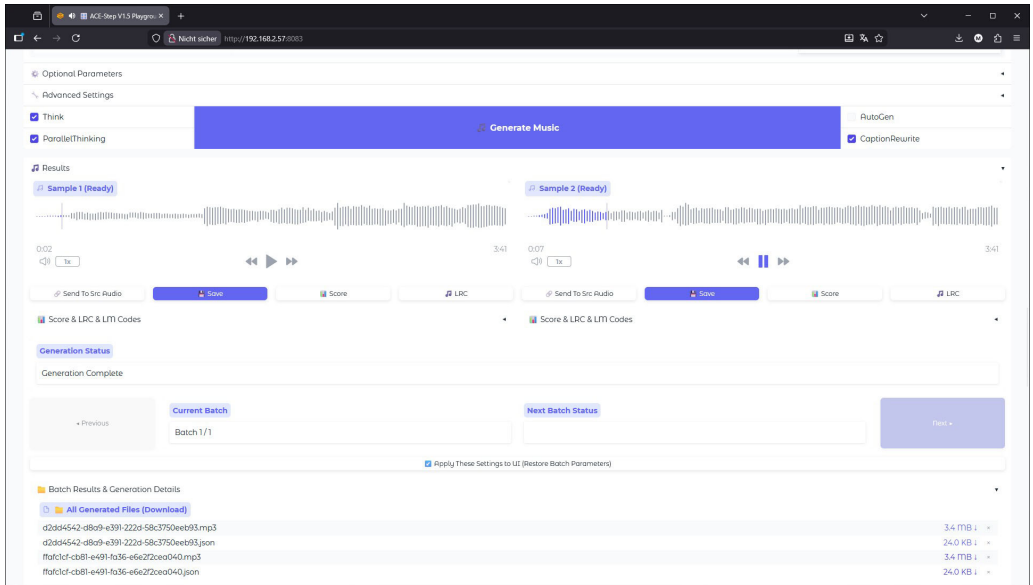


Abbildung B.3: ACE-Step V1.5 Playground – generierte Musik

B.3 Operationalisierung: Autostart und Systemintegration

Um ACE-Step 1.5 als zuverlässigen Dienst in der IT-Infrastruktur zu etablieren, integrieren wir die Anwendung via `systemd` in das Betriebssystem. Dies gewährleistet eine hohe Verfügbarkeit nach Systemneustarts oder Fehlern.

B.3.1 Schritt 1: Erstellung der Unit-Datei

Sie legen jetzt im Terminal-Fenster eine neue Konfigurationsdatei an. Auch hier nutzen wir wieder den bewährten Bash-Wrapper, um sicherzustellen, dass `uv` und alle Pfade in der spezialisierten `systemd`-Umgebung korrekt erkannt werden. Damit Sie es leichter haben und nicht so viel aus dem Buch abtippen müssen, finden Sie die `acestep.service`-Datei im Verzeichnis `ACE-Step_1-5` der ZIP-Datei, die es begleitend zum Buch beim Verlag zum Download gibt.

Befehl: `sudo nano /etc/systemd/system/acestep.service`

Kopieren Sie den Inhalt dieser bereitgestellten Dateien in die gerade frisch angelegte Datei auf Ihrem System. Achten Sie darauf, den Benutzernamen (hier den Platzhalter `<Ihr Benutzername>`) und die Pfade an Ihre tatsächliche Umgebung anzupassen.

Wenn Sie alle Änderungen vorgenommen haben, speichern Sie sie ab.

B.3.2 Schritt 2: Dienst aktivieren und starten

Nachdem die Datei gespeichert ist, müssen Sie `systemd` noch anweisen, die neue Konfiguration einzulesen und den Dienst für den automatischen Start scharfzuschalten. Dazu folgt jetzt eine Reihe von Befehlen, die Sie bitte einen nach dem anderen ausführen.

Wenn Sie eine Änderung an einem Service vorgenommen haben, müssen die Services neu geladen werden. Das machen Sie mit dem folgenden Befehl:

Befehl: `sudo systemctl daemon-reload`

Damit der Service aktiv ist, müssen Sie den folgenden Befehl ausführen. Jetzt startet ACE-Step 1.5 automatisch nach jedem Neustart Ihres Servers.

Befehl: `sudo systemctl enable acestep.service`

Jetzt starten Sie den ACE-Step-1.5-Service noch einmal von Hand:

Befehl: `sudo systemctl start acestep.service`

B.3.3 Schritt 3: Erfolgskontrolle

Wichtig sind auch immer die Befehle, mit denen Sie genauer unter die Werkbank schauen können, um zu prüfen, was da los ist. Wenn Sie also prüfen möchten, ob Ihre Audio-KI nun im Hintergrund »musiziert«, prüfen Sie das mit dem `status`-Befehl oder durch einen Blick in die Log-Datei:

Befehl: `sudo systemctl status acestep.service`

Wenn Sie die Logs lesen möchten oder live die Logs verfolgen wollen, nutzen Sie diesen Befehl:

Befehl: `journalctl -u acestep.service -f`

Wenn Sie jetzt wieder die URL aufrufen, über die Sie auf das ACE-Step 1.5 Dashboard zugreifen, sollte dieses wieder online sein.

B.4 Compliance und Strategie: KI-Audio im rechtlichen Raum

Der Einsatz von KI-generierter Musik im kommerziellen Sektor erfordert eine differenzierte Betrachtung der aktuellen Rechtslage – soweit ich das als Laie beurteilen kann (Stand: Frühjahr 2026).

Nach aktueller EU-Rechtsprechung genießen rein KI-generierte Werke oft keinen Urheberrechtsschutz, da die menschliche Schöpfungshöhe fehlt. Für Unternehmen bedeutet das Folgendes:

1. **Schutzfähigkeit:** KI-Musik ist ein effizientes Werkzeug, aber schwer als geistiges Eigentum gegen Nachahmer zu schützen.

2. **Hybrid-Workflow:** Im professionellen Umfeld sollte die KI als »Co-Komponist« dienen. Die finale Bearbeitung durch einen Sound-Engineer sichert nicht nur die Qualität, sondern erhöht die Chancen auf rechtlichen Schutz des Gesamtwerks.
3. **Transparency Act:** Beachten Sie die Kennzeichnungspflichten gemäß des *EU AI Acts* für KI-generierte Inhalte im öffentlichen Raum.

Fazit für Entscheider: ACE-Step 1.5 ist kein Ersatz für kreative Visionen, sondern ein Hochleistungswerkzeug zur Skalierung von Audio-Content. Souveränität bedeutet hier, die Kontrolle über den Trainingsdatensatz zu behalten und die KI als Werkbank für den eigenen, unverwechselbaren Corporate Sound zu nutzen.