Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

SP Signal Processing

Vadim Titov, Henning Möllers, Leon Mannweiler

Multichannel Speech Enhancement with Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN)

Multichannel Speech Enhancement

Universität Hamburg
Signal Processing (SP)

December 1, 2022

Gliederung

1. Überblick
2. Datengenerierung
3. Datenset
4. Neuronales Netz
5. Evaluation
6. Ausblick

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

SP
Signal Processing

Überblick

Überblick

- Ziel: Mehrkanälige Sprachverbesserung / Rauschunterdrückung.
- Methode: Neuronales Netz statt klassiche Beamformer.
- Motivation: Perspektive auf Verbesserung gegenüber klassischen Methoden.

Vorgehen

- Datengenerierung (Henning)
- Datenverarbeitung (Leon)
- Neuronales Netz (Vadim)
- Evaluation (Leon)

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

SP Signal Processing

Datengenerierung

Datengenerierung - Trainingsziel

Wir wollen ein Signal verstärken, welches:

- aus einem **relativ zum Mic immer gleichen Winkel** kommt
  $\rightarrow$ Entsprechende Rotation des Mics nötig, da absoluter Winkel des Mics beliebig

- von einer **beliebigen Quellposition** ausgesendet wird
- durch mehrere andere Signale an beliebiger Position gestört wird

$\Rightarrow$ Unser Netz soll darauf trainiert werden, **das Signal aus einem relativ zum Mic bestimmten Winkel unter variablen Umständen** (Position/Frequenzen Störquellen, Mic-Abstand zur Wand usw.) bestmöglich zu verstärken.

Datengenerierung - Trainingsziel

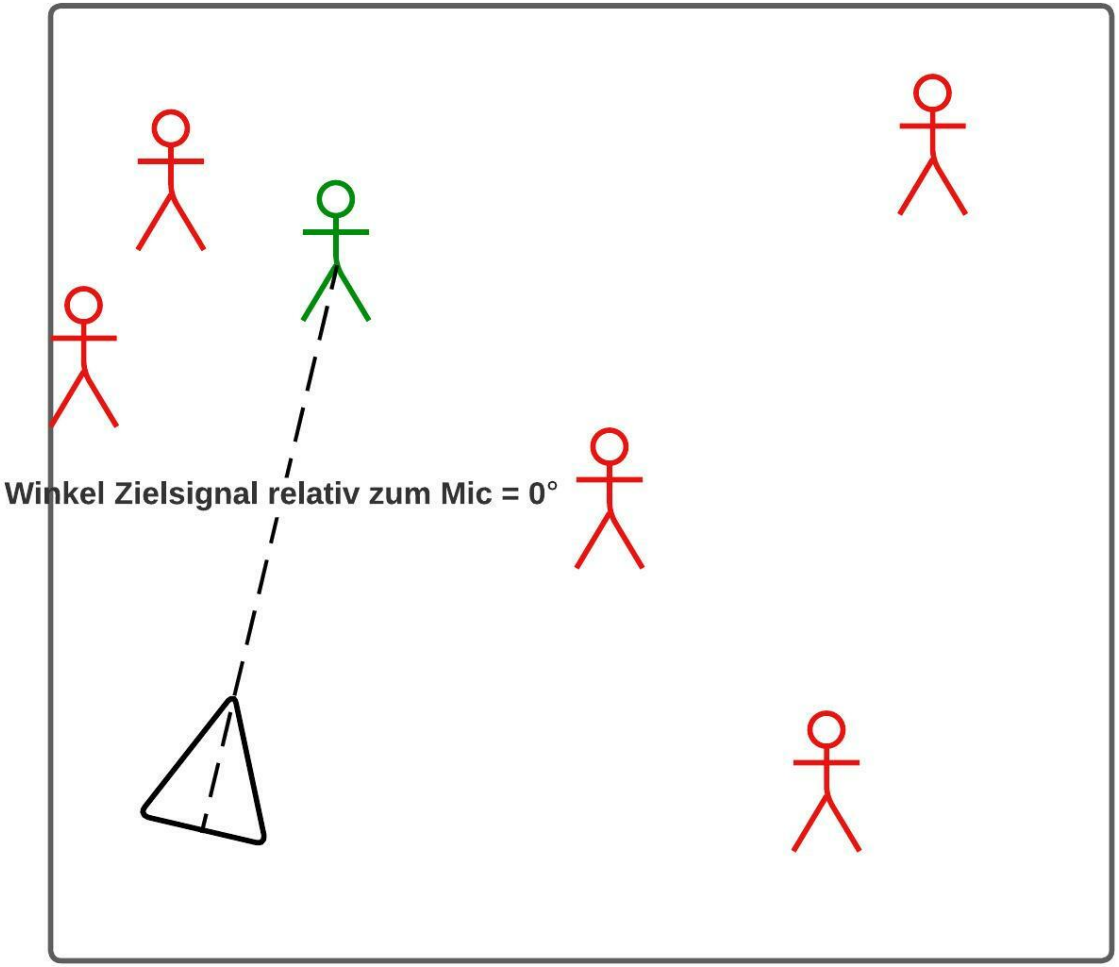

Figure: Red: Noise sources, Green: Target source, Triangle: Mic-Array

Datengenerierung - Trainingsziel

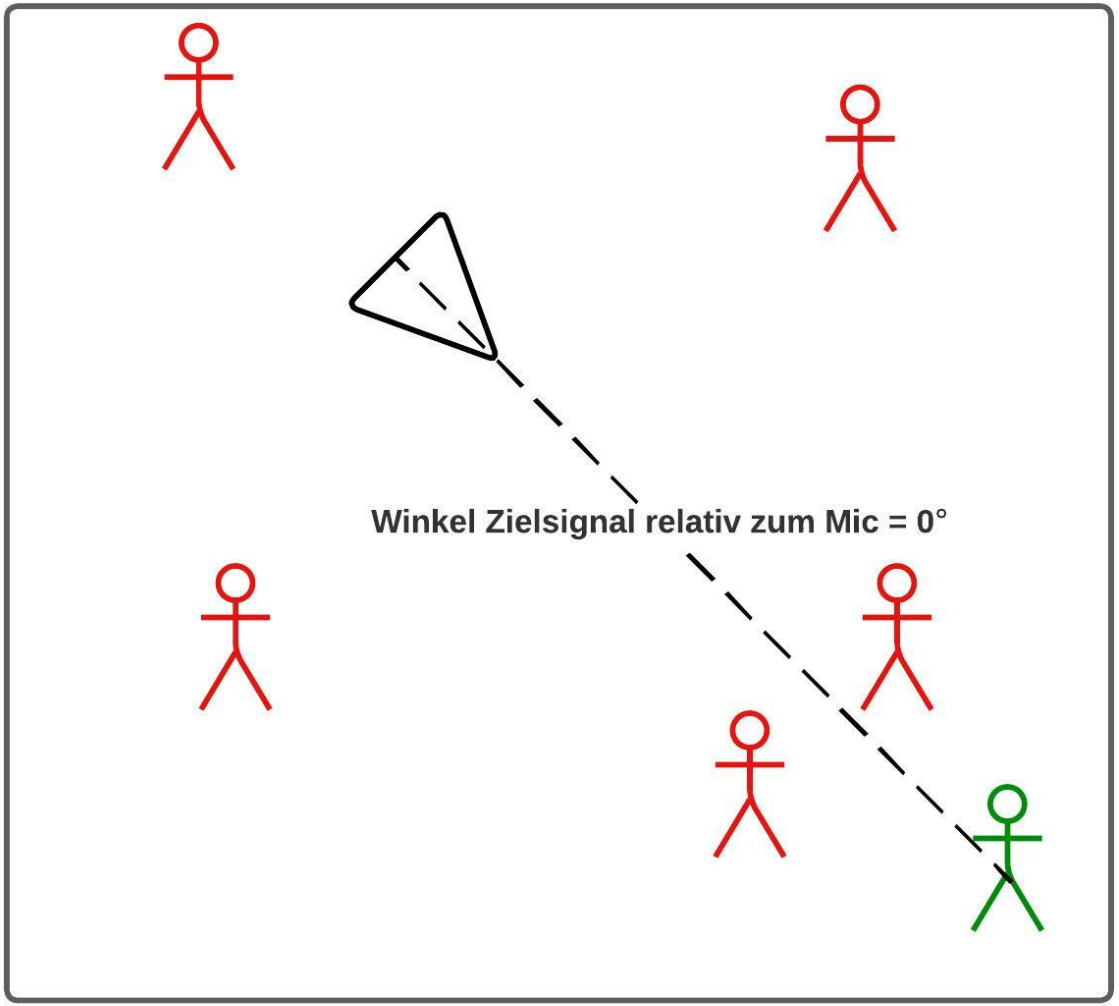

Figure: Red: Noise sources, Green: Target source, Triangle: Mic-Array

Datengenerierung - Raum SetUp

- Raumsimulation mit PyRoomAcoustics
- 2-dimensionaler Raum
- Breite (X-axis): 5 Meter
- Länge (Y-axis): 9 Meter
- Nachhallzeit (RB60): 0.5 Sekunden
- TODO?: 3 Dimensionen, Random Raum-Maße

Datengenerierung - Mic SetUp

- 2-dimensionales Mikrofon-Array aus 3 Mikrofonen
- Kreisförmige Anordnung mit Radius 10 Zentimeter?
- Rotation wird random bestimmt $\rightarrow$ Grundlage für Target Position

Datengenerierung - Mic Position

- Random coordinate im Raum
- Mindestems 1 Meter Abstand zu Wänden
- Rotation $=$ random Grad-Wert zwischen 0 and 360
- Rotation wird in Kombination mit der Position validiert (ist besteht genug Platz entlang der Ausrichtung um Target Source zu platzieren?)

Datengenerierung - Target Source Position

- Bestimmung von Koordianate "auf" der Rotation des Mic-Arrays
- Random Distanz auf diesem Winkel
- Distanz beschränkt durch nutzerdefinierte Min.- und Max.-Abstände
- $x = xMic + (distance * \cos MicRotationInRadiants)$
- $y = yMic + (distance * \sin MicRotationInRadiants)$

Datengenerierung - Trainingsziel

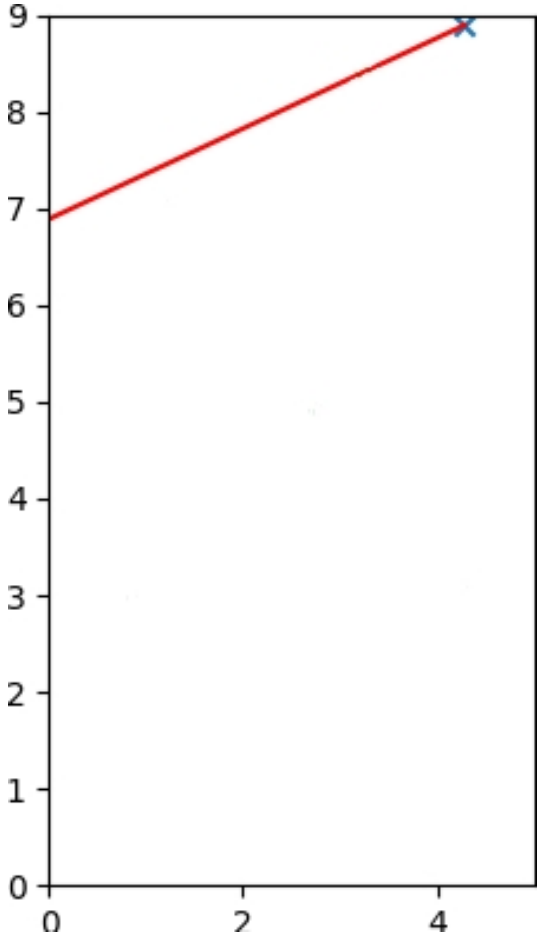

Figure: Mic Array in the room with with its rotation (red line)

Datengenerierung - Noise Sources Positionen

- Random Koordinate im Raum
- Validität der Position wird geprüft: Min.-Distanzen zu Mic-Array und zu Target-Position
- TODO?: Prüfen, ob außerhalb der Mic-Ausrichtung

Datengenerierung - Noise Source Positioning

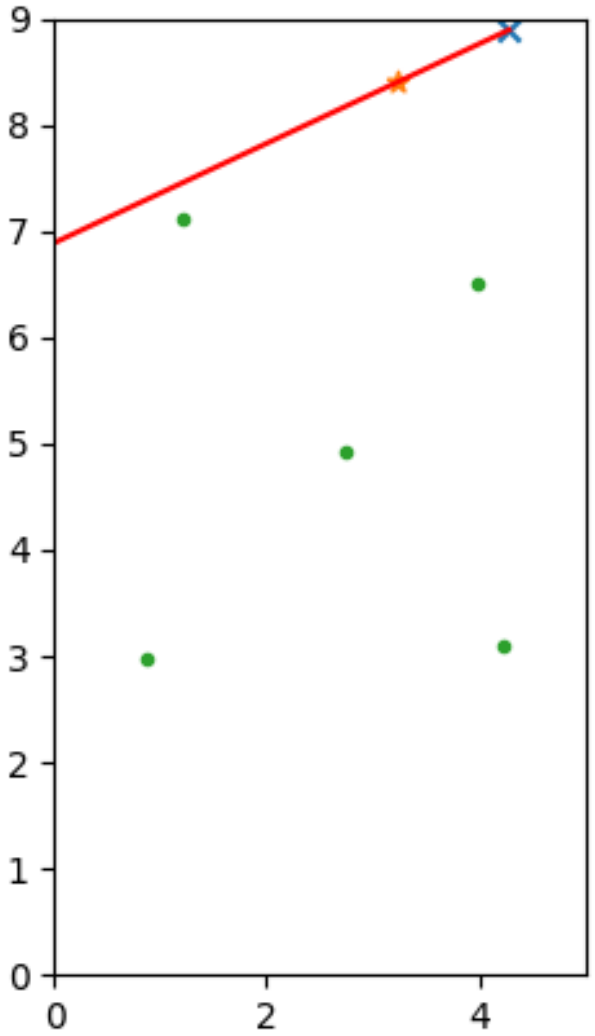

Figure: Blau: Mic-Array, Grün: Noise Sources, Orange: Target Source, Rot: Ausrichtung Mic Array

Datengenerierung - Zuweisung Audiosignale

- Je Source wird ein random Unterordner/=Sprecher ("01a, 01b usw.") aus dem vorgegebenen "si_tr_s" Datenset ausgewählt
- Dabei wird niemals derselbe Sprecher/Unterordner für mehrere Quellen ausgewählt
- Wähle aus Unterordner eine random Audiodatei
- Ergebnis: Liste mit Target und Noise Audio-Signalen

  $\Rightarrow$ Verwende Audio-Liste beim Einfügen der Sources in den Raum

Datengenerierung - Extraktion Mikrofon-Signale

- Simuliere die Raumakustik mit PRA und erhalte das Mic-Input-Signal ("Premix")
- Wähle aus Premix das Target-Signal aus für Clean-Datei
- Wähle aus Premix die Noise-Signale aus für Noise-Datei
- Erzeuge eine Clean- und Noise-Datei im Zielpfad
- Jede Datei enthält jeweils 3 Channel (einzelne Mics)

Plotting

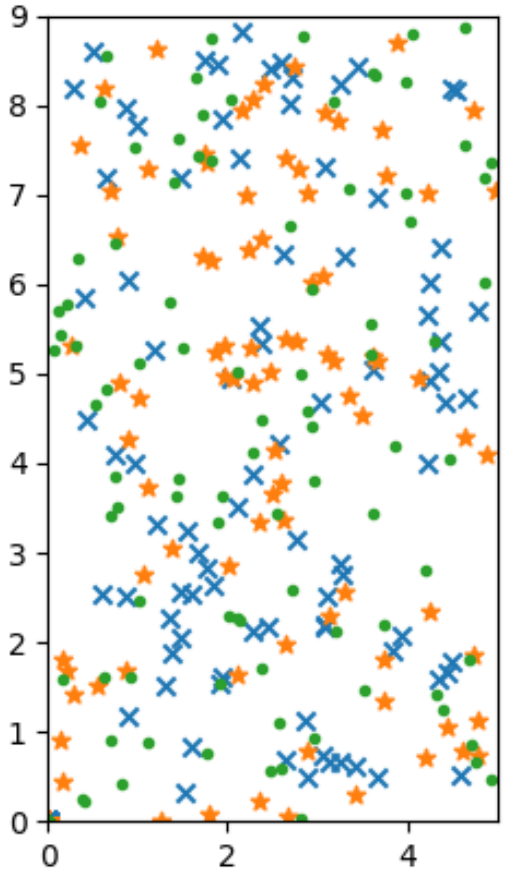

Figure: Blau: Mic-Arrays, Grün: Noise Sources, Orange: Target Sources

Überblick

1. Raum erstellen
2. Erstellen und einfügen des Mic-Array mit random rotation
3. Erstellen und einfügen der Target Source auf Mic rotation
4. Erstellen und einfügen der Noise Sources
5. Raum simulieren und Clean- und Noise-Datei erzeugen

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

SP
Signal Processing

Datenset

Dataset Klasse - Ziel

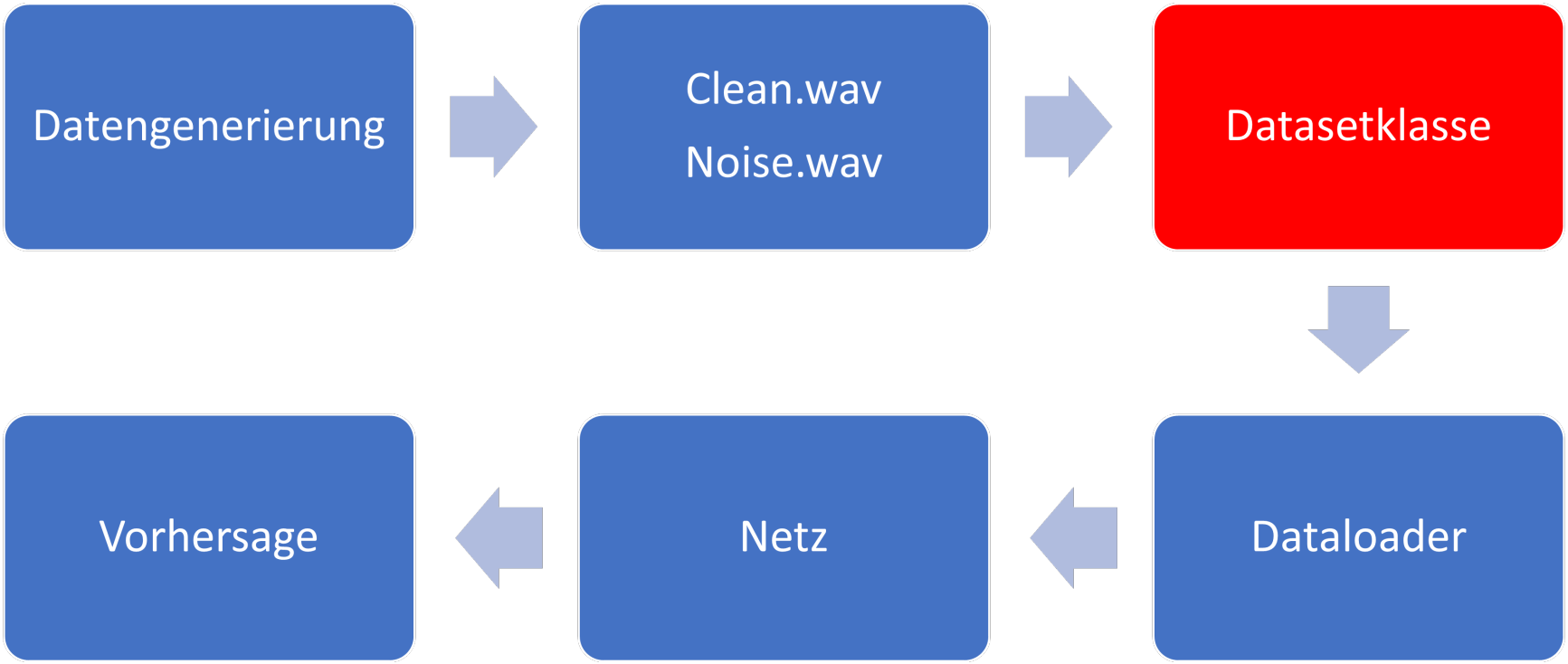

Figure: Schaubild zur Zielsetzung und Einordnung der Dataset Klasse

Dataset Klasse - Funktionen

Die Dataset Klasse definiert bestimmte Operationen, die dann vom Dataloader benutzt werden können.

- **length-Funktion**, die die Anzahl der Audiodateien im Dataset angibt
- **get_item-Funktion**, die definiert, wie eine Audiodatei eingelesen und evtl. weiterverarbeitet werden soll, bevor sie dann übergeben wird
- **cut-Funktion**, die bestimmt in welchen Situationen eine Audiodatei zu einem z.B. 3sek Schnipsel geschnitten werden soll

Dabei funktionieren die Operationen teilweise unterschiedlich, je nachdem, ob wir gerade trainieren, validieren oder testen.

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

SP Signal Processing

Netz
Grundidee, Architektur, Training, Vorhersagen

Netz: Grundidee

- Zeitliche/räumliche Eigenschaften von Sprache und Rauschen:
  - Sprache ist nicht-stationär und kohärent.
  - Rauschen ist stationär und weniger räumlich korreliert.

Netz: Grundidee

- Long short-term memory (LSTM) recurrent neural network (RNN) nutzt dies aus und lernt:
  - Zwischen den räumlichen Eigenschaften einer gerichteten Sprachquelle und Rauschen zu unterscheiden.
  - Ein gemeinsames Modell für alle Teilbänder.
- Daher unempfindlich gegenüber Position der Sprachquelle.

Netz: Grundidee

- **Input:**
  - STFT einer verrauschten mehrkanäligen Audiosequenz.
  - Zugehörige klare einkanälige Audiosequenz.
- **Output:** Komprimierte binäre Maske.
- **Anwendung:** Maske wird dekomprimiert und auf das verrauschte Signal angewandt.
- **Ergebnis:** Entrauschte Audiosequenz als Schätzung der klaren Audiosequenz.

Netz: Architektur

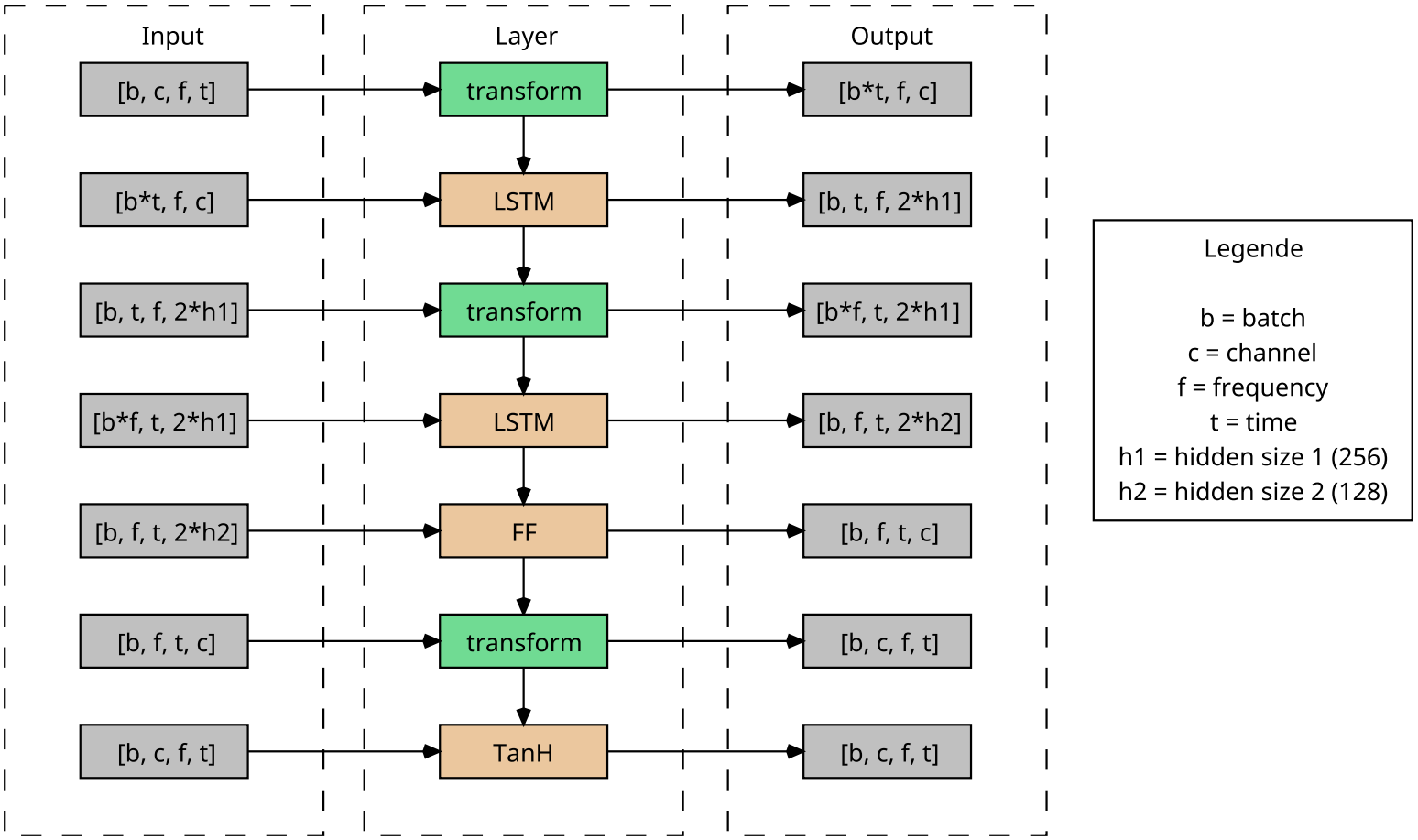

Figure: Netzarchitektur.

Netz: Features

- **Early stopping:** Dropout nach 10 Epochen ohne Verbesserung des Validierungsverlusts, um Overfitting vorzubeugen.
- **Checkpointing:** Speichern der Modellparameter der besten 2 Epochen (bezüglich minimalen Validierungsverlusts) um ein traniertes Modell für Vorhersagen laden zu können.

Netz: Training

- **Verlustfunktion:**
  - Mean Squared Error (MSE)
  - Scale-Invariant Signal-to-Distortion Ratio (SI-SDR)
- **Optimierer:**
  - Adaptive Moment Estimation (Adam)
  - Root Mean Squared Propagation (RMSProp)
- **Lernrate:** 0.01 - 0.00001

Netz: Training

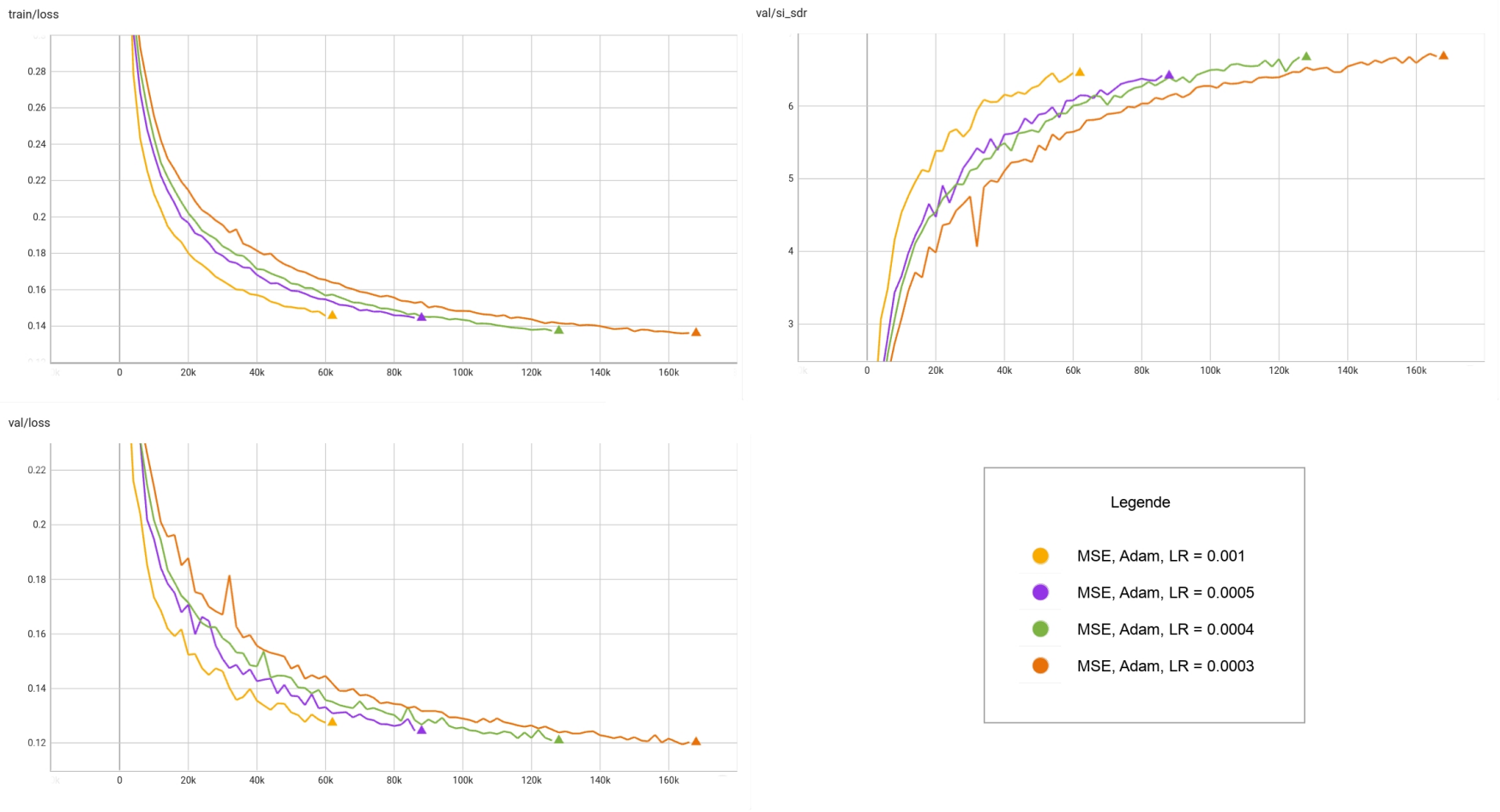

Figure: Training Performance Adam Optimizer different LR (reference data).

Netz: Training

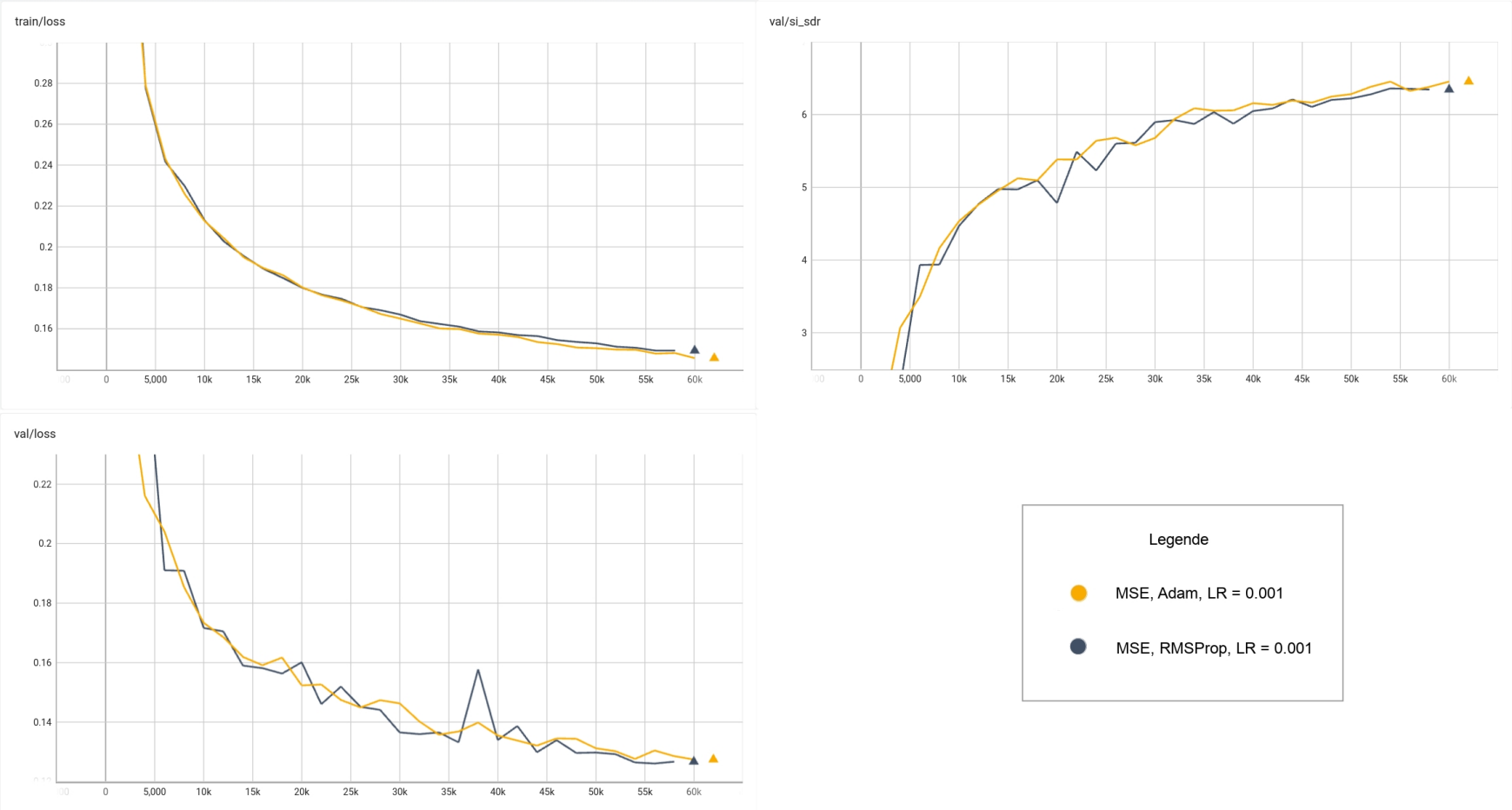

Figure: Training Performance Adam vs. RMSProp Optimizer (reference data).

Netz: Training

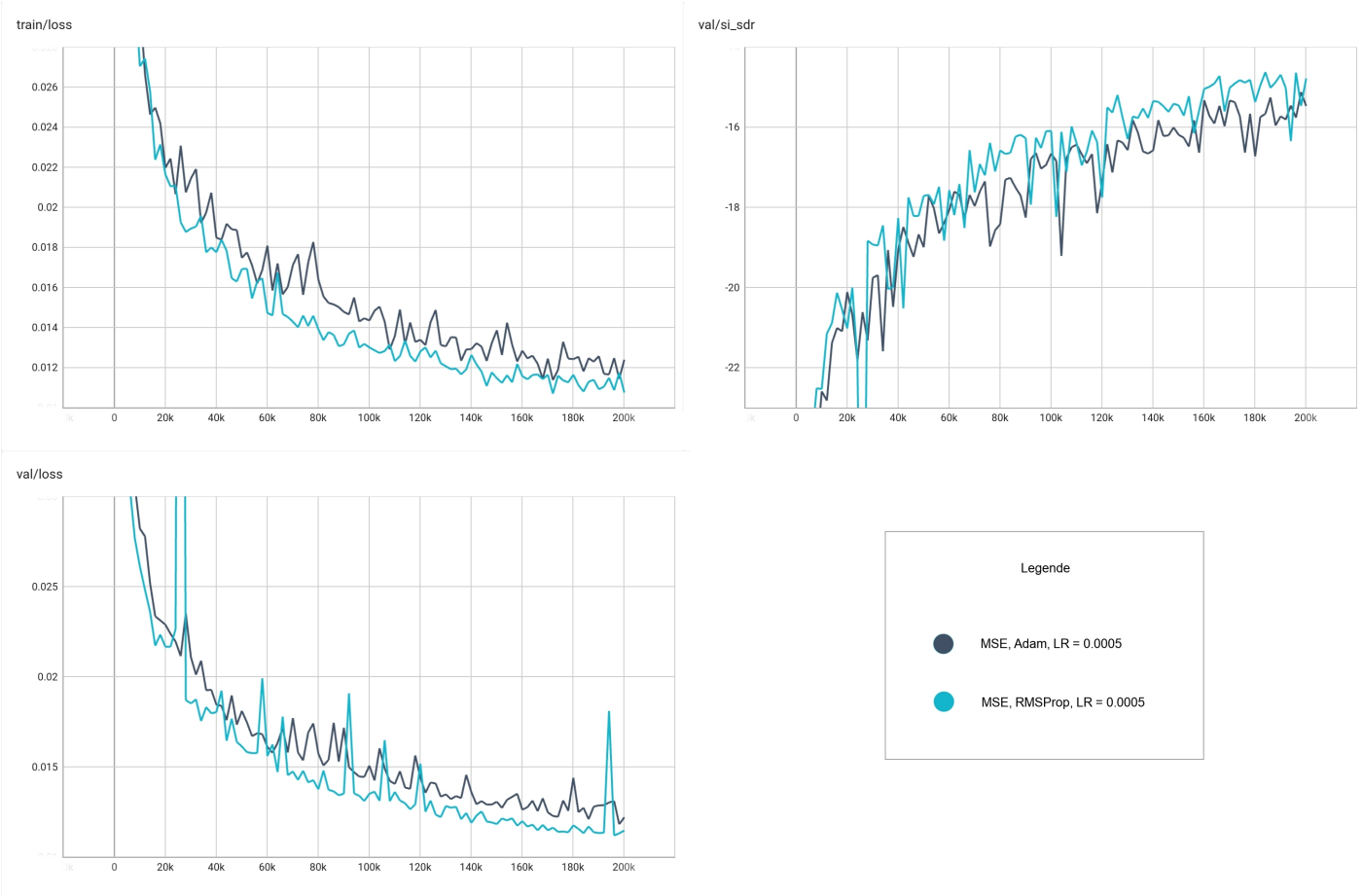

Figure: Training Performance Adam vs. RMSProp Optimizer (our data).

Netz: Vorhersage

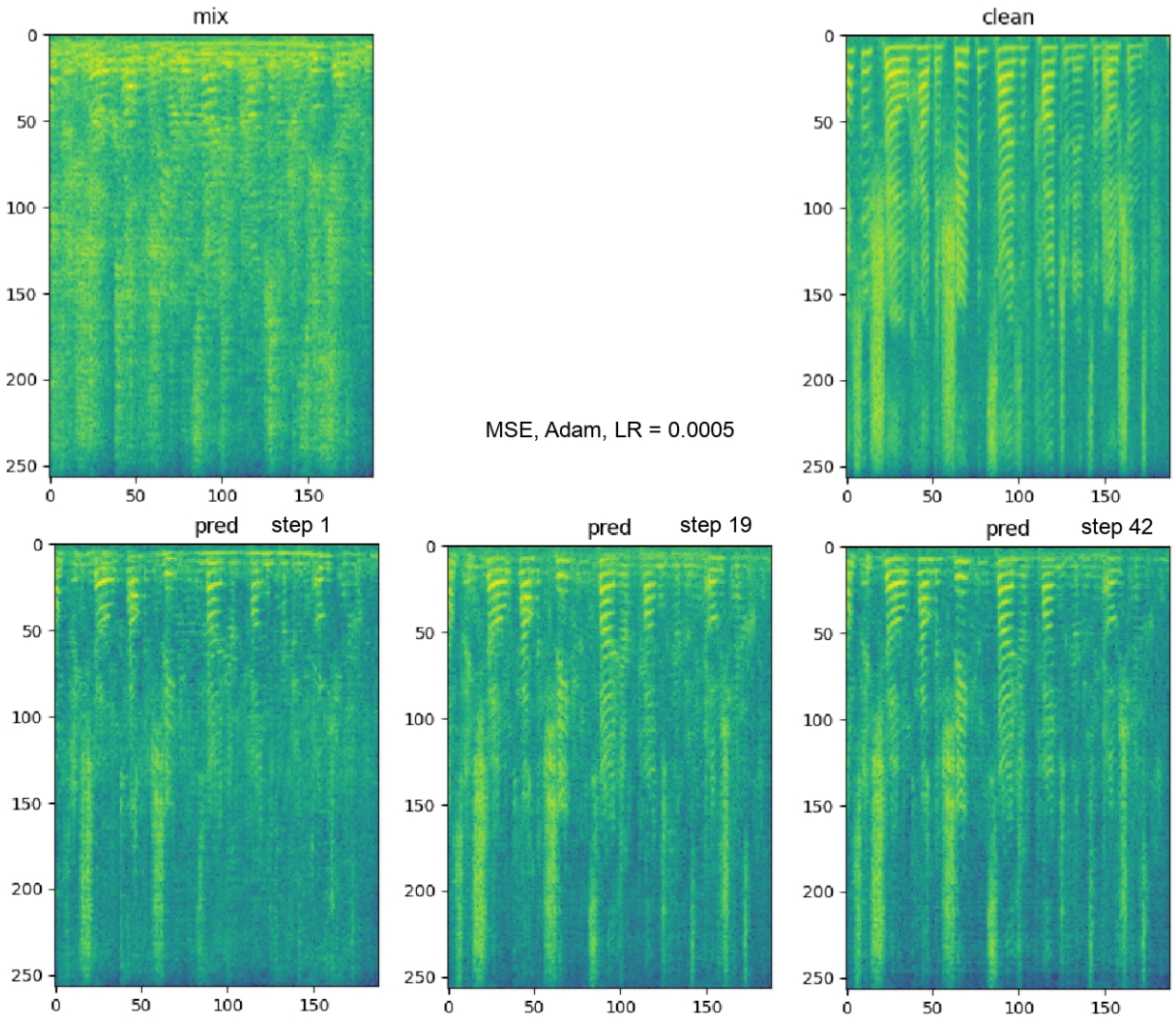

Figure: Training Performance Spectrograms (reference data).

Netz: Vorhersage

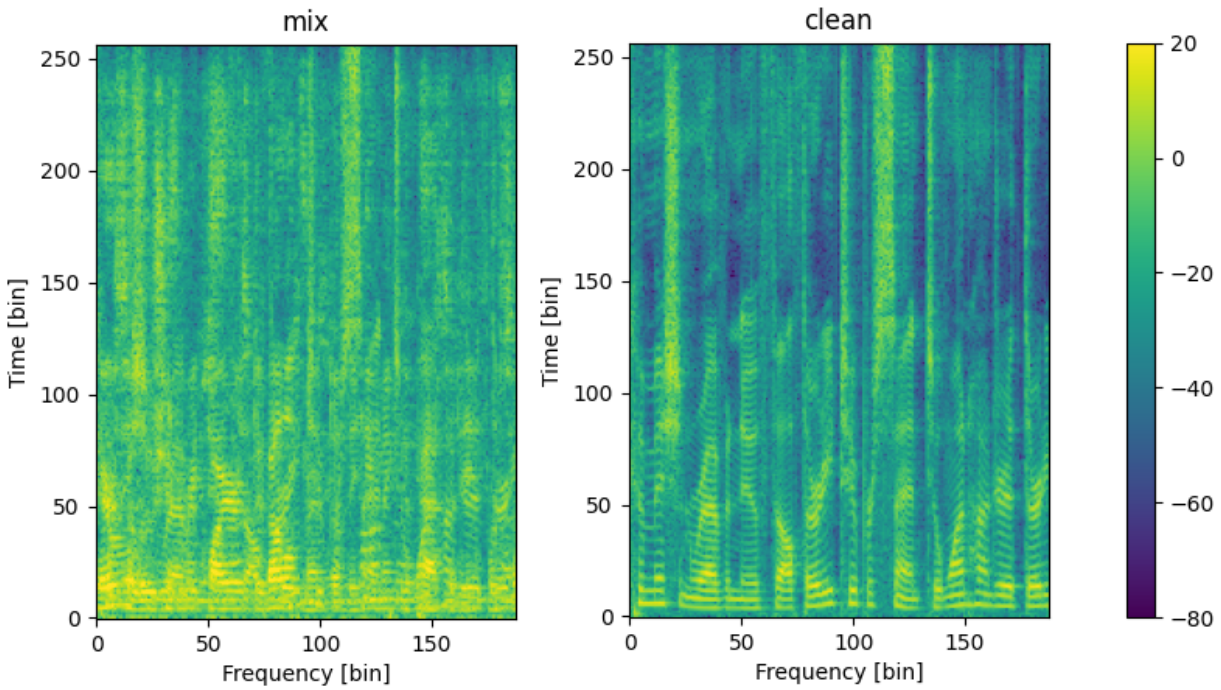

Figure: Training Performance Spectrograms (our data).

Netz: Vorhersage

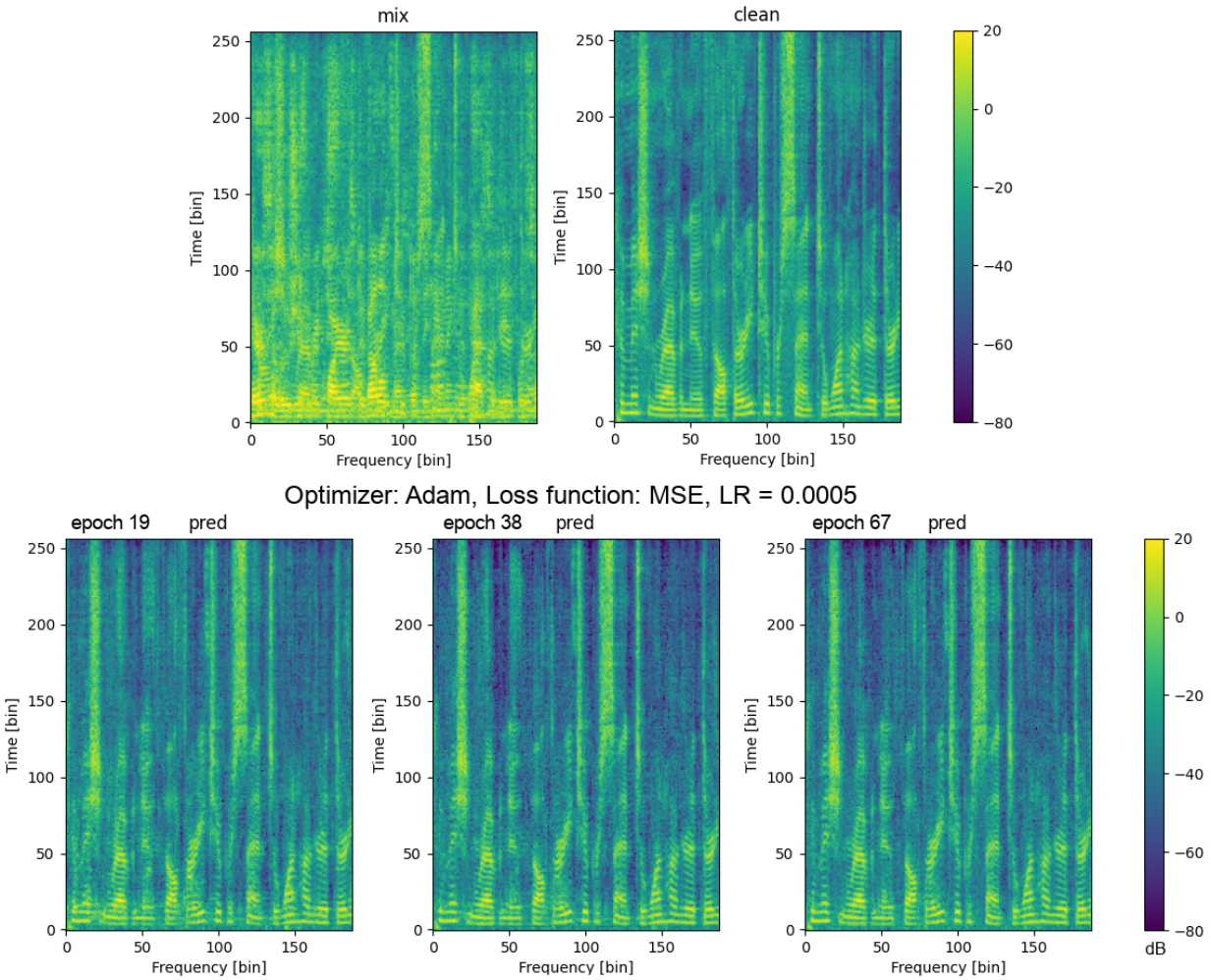

Figure: Training Performance Spectrograms, Adam optimizer (our data).

Netz: Vorhersage

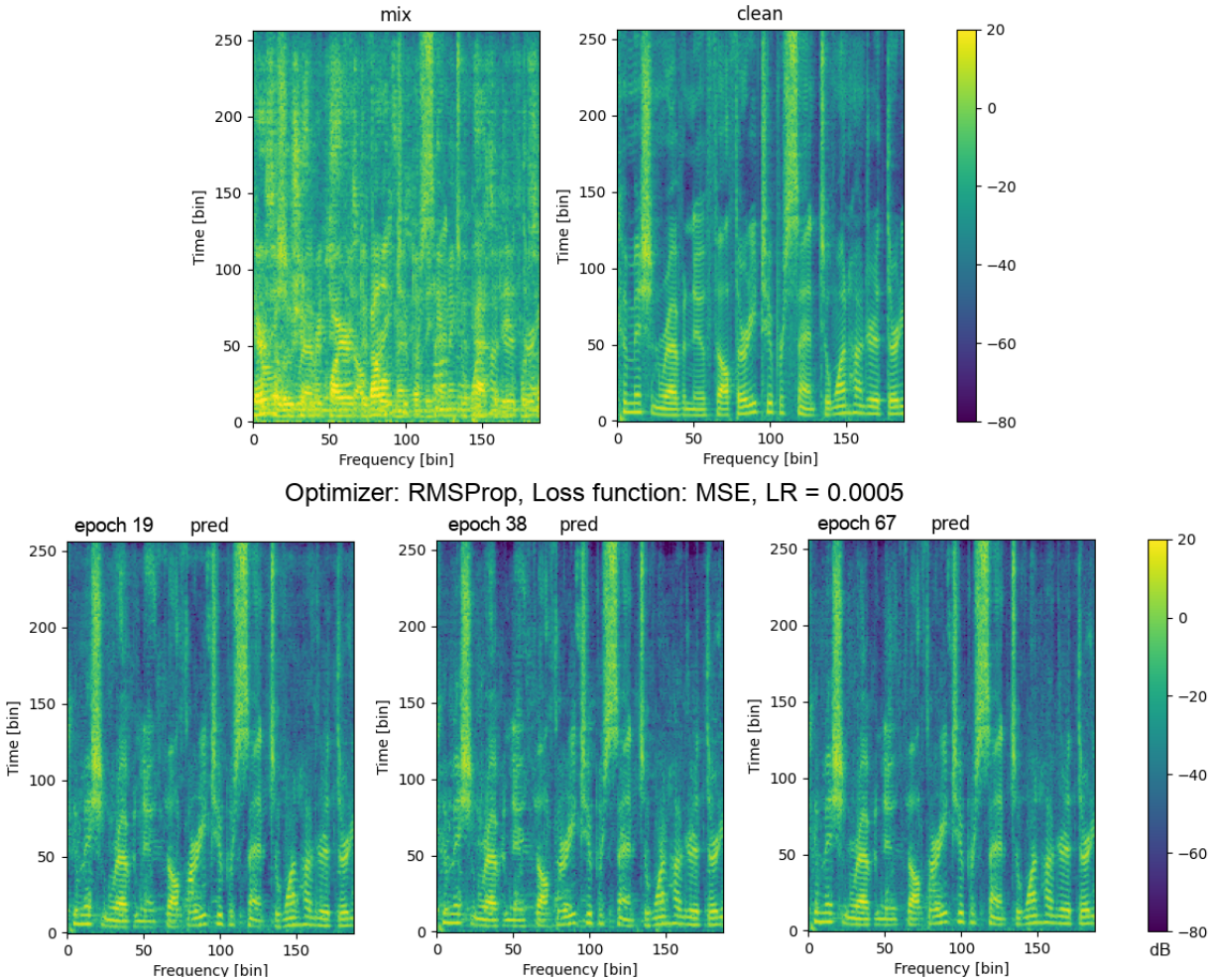

Figure: Training Performance Spectrograms, RMSProp optimizer (our data).

Netz: Vorhersage

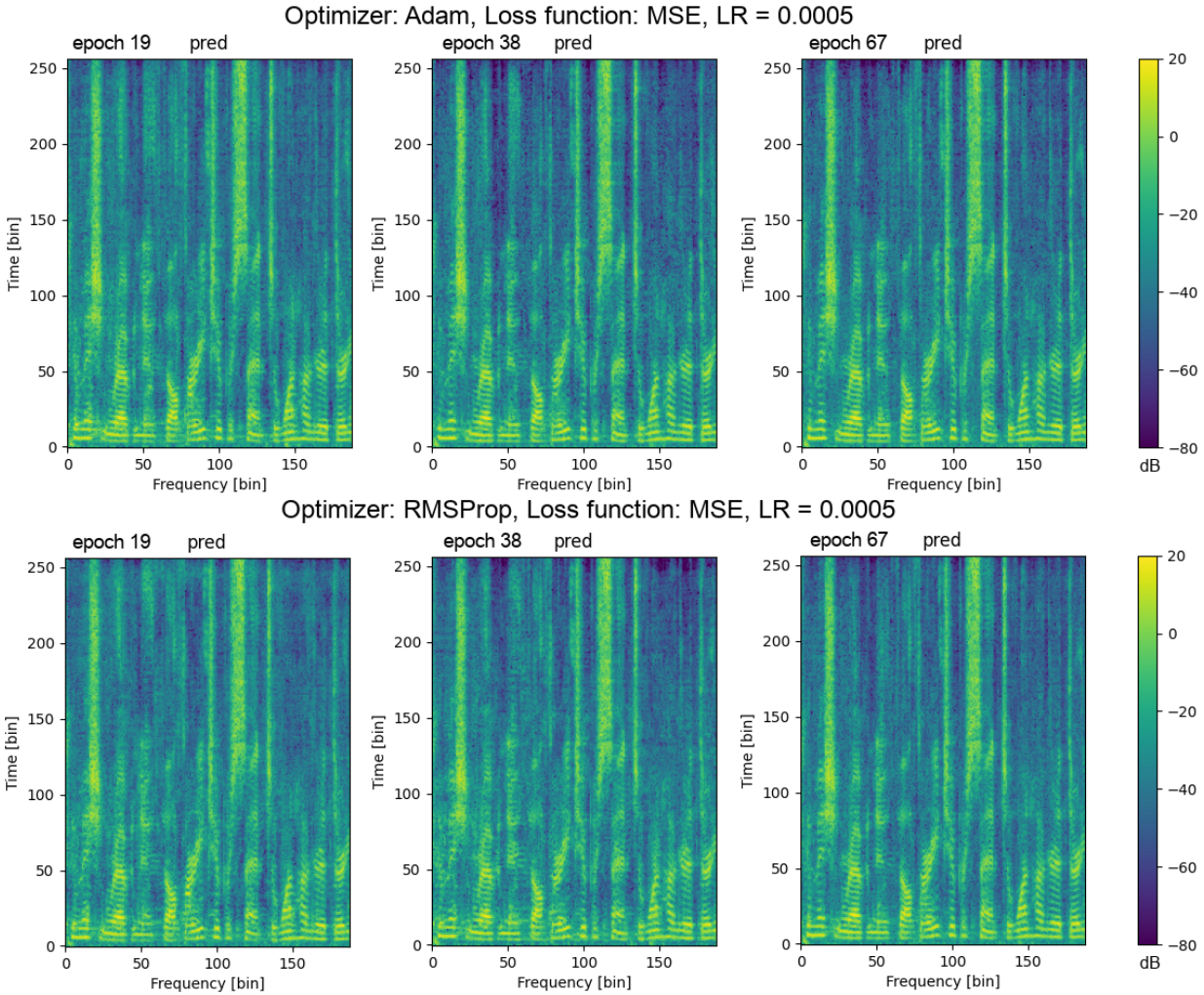

Figure: Training Performance Spectrograms, Adam vs. RMSProp (our data).

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

SP
Signal Processing

Evaluation

Evaluation des Ergebnisses

Qualität unseres Filters bewerten wir mit der "Perceptual Evaluation of Speech" kurz PESQ.

- PESQ von der Prediction im Verhältnis zum Clean-Signal
- PESQ von der Prediction im Verhältnis zum Noise-Signal
- **Differenz/Delta** von PESQ(Clean) und PESQ(Noise)

$\Rightarrow$ Noch können wir leider keinen Wert hierfür vorweisen, aber wir sind nah dran!

Evaluation unserer Arbeit

Was gut lief:

- Aufteilung der Arbeitsschritte
- Absprache bzw. Kommunikation untereinander
- hoher Lerneffekt durchs Fehler-machen und "learning by doing"

Evaluation unserer Arbeit

Wo es Schwierigkeiten gab:

- Schnittstelle zwischen Audiodateien, Dataset-Klasse und Netz (wer braucht was?)
- Fehlersuche (wo bzw. bei wem liegt der Fehler eigentlich?)

$\Rightarrow$ Netz lernt nicht dazu beim Trainieren

$\Rightarrow$ Ursache: Audiodateien

$\Rightarrow$ Effekt: Zeitverlust

Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

SP Signal Processing

Ausblick

Probleme

- Der erste clean-Datensatz besteht nur aus Nullen
- Ab und zu gibt die Lossfunction `nan` zurück
- Wenn wir die cut-Funktion nicht verwenden (beim Testen) sind die Audiodateien nicht gleich lang
- Mit unseren Audiodateien bekommen wir keine positive SISDR

Lösungsansätze

Für die `nan's` könnte man versuchen die Torch-Funktion nan_to_num zu verwenden.

$\Rightarrow$ Ein allgemeiner Verbesserungsvorschlag wäre, einen 3D statt nur 2D Raum für die Datenerzeugung zu verwenden.

Danke fürs Zuhören