Project Deliverable D1.3

Team

Vladislav Kalinichenko v.kalinichenko@innopolis.university
Polina Korobeinikova p.korobeinikova@innopolis.university
Janna Ivanova j.ivanova@innopolis.university

Repository

https://github.com/poinka/NewWave.git

Project Topic

Natural Language Music Recommender System

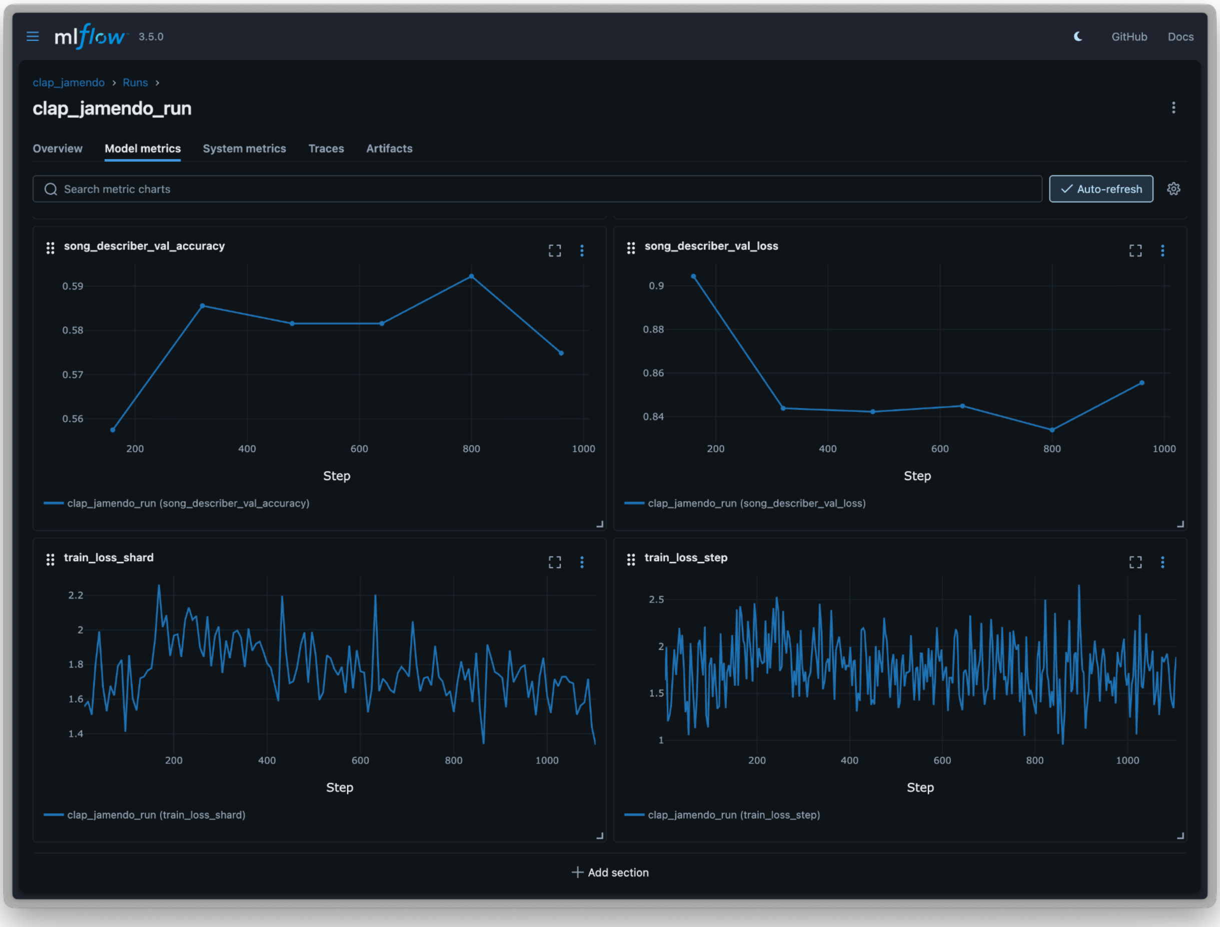
What has been done so far

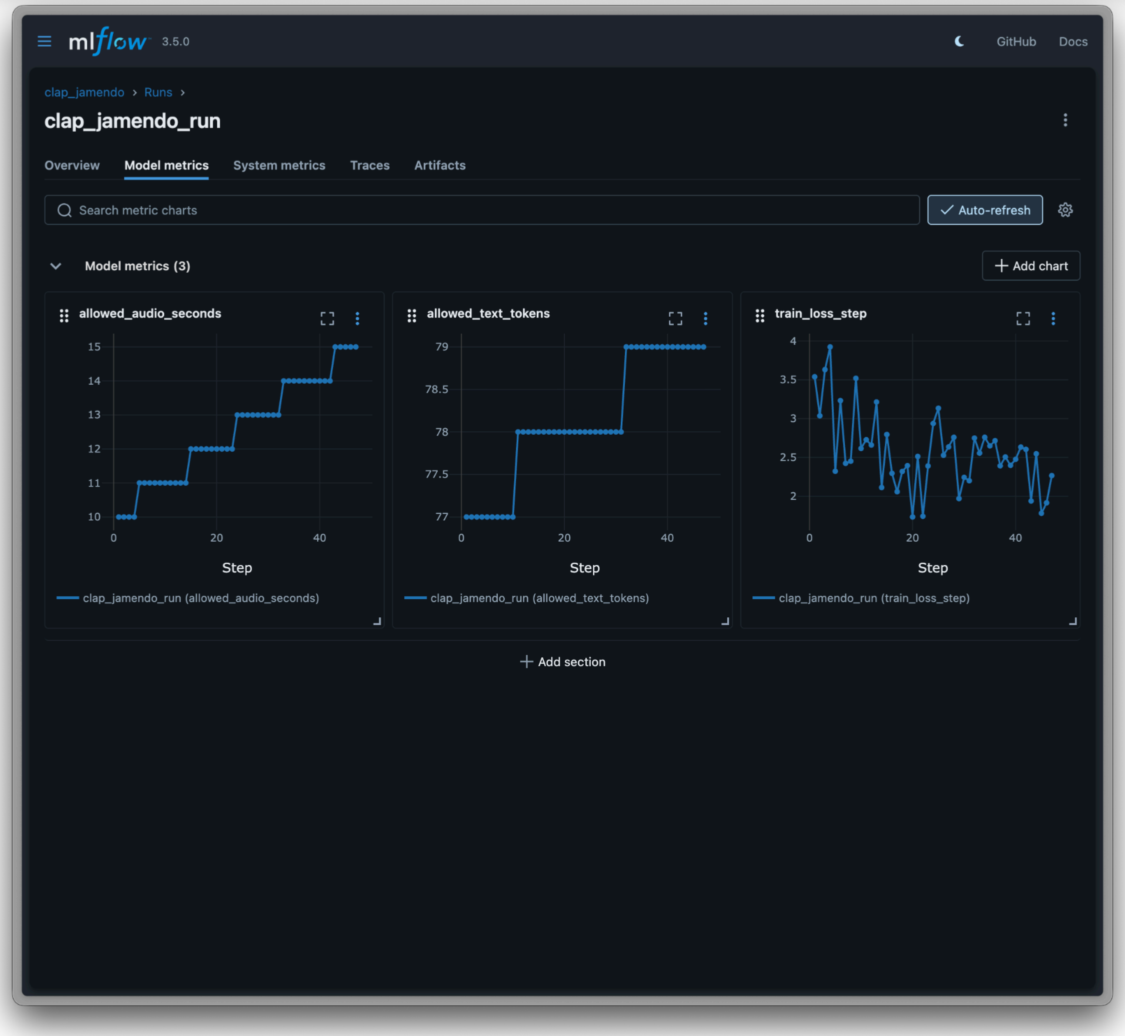
- Performed exploratory data analysis on Jamendo (large-scale, ~1 TB with many songs) and Sound Describer (tiny, higher-quality) to decide dataset roles; used Jamendo for training and some evaluation batches, and Sound Describer only for evaluation to gauge generalization under description shifts.
- Implemented a parquet-by-parquet pseudo-streaming pipeline to overcome storage limits and unstable direct streaming, enabling per-file load-and-delete to simulate Hugging Face streaming during training with controllable batches.
- Extended the CLAP context window via a linear schedule: audio clips increased from 10 seconds toward 240 seconds and text descriptions expanded beyond 77 tokens, approximately adding one second per several batches.
- Ran long-context CLAP training on A100 for about one day, covering roughly 10% of the dataset; observed both stagnation and sharp validation gains across runs on Jamendo subsets and Sound Describer, and continued training to improve generalization.
- Iterated the lyrics/description retrieval encoder: removed Longformer due to poor convergence and inefficiency; trained all-MiniLM-L6-v2 with hard negatives (artists, then tags); tested an average-pooled text variant; switched to bge-m3, increased batch for stability, and extended to 25 epochs for best results.

Results

CLAP

- Explored two datasets with distinct roles: Jamendo as a very large corpus (~1 TB with many songs) used for training and some evaluation batches, and Sound Describer as a tiny but higher-quality set used only for evaluation to assess generalization under description style shifts.
- Due to storage limits and unstable direct streaming from the hub, implemented a parquet-by-parquet pseudo-streaming loader: load each parquet shard, process batches, then delete to free space, effectively simulating streaming with controllable batch flow on both laptops and InnoDataHub servers.
- With this pipeline, a full training day on A100 covered roughly 10% of the dataset, confirming throughput feasibility while indicating more epochs/passes are needed for stable generalization.
- Training objective was to extend CLAP beyond its pretraining limits of 10 s audio and 77-token text by applying a linear schedule that gradually increases audio duration toward 240 s and expands text length, roughly adding about one second of audio per several batches while similarly loosening the token cap on descriptions.
- Validation behavior varied by run: some runs showed stagnation while others produced sharp gains on Jamendo subsets and on Sound Describer evaluation, consistent with sensitivity to schedule, batch regime, and data stream composition; extended training is ongoing to consolidate gains and improve generalization across description styles.
  Metrics:
  MRR: 0.1122
  Recall@k:
  Recall@1: 0.0442
  Recall@5: 0.1521
  Recall@10: 0.2487
  Recall@20: 0.3820
  Precision@k:
  Precision@1: 0.0442
  Precision@5: 0.0304
  Precision@10: 0.0249
  Precision@20: 0.0191


mlflow 3.5.0
GitHub
Docs
clap_jamendo › Runs ›
clap_jamendo_run
Overview
Model metrics
System metrics
Traces
Artifacts
Search metric charts
Auto-refresh
song_describer_val_accuracy
0.59
0.58
0.57
0.56
200
400
600
800
1000
Step
clap_jamendo_run (song_describer_val_accuracy)
song_describer_val_loss
0.9
0.88
0.86
0.84
200
400
600
800
1000
Step
clap_jamendo_run (song_describer_val_loss)
train_loss_shard
2.2
2
1.8
1.6
1.4
200
400
600
800
1000
Step
clap_jamendo_run (train_loss_shard)
train_loss_step
2.5
2
1.5
1
200
400
600
800
1000
Step
clap_jamendo_run (train_loss_step)
Add section


mlflow 3.5.0
GitHub
Docs
clap_jamendo › Runs ›
clap_jamendo_run
Overview
Model metrics
System metrics
Traces
Artifacts
Search metric charts
Auto-refresh
Model metrics (3)
Add chart
allowed_audio_seconds
15
14
13
12
11
10
0
20
40
Step
clap_jamendo_run (allowed_audio_seconds)
allowed_text_tokens
79
78.5
78
77.5
77
0
20
40
Step
clap_jamendo_run (allowed_text_tokens)
train_loss_step
4
3.5
3
2.5
2
0
20
40
Step
clap_jamendo_run (train_loss_step)
Add section

SOTA comparison

- Against recent state of the art on the same evaluation set (Song Describer), CLaMP 3 (2025) reports MRR = 0.1985; our best current retrieval run reaches MRR = 0.1169 on the same dataset, i.e., about 1.76× behind the latest SOTA; the paper also reports a CLAP fine-tuned baseline of 1.131 (same family as ours), which aligns with our setup and supports the fairness of the comparison (source: **https://arxiv.org/pdf/2502.10362v1**)
- Relative to prior CLAP results, our R@10 = 0.2487 is within 4.4% of the CLAP 2023 figure (0.2601), and our R@1 = 0.0442 matches CLAP 2023 exactly, indicating competitive early-rank retrieval while long-context training continues (source: **https://arxiv.org/pdf/2311.10057**)

Lyrics/Description retrieval

Model iterations and rationale

- Removed the Longformer model because it failed to train, was very slow, and consumed too much memory; as a result, lyric embeddings had to be refreshed only every 2000 steps, which was inefficient. Literature checks also showed it is not well-suited for semantic retrieval unless further adapted on very large datasets.
- Switched to all-MiniLM-L6-v2. Training was significantly faster, and it was possible to add hard negatives (first by artists, then by tags) without excessive memory pressure. With 256-token truncation for both descriptions and lyrics (the best-performing length), the best test results were:

  MRR: 0.0685
  Recall@1: 0.0288
  Precision@1: 0.0288
  Recall@5: 0.0984
  Precision@5: 0.0197
  Recall@10: 0.1479
  Precision@10: 0.0148
  Recall@20: 0.2065
  Precision@20: 0.0103

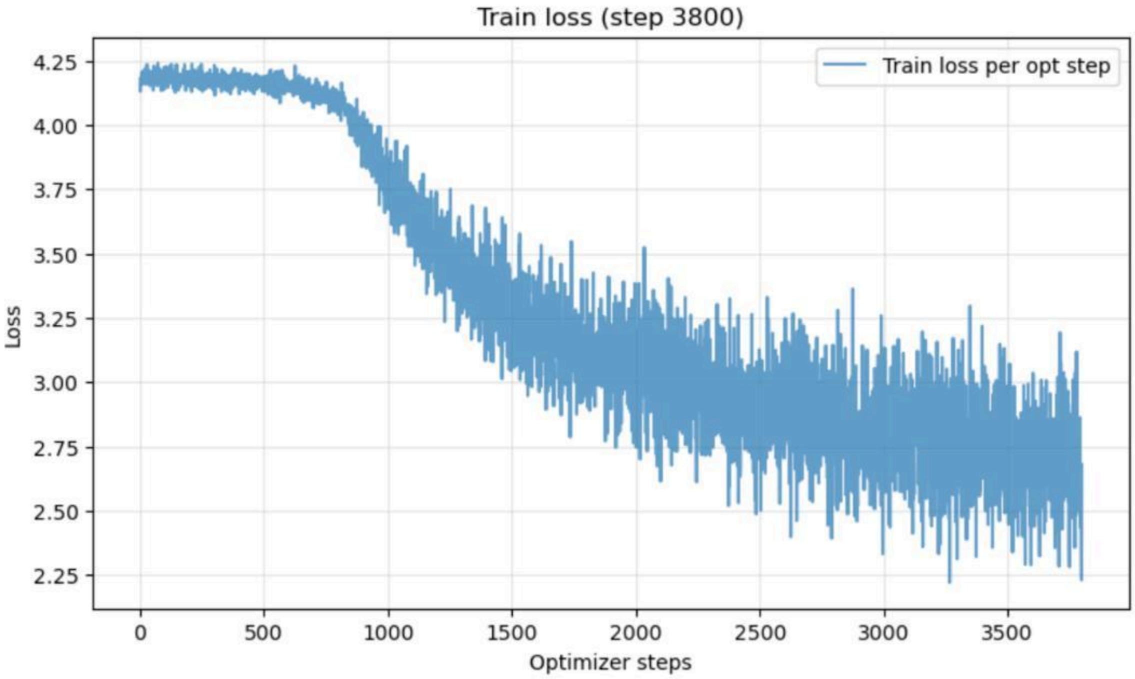
  The problem was that we cut the text, description, and lyrics to 256 tokens because the model produced the best results at that length, which meant a lot of the text simply didn't make it into the model.

- Tried average pooling over the full text; training became much longer and did not pay off:

  MRR: 0.0086
  Recall@1: 0.0016
  Precision@1: 0.0016
  Recall@5: 0.0084
  Precision@5: 0.0017
  Recall@10: 0.0164
  Precision@10: 0.0016
  Recall@20: 0.0298
  Precision@20: 0.0015

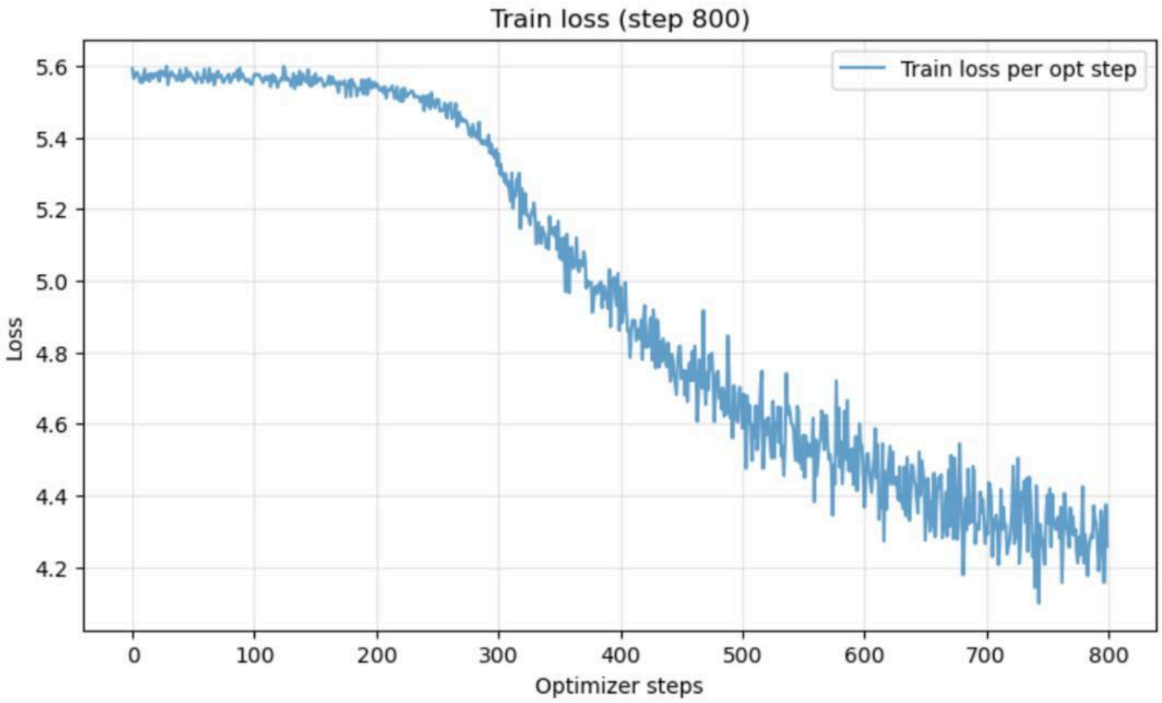
- Moved to a larger model that can take both the full description and lyrics: bge-m3. Trained separate heads for lyrics and descriptions. Initial results:

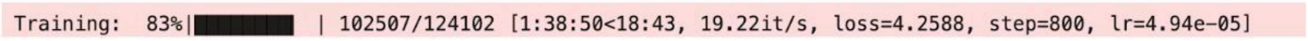
  MRR: 0.0871
  Recall@1: 0.0404
  Precision@1: 0.0404
  Recall@5: 0.1258
  Precision@5: 0.0252
  Recall@10: 0.1849
  Precision@10: 0.0185
  Recall@20: 0.2499
  Precision@20: 0.0125



- Increased batch size to stabilize training; metrics improved:

  MRR: 0.0932
  Recall@1: 0.0441
  Precision@1: 0.0441
  Recall@5: 0.1325
  Precision@5: 0.0265
  Recall@10: 0.1947
  Precision@10: 0.0195
  Recall@20: 0.2641
  Precision@20: 0.0132



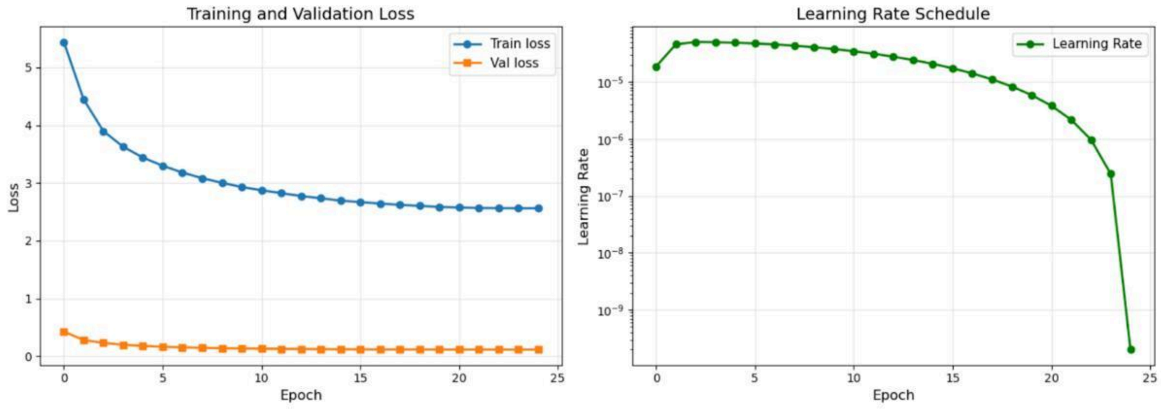


```
Training:  83%|████████   | 102507/124102 [1:38:50<18:43, 19.22it/s, loss=4.2588, step=800, lr=4.94e-05]
```

- Extended training to 25 epochs (from 5 previously), achieving the best results so far:

  MRR: 0.1169
  Recall@1: 0.0542
  Precision@1: 0.0542
  Recall@5: 0.1734
  Precision@5: 0.0347
  Recall@10: 0.2496
  Precision@10: 0.0250
  Recall@20: 0.3335
  Precision@20: 0.0167

last model training



Work distribution

- **Polina Korobeinikova:** Conduct lyrics/description encoder experiments; removed Longformer; trained all-MiniLM-L6-v2 with hard negatives; tested average pooling; switched to bge-m3, stabilized with larger batch, and scaled to 25 epochs to reach the best metrics.
- **Janna Ivanova:** Conduct EDA on Jamendo and Sound Describer; determined dataset roles (Jamendo for training and partial eval, Sound Describer for eval) to test generalization under description shifts. Create this report.
- **Vladislav Kalinichenko**: Built the parquet-based pseudo-streaming loader to handle storage/streaming constraints; orchestrated long-context CLAP runs and monitored validation behavior and throughput on A100.

Plan for the Next Weeks

- develop a demo application
- download songs for a retrieval
- connect endpoints to work with the application
- train a retrieval system that combines two current models