Robust Neural Audio Fingerprinting using Music Foundation Models

Shubhr Singh[1*] **Kiran Bhat**[1*] **Xavier Riley**[1] **Benjamin Resnick**[1]
John Thickstun[1,2] **Walter De Brouwer**[1]

[1]**SoundPatrol** [2]**Cornell University**

Abstract

The proliferation of distorted, compressed, and manipulated music on modern media platforms like TikTok motivates the development of more robust audio fingerprinting techniques to identify the sources of musical recordings. In this paper, we develop and evaluate new neural audio fingerprinting techniques with the aim of improving their robustness. We make two contributions to neural fingerprinting methodology: (1) we use a pretrained music foundation model as the backbone of the neural architecture and (2) we expand the use of data augmentation to train fingerprinting models under a wide variety of audio manipulations, including time streching, pitch modulation, compression, and filtering. We systematically evaluate our methods in comparison to two state-of-the-art neural fingerprinting models: NAFP and GraFPrint. Results show that fingerprints extracted with music foundation models (e.g., MuQ, MERT) consistently outperform models trained from scratch or pretrained on non-musical audio. Segment-level evaluation further reveals their capability to accurately localize fingerprint matches, an important practical feature for catalog management.

1 Introduction

Audio fingerprinting identifies unknown audio by extracting compact feature representations, or *fingerprints*, from a query and matching them against a reference database [1]. Fingerprinting has a wide range of applications such as music identification [2], integrity verification [3], and broadcast monitoring [4]. Queries often differ from reference tracks due to environmental degradations (e.g., noise, reverberation, microphone coloration) or deliberate modifications (e.g., pitch shifts, time stretches, lossy compression). Effective fingerprints should be both robust to such variations and discriminative enough to distinguish tracks.

Recent progress in neural audio fingerprinting [5] has shifted the field beyond classical methods like Shazam [2], towards contrastive learning [6] approaches that align representations of original and modified audio [5, 7]. Prior approaches to neural audio fingerprinting either learn these representations from scratch [5, 7] or adapt them from general-purpose audio models [8]. In this work we extend this line of research by exploring music foundation models as pretrained backbones for fingerprinting. Furthermore, we systematically evaluate each model's robustness against a broader set of manipulations that characterize modern media ecosystems, extending beyond noise and reverberation to include time steching [9], pitch shifting, compression, and filtering.

We focus on fingerprints derived from two music foundation models, MuQ [10] and MERT [11], as well as the general-purpose audio foundation model BEATs [12], previously considered for fingerprinting by [8]. To contextualize the performance of these fingerprints, we benchmark against two state of the art neural fingerprinting models, NAFP [5] and GraFPrint [7], as well as a Shazam-like baseline implemented with open source library Dejavu [13]. All models are assessed under a broad

[*]Equal contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: .

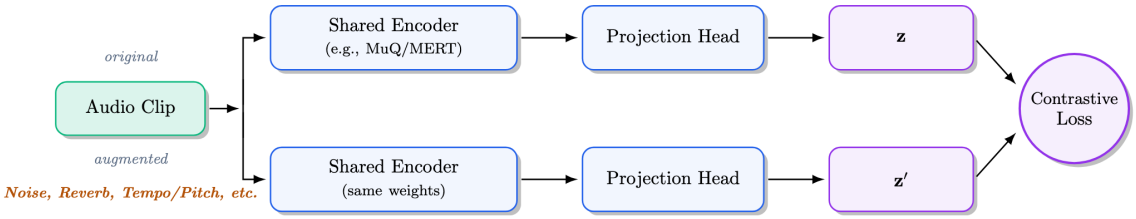

Figure 1: The contrastive learning framework for neural audio fingerprinting. Original and augmented audio (e.g., audio with noise, reverb, time/pitch changes) are passed through a shared encoder, followed by a projection head. The resulting embeddings (z and z') are optimized using a contrastive loss to encourage invariance to audio degradations.

collection of audio distortions and manipulations, for both track-level and segment-level identification tasks.

Previous work on neural audio fingerprinting focuses on track-level evaluation with the Free Music Archive (FMA) dataset [14], using random splits of the same dataset for both training and testing. This setup offers only a limited view of real-world deployment, where query and reference databases can come from different distributions. To address this, we train and evaluate fingerprints on separate datasets to assess a fingerprinting model's robustness to subtle shifts in the data distributions. Fingerprinting experiments are conducted for two retrieval tasks: traditional *track-based* identification [5] and additionally *segment-based* identification, following the Pexeso Benchmark [15],a standardized open-source framework for evaluating fingerprint retrieval and temporal alignment under controlled degradations.

2 Methodology

This section describes our methodology for fingerprinting: datasets (Section 2.1), fingerprint representation learning (Section 2.2), data augmentations (Section 2.3), and inference algorithms (Section 2.4).

2.1 Datasets

For training, we use 300,000 samples from the Disco-10M dataset [16], ensuring no overlap with the FMA dataset [14]. For the *track-based* evaluation, 5,000 reference tracks are drawn from FMA, with 1,000 of these used to generate query files by extracting random 10-second segments and applying a range of audio distortions (see Section 2.3). For the *segment-based* evaluation, we follow the Pexeso Audio Fingerprinting Benchmark Toolkit [15] using the `pexafb_hard_small` and `pexafb_hard_medium` difficulty levels, which provide reference sets of 99 and 953 files and corresponding query sets of 100 and 1,000 files. In this setting, queries are constructed by concatenating one or more 10-second segments, each modified with distortions such as time streching, pitch shifts, echo, reverb, filtering, or noise, and concatenated using techniques such as fades or overlaps.

2.2 Models

We evaluate a unified fingerprinting architecture consisting of a pretrained encoder followed by a non-linear projection head. Given an embedding $\mathbf{x} \in \mathbb{R}^{d_{\text{in}}}$ from the backbone ($d_{\text{in}} = 1024$), we project $\mathbf{x}$ to $\mathbf{z} \in \mathbb{R}^{d_{\text{out}}}$ ($d_{\text{out}} = 256$) using a two-layer MLP (Projection Head in Fig 1).

$$\mathbf{z} \;=\; \mathbf{W}_2\,\phi(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2,$$

where $\mathbf{W}_1 \in \mathbb{R}^{d_h \times d_{\text{in}}}$, $\mathbf{b}_1 \in \mathbb{R}^{d_h}$, $\mathbf{W}_2 \in \mathbb{R}^{d_{\text{out}} \times d_h}$, $\mathbf{b}_2 \in \mathbb{R}^{d_{\text{out}}}$, $d_h = 4096$, and ϕ is ELU [17] nonlinearity. In early ablations, we found that this two-layer projection head with non-linearity outperforms the linear projection head used in prior works such as NAFP and GraFPrint; use of a large hidden width d_h further improves performance.

For the backbone, we consider three foundation models. MuQ [10] is a self-supervised music representation model based on masked token prediction with Conformer blocks [18], designed to

capture both local and long-range musical structure. MERT [11] adapts masked language modeling to musical audio using a convolutional front-end and Transformer encoder to jointly predict masked acoustic tokens. BEATs [12] is a general-purpose audio foundation model pretrained on AudioSet-2M [19]. For MuQ and MERT, we set the learning rate to 3×10^{-5}, while BEATs is trained with 5×10^{-5}. We compare these methods against NAFP [5], which employs a convolutional backbone trained from scratch with contrastive learning, and GraFPrint [7], which extends NAFP with a graph neural network (also trained from scratch). Dejavu [13] serves as a Shazam-like baseline that extracts constellation maps of spectral peaks, encodes them as landmark-based hashes, and retrieves matches using hash-table lookups.

2.3 Data Augmentations

We employ a range of augmentations during both training and evaluation to improve robustness against common acoustic and signal-domain variations. Temporal modifications include *time streching* (uniform in $[0.7, 1.5]$) and *pitch-shifting* (uniform in $[-5, 5]$ semitones), applied individually or sequentially. Additive noise augmentation uses $\sim$6 hours of MUSAN [20] recordings (restaurant, home, street) at varying SNRs, optionally combined with reverberation simulated using RIRs from the Aachen database [21] (RIR SNR in $[0.1, 1.5]$). Spectral filtering is applied using band-pass (300–1800 Hz), high-pass ($[1800, 3400]$ Hz), or low-pass ($[300, 1500]$ Hz) filters to emulate telephony, bandwidth-limited playback, or lo-fi effects. Echo is introduced with delays in $[100, 200]$ ms, while low-bitrate artifacts are simulated using Encodec [22] at 6-bit and 12-bit quantization (24 kHz model).

2.4 Inference

For track-level retrieval, we follow a conventional inference procedure. Query and reference tracks are passed through the trained model to generate embeddings, which serve as audio fingerprints. Reference fingerprints form a database and queries are matched using FAISS [23],an efficient library for approximate nearest-neighbour search over large embedding databases. A query is counted as correct if this retrieved reference matches the ground-truth reference for that query.

The segment-level retrieval task defined by the Pexeso benchmark is a novel setting for fingerprinting, where queries are constructed from multiple snippets originating from different reference tracks. This requires a new approach to inference that both identifies matches and localizes them with temporal alignment. To address this, we obtain the top-5 FAISS neighbors per query segment, filter out candidates below a fixed similarity threshold (0.7), and group the remaining matches by (query file, reference file). Within each group, every retained match gives two numbers: the segment's start time in the query and the corresponding start time in the reference. We then fit a linear model $t_{\mathrm{ref}} \approx a\, t_{\mathrm{qry}} + b$ using Huber regression [24], which down-weights outliers. The parameter a is a *time-scaling factor* that captures a uniform speed discrepancy between the query and the reference. When $a = 1$, the two run at the same speed, while $a > 1$ indicates the query runs more slowly.

The inlier matches within each $(\text{query}, \text{reference})$ group trace one or more candidate timing trajectories. For each group, we evaluate the trajectories generated from different seeds and keep the strongest one, giving priority to trajectories with more inliers and higher goodness of fit (larger R^2). The selected trajectory is then converted to segment boundaries by taking the earliest inlier as the start time and the latest inlier plus segment length as the end time on both query and reference segment boundaries. The resulting interval is taken as the aligned match and is scored against the ground-truth annotations.

3 Results & Discussion

Track-level retrieval results on the curated FMA dataset highlight clear differences between pretrained backbones, neural baselines, and the Shazam-like system. As shown in Table 1, the pretrained backbones (MuQ, MERT, and BEATs) consistently surpass state-of-the-art models trained from scratch and classical methods, highlighting the value of pretraining. For the music foundation models (MuQ and MERT), we show two settings: frozen (encoder weights fixed) and unfrozen (encoder fine-tuned). In the case of BEATs, we evaluate with the unfrozen version. The unfrozen MuQ model achieves the highest overall accuracy, clearly outperforming all other models, and is particularly robust to Encodec compression, a setting where other models struggle. However, relative to their own

Table 1: Top-1 hit rate (%) on track-level evaluation. T+P denotes both time stretch and pitch shift applied, R+N denotes reverb and noise combinations, B.P. denotes band-pass filtering, H.P. denotes high-pass filtering, L.P. denotes low-pass filtering, and Enc. denotes Encodec compression.

Model	Time	Pitch	T+P	Noise	Reverb	R+N	B.P.	H.P.	L.P.	Echo	Enc.	Overall
MuQ-Unfrozen	96	94	**87**	**97**	**100**	**90**	**63**	**73**	74	**100**	**96**	**88.18**
MuQ-Frozen	90	91	86	90	98	84	60	72	69	93	90	83.91
MERT-Unfrozen	**100**	92	81	87	98	78	32	35	70	**100**	44	74.27
MERT-Frozen	97	89	81	86	95	71	30	29	68	96	38	70.91
BEATs-Unfrozen	84	89	73	84	91	77	27	39	76	**100**	33	70.27
GraFPrint	58	**97**	67	90	84	80	15	47	95	96	17	67.82
NAFP	39	91	55	84	86	78	18	42	**96**	99	10	63.45
Dejavu	25	91	12	71	80	52	9	12	87	99	3	49.18

Table 2: Segment-level evaluation (F1 scores, %) on Pexeso benchmark (hard settings). BBox refers to bounding-box.

Model	Pex-Hard-Small			Pex-Hard-Medium		
	Track F1	BBox F1	Length F1	Track F1	BBox F1	Length F1
MuQ-Unfrozen	**95.5**	**86.4**	**90.8**	**87.3**	**74.7**	**81.4**
MuQ-Frozen	91.1	83.0	88.1	84.8	73.8	80.1
MERT-Unfrozen	92.47	80.20	86.3	85.55	71.00	78.85
MERT-Frozen	84.10	71.75	70.88	80.75	59.10	69.10
BEATs-Unfrozen	80.8	73.10	76.4	85.2	70.2	76.70
GraFPrint	78.00	49.40	67.60	81.30	61.90	70.40
NAFP	77.47	40.50	66.41	80.6	57.4	66.8
Dejavu	67.8	40.2	63.11	73.2	58.4	68.8

performance across augmentations, both MuQ and MERT show reduced accuracy for filtering-based augmentations. Interestingly, the NAFP model outperforms all the other models in case of low pass filtering augmentation by a significant margin. The causes of this sensitivity will be investigated in future work. Across all conditions, neural approaches consistently surpass the Shazam-like baseline implemented with Dejavu.

Segment-level evaluation results on the Pexeso benchmark further confirm the contrast between the pretrained backbones and neural baselines. Table 2 shows that the unfrozen MuQ model achieves the highest scores across all metrics, namely **track-level F1** (reference retrieval), **length-level F1** (alignment accuracy), and **bounding-box F1** (segment boundary precision [25]).Consistent with the track-level results, MuQ, MERT, and BEATs outperform NAFP and GraFPrint by a clear margin. Interestingly, NAFP and GraFPrint improve from the smaller to the medium-sized dataset, while pretrained backbones slightly decline, highlighting a scalability contrast that warrants further investigation in large-scale fingerprinting scenarios

4 Conclusion

This work presents a systematic evaluation of self-supervised music foundation models (MuQ, MERT) and a general-purpose audio foundation model (BEATs) against state-of-the-art neural audio fingerprinting approaches (NAFP, GraFPrint), under a broad set of audio modifications at both track and segment levels. Models with pretrained backbones consistently outperform those trained from scratch, showing superior robustness and generalization, especially under challenging conditions. Segment-level evaluation further highlights their ability to accuratsly localize matched regions, an important capability for large-scale catalog management. Our findings suggest that pretrained music foundation models can serve as powerful backbones for audio fingerprinting, but also reveal weaknesses in handling certain transformations, such as spectral filtering. Future work will explore targeted augmentation strategies to address these weaknesses and extend the evaluation to include new types of adversarial audio changes that are intentionally used to avoid detection on modern content-sharing platforms.

References

[1] R Oguz Araz, Guillem Cortès-Sebastià, Emilio Molina, Joan Serrà, Xavier Serra, Yuki Mitsufuji, and Dmitry Bogdanov. Enhancing neural audio fingerprint robustness to audio degradation for music identification. *arXiv e-prints*, pages arXiv–2506, 2025.

[2] Avery Wang. An industrial-strength audio search algorithm. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 7–13, 2003.

[3] Emilia Gomez, Pedro Cano, L Gomes, Eloi Batlle, and Madeleine Bonnet. Mixed watermarking-fingerprinting approach for integrity verification of audio recordings. In *Proceedings of the International Telecommunications Symposium*, 2002.

[4] Guillem Cortès, Alex Ciurana, Emilio Molina, Marius Miron, Owen Meyers, Joren Six, and Xavier Serra. Baf: an audio fingerprinting dataset for broadcast monitoring. 2022.

[5] Sungkyun Chang, Donmoon Lee, Jeongsoo Park, Hyungui Lim, Kyogu Lee, Karam Ko, and Yoonchang Han. Neural audio fingerprint for high-specific audio retrieval based on contrastive learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3025–3029. IEEE, 2021.

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[7] A. Bhattacharjee et al. GraFPrint: A GNN-based approach for audio identification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

[8] C. Nikou et al. Contrastive and transfer learning for effective afp. *International Journal of Multimedia and Signal Technologies and Applications (IJMSTA)*, 2025.

[9] David. Universal sues for tempo shifting and mixing its songs. Music licensing blog post on ClicknClear, November 2024. published online.

[10] Haina Zhu, Yizhi Zhou, Hangting Chen, Jianwei Yu, Ziyang Ma, Rongzhi Gu, Yi Luo, Wei Tan, and Xie Chen. Muq: Self-supervised music representation learning with mel residual vector quantization. *CoRR*, 2025.

[11] Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghao Xiao, Chenghua Lin, Anton Ragni, Emmanouil Benetos, Norbert Gyenge, Roger Dannenberg, Ruibo Liu, Wenhu Chen, Gus Xia, Yemin Shi, Wenhao Huang, Zili Wang, Yike Guo, and Jie Fu. MERT: Acoustic music understanding model with large-scale self-supervised training. In *The Twelfth International Conference on Learning Representations*, 2024.

[12] Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, Wanxiang Che, Xiangzhan Yu, and Furu Wei. BEATs: Audio pre-training with acoustic tokenizers. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 5178–5193. PMLR, 23–29 Jul 2023.

[13] Will Drevo. Dejavu: open-source audio fingerprinting project. GitHub, 2014. `https://github.com/worldveil/dejavu`.

[14] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. FMA: A dataset for music analysis. In *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[15] Pexeso Team. Audio fingerprinting benchmark toolkit. GitHub, 2025. `https://github.com/Pexeso/audio-fingerprinting-benchmark-toolkit`.

[16] Luca Lanzendörfer, Florian Grötschla, Emil Funke, and Roger Wattenhofer. Disco-10m: A large-scale music dataset. *Advances in Neural Information Processing Systems*, 36:54451–54471, 2023.

[17] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 4(5):11, 2015.

[18] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

[19] Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 776–780. IEEE, 2017.

[20] David Snyder, Guoguo Chen, and Daniel Povey. Musan: A music, speech, and noise corpus. *arXiv preprint arXiv:1510.08484*, 2015.

[21] Marco Jeub, Magnus Schafer, and Peter Vary. A binaural room impulse response database for the evaluation of dereverberation algorithms. In *2009 16th International Conference on Digital Signal Processing*, pages 1–5. IEEE, 2009.

[22] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *Transactions on Machine Learning Research*.

[23] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[24] Peter J Huber. Robust estimation of a location parameter. In *Breakthroughs in statistics: Methodology and distribution*, pages 492–518. Springer, 1992.

[25] Sifeng He, Xudong Yang, Chen Jiang, Gang Liang, Wei Zhang, Tan Pan, Qing Wang, Furong Xu, Chunguang Li, JinXiong Liu, et al. A large-scale comprehensive dataset and copy-overlap aware evaluation protocol for segment-level video copy detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21086–21095, 2022.