Interspeech 2024
1-5 September 2024, Kos, Greece

Audio Fingerprinting with Holographic Reduced Representations

Yusuke Fujita, Tatsuya Komatsu

LY Corporation, Tokyo, Japan
yusuke.fujita@lycorp.co.jp

Abstract

This paper proposes an audio fingerprinting model with holographic reduced representation (HRR). The proposed method reduces the number of stored fingerprints, whereas conventional neural audio fingerprinting requires many fingerprints for each audio track to achieve high accuracy and time resolution. We utilize HRR to aggregate multiple fingerprints into a composite fingerprint via circular convolution and summation, resulting in fewer fingerprints with the same dimensional space as the original. Our search method efficiently finds a combined fingerprint in which a query fingerprint exists. Using HRR's inverse operation, it can recover the relative position within a combined fingerprint, retaining the original time resolution. Experiments show that our method can reduce the number of fingerprints with modest accuracy degradation while maintaining the time resolution, outperforming simple decimation and summation-based aggregation methods.

Index Terms: audio fingerprinting, contrastive learning, holographic reduced representation

1. Introduction

Audio fingerprinting identifies a song within a database using a segment of an audio signal as a query. Applications of audio fingerprinting include identifying a user's unknown songs from a microphone input, finding duplicated music in a database, and checking copyrights. Peak-based matching [1], which detects peaks in a spectrogram and encodes their relative positions using hash functions, has traditionally been widely used. Various approaches to extract more discriminative and robust features than spectrogram peaks have been studied [2–6]. Most approaches use binary hashing functions for efficient search with hamming distance. Although the hash-based fingerprint is efficient, noise or distortions in the query audio affect the feature extraction performance, leading to incorrect fingerprints.

Neural-network-based fingerprinting methods, which learn to generate robust embeddings against noise, have advanced the field. Now Playing [7] uses a neural network trained with a semi-hard triplet loss function, which minimizes the distance between the reference audio segment and their noisy version while maintaining their distances to other audio segments larger. Neural audio fingerprinting (NAFP) [8] further exploits an advanced contrastive learning framework and extracts fingerprints with small window shifts (e.g., 0.5 sec), leading to better search accuracy while precisely determining the matched position within a song.

In exchange for better accuracy with high time resolution, NAFP requires significantly larger storage than traditional hash-based fingerprinting because the fingerprint is a real-valued vector of hundreds of dimensions. Hashing-based embedding mappings, such as Locality-Sensitive Hashing (LSH) [9], and vector quantization methods like Product Quantization (PQ) [10] reduce storage and improve computational efficiency by aggregating the similarity calculations of multiple, partially similar embeddings with a query in a single computation. In particular, PQ is widely used in general maximum inner-product search (MIPS) systems.

However, current approaches to improve MIPS do not reduce the number of fingerprints that must be searched, which could be considered another dimension of efficiency. If we could represent a group of fingerprints, e.g., in the same audio track, as another *composite* fingerprint, the number of fingerprints can be reduced. Moreover, we could efficiently handle a *containment* search query like "find a group of fingerprints in which a query fingerprint exists." Though any simple aggregation operation, such as decimation or summation within a group, could reduce the number of fingerprints, it leads to a loss of accuracy and time resolution. Our motivation is to find an appropriate aggregation operation that maintains both accuracy and time resolution.

In this paper, we propose a method to reduce the number of fingerprints by utilizing holographic reduced representations (HRRs). HRR [11] is a representation of a compositional structure in distributed representations. With HRR, circular convolution (denoted by $\circledast$) binds two items, and summation integrates the bounded items in the same vector space. An illustrative example of HRR is shown in [12]. According to the example, one can compose a sentence like $\boldsymbol{s} = red \circledast cat + blue \circledast dog$ to represent the co-existence of a red cat and a blue dog, where red, cat, $blue$, dog are item vectors in the same dimensional space. Then, one can retrieve the cat's color with the inverse operation, $\boldsymbol{s} \circledast cat^{\dagger} \approx red$ under some assumptions in the associate vectors. Our proposed method uses this composition scheme to group a sequence of fingerprints. Each fingerprint is associated with its relative position in a sequence using circular convolution, and then the results are summed together to obtain a composite fingerprint. It reduces the number of fingerprints stored in the database, while we can retrieve the original fingerprint location through the inverse operation.

We conducted fingerprint search experiments using the FMA dataset [13]. We followed a similar setup used in the NAFP paper [8]. The experimental results show that the proposed method can aggregate fingerprints with a slight accuracy degradation compared with non-reduced fingerprints. It outperforms simple decimation-based and summation-based aggregation methods, which make it hard to recover the original fingerprint location within a sequence. Though our proposed method can work with any pretrained fingerprinter such as NAFP, we further explored the possibility of using HRR-aware training for a neural fingerprinter. A similar training strategy with HRR has

10.21437/Interspeech.2024-245

been proposed for extreme multi-label classification in [14]. We are believed to be among the first to apply HRR-based training to contrastive learning. These additional experiments exhibit that considering the HRR's noise in the training does not offer a significant improvement. Finally, we discuss the limitations and future work based on the results of HRR-aware training.

2. Background

2.1. Neural Audio fingerprinting

NAFP [8] has introduced the contrastive learning framework to extract a fingerprint for short audio segments. A neural-network-based function f transforms T-length audio feature sequence $\boldsymbol{A} \in \mathbb{R}^{F \times T}$ into an fingerprint $\boldsymbol{x} \in \mathbb{R}^D$:

$$\boldsymbol{x} = f(\boldsymbol{A}). \tag{1}$$

A replica audio is prepared for each audio segment $\boldsymbol{A}$ in a training set with various augmentations, and the fingerprint $\boldsymbol{r}$ for the replica is produced:

$$\boldsymbol{r} = f(\mathsf{Aug}(\boldsymbol{A})). \tag{2}$$

We train the fingerprint function f with contrastive learning. Given a training batch of B fingerprints $\boldsymbol{X} = [\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(B)}]$ and their replicas $\boldsymbol{R} = [\boldsymbol{r}^{(1)}, \ldots, \boldsymbol{r}^{(B)}]$, a contrastive loss is calculated as follows:

$$\mathcal{L}(\boldsymbol{R}, \boldsymbol{X}) = -\sum_{i=0}^{B} \log \frac{\exp(\mathsf{sim}(\boldsymbol{x}^{(i)}, \boldsymbol{r}^{(i)})/\tau)}{\sum_{j=1}^{B} \exp(\mathsf{sim}(\boldsymbol{x}^{(j)}, \boldsymbol{r}^{(i)})/\tau)}, \tag{3}$$

where τ is a temperature hyperparameter, cosine-similarity is used as a similarity measure sim. Since this contrastive loss only considers mapping from $\boldsymbol{R}$ to $\boldsymbol{X}$, we also calculate the loss in the reverse direction to encourage one-to-one correspondence between $\boldsymbol{R}$ and $\boldsymbol{X}$:

$$\mathcal{L}_{\mathsf{NAFP}} = (\mathcal{L}(\boldsymbol{R}, \boldsymbol{X}) + \mathcal{L}(\boldsymbol{R}, \boldsymbol{X}))/2. \tag{4}$$

2.2. Holographic reduced representations

HRR [11] uses circular convolution to *bind* two vectors.

$$\boldsymbol{a} \circledast \boldsymbol{b} = \mathcal{F}^{-1}(\mathcal{F}(\boldsymbol{a})\mathcal{F}(\boldsymbol{b})), \tag{5}$$

where $\mathcal{F}$ is the discrete Fourier transform. Then, summation can *bundle* multiple vectors into a single composite vector with the same dimension:

$$\boldsymbol{s} = \boldsymbol{a} \circledast \boldsymbol{b} + \boldsymbol{c} \circledast \boldsymbol{d} \in \mathbb{R}^D. \tag{6}$$

Assuming that the elements of vectors are i.i.d. with zero mean and variance $1/D$, which is a reasonable assumption given that contrastive learning ensures uniformity in the learned representations [15], $\boldsymbol{a}$ can be recovered using the following inverse operation:

$$\hat{\boldsymbol{a}} = \boldsymbol{s} \circledast \boldsymbol{b}^{\dagger} = \mathcal{F}^{-1}(\mathcal{F}(\boldsymbol{s})\mathcal{F}(\frac{1}{\boldsymbol{b}})), \tag{7}$$

where $\boldsymbol{b}^{\dagger} = \mathcal{F}^{-1}(\mathcal{F}(\frac{1}{\boldsymbol{b}}))$. The recovered vector has noise due to other bundled vectors ($\boldsymbol{c}$ and $\boldsymbol{d}$). The capacity, the number of acceptable vectors to be bundled, increases linearly as the vector dimension D increases.

Note that we do not need to use the inverse operation to check if the vector is in a composite vector. We can ask if two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$) are bounded and bundled in $\boldsymbol{s}$ by checking that $(\boldsymbol{a} \circledast \boldsymbol{b})^{\top} \boldsymbol{s} \approx 1$. We use this property for efficient fingerprint search.

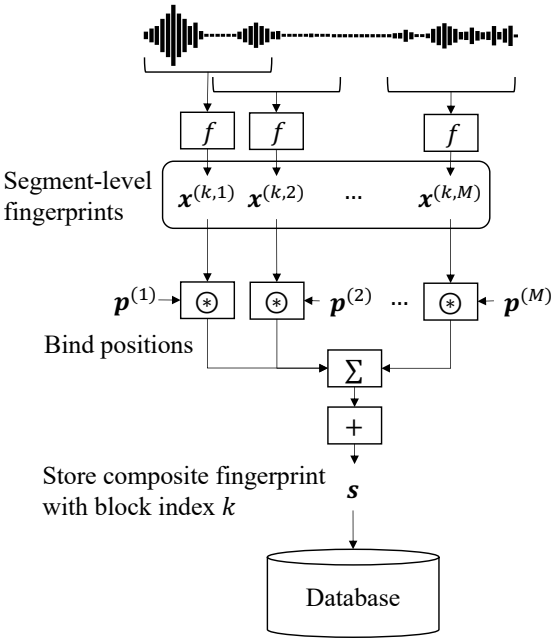

Figure 1: *Composition of fingerprints*

3. Proposed method

In general fingerprinting methods, such as NAFP, many fingerprints are handled independently. Contrary to such a trend, we attempt to aggregate a sequence of audio fingerprints utilizing the compositional structure of HRR. As described in the introduction, HRR can represent a composite sentence "a red cat and blue dog" as a vector $\boldsymbol{s} = red \circledast cat + blue \circledast dog$. Similarly, the proposed method considers a structure of fingerprints; in this study, a "sequence" structure is encoded using HRR. We empirically demonstrate that the HRR-based fingerprint enables us to perform the containment search to determine whether the query is bounded in the composite representation of a sequence.

3.1. Composition of fingerprint sequence with HRR

The proposed composition method is depicted in Fig. 1.

We initialize M position vectors $\boldsymbol{p}^{(1)}, \ldots, \boldsymbol{p}^{(M)} \in \mathbb{R}^D$. A sequence of N fingerprints $\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(N)} \in \mathbb{R}^D$ in the audio database is first segmented into blocks, each with M consecutive fingerprints. We bind a sequence of fingerprints for each block with the position vectors, resulting in a composite fingerprint $\boldsymbol{s}^{(k)}$:

$$\boldsymbol{s}^{(k)} = \sum_{m=1}^{M} \boldsymbol{x}^{(k,m)} \circledast \boldsymbol{p}^{(m)} \quad (1 \leq k \leq \lceil N/M \rceil), \tag{8}$$

where k is a block index and $\boldsymbol{x}^{(k,m)} = \boldsymbol{x}^{((k-1)M+m)}$ is the m-th fingerprint in the k-th block. We only store the composite fingerprint instead of all M fingerprints, which requires M times smaller storage.

3.2. Search composite fingerprint with positions

Fig. 2 shows our search method for composite fingerprints. Given a fingerprint $\boldsymbol{q} \in \mathbb{R}^D$ extracted from a query audio, we

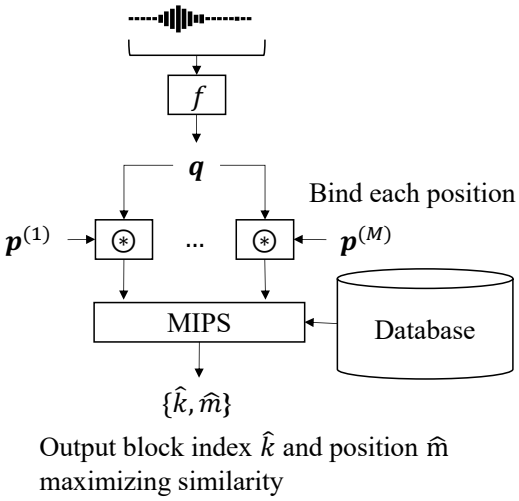

Figure 2: *Search composite fingerprint with maximum inner-product search*

search a block in which $\boldsymbol{q}$ exists:

$$\hat{k}, \hat{m} = \arg\max_{k,m} \mathsf{sim}(\boldsymbol{q} \circledast \boldsymbol{p}^{(m)}, \boldsymbol{s}^{(k)}) \quad (1 \le m \le M). \quad (9)$$

Here, we can use an efficient K-nearest neighbor search algorithm for MIPS and produce top-K block indices for each m. Then, we easily obtain a relative position $\hat{m}$ in the retrieved block $\hat{k}$ having the maximum similarity score.

The proposed composition method preserves the distinction between positions, unlike simple decimation or summation operations. We compare the composition operations in the experiment section.

3.3. Search for a sequence of fingerprints

When query audio is longer than T (one segment), the system can gather similarity scores for multiple consecutive segments to improve search accuracy. Given L consecutive query fingerprints $\boldsymbol{q}^{(1)}, \ldots, \boldsymbol{q}^{(L)}$, we construct a sequence of composite queries as follows:

$$\boldsymbol{q}'^{(i)} = \sum_{m=1}^{M} \boldsymbol{q}^{(i+m-1)} \circledast \boldsymbol{p}^{(m)} \quad (1 \le i \le L - M + 1). \quad (10)$$

Then, we find top-K similar blocks for each $\boldsymbol{q}'^{(i)}$. The offset in the retrieved block index $\hat{k}$ for i-th query is compensated by $\hat{k} - i$. The sequence-level similarity score is the sum of all similarity scores assigned to the same block index, and finally the system outputs the index with the highest score.

3.4. HRR-aware training of neural fingerprinter

Although the proposed method can work with any pretrained fingerprinter, this section further investigates the possibility of learning with HRR's characteristics.

As described in Sec 2.2, the recovered vectors from HRR have noise, leading to degraded search performance. To mitigate the issue, we train the fingerprint function f to be aware of the HRR's noise.

Given a training batch of B fingerprints $\boldsymbol{X}$ and their replicas $\boldsymbol{R}$, we first generate a batch of composite fingerprints $\boldsymbol{S} = [\boldsymbol{s}^{(1)}, \ldots, \boldsymbol{s}^{(B/M)}]$ using Eq. 8. The replicas are bounded with position vectors aligned with the batch of composite fingerprints. A batch of the bounded vectors $\boldsymbol{V} = [\boldsymbol{v}^{(1)}, \ldots, \boldsymbol{v}^{(B)}]$ is calculated as:

$$\boldsymbol{v}^{(b)} = \boldsymbol{r}^{(b)} \circledast \boldsymbol{P}_{b \bmod M} \quad (1 \le b \le B). \quad (11)$$

Then, we compute the following contrastive loss instead of Eq. 3:

$$\mathcal{L}'(\boldsymbol{V}, \boldsymbol{S}) = -\sum_{b=1}^{B} \log \frac{\exp(\mathsf{sim}(\boldsymbol{s}^{(\lceil b/M \rceil)}, \boldsymbol{v}^{(b)}))}{\sum_{k=1}^{B/M} \exp(\mathsf{sim}(\boldsymbol{s}^{(k)}, \boldsymbol{v}^{(b)}))} \quad (12)$$

Unlike the original loss $\mathcal{L}$, the modified loss considers noise added to the composite fingerprint and maximizes the similarity only if a query is placed at the specified position. The loss for mapping from $\boldsymbol{S}$ to $\boldsymbol{V}$ is slightly different because M consecutive vectors in $\boldsymbol{V}$ should be mapped to one composite fingerprint in $\boldsymbol{S}$. To force one-to-one mapping in the loss calculation, we split V according to their positions:

$$\mathcal{L}''(\boldsymbol{S}, \boldsymbol{V}) = -\sum_{m=1}^{M} \sum_{k=1}^{B/M} \log \frac{\exp(\mathsf{sim}(\boldsymbol{s}^{(k)}, \boldsymbol{v}^{(k,m)}))}{\sum_{k'=1}^{B/M} \exp(\mathsf{sim}(\boldsymbol{s}^{(k)}, \boldsymbol{v}^{(k',m)}))}, \quad (13)$$

where $\boldsymbol{v}^{(k,m)} = \boldsymbol{v}^{((k-1)M+m)}$. Then, we mix the two losses:

$$\mathcal{L}_{\mathsf{NAPF-HRR}} = (\mathcal{L}''(\boldsymbol{S}, \boldsymbol{V}) + \mathcal{L}''(\boldsymbol{V}, \boldsymbol{S}))/2. \quad (14)$$

4. Experimental setup

4.1. Data

We conducted audio fingerprinting experiments on the FMA dataset [13] according to the NAFP paper [8]. Note that the dataset described below is the *mini* version and can be downloaded from [16], which is different from the *full* version reported in the paper [8].

The training data set for the fingerprinting function is sampled from `fma_medium`, comprising 10,000 songs, each with 30-second audio. The test-DB dataset is another subset from `fma_medium`, comprising 500 songs of 30 seconds each. The test-query dataset is a noisy copy of the test DB with a random augmentation pipeline, including time offset modulation up to $\pm$200 ms, background noise mixing using AudioSet [17] in the SNR range from 0 to 10 dB, and impulse response convolution using two public datasets [18, 19]. The test-dummy-DB dataset is used as a set of distractors; they should not be matched to the test query. The dummy dataset is sampled from `fma_full`, consisting of 10,000 songs, each with 30 seconds.

4.2. Network architecture and training configurations

We also used the same network architecture with NAFP [8]. The network accepts a log-scaled Mel-spectrogram representing 1-second audio with the 0.5-second shift. The input runs through eight convolutional encoders with separable convolution, layer normalization, and ReLU activation, followed by a projection layer and L2-normalization. We mainly used the fingerprint dimension $D = 512$, larger than $D = 128$ reported in [8], because our HRR requires a sufficient dimension to bind multiple vectors. We set the batch size B to 640. We trained the network using Adam optimizer with an initial learning rate of 1e-4 and cosine decay to 1e-6 in 100K steps. The temperature τ was set to 0.05. We implemented the training pipeline by ourselves with PyTorch.

Table 1: *Top-1 hit rate (%) with different aggregation methods. The fingerprint dimension is 512. M is the block size for aggregation.*

Method	M	Query length (s) 1	2	3	5	10
No-aggregation	1	71.1	90.4	95.1	97.9	99.4
Summation	2	31.2	74.2	84.9	92.7	96.4
Decimation	2	39.0	74.0	86.0	93.4	95.9
HRR (proposed)	2	**58.8**	**83.8**	**90.9**	**95.1**	**97.8**
Summation	4	3.0	6.6	39.4	44.3	89.8
Decimation	4	19.3	45.8	37.7	71.3	92.8
HRR (proposed)	4	**31.0**	**58.3**	**45.0**	**79.0**	**96.0**

For training with HRR described in Sec. 3.4, we tested different numbers of positions $M = 2, 4$

4.3. Search algorithm

Faiss [20] is used for efficient MIPS. We used the inverted file index structure with PQ (IVF-PQ). For the IVF-PQ, we had 200 centroids with a code size of 64 and 8 bits per index.

4.4. Evaluation protocol

We use the Top-1 hit rate (%) to measure the search performance. Assuming we have Q query fingerprints and Q_{hit} queries with the maximum similarity are hit as top-1, the Top-1 hit rate can be measured as Q_{hit}/Q.

5. Results and discussion

5.1. Comparison of fingerprint aggregation methods

The proposed aggregation method **HRR** (Eq. 8) was compared with two simple alternatives, 1) **Summation**: $\boldsymbol{s}^{(k)} = \sum_{m=1}^{M} \boldsymbol{x}^{(k,m)}$, and 2) **Decimation**: $\boldsymbol{s}^{(k)} = \boldsymbol{x}^{(k,1)}$.

Table 1 shows the results on the Top-1 hit rate. We observed that for all query lengths and block sizes, the proposed HRR outperformed other aggregation methods. In particular, when query length is 1-sec, i.e., one segment, HRR produced significantly better accuracy than the summation and decimation methods. Unlike the other methods, it demonstrates that HRR can preserve the original time resolution. Although we can see significant performance degradation compared with the no-aggregation system, the proposed method can recover the accuracy according to the query length.

Table 2 shows the Top-1 *near* match rate for the 1-second query. The near match means that the Top-1 hypothesis is within ± 500 msec. With $M = 2$, the summation method can produce a better Top-1 near match, suggesting the summation does reasonable aggregation while ignoring the position in a sequence. With $M = 4$, the summation and decimation methods failed even for near matches. The proposed HRR method showed no significant difference between the Top-1 exact and Top-1 near values. It suggests that HRR can accurately discriminate the position in a sequence.

5.2. Effect of HRR-aware training

Table 3 shows the results of HRR-aware training. For $M = 2$, HRR-aware training was slightly better than that without HRR-aware training. However, the difference was not significant. For $M = 4$, HRR-aware training was only slightly better at the query length of 10. We hypothesize that the linear operations of HRR limit the capacity of representations. Adding some non-linear operations for aggregation could lead to an improvement in the proposed training scheme. We leave this direction for future work.

Table 2: *Comparison of Top-1 exact/near match rate for 1-second query with different aggregation methods. The fingerprint dimension is 512. M is the block size for aggregation.*

Method	M	Top-1 exact	Top-1 near
Summation	2	31.2	**62.9**
Decimation	2	39.0	45.0
HRR (proposed)	2	**58.8**	60.4
Summation	4	3.0	7.7
Decimation	4	19.3	22.9
HRR (proposed)	4	**31.0**	**32.4**

Table 3: *Top-1 hit rate (%) with and without HRR-aware training. The fingerprint dimension is 512. M is the block size for aggregation.*

Method	M	Query length (s) 1	2	3	5	10
HRR	2	58.8	83.8	90.9	95.1	97.8
+ HRR-aware train.	2	59.0	83.8	91.5	95.4	98.1
HRR	4	31.0	58.3	45.0	79.0	96.0
+ HRR-aware train.	4	24.4	57.0	44.5	67.4	97.0

6. Conclusion

We proposed an audio fingerprinting model with holographic reduced representation (HRR). The proposed method can reduce the number of stored fingerprints by utilizing HRR to aggregate multiple fingerprints into a composite fingerprint. We conducted fingerprint search experiments using the FMA dataset. The results show that the proposed method can aggregate fingerprints with a slight accuracy degradation compared with non-reduced fingerprints. It significantly outperformed decimation-based and summation-based aggregation methods. While the baseline aggregation methods make it hard to recover the original fingerprint position within a sequence, the proposed HRR-based aggregation successfully preserved it. Experiments with the HRR-aware training of the neural fingerprinting model did not show an improvement.

This study paves the way for several areas of future research. Investigating alternative methods for aggregating fingerprints that reduce storage while maintaining accuracy is a promising direction. Additionally, evaluating the proposed HRR-based method on larger and more diverse datasets would help assess its scalability and applicability. Furthermore, refining HRR parameters, such as vector dimensionality and the number of position vectors, could enhance the search accuracy.

7. References

[1] A. Wang, "An industrial strength audio search algorithm." in *ISMIR*, 2003.

[2] P. Cano and E. Batlle, "A review of audio fingerprinting," *Journal*

of VLSI Signal Processing, vol. 41, pp. 271–284, 11 2005.

[3] S. Baluja and M. Covell, "Waveprint: Efficient wavelet-based audio fingerprinting," *Pattern Recognition*, 2008.

[4] T.-K. Hon, L. Wang, J. D. Reiss, and A. Cavallaro, "Audio fingerprinting for multi-device self-localization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 10, pp. 1623–1636, 2015.

[5] Y. Jiang, C. Wu, K. Deng, and Y. Wu, "An audio fingerprinting extraction algorithm based on lifting wavelet packet and improved optimal-basis selection," *Multimedia Tools and Applications*, vol. 78, no. 21, pp. 30 011–30 025, 2019.

[6] X. Wu and H. Wang, "Asymmetric contrastive learning for audio fingerprinting," *IEEE Signal Processing Letters*, vol. 29, pp. 1873–1877, 2022.

[7] B. A. y Arcas, B. Gfeller, R. Guo, K. Kilgour, S. Kumar, J. Lyon, J. Odell, M. Ritter, D. Roblek, M. Sharifi, and M. Velimirović, "Now playing: Continuous low-power music recognition," in *NeurIPS*, 2017.

[8] S. Chang, D. Lee, J. Park, H. Lim, K. Lee, K. Ko, and Y. Han, "Neural audio fingerprint for high-specific audio retrieval based on contrastive learning," in *ICASSP*, 2021, pp. 3025–3029.

[9] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proceedings of the 25th International Conference on Very Large Data Bases*, 1999, p. 518–529.

[10] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 1, pp. 117–128, 2011.

[11] T. Plate, "Holographic reduced representations," *IEEE Transactions on Neural Networks*, vol. 6, no. 3, pp. 623–641, 1995.

[12] M. Nickel, L. Rosasco, and T. Poggio, "Holographic embeddings of knowledge graphs," in *AAAI*, 2016, p. 1955–1961.

[13] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," in *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017. [Online]. Available: https://arxiv.org/abs/1612.01840

[14] A. Ganesan, H. Gao, S. Gandhi, E. Raff, T. Oates, J. Holt, and M. McLean, "Learning with holographic reduced representations," in *Advances in Neural Information Processing Systems*, 2021.

[15] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *International conference on machine learning*. PMLR, 2020, pp. 9929–9939.

[16] S. Chang, "Neural audio fingerprint dataset," 2021. [Online]. Available: https://dx.doi.org/10.21227/ahym-e477

[17] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 776–780.

[18] Xaudia, "Microphone impulse response project," 2017. [Online]. Available: https://micirp.blogspot.com/

[19] M. Jeub, M. Schafer, and P. Vary, "A binaural room impulse response database for the evaluation of dereverberation algorithms," in *2009 16th International Conference on Digital Signal Processing*, 2009, pp. 1–5.

[20] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Transactions on Big Data*, vol. 7, no. 3, pp. 535–547, 2019.