Simultaneously Learning Robust Audio Embeddings and Balanced Hash Codes for Query-by-Example

ICASSP 2023

Anup Singh[1,2], Kris Demuynck[1], Vipul Arora[2]

[1]Ghent University, [2]Indian Institute of Technology-Kanpur

AUDIO FINGERPRINTING

- Audio fingerprinting generates content-based summary of an audio signal.
- Allows efficient storage and retrieval of similar audio in large audio database.
- Includes audio segmentation, feature extraction and indexing.

APPLICATIONS

Music Identification

Broadcast monitoring

Second screen applications

CHALLENGES

- **Robustness** against audio distortions such as noise and reverberation.
- **High efficiency** to expedite the retrieval process to enhance user experience.
- **Less complexity** for practical deployment.

PROBLEMS

- Conventional approaches rely on handcrafted audio features, which are not robust against high distortion levels - Robustness ✗
- Recent deep learning based approaches generate compact audio fingerprints; however, their performance degrade at high distortion levels. Also, it requires long queries to deliver reliable results - Efficacy ✗
- Previous approaches do not perform fine-grained audio search - Efficacy ✗
- No prior work addresses the indexing performance - Efficiency ✗
- Need a solution that improves all aspects of the system.

SOLUTION

- Contrastive learning framework for robust audio embeddings - Robustness ✔
- Transformer encoder that embeds contextual information to generate more discriminative audio embeddings than CNNs - Efficacy ✔
- Performs fine-grained audio search - Efficacy ✔
- Adopt optimal transport framework to optimize the indexing performance - Efficiency ✔
- Our indexing involves building a single hash table and lesser hash buckets probes for retrieval. Moreover, our indexing is less memory consuming - Complexity ✔

METHOD - AUDIO EMBEDDINGS

- We learn an encoder $\mathcal{F}_\theta$ using contrastive learning framework such that pairs of clean audio segment and its distorted counterpart, $\{x, x^+\}$, are closer in $L^2(\mathbb{R}^d)$ space.

$$\mathcal{L}_c = -\log \frac{e^{(\mathcal{F}_\theta(x)\cdot\mathcal{F}_\theta(x^+))/\tau}}{e^{(\mathcal{F}_\theta(x)\cdot\mathcal{F}_\theta(x^+))/\tau} + \sum_{x^-} e^{(\mathcal{F}_\theta(x)\cdot\mathcal{F}_\theta(x^-))/\tau}}$$

- $\mathcal{F}_\phi$ projects encodings into $L^2(\mathbb{R}^K)$ space as: $q(x) = \mathcal{F}_\theta(\mathcal{F}_\phi(x))$

$$\mathcal{L}_h = -\log \frac{e^{(q(x)\cdot q(x^+))/\tau}}{e^{(q(x)\cdot q(x^+))/\tau} + \sum_{x^-} e^{(q(x)\cdot q(x^-))/\tau}}$$

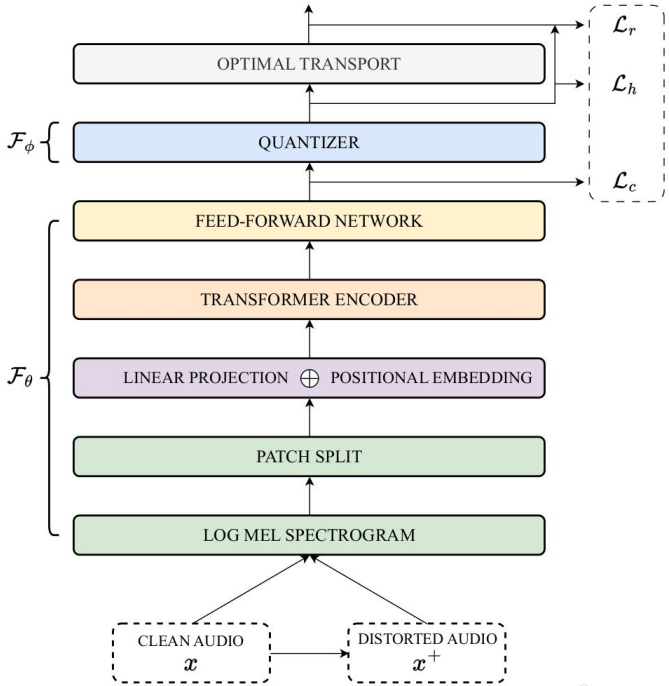

Figure 1: An overview of the training framework

METHOD - QUANTIZATION

- We propose quantizing q(x) as a clustering process.
- We consider all possible 2^K K-bits hash codes as fixed cluster centroids in $L^2(\mathbb{R}^K)$ space. $\tilde{h} \in [-1, 1]^K$
- We map q(x) to its nearest cluster centroid such that all q(x) is uniformly assigned to the centroids.
- To achieve robust hash codes, we formulate clustering process as:

$$\mathcal{L}_r = -\max_k \left[\text{softmax}\left(\frac{s_k(x)}{\tau'}\right) + \text{softmax}\left(\frac{s_k(x^+)}{\tau'}\right)\right],$$

$$\text{where } s_k(x) = q(x) \cdot \frac{\tilde{h}_k}{||\tilde{h}_k||_2}, \quad k = 1, 2, .., 2^K$$

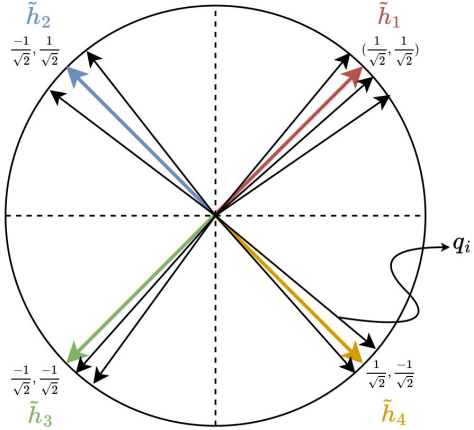

Figure 2: An illustration of clustering in 2-D space.

METHOD – BALANCED CLUSTERING

$$\max_A \ \mathbb{E}_x \Big[\sum_{k=1}^{2^K} A_k(x) s_k(x) \Big] \quad \text{subject to}$$

$$A_k(x) \in \{0,1\}, \ \sum_{k=1}^{2^K} A_k(x) = 1 \ \forall x \ , \ \mathbb{E}_x[A_k(x)] = \frac{1}{2^K} \ \forall k$$

➜ difficult to solve :(➜ Optimal Transport :)

$$\mathcal{L}_r = -\Big[\text{softmax}\Big(\frac{s_{k^*}(x)}{\tau'}\Big) + \text{softmax}\Big(\frac{s_{k^*}(x^+)}{\tau'}\Big)\Big]$$

$$\text{where } k^* = \text{argmax}_k \ A_k(x)$$

➜ Clustering objective redefined

METHOD - SUBSEQUENCE SEARCH

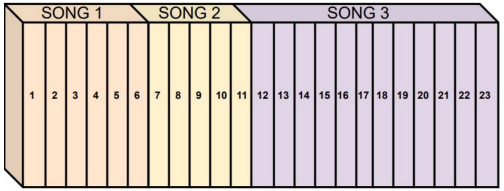

Figure 3: Sequential storage of audio embeddings in the database.

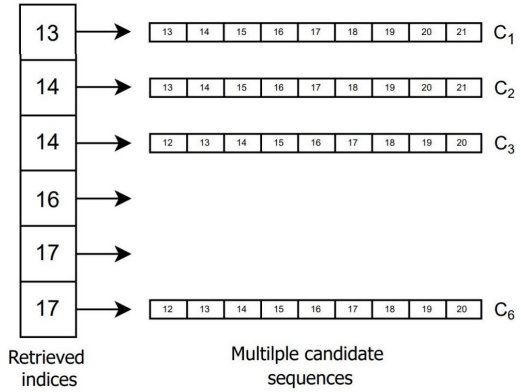

Figure 4: Generation of multiple candidate sequence for fine-grained search.

- Generate multiple sequence candidates C_i with their starting indice as $I_i = I_m - m$, where I_m is the retrieved index at m^{th} position.
- Select I_i (time offset) with maximum agreement among candidates.

SYSTEM OVERVIEW

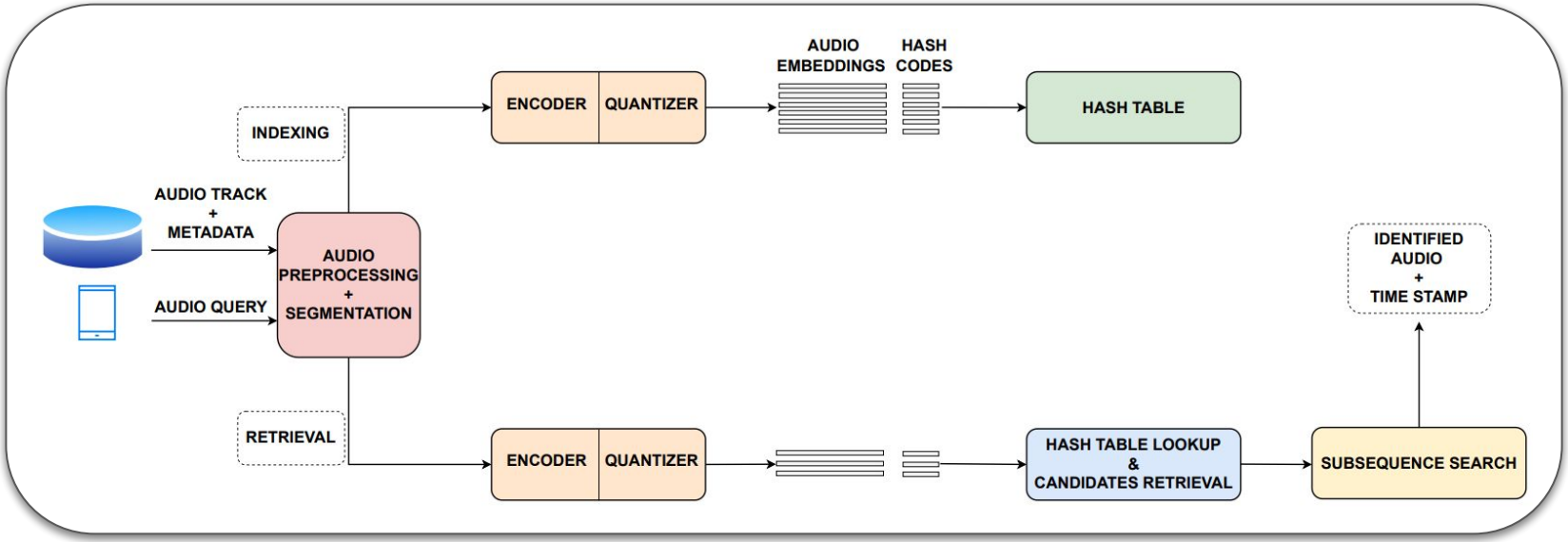

Figure 5: An overview of the system for indexing reference database and audio identification.

RESULTS

		Noise					Noise + Reverb					Reverb					
Length		0	5	10	15	20	0	5	10	15	20	0.2	0.4	0.5	0.7	0.8	speedup
1	NAFP + LSH	50.7	69.7	73.7	76.0	76.9	20.8	43.9	55.5	58.5	58.9	62.5	61.5	58.9	49.3	45.0	1x
	FE + LSH	**66.6**	**82.6**	87.6	90.0	90.6	**44.8**	**62.6**	73.8	**79.4**	**82.0**	83.8	83.8	78.4	**69.4**	64.4	1x
	FE + HT (Ours)	64.9	81.7	**87.9**	**90.1**	**90.9**	41.9	62.0	**74.6**	79.2	81.2	**84.6**	**82.3**	**80.0**	68.7	**64.5**	**2.4x**
2	NAFP + LSH	71.0	83.4	85.5	87.6	87.5	37.8	65.6	73.5	75.2	76.7	78.6	76.4	75.7	68.0	59.5	1x
	FE + LSH	**80.4**	88.2	91.6	93.2	94.8	**63.4**	78.2	84.6	86.0	85.4	90.4	86.4	85.0	74.8	67.8	1x
	FE+HT(Ours)	80.2	**89.4**	**93.1**	**94.1**	**94.9**	61.4	**80.8**	**85.0**	**87.5**	**88.3**	**92.4**	**87.7**	**86.4**	**77.3**	**70.7**	**2.4x**
3	NAFP + LSH	77.7	84.8	88.9	89.2	89.1	50.1	72.6	80.0	79.5	80.1	83.6	79.9	77.9	70.2	63.8	1x
	FE + LSH	83.2	88.4	92.6	94.4	95.2	**71.6**	82.6	85.6	86.2	87.4	91.8	87.8	85.6	74.4	68.4	1x
	FE+HT(Ours)	**84.7**	**90.8**	**95.5**	**96.3**	**97.1**	70.7	**84.1**	**88.6**	**89.0**	**90.1**	**93.8**	**90.3**	**87.9**	**77.9**	**71.6**	**2.3x**
5	NAFP + LSH	82.6	89.2	90.2	90.5	91.2	60.2	79.1	83.4	82.8	83.1	85.6	83.4	79.3	74.1	65.7	1x
	FE + LSH	85.6	90.0	92.8	94.2	95.8	80.0	87.0	87.1	87.6	87.2	93.8	88.8	86.6	75.0	69.0	1x
	FE+HT(Ours)	**88.0**	**93.4**	**95.3**	**96.2**	**97.4**	**80.6**	**88.6**	**90.8**	**91.3**	**91.5**	**96.4**	**91.1**	**88.9**	**78.7**	**71.2**	**2.4x**

- Efficacy: Our system achieves ~20% better hit-rate performance than baseline in adverse distortion environments (noise + reverb) with shorter query lengths (1s or 2s).
- Efficiency: Our system is 2.4x faster than LSH in search process.

RESULTS

		Noise					Noise + Reverb					Reverb					
Length		0	5	10	15	20	0	5	10	15	20	0.2	0.4	0.5	0.7	0.8	speedup
1	NAFP + LSH	50.7	69.7	73.7	76.0	76.9	20.8	43.9	55.5	58.5	58.9	62.5	61.5	58.9	49.3	45.0	1x
	FE + LSH	**66.6**	**82.6**	87.6	90.0	90.6	**44.8**	**62.6**	73.8	**79.4**	**82.0**	83.8	83.8	78.4	**69.4**	64.4	1x
	FE + HT (Ours)	64.9	81.7	**87.9**	**90.1**	**90.9**	41.9	62.0	**74.6**	79.2	81.2	**84.6**	**82.3**	**80.0**	68.7	**64.5**	**2.4x**
2	NAFP + LSH	71.0	83.4	85.5	87.6	87.5	37.8	65.6	73.5	75.2	76.7	78.6	76.4	75.7	68.0	59.5	1x
	FE + LSH	**80.4**	88.2	91.6	93.2	94.8	**63.4**	78.2	84.6	86.0	85.4	90.4	86.4	85.0	74.8	67.8	1x
	FE+HT(Ours)	80.2	**89.4**	**93.1**	**94.1**	**94.9**	61.4	**80.8**	**85.0**	**87.5**	**88.3**	**92.4**	**87.7**	**86.4**	**77.3**	**70.7**	**2.4x**
3	NAFP + LSH	77.7	84.8	88.9	89.2	89.1	50.1	72.6	80.0	79.5	80.1	83.6	79.9	77.9	70.2	63.8	1x
	FE + LSH	83.2	88.4	92.6	94.4	95.2	**71.6**	82.6	85.6	86.2	87.4	91.8	87.8	85.6	74.4	68.4	1x
	FE+HT(Ours)	**84.7**	**90.8**	**95.5**	**96.3**	**97.1**	70.7	**84.1**	**88.6**	**89.0**	**90.1**	**93.8**	**90.3**	**87.9**	**77.9**	**71.6**	**2.3x**
5	NAFP + LSH	82.6	89.2	90.2	90.5	91.2	60.2	79.1	83.4	82.8	83.1	85.6	83.4	79.3	74.1	65.7	1x
	FE + LSH	85.6	90.0	92.8	94.2	95.8	80.0	87.0	87.1	87.6	87.2	93.8	88.8	86.6	75.0	69.0	1x
	FE+HT(Ours)	**88.0**	**93.4**	**95.3**	**96.2**	**97.4**	**80.6**	**88.6**	**90.8**	**91.3**	**91.5**	**96.4**	**91.1**	**88.9**	**78.7**	**71.2**	**2.4x**

- Efficacy: Our system achieves ~20% better hit-rate performance than baseline in adverse distortion environments (noise + reverb) with shorter query lengths (1s or 2s).
- Efficiency: Our system is 2.4x faster than LSH in search process.

RESULTS

		Noise					Noise + Reverb					Reverb					
Length		0	5	10	15	20	0	5	10	15	20	0.2	0.4	0.5	0.7	0.8	speedup
1	NAFP + LSH	50.7	69.7	73.7	76.0	76.9	20.8	43.9	55.5	58.5	58.9	62.5	61.5	58.9	49.3	45.0	1x
	FE + LSH	**66.6**	**82.6**	87.6	90.0	90.6	**44.8**	**62.6**	73.8	**79.4**	**82.0**	83.8	83.8	78.4	**69.4**	64.4	1x
	FE + HT (Ours)	64.9	81.7	**87.9**	**90.1**	**90.9**	41.9	62.0	**74.6**	79.2	81.2	**84.6**	**82.3**	**80.0**	68.7	**64.5**	**2.4x**
2	NAFP + LSH	71.0	83.4	85.5	87.6	87.5	37.8	65.6	73.5	75.2	76.7	78.6	76.4	75.7	68.0	59.5	1x
	FE + LSH	**80.4**	88.2	91.6	93.2	94.8	**63.4**	78.2	84.6	86.0	85.4	90.4	86.4	85.0	74.8	67.8	1x
	FE+HT(Ours)	80.2	**89.4**	**93.1**	**94.1**	**94.9**	61.4	**80.8**	**85.0**	**87.5**	**88.3**	**92.4**	**87.7**	**86.4**	**77.3**	**70.7**	**2.4x**
3	NAFP + LSH	77.7	84.8	88.9	89.2	89.1	50.1	72.6	80.0	79.5	80.1	83.6	79.9	77.9	70.2	63.8	1x
	FE + LSH	83.2	88.4	92.6	94.4	95.2	**71.6**	82.6	85.6	86.2	87.4	91.8	87.8	85.6	74.4	68.4	1x
	FE+HT(Ours)	**84.7**	**90.8**	**95.5**	**96.3**	**97.1**	70.7	**84.1**	**88.6**	**89.0**	**90.1**	**93.8**	**90.3**	**87.9**	**77.9**	**71.6**	**2.3x**
5	NAFP + LSH	82.6	89.2	90.2	90.5	91.2	60.2	79.1	83.4	82.8	83.1	85.6	83.4	79.3	74.1	65.7	1x
	FE + LSH	85.6	90.0	92.8	94.2	95.8	80.0	87.0	87.1	87.6	87.2	93.8	88.8	86.6	75.0	69.0	1x
	FE+HT(Ours)	**88.0**	**93.4**	**95.3**	**96.2**	**97.4**	**80.6**	**88.6**	**90.8**	**91.3**	**91.5**	**96.4**	**91.1**	**88.9**	**78.7**	**71.2**	**2.4x**

- Efficacy: Our system achieves ~20% better hit-rate performance than baseline in adverse distortion environments (noise + reverb) with shorter query lengths (1s or 2s).
- Efficiency: Our system is 2.4x faster than LSH in search process.

SUMMARY

- An audio fingerprinting system robust against high noise and reverberation levels.
- Balanced hash codes using optimal transport framework.
- Improved the performance of the system in terms of efficiency and efficacy.
- Demonstrated better performance compared to baselines.

FOR MORE INFORMATION...

- Join us at our poster presentation
  - Date/Time: June 09, 2023, 08:15 AM (EEST)
  - Session name: Music Information Retrieval
- Or contact me at:
  - Email: anup.singh@ugent.be
  - Twitter: @15_anup