國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Graduate Institute of Communication Engineering
College of Electrical Engineering and Computer Science
National Taiwan University

Master's Thesis

基於聲紋辨識方法之改良式哼唱檢索系統

Improved Query by Humming System based on Audio Fingerprinting Method

龔鈺翔

Yu-Hsiang Kung

指導教授：丁建均 博士

Advisor: Jian-Jiun Ding, Ph.D.

中華民國 114 年 7 月

July, 2025

doi:10.6342/NTU202503400

誌謝

能夠順利完成這篇論文，我要由衷感謝我的指導教授丁建均老師。在研究的過程中，丁老師不僅在方向上給予我明確的指引，也在我迷惘時耐心指導與鼓勵，使我得以持續前進。同時，也感謝實驗室的學長姐們，您們的研究成果為我奠定了重要的基礎，讓我對本論文主題有了更深入且清晰的理解。此外，我要感謝實驗室的同學們，當我在研究上感到迷惘時，總是願意傾聽我的想法，並給予我寶貴的意見與支持。

最後，我要特別感謝我的家人，謝謝你們一路以來的支持與陪伴，讓我能夠無後顧之憂地專注於學業與研究不被其他因素影響。

doi:10.6342/NTU202503400

中文摘要

哼唱檢索系統(Query by Humming)是設計用在不知道傳統歌曲搜尋的資訊(如:歌名、歌手、歌詞等)的情況下，透過哼出一段旋律來搜尋出期望之歌曲。與常見的歌曲辨識不同，哼唱檢索是使用者哼出一段旋律，而非從背景聲音中找出撥放中的歌曲，這樣可能會導致哼唱的音高、速度都與使用者期望得到的歌曲有所出入。常見的哼唱檢索系統分為三個部分:音符切割(或稱為發端檢測)、音高辨識、資料比對，其中音符切割又分為兩種做法:音框導向及音符導向。

音框導向透過將輸入切割成固定長度的片段，辨識這個片段的音高後，透過所有片段的音符序列與資料庫的序列做比較。

另一種做法是音符導向，為了提升音高辨識的準確率，降低哼唱時的節奏差異以及音高抖動帶來的影響。音符導向透過偵測每個音的開始，藉此來切割出不同的音符片段，用來做音高的辨識。

相較於傳統將問題分成三個子問題來完成。也有些論文透過機器學習的方式來改善前兩個子問題的準確性，但大多受限於公開訓練資料的不足，導致效果不慎理想。

不過近年有論文提出將哼唱檢索系統視為是翻唱歌曲辨識的特殊情況，可以藉此透過翻唱歌曲辨識更多的公開資料來改善訓練資料的不足。本篇論文基於以上的假設，藉由機器學習的方式，將輸入的哼唱音訊轉換成一個高維的特徵，透過比對資料庫內的特徵相似度，來獲得最相近的歌曲排序，能獲得比傳統的方法更加準確的結果，同時也能規避哼唱檢索系統的公開資料不足的影響。

關鍵字：哼唱檢索、深度學習、聲紋辨識

doi:10.6342/NTU202503400

ABSTRACT

A Query by Humming (QBH) system is designed for situations where traditional song search information (such as title, artist, or lyrics) is unknown, allowing a user to find a desired song by humming a part of its melody. Unlike common song recognition, which identifies a song playing from a background source, QBH involves the user producing the melody themselves. This can result in discrepancies in pitch and tempo compared to the original song the user is trying to find.

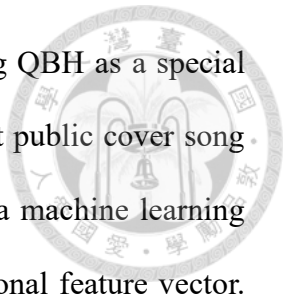
Conventional Query by Humming systems are typically composed of three main parts: note segmentation (or onset detection), pitch recognition, and data matching. Within note segmentation, there are two common approaches: frame-based and note-based.

The frame-based approach segments the input audio into fixed-length frames. After identifying the pitch of each frame, the resulting sequence of notes is compared against sequences in the database.

The other approach is note-based, which aims to improve pitch recognition accuracy and reduce the impact of rhythmic variations and pitch fluctuations inherent in humming. The note-based method works by detecting the start of each note, thereby segmenting the audio into distinct note fragments that are then used for pitch recognition.

In contrast to the traditional method of dividing the problem into these three sub-problems, some recent studies have leveraged machine learning to improve the performance of the first two components. However, these approaches are often limited by the scarcity of large-scale, publicly available QBH datasets, resulting in suboptimal performance.

doi:10.6342/NTU202503400



To address this limitation, recent research has proposed treating QBH as a special case of cover song identification, allowing the use of more abundant public cover song datasets for training. Based on this assumption, this work employs a machine learning approach that transforms input humming audio into a high-dimensional feature vector. The system then obtains a ranked list of the most similar songs by comparing feature similarity within the database. This method can achieve more accurate results than traditional approaches and also helps to circumvent the challenges posed by the limited availability of public data for Query by Humming systems.

Index Terms- Query by humming, deep learning, Audio fingerprinting

doi:10.6342/NTU202503400

CONTENTS

doi:10.6342/NTU202503400

doi:10.6342/NTU202503400

LIST OF FIGURES

doi:10.6342/NTU202503400

LIST OF TABLES

doi:10.6342/NTU202503400

Chapter 1 Introduction

In our daily life, there comes a moment that we want to find a song. Unfortunately, we cannot recognize either lyrics or title. The only thing that we have is a piece of melody. Query by Humming(QBH) system is aimed to solve this problem by allowing user to retrieve music through their humming melody.

In traditional QBH implement, we split QBH system into two main part melody extraction which also known as pitch estimate and melody matching. A typical process of a QBH system is when user send a query by humming the melody they remembered, QBH system first using melody extraction module to extract the melody. Then use this melody to compare with database to find most similar piece, the song name of this piece will return to user as the result.

Melody extraction, which means to generate a sequence of pitch or frequency to describe the audio signal. For an example, using staff notation to label a piano music is called melody extraction. There are two types of pitch labeling, single or chord. The example just mention is apparently chord type since piano music mostly comes with chord. It's not accurate using one single frequency to describe the music in any point. Fortunately, human can only sing a tone at same time. This leads to song melody can be noted as a sequence of main frequency. That is to say, we only need to extract one frequency in the same time which is easier than finding every tone.

On the other perspective of melody extraction, it can also split into two different types frame based and note based. Frame based using a fix length to sample a frequency. For example, the audio signal is sampled as 44100Hz, we can split this signal into some piece which is 512 samples long. Then we calculate the most significant frequency and

doi:10.6342/NTU202503400

use it to represent this frame. Another way to separate signal is note based, in this method we use note to pronounce the audio. To do this, we have to detect onset and offset perfectly. Otherwise the pitch could be calculated wrong or combine two notes into one note. There are two main advantages of this procedure. First, using this way to separate audio can benefit pitch estimation since it usually has longer samples compare to frame based. Also, the unstable frequency at the beginning of a note can be ignored which is not possible in frame based procedure. Second, we can ignore the tempo difference between query and reference which needs some matching tricks to reduce the impact in frame based system.

The other part of QBH system is melody matching, this can be seen as sequence matching problem. Given a short sequence, calculate the similarity with every long sequence generate by reference song or man labeled data. In this sub problem, there are some algorism that perform well in small dataset such as Earth Mover's Distance or dynamic time warp. There also have some algorism aims to improve searching speed for example local sensitive hash or multistage matching.

These work mentioned above have some challenges. In melody extraction, framed based method will suffer from tempo difference between query and reference song which need to use more complex matching algorism to collaborate. On the other side, although note based method can ignore tempo difference, this method still has its own problem such as missing note (like onset undetected make two notes seen as one note), poor performance on gliding pitch. In melody matching, since the query can be start at any time in reference song and the length of query notes are not fixed. The matching algorism will get slower in large database and long reference song. Furthermore, the matching algorism usually have to collaborate the imperfect extraction of melody. These problems make query by humming system impractical in real scenario due to there always have large amount of songs.

doi:10.6342/NTU202503400

According to these disadvantages, there is a need for more robust method that can speed up in matching stage, handle noise, tone difference. To address this problem, we propose a novel QBH system that using fingerprinting to describe segment of audio. This method has some advantages that can improve the performance of QBH system. First, using machine learning method as audio fingerprinting can make dataset not rely on human labeled note sequence. Second, conformer structure model can make classification more accuracy without adding more dimension on fingerprint vector. Lastly, as the query and reference are using fingerprint vector to matching not note sequence, we use cosine similarity as the matching algorism to improve the matching speed.

In summary, this work addresses the limitations of traditional QBH systems—such as limited tolerance to pitch and tempo variations, and dependence on note-based symbolic representation—by leveraging a deep neural architecture. We propose a Conformer-based embedding model that integrates harmonic attention and domain-adversarial training to improve robustness and generalization. Our contributions are threefold: (1) by using audio fingerprinting, we can use cosine similarity to boost retrieval speed, (2) we employ harmonic block to enhance pitch structure modeling, and (3) we use conformer model to achieves superior performance using a comparable embedding dimension compared to ByteHum[8].

1.1 Terms Definitions

Before introduce our proposed method, there are some related terms that people need to know. The definitions of some terms are shown below.

Table 1-1 Definitions of terms

Note	A symbol to represent pitch and duration of a sound.

doi:10.6342/NTU202503400

Frame	A small clip of a sound. Usually split sound into same frame size.
Pitch	Quantize frequency using 12-tone equal temperament scale
Octave	An octave contain 12 tones, an octave higher means double of frequency in Hz
Constant-Q Transform(CQT)	Transform audio into a time-frequency spectrum with pitch representation. Higher frequency resolution in lower frequencies.

doi:10.6342/NTU202503400

Chapter 2 Related work

2.1 Traditional approach

2.1.1 Frame-based approach

In frame-based systems, the audio signal is divided into fixed-size overlapping frames (e.g., 512 samples), and pitch is estimated for each frame independently. This method simplifies implementation and aligns well with spectral-based pitch trackers such as YIN or autocorrelation-based methods.

However, frame-based systems are sensitive to tempo differences between the query and reference. A faster or slower humming speed may shift the temporal alignment, requiring additional matching algorithms such as Dynamic Time Warping (DTW) to compensate for timing variations [4], [13]. Moreover, frame-level pitch estimation may suffer from inaccuracies in low-energy or gliding regions, where pitch salience is weak or unstable.

2.1.2 Note-based approach

The note-based approach is a widely adopted strategy in traditional Query-by-Humming (QBH) systems. Unlike frame-based methods that estimate pitch at fixed time intervals, note-based systems attempt to segment the audio into discrete musical notes by detecting onset and offset events. Each note is then assigned a pitch value, and the resulting symbolic sequence serves as the melodic representation of the audio. This representation is naturally invariant to tempo variations and is more aligned with human perception of melody.

Several works from National Taiwan University have focused on improving note-

doi:10.6342/NTU202503400

based QBH pipelines. Lin [1] proposed an onset-based segmentation method combined with dynamic time warping (DTW) to align pitch sequences between query and reference songs. Hu [4] similarly employed onset detection and developed a modified melody matching algorithm to improve alignment accuracy. Both systems rely heavily on symbolic note representations and traditional signal processing techniques.

To improve onset detection accuracy, Chen [2] introduced a CNN-based model that integrates multiple acoustic features—such as spectral flux and energy envelope—into a unified onset detector. Although deep learning is used for onset estimation, the overall retrieval process still follows a traditional note-based pipeline, including pitch estimation and symbolic matching. Building upon this, Hung [3] further replaced the CNN with a more robust model and introduced a waveform-level autoencoder to denoise vocal inputs before pitch extraction. [17] proposed using a vocal linguistic feature to improve the onset/offset detection. While these enhancements improve the reliability of melody extraction, they do not change the fundamental nature of the system, which remains rooted in symbolic, note-level representation and DTW-based matching.

Despite their conceptual clarity, note-based approaches face challenges in practical scenarios. Accurate onset detection remains difficult, especially in noisy, gliding, or expressive vocal input. Misdetected or missing onsets can lead to pitch merging or segmentation errors, ultimately affecting retrieval accuracy. Additionally, pitch estimation from short segments is susceptible to harmonic interference, which can misidentify overtones as the fundamental frequency.

Beyond conventional onset/offset detectors, some recent studies reformulate onset/offset detection as an object detection task on spectrograms. Musicyolo[18] demonstrates that treating a note as an object, then the boundaries of the object boxes can be seen as the onset and offset of the note. This method allows notes that are very close

doi:10.6342/NTU202503400

to other notes to be detected more accurately.

2.1.3 Pitch estimation

Pitch estimation is one of the most critical components in traditional Query-by-Humming (QBH) systems, as it directly determines the accuracy of the extracted melody. The goal is to estimate the fundamental frequency (f_0) of the input audio segment, which is the basis of melody.

Conventional pitch estimation algorithms typically work on short-time frames using frequency-domain techniques. Notable examples include time-domain based methods ACF, AMDF and frequency-domain based methods HPS, Cepstrum. Both methods perform well in ideal scenarios.

However, these methods can easily be affected by noise, harmonic frequency, and instability of the vocal. One of the most common failures is identifying harmonic frequency as fundamental frequency, also known as octave error. To eliminate the effect of this problem, several enhancements have been proposed, such as energy-based thresholding and pitch smoothing across frames [6],[9],[12].

Some note-based approaches attempt to improve pitch estimation by leveraging onset and offset information, given stable note regions for pitch estimation. While this can reduce transient noise and vibrato effects, it also introduces new dependencies on the accuracy of onset detection.

Due to these limitations, recent research has shifted toward learning-based pitch estimation. Methods such as CREPE and other neural network-based f_0 trackers[26], [30] have demonstrated superior robustness in noisy and unconstrained environments, motivating their integration into modern QBH pipelines.

In addition, [16] further proposed a lightweight pitch estimation encoder-decoder

doi:10.6342/NTU202503400

model with an integrated detector to determine whether a melody exists in each input frame. With this method, it can perform better in some scenarios.

2.1.4 Sequence Matching Algorithms

After melody extraction, the next step is to compare the query sequence with entries in the database. Several algorithms have been proposed for sequence matching:

1 Dynamic Time Warping (DTW) [4] aligns sequences with non-linear time distortion, allowing tolerance to tempo variation.

2 Earth Mover's Distance (EMD) [14] computes similarity by measuring the minimum effort to transform one sequence into another.

3 Longest Common Subsequence (LCS) captures symbolic similarity by counting matching note subsequences.

4 Multistage Matching [13] and Locality-Sensitive Hashing (LSH) are used to accelerate matching in large-scale databases.

While effective in small or mid-scale setups, these methods often struggle with scalability. The retrieval time grows with the number and length of candidate songs, making them less suitable for real-world music libraries containing millions of tracks.

Additionally, some studies like [12] use smoothing techniques to refine the sequence and eliminate certain octave errors to generate a more accurate sequence that can benefit the matching result.

2.2 Machine learning approach

Deep learning has significantly reshaped the landscape of Query-by-Humming

doi:10.6342/NTU202503400

(QBH) systems. Historically, QBH pipelines have drawn heavily from singing melody transcription methods, which aim to predict a pitch sequence from audio. These transcription approaches can be categorized into frame-based and note-based paradigms, often coupled with pitch estimation modules. More recently, a novel perspective emerged—proposing that QBH can be seen as a special case of the cover song identification (CSI) problem, where a user-generated humming query serves as the cover variant. This view laid the foundation for transferring CSI models to the QBH task, as exemplified by ByteHum [8], which leverages this framing to use large-scale CSI datasets for supervised training [32].

Several fully deep learning-based systems have been proposed. [5] treats QBH as a classification problem, using a chromagram to reduce unnecessary information and a CNN-based model for classification. ByteHum [8] adopts a CSI framework by applying neural fingerprinting to CQT spectrograms extracted from vocal-separated tracks. Using triplet loss for metric learning, it produces robust segment-level embeddings that are matched using cosine similarity. CoverHunter [20] shows that the combination of attention mechanisms and convolutional networks can achieves better performance on the CSI problem compared to convolutional networks alone. DisCover [21] further separates melody-relevant content from confounding factors using disentangled representation learning, increasing retrieval performance under varied acoustic conditions.

Other works focused on deep learning-based melody extraction. For example, [7] uses both temporal and frequency domain attention mechanisms to improve melody extraction accuracy. HANet [26] further proposes a harmonic attention mechanism to improve pitch estimation, especially in polyphonic settings. HKDSME [15] uses harmonic supervision to improve melody extraction under semi-supervised conditions. These systems, while not end-to-end QBH pipelines, provide important building blocks

doi:10.6342/NTU202503400

that enhance the quality of humming representations.

A key distinction should be made between systems that only incorporate neural components (e.g., using CNNs for onset detection [2], or denoising autoencoders [3]) and those where the entire pipeline, from spectrogram input to retrieval result, is optimized via deep learning. The latter enables robustness to key variations, tempo changes, and background noise, making them better suited for real-world deployment.

2.3 Audio Fingerprinting and Metric Learning

Among the deep learning-based approaches to QBH, one of the most transformative techniques is neural audio fingerprinting. Unlike symbolic methods that rely on explicit pitch sequences, fingerprinting encodes audio segments into dense, discriminative embeddings in a fixed-dimensional vector space. These fingerprints allow fast and accurate retrieval via vector similarity, even in noisy or distorted queries.

Neural fingerprinting can be viewed as an extension of traditional audio fingerprinting systems, but instead of using hand-crafted features, it uses convolutional or transformer-based encoders trained with metric learning objectives such as triplet loss [22] or contrastive loss. The model learns to pull embeddings of similar (hummed vs. original) segments closer, while pushing dissimilar ones apart. This enables efficient Maximum Inner Product Search (MIPS) over large-scale reference databases.

Systems like “Now playing”[19], ByteHum [8], and CoverHunter [20] implement this concept effectively. In ByteHum, the vocal-separated audio is converted into CQT spectrograms and processed by a CNN-based encoder followed by L2 normalization. The resulting 128-dimensional vectors serve as neural fingerprints. CoverHunter enhances this architecture with attention mechanisms and alignment refinements, improving precision in partial or noisy queries.

doi:10.6342/NTU202503400

There are three main advantages of audio fingerprinting:

1. Speed: Fingerprints can be indexed and retrieved using fast approximate nearest neighbor (ANN) methods.
2. Robustness: Embeddings tolerate pitch drift, tempo variation, and background noise better than symbolic features.
3. Scalability: Retrieval remains efficient even when searching across tens of thousands of songs.

By framing QBH as a segment-level retrieval task with neural fingerprints, these systems sidestep many limitations of traditional note-based approaches, such as onset misalignment and gliding pitch distortion. Neural fingerprinting thus represents a paradigm shift in QBH system design.

doi:10.6342/NTU202503400

Chapter 3 Method

In this section, we present the architecture of our proposed method, including overall system, data preprocessing, model architecture, and loss function. The overall workflow is shown in Figure 3.1.

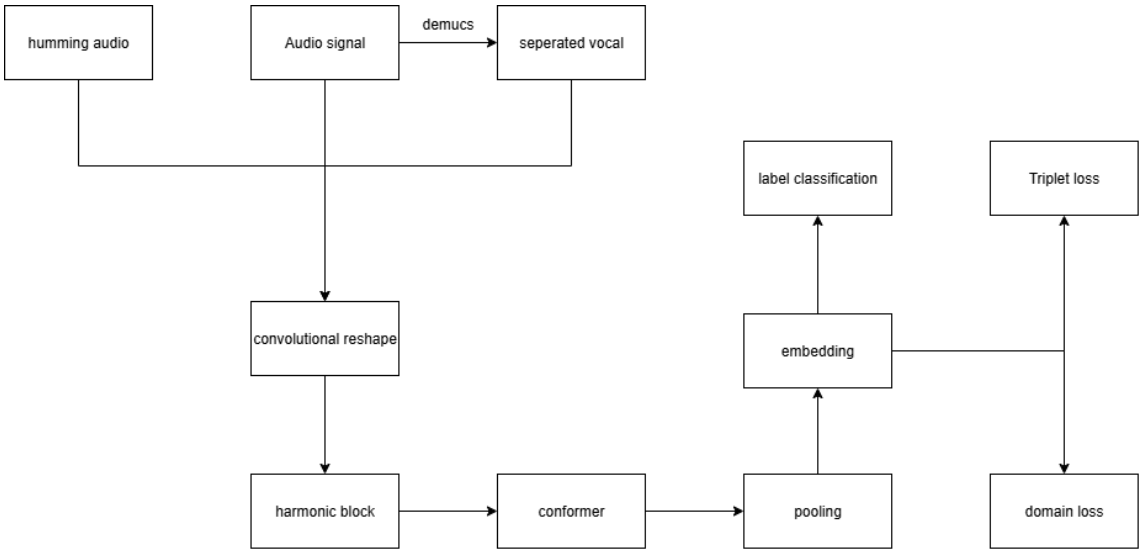


Figure 3.1 System architecture overview

3.1 Overall system

The architecture of our proposed system is inspired by ByteHum[8], as shown in Figure 3.1, the audio will be split into 8-second segments with a 3-seconds hop. Then we transform each segment into time-frequency image by Constant-Q Transform(CQT). After this step, we feed the images into model, which will output a 128-dimensional fingerprint. We use this fingerprint to compare the cosine similarity with a database that is generated through the same process using reference songs.

doi:10.6342/NTU202503400

3.2 Data preprocess

In traditional QBH systems, we only need some samples to validate the performance of a system, it leads to the lack of humming audio data on the internet. To deal with this problem, in [32], they proposed to consider QBH as a special case of Cover Song Identification(CSI). Following this thought, we separate the vocal soundtrack from dataset by demucs[25] to simulate humming audio.

After separate vocal tracks, we split vocal tracks into 8-seconds segment with a 3-seconds hop, and use RMS energy to check whether segment has voice. For those segments containing voice, we take same time shift window to original songs. These vocal clips, original song clips and some MIR-QBSH segments will be conbined as our training set.

Segments in training set will apply Constant-Q Transform with 96 bins which means 8 octave from C0 to B7 and 256 samples hop.

The vocal tracks were separated using Demucs[25] pre-trained model, applied on both synthetic datasets ST500[28] and Kaggle Humming Audio[29]. For quality control, we used RMS thresholding to exclude silent or low-energy segments; segments with mean RMS below 0.02 were discarded. To simulate real-world humming input, no additional pitch correction or post-processing was applied to the separated vocal.

3.3 Model Architecture

Our proposed model architecture shown in Figure 3.1, can be separated into four parts, each part is describe below:

doi:10.6342/NTU202503400

3.3.1 Down sampling and projecting to higher dimension space

In common, when we need to down sampling in time domain, we can either use 1D convolution or pooling to achieve. These functions can maintain the same dimension along frequency axis. However, in our scenario we also need to projection to upper space in frequency axis. It means maintain same dimension is unimportant, so we use 2D convolution to make down sampling consist both time and frequency information. Then use linear projection to reshape frequency axis to embedding dimension we want.

Specifically, we use 2 layers of 2D convolution with 2 sample stride to reduce both dimension to 1/4. In order to not lose frequency information, we set channel to same as embedding dimension. Then we reshape it into (T/4,Demb*D/4) and use a linear project to project to embedding dimension(Demb).

3.3.2 Harmonic block

From previous works[2], they showed that one of the difficulty of QBH system is to estimate pitch perfectly. Furthermore, the main factor affecting pitch estimation accuracy is the incorrect identification of harmonics as the fundamental frequency.

doi:10.6342/NTU202503400

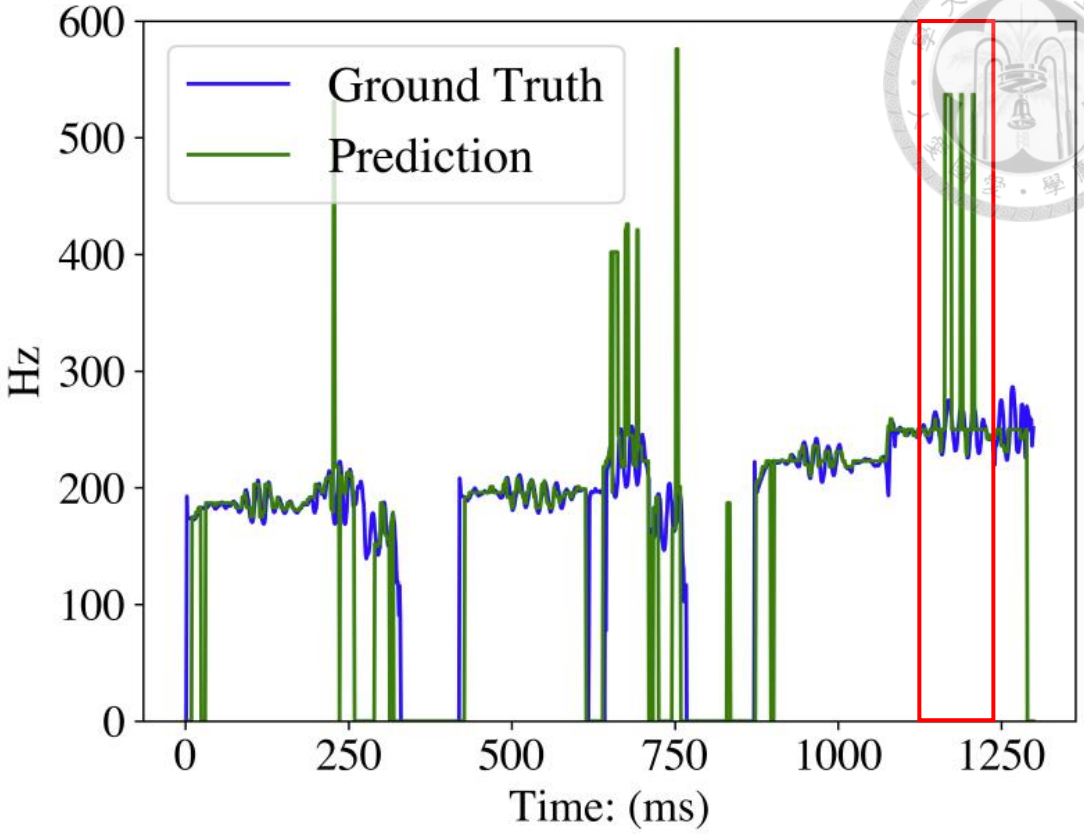


Figure 3.2 [30] shows an example of harmonic frequency predict as fundamental frequency

This shows that the relationships between the fundamental frequency and harmonics are important in QBH system. To better capture these relations, HANET[26] proposed a method using temporal and frequency two branch to capture both side features shown in Figure 3.3.

doi:10.6342/NTU202503400

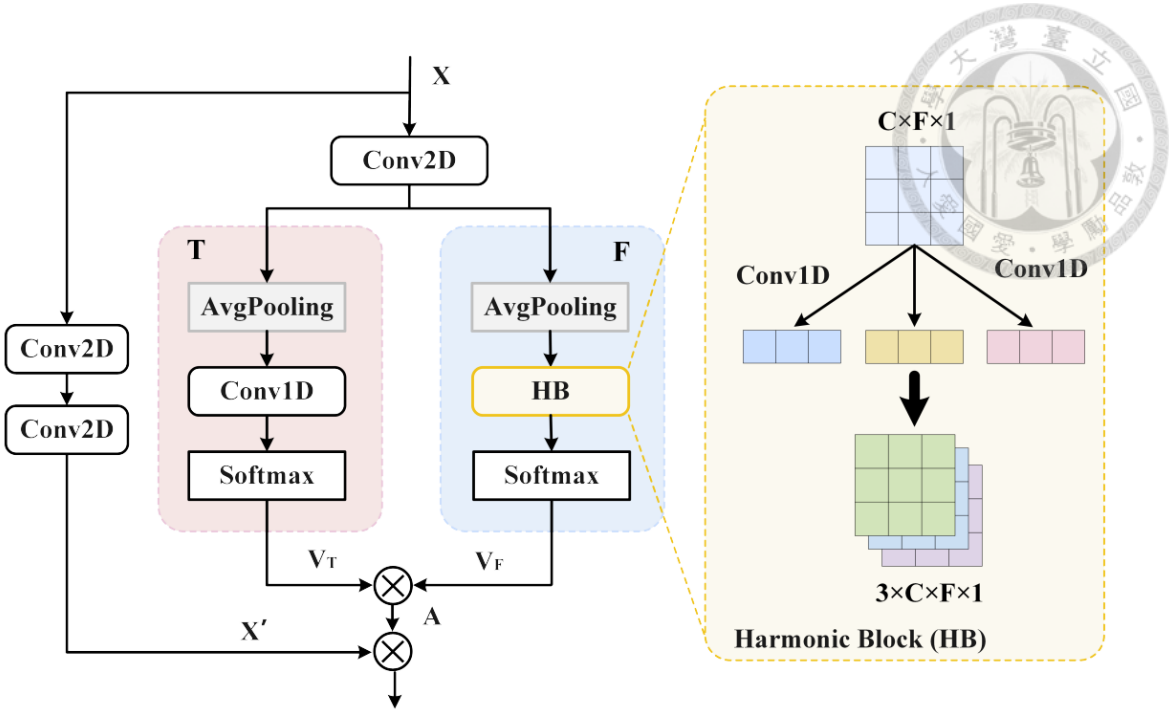


Figure 3.3 HANet[26] proposed Harmonic Block to capture harmonic relations

In this block, they use two 2D convolution layers to upscale channel to 3X. Then in the temporal branch, they use an average pooling to smooth the image, then use an 1d convolution along time axis to get temporal features. On the other side, frequency branch use another average pooling to smooth the image, then use 3 different kernel sizes 1d convolution to capture different harmonic features. After this, they concatenate 3 images from different convolutions and multiply with temporal features.

To better capture the difference harmonic relations, they adopt 3 different Harmonic Block and use a modified channel attention block shown in Figure 3.4 to fusion different Harmonic Block's output.

doi:10.6342/NTU202503400

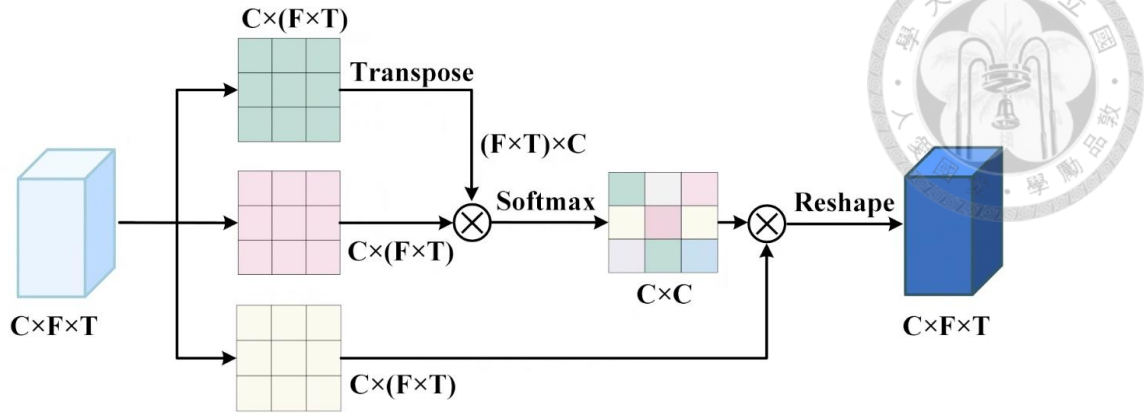


Figure 3.4 HANet[26] proposed modified channel attention block

We adopt this method after down sampling. According to different input shape, specific bins per octave, we modified the kernel size to match the original concept[26] shown in Table 3-1.

Table 3-1 Kernel size comparison with original setting

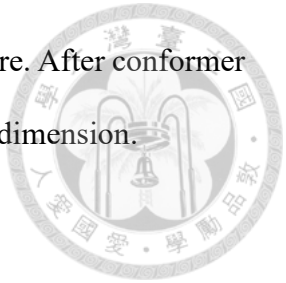
	Kernel size	Kernel size(original)
Full	[12, 42, 72]	[60, 210, 360]
Mid	[12, 30, 48]	[60, 150, 240]
Tiny	[12, 18, 24]	[60, 90, 120]

3.3.3 Conformer

When the transformer model published, its strong performance on language task such as translation has generated significant interest. As previous work[23], they shows that the combination of transformer and convolution neural network(CNN) which known as conformer has strong ability on speech task. On the other hand, query by humming can be seen as a type of speech recognition task. Based on this assumption, we use conformer as the main component of our proposed method. In detail, we use 128 dimension as our model token size, with 256 feed forward neural and 4 attention head as one conformer

doi:10.6342/NTU202503400

layer. In our model, we use 4 layers of conformer to generate the feature. After conformer block, we use a pooling layer to make final fingerprint output as 128 dimension.



3.4 Loss function

To train the model effectively, we employ a composite loss function that combines three components: focal loss, triplet loss, and domain adversarial loss. The total loss is defined as:

$$L_{total} = L_{focal} + 0.3 * L_{triplet} + 0.1 * L_{domain} \tag{3.1}$$

This weighted combination encourages the model to simultaneously perform accurate embedding learning, sample discrimination, and domain-invariant representation extraction.

3.4.1 Focal loss

In [24], they proposed a new loss function called focal loss to handle imbalanced data between classes during training, particularly in the classification-based auxiliary tasks. It modulates the standard cross-entropy loss by focusing more on hard-to-classify examples. The focal loss is defined as:

$$L_{focal} = -\alpha_t(1 - p_t)^{\gamma} \log(p_t) \tag{3.2}$$

where p_t is the model's estimated probability for the true class, α_t is a balancing factor, and γ is the focusing parameter. In our experiments, we set α_t = 1 and γ = 2, for our training process.

doi:10.6342/NTU202503400

3.4.2 Triplet loss

To encourage embeddings of similar audio segments to be close in the embedding space, we apply triplet loss. Each training sample consists of an anchor **a**, a positive sample **p** from the same song or vocal variation, and a negative sample **n** from a different song. The triplet loss is computed as:

$$L_{triplet} = \max(0, \mathrm{d}(\mathrm{a}, \mathrm{p}) - \mathrm{d}(\mathrm{a}, \mathrm{n}) + \mathrm{m}) \tag{3.3}$$

where $\mathrm{d}(\cdot,\cdot)$ denotes cosine distance, and m is a predefined margin which is set to 0.3

3.4.3 Domain loss

To reduce the domain gap between humming queries and original vocal segments, we introduce a domain adversarial loss based on the Gradient Reversal Layer (GRL) framework [31].

In this setup, a domain classifier is appended to the shared feature encoder. During training, the GRL inverts the gradient of the domain classification loss before it is propagated to the encoder. This encourages the encoder to produce domain-invariant embeddings while allowing the classifier to learn to discriminate between domains.

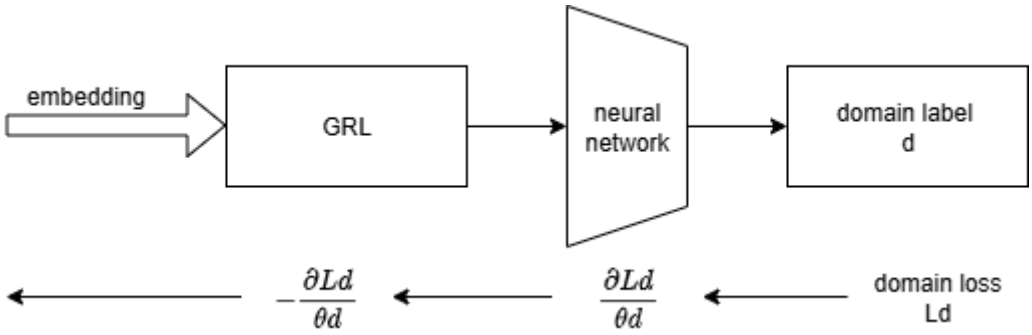


Figure 3.5 domain loss architecture

Let D be the binary domain label (0 for singing, 1 for humming). The binary cross-

doi:10.6342/NTU202503400

entropy loss is defined as:

$$L_{domain} = cross\ entropy(D) \tag{3.4}$$

During backpropagation, the gradient passed to the encoder through the GRL is multiplied by $-\lambda$, effectively reversing the optimization objective. While the domain classifier aims to minimize classification error, the encoder learns to confuse the classifier, thus producing more domain-agnostic features. In our experiments, we set the GRL coefficient $\lambda = 1.0$.

doi:10.6342/NTU202503400

Chapter 4 Experiment

4.1 Experimental Setup

In this work, we use several datasets in both training and testing. The training dataset contains three different datasets, the detailed information is listed below in Table 4-1.

Table 4-1 Dataset infomation

#	1	2	3
source	MIR-QBSH[27]	ST500[28]	Kaggle humming audio[29]
usage	Training & testing	Training	Training
length	4431(3987/444)	384songs	200 songs/206 humming record
Type	humming	song	Song&humming

Dataset1 is split into 90% of training and reference data, and other 10% for testing query. The other 2 datasets are used for training, combined with separated vocal tracks and original song to gather the training set.

4.2 Evaluation Metrics

In our work, we propose an end-to-end QBH system, to assess the performance of the proposed Query-by-Humming (QBH) system, we adopt several commonly used evaluation metrics in music retrieval tasks such as MRR and Top-k ratio. These measurement parameters are come from the MIREX query by singing/humming standard.

doi:10.6342/NTU202503400

4.2.1 Top-K ratio

The Top-K ratio is used to calculate the ratio of right answer in first K candidates for all given query. We use Top-1, and Top-10 as our evaluation metrics. The Top-K ratio formula is:

$$Top-K = \frac{number\ of\ right\ answer\ in\ first\ K\ candidates}{number\ of\ all\ query} \tag{4.1}$$

4.2.2 Mean Reciprocal Rank (MRR)

Mean Reciprocal Rank(MRR) is used to evaluate the overall ranking quality of the retrieval results. For each query, the reciprocal rank is calculated as the inverse of the position at which the correct song is first retrieved. Then calculate the mean across all query. MRR is defined as:

$$MRR = \frac{1}{query\ number} \sum_{x}^{Queries} \frac{1}{Rank(x)} \tag{4.2}$$

4.3 Main Results

In this section, we compare our proposed QBH system with several models and previous work on MIR-QBSH dataset. Table 4-2 shows the performance of Mean Reciprocal Rank(MRR), top-1,and top-10 hit rates of our system and others.

Our method achieves MRR of 0.971 and Top-1 hit rate of 0.95 which is the highest of the table, outperforming ByteHum[8] and other traditional or learning-based methods. Although our method achieves the same Top-10 accuracy as ByteHum[8], our proposed method shows better performance in Top-1, which indicates our system performs better in early rankings. This also shows the reason of better performance on MRR.

doi:10.6342/NTU202503400

Compared to traditional note-based method proposed by Lin[1], which achieves 0.806 in MRR, our system improves by over 20% or 16.5% in absolute. Similarly, our method outperforms Chen[2]'s method, and other note-based systems. These results demonstrate the effectiveness of the neural fingerprinting approach.

Table 4-2 Evaluation matric comparison with different method

Method		MRR	hitrate	
			Top1	Top10
proposed		0.971	0.95	0.99
ByteHum [8]		0.94	0.9	0.99
Lin [1]		0.806	0.7415	0.9438
Chen [2]		0.9547	-	0.989
CHAD [10]		-	-	0.921
Mostafa & Fung [14]		0.919	-	-
Ulfi & Mandala [7]		0.33	0.17	0.73
Ranjan & Arora [6]		0.771	-	-
Triastanto & Mandala [9]		~0.3*	~0.2*	~0.7*
Alfaro-Paredes et al. [11]	DTW	0.26	0.1583	0.5
	Qmax	0.24	0.1667	0.375

* Denotes values that were estimated from the corresponding figure in the source.

4.4 Ablation Studies

doi:10.6342/NTU202503400

To validate the effectiveness of individual components in our proposed QBH system, we conducted a series of ablation studies. Specifically, we examined the impact of the Conformer encoder structure, harmonic block, and domain adversarial training. Table 4-3 summarizes the performance of different architectural variants in terms of Mean Reciprocal Rank (MRR) and Top-1 rate.

Table 4-3 Ablation Study result

Method	MRR	difference	Top1	difference
conformer	0.90832	-	0.86486	-
transformer	0.53829	-40.74%	0.40766	-52.86%
proposed	0.93454	-	0.8964	-
conformer without harmonic block	0.90832	-2.81%	0.86486	-3.52%
conformer without convolution prelayer	0.87589	-6.28%	0.81532	-9.05%
conformer without domain loss	0.89574	-4.15%	0.84685	-5.53%

4.4.1 Effect of Conformer vs. Transformer Encoder

To evaluate the impact of temporal modeling architecture, we compare our proposed Conformer-based encoder with a Transformer-based counterpart using the same input preprocessing and convolutional reshaping method. Both variants are trained under the same loss functions and dataset conditions to ensure a fair comparison.

doi:10.6342/NTU202503400

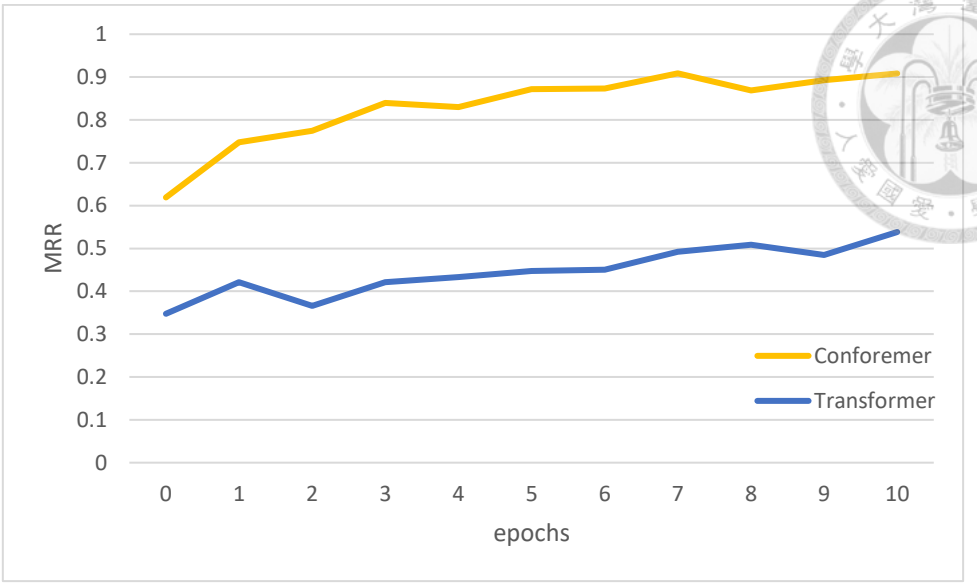


Figure 4.1 MRR comparison between Conformer and Transformer

As shown in Table 4-3 and Figure 4.1, the Conformer-based model achieves an MRR of 0.908, significantly outperforming the Transformer-based variant, which only achieves 0.538. The Top-1 rate also drops from 0.896 to 0.408. This demonstrates that the combination of self-attention and local convolution in the Conformer structure is better suited for modeling the time-frequency patterns of humming audio.

The Transformer encoder, while capable of capturing long-range dependencies via self-attention, lacks the ability to model local acoustic continuity and harmonic structure effectively. In contrast, the Conformer integrates convolutional modules that capture local sequential features, which are essential for robust melody representation in humming queries—especially under gliding pitch or expressive singing.

These results validate the importance of using convolution-augmented attention mechanisms in QBH systems, where both global and local temporal context are crucial for matching melodic patterns.

doi:10.6342/NTU202503400

4.4.2 Effect of Harmonic Block

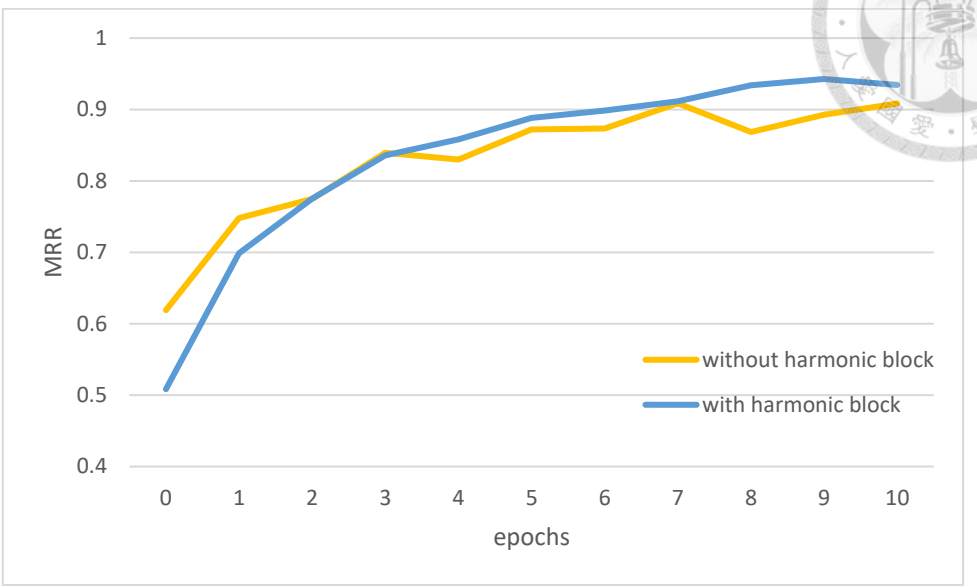


Figure 4.2 Comparison the effect of harmonic block

Since the previous section demonstrated the effectiveness of Conformer model, we evaluate the effect of the harmonic block by comparing the full Conformer model with its harmonic block removed. When harmonic modeling is disabled, the performance drops noticeably across all metrics (MRR drops from 0.935 to 0.908; Top-1 from 0.896 to 0.865).

The harmonic block is specifically designed to capture pitch-relevant frequency relationships by using multiple 1D convolutions with different kernel sizes across the frequency axis. This allows the model to explicitly learn harmonic intervals, such as octaves and fifths, which often occur in music and humming inputs. By aggregating these multi-resolution harmonic cues, the model becomes more sensitive to subtle pitch structures, even when the fundamental frequency is weak or slightly off-key.

Without this module, the model relies solely on the downstream encoder to infer frequency structure, which may lead to confusion between closely spaced overtones and reduce the discriminative power of embeddings. The ablation results confirm that removing the harmonic block leads to a significant degradation in both MRR and Top-1

doi:10.6342/NTU202503400

accuracy, reinforcing its importance in enhancing melody contour representation and improving retrieval robustness under pitch fluctuation or vibrato.

4.4.3 Effect of convolution reshape

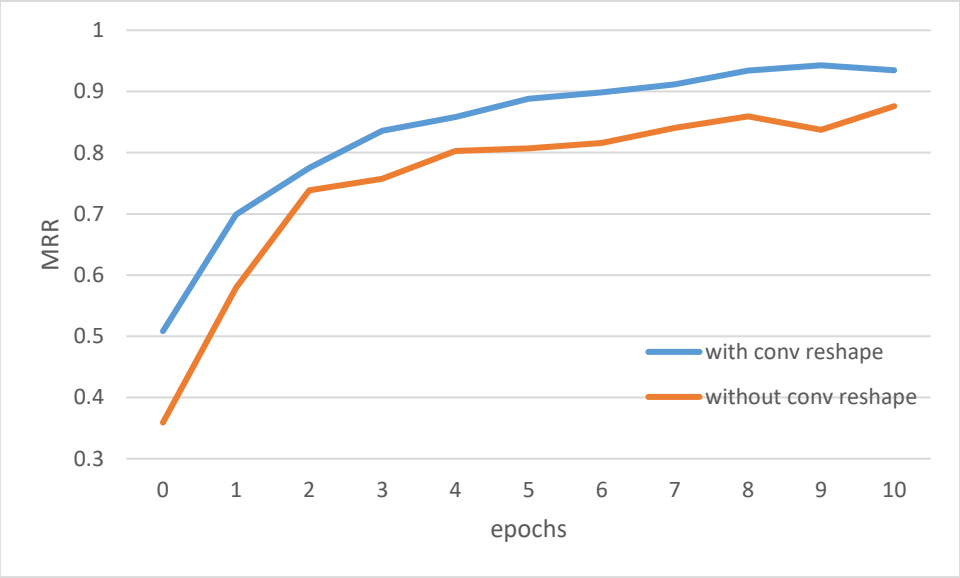


Figure 4.3 Comparison the effect of convolution reshape

In [20], a 2D convolutional layer is used to reduce the temporal length of the CQT input and to project the frequency axis into the embedding dimension of the backbone model. Inspired by this, we adopt a similar convolutional reshape mechanism in our model to not only accelerate inference but also replace the need for a separate linear projection.

Figure 4.3 compares the performance of the model with convolutional reshape and the variant using linear projection instead. As shown in the figure, the model with convolutional reshape achieves higher MRR across all epochs. According to Table 4-3, the final model without reshape suffers a 6.28% drop in MRR and a 9.05% drop in Top-1 accuracy, highlighting the importance of this design choice.

We hypothesize that the convolutional reshape facilitates better alignment between

doi:10.6342/NTU202503400

frequency-domain features and the encoder's input space, enabling more stable and effective learning of melodic patterns. Without this step, the model may require more capacity to learn this transformation implicitly, leading to slower convergence and degraded performance.

4.4.4 Effect of Domain Adversarial Loss

According to the lack of aligned humming audio resource, we use separated singing vocal tracks as a kind of humming audio which is inspired by [32]. To minimize the difference between separated vocal and humming audio, we adopt an adversarial loss

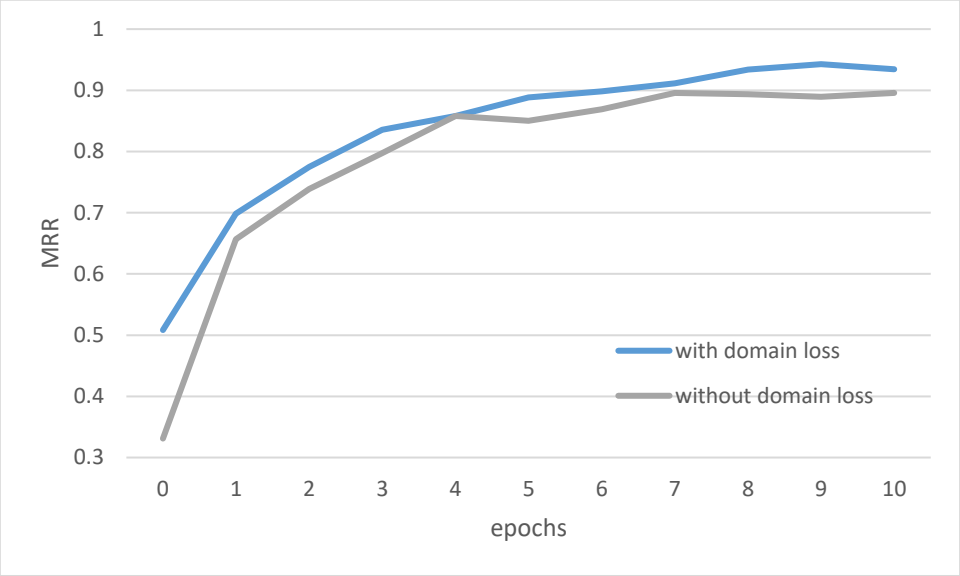


Figure 4.4 Comparison the effect of domain loss

We also examine the impact of domain loss by removing the GRL-based domain classifier. As shown in "conformer wo domain loss," the MRR drops from 0.935 to 0.896, and Top-10 accuracy decreases from 0.896 to 0.847.

The domain adversarial loss plays a crucial role in promoting domain-invariant representations between training and query inputs, which may differ in style, tone quality,

doi:10.6342/NTU202503400

or recording conditions. Specifically, we adopt a Gradient Reversal Layer (GRL) that connects the embedding output to a domain classifier. During backpropagation, the GRL inverts gradients flowing into the encoder, encouraging it to produce embeddings that confuse the domain classifier and thus become agnostic to domain-specific characteristics.

This mechanism helps bridge the distributional gap between clean singing vocals (e.g., from ST500[28]) and user-generated humming inputs (e.g., from MIR-QBSH[27]). Without this alignment, the model may overfit to synthetic vocal training data and fail to generalize to real-world queries. The performance drop observed in the ablation confirms that domain adversarial training enhances robustness and consistency across diverse input sources.

doi:10.6342/NTU202503400

Chapter 5 Conclusion

In this thesis, we proposed a robust and scalable deep learning-based Query-by-Humming (QBH) system that combines harmonic-aware audio fingerprinting with advanced temporal modeling. Our method integrates a harmonic block to enhance pitch-related features and adopts a Conformer-based encoder to effectively capture both local and global temporal dependencies in humming queries. To further improve generalization across different audio domains, we introduced a domain-adversarial training objective using a gradient reversal layer (GRL).

We trained our system on datasets including ST500 and Kaggle Humming Audio and evaluated the system on the MIR-QBSH benchmark. Experimental results demonstrated that our method outperforms previous approaches, including traditional pitch-based DTW systems and recent deep learning baselines such as ByteHum. The system achieved state-of-the-art performance in terms of MRR and Top-k accuracy, thereby confirming the effectiveness of the proposed architecture.

In addition, we conducted ablation studies to validate the contributions of each component, including the harmonic block, Conformer encoder, and domain loss. The results highlight the importance of jointly modeling pitch harmonics and time-frequency structures for accurate and efficient humming-based music retrieval.

While the proposed system performs well on clean and moderately noisy queries, limitations remain in handling heavily distorted, fragmented, or off-pitch queries in real-world environments. Future work may explore stronger augmentation techniques, semi-supervised pretraining, and broader query modalities such as whistling or beatboxing to further enhance the robustness and applicability of QBH systems.

doi:10.6342/NTU202503400

In conclusion, this work contributes to the ongoing development of content-based music retrieval systems by demonstrating the potential of neural fingerprinting and harmonic modeling in QBH tasks. We hope this research serves as a foundation for more intelligent, efficient, and user-friendly music search technologies.

doi:10.6342/NTU202503400

Chapter 6 Reference

[1] 林巧薇 (2016). "應用節奏與頻率資訊之改良式哼唱檢索系統及改良式發端偵測與旋律匹配"

[2] 陳秉鴻 (2020). "深度學習，池化運算及改良式動態規劃應用於哼唱檢索系統."

[3] 洪譽承 (2022). "基於深度學習原音自編碼器去噪應用於哼唱式系統"

[4] 胡哲銘 (2010). "歌聲檢索系統：改良式發端識別以及修正式旋律比對"

[5] K. -Y. Chen and J. -J. Ding, "Chromagram Features Analysis for Learning-Based Query by Humming Systems," 2025 International Conference on Electronics, Information, and Communication (ICEIC), Osaka, Japan, 2025, pp. 1-4, doi: 10.1109/ICEIC64972.2025.10879656.

[6] S. Ranjan and V. Arora, "A Bioinformatic Method Of Semi-Global Alignment For Query-By-Humming," 2020 IEEE 4th Conference on Information & Communication Technology (CICT), Chennai, India, 2020, pp. 1-5, doi: 10.1109/CICT51604.2020.9312085.

[7] M. Ulfi and R. Mandala, "Improving Query by Humming System using Frequency-Temporal Attention Network and Partial Query Matching," 2022 9th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA), Tokoname, Japan, 2022, pp. 1-6, doi: 10.1109/ICAICTA56449.2022.9933001.

[8] X. Du, P. Zou, M. Liu, X. Liang, M. Chu and B. Zhu, "ByteHum: Fast and Accurate Query-by-Humming in the Wild," ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Seoul, Korea, Republic of, 2024, pp. 1111-1115, doi: 10.1109/ICASSP48485.2024.10448117.

doi:10.6342/NTU202503400

[9] A. N. Dwi Triastanto and R. Mandala, "Query by Humming Music Information Retrieval using DNN-LSTM based Melody Extraction and Noise Filtration," *2022 5th International Conference on Information and Communications Technology (ICOIACT)*, Yogyakarta, Indonesia, 2022, pp. 503-508, doi: 10.1109/ICOIACT55506.2022.9972121.

[10] A. Amatov, D. Lamanov, M. Titov, I. Vovk, I. Makarov, and M. Kudinov, "A Semi-Supervised Deep Learning approach to dataset collection for Query-By-Humming task," *arXiv.org*, Dec. 02, 2023. https://arxiv.org/abs/2312.01092

[11] E. Alfaro-Paredes, L. Alfaro-Carrasco, and W. Ugarte, "Query by humming for song identification using voice isolation," in *Lecture notes in computer science*, 2021, pp. 323–334. doi: 10.1007/978-3-030-79463-7_27.

[12] S. Ranjan and V. Srivastava, "Incorporating Total Variation Regularization in the design of an intelligent Query by Humming system," *arXiv.org*, Feb. 09, 2023. https://arxiv.org/abs/2302.04577

[13] M. Li, Z. Zhao and P. Shi, "Query by humming based on the hierarchical matching algorithm," *2015 IEEE International Conference on Computer and Communications (ICCC)*, Chengdu, China, 2015, pp. 82-86, doi: 10.1109/CompComm.2015.7387545.

[14] N. Mostafa and P. Fung, "A Note Based Query By Humming System Using Convolutional Neural Network," *Interspeech 2017*, pp. 3102–3106, Aug. 2017, doi: https://doi.org/10.21437/interspeech.2017-1590.

[15] S. Yu, X. He, K. Chen, and Y. Yu, "HKDSME: Heterogeneous Knowledge Distillation for Semi-supervised Singing Melody Extraction Using Harmonic Supervision," pp. 545–553, Oct. 2024, doi: 10.1145/3664647.3681288.

[16] T. -H. Hsieh, L. Su and Y. -H. Yang, "A Streamlined Encoder/decoder Architecture for Melody Extraction," *ICASSP 2019 - 2019 IEEE International Conference on*

doi:10.6342/NTU202503400

Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 2019, pp. 156-160, doi: 10.1109/ICASSP.2019.8682389.

[17] S. Yong, L. Su and J. Nam, "A Phoneme-Informed Neural Network Model For Note-Level Singing Transcription," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10096707.

[18] X. Wang, W. Xu, W. Yang and W. Cheng, "Musicyolo: A Sight-Singing Onset/Offset Detection Framework Based on Object Detection Instead of Spectrum Frames," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, Singapore, 2022, pp. 396-400, doi: 10.1109/ICASSP43922.2022.9746684.

[19] B. Agüera y Arcas *et al.*, "Now Playing: Continuous low-power music recognition," Nov. 2017, [Online]. Available: https://arxiv.org/abs/1711.10958

[20] F. Liu, D. Tuo, Y. Xu and X. Han, "CoverHunter: Cover Song Identification with Refined Attention and Alignments," *2023 IEEE International Conference on Multimedia and Expo (ICME)*, Brisbane, Australia, 2023, pp. 1080-1085, doi: 10.1109/ICME55011.2023.00189.

[21] J. Xun *et al.*, "DisCover: Disentangled Music Representation Learning for Cover Song Identification," *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 453–463, Jul. 2023, doi: https://doi.org/10.1145/3539618.3591664.

[22] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, 2015, pp. 815-823, doi: 10.1109/CVPR.2015.7298682.

doi:10.6342/NTU202503400

[23] A. Gulati et al., “Conformer: Convolution-augmented transformer for speech recognition,” arXiv.org, May 16, 2020. https://arxiv.org/abs/2005.08100

[24] T. -Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal Loss for Dense Object Detection," *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, pp. 2999-3007, doi: 10.1109/ICCV.2017.324.

[25] S. Rouard, F. Massa, and A. Défossez, “Hybrid transformers for music source separation,” *arXiv.org*, Nov. 15, 2022. https://arxiv.org/abs/2211.08553

[26] S. Wang, X. Kong, H. Huang, K. Wang and Y. Hu, "HANet: A Harmonic Attention-Based Network for Singing Melody Extraction from Polyphonic Music," *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Hyderabad, India, 2025, pp. 1-5, doi: 10.1109/ICASSP49660.2025.10889955.

[27] R. Jang “MIR-QBSH-corpus,” MIR Lab, CS Dept., Tsing Hua Univ., Taiwan. Link: http://mirlab.org/dataSet/public/MIR-QBSH.zip

[28] R. Jang “MIR-ST500,” MIR Lab, CS Dept., Tsing Hua Univ., Taiwan. Link: http://mirlab.org/dataset/public/MIR-ST500_20201014.zip

[29] J. Z. M. Lim, "Query by Humming (QBH) audio dataset," Kaggle, 2021. [Dataset]. [Online]. Available: https://www.kaggle.com/datasets/limzhiminjessie/query-by-humming-qbh-audio-dataset

[30] K. Chen, S. Yu, C. -i. Wang, W. Li, T. Berg-Kirkpatrick and S. Dubnov, "Tonet: Tone-Octave Network for Singing Melody Extraction from Polyphonic Music," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Singapore, Singapore, 2022, pp. 621-625, doi: 10.1109/ICASSP43922.2022.9747304.

[31] Yaroslav Ganin *et al.*, “Domain-Adversarial Training of Neural Networks,” *Journal*

doi:10.6342/NTU202503400

of Machine Learning Research, vol. 17, no. 59, pp. 1–35, 2016. Available: http://www.jmlr.org/papers/v17/15-239.html

[32] J. Salamon, J. Serrà, and E. Gómez, "Tonal representations for music retrieval: from version identification to query-by-humming," *International Journal of Multimedia Information Retrieval*, vol. 2, no. 1, pp. 45–58, Dec. 2012, doi: 10.1007/s13735-012-0026-0.

doi:10.6342/NTU202503400