DisCover: Disentangled Music Representation Learning for Cover Song Identification

Jiahao Xun
Zhejiang University
Hangzhou, China
jhxun@zju.edu.cn

Shengyu Zhang[†]
Zhejiang University
Hangzhou, China
sy_zhang@zju.edu.cn

Yanting Yang
Zhejiang University
Hangzhou, China
yantingyang@zju.edu.cn

Jieming Zhu
Huawei Noah's Ark Lab
Shenzhen, China
jiemingzhu@ieee.org

Liqun Deng
Huawei Noah's Ark Lab
Shenzhen, China
dengliqun.deng@huawei.com

Zhou Zhao[†]
Zhejiang University
Hangzhou, China
zhaozhou@zju.edu.cn

Zhenhua Dong
Huawei Noah's Ark Lab
Shenzhen, China
dongzhenhua@huawei.com

Ruiqi Li
Zhejiang University
Hangzhou, China
rickyli@zju.edu.cn

Lichao Zhang
Zhejiang University
Hangzhou, China
zju_zlc@zju.edu.cn

Fei Wu
Zhejiang University
Hangzhou, China
wufei@zju.edu.cn

ABSTRACT

In the field of music information retrieval (MIR), cover song identification (CSI) is a challenging task that aims to identify cover versions of a query song from a massive collection. Existing works still suffer from high intra-song variances and inter-song correlations, due to the entangled nature of version-specific and version-invariant factors in their modeling. In this work, we set the goal of disentangling version-specific and version-invariant factors, which could make it easier for the model to learn invariant music representations for unseen query songs. We analyze the CSI task in a disentanglement view with the causal graph technique, and identify the intra-version and inter-version effects biasing the invariant learning. To block these effects, we propose the disentangled music representation learning framework (DisCover) for CSI. DisCover consists of two critical components: (1) Knowledge-guided Disentanglement Module (KDM) and (2) Gradient-based Adversarial Disentanglement Module (GADM), which block intra-version and inter-version biased effects, respectively. KDM minimizes the mutual information between the learned representations and version-variant factors that are identified with prior domain knowledge. GADM identifies version-variant factors by simulating the representation transitions between intra-song versions, and exploits adversarial distillation

for effect blocking. Extensive comparisons with best-performing methods and in-depth analysis demonstrate the effectiveness of DisCover and the and necessity of disentanglement for CSI.

CCS CONCEPTS

• **Information systems** → **Information retrieval**; • **Computing methodologies** → **Artificial intelligence**.

KEYWORDS

Cover Song Identification; Disentanglement Representation; Music Representation

1 INTRODUCTION

Nowadays, online digital music platforms, such as Spotify and Apple Musiccontain a massive number of music tracks for consumption, intensifying the need of music retrieval techniques for discovering related songs. One of the key techniques for music discovery is cover song identification (CSI), which aims to retrieve the cover versions from a music collection given a query song. Specially, a cover version/song is an alternative interpretation of the original version with different musical facets (*e.g.* timbre, key, tempo, or structure).

[†]Corresponding Authors.
The source code will be available at https://gitee.com/mindspore/models.

arXiv:2307.09775v1 [cs.IR] 19 Jul 2023

In real-world scenarios, the music collection can be massive and rapidly updated, potentially amplifying the intra-song variances (*c.f.* Figure 1) and inter-song correlations. These characteristics drive the CSI problem hard to handle due to the ubiquitous spurious correlations among songs of different collections. Intuitively, CSI requires a fine-grained analysis of music facets and semantics such that the intra-song correlations and inter-song differences can be adequately distinguished.

Figure 1: An illustration of the data distribution in CSI task with the raw waveform (top) and CQT spectrogram (bottom). (a) and (b) are two different cover version for the same song "Don't Let It Bring You Down" in Covers80 dataset.

Recently, with the development of artificial intelligence in other domains [1, 29, 50–53, 59, 60, 65], deep learning based CSI models have presented superior performance compared with traditional sequence matching methods [12, 31, 44, 45]. Most of them [10, 11, 16, 56, 62, 63] treat CSI as a classification task and utilize CNN-based architecture for music content understanding. Furthermore, some state-of-the-art works [10, 11, 16] explore metric learning techniques to narrow the gap between different cover versions of the same song and simultaneously expand the distance among the different version groups. Despite the significant advances made by these methods, we argue that song-specific and song-sharing musical factors are highly entangled in their modeling, thus being inadequate to distinguish unseen cover versions and songs. For example, as shown in Figure 1, the testing cover version (b) of the query song (a) shows significant differences in pitch/F0 (the orange curve in the spectrogram), timbre, or rhythm. If the model is unable to disentangle these factors and identifies them as version-variant, it might fail to generalize on this testing version and identify it as negative ones. On the other hand, if the model fails to disentangle and recognize version-invariant factors, it might falsely correlate some other songs with the given query based on the high similarity of version-specific factors. To bridge the gap, we set the goal of explicitly disentangling version-variant and version-invariant factors and thus learning invariant musical representations for unseen cover song identification.

To better understand the underlying mechanism of the disentanglement in CSI task, we resort the causal graph technique [34] for illustration (*c.f.* Figure 2). The nodes denote cause or effect factors. An edge $A \rightarrow B$ means the A can directly affect B.

- Firstly, we illustrate the causal graph from the model's perspective in Figure 2(a): Z_i denotes the set of factors that are specific to

(a) Model's perspective   (b) Searcher's perspective

Figure 2: Causal graph of CSI from different perspectives.

the i-th cover version and are mostly version-variant. X_i denotes the learned musical representation of the i-th cover version. Z denotes the set of version-invariant factors. Y_i denotes the retrieval results (*e.g.* a candidate playlist) given the learned representation X_i. Intuitively, during the co-training of various cover versions where different factors are highly entangled, Z_1 could have a direct effect on X_1 and also X_2, which is the musical representation of the second cover version. Therefore, Y_1 and Y_2 will be indirectly affected through causal path $Z_1 \rightarrow X_1 \rightarrow Y_1$ and $Z_1 \rightarrow X_2 \rightarrow Y_2$ respectively, which will lead to spurious correlations and mismatching during unseen cover song identification.

- Secondly, as illustrated in Figure 2(b), we further consider the causal graph from searcher's perspective. It is a relatively ideal causal graph that $\hat{X}_i$ is only affected by Z. In other words, version information has no effect on the learned music representation, such that intra-song versions can be adequately distinguished from the others.

In this work, we aim to develop a disentanglement framework that could realize the transition of models' underlying behavior from Figure 2(a) to Figure 2(b) for debiased and effective cover song identification. We identify two critical challenges in achieving disentanglement in CSI: (1) Mitigating the negative effect from cover information and extracting the commonness for the versions (cutoff $Z_i \rightarrow X_i \rightarrow Y_i$), which aims to make the model more focused on the version-invariant factors Z and learn invariant representations for different cover versions. (2) Identifying the differences between versions and alleviating the negative transfer (cutoff $Z_i \rightarrow X_j \rightarrow Y_j$), which attempts to bridge the intra-group gap and avoid biased representation learning. It is non-trivial to block paths $Z_i \rightarrow X_i \rightarrow Y_i$ and $Z_j \rightarrow X_i \rightarrow Y_i$ due to the implicit nature of version-specific factors Z_i, Z_j and the effects in deep neural networks. In this regard, we introduce a disentanglement module for identifying version-specific factors, followed by an effect-blocking module for learning invariant representations. As for the path $Z_i \rightarrow X_i \rightarrow Y_i$, disentangling Z_i is challenging without supervision signals since different factors (*e.g.* F0 and timbre) in raw music are highly entangled. In this regard, we introduce prior domain knowledge as guidance for disentanglement. As for the path $Z_j \rightarrow X_i \rightarrow Y_i$, the challenge lies in how to identify the factors Z_j in the j-th sample that could affect the representation learning of X_i. Intuitively, we regard the modified factors during the transition from X_j to X_i as version-variant factors that are critical to X_i in the j-th sample.

Technically, we propose a ***Dis***entangled music representation learning framework for ***Cover*** song identification, denoted as DisCover, which encapsulates two key components: (1) Knowledge-guided Disentanglement Module (**KDM**) and (2) Gradient-based Adversarial Disentanglement Module (**GADM**) for blocking biased

effects $Z_i \rightarrow X_i \rightarrow Y_i$ and $Z_j \rightarrow X_i \rightarrow Y_i$, respectively. KDM employs off-the-shelf music feature extractors as the domain knowledge for disentanglement, and minimizes the mutual information (MI) between the learned representations and version-variant factors. GADM identifies version-variant factors by simulating the representation transitions between intra-song versions and adopting gradient-based adaptive masking. Since the discrete-valued mask might distort the continuity of representations in the hypersphere, it would be less effective to use MI to measure the effect $Z_j \rightarrow X_i \rightarrow Y_i$. Instead, GADM incorporates an adversarial distillation sub-module for distribution-based effect blocking.

The main highlights of this work are summarized as follows:

- We analyze the cover song identification problem in a disentanglement view with causal graph, a powerful tool but is seldom used in the community. We identify the bad impact of version-variant factors with two effect paths that needed to be blocked.
- We propose the DisCover framework that disentangles version-variant factors among intra-song versions and blocks two biased effect paths via knowledge-guided MI minimization and gradient-based adversarial distillation.
- We conduct in-depth experimental analyses along on both quantitative and qualitative results, which have demonstrated the effectiveness and necessity of disentanglement for CSI.

2 RELATED WORK

2.1 Cover Song identification

With the increasing amount of music data on the Internet, cover song identification (CSI) has long been a popular task in the music information retrieval community. CSI aims to retrieve the cover versions of a given song in a dataset, which can also be seen as measuring the similarity between music signals without meta-information (*e.g.*, title, author, genre). Specifically, meta-information might ease the problem but also introduce spurious correlations that many different songs have quite similar or even the same short title. Moreover, users humming the query songs might not necessarily know/provide the meta-information. Overall, CSI as a challenging task has long attracted lots of researchers due to its potential applications in music representation learning [25, 57], retrieval [32, 46, 61] and recommendation [14, 35]. However, those cover songs may differ from the original song in key transposition, speed change, and structural variations, which challenges identifying the cover song. To solve these problems, [45] developed music sequences alignment algorithms for version identification by measuring the similarity between time series, and [13] generated fixed-length vectors for cover song identification. In addition, deep learning approaches are introduced to CSI. For instance, CNNs are utilized to measure the similarity matrix [2] or learn features [9, 37, 56, 62]. On this basis, TPPNet [62] uses a temporal pyramid pool to extract information at a different scale. CQTNet [63] proposes a special CNN architecture to extract musical representations and train the network through classification strategies. Although these methods have made significant progress, they ignore the entanglement of cover song representations and may incorrectly correlate some other songs with a given query. Thus, we propose a framework that disentangles version-variant factors among intra-song versions.

(a) Cutoff $Z_i \rightarrow X_i$ **(b) Cutoff $Z_i \rightarrow X_j$**

Figure 3: Interventions on causal graph of DisCover from the perspective of modelling.

2.2 Disentangled Representation Learning

Disentangled representation learning (DRL) focuses on encoding data points into separate independent embedding subspaces, where different subspaces represent different data attributes. To prevent information leakage from each other, the correlation between the two embedding parts is still required to be reduced. Some correlation-reducing methods mainly focus on Mutual Information (MI) minimization, where MI is a fundamental measure of the dependence between two random variables. To accurately estimate MI upper bound, CLUB [5] bridges mutual information estimation with contrastive learning. This method has gained a lot of attention and applications in scenarios such as domain adaption, style transfer, and causal inference. For instance, IDE-VC [64] and VQMIVC [49] achieves proper disentanglement of speech representations. MIM-DRCFR [4] learns disentangled representations for counterfactual regression. In addition, as analyzed in [3, 42, 66], the gradients of the final predicted score convey the task-discriminative information, which correctly identifies the task-relevant features. For instance, Grad-CAM [42] visualizes the importance of each class by leveraging the gradient information. On the basis of this, ToAlign [54] decomposes a source feature into a task-relevant one and a task-irrelevant one for performing the classification-oriented alignment. RSC [24] discards the task-relevant representations associated with the higher gradients. DropClass [8] uses gradient information to extract class-specific information from the entangled feature map. However, most of these works learn to disentangle representations from a single perspective. This paper blocks two biased effect paths via knowledge-guided MI minimization and gradient-based adversarial distillation.

2.3 Music Representation Learning

An effective musical representation is essential for learning different music-related tasks, such as music classification [7, 27, 36, 48, 55, 56], cover song identification [56, 58, 62, 63], music generation [17, 18, 28, 40]. Most of them rely on large amounts of labeled datasets to learn music representations. As the labeled datasets on which supervised learning methods require extensive manual labeling, it is often costly and time-consuming, leading to limitations in the performance of supervised learning methods. For this reason, some audio researchers have adopted a self-supervised learning approach to learning musical representations [41, 47, 57, 67]. For example, MusicBERT [67] models music self-representation with a multi-task learning framework. PEMR [57] proposes a positive-negative frame mask for music representation with contrastive learning. Many approaches to music representation learning focus on key pieces of music, while CSI focuses more on the whole song.

3 PROPOSED METHOD

3.1 PRELIMINARIES

Problem Formulation. Following the common practice in modern cover song identification task [56, 62], we formulate cover song identification as an information retrieval problem and specifically focus on music representation learning. We use q to denote one query song and $\mathcal{S} = \{s_i\}_{i=1,...,|\mathcal{S}|}$ to denote the song collections on an online music platform.

Given the query q, cover song identification aims to retrieve the most similar candidates $C = \{c_i\}_{i=1,...,k}$ from the song collections $\mathcal{S}$ in a top-k manner. A deep learning-based CSI model $f(\cdot)$ encodes the q and s_i into the fixed dimension representation q and s_i separately. Then we use cosine distance to calculate the similarity for all the pairs $P = \{(q, s_i)\}_{i=1,...,|S|}$. During testing and serving, top-k candidates C will be ranked by the similarity and displayed on the music platform in a position consistent with the rank.

Prior Knowledge Selection. There are usually multiple variations of musical facets for the cover version, such as timbre, key, tempo, timing, or structure [43]. Hence it meets a problem of how to select the appropriate musical facets as expert knowledge. Inspired by the common practice in the disentanglement-based voice conversion [38, 49] and singing voice synthesis [6] and other speech-related tasks [19–23], we consider that fundamental frequency (F0) and timbre are relatively more sensitive to cover versions among different facets since they often change when different artists perform the same piece of song/music. Therefore, we select the F0 and timbre as representatives of the prior knowledge in our work. Specifically, F0 is the musical pitch, representing the high or low notes in the song/music. Timbre describes the vocal characteristics of the artist or instrument, which strongly influences how song/music is heard by trained as well as untrained ears.

3.2 Framework Overview

To block the intra-version and inter-version biased effects for learning version-invariant representations, we propose DisCover, as shown in Figure 4. DisCover consists of two modules: (1) Knowledge-guided Disentanglement Module (KDM), which mitigates the negative effect from cover information and extracting the commonness for the versions (green area in the upper left of Figure 4). (2) Gradient-based Adversarial Disentanglement Module (GADM), which identifies the differences between versions and alleviates the negative transfer (blue area in the lower right of Figure 4). The two modules are jointly trained in a parallel manner.

3.3 Knowledge-guided Disentanglement

As shown in Figure 3(a), the Knowledge-guided Disentanglement module (KDM) aims to block the bias between intra-song versions (cutoff $Z_i \rightarrow X_i \rightarrow Y_i$), which attempts to make the model more focused on the version-invariant factors Z and learn invariant representations for different cover versions. Considering that the model is hard to identify the version-specific factors entangled in the representation, as mentioned in Sec 3.1, we introduce the prior knowledge (*e.g.* F0 and timbre) to serve as the teacher that provides version-variant factors Z_i. In contrast to the goal of knowledge transfer, the model aims to minimize the correlation between the

Figure 4: Schematic illustration of DisCover framework. KDM minimizes the MI between the learned representations and version-variant factors that are identified with prior domain knowledge. GADM identifies and decomposes version-variant factors by simulating the representation transitions between intra-song versions, and exploits adversarial distillation for effect blocking.

learned representations X_i and the version-variant factors Z_i. In this way, we can explicitly disentangle representation X_i from version-variant factors Z_i.

Here, we denote $x \in \mathbb{R}^{dim}$ as the learned representations and $z \in \{o, t\}$ as the knowledge bank of version-variant factors, where $o \in \mathbb{R}^{dim}$ represents the fundamental frequency (F0) features, $t \in \mathbb{R}^{dim}$ represents the timbre representations.

3.3.1 factors-invariant Representation Modeling. To minimize the correlation between the learned representations x and the version-variant factors z, we introduce mutual information (MI) to serves as the measurement, which is defined as the Kullback-Leibler (KL) divergence between their joint and marginal distributions as:

$$I(x; z) = \mathbb{E}_{p(x,z)}[\log \frac{p(z|x)}{p(x)}] \qquad (1)$$

Since the conditional distribution $p(z|x)$ is intractable, we adopt vCLUB [5] to approximate the upper bound of MI as:

$$I(x, z) = \mathbb{E}_{p(x,z)}[\log q_{\theta_{x,z}}(z|x)] - \mathbb{E}_{p(x)}\mathbb{E}_{p(z)}[\log q_{\theta_{x,z}}(z|x)] \quad (2)$$

where $q_{\theta_{x,z}}(\cdot)$ represents the variational estimation network between x and z. Therefore the unbiased estimation for vCLUB between learned representation and version-variant factors can be reformulated as:

$$\mathcal{L}_{I(x;o)} = \frac{1}{N}\sum_{i=1}^{N}[\log(q_{\theta_{x,o}}(o_i|x_i)) - \frac{1}{N}\sum_{j=1}^{N}\log(q_{\theta_{x,o}}(o_j|x_i))] \tag{3}$$

$$\mathcal{L}_{I(x;t)} = \frac{1}{N}\sum_{i=1}^{N}[\log(q_{\theta_{x,t}}(t_i|x_i)) - \frac{1}{N}\sum_{j=1}^{N}\log(q_{\theta_{x,t}}(t_j|x_i))] \tag{4}$$

where N represents the batch size. By minimizing the Eq. (3) and (4), we can decrease the correlation between learned representation and version-variant factors and the total MI loss is:

$$\mathcal{L}_{MI} = \mathcal{L}_{I(x;o)} + \mathcal{L}_{I(x;t)} \tag{5}$$

To obtain the reliable upper bound approximation, a robust variational estimator $q_{\theta_{x,z}}(\cdot)$ is required. We train the variational estimator by minimizing the log-likelihood:

$$\mathcal{L}_{q_{\theta_{x,z}}} = -\frac{1}{N} \sum_{i=1}^{N} [\log(q_{\theta_{x,z}}(x|z))], z \in \{o, t\} \qquad (6)$$

3.3.2 knowledge tradeoff. However, we argue that vCLUB might be at risk of posterior collapse [15, 30, 39] due to the KL-Vanishinig. For example, if the weights of the variational estimator become randomized due to undesirable training, the introduction of prior knowledge would be meaningless. Therefore, knowledge tradeoff is the self-supervised way to relieve the posterior collapse and ensure training stability for variational estimator. Furthermore, considering knowledge extractors' ability, little beneficial version-invariant information might still remain in the z. To address these concerns, we provide two alternatively simple methods. Firstly, we can fuse task-oriented representation e as:

$$a = \sigma(g(e, q(z))), \qquad (7)$$

$$e^* = a * e + (1 - a) * q(z) \qquad (8)$$

where $\sigma(\cdot)$ denotes the sigmoid function, $g(\cdot)$ is the linear transformation, $q(\cdot)$ is the shared MLP in the variational estimator, and $a \in \mathbb{R}$ serves as the tradeoff between e and $q(z)$. Secondly, we can use clustering models (*e.g.* k-means) to annotate the pseudo labels for z to supervise the variational estimator with classification task:

$$\mathcal{L}_{z_{cls}} = -\sum_{i=1}^{N} y_{z_i} \log(\hat{y}_{z_i}) + (1 - y_{z_i}) \log(1 - \hat{y}_{z_i}) \qquad (9)$$

where y_{z_i} is the pseudo label for z_i, and $\hat{y}_{z_i}$ is the output of the knowledge classifier.

3.4 Gradient-based Adversarial Disentanglement

As shown in Figure 3(b), the Gradient-based Adversarial Disentanglement module (GADM) aims to block the bias between inter-song versions (cutoff $Z_j \rightarrow X_i \rightarrow Y_i$), which attempts to bridge the intra-group gap and avoid biased representation learning. As analyzed in [3, 42, 66], the gradients of the predictive score contain the discriminative information for the downstream tasks. Analogously, the gradients of the transition cost between two versions might convey important information for version-variant factors. For this purpose, we randomly construct the positive query-target pairs with different versions and obtain the corresponding representation pairs (x, x^+) with the same backbone model. GADM has three main steps: identification, decomposition, and alignment.

3.4.1 Identification. The main idea of identification is to recognize the version-variant factors that are entangled in the elements of learned representations. Since the backbone encoder maps the samples into the hyperspace, positive representation pairs x and x^+ can be regarded as two points in the same high dimensional space. In the ideal case, different versions of the same song should have similar representations. In other words, these points should cluster together in the hyperspace. However, the distance between two points would be enlarged due to the disruption of version-variant

factors that are highly entangled in the representations. Therefore, we treat the distance between query-target pair x and x^+ as the transition cost caused by entangled version-variant factors. Here, we can use metric function (*e.g.* Euclidean, Manhattan, or Cosine) to serve as the transitions cost $C_{trans} \in \mathbb{R}^+$ between the representations of intra-song versions x and x^+ as:

$$C_{trans} = h(x, x^+) \qquad (10)$$

where $h(\cdot, \cdot)$ denotes the metric function. Motivated by the GradCAM-like methods [3, 42, 66], which utilize the saliency-based class information from the gradient perspective. We can obtain version-variant information by calculating the gradients of the transition cost C_{trans} *w.r.t.* the representation x as:

$$g_x = \frac{\partial C_{trans}}{\partial x} \qquad (11)$$

where $g_x \in \mathbb{R}^{dim}$ denotes the gradient vector of x. Since the partial derivative operation for query x utilizes the information from target x^+, gradient vector g_x probably conveys the element-wise importance information of representation x for measuring the difference to its target x^+. Specifically, as shown in the bottom right corner of Figure 4, each element $g_x(i)$ in g_x represents the fusion result between query element $x(i)$ and whole target representation x^+. The process allows element $g_x(i)$ to automatically search for the elements of the query representation x that are relevant to the transition cost. That's why g_x can identify the version-variant factors hiding in the x^+. Furthermore, the value of $g_x(i)$ represents the sensitivity to the changes of transition cost C_{trans}, where the element with the higher value is more relevant to the version-variant factors based on the nature of gradient.

3.4.2 Decomposition. After identifying the version-variant factors, we attempt to decompose the version-invariant representation $\hat{x}$ from x. Inspired by ToAlign [54], which decomposes a source feature into a task-relevant/irrelevant one with a gradient-based attention weight vector. We further exploit the numeric order in g_x to ensure that the element with the higher gradient has the lower attention weight. Specifically, given the gradient vector g_x, we will construct the corresponding mask vector and decompose it as:

$$m_x(i) = \begin{cases} 1 - \dfrac{\exp(g_x(i))}{\sum_{k \in \{k|g_x(k) \geq q_p\}} \exp(g_x(k))}, & if \ g_x(i) \geq q_p \\ 1, & otherwise \end{cases} \qquad (12)$$

$$\hat{x} = m_x \odot x \qquad (13)$$

where $m_x(i)$ denotes i-th element in the mask, q_p denotes the p-th largest percentile in g_x, and $\odot$ denotes the hadamard product. Moreover, in view of the self-challenging method [24], the decomposition process adaptively re-weights x based on the knowledge from g_x and forces the backbone to lower the attention on version-specific elements, so as to obtain the version-invariant representation $\hat{x}$.

3.4.3 Alignment. To alleviate the negative transfer, we adopt the adversarial distillation sub-module to align entangled representation x to the disentangled one $\hat{x}$. In the beginning, x and $\hat{x}$ belong to different hyperspheres, where the x is considered as the negative source and the $\hat{x}$ is the positive target. We use them to train the discriminator D to distinguish which hypersphere the representation belongs to, with the classification loss $\mathcal{L}_{D_1}$. Meanwhile, the

backbone encoder is trained to fool the discriminator to learn the version-invariant representation by minimizing task-oriented loss while maximizing $\mathcal{L}_{D_2}$:

$$\mathcal{L}_{D_1} = \frac{1}{N} \sum_{i=1}^{N} [\log D(\hat{x}_i) + \log (1 - D(x_i))] \qquad (14)$$

$$\mathcal{L}_{D_2} = \frac{1}{N} \sum_{i=1}^{N} [\log (1 - D(x_i))] \qquad (15)$$

Furthermore, considering the symmetry of the query-target pair, we can similarly obtain the version-invariant target representation $\hat{x}^+ \in \mathbb{R}^{dim}$. To ensure the semantic consistency between query and target, it is better to minimize transition cost as:

$$\mathcal{L}_{trans} = h(\hat{x}, \hat{x}^+) \qquad (16)$$

3.5 Training

Given the output of the task-oriented classifier $\hat{y}_{\hat{x}}$, we treat CSI as the classification task, where the task-oriented learning objective can be formulated as follows:

$$\mathcal{L}_{task} = - \sum_{i=1}^{N} y_{\hat{x}} log(\hat{y}_{\hat{x}}) + (1 - y_{\hat{x}}) log(1 - \hat{y}_{\hat{x}}) \qquad (17)$$

where $y_{\hat{x}}$ is the groundtruth label. To be clear, the overall optimization objective of our proposed DisCover is summarized as follows:

$$\mathcal{L}_1 = \mathcal{L}_{task} + \mathcal{L}_{trans} + \lambda_1 \mathcal{L}_{MI} + \mathcal{L}_{z_{cls}} - \mathcal{L}_{D_2} \qquad (18)$$

$$\mathcal{L}_2 = \mathcal{L}_{D_1} + \lambda_2 \mathcal{L}_{q_{\theta_{x,z}}} \qquad (19)$$

where $\mathcal{L}_1$ and $\mathcal{L}_2$ are optimized alternately.

4 EXPERIMENTS

We analyze the DisCover framework and demonstrate its effectiveness by answering the following research questions:

- **RQ1**: How does DisCover perform compared with existing best-performing cover song identification methods in different scenarios (*e.g.*, unseen songs/versions) ?
- **RQ2**: Do knowledge-guided disentanglement and gradient-based disentanglement all contribute to the effectiveness over various base models in a model-agnostic manner?
- **RQ3**: How does different architecture and hyper-parameter settings will affect the performance of DisCover?
- **RQ4**: Does DisCover disentangle the version-variant factors?

Table 1: Dataset statics

Dataset	Songs	Recordings	Avg. versions	Language
SHS100K	10000	104641	10.5	English
Karaoke30K	11500	31629	2.8	Chinese
Covers80	80	160	2.0	English

4.1 Experimental Setting

4.1.1 Dataset. We conduct experiments on two open source datasets commonly used in cover song identification and one self-collected real-world dataset. Statistics of these datasets are shown in Table 1.

- **Second Hand Songs 100K (SHS100K)**: We downloaded raw audios through youtube-dl[2] using the URLs provided on GitHub[3]. It has 10000 songs with 104641 recordings. Notably, there are 25% of test songs seen during model training in the setting of [62]. To further explore the generalization performance, we also construct another scenario setting where all test songs are unseen during training. For both scenarios, the ratio among the training set, validation set, and testing set is 8:1:1.
- **Covers80**[4]: It has 80 songs with 160 recordings, where each song has 2 cover versions. Due to the small amount of data, it is commonly used only for evaluating models.
- **Karaoke30K**: A real-world Chinese karaoke dataset collected by ourselves. It has 11500 songs with 31629 recordings, where each song has 1 to 3 cover versions. Following the SHS100K, we also construct the two scenarios with the same setting.

4.1.2 Evaluation Metrics. Following the evaluation protocol of the Mirex Audio Cover Song Identification Contest[5], we employ three widely used metrics for evaluation, *i.e.*, MAP (mean average precision), P@10 (precision at 10), and MR1 (mean rank of the first correctly identified cover).

4.1.3 Comparison Baselines.

- **2DFM** [13]: 2DFM transforms a beat-synchronous chroma matrix with a 2D Fourier transformer and poses the search for cover songs as estimating the Euclidean distance.
- **ki-CNN** [56]: ki-CNN uses a key-invariant convolutional neural network robust against key transposition for classification.
- **TPPNet** [62]: TPPNet combines CNN architecture with temporal pyramid pooling to extract information on different scales and transform songs with different lengths into fixed-dimensional representations.
- **CQTNet** [63]: CQTNet uses carefully designed kernels and dilated convolutions to extend the receptive field, which can improve the model's representation learning capacity.
- **PICKiNet** [33]: PICKiNet devises pitch class blocks to obtain the key-invariant musical features.

4.1.4 Implementation Details. We train models on the SHS100K and Karaoke30K and report the evaluation metrics on them with different scenarios. Covers80 is used to evaluate the models trained on SHS100K since their languages are the same. We use parselmouth[6] and resemblyzer[7] to extract F0 and timbre respectively. In KDM, we apply Eq. (8) to F0 feature and Eq. (9) to timbre representation, where the number of the clusters for generating pseudo label $N = 100$. Following the default MI-related setting in [49], we set hyper-parameters $\lambda_1 = 0.05$, $\lambda_2 = 1$. In GADM, we select Euclidean distance as the metric function, and the mask ratio is set

---

[2]https://github.com/ytdl-org/youtube-dl
[3]https://github.com/NovaFrost/SHS100K2
[4]https://labrosa.ee.columbia.edu/projects/coversongs/covers80/
[5]https://www.music-ir.org/mirex/wiki/2020:Audio_Cover_Song_Identification
[6]https://github.com/YannickJadoul/Parselmouth
[7]https://github.com/resemble-ai/Resemblyzer

Table 2: Improvement over the best-performing baselines across different scenarios.

| Model | SHS100K | | | | | | Covers80 | | | | | |
| | Scenario 1 : | | | Scenario 2 : | | | Scenario 1 : | | | Scenario 2 : | | |
	MAP↑	P@10↑	MR1↓	MAP↑	P@10↑	MR1↓	MAP↑	P@10↑	MR1↓	MAP↑	P@10↑	MR1↓
2DFM	-	-	-	0.104	0.113	415	-	-	-	0.381	0.053	33.60
Ki-CNN	0.176	0.224	105.79	0.215	0.183	147.3	0.485	0.069	16.18	0.509	0.071	15.45
TPPNet	0.419	0.455	45.85	0.471	0.338	74.38	0.757	0.084	5.81	0.786	0.087	8.39
CQTNet	0.571	0.573	**31.69**	0.624	0.340	**61.31**	0.805	0.087	6.58	0.846	0.089	5.13
PICKiNet	0.617	0.602	38.66	0.626	0.408	84.12	0.818	0.085	7.11	0.858	0.091	4.27
TPPNet-Dis	0.565	0.567	41.60	0.561	0.384	74.26	0.814	0.091	7.81	0.849	0.091	4.74
CQTNet-Dis	**0.658**	0.627	37.98	0.640	0.417	76.41	**0.856**	**0.091**	**5.11**	**0.912**	**0.095**	**2.43**
PICKiNet-Dis	0.657	**0.627**	46.80	**0.653**	**0.421**	72.30	0.830	0.087	5.88	0.882	0.093	3.26

Table 3: Comparing different methods on Karaoke30K with different scenarios.

| Model | Scenario 1 : | | | Scenario 2 : | | |
	MAP↑	P@10↑	MR1↓	MAP↑	P@10↑	MR1↓
Ki-CNN	0.483	0.119	52.53	0.524	0.116	52.01
TPPNet	0.760	0.165	17.65	0.777	0.154	13.86
CQTNet	0.863	0.182	7.84	0.831	0.161	11.93
PICKiNet	0.944	0.194	4.41	0.959	0.178	4.12
TPPNet-Dis	0.935	0.192	5.23	0.957	0.177	3.24
CQTNet-Dis	**0.976**	0.198	2.66	**0.983**	**0.180**	**3.20**
PICKiNet-Dis	0.974	**0.198**	**2.61**	0.973	0.179	3.52

to 1. Following the setting of [62], we also apply a multi-length training strategy. Adam [26] is used as the optimizer for backbone, discriminator, and variational estimator. The training batch size N is 32, initial learning rate is 4e-4, weight decay is 1e-5. Notably, LayerNorm is applied in DisCover to obtain normalized representation to ensure numerical stability in similarity-based retrieval.

4.2 Overall Results (RQ1)

We instantiate the proposed DisCover framework on three best-performing CSI methods, i.e., TPPNet, CQTNet, and PICKiNet, and obtain TPPNet-Dis, CQTNet-Dis and PICKiNet-Dis. Table 2 and 3 list the comparison results of the best-performing models and those enhanced by DisCover on the SHS100K, Karaoke30K and Covers80 datasets under two different scenarios. Specifically, in Scenario #1, all test songs are unseen during training, while in Scenario #2, models have seen 25% class of test songs during training. According to the results, we have the following observations:

- Overall, the results across multiple evaluation metrics consistently indicate that TPPNet-Dis, CQTNet-Dis, and PICKiNet-Dis achieve better results than their base models among different datasets and scenarios. Especially, CQTNet-Dis and PICKiNet-Dis show comparable performance and outperform other best-performing methods. We attribute the improvements to the fact that baselines succeed in learning the version-invariant representations by disentangling version-specific musical factors.

- DisCover can boost the performance of models in different scenarios, especially in scenario #1, where all test songs are unseen. It suggests that the version-variant factors have been highly disentangled. In addition, in Karaoke30k where the cover versions of a particular song are fewer, DisCover could still significantly improve the baselines. These results demonstrate the practical merits of DisCover, i.e., identifying version-variant factors with limited number of annotated versions. Note that in real-world scenarios, less popular songs constitute the majority of the music collections and have fewer cover versions. In summary, these results demonstrate the strengths of DisCover in generalization and few-shot learning, which is critical for industrial scenarios where music collections could be rapidly updated and too massive to sample the full cover versions for training.

- Surprisingly, MR1 scores in scenario #1 are mostly worse than those in scenario #2 for all models, especially in SHS100K. These results might suggest that entangled training leads to spurious correlations among songs, including those testing songs seen during training. We also observe that on the SHS100K dataset, the proposed method could not beat some baselines w.r.t. MR1. SHS100K are known to have unusual audio manifestations in recordings and vocal concert songs (with strong background noises e.g. claps, shouts, or whistles), where MR1 scores are sensitive to these noises and exhibit high variances. On the Karaoke30K dataset where the manifestations in recordings are closer to real-world search scenarios, we observe consistent performance improvement brought by DisCover across all metrics.

4.3 Model Analysis (RQ2, RQ3)

4.3.1 Analysis of key building modules. knowledge-guided disentanglement and gradient-based adversarial disentanglement are two key components of DisCover framework. We conduct the ablation study on them to reveal the efficacy of the architectures and the benefits of disentangling version-variant factors. Specifically, we selectively discard the KDM and GADM from CQTNet-Dis and TPPNet-Dis to obtain ablation architectures, i.e., w/o. KDM, and w/o. GADM, respectively to show the model-agnostic capability of these two modules. The results are shown in Table 4. We can observe that:

Table 4: Ablation studies by selectively discarding the knowledge-guided disentanglement module (w/o. KDM) and gradient-based adversarial disentanglement module (w/o. GADM). We study both TPPNet-Dis and CQTNet-Dis on different datasets to reveal the model-agnostic capability of the proposed modules.

Scenario 1 :	SHS100K			Covers80			Karaoke30K		
Model	MAP↑	P@10↑	MR1↓	MAP↑	P@10↑	MR1↓	MAP↑	P@10↑	MR1↓
TPPNet-Dis	0.565	0.567	41.60	0.814	0.091	7.81	0.935	0.192	5.23
w/o. KDM	0.542	0.551	39.98	0.805	0.085	7.06	0.921	0.190	3.79
w/o. GADM	0.497	0.522	48.55	0.790	0.087	8.34	0.845	0.180	8.52
TPPNet	0.419	0.455	45.85	0.757	0.084	5.81	0.760	0.165	17.65
CQTNet-Dis	0.658	0.627	37.98	0.856	0.091	5.11	0.976	0.198	2.66
w/o. KDM	0.649	0.622	32.34	0.843	0.093	4.45	0.961	0.196	3.73
w/o. GADM	0.619	0.607	36.66	0.833	0.088	7.17	0.887	0.186	7.88
CQTNet	0.571	0.573	31.69	0.805	0.087	6.58	0.863	0.182	7.84

Table 5: Study of different prior knowledge. The disentanglement of both F0 and timbre can be beneficial.

Scenario 1:	SHS100K			Covers80		
Factors	MAP↑	P@10↑	MR1↓	MAP↑	P@10↑	MR1↓
TPPNet	0.420	0.454	44.33	0.757	0.084	5.81
F0	0.457	0.485	48.20	0.764	0.086	7.54
w/. tradeoff	0.463	0.490	43.65	0.778	0.086	8.31
Timbre	0.443	0.475	55.75	0.772	0.086	8.38
w/. tradeoff	0.466	0.495	52.69	0.784	0.088	7.14
Timbre & F0	0.469	0.496	43.96	0.782	0.086	8.73
w/. tradeoff	0.497	0.522	48.55	0.790	0.087	8.34
CQTNet	0.569	0.572	31.90	0.805	0.087	6.58
F0	0.585	0.580	34.52	0.814	0.093	4.00
w/. tradeoff	0.603	0.597	34.46	0.821	0.091	3.97
Timbre	0.586	0.585	38.17	0.816	0.093	4.59
w/. tradeoff	0.606	0.599	37.17	0.829	0.094	4.53
Timbre & F0	0.590	0.586	37.83	0.824	0.089	5.51
w/. tradeoff	0.619	0.607	36.66	0.833	0.088	7.17

- Removing either KDM or GADM leads to performance degradation, while removing both modules (*i.e.*, the base model) leads to the worst performance. These results demonstrate the effectiveness of the proposed two modules as well as the benefits of disentanglement for CSI. We attribute this superiority to the fact that the models would absorb less spurious correlations among songs and versions by learning version-invariant representations and blocking intra/inter-version biased effects.
- Removing GADM leads to more performance drops than removing KDM, which indicates that introduced prior knowledge only contains the part of the version-variant factors. Therefore it is necessary to identify the remained factors that hide in the representation. These results again verify the effectiveness of the end-to-end disentanglement module GADM.

Table 6: Analysis of the number of clustering centers N for timbre in knowledge tradeoff on CQTNet and TPPNet under scenario #1.

Model	N	SHS100K			Covers80		
		MAP↑	P@10↑	MR1↓	MAP↑	P@10↑	MR1↓
TPPNet	100	0.466	0.495	52.69	0.784	0.088	7.14
	1K	0.463	0.492	43.06	0.785	0.088	9.83
	5K	0.462	0.492	44.06	0.792	0.084	8.68
	10K	0.467	0.496	44.46	0.777	0.086	8.14
CQTNet	100	0.606	0.599	37.17	0.829	0.094	4.53
	1K	0.601	0.594	32.39	0.827	0.092	2.95
	5K	0.593	0.590	38.47	0.832	0.096	3.32
	10K	0.609	0.602	34.34	0.833	0.092	4.18

- The results are consistent across different baselines, which indicates that the proposed two modules can easily boost the best-performing CSI baselines in a plug-and-play and model-agnostic manner.

4.3.2 Study of different prior knowledge introduced in KDM. F0 and timbre are two commonly used features in singing voice conversion/synthesis tasks, which can reflect music pitch and voice characteristics, respectively. To further study the impact of different prior knowledge, we selectively use F0 and timbre to serve as the version-variant factors. We conduct experiments on CQTNet and TPPNet with SHS100K dataset. The results are shown in table 5 where we can find that:

- Introducing either F0 or timbre can improve the baseline performance and introducing both of them will achieve better results. These results further demonstrate the effectiveness of minimizing the correlation between the learned representations and the version-variant factors.
- Different verison-variant factors play different roles in exerting a bad impact on model learning. Compared with F0, disentangling timbre appears to be more beneficial to the baseline models. The reason might be that voice characteristic vary from person to person, which leads to high intra-song variances among versions that are performed by different people.
- Learning with knowledge tradeoff leads to better performance with different baselines and datasets, which suggests that this technique can further exploit the useful information hiding in prior knowledge and is helpful in relieving the posterior collapse of variational estimator [15, 30, 39].

4.3.3 Analysis of the number of clustering centers for timbre in knowledge tradeoff. In this experiment, we analyze the impact of the number (N) of clusters used to generate pseudo-labels on the model performance, which uncovers the hyper-parameter sensitivity. As shown in Table 6, the model performance is overall insensitive to the number of clusters. In other words, the model can achieve comparable performance with relatively few pseudo-labels (*e.g.* N = 100) and lower complexity, which is suitable for real-world scenarios to reduce resource consumption.

4.3.4 Analysis of transition simulation in GADM. As analyzed in Sec. 3.4.1, we use metric function to serve as the transition cost

Table 7: Analysis of transition simulation in GADM on CQT-Net and TPPNet under scenario #1.

Model	Method	SHS100K			Covers80		
		MAP↑	P@10↑	MR1↓	MAP↑	P@10↑	MR1↓
TPPNet	Manhattan	0.344	0.389	79.87	0.724	0.082	8.58
	Euclidean	0.542	0.551	39.98	0.805	0.085	7.06
	Cosine	0.473	0.495	44.08	0.730	0.084	10.35
CQTNet	Manhattan	0.620	0.603	34.01	0.814	0.089	4.41
	Euclidean	0.649	0.622	32.34	0.843	0.093	4.45
	Cosine	0.525	0.538	48.75	0.784	0.091	5.15

between two versions of a song. Therefore a reliable metric function is vital for identifying the version-variant factors between different versions. In this experiment, we select three commonly used distances (*e.g.* Euclidean, Manhattan, and Cosine) to serve as the transition cost. Surprisingly, as shown in Table 7, the Euclidean distance, which is less explored in the CSI literature, shows a clear advantage over other widely used metric functions. This is an interesting finding that might be potentially inspirational. We plan to further uncover the underlying mechanisms in the future.

4.4 Qualitative Analysis (RQ4)

The above analysis quantitatively shows the effectiveness of disentanglement in cover song identification. To evaluate whether the model can learn the version-invariant and unbiased representations via disentangled learning, we visualize the t-SNE transformed embeddings. We adopt CQTNet, TPPNet and PICKiNet as baseline and equip them with DisCover framework and plot the twenty randomly sampled songs and each song has three versions with the representations encoded by the corresponding model. As shown in Figure 5, we can observe that:

- Overall, different versions of a song exhibit more noticeable clusters with the help of DisCover. The base model is more likely to falsely correlate songs based on the similarity of version-variant factors. For example, the versions of song #12 in Figure 5(a) are closer to the other songs, which suggests that CQTNet fail to learn the discriminative representation for them. However, in Figure 5(b), different versions of song #12 are more compact, which demonstrates the capability of disentanglement.

- Moreover, equipped with DisCover, all of CQTNet, TPPNet and PICKiNet show better performance in learning more discriminative representations compared to the baselines, which further reveals the model-agnostic capability of DisCover.

- Furthermore, although the training samples for each song are limited (2 to 3 cover versions for a song), DisCover can still learn the discriminative representations for unseen songs. These results again verify the strengths of DisCover in generalization and few-shot learning.

5 CONCLUSION

In this paper, we first analyze the cover song identification problem in a disentanglement view with causal graph. We identify the bad impact of version-variant factors with two effect paths that need to be blocked. Then, we propose the disentangled music representation learning framework DisCover to block these effects. DisCover consists of two modules: (1) Knowledge-guided Disentanglement

(a) CQTNet Baseline

(b) CQTNet with DisCover

(c) TPPNet Baseline

(d) TPPNet with DisCover

(e) PICKiNet Baseline

(f) PICKiNet with DisCover

Figure 5: Case study with t-SNE transformed embeddings derived from different baselines with our DisCover framework, where colored nodes represent the different songs.

module, it mitigates the negative effect of cover information and extracts the commonness for the versions, which makes the model more focused on the version-invariant factors and learning invariant representations for different cover versions. (2) Gradient-based Adversarial Disentanglement module, it identifies the differences between versions and alleviates the negative transfer, which bridges the intra-group gap and avoids biased representation learning. Extensive comparisons with best-performing methods and in-depth analysis demonstrate the effectiveness of DisCover and the necessity of disentanglement for CSI.

6 ACKNOWLEDGEMENTS

---

[8]https://www.mindspore.cn

REFERENCES

[1] Yin Aoxiong, Zhong Tianyun, Tang Li, Jin Weike, Jin Tao, and Zhao Zhou. 2023. Gloss Attention for Gloss-free Sign Language Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE.

[2] Sungkyun Chang, Juheon Lee, Sang Choe, and Kyogu Lee. 2017. Audio Cover Song Identification using Convolutional Neural Network.

[3] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. 2018. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 839–847.

[4] Mingyuan Cheng, Xinru Liao, Quan Liu, Bin Ma, Jian Xu, and Bo Zheng. 2022. Learning Disentangled Representations for Counterfactual Regression via Mutual Information Minimization. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1802–1806.

[5] Pengyu Cheng, Weituo Hao, Shuyang Dai, Jiachang Liu, Zhe Gan, and Lawrence Carin. 2020. Club: A contrastive log-ratio upper bound of mutual information. In *International conference on machine learning*. PMLR, 1779–1788.

[6] Hyeong-Seok Choi, Jinhyeok Yang, Juheon Lee, and Hyeongju Kim. 2022. NANSY++: Unified Voice Synthesis with Neural Analysis and Synthesis. *arXiv preprint arXiv:2211.09407* (2022).

[7] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. 2017. Convolutional recurrent neural networks for music classification. In *2017 IEEE International conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2392–2396.

[8] Sanghyeok Chu, Dongwan Kim, and Bohyung Han. 2021. Learning Debiased and Disentangled Representations for Semantic Segmentation. *Advances in Neural Information Processing Systems* 34 (2021), 8355–8366.

[9] Guillaume Doras and Geoffroy Peeters. 2019. Cover detection using dominant melody embeddings. In *ISMIR 2019*.

[10] Xingjian Du, Ke Chen, Zijie Wang, Bilei Zhu, and Zejun Ma. 2022. Bytecover2: Towards dimensionality reduction of latent embedding for efficient cover song identification. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 616–620.

[11] Xingjian Du, Zhesong Yu, Bilei Zhu, Xiaoou Chen, and Zejun Ma. 2021. Bytecover: Cover song identification via multi-loss training. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 551–555.

[12] Daniel PW Ellis and Graham E Poliner. 2007. Identifying cover songs' with chroma features and dynamic programming beat tracking. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07*, Vol. 4. IEEE, IV–1429.

[13] Daniel PW Ellis and Bertin-Mahieux Thierry. 2012. Large-scale cover song recognition using the 2d fourier transform magnitude. (2012).

[14] Casper Hansen, Christian Hansen, Lucas Maystre, Rishabh Mehrotra, Brian Brost, Federico Tomasi, and Mounia Lalmas. 2020. Contextual and sequential user embeddings for large-scale music recommendation. In *Fourteenth ACM conference on recommender systems*. 53–62.

[15] Junxian He, Daniel Spokoyny, Graham Neubig, and Taylor Berg-Kirkpatrick. 2019. Lagging Inference Networks and Posterior Collapse in Variational Autoencoders. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[16] Shichao Hu, Bin Zhang, Jinhong Lu, Yiliang Jiang, Wucheng Wang, Lingcheng Kong, Weifeng Zhao, and Tao Jiang. 2022. WideResNet with Joint Representation Learning and Data Augmentation for Cover Song Identification. In *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*. ISCA, 4187–4191.

[17] Rongjie Huang, Feiyang Chen, Yi Ren, Jinglin Liu, Chenye Cui, and Zhou Zhao. 2021. Multi-singer: Fast multi-singer singing voice vocoder with a large-scale corpus. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3945–3954.

[18] Rongjie Huang, Chenye Cui, Feiyang Chen, Yi Ren, Jinglin Liu, Zhou Zhao, Baoxing Huai, and Zhefeng Wang. 2022. Singgan: Generative adversarial network for high-fidelity singing voice generation. In *Proceedings of the 30th ACM International Conference on Multimedia*. 2525–2535.

[19] Rongjie Huang, Jiawei Huang, Dongchao Yang, Yi Ren, Luping Liu, Mingze Li, Zhenhui Ye, Jinglin Liu, Xiang Yin, and Zhou Zhao. 2023. Make-an-audio: Text-to-audio generation with prompt-enhanced diffusion models. *arXiv preprint arXiv:2301.12661* (2023).

[20] Rongjie Huang, Max WY Lam, Jun Wang, Dan Su, Dong Yu, Yi Ren, and Zhou Zhao. 2022. FastDiff: A Fast Conditional Diffusion Model for High-Quality Speech Synthesis. *arXiv preprint arXiv:2204.09934* (2022).

[21] Rongjie Huang, Yi Ren, Jinglin Liu, Chenye Cui, and Zhou Zhao. 2022. GenerSpeech: towards style transfer for generalizable out-of-domain text-to-speech. *Advances in Neural Information Processing Systems* 35 (2022), 10970–10983.

[22] Rongjie Huang, Zhou Zhao, Huadai Liu, Jinglin Liu, Chenye Cui, and Yi Ren. 2022. Prodiff: Progressive fast diffusion model for high-quality text-to-speech. In *Proceedings of the 30th ACM International Conference on Multimedia*. 2595–2605.

[23] Rongjie Huang, Zhou Zhao, Jinglin Liu, Huadai Liu, Yi Ren, Lichao Zhang, and Jinzheng He. 2022. TranSpeech: Speech-to-Speech Translation With Bilateral Perturbation. *arXiv preprint arXiv:2205.12523* (2022).

[24] Zeyi Huang, Haohan Wang, Eric P Xing, and Dong Huang. 2020. Self-challenging improves cross-domain generalization. In *European Conference on Computer Vision*. Springer, 124–140.

[25] Junyan Jiang, Gus G Xia, Dave B Carlton, Chris N Anderson, and Ryan H Miyakawa. 2020. Transformer VAE: A hierarchical model for structure-aware and interpretable music representation learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 516–520.

[26] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization.. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[27] Jongpil Lee, Jiyoung Park, Luke Kim, and Juhan Nam. 2017. Sample-level Deep Convolutional Neural Networks for Music auto-tagging Using Raw Waveforms. In *The 14th Sound and Music Computing Conference*. SMCNetwork.

[28] Jinglin Liu, Chengxi Li, Yi Ren, Feiyang Chen, and Zhou Zhao. 2022. Diffsinger: Singing voice synthesis via shallow diffusion mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 11020–11028.

[29] Luping Liu, Yi Ren, Zhijie Lin, and Zhou Zhao. 2022. Pseudo numerical methods for diffusion models on manifolds. *arXiv preprint arXiv:2202.09778* (2022).

[30] James Lucas, George Tucker, Roger B. Grosse, and Mohammad Norouzi. 2019. Don't Blame the ELBO! A Linear VAE Perspective on Posterior Collapse. (2019), 9403–9413.

[31] Matija Marolt. 2006. A Mid-level Melody-based Representation for Calculating Audio Similarity.. In *ISMIR*. Citeseer, 280–285.

[32] Meinard Müller, Andreas Arzt, Stefan Balke, Matthias Dorfer, and Gerhard Widmer. 2018. Cross-modal music retrieval and applications: An overview of key methodologies. *IEEE Signal Processing Magazine* 36, 1 (2018), 52–62.

[33] Ken O'Hanlon, Emmanouil Benetos, and Simon Dixon. 2021. Detecting Cover Songs with Pitch Class Key-Invariant Networks. In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 1–6.

[34] Judea Pearl. 2009. *Causality*. Cambridge university press.

[35] Bruno L Pereira, Alberto Ueda, Gustavo Penha, Rodrygo LT Santos, and Nivio Ziviani. 2019. Online learning to rank for sequential music recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 237–245.

[36] Jordi Pons and Xavier Serra. 2019. musicnn: Pre-trained convolutional neural networks for music audio tagging. *arXiv preprint arXiv:1909.06654* (2019).

[37] Xiaoyu Qi, Deshun Yang, and Xiaoou Chen. 2017. Audio feature learning with triplet-based embedding network. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[38] Kaizhi Qian, Yang Zhang, Shiyu Chang, Mark Hasegawa-Johnson, and David Cox. 2020. Unsupervised speech decomposition via triple information bottleneck. In *International Conference on Machine Learning*. PMLR, 7836–7846.

[39] Ali Razavi, Aäron van den Oord, Ben Poole, and Oriol Vinyals. 2019. Preventing Posterior Collapse with delta-VAEs. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[40] Yi Ren, Jinzheng He, Xu Tan, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2020. Popmag: Pop music accompaniment generation. In *Proceedings of the 28th ACM International Conference on Multimedia*. 1198–1206.

[41] Aaqib Saeed, David Grangier, and Neil Zeghidour. 2021. Contrastive learning of general-purpose audio representations. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 3875–3879.

[42] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.

[43] Joan Serrà, Emilia Gómez, and Perfecto Herrera. 2010. Audio Cover Song Identification and Similarity: Background, Approaches, Evaluation, and Beyond. *Advances in Music Information Retrieval* (2010).

[44] Joan Serra, Emilia Gómez, Perfecto Herrera, and Xavier Serra. 2008. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech, and Language Processing* 16, 6 (2008), 1138–1151.

[45] Joan Serra, Xavier Serra, and Ralph G Andrzejak. 2009. Cross recurrence quantification for cover song identification. *New Journal of Physics* 11, 9 (2009), 093017.

[46] Federico Simonetta, Stavros Ntalampiras, and Federico Avanzini. 2019. Multimodal music information processing and retrieval: Survey and future challenges. In *2019 international workshop on multilayer music representation and processing (MMRP)*. IEEE, 10–18.

[47] Janne Spijkervet and John Ashley Burgoyne. 2021. Contrastive learning of musical representations. *arXiv preprint arXiv:2103.09410* (2021).

[48] Aäron Van Den Oord, Sander Dieleman, and Benjamin Schrauwen. 2014. Transfer learning by supervised pre-training for audio-based music classification. In *Conference of the International Society for Music Information Retrieval (ISMIR 2014)*.

[49] Disong Wang, Liqun Deng, Yu Ting Yeung, Xiao Chen, Xunying Liu, and Helen Meng. 2021. Vqmivc: Vector quantization and mutual information-based unsupervised speech representation disentanglement for one-shot voice conversion. (2021).

[50] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising implicit feedback for recommendation. In *WSDM*. 373–381.

[51] Wenjie Wang, Fuli Feng, Xiangnan He, Xiang Wang, and Tat-Seng Chua. 2021. Deconfounded recommendation for alleviating bias amplification. In *KDD*. 1717–1725.

[52] Wenjie Wang, Fuli Feng, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. 2021. Clicks can be cheating: Counterfactual recommendation for mitigating clickbait issue. In *SIGIR*. 1288–1297.

[53] Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, Min Lin, and Tat-Seng Chua. 2022. Causal Representation Learning for Out-of-Distribution Recommendation. In *WWW*. 3562–3571.

[54] Guoqiang Wei, Cuiling Lan, Wenjun Zeng, Zhizheng Zhang, and Zhibo Chen. 2021. ToAlign: Task-oriented Alignment for Unsupervised Domain Adaptation. *Advances in Neural Information Processing Systems* 34 (2021), 13834–13846.

[55] Ho-Hsiang Wu, Chieh-Chi Kao, Qingming Tang, Ming Sun, Brian McFee, Juan Pablo Bello, and Chao Wang. 2021. Multi-task self-supervised pre-training for music classification. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 556–560.

[56] Xiaoshuo Xu, Xiaoou Chen, and Deshun Yang. 2018. Key-invariant convolutional neural network toward efficient cover song identification. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 1–6.

[57] Dong Yao, Zhou Zhao, Shengyu Zhang, Jieming Zhu, Yudong Zhu, Rui Zhang, and Xiuqiang He. 2022. Contrastive Learning with Positive-Negative Frame Mask for Music Representation. In *Proceedings of the ACM Web Conference 2022*. 2906–2915.

[58] Furkan Yesiler, Joan Serrà, and Emilia Gómez. 2020. Accurate and scalable version identification using musically-motivated embeddings. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 21–25.

[59] Aoxiong Yin, Zhou Zhao, Weike Jin, Meng Zhang, Xingshan Zeng, and Xiaofei He. 2022. MLSLT: Towards Multilingual Sign Language Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5109–5119.

[60] Aoxiong Yin, Zhou Zhao, Jinglin Liu, Weike Jin, Meng Zhang, Xingshan Zeng, and Xiaofei He. 2021. Simulslt: End-to-end simultaneous sign language translation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 4118–4127.

[61] Yi Yu, Suhua Tang, Francisco Raposo, and Lei Chen. 2019. Deep cross-modal correlation learning for audio and lyrics in music retrieval. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 15, 1 (2019), 1–16.

[62] Zhesong Yu, Xiaoshuo Xu, Xiaoou Chen, and Deshun Yang. 2019. Temporal Pyramid Pooling Convolutional Neural Network for Cover Song Identification.. In *IJCAI*. 4846–4852.

[63] Zhesong Yu, Xiaoshuo Xu, Xiaoou Chen, and Deshun Yang. 2020. Learning a representation for cover song identification using convolutional neural network. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 541–545.

[64] Siyang Yuan, Pengyu Cheng, Ruiyi Zhang, Weituo Hao, Zhe Gan, and Lawrence Carin. 2021. Improving Zero-Shot Voice Style Transfer via Disentangled Representation Learning. In *ICLR*.

[65] Lichao Zhang, Yi Ren, Liqun Deng, and Zhou Zhao. 2022. Hifidenoise: High-fidelity denoising text to speech with adversarial networks. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 7232–7236.

[66] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2921–2929.

[67] Hongyuan Zhu, Ye Niu, Di Fu, and Hao Wang. 2021. MusicBERT: A Self-supervised Learning of Music Representation. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3955–3963.