THE POWER OF DEEP WITHOUT GOING DEEP? A STUDY OF HDPGMM MUSIC REPRESENTATION LEARNING

First Author
Affiliation1
author1@ismir.edu

Second Author
Retain these fake authors in submission to preserve the formatting

Third Author
Affiliation3
author3@ismir.edu

ABSTRACT

In the previous decade, Deep Learning (DL) has proven to be one of the most effective machine learning methods to tackle a wide range of Music Information Retrieval (MIR) tasks. It offers the highly expressive learning capacity that can fit any music representations in need for the downstream tasks, which sacrifices interpretability. On the other hand, the Bayesian nonparametric approach promises similar properties, such as the high flexibility while being robust to the overfitting and preserving the interpretability. The primary motivation of this work is to explore the potential of Bayesian nonparametric models in MIR tasks by focusing on the representation learning aspect especially compared to the DL models. We assess the music representation learned from the Hierarchical Dirichlet Process Gaussian Mixture Model (HDPGMM), an infinite mixture model based on the Bayesian nonparametric approach, to MIR tasks, including classifications, auto-tagging, and recommendation. The experimental result suggests that the HDPGMM music representation can outperform DL representations in certain scenarios, and overall comparable.

1. INTRODUCTION

Deep Learning (DL) became one of the most popular and successful methodologies to tackle a various range of Music Information Retrieval (MIR) tasks; music classification [1], music generation [2], recommendation [3] and more. Some of the known benefits are: 1) the high flexibility towards any data structure and expressiveness, 2) effective ways to handling the overfitting at the same time, and finally 3) the rapidly improving infrastructural supports, including both hardwares (i.e., accelerators such as GPU) and softwares (i.e., deep learning frameworks).

Fuelled by these benefits, DL models can learn useful representation of the diverse data structures, which is one of the key reasons of its success [4]. At its cost, on the other hand, DL models often are criticized by its opacity and the lack of interpretability [5], which also holds for the applications within MIR domain [6, 7].

© F. Author, S. Author, and T. Author. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** F. Author, S. Author, and T. Author, "The power of deep without going deep? A study of HDPGMM music representation learning", in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.,* Bengaluru, India, 2022.

The Bayesian nonparametric approach is an alternative approach to achieve what DL does well in an interpretable way. As a non-parametric method, it can overcome the underfitting by unlimited model capacity, while resisting against the overfitting with its Bayesian nature [8]. At the same time, the model can be interpretable through its probabilistic model structure itself, as the modelling procedure is to state, or infer the data generation process in the first place.

Bayesian nonparametric models have been applied in various MIR tasks for previous decades. As latent representation of music audio, it was applied to esitimate music (self) similarity [9, 10]. It also was employed to conduct harmonic and spectral analyses by decomposing the time-frequency representation of the music audio into the mixture of infinite latent components [11, 12]. These analyses also have shown to be useful for downstream tasks such as the structural analysis [13] or music recommentation [14]. Further, discriminative model based on the Bayesian nonparametric is developed for the music emotion recognition [15].

In spite of the similar premises they propose, however, Bayesian nonparametric models have not yet gained as much attention as DL recieved in MIR field. To our best knowledge, in particular, they have not been compared under similar experimental control within MIR context. As they promise quite similar high-level properties, it would be worth to explore to what extent they compare.

The research goal of this work is to assess the music representation learned from Bayesian nonparametric models by comparing them to the modern deep neural network models. In particular, we consider the Hierarchical Dirichlet Process Gaussian Mixture Model (HDPGMM) [14, 16, 17] as our model of interest. The comparison can confirm the potential of Bayesian nonparametric models as a representation learner over the DL methods. To concretize the comparison, we employ the transfer learning experimental framework [18–20], which is commonly used for assessing the potential of representation itself. The contribution of this work can be listed as follows:

1. We explore and suggest insight how "good" the music representation learned by Bayesian nonparametric models is under a range MIR tasks.

2. We provide an efficient, GPU accelerated inference algorithm for HDPGMM, which can handle relatively scaled data.

The remaining of the paper will present the HDPGMM (Section 2), and discuss the experimental setup (Section 3), followed by its result and discussion (Section 4). Lastly, we conclude the paper with a few pointers for the future works (Section 5).

2. HDPGMM

In this subsection, we introduce the main model we employ for this study, the Hiararchical Dirichlet Process Gaussian Mixture Model (HDPGMM) [10, 17]. We first discuss the Dirichlet Process Mixture Model (DPMM), the model on which HDPGMM is extended.

2.1 Dirichlet Process Mixture Models

Mixture models such as GMMs assume a finite number of components from which each of observed feature vectors are drawn. It is well known that finding the optimal number K of the mixuture components is a difficult problem. There are a few approaches that can be useful to estimate K such as using the cross-validation. Dirichlet Process Mixture Model (DPMM) circumbents this problem by parametrizing the number of mixtures as part of the model. DP plays a central role for such models.

DP is a stochastic process which draws random probability distributions. Due to this property, it is often described as a distribution over distributions [8]. It can be also seen as the infinite dimensional generalization of the Dirichlet distributions [8]. This aspect is the core building block of the Bayesian non-parametric models such as infinite mixture models. Setting DP as a prior distribution for the responsibility π (also often referred as mixture probability) of mixture components allows the mixture model to infer both the relative weight among components and the appropriate number of maximum components for given observations.

Among a several ways to represent DP, we introduce the stick-breaking construction [21]. [1] Stick-breaking constructs DP in a simple and general manner. Formally, it is as follows:

$$\begin{aligned} \beta'_k &\sim \text{Beta}(1, \gamma) & \phi_k &\sim H \\ \beta_k &= \beta'_k \prod_{l=1}^{k-1}(1-\beta'_l) & G_0 &= \sum_{k=1}^{\infty} \beta_k \delta_{\phi_k} \end{aligned} \tag{1}$$

where two euqations in the left column represent the draw of infinite dimensional weight β_k which sums to one. Notably, the distribution for β is also referred as $\beta \sim \text{GEM}(\gamma)$ [22]. In the right column, H denotes the base distribution from which variable ϕ_k lying in some space Φ is drawn. The right bottom equation defines the draw of the probability measure G_0, where δ_{ϕ_k} means the point mass centered at the component ϕ_k. Altogether, Eq. 1 constructs the DP $G_0 \sim \text{DP}(\gamma, H)$. Figure 1 depicts the process in graphical way. In mixture model context, we want to infer

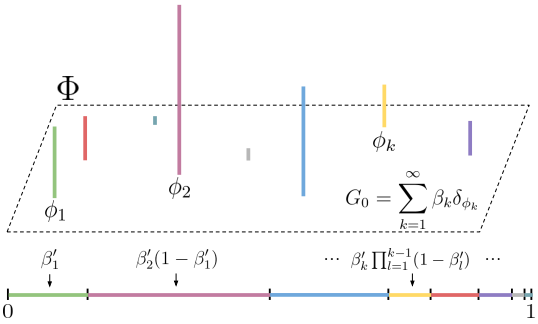


Figure 1: Illustration of stick-breaking construction in Eq. 1

mixture components $\{\phi_1, \phi_2, \cdots, \phi_k, \cdots\}$ that fits to the data observations $\{x_1, x_2, \cdots, x_n\}$, which are assumed to be drawn from distributions $F(\phi)$ parameterized with variable ϕ (i.e., mean and covariance $\phi = \{\mu, \Sigma\}$ in case of the multivariate Gaussian F). we can now use DP to draw ϕ as ith mixture components corresponds to the ith observation x_i by introducing the cluster assignment variable $y_i \sim \text{Mult}(\beta)$:

$$\begin{aligned} \beta|\gamma &\sim \text{GEM}(\gamma) & \phi_k|H &\sim H \\ y_i|\beta &\sim \text{Mult}(\beta) & x_i|y_i, \{\phi_k\} &\sim F(\phi_{y_i}) \end{aligned} \tag{2}$$

where ϕ_{y_i} denotes the component parameter ϕ indexed by assignment variable y_i corresponding to the ith observation x_i.

2.2 Hierarchical DPMM

In many data structure, groupings of atomic data points arise naturally (i.e., audio frames within a song, songs from an artist, words of a lyrics). Hierarchical DP (HDP) is an extention of DP modelling "groupings" by imposing group-level DPs derived from the "global-level" or "corpus-level" DP as the global pool of components [17]. Following Sethuraman's stick-breaking construction [16], jth group-level DP can be expressed as follows:

$$\begin{aligned} \pi'_{jt} &\sim \text{Beta}(1, \alpha_0) & \psi_{jt} &\sim G_0 \\ \pi_{jt} &= \pi'_{jt} \prod_{l=1}^{t-1}(1-\pi'_{jl}) & G_j &= \sum_{t=1}^{\infty} \pi_{jt} \delta_{\psi_{jt}} \end{aligned} \tag{3}$$

As seen above, HDP indeed appears as the recursion of multiple levels of DPs [2] . Notably, the base distribution G_0 of each group-level DP is from the corpus-level DP. This relationship allows to map group-level atoms ψ_{jt} to the corpus-level atoms ϕ_k. Wang et al. introduce a series of indicator variables c_{jt} which maps ψ_{jt} and ϕ_k as follows [16]:

$$\begin{aligned} c_{jt} &\sim \text{Mult}(\beta) \\ \psi_{jt} &= \phi_{c_{jt}} \end{aligned} \tag{4}$$

[1] Literature commonly chooses the Chinese Restaurant Process (CRP) for a illustrative metaphor for DP as it is intuitive and well explains various properties of DP [8]. We mainly discuss DP with the stick-breaking construction due to its further usage in the model inference within the work. For readers interested in other metaphorical descriptions of DP, we kindly refer them to [8, 10]

[2] It implies naturally that multiple levels are possible (i.e., corpus - author - document), if it suits to the data structure.

where β is drawn from the corpus-level DP in Eq. 1. It simplifies the model as we do not need to explicitly represent ψ_{jt} [16]. Finally, we can represent HDPMM by introducing another indicator variable $z_{jn} \sim \text{Mult}(\pi_j)$ for nth observation x_{jn} within the jth group, similarly to Eq. 2:

$$\begin{aligned} \pi_j|\alpha_0 &\sim \text{GEM}(\alpha_0) & \theta_{jn} &= \psi_{jz_{jn}} = \phi_{c_{jz_{jn}}} \\ z_{jn}|\pi_j &\sim \text{Mult}(\pi_j) & x_{jn}|z_{jn}, c_{jt}, \{\phi_k\} &\sim F(\theta_{jn}) \end{aligned} \tag{5}$$

where we use the indicator z_{jn} to select ψ_{jt}, which eventually is mapped as $\phi_{c_{jz_{jn}}}$ that represents the parameter θ_{jn} to draw the observation x_{jn}. HDPGMM is then defined by simply setting F as the (multivariate) Gaussian distribution and H as one of distribution from which we can sample the mean and covariance (i.e., Gaussian-inverse Wishart distribution). Figure 2 depicts the HDPGMM graphically.

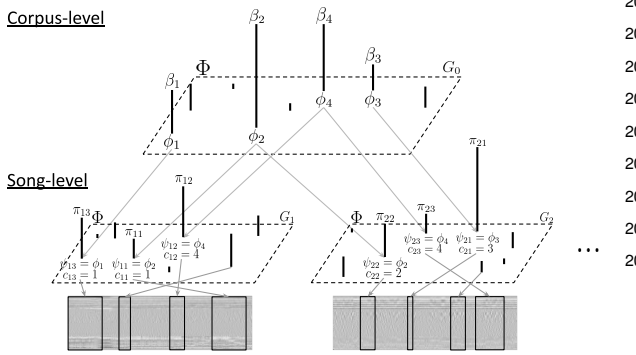


Figure 2: Illustration of 2-level HDPMM within corpus-song context. The top illustration depicts the corpus-level DP similar to Figure 1. The second row illustrations describe the draw of second level (song-level) DPs per song from the corpus-level DP. DPMM assumes that song features are drawn from each song-level DPs as depicted in the third rows of the image.

2.3 Inference Algorithm

In this section we discuss the inference (training) algorithm. We employ the online Variational Inference (VI) [16]. VI is one of the common choices to infer a fully Bayesian models and usually significantly faster than other methods such as Markov Chain Monte Carlo (MCMC) with the expense of its relative precision. VI seeks the simpler, approximated version of the true posterior by minimizing the Kullback-Leibler (KL) divergence between approximation $q(Z)$ and the true posterior $p(Z|X)$, where Z denote the set of latent variables and parameters that we want to find, and X refers a set of observations [23]. One of the popular simpliciation is full-factorization of the distribution $q(Z) = \prod_{i=1}^{|Z|} q_i(Z_i)$. In the context of HDPGMM, we have the following factorization:

$$q(\beta', \pi', c, z, \phi) = q(\beta')q(\pi')q(c)q(z)q(\phi) \tag{6}$$

where β', π', c, z denote the corpus-level and group-level stick proportion, group-level component selection variable, and finally the observation-level component selection variable, respectively. ϕ refers the parameter(s) for the F which draws the atomic observation, which is set as (multivariate) Gaussian in our context. Thus, ϕ includes the means μ and precision matrices Λ of each Gaussian component [3].

Each variational distributions further factorize as follows:

$$\begin{aligned} q(\beta') &= \textstyle\prod_k^{K-1} \text{Beta}(\beta'_k|u_k, v_k) \\ q(\pi') &= \textstyle\prod_t^{T-1} \text{Beta}(\pi'_t|a_t, b_t) \\ q(c) &= \textstyle\prod_j \prod_t \text{Mult}(c_{jt}|\varphi_{jt}) \\ q(z) &= \textstyle\prod_j \prod_n \text{Mult}(z_{jn}|\zeta_{jn}) \end{aligned} \tag{7}$$

where u_k, v_k, a_t, b_t denote the variational parameters for the Beta distributions for corpus-level and group-level stick proportions, respectively. $\varphi_{jt} \in \mathbb{R}^K, \zeta_{jn} \in \mathbb{R}^T$ are the variational parameters for the Multinomial distribution to draw the selector c and z.

Notably, we truncate the inifinite Beta distributions by K and T, which is a common method for applying VI on the infinite mixture model. With sufficiently large number for the truncation, the model will still not be limited to the truncation, and will only use the number of components that suits for given dataset. Final variational distribution is the Gaussian-Wishart prior distribution which draws the Gaussian parameters $\phi = \{\mu, \Lambda\}$ for distribution F:

$$q(\phi) = \textstyle\prod_k^K \mathcal{N}(\mu_k|m_k, (\lambda_k\Lambda_k)^{-1})\mathcal{W}(\Lambda_k|W_k, \nu_k) \tag{8}$$

where we draw the precision $\Lambda_k \in \mathbb{R}^{d\times d}$ from Wishart distribution with the variational parameter $W_k \in \mathbb{R}^{d\times d}$ and $\nu_k \in \mathbb{R}$, and the mean $\mu_k \in \mathbb{R}^d$ is drawn by the precision weighted by $\lambda_k \in \mathbb{R}$ and mean $m_k \in \mathbb{R}^d$.

We then can obtain the optimal model by maximizing the lowerbound of the marginal log likelihood $\log p(X|Z)$ [23–25]:

$$\begin{aligned} \log p(X|Z) &\geq \mathbb{E}_q[\log p(X, \beta', \pi', c, z, \phi)] + H(q) \\ &= \textstyle\sum_j \big\{\mathbb{E}_q[\log\left(p(X_j|c_j, z_j, \phi)p(c_j|\beta')p(z_j|\pi'_j)p(\pi'_j|\alpha_0)\right)] \\ &\quad + H(q(c_j)) + H(q(z_j)) + H(q(\pi'_j))\big\} \\ &\quad + \mathbb{E}_q[\log p(\beta')p(\phi)] + H(q(\beta')) + H(q(\phi)) \end{aligned} \tag{9}$$

where $H(\cdot)$ denotes the entropy of given distribution, and $X_j = \{x_{j1}, x_{j2}, \cdots, x_{jN_j}\}$ is a set of observations within jth group. [4]

Using the standard result of VI [16, 23, 25], the update rules for group-level parameters are dreived as follows:

$$a_{jt} = 1 + \textstyle\sum_n \zeta_{jnt} \tag{10}$$

$$b_{jt} = \alpha_0 + \textstyle\sum_n \sum_{s=t+1}^T \zeta_{jnt} \tag{11}$$

$$\varphi_{jtk} \propto \exp(\textstyle\sum_n \zeta_{jnt}\mathbb{E}_q[\log p(x_{jn}|\phi_k)] + \mathbb{E}_q[\log \beta_k]) \tag{12}$$

$$\zeta_{jnt} \propto \exp(\textstyle\sum_{k=1}^K \varphi_{jtk}\mathbb{E}_q[\log p(x_{jn}|\phi_k)] + \mathbb{E}_q[\log \pi_{jt}]) \tag{13}$$

[3] We adopt the result from [23], where the Gaussian-Wishart distribution is used for the prior.

[4] Thus, the terms inside the sum over the group would further factorized into sums of these per-group observation. We omit them to avoid the equation being too crammed.

Similarly, the update rules for the corpus-level parameters are as follows [23]:

$$u_k = 1 + \sum_j \sum_{t=1}^{T} \varphi_{jtk} \tag{14}$$

$$v_k = \gamma + \sum_j \sum_{t}^{T} \sum_{l=k+1}^{K} \varphi_{jtl} \tag{15}$$

$$\lambda_k = \lambda_0 + N_k \tag{16}$$

$$m_k = \lambda_k^{-1}(\lambda_0 m_0 + N_k \bar{x}_k) \tag{17}$$

$$W_k^{-1} = W_0^{-1} + N_k S_k + \frac{\lambda_0 N_k}{\lambda_0 + N_k}(\bar{x}_k - m_0)(\bar{x}_k - m_0)^\intercal \tag{18}$$

$$\nu_k = \nu_0 + N_k \tag{19}$$

where $\lambda_0 \in \mathbb{R}, m_0 \in \mathbb{R}^d, \nu_0 \in \mathbb{R}, W_0 \in \mathbb{R}^{d \times d}$ are the hyperparameters corresponding to the weight, location, degrees of freedom, and scale of Gaussian-Wishart distribution. The sufficient statistics and expectations used above update rules are defined as follows:

$$\mathbb{E}_q[\log \beta_k] = \mathbb{E}_q[\log \beta'_k] + \sum_{l=1}^{k-1} \mathbb{E}_q[\log(1-\beta'_l)]$$
$$\mathbb{E}_q[\log \beta'_k] = \Psi(u_k) - \Psi(u_k + v_k)$$
$$\mathbb{E}_q[\log(1-\beta'_k)] = \Psi(v_k) - \Psi(u_k + v_k)$$
$$\mathbb{E}_q[\log \pi_{jt}] = \mathbb{E}_q[\log \pi'_{jt}] + \sum_{s=1}^{t-1} \mathbb{E}_q[\log(1-\pi'_s)]$$
$$\mathbb{E}_q[\log \pi'_{jt}] = \Psi(a_{jt}) - \Psi(a_{jt} + b_{jt})$$
$$\mathbb{E}_q[\log(1-\pi'_{jt})] = \Psi(b_{jt}) - \Psi(a_{jt} + b_{jt})$$
$$N_k = \sum_j \sum_n r_{jnk}$$
$$\bar{x}_k = \frac{1}{N_k} \sum_j \sum_n r_{jnk} x_{jn}$$
$$S_k = \frac{1}{N_k} \sum_j \sum_n r_{jnk}(x_{jn} - \bar{x}_k)(x_{jn} - \bar{x}_k)^\intercal$$

where $\Psi(\cdot)$ refers the digamma function, and $r_{jnk} = \sum_{t=1}^{T} \zeta_{jnt}\varphi_{jtk}$ is the inferred responsibility of nth observation of jth group on kth component. Standard batch update would compute statistics across the entire corpus, and updates the corpus level parameters. However, it may be slow or may suffer by too large or small numbers accumulated from a large scale corpus.

In this work, we employ the online VI where corpus-level parameters are updated per every group or mini-batch of group passed. In this way, the early phase of model inference can be accelerated substantially compared to the full-batch update [16, 26]. The corpus level update then contoled by the learning rate $\rho_t = (\tau_0 + t)^{-\kappa}$, which is decayed over iterations controlled by the offset parameter $\tau_0 > 0$ and scale $\kappa \in (0.5, 1]$:

$$Z_t = (1 - \rho_t) Z_{t-1} + \rho_t \tilde{Z}_t \tag{20}$$

where $\tilde{Z}_t$ means one of the corpus-level parameter in Eq. (14) to (19) updated by the given mini-batch at tth iteration, while Z_{t-1} is the current parameter. To properly scale the update with respect to the mini-batch size, we weight them by the factor of $w = \frac{|\tilde{X}|}{|X|}$, where $|X|, |\tilde{X}|$ denote the number of groups within the entire observation set X and the mini-batch of groups $\tilde{X}$.

Combining altogether, the overall inference algorithm is described in Algorithm 1.

Algorithm 1: Online VI for HDPGMM

Initialize $\phi = (\phi_k)_{k=1}^K, u = (u_k)_{k=1}^K, v = (v_k)_{k=1}^K$ randomly. Set $t = 1$.
while *Stopping critrion is not met* **do**
  Fetch a random mini-batch of groups $\tilde{X}$
  repeat
    Update a_j, b_j, ζ_jandφ_j using Eq. (10) to (13)
  until *mini-batch likelihood stops improving*;
  Compute $u_k, v_k, \lambda_k, m_k, W_k,$ andν_k using Eq. (14) to (19)
  Set $\rho_t = (\tau_0 + t)^{-\kappa}, t \leftarrow t + 1$
  Update $u_k, v_k, \lambda_k, m_k, W_k,$ andν_k using Eq. 20
end

2.4 Further Regularization

Inspired by the implementation of [16, 25], we apply the further regularization on the model. Specifically, we "splash" the uniform noise to the inferred responsibility r_{jn} as it can be biased if the groups are corrupted or impcomplete, such as the preview audio of the entire song:

$$\tilde{r}_{jn} = (1 - \eta_t) r_{jn} + \eta_t e \tag{21}$$

η_t is the regularization coefficient that determines the extent uniform noise $e = (e_k)_{k=1}^K$ is mixed into r_{jn}. $\eta_t = \frac{\eta_0 * 10^3}{(t+10^3)}$ also is defined as decaying function similar to the learning rate ρ_t.

3. EXPERIMENTAL SETUP

In this section, we discuss the details of the empirical experiment we conduct. The overall design is adopted from the recent music representation learning studies [18, 19, 27], where multiple downstream music machine learning tasks are tested with the feature set learned from the representation models.

3.1 Datasets, Features & Evaluation

We used the subset of Million Song Dataset (**MSD**) [28] as the "source" dataset for the representation learning (i.e., inference for HDPGMM). Specifically, we used the preview music excerpts from the "train" subset of MSD introduced by [29, 30] [5] . For the evaluation of the representation, we employ three target (downstream) tasks encompassing *music genre classification*, *music auto tagging*, *music recommendation*. For each target task, we chose the **GTZAN** dataset [31] with the fault-filtered split [32], MagnaTagaTune (**MTAT**) dataset [33] with the split from [34], and finally the Echo Nest taste profile subset as part of MSD (**Echonest**) [28] filtered by user and item frequency [35] for the genre classification, auto-tagging and recommendation, respectively.

[5] Length of preview audio differs per clips, where about 70% of samples are approximately 30 seconds and rest are 1 minute, with a small subset is longer than 1 minute.

The evaluation is done by the cross-validation of task-specific prediction models. We fit the logistic regression models for GTZAN and MTAT dataset which takes learned music represetnation as input and predicts the genre or music tags respectively. We find the hyper-parameters of logistic regression model by the randomized parameter search [36] for each run.

As for the recommendation, we apply the item *K* Nearest Neighbor (*item-kNN*) method [37] where the song-song similarity is measured by the *cosine simiarlity* of the learned music representation. We adopted the data split introduced in [35]. [6] We repeat 5 "runs" for each of downstream evalution to take account the various random effect (i.e., initialization, randomized search). The accuracy performance of each target task is assessed by commonly used accuracy measure in each task, which are further elaborated in Table 1.

Id	# Samples	# Classes	Acc. Measure
MSD	$213,354$	N/A	N/A
GTZAN	$1,000$	10	F1 [38]
MTAT	$25,863$	50	AUROC [39]
Echonest	$40,980$	$571,355$ [7]	nDCG [40]

Table 1: Details on dataset used. We use *macro* average strategy for F1 and Area Under Receiver Operating Characteristic curve (AUROC) measure, and we consider top 500 recommendationed songs for computing the normalized Discounted Cumulative Gain (nDCG).

We select a set of audio features to infer the HDPGMM and non-DL baseline models inspired by [15]. The detailed curation of the features can be found in Table 2 [8].

Feature	Aspect	Dim.	Transform
MFCC	Timbre	13	-
ΔMFCC	Timbre	13	-
$\Delta\Delta$MFCC	Timbre	13	-
Onset Strength [42]	Rhythm	1	log
Chroma [43]	Tonal	12	log

Table 2: Details on the base audio features we employed for the experiment.

3.2 Comparisons

We evaluate 4 comparisons against HDPGMM, which includes both DL based models and non-DL based models.

- **G1** is the simplist model among the comparison. The song-wise feature is represented by the concatenated parameters $\phi = \{\mu, \Sigma\}$ of the single multivariate Gaussian fitted on the frame-level base audio feature vectors within a song [9].
- **VQ Codebook** can be deemed as the simple approximation of HDPGMM models [10]. First a corpus-wise K means clustering model is fitted on the song features in the frame-level. Later song-level feature is represented by the normalized frequency of cluster assignment of each frame-level base audio feature vector within the song. It can be interpreted similarly to the inferred cluster assignment variable π_j in Eq. (5). We set the number of cluster as 256, same to the maximum number of corpus-level components K of HDPGMM models.
- **KIM** is a DL based music representation trained in a simple self-supervised learning framework [18]. The objective function of the model is to measure the extent which a siamenese neural network [44] can predict whether the two song clips it takes are cropped from the same song or different songs. We use the original implementation of the work, which takes the stereo mel-spectrogram as input. The representation is extracted from the last hidden layer before the prediction, and summarized with global average and standard devication pooling, through the entire song excerpt.
- **CLMR** [27] is recent DL based music representation employing the self-supervised learning through the contrastive learning method [45]. We employed the original implementation of [27] that uses the raw audio samples as input feature [10]. The representation is drawn from the last hidden layer and pooled with global average over the entire song excerpt.
- **HDPGMM** [11] uses the base audio feature with the whitening following [47]. As for song-level representation y_j, we employ the expectation $\tilde{y}_{jk} = \mathbb{E}_q[\log p(X_j|c_j, z_j, \phi_k)]$ and normalize them over components and the length of the clip $y_{jk} = N_j^{-1} \frac{\tilde{y}_{jk}}{\sum_{k'=1}^{K} \tilde{y}_{jk'}}$. We adopt most of the hyper-parameter setup found from previous work [16]; we set the maximum number of components on the corpus level K and the song level T as 256 and 32 respectively, mini-batch size to 1024, learning rate parameters $\kappa = 0.6$ and $\tau_0 = 64$. We, however, explore the regularization rate $\eta_0 \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$ and total corpus size $|X| \in \{2 \times 10^3, 2 \times 10^4, 213354\}$ to examine their effect on the representation.

For the learning based models (i.e., KIM, CLMR,

[6] For every trial, disjoint validation and test users are uniformly sampled, which includes 10% of population for each. Then 30% of listening records of these users are hold out as testing interactions. We find the best *K* by conducting the grid search on the range $K \in \{2^4, 2^5, 2^6, 2^7, 2^8, 2^9\}$ by measuring the accuracy on the validation users. Using the best *K*, we compute final score on the testing user, holding out the same amount of the interaction. For each run, We take the average score over 5 trials to incorperate the random split effect.

[7] It refers the number of users in this dataset.

[8] We use the implementation of `librosa` [41] with the default parameters for all the features.

[9] We chose the diagonalized covariance, which means the feature is concatenation of *mean* $\mu \in \mathcal{R}^{52}$ and *standard deviation* $\sigma \in \mathcal{R}^{52}$ of features.

[10] Unlike the original work, we set the dimensionality of hidden layer where we extract the representation as 256 to control the representation dimensionality with other comparisons. Also, we do not apply the data augmentation for a fair comparison and also as it is beyond the main scope of the work. However, as it is reported that it can meaningfully improve the representation, we assume that applying the method would benefit similarly to other models as well.

[11] inference algorithm in algorithm 1 is implemented on `PyTorch` [46], and thus support GPU acceleration.

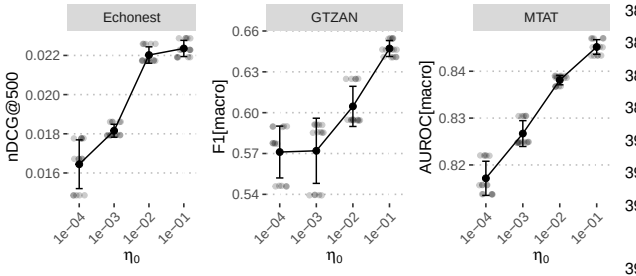
HDPGMM), we repeated 3 trials to incorporate any random effect of learning algorithms. Within a trial, we iterated the update for 100 epochs. Further, we set the dimensionality of the representation is set to 256, homogeneous accross all comparisons except $G1$. Finally, we take the logarithm of the representation $\hat{y_j} = \log(\max(y_j, 10^{-8}))$ if it is given as the probability simplex (i.e., VQCodebook, HDPGMM).

4. RESULT AND DISCUSSION

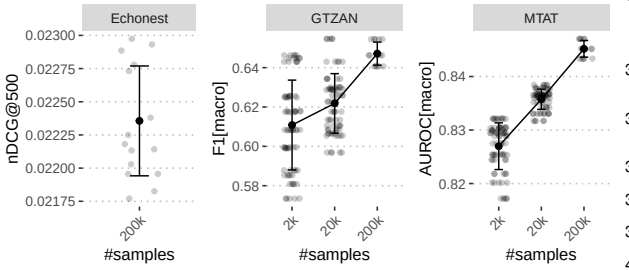
4.1 Effect of learning factors

Overall, we observe that the learning factors of HDPGMM can make a substantial difference. As Figure 3 shows, the result suggests that the additional regularization in Eq (21) generally effect positvely on the performance of all the downstream tasks we tested. It implies that the entire song inputs would likely improve the representation further, as the regularization crudely mask the effect of the missing data due to the preview clipping to some extent.

Further, we found that the number of training samples also have a rather clear positive effect on the performances. We measured the effect by repeately sub-sampling the MSD training data 5 times in two different sizes mentioned in 3.2, and conduct the same evalution routine as the full training set. The result shows that the positive effect is logarithmic, which suggest that to get a linear improvement in the performance, the representation should be learned with exponentially larger dataset.



(a) Effect of regularization factor η_0. ($|X| = 213,354$)



(b) Effect of the number of training samples. ($\eta_0 = 10^{-1}$)

Figure 3: Effect of the learning factors on HDPGMM. Error bar indicates the standard deviation, and grey dots are the raw datapoints.

4.2 Model Comparison

Model comparison result suggests that the HDPGMM model is comparable to the DL representation on average, and can be outperforming in a certain scenario. Reported in Table 3, KIM achieves worse performance on GTZAN and MTAT than HDPGMM, while better at recommendation (Echonest). On the other hand, CLMR, which adopts a more sophisticated learning strategies, performed better in two tasks (GTZAN, Echonest) but still worse in the auto-tagging. It seems, however, Hierarchical mixture model variants (VQCodebook, HDPGMM) performs particularly worse in the recommendation task. We hypothesize that the cosine similarity might not be an optimal choice for the probability simplex representation, which however requires the further study.

Dataset	Model	Mean Acc.(±SD)
Echonest	G1	0.0252 (± 0.0000)
	VQCodebook	0.0158 (± 0.0000)
	KIM	0.0262 (± 0.0012)
	CLMR	**0.0279 (± 0.0000)**
	HDPGMM	0.0224 (± 0.0004)
GTZAN	G1	0.5409 (± 0.0000)
	VQCodebook	0.5776 (± 0.0000)
	KIM	0.5448 (± 0.0245)
	CLMR	**0.6681 (± 0.0000)**
	HDPGMM	0.6471 (± 0.0059)
MTAT	G1	0.8435 (± 0.0000)
	VQCodebook	0.8373 (± 0.0000)
	KIM	0.8021 (± 0.0041)
	CLMR	0.8248 (± 0.0000)
	HDPGMM	**0.8452 (± 0.0016)**

Table 3: Evaluation result on downstream tasks.

Except this corner case, the HDPGMM representation outperforms the other non-DL comparisons in general. Especially, it achieves significant improvement over the simple version of it (VQCodebook) for all tasks, and mostly outperforming G1.

4.3 Component Interpretability

Finally, we

5. CONCLUSION AND FUTURE WORKS

6. REFERENCES

[1] M. Won, J. Spijkervet, and K. Choi, *Music Classification: Beyond Supervised Learning, Towards Real-world Applications*. https://music-classification.github.io/tutorial, 2021. [Online]. Available: https://music-classification.github.io/tutorial

[2] J. Briot, G. Hadjeres, and F. Pachet, *Deep Learning Techniques for Music Generation*, ser. Computational Synthesis and Creative Systems. Springer International Publishing, 2019.

[3] M. Schedl, "Deep learning in music recommendation systems," *Frontiers in Applied Mathematics and Statistics*, vol. 5, 2019. [Online]. Available: https://www.frontiersin.org/article/10.3389/fams.2019.00044

Hip-Hop	country	female vocalists	pop	electronic	pop	oldies
pop	rock	singer-songwriter	female vocalists	dance	soul	blues
rnb	pop	pop	female vocalist	electronica	female vocalists	country
soul	oldies	acoustic	rock	funk	rnb	60s
male vocalists	indie	Mellow	Love	electro	dance	soul
rock	singer-songwriter	folk	dance	Hip-Hop	Hip-Hop	rock
hip hop	classic rock	female vocalist	female	House	rock	classic rock
0.0394	0.0308	0.0299	0.0269	0.0268	0.0248	0.0227

Table 4: Top tags for a few mostly "loaded" components (column-wise). The last row is the normalized total responsibility $\tilde{N}_k = \frac{N_k}{\sum_{k'}^{K} N_{k'}}$, meaning the proportional amount of songs having the component.

[4] E. J. Humphrey, J. P. Bello, and Y. LeCun, "Moving beyond feature design: Deep architectures and automatic feature learning in music informatics," in *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012, Mosteiro S.Bento Da Vitória, Porto, Portugal, October 8-12, 2012*, F. Gouyon, P. Herrera, L. G. Martins, and M. Müller, Eds. FEUP Edições, 2012, pp. 403–408. [Online]. Available: http://ismir2012.ismir.net/event/papers/403-ismir-2012.pdf

[5] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. A. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018, Turin, Italy, October 1-3, 2018*, F. Bonchi, F. J. Provost, T. Eliassi-Rad, W. Wang, C. Cattuto, and R. Ghani, Eds. IEEE, 2018, pp. 80–89. [Online]. Available: https://doi.org/10.1109/DSAA.2018.00018

[6] B. L. Sturm, "A simple method to determine if a music information retrieval system is a "horse"," *IEEE Trans. Multim.*, vol. 16, no. 6, pp. 1636–1644, 2014. [Online]. Available: https://doi.org/10.1109/TMM.2014.2330697

[7] ——, "The "horse" inside: Seeking causes behind the behaviors of music content analysis systems," *Comput. Entertain.*, vol. 14, no. 2, pp. 3:1–3:32, 2016. [Online]. Available: https://doi.org/10.1145/2967507

[8] Y. W. Teh, "Dirichlet process," in *Encyclopedia of Machine Learning and Data Mining*, C. Sammut and G. I. Webb, Eds. Springer, 2017, pp. 361–370. [Online]. Available: https://doi.org/10.1007/978-1-4899-7687-1_219

[9] Y. Qi, J. W. Paisley, and L. Carin, "Dirichlet process HMM mixture models with application to music analysis," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2007, Honolulu, Hawaii, USA, April 15-20, 2007*. IEEE, 2007, pp. 465–468. [Online]. Available: https://doi.org/10.1109/ICASSP.2007.366273

[10] M. D. Hoffman, D. M. Blei, and P. R. Cook, "Content-based musical similarity computation using the hierarchical dirichlet process," in *ISMIR 2008, 9th International Conference on Music Information Retrieval, Drexel University, Philadelphia, PA, USA, September 14-18, 2008*, J. P. Bello, E. Chew, and D. Turnbull, Eds., 2008, pp. 349–354. [Online]. Available: http://ismir2008.ismir.net/papers/ISMIR2008_130.pdf

[11] ——, "Finding latent sources in recorded music with a shift-invariant hdp," in *in International Conference on Digital Audio Effects (DAFx) (under review)*, 2009.

[12] M. Nakano, J. L. Roux, H. Kameoka, N. Ono, and S. Sagayama, "Infinite-state spectrum model for music signal analysis," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2011, May 22-27, 2011, Prague Congress Center, Prague, Czech Republic*. IEEE, 2011, pp. 1972–1975. [Online]. Available: https://doi.org/10.1109/ICASSP.2011.5946896

[13] M. Nakano, Y. Ohishi, H. Kameoka, R. Mukai, and K. Kashino, "Bayesian nonparametric music parser," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2012, Kyoto, Japan, March 25-30, 2012*. IEEE, 2012, pp. 461–464. [Online]. Available: https://doi.org/10.1109/ICASSP.2012.6287916

[14] K. Yoshii and M. Goto, "Continuous plsi and smoothing techniques for hybrid music recommendation," in *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe International Conference Center, Kobe, Japan, October 26-30, 2009*, K. Hirata, G. Tzanetakis, and K. Yoshii, Eds. International Society for Music Information Retrieval, 2009, pp. 339–344. [Online]. Available: http://ismir2009.ismir.net/proceedings/OS4-1.pdf

[15] J. Wang, Y. Lee, Y. Chin, Y. Chen, and W. Hsieh, "Hierarchical dirichlet process mixture model for music emotion recognition," *IEEE Trans. Affect. Comput.*, vol. 6, no. 3, pp. 261–271, 2015. [Online]. Available: https://doi.org/10.1109/TAFFC.2015.2415212

[16] C. Wang, J. W. Paisley, and D. M. Blei, "Online variational inference for the hierarchical dirichlet process," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics,*

AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011, ser. JMLR Proceedings, G. J. Gordon, D. B. Dunson, and M. Dudík, Eds., vol. 15. JMLR.org, 2011, pp. 752–760. [Online]. Available: http://proceedings.mlr.press/v15/wang11a/wang11a.pdf

[17] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei, "Hierarchical dirichlet processes," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1566–1581, 2006. [Online]. Available: https://doi.org/10.1198/016214506000000302

[18] J. Kim, J. Urbano, C. C. S. Liem, and A. Hanjalic, "One deep music representation to rule them all? A comparative analysis of different representation learning strategies," *Neural Comput. Appl.*, vol. 32, no. 4, pp. 1067–1093, 2020. [Online]. Available: https://doi.org/10.1007/s00521-019-04076-1

[19] K. Choi, G. Fazekas, M. B. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, S. J. Cunningham, Z. Duan, X. Hu, and D. Turnbull, Eds., 2017, pp. 141–149. [Online]. Available: https://ismir2017.smcnus.org/wp-content/uploads/2017/10/12_Paper.pdf

[20] S. Dieleman, P. Brakel, and B. Schrauwen, "Audio-based music classification with a pretrained convolutional network," in *Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, October 24-28, 2011*, A. Klapuri and C. Leider, Eds. University of Miami, 2011, pp. 669–674. [Online]. Available: http://ismir2011.ismir.net/papers/PS6-3.pdf

[21] J. Sethuraman, "A constructive definition of Dirichlet priors," *Statistica Sinica*, vol. 4, pp. 639–650, 1994.

[22] J. Pitman, "Poisson-dirichlet and gem invariant distributions for split-and-merge transformations of an interval partition," *Comb. Probab. Comput.*, vol. 11, no. 5, pp. 501–514, 2002. [Online]. Available: https://doi.org/10.1017/S0963548302005163

[23] C. M. Bishop and N. M. Nasrabadi, "*Pattern Recognition and Machine Learning*," *J. Electronic Imaging*, vol. 16, no. 4, p. 049901, 2007. [Online]. Available: https://doi.org/10.1117/1.2819119

[24] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, no. 2, pp. 183–233, 1999. [Online]. Available: https://doi.org/10.1023/A:1007665907178

[25] D. M. Blei and M. I. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian Analysis*, vol. 1, no. 1, pp. 121 – 143, 2006. [Online]. Available: https://doi.org/10.1214/06-BA104

[26] M. D. Hoffman, D. M. Blei, and F. R. Bach, "Online learning for latent dirichlet allocation," in *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada*, J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, Eds. Curran Associates, Inc., 2010, pp. 856–864. [Online]. Available: https://proceedings.neurips.cc/paper/2010/hash/71f6278d140af599e06ad9bf1ba03cb0-Abstract.html

[27] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, J. H. Lee, A. Lerch, Z. Duan, J. Nam, P. Rao, P. van Kranenburg, and A. Srinivasamurthy, Eds., 2021, pp. 673–681. [Online]. Available: https://archives.ismir.net/ismir2021/paper/000084.pdf

[28] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[29] J. Pons, O. Nieto, M. Prockup, E. M. Schmidt, A. F. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, E. Gómez, X. Hu, E. Humphrey, and E. Benetos, Eds., 2018, pp. 637–644. [Online]. Available: http://ismir2018.ircam.fr/doc/pdfs/191_Paper.pdf

[30] J. Lee, J. Park, K. L. Kim, and J. Nam, "Samplecnn: End-to-end deep convolutional neural networks using very small filters for music classification," *Applied Sciences*, vol. 8, no. 1, 2018. [Online]. Available: https://www.mdpi.com/2076-3417/8/1/150

[31] G. Tzanetakis and P. R. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, 2002. [Online]. Available: https://doi.org/10.1109/TSA.2002.800560

[32] C. Kereliuk, B. L. Sturm, and J. Larsen, "Deep learning and music adversaries," *IEEE Trans. Multim.*, vol. 17, no. 11, pp. 2059–2071, 2015. [Online]. Available: https://doi.org/10.1109/TMM.2015.2478068

[33] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, "Evaluation of algorithms using games: The case of music tagging," in *Proceedings of the 10th International Society for Music Information Retrieval Conference, ISMIR 2009, Kobe International Conference Center, Kobe, Japan, October 26-30, 2009*, K. Hirata, G. Tzanetakis, and K. Yoshii, Eds. International Society for Music Information

Retrieval, 2009, pp. 387–392. [Online]. Available: http://ismir2009.ismir.net/proceedings/OS5-5.pdf

[34] J. Lee and J. Nam, "Multi-Level and Multi-Scale Feature Aggregation Using Pretrained Convolutional Neural Networks for Music Auto-Tagging," *IEEE Signal Processing Letters*, vol. 24, no. 8, pp. 1208–1212, Aug. 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7944704/

[35] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, P. Champin, F. Gandon, M. Lalmas, and P. G. Ipeirotis, Eds. ACM, 2018, pp. 689–698. [Online]. Available: https://doi.org/10.1145/3178876.3186150

[36] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, pp. 281–305, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2188395

[37] M. Deshpande and G. Karypis, "Item-based top-N recommendation algorithms," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, 2004. [Online]. Available: https://doi.org/10.1145/963770.963776

[38] C. J. van Rijsbergen, *Information Retrieval*. Butterworth, 1979.

[39] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006. [Online]. Available: https://doi.org/10.1016/j.patrec.2005.10.010

[40] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, 2002. [Online]. Available: http://doi.acm.org/10.1145/582415.582418

[41] B. McFee, A. Metsai, M. McVicar, S. Balke, C. Thomé, C. Raffel, F. Zalkow, A. Malek, Dana, K. Lee, and et al., "librosa/librosa: 0.9.1," Feb 2022.

[42] S. Böck and G. Widmer, "Maximum filter vibrato suppression for onset detection," in *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13)*, Maynooth, Ireland, September 2013.

[43] D. P. Ellis, Apr 2007, accessed: 2022-05-12. [Online]. Available: https://www.ee.columbia.edu/~dpwe/resources/matlab/chroma-ansyn

[44] G. Koch, R. Zemel, R. Salakhutdinov *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2. Lille, 2015, p. 0.

[45] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 1597–1607. [Online]. Available: http://proceedings.mlr.press/v119/chen20j.html

[46] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning.pdf

[47] J. Nam, J. Herrera, M. Slaney, and J. O. S. III, "Learning sparse feature representations for music annotation and retrieval," in *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012, Mosteiro S.Bento Da Vitória, Porto, Portugal, October 8-12, 2012*, F. Gouyon, P. Herrera, L. G. Martins, and M. Müller, Eds. FEUP Edições, 2012, pp. 565–570. [Online]. Available: http://ismir2012.ismir.net/event/papers/565-ismir-2012.pdf