PERCEPTUAL MUSICAL SIMILARITY METRIC LEARNING WITH GRAPH NEURAL NETWORKS

Cyrus Vahidi[1], *Shubhr Singh*[1], *Emmanouil Benetos*[1], *Huy Phan*[2], *Dan Stowell*[3], *György Fazekas*[1], and *Mathieu Lagrange*[4]

[1] Centre for Digital Music, Queen Mary University of London, UK
[2] Amazon Alexa, Cambridge, MA, USA
[3] Tilburg University, Bijsterveldenlaan, Tilburg, Netherlands
[4] LS2N, Nantes Université, CNRS, École Centrale Nantes, France

ABSTRACT

Sound retrieval for assisted music composition depends on evaluating similarity between musical instrument sounds, which is partly influenced by playing techniques. Previous methods utilizing Euclidean nearest neighbours over acoustic features show some limitations in retrieving sounds sharing equivalent timbral properties, but potentially generated using a different instrument, playing technique, pitch or dynamic. In this paper, we present a metric learning system designed to approximate human similarity judgments between extended musical playing techniques using graph neural networks. Such structure is a natural candidate for solving similarity retrieval tasks, yet have seen little application in modelling perceptual music similarity. We optimize a Graph Convolutional Network (GCN) over acoustic features via a proxy metric learning loss to learn embeddings that reflect perceptual similarities. Specifically, we construct the graph's adjacency matrix from the acoustic data manifold with an example-wise adaptive k-nearest neighbourhood graph: Adaptive Neighbourhood Graph Neural Network (AN-GNN). Our approach achieves 96.4% retrieval accuracy compared to 38.5% with a Euclidean metric and 86.0% with a multilayer perceptron (MLP), while effectively considering retrievals from distinct playing techniques to the query example.

***Index Terms*—** auditory similarity, content-based music retrieval, graph neural networks, metric learning

1. INTRODUCTION

An important aspect of music information retrieval (MIR) is the retrieval of sound samples based on perceived similarity in style or orchestration. In digital music composition, this plays an important role in categorization, organization, and exploration of large musical libraries for creative purposes. At a first level of approximation, MIR research considers the identity of a musical instrument a sufficient proxy for similarity between musical timbres. However, differences in musical instrument playing techniques lead to perceptually distinct sounds even when played on the same instrument [1].

While timbre perception research often employs dimensionality reduction techniques to identify explanatory acoustic dimensions for perceptual data, metric learning over acoustic features has been proposed as a means to model perceptual similarity [2]. However, this approach has limited practical application in music information retrieval due to the use of very small sets of simple sound stimuli.

A recent publication [3] sought to address these limitations by focusing on similarity retrieval in terms of human similarity judgements between musical instrument playing techniques (IPTs), or instrument-mute-technique classes (IMT), from the Studio-on-Line (SOL) dataset. By using time-frequency scattering acoustic features [4], they reported 90% (with a Euclidean metric) and 99% (with a learned linear metric) top-5 similarity retrieval accuracy. However, the study had some limitations: relevant similarity retrievals typically belonged to the same IMT class as the query, suggesting that the system was mainly capable of demonstrating invariance to differences in pitch and dynamics.

In this paper, we demonstrate that retrieval accuracy under the Euclidean metric significantly declines when only considering relevant retrievals with a distinct IMT to the query. We focus on modeling human judgments at the level of clusters, which consist of perceptually similar sounds that belong to different IMT categories. By capturing the perceptual similarities across distinct IMTs, we aim to develop a more robust and accurate system for timbre modeling in a musical sound retrieval context.

Graph Neural Networks (GNNs) have recently gained popularity in the field of similarity retrieval due to their ability to capture complex relationships between data points. GNNs can leverage graph structure of data and can learn representations that are more informative than traditional methods. Euclidean distance or cosine similarity metrics, which are based on pairwise comparisons of feature vectors, can be limited in their ability to capture high-dimensional and non-linear relationships between data points.

Recent research has shown that GNN-based methods can achieve state-of-the-art performance in image [5] and text retrieval [6]. A recent publication demonstrated the effectiveness of GNNs trained under triplet metric learning to model music artist similarity from acoustic features [7]. In contrast, our work does not rely on known similarity relationships derived from metadata (as in [7]), but leverages an adaptive similarity measure that learns from structure within the input data.

The key idea of our approach consists of the following components:

C. Vahidi is a research student at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported jointly by UK Research and Innovation [grant number EP/S022694/1] and Music Tribe. S.Singh is a research student at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported jointly by UK Research and Innovation [grant number EP/S022694/1] and Queen Mary University of London. This work was supported by the Engineering and Physical Sciences Research Council [grant number EP/V062107/1].

979-8-3503-2372-6/23$31.00

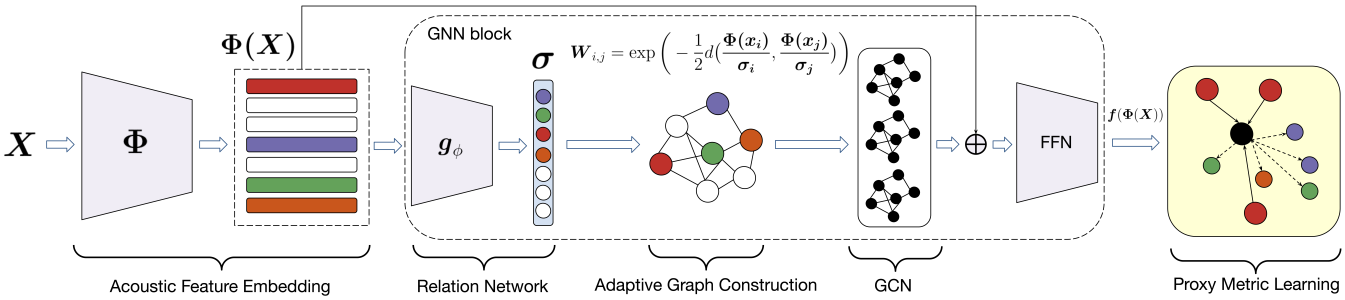

Figure 1: Overview of the system architecture. We transform the input audio waveform into an acoustic feature representation via the operator $\mathbf{\Phi}$ (e.g. Open-L3 deep feature embedding). We compute embeddings for the dataset X using our graph neural network (AN-GNN) architecture (Section 4). We use 2 GNN blocks in series. We construct a graph using input embeddings as nodes, while adaptively computing its adjacency matrix via the *relation network* with a learned example-wise scaling parameter σ (Section 4.2). Graph convolution is performed over the graph (Section 4.1), producing updated node embeddings. The graph convolution output is then passed through a FFN. We optimize the parameters of the network under the Proxy-Anchor metric learning objective [8] (see Section 3.1), using human judgments to determine similarity relationships (see Section 2).

- Similarity Retrieval: to pose a more robust retrieval task, we only consider relevant retrievals from the corpora that do not belong to the same IMT as the query sound.
- Metric learning: We utilize metric learning over Open-L3 deep acoustic features [9, 10] to approximate similarity among the perceptual clusters. We adopt triplet metric learning with the Proxy-Anchor Loss [8], which helps refine the learned embeddings by minimizing intra-cluster distances and maximizing inter-cluster distances. This approach enhances the overall performance of our system in approximating human judgments of similarity across distinct IMTs.
- Learning embeddings: Our approach employs an architecture that combines a multi-layered graph convolutional network (GCN) frontend with a feed-forward network (FFN) backend to learn embeddings. This hybrid architecture enables us to capture complex relationships within the input acoustic features.
- Adaptive neighbourhood adjacency matrix: We consider a GCN architecture that constructs an adjacency matrix by learning an example-wise scaling parameter from data in order to determine a node's neighbourhood.

Overall, our work demonstrates an effective approach for modeling auditory similarity with applications in similarity retrieval, orchestration, assisted composition, and objective music similarity. We provide a publicly available code repository for reproduction of this work[1].

2. DATASET

The Cyberlioz dataset, introduced in [3] as Cyberlioz-v1, is a collection of human similarity judgements between unique musical instrument playing techniques from 31 subjects. The dataset is based on the perceptual similarity judgements of 78 reference sounds, sourced from the comprehensive Studio-on-Line (SOL) (v0.9) dataset [11], sampled at 44.1 kHz. The reference sounds are of equal pitch and dynamics but belong to distinct instrument-mute-technique (IMT) class (e.g. violin+sordino–pizzicato). A 'mute' refers to a device attached to a musical instrument to alter the sound it produces; combined with a range of playing techniques, a diverse range of timbres can be produced by a single instrument. Therefore, the combination of instrument, mute and technique is indicative of the timbral identity of a sample. Participants provided similarity judgments by arranging the 78 sounds into clusters on a 2-dimensional grid. The 31 cluster graphs were aggregated via hypergraph partitioning, resulting in an ensemble of 19 distinct similarity clusters that the 78 sounds are assigned to. The SOL dataset consists of a much larger set of sounds that are equivalent to one of the 78 reference IMT classes, but vary in pitch and dynamics.

As in Cyberlioz-v1, for each reference sound, we include every sample of SOL from the same IMT. As a result, the set of 78 reference sounds (the *seed set*) extends to 7332 sound samples, which we call the *extended set*. In Cyberlioz-v1, only instrument and playing technique are considered as the selection criteria. While this decision seemed reasonable at the time of this study, mute and other kinds of extension can have a drastic effect on timbre. Thus, in Cyberlioz-v2, mute is also considered as a selection criterion. As a consequence, our extended set is fewer in number than [3], due to a more restrictive IMT class equivalence. With this new dataset, Cyberlioz-v2, the task is also made harder, hence we observe reduction in retrieval accuracy when considering the system proposed in [3] (see Section 5.3).

3. METRIC LEARNING OF AUDITORY SIMILARITY

Distance metric learning is of relevance in MIR applications such as content-based music retrieval, music similarity [12], artist similarity [13, 7], few-shot learning for instrument classification [14], representation learning [15] and modelling auditory similarity judgements [3, 16]. In metric learning, the goal is to adapt an embedding function to satisfy some semantic or perceptual similarity constraints of interest.

Our task is to learn a distance metric on acoustic features using human judgements of binary similarity. We aim to learn an embedding representation $z = \boldsymbol{f}(\mathbf{\Phi}(\boldsymbol{x}))$, where $\boldsymbol{f}$ is a learned metric on acoustic features $\mathbf{\Phi}(\boldsymbol{x})$. The metric should satisfy the following must-link ($\boldsymbol{S}$) and not-link ($\boldsymbol{D}$) constraints [17]:

[1]Experiments repository: https://github.com/cyrusvahidi/ipt-similarity

$$\boldsymbol{S} = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) : \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ should be similar}\} \quad (1)$$

$$\boldsymbol{D} = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) : \boldsymbol{x}_i \text{ and } \boldsymbol{x}_j \text{ should be dissimilar}\} \quad (2)$$

3.1. Proxy-Anchor Loss

Metric learning loss functions typically compare pairs or triplets [18] (pair-based), or relate datapoints to learned embeddings per class (proxy-based) [19, 8]. Proxy-Anchor loss [8] significantly mitigates the number of pairwise comparisons required by pair and triplet losses, by only using learned proxy embeddings in the role of "anchor". This solves the computational challenges associated with online sampling of informative tuples in triplet and pair-based losses. The Proxy-Anchor loss is defined as:

$$\begin{aligned}\boldsymbol{\mathcal{L}}(X) = &\frac{1}{|P^+|}\sum_{p\in P^+}\log\big(1+\sum_{x\in X_p^+} e^{-\alpha(s(x,p)-\delta)}\big) \\ &+\frac{1}{|P|}\sum_{p\in P}\log\big(1+\sum_{x\in X_p^-} e^{\alpha(s(x,p)+\delta)}\big)\end{aligned} \quad (3)$$

where X is a batch of embeddings output by the network, $\alpha > 0$ is a scaling factor, $\delta > 0$ is a margin, and s denotes cosine similarity. The set of positive proxies in the batch, P^+, contains the proxies for the unique cluster labels in the batch, while the set P contains all of the available proxies. For a single proxy embedding, p, X_p^+ and X_p^- are the positive and negative embedding vectors in a batch. Hence Eqn. (3) encourages that each embedding $f\Phi(x)$ in a batch is pulled towards its positive proxy, while being repelled from all negative proxies. In our experiments, we initialize 19 learnable proxy embeddings of size 512, i.e. one for each of the perceptual clusters labels in our dataset. Following [8], we set $\alpha = 32$ and $\delta = 0.1$. We refer the reader to [8] for more details on the loss function and its hyperparameters

4. GRAPH NEURAL NETWORKS

4.1. Graph Convolution

Graph Convolutional Networks (GCNs) were originally proposed in [20] to perform classification on graph structured data. The core idea was to update node representations by exchanging information across nodes. GCNs learn a function $f(\cdot,\cdot)$ on a graph G which takes two inputs: a node feature matrix $H_l \in \mathbb{R}^{n\times d}$ and corresponding adjacency matrix $A \in \mathbb{R}^{n\times n}$. The output is an updated node feature matrix $H_{l+1} \in \mathbb{R}^{n\times d}$. In this expression, n denotes the number of nodes, d the dimensionality of node features and l denotes the layer number in the GCN. Based on [20], the convolution expression can be rewritten as:

$$H_{l+1} = h(AH_lW_l) \quad (4)$$

where $W_l \in \mathbb{R}^{d\times d}$ is a transformation matrix to be learned, $A \in \mathbb{R}^{n\times n}$ is the normalized version of the correlation matrix A, and $h(\cdot)$ denotes a non-linear operation, which is the Exponential Linear Unit (ELU) [21] in this work.

4.2. Constructing the Graph

Given a collection of n audio samples, we extract acoustic features with an operator Φ. Each feature vector is treated as the node of our graph. Since the adjacency matrix A is unknown, we derive it directly from acoustic features.

A simple approach is to construct the edges via Euclidean k-nearest neighbours (k-NN) on the node features. For each node v_i, we initialize an empty set of neighbours $\mathcal{N}_0 = \emptyset$. For K iterations of neighbour selection, we select the closest neighbour (that was not already selected) by computing the pairwise Euclidean distance, resulting in a set of K-nearest neighbours:

$$\mathcal{N}_{k+1}(\boldsymbol{v}_i) = \mathcal{N}_k(\boldsymbol{v}_i) \cup \left\{ \underset{\boldsymbol{v}_j \notin \mathcal{N}_k(\boldsymbol{v}_i)}{\arg\min} \|\boldsymbol{\Phi}(\boldsymbol{v}_j) - \boldsymbol{\Phi}(\boldsymbol{v}_i)\|_2 \right\} \quad (5)$$

For each node v_i, we add an edge e_{ji} from v_j to v_i for all $v_j \in \mathcal{N}_k(v_i)$, thus obtaining the graph $G = (V, E)$, where V and E respectively denote the nodes and edges.

Alternatively, we can construct the graph using a Gaussian similarity function:

$$A_{ij} = \exp\left(-\frac{d(v_i, v_j)}{2\sigma^2}\right) \quad (6)$$

where d is the Euclidean distance and σ is an example-wise scale parameter. We predict σ with a learned function over batch samples. By tuning σ according to the data, the neighbourhood structure can be adaptively controlled [22]. Eqn. (6) can be rewritten as

$$A_{ij} = \exp\left(-\frac{1}{2}d\big(\frac{\Phi(v_i)}{\sigma_i}, \frac{\Phi(v_j)}{\sigma_j}\big)\right) \quad (7)$$

where $\sigma_i = g_\phi(v_i)$ and $\sigma_j = g_\phi(v_j)$ are the learned scaling parameters for nodes v_i and v_j. g_ϕ (*relation network* in Fig. 1) is learned by a multi-layered perceptron (MLP) comprising of two linear layers with ReLU activation [23], followed by a linear layer with no non-linearity. The adjacency matrix A is sparsified by keeping the k-max values for each row in A.

4.3. Adaptive Neighbourhood Graph Neural Network (AN-GCN)

Recent works [24, 25] have shown that dynamic graph convolution, where the graph structure is allowed to change, can learn better graph representations than GCNs with fixed graph structure. Based on this, we adapt our model to recompute the graph at each layer from the updated node embeddings. In Fig. 1, we refer to this as Adaptive Neighbourhood Graph Neural Network (AN-GNN).

A single block of our model consists of three components in sequence: a relation network (RN), a GCN network and a feed-forward network (FFN). The RN is used to predict the example-wise scale parameter σ given the node feature embeddings. The GCN processes the input graph and updates the node representations and finally the FFN network (a simple multi-layer perceptron with two fully-connected layers and non-linearity in between) is used on each node to encourage feature diversity and mitigate the over-smoothing phenomenon [26]. Our AN-GNN uses two of such blocks in sequence and $k = 3$ nearest neighbours to compute the adjacency matrix, A, in each block.

As a point of comparison, we replace the adaptive neighbourhood graph with a Euclidean k-nearest neighbour graph ($k = 3$) (kNN-GNN).

5. EXPERIMENTS

5.1. Acoustic Feature Extraction

For the AN-GNN's input nodes, we extract OpenL3 deep embeddings [10] of dimensionality 512 using the music content encoder. For the sake of comparison, we also assess the system with MFCCs and joint time–frequency scattering (JTFS) coefficients [4] as input. We use librosa v0.10.0 [27] to extract 40 MFCCs from the log-power Mel spectrogram of 128 bins. We compute JTFS coefficients with the same hyperparameters and preprocessing steps outlined in [3], but using a recently introduced implementation in Kymatio [28, 29]. We compute each acoustic feature using the first second of every audio example in the dataset and globally average across time. We perform affine standardization of the input features to enforce zero mean and unit variance. We compute the statistics on the train set and propagate them to the test set.

5.2. Training setup

We used the AdamW optimizer with a learning rate of 0.05, weight decay of 1e-5, and trained for 50 epochs using a batch size of 512. To adjust the learning rate, we use a StepLR scheduler with a step size of 10 and gamma value of 0.8. Our model's loss function is the Proxy-Anchor Loss, configured with a margin of 0.1, 19 classes, and an embedding size of 512. To validate the model, we used 5-fold cross-validation with random train-test splits. We train and compare 4 models: a full-rank linear projection, MLP, kNN-GNN, and AN-GNN. The MLP and GNN models have comparable parameter counts. The MLP architecture has two nonlinear layers (ReLU activation, batch normalization, dropout 0.5) with 256 and 512 units, followed by output linear layer of 512 units.

5.3. Evaluation by similarity retrieval

To evaluate the performance of the learned system, we consider a retrieval-by-similarity task.

Let $\mathbb{G}(n)$ denote the perceptual cluster label for audio sample x_n. Using our learned embedding operator, $\phi_k(x_n)$ returns x_n's k-th nearest neighbour in a test dataset of a total of N samples. We define the precision-at-rank-k as follows:

$$p(n, K) = \frac{1}{K} \sum_{k=1}^{K} \mathbf{1}(\phi_k(x_n) \in \mathbb{G}(n)) \tag{8}$$

$$P_(K) = \frac{1}{N} \sum_{n=1}^{N} p_(n, K) \tag{9}$$

where the indicator function $\mathbf{1}(\phi_k(x_n) \in \mathbb{G}(n))$ deems x_n's k-th nearest neighbour as a relevant sample only if it belongs to the same perceptual cluster in Cyberlioz. We consider $K = 5$ nearest neighbours for every query in the test set and only consider retrievals within the test set.

In [3], the authors reported precision-at-k (P@k) of 90% (Euclidean) and 99% (LMNN), where the entire Cyberlioz-v1 dataset was used for train and test.

In this paper, we pose a more challenging retrieval task by:

1. considering a more rigorous annotation scheme with Cyberlioz-v2
2. using a 5-folds cross-validation learning scheme
3. evaluating the system with a figure of merit that ignores relevant retrievals of *identical IMT* class to the query example, termed inter-IMT-precision-at-k (ii-P@k).

	# params	ii-P@5	
		JTFS	**OpenL3**
Euclidean	*0*	33.22 ± 1.9	38.5 ± 1.2
Linear Projection	*262K*	37.6 ± 1.4	56.9 ± 1.0
MLP	*526K*	47.8 ± 4.6	86.0 ± 6.6
kNN-GNN	*529K*	65.3 ± 10.6	95.9 ± 3.2
AN-GNN	*563K*	66.2 ± 9.7	**96.4 ± 3.4**

Table 1: Inter-IMT-precision-at-rank-5 (ii-P@5) retrieval accuracy (see Section 5.3) across 5 cross-validation folds for various architectures: Euclidean distance on identity features, a linear projection, MLP, kNN-GNN and AN-GNN. We compare JTFS (429 dimensions) and OpenL3 (512 dimensions) input features. We list the number of learnable parameters for each setting for OpenL3 input.

For the sake of comparison, we follow the approach outlined in [3] for similarity retrieval with JTFS under a Euclidean metric and a metric learned with the large margin nearest neighbours (LMNN), but Cyberlioz-v2. For this reference system, we observe ii-P@5 of 34.9% (Euclidean) and 37.8% (LMNN), and P@5 of 75.2% and 82.5%, respectively.

In Table 1, we report the ii-P@5 for each architecture with JTFS (429 dimensions) and OpenL3 (512 dimensions) input features. To test for sensitivity to dimensionality in the metric learning approach, we assess performance on random input features of 512 dimensions, observing about 8% ii-P@5 for all architectures. With MFCCs as input, we observe a highest ii-P@5 (40.2%) with AN-GNN. Open-L3 features perform best under every architecture, while the AN-GNN consistently achieves the highest ii-P@5 across 5 cross-validation folds. We observe that the ii-P@k is always substantially lower than the P@k, since many of the top P@k retrievals are likely to be acoustically similar to the query. With MLP, kNN-GNN or AN-GNN performance consistently improves for every input feature. With JTFS and OpenL3 input features, our kNN-GNN and AN-GNN approaches bring substantial improvements in performance over an MLP of comparable parameter count.

6. CONCLUSIONS

We introduced a graph neural network approach for modelling perceptual similarity between musical playing techniques. Our GNN adaptively constructs its adjacency matrices and is trained under a Proxy-Anchor metric learning loss. Our AN-GNN approach demonstrates a highly effective system for learning from human judgements and auditory similarity retrieval, and significantly outperforms previous methods. Our approach offers applications in assisted music composition, content-based sound retrieval, sound sample library navigation and timbre similarity evaluation.

A limitation of this work is in our assumption of absolute invariance of the human similarity judgements to differences in pitch and dynamics between sounds. In future work, we plan to investigate strategies that consider varying degrees of invariance. We intend to explore approaches to graph construction that account for pitch relationships. This work provides several potential avenues for further research, including: expansion beyond musical instrument sounds, learning features directly from data and modelling continuous dissimilarity scores beyond binary supervised clusters.

7. REFERENCES

[1] V. Lostanlen, J. Andén, and M. Lagrange, "Extended playing techniques: the next milestone in musical instrument recognition," in *Proceedings of the 5th international conference on digital libraries for musicology*, 2018, pp. 1–10.

[2] E. Thoret, B. Caramiaux, P. Depalle, and S. Mcadams, "Learning metrics on spectrotemporal modulations reveals the perception of musical instrument timbre," *Nature Human Behaviour*, vol. 5, no. 3, pp. 369–377, 2021.

[3] V. Lostanlen, C. El-Hajj, M. Rossignol, G. Lafay, J. Andén, and M. Lagrange, "Time–frequency scattering accurately models auditory similarities between instrumental playing techniques," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2021, no. 1, pp. 1–21, 2021.

[4] J. Andén, V. Lostanlen, and S. Mallat, "Joint time–frequency scattering," *IEEE Transactions on Signal Processing*, vol. 67, no. 14, pp. 3704–3718, 2019.

[5] X. Zhang, M. Jiang, Z. Zheng, X. Tan, E. Ding, and Y. Yang, "Understanding image retrieval re-ranking: a graph neural network perspective," *arXiv preprint arXiv:2012.07620*, 2020.

[6] Y. Zhang, J. Zhang, Z. Cui, S. Wu, and L. Wang, "A graph-based relevance matching model for ad-hoc retrieval," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4688–4696.

[7] F. Korzeniowski, S. Oramas, and F. Gouyon, "Artist similarity for everyone: A graph neural network approach," *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, 2022.

[8] S. Kim, D. Kim, M. Cho, and S. Kwak, "Proxy anchor loss for deep metric learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3238–3247.

[9] R. Arandjelovic and A. Zisserman, "Look, listen and learn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 609–617.

[10] J. Cramer, H.-H. Wu, J. Salamon, and J. P. Bello, "Look, listen, and learn more: Design choices for deep audio embeddings," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 3852–3856.

[11] C. E. Cella, D. Ghisi, V. Lostanlen, F. Lévy, J. Fineberg, and Y. Maresz, "OrchideaSOL: a dataset of extended instrumental techniques for computer-aided orchestration," *arXiv preprint arXiv:2007.00763*, 2020.

[12] J. Lee, N. J. Bryan, J. Salamon, Z. Jin, and J. Nam, "Disentangled multidimensional metric learning for music similarity," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6–10.

[13] J. Park, J. Lee, J. Park, J.-W. Ha, and J. Nam, "Representation learning of music using artist labels," in *International Society for Music Information Retrieval Conference*. International Society for Music Information Retrieval, 2018.

[14] H. Flores Garcia, A. Aguilar, E. Manilow, and B. Pardo, "Leveraging hierarchical structures for few-shot musical instrument recognition," in *Proc. of the 22nd Int. Society for Music Information Retrieval Conference*, 2021.

[15] J. Lee, N. J. Bryan, J. Salamon, Z. Jin, and J. Nam, "Metric learning vs classification for disentangled music representation learning," in *The 21th International Society for Music Information Retrieval Conference (ISMIR)*. International Society for Music Information Retrieval, 2020.

[16] P. Manocha, A. Finkelstein, R. Zhang, N. J. Bryan, G. J. Mysore, and Z. Jin, "A differentiable perceptual audio metric learned from just noticeable differences," *Proc. Interspeech 2020*, pp. 2852–2856, 2020.

[17] A. Bellet, A. Habrard, and M. Sebban, "Metric learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 9, no. 1, pp. 1–151, 2015.

[18] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3*. Springer, 2015, pp. 84–92.

[19] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[20] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[21] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *arXiv preprint arXiv:1511.07289*, 2015.

[22] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, and Y. Yang, "Learning to propagate labels: Transductive propagation network for few-shot learning," in *7th International Conference on Learning Representations, ICLR 2019*. International Conference on Learning Representations, ICLR, 2019.

[23] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[24] G. Li, M. Muller, A. Thabet, and B. Ghanem, "Deepgcns: Can gcns go as deep as cnns?" in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9267–9276.

[25] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[26] K. Han, Y. Wang, J. Guo, Y. Tang, and E. Wu, "Vision gnn: An image is worth graph of nodes," *Advances in Neural Information Processing Systems*, vol. 35, pp. 8291–8303, 2022.

[27] B. McFee, M. McVicar, D. Faronbi, I. Roman, M. Gover, S. Balke, S. Seyfarth, A. Malek, C. Raffel, V. Lostanlen, and B. van Niekirk et al., "librosa/librosa: 0.10.0," Feb. 2023. [Online]. Available: https://doi.org/10.5281/zenodo.7657336

[28] M. Andreux, T. Angles, G. Exarchakisgeo, R. Leonardu, G. Rochette, L. Thiry, J. Zarka, S. Mallat, J. Andén, E. Belilovsky, *et al.*, "Kymatio: Scattering transforms in python," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 2256–2261, 2020.

[29] C. Vahidi, H. Han, C. Wang, M. Lagrange, G. Fazekas, and V. Lostanlen, "Mesostructures: Beyond spectrogram loss in differentiable time-frequency analysis," *arXiv preprint arXiv:2301.10183*, 2023.