ability to make them relevant [41] since trustworthy AI is not achieved automatically with XAI [55]. One taxonomic distinction we would like to underline is whether methods aim to explain models or data [56]. Our method is intended to belong to the latter category for which the purpose is not to explain how models work (*e.g.* troubleshooting learned filters) but rather to use them as proxies to gain new insights about reality (*e.g.* causal inference [57]).

With all these aspects in mind and based on previous literature, we formalise our three explanation goals:

1. **Attribution.** Identify musical concepts associated with a given track's audio signal.
2. **Transferability.** Detected concepts should be applicable to various settings and tasks.
3. **Generality.** The hierarchy should be independent of the signals and make sense for various settings.

Our first goal is linked to the traditional explanation literature supporting *transparency* [45,47] and is common in concept-learning. We have defined our second and third goals to stress that the finality of our work is to be as universal as possible.

3. LEARNING MUSIC CONCEPTS

In this section, we first adopt the few-shot supervision setting of concept learning to learn the music concept $\vec{v}_C$ of each playlist, viewed as a set C of audio signals examples.

3.1 Background on CAVs

We review some essential notions from the concept learning framework we use to learn playlist concepts [3].

We denote $\vec{f}^{[\ell]}$ the output of the l-th layer of a trained neural network f taking as input some samples $x \sim X$. A "concept activation vector" $\vec{v}_C$ is defined for a set of positive examples C as the normal to a separating hyperplane between $\{\vec{f}^{[\ell]}(x) \mid x \in C\}$ versus negative examples – usually $\{\vec{f}^{[\ell]}(x) \mid x \notin C\}$, we provide an illustration in Figure 1(a). In practice, the hyperplane is learned through a logistic regression on a subset of C, while holding out the remaining samples for validation and testing [1] .

As additional insights, this setting is "sound" for few-shot learning because it uses a pretrained model f with prior knowledge on the domain – relating to transfer learning [58] – and learns a linear mapping on its space, hence reducing the risk of over-parametrisation and justifying why more complex mappings are not considered.

3.2 Sets of examples C

The work of Kim et al. [3] lends itself naturally to learning concepts from playlists. Indeed, playlists can be seen as small sets of positive music examples built around a particular musical idea. As it will be further detailed in our experiments, we can use data from music streaming services where playlists are abundant. We will denote $\mathcal{C} = \{C_i\}_i^K$ the set of K sets of music tracks that will enable to learn a CAV set $\mathcal{V} = \{\vec{v}_i\}_i^K$ discriminating $\mathcal{C}$ in the $\vec{f}^{[\ell]}$ space.

It must be noted that playlists are more than a set of tracks, we usually have access to textual data with a title and a description that help rationalise the learnt concepts.

3.3 Backbone model $\vec{f}^{[\ell]}$

As often in few-shot learning, CAVs learning makes use of a pretrained model f – referred to as *backbone* – to embed samples into a latent space with a higher level of abstraction. In this paper, our working hypothesis is that **music concepts can be described solely through audio signals**. We wanted to fully explore the potential of audio signals, which is a known challenging setting in MIR and recommendation [59], and leave its combination with other sources of features for future work. In addition, having audio inputs enables using our model with any music track from any dataset, hence supporting goal 2, while relying on features such as collaborative filtering or other usage data would have tied our model to a given dataset.

After having experimented with several models, as the recent self-supervised model *CLMR* [60], we have chosen to use *MusiCNN* [61] as the backbone since it had demonstrated consistent performances across various musical tasks, which we deemed on par with our goal 2 and 3. MusiCNN was trained on the *Million Song Dataset* [62].

3.4 Preliminary results

We provide some experimental observations to fix ideas and prepare the next section. If we train CAVs as in the original paper [3] – further detailed in section 5.2, we obtain a set of vectors $\mathcal{V}$ that allows to detect a set of K concepts given any music input spectrogram. An example result of learned concepts is given in Figure 1(b) for one of the studied datasets. Unsurprisingly, we find that many associated concepts end up having close vector weights [2] . If we were to use our concept detectors directly, we would have many concepts activated simultaneously, which could make predictions hard to read when K is very big. On the positive side, note that **vector proximity could be used as a means to quantify concept similarities**. With this idea, we go a step further and next present our method to build a hierarchy to help rationalise concept dependencies.

4. HIERARCHISING MUSIC CONCEPTS

Having learned numerous non-independent concepts $\mathcal{V} = \{\vec{v}_i\}_i^K$, we next elaborate on how to organise them with a structure to make them usable (goals 2, 3).

Many classes of structures are possible, and it would be a mistake to select one with purely theoretical considerations since cognitive and social sciences [63, 64] have underlined that explanatory preferences are shaped by diverse factors that cannot be reduced to a set of so-called golden explanation principles. Therefore, from an empirical standpoint, taxonomies seem to be frequent with music

[1] By definition, $\vec{v}_C$ is the learned parameter of a logistic regression.

[2] In the figure, proximity is defined as the cosine similarity.

concepts (*e.g.* genres [8], instruments [25]). We thus argue that this class of graph is rather natural for humans and chose to consider it only.

4.1 Similarity graph computation

Before mining a hierarchy, it is necessary to determine how concepts – now acting as graph nodes – relate to one another. Essentially, we are trying to define a similarity measure between concepts.

When sampling uniformly from a music dataset, concept activations are rare signals overshadowed by uninformative negative detection. The straightforward solution of estimating the empirical covariance of concept activations is thus not a well-behaved measure of similarity. Fortunately, there is another reliable, interpretable (and faster) way to do it. We propose to consider the positive examples of each concept playlist – embedded in $\vec{f}^{[\ell]}$, and to check on what side of other concept hyperplanes their centroid lies. Formally, given $\mathcal{V} = \{\vec{v}_i\}_1^K$, we compute the matrices of concept similarities S and adjacency A (*i.e.* A is a binarised matrix representing graph edges):

$$S_{i,j}(\mathcal{V}) = \mathbb{E}_{x \sim \mathcal{C}_i} \left[\sigma(\langle \vec{v}_j, \vec{f}^{[\ell]}(x) \rangle) \right] \tag{1}$$

$$A_{i,j}(\mathcal{V}) = \left[S_{i,j}(\mathcal{V}) \geq \frac{1}{2} \right] \tag{2}$$

where σ denotes the logistic sigmoid function and $\mathcal{C}_i$ a random variable sampling spectrogram excerpts from tracks of the playlist C_i. Bias is integrated in $\vec{v}_j$ for simplicity. The threshold at $\frac{1}{2}$ for A translates having each playlist centroid on the positive side of the concept hyperplanes. Because $\mathcal{V}$ is learned through logistic regressions that return well-calibrated probabilities, S and A return similarities with well-defined interpretations.

4.2 Hierarchy extraction

Having defined similarities between concepts, we next simplify S and A to a hierarchy that highlights what concepts are similar to many others and what others represent unique and niche ideas.

For our problem and its scale, we propose to adapt the greedy hierarchy construction of Heymann et Garcia-Molina [33] that makes use of the notion of **betweenness graph centrality** [65]. In this method, given an unweighted graph of similarities, a tree is greedily grown by picking each node in decreasing order of centrality in the similarity graph and linking it to the most similar and already-added node in the tree. We denote by $H(S(\mathcal{V}))$ the result of this algorithm on the similarity graph given by the adjacency matrix $A(\mathcal{V})$ and using $S(\mathcal{V})$ as similarity.

Though this algorithm was originally applied to word tags, we argue that the inductive bias of the targeted hierarchy is applicable to our problem. In particular, the use of centrality is justified by the existence of noisy links in the similarity graph, said links are assumed to be more frequent higher up in the hierarchy (*general-general assumption*). For us, this means that general concepts – *e.g. Pop* and *Rock* – should be more likely to be similar than niche concepts – *Bedroom Pop* and *Goth Rock*. This sounds reasonable for music content since we expect niche concepts to relate to a specific musical idea, style, or instrument.

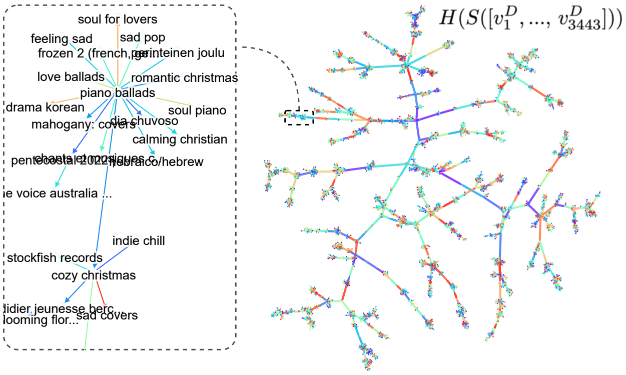

Figure 2. Obtained hierarchy on the Deezer dataset. An interactive plot of this figure, as well as many others, is available at `research.deezer.com/concept_hierarchy/`.

5. EXPERIMENTS

We first detail our experimental setups, then validate that concepts can be reliably learned from audio, and finally show that our proposed structuring method leads to consistent hierarchies for MIR applications.

5.1 Datasets

The set of playlists $\mathcal{C}_D$ we use was exported through the API of the streaming platform Deezer [3]. We have filtered **3635 playlists of mixed types** to be used as concepts: all 1498 editorial playlists available – considered thoughtfully curated but biased by popularity, and 2137 public user playlists with the lower artist popularity – to improve diversity. Playlists have a mean length of 75 tracks and a minimum of 40. This dataset includes the playlist ids, titles, descriptions, and public 30s audio preview for each track. There are 245074 unique tracks from 116497 unique artists: playlists overlap is thus very limited.

In order to compare our computed hierarchy to existing taxonomies, we make use of the APM Music dataset [66]. This is a more traditionally tagged dataset, but its relevance for our task is to include a ground-truth two-level hierarchy of the *music genres* (219) and *moods* (165) tag types. We aggregate sets of positively tagged examples as $\mathcal{C}_{\text{genre}}$ for each genre tags and respectively $\mathcal{C}_{\text{mood}}$ for moods.

5.2 Validating concept learning

We first present experimental evidence related to learning sets of concepts (section 3), to validate goal 1.

5.2.1 Experimental setup

Concept examples are split in a 70% (*train*) / 10% (*validation*) / 20% (*test*) fashion. According to the requirements of our backbone [61], we convert audio inputs to

[3] `https://developers.deezer.com/api`

Source	Audio (↓)	CF (↓)	BERT (↑)	W2V$_1$ (↑)	W2V$_2$ (↑)
$H(S(\mathcal{V}_D))$	**2.449 ± 0.022**	0.845 ± 0.013	0.345 ± 0.007	0.286 ± 0.009	0.542 ± 0.007
$H(S_{\text{CF}})$	**2.413 ± 0.021**	**0.345 ± 0.007**	0.416 ± 0.008	0.336 ± 0.010	0.601 ± 0.008
$H(S_{\text{BERT}})$	2.858 ± 0.028	0.868 ± 0.013	**0.726 ± 0.005**	0.505 ± 0.011	0.652 ± 0.008
$H(S_{\text{W2V-1}})$	2.952 ± 0.028	0.932 ± 0.012	0.523 ± 0.008	**0.804 ± 0.005**	0.721 ± 0.007
$H(S_{\text{W2V-2}})$	2.843 ± 0.026	0.847 ± 0.012	0.531 ± 0.008	0.596 ± 0.009	**0.836 ± 0.004**
Random	3.388 ± 0.027	1.104 ± 0.006	0.239 ± 0.004	0.142 ± 0.006	0.452 ± 0.006

Table 1. Evaluation of the full hierarchy on audio, user-logs and title semantic similarities with 95% confidence. *Audio* and *CF* source are endowed with an **Euclidean distance**, the last three semantic sources use **cosine similarities**.

mel-spectrograms with log-magnitudes. Representations are extracted from the penultimate layer of the backbone, denoted as $f^{[\ell]}$ – which gave us the best performances for concept learning overall. Our complete training code, with further details and exact parameters, is provided for reproducibility at `github.com/deezer/concept_hierarchy`.

5.2.2 *Learning metrics evaluation*

On the Deezer dataset $\mathcal{C}_D$, the mean balanced accuracy of learning concepts is **83.8%** ± **0.3**, with a 95% confidence interval. In detail, we display a histogram of test balanced accuracies in Figure 3, showing that a **majority of playlist concepts can be learned reliably from audio**.

As a rationalisation of failure cases, the hypothesis we made in section 3.4 to detect concepts from audio may not always hold. For instance, it fails when playlist concepts rely on factors extraneous to audio (*e.g.* playlist of a movie soundtrack) or when the backbone space is not expressive enough to discriminate concepts. This actually happens in these experiments with concepts similar in all respects but for their singing languages: *e.g.* the model makes no difference between Japanese and British jazz. We filter concepts v_i below a given test accuracy threshold to account for this effect. Fixed at 70%, 192 concepts are filtered out, leading to $K = 3443$ remaining concepts. This threshold was set empirically by checking that those left-out concepts were indeed unclear from audio (*e.g.* OST, new releases, charts). We denote the resulting concept set $\mathcal{V}_D$.

The two smaller APM datasets lead to similar results with a **78.3%** ± **0.8** for genres, denoted $\mathcal{V}_{\text{genre}}$, and a **71.3%** ± **0.6** accuracy for mood tags, denoted $\mathcal{V}_{\text{mood}}$.

As a wrap-up of this section, goal 1 of detecting concepts from audio seems satisfied in our experiments.

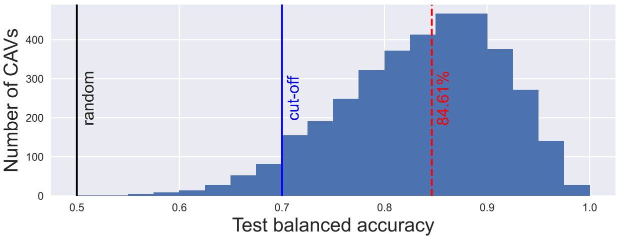

Figure 3. Concept learning performances. The middle line indicates the cut-off at 70% accuracy, the right line is the mean accuracy of the remaining CAVs.

5.3 Hierarchy Mining

We conduct a three-fold evaluation of our proposed hierarchy extraction method (section 4) in order to evaluate goal 3: structure; ground-truth comparison; alignment with other sources of concept similarities.

5.3.1 *Structure priors*

We first provide evidence supporting our choice to extract a hierarchical tree graph over other choices of structures. Though we do not have access to a ground-truth graph of the relations between concepts in $\mathcal{C}_D$, we have some priors on what a satisfying result should be like. For instance, we know that musical concepts are often blended (*e.g.* genres). It is thus safe to assume that a good graph of concepts should be *connected*. Extending this property, isolated nodes should strongly be discouraged. As another prior assumption on our target graph, we know that we can usually find several similar playlists to a given one, but that very strongly connected nodes should be infrequent, *sparsity* should thus also be valued.

With these two priors in mind, we inspect various graph baselines that could straightforwardly be obtained from $\mathcal{V}_D$ without our hierarchy extraction step: the similarity graph $A(\mathcal{V}_D)$ from equation (2); a similarity graph obtained from $S(\mathcal{V}_D)$ with an adjusted threshold to match the sparsity of $H(S(\mathcal{V}_D))$ (*Sparse* A); a top-1 most similar neighbours graph of each node; reference random graphs. Their respective sparsity and connectivity is shown in Table 2. It appears that baseline graphs become disconnected as soon as sparsity drops. We also underline that random baselines provide better-behaved graphs than their counterparts, indicating that similarity links are not evenly distributed among nodes, thus pointing to the existence of communities in the graph that justifies the use of betweenness [67]. Using a hierarchy extraction algorithm helps maintain a low sparsity while connecting every node in the graph.

Graph	#Edges (↓)	#CC (↓)	#IN (↓)
H **(proposed)**	3442 (0.03%)	1	0
$A(\mathcal{V}_D)$	1309163 (11%)	1	0
Sparse A	3443	2766	2735
Top-1	3443	134	0
Random Sparse A	3443	516	430
Random Top-1	3443	7	0

Table 2. Structure evaluation. We count the number of edges in each graph, of connected components (*CC*), and of isolated nodes (*IN*). Lower is better.

This study does not prevent using sparse and connected graphs other than hierarchies. We believe that other relevant structures may exist. Nonetheless, as additional benefits of using hierarchies, we note that tree-like graphs are always planar, which eases their visualisation and navigability by having only one most-relevant parent per node.

5.3.2 Ground-truth evaluation

To validate our generated concept hierarchy, we compare $H(S(\mathcal{V}_{\text{mood}}))$ and $H(S(\mathcal{V}_{\text{genre}}))$ to the expert tag hierarchy provided with the APM dataset – which is unavailable for $\mathcal{V}_D$. As those hierarchies are two-levelled – *i.e.* clustering of tags, we evaluate the accuracy of the mined edges at linking neighbours from the same cluster and compute a silhouette score [68]. Judging from the results given in Table 3, the generation is promising but far from perfect.

The estimated hierarchy of genres is illustrated in Figure 1(c) to help interpret the results. As a typical failure case, the orange cluster ("Electronica") ends up in two separate branches, which can happen because of the greedy construction of our hierarchy. In other cases, however, the ground-truth may be questioned: "Latin influence", "Latin", "Ethnic Dance" and "Calypso" form a link in our hierarchy, but belong to four different clusters in the ground-truth – due to non-musical taxonomic considerations – despite being musically similar. It is not easy to quantify the performance upper-bound we have between our musical concept detection and the practical expert taxonomies we compare to. Nevertheless, seeing that our generated hierarchies roughly align with ground-truth structures is a good sanity check.

Tags	Accuracy (%)	Silhouette
H_{mood}	45.1	-0.09 ± 0.05
H_{genre}	49.1	-0.17 ± 0.04

Table 3. Evaluation of our unsupervised hierarchy against ground-truth clustering of moods and genres tags.

5.3.3 Alignment to various sources

Coming back to the difficult setting of $\mathcal{V}_D$, we finally evaluate our hierarchy of concepts with mixed types, illustrated in Figure 2. As we do not have a ground-truth in this general case, we evaluate whether the edges of our hierarchy – built from $S(\mathcal{V})$ – make sense on other sources of similarity. Specifically, since our data is collected from a streaming service, collaborative filtering embeddings [69] based on listening logs are available to estimate playlist similarities (S_{CF}). Playlists similar in concepts could indeed be expected to be co-listened by users, though popularity biases are also at play. We also leverage playlist titles and expect neighbours to be rather close semantically. To that end, we use a large language model [70] that can embed any text prompt (S_{BERT}), and two music-specialised word embedders for which we average representations of in-vocabulary words of each concept names: $S_{\text{W2V-1}}$ [9] and $S_{\text{W2V-2}}$ [71]. For reference, we generate hierarchies from each domain-specific source, include a random hierarchy, and a *Audio* measure based on CAVs weights, following our observations made in section 3.4.

We compute the average similarity on each hierarchy's edges given those several sources of similarities. Results are provided in Table 1. By construction, each hierarchy maximises performance on the source it was built on. We rather inspect how one source of knowledge transfers to other sources. Collaborative filtering being the canonical way of estimating similarities in the streaming industry our dataset is extracted from and in general [45], we underline that our hierarchy – solely based on audio – closely matches its transfer performances, which is a good result [4]. We have thus shown that our hierarchy transfers to other sources of knowledge (goal 3).

5.4 Qualitative discussion

We retrieve many associations that make sense for humans in $H(S(\mathcal{V}_D))$: blues and jazz; rock and pop; motivation, dancing, running, and party are close to one another – which aligns with previous work [72]. More interestingly, we observe that *"LSD Trip"* is the parent of *"Surf"*, *"Summer of love"*, and *"Stoner Rock"*, effectively associating an activity to a sport, an historic phenomenon, and a music genre. Yet, we also find some association that make sense for the model but not for humans: *e.g.* *"Rock Christmas"* is also a child of *"LSD Trip"*. This phenomenon is hard to avoid with fully unsupervised methods [53, 73]. We can find many more interesting associations, but we have to beware of confirmation biases, as often in XAI [51, 52, 74]. We provide online visualisations of the results [5].

As observable limitations, $\mathcal{V}_D$ is biased by the streaming platform usage towards French, English, and Brazilian content, which could be addressed in future work with importance sampling. Then, some concept titles are misleading, *e.g.* playlists "City sounds: <city>" are confounded by the taste of their curator, resulting in city concepts being close to the 60's genre, despite this not being obvious solely judging from the title. Finally, as already mentioned, the backbone fails to discriminate linguistic information: *e.g.* *"Queer Pop"* and *"[Current] Pop"* differ by their lyric theme but are neighbours in the hierarchy.

6. CONCLUSION

Spectrograms are hard to interpret. We propose to extend concept learning to learn hierarchies of music concepts from playlists, with greater flexibility than usual music taggers. Our results should be viewed as complementary to expert music ontologies, as a means to witness how music is organically described by users and editors, and thus to capture new or evolving salient aspects of music.

This topic is novel and further steps are necessary in order to overcome cultural biases of our data, and to discover causal relations for our structure. Future work include leveraging the found hierarchy for dynamic music recommendation – e.g. exploring and switching branches.

[4] Why not use CF embeddings then? As a reminder, they are tied to the dataset, and thus fail goal 2. They are only used for comparison.

[5] `research.deezer.com/concept_hierarchy/`

7. REFERENCES

[1] A. Liberman, F. S. Cooper, D. P. Shankweiler, and M. Studdert-Kennedy, "Why are speech spectrograms hard to read?" *American Annals of the Deaf*, pp. 127–133, 1968.

[2] M. Won, S. Chun, and X. Serra, "Visualizing and understanding self-attention based music tagging," *Machine Learning for Music Discovery Workshop, ICML*, 2019.

[3] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and R. sayres, "Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV)," in *International Conference on Machine Learning (ICML)*. PMLR, 2018, pp. 2668–2677.

[4] C.-K. Yeh, B. Kim, and P. Ravikumar, "Human-centered concept explanations for neural networks," in *Neuro-Symbolic Artificial Intelligence: The State of the Art*, vol. 342, 2022, pp. 337–352.

[5] A. Ghandeharioun, B. Kim, C.-L. Li, B. Jou, B. Eoff, and R. W. Picard, "Dissect: Disentangled simultaneous explanations via concept traversals," *International Conference on Learning Representations (ICLR)*, 2022.

[6] C. J. Cai, E. Reif, N. Hegde, J. Hipp, B. Kim, D. Smilkov, M. Wattenberg, F. Viegas, G. S. Corrado, M. C. Stumpe *et al.*, "Human-centered tools for coping with imperfect algorithms during medical decision-making," in *CHI conference on human factors in computing systems*, 2019, pp. 1–14.

[7] P. Arendsen, D. Marcos, and D. Tuia, "Concept discovery for the interpretation of landscape scenicness," *Machine Learning and Knowledge Extraction*, vol. 2, no. 4, pp. 397–413, 2020.

[8] R. Hennequin, J. Royo-Letelier, and M. Moussallam, "Audio based disambiguation of music genre tags," *International Society of Music Information Retrieval Conference (ISMIR)*, 2018.

[9] S. Doh, J. Lee, T. H. Park, and J. Nam, "Musical word embedding: Bridging the gap between listening contexts and music," *ICML Workshop: Machine Learning for Music Discovery*, 2020.

[10] E. M. Von Hornbostel and C. Sachs, "Classification of musical instruments: Translated from the original german by anthony baines and klaus p. wachsmann," *The Galpin Society Journal*, pp. 3–29, 1961.

[11] M. Antović, "From expectation to concepts: Toward multilevel grounding in musical semantics," *Cognitive Semiotics*, vol. 9, no. 2, pp. 105–138, 2016.

[12] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *12th International Conference on Computer Vision (ICCV)*. IEEE, 2009, pp. 365–372.

[13] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 951–958.

[14] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang, "Concept bottleneck models," in *International Conference on Machine Learning (ICML)*. PMLR, 2020, pp. 5338–5348.

[15] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su, "This looks like that: deep learning for interpretable image recognition," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[16] B. Zhou, Y. Sun, D. Bau, and A. Torralba, "Interpretable basis decomposition for visual explanation," in *European Conference on Computer Vision (ECCV)*, 2018, pp. 119–134.

[17] A. Lucieri, M. N. Bajwa, S. A. Braun, M. I. Malik, A. Dengel, and S. Ahmed, "On interpretability of deep learning based skin lesion classifiers using concept activation vectors," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2020, pp. 1–10.

[18] A. R. Asokan, N. Kumar, A. V. Ragam, and S. S. Sharath, "Interpretability for multimodal emotion recognition using concept activation vectors," *arXiv preprint arXiv:2202.01072*, 2022.

[19] C. Göpfert, Y. Chow, C.-w. Hsu, I. Vendrov, T. Lu, D. Ramachandran, and C. Boutilier, "Discovering personalized semantics for soft attributes in recommender systems using concept activation vectors," *The Web Conference (WWW)*, 2022.

[20] A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim, "Towards automatic concept-based explanations," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[21] Z. Chen, Y. Bei, and C. Rudin, "Concept whitening for interpretable image recognition," *Nature Machine Intelligence*, vol. 2, no. 12, pp. 772–782, 2020.

[22] E. Strumbelj and I. Kononenko, "An efficient explanation of individual classifications using game theory," *The Journal of Machine Learning Research*, vol. 11, pp. 1–18, 2010.

[23] C.-K. Yeh, B. Kim, S. Arik, C.-L. Li, T. Pfister, and P. Ravikumar, "On completeness-aware concept-based explanations in deep neural networks," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 20 554–20 565, 2020.

[24] Y. Goyal, A. Feder, U. Shalit, and B. Kim, "Explaining classifiers with causal concept effect (cace)," *arXiv preprint arXiv:1907.07165*, 2019.

[25] H. F. Garcia, A. Aguilar, E. Manilow, and B. Pardo, "Leveraging hierarchical structures for few-shot musical instrument recognition," *International Society of Music Information Retrieval Conference (ISMIR)*, 2021.

[26] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[27] T. Griffiths, M. Jordan, J. Tenenbaum, and D. Blei, "Hierarchical topic models and the nested chinese restaurant process," *Advances in Neural Information Processing Systems (NIPS)*, vol. 16, 2003.

[28] V.-A. Nguyen, J. L. Ying, P. Resnik, and J. Chang, "Learning a concept hierarchy from multi-labeled documents," *Advances in Neural Information Processing Systems (NIPS)*, vol. 27, 2014.

[29] G. N. Lance and W. T. Williams, "A general theory of classificatory sorting strategies: 1. hierarchical systems," *The computer journal*, vol. 9, no. 4, pp. 373–380, 1967.

[30] X. Liu, Y. Song, S. Liu, and H. Wang, "Automatic taxonomy construction from keywords," in *SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 1433–1441.

[31] J. Völker and M. Niepert, "Statistical schema induction," in *Extended Semantic Web Conference*. Springer, 2011, pp. 124–138.

[32] V. Anoop, S. Asharaf, and P. Deepak, "Unsupervised concept hierarchy learning: a topic modeling guided approach," *Procedia computer science*, vol. 89, pp. 386–394, 2016.

[33] P. Heymann and H. Garcia-Molina, "Collaborative creation of communal hierarchical taxonomies in social tagging systems," Stanford, Tech. Rep., 2006.

[34] D. Benz, A. Hotho, and G. Stumme, "Semantics made by you and me: Self-emerging ontologies can capture the diversity of shared knowledge," in *Web Science Conference (WebSci09)*, 2010.

[35] A. Ross and F. Doshi-Velez, "Benchmarks, algorithms, and metrics for hierarchical disentanglement," in *International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 9084–9094.

[36] R. Jenatton, J. Mairal, G. Obozinski, and F. R. Bach, "Proximal methods for sparse hierarchical dictionary learning," in *International Conference on Machine Learning (ICML)*, 2010.

[37] E. Bart, I. Porteous, P. Perona, and M. Welling, "Unsupervised learning of visual taxonomies," in *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2008, pp. 1–8.

[38] P. Hase, C. Chen, O. Li, and C. Rudin, "Interpretable image recognition with hierarchical prototypes," in *AAAI Conference on Human Computation and Crowdsourcing*, vol. 7, 2019, pp. 32–40.

[39] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[40] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," *Advances in neural information processing systems (NeurIPS)*, 2018.

[41] Z. C. Lipton, "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." *Queue*, vol. 16, no. 3, pp. 31–57, 2018.

[42] D. Janzing, L. Minorics, and P. Blöbaum, "Feature relevance quantification in explainable ai: A causal problem," in *International Conference on artificial intelligence and statistics*. PMLR, 2020, pp. 2907–2916.

[43] A. Mahinpei, J. Clark, I. Lage, F. Doshi-Velez, and P. Weiwei, "Promises and pitfalls of black-box concept learning models," *ICML Workshop on Theoretic Foundation, Criticism, and Application Trend of Explainable AI*, 2021.

[44] S. Krishna, T. Han, A. Gu, J. Pombra, S. Jabbari, S. Wu, and H. Lakkaraju, "The disagreement problem in explainable machine learning: A practitioner's perspective," *arXiv preprint arXiv:2202.01602*, 2022.

[45] D. Afchar, A. B Melchiorre, M. Schedl, R. Hennequin, E. V. Epure, and M. Moussallam, "Explainability in music recommender systems," in *AI Magazine*, 2022.

[46] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, vol. 2, no. 11, pp. 665–673, 2020.

[47] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," *Information Fusion*, vol. 58, pp. 82–115, 2020.

[48] D. Afchar, V. Guigue, and R. Hennequin, "Towards rigorous interpretations: a formalisation of feature attribution," in *International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 76–86.

[49] P. Hase, H. Xie, and M. Bansal, "The out-of-distribution problem in explainability and search methods for feature importance explanations," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021.

[50] D. Slack, S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju, "Fooling lime and shap: Adversarial attacks on post hoc explanation methods," in *AAAI/ACM Conference on AI, Ethics, and Society*, 2020, pp. 180–186.

[51] H. Kaur, H. Nori, S. Jenkins, R. Caruana, H. Wallach, and J. Wortman Vaughan, "Interpreting interpretability: understanding data scientists' use of interpretability tools for machine learning," in *CHI conference on human factors in computing systems*, 2020, pp. 1–14.

[52] J. Dinu, J. Bigham, and J. Z. Kolter, "Challenging common interpretability assumptions in feature attribution explanations," *NeurIPS Workshop: ML Retrospectives, Surveys & Meta-Analyses (ML-RSA)*, 2020.

[53] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 4114–4124.

[54] L. Sixt, M. Schuessler, O.-I. Popescu, P. Weiß, and T. Landgraf, "Do users benefit from interpretable vision? a user study, baseline, and dataset," in *International Conference on Learning Representations (ICLR)*, 2022.

[55] C. Rudin, C. Chen, Z. Chen, H. Huang, L. Semenova, and C. Zhong, "Interpretable machine learning: Fundamental principles and 10 grand challenges," *Statistics Surveys*, 2021.

[56] H. Chen, J. D. Janizek, S. Lundberg, and S.-I. Lee, "True to the model or true to the data?" *ICML Workshop on Human Interpretability in Machine Learning*, 2020.

[57] J. Pearl, "Causal inference in statistics: An overview," *Statistics Surveys*, vol. 3, pp. 96 – 146, 2009.

[58] S. Bozinovski and A. Fulgosi, "The influence of pattern similarity and transfer of learning upon training of a base perceptron b2.(original in croatian: Utjecaj slicnosti likova i transfera ucenja na obucavanje baznog perceptrona b2)," in *Proc. Symp. Informatica*, 1976, pp. 3–121.

[59] O. Celma, P. Herrera, and X. Serra, "Bridging the music semantic gap," *International Semantic Web Conference*, 2006.

[60] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," *International Society of Music Information Retrieval Conference (ISMIR)*, 2021.

[61] J. Pons and X. Serra, "musicnn: Pre-trained convolutional neural networks for music audio tagging," *Late breaking/demo session of the International Society for Music Information Retrieval Conference (LBD-ISMIR)*, 2019.

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

[63] T. Lombrozo, "Explanatory preferences shape learning and inference," *Trends in Cognitive Sciences*, vol. 20, no. 10, pp. 748–759, 2016.

[64] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artificial intelligence*, vol. 267, pp. 1–38, 2019.

[65] S. P. Borgatti and M. G. Everett, "A graph-theoretic perspective on centrality," *Social networks*, vol. 28, no. 4, pp. 466–484, 2006.

[66] B. Gao, E. Dellandréa, and L. Chen, "Music sparse decomposition onto a midi dictionary of musical words and its application to music mood classification," in *International Workshop on Content-Based Multimedia Indexing (CBMI)*. IEEE, 2012, pp. 1–6.

[67] M. Girvan and M. E. Newman, "Community structure in social and biological networks," *Proceedings of the national academy of sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[68] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[69] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *International Conference on Data Mining*. IEEE, 2008, pp. 263–272.

[70] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," in *Conference on Empirical Methods in Natural Language Processing*. ACL, 2019.

[71] M. Won, S. Oramas, O. Nieto, F. Gouyon, and X. Serra, "Multimodal metric learning for tag-based music retrieval," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 591–595.

[72] K. M. Ibrahim, J. Royo-Letelier, E. V. Epure, G. Peeters, and G. Richard, "Audio-based auto-tagging with contextual tags for music," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 16–20.

[73] F. Dalvi, A. R. Khan, F. Alam, N. Durrani, J. Xu, and H. Sajjad, "Discovering latent concepts learned in bert," in *International Conference on Learning Representations (ICLR)*, 2022.

[74] R. S. Nickerson, "Confirmation bias: A ubiquitous phenomenon in many guises," *Review of General Psychology*, vol. 2, no. 2, pp. 175–220, 1998.

MULTI-OBJECTIVE HYPER-PARAMETER OPTIMIZATION OF BEHAVIORAL SONG EMBEDDINGS

Massimo Quadrana
Apple
mquadrana@apple.com

Antoine Larreche-Mouly
Apple
alarreche@apple.com

Matthias Mauch
Apple
mmauch@apple.com

ABSTRACT

Song embeddings are a key component of most music recommendation engines. In this work, we study the hyper-parameter optimization of behavioral song embeddings based on Word2Vec on a selection of downstream tasks, namely next-song recommendation, false neighbor rejection, and artist and genre clustering. We present new optimization objectives and metrics to monitor the effects of hyper-parameter optimization. We show that single-objective optimization can cause side effects on the non optimized metrics and propose a simple multi-objective optimization to mitigate these effects. We find that next-song recommendation quality of Word2Vec is anti-correlated with song popularity, and we show how song embedding optimization can balance performance across different popularity levels. We then show potential positive downstream effects on the task of play prediction. Finally, we provide useful insights on the effects of training dataset scale by testing hyper-parameter optimization on an industry-scale dataset.

1. INTRODUCTION

Modern Recommendation Systems (RS) rely on embedding vectors to represent the latent user and item factors [1]. They can be employed for several downstream applications, ranging from song recommendation in radio stations or playlists [2–7], search [8, 9], tagging [10], to the generation of artist and genre representations for annotation and recommendation tasks [11–13]. Embeddings are usually generated in the early stages of complex RS pipelines, often through self-supervised methods like Word2Vec [14], which come with default hyper-parameters tuned on non-RS tasks (e.g., NLP).

Practitioners and researchers in the field hence need feasible optimization objectives and reliable metrics to compare against when optimizing embeddings. Recent research shows that hyper-parameter optimization can significantly improve recommendation quality [15, 16]. While prior work mainly focuses on recommendation tasks, it is worth considering other tasks like false positive rejection and clustering during optimization.

Additionally, embedding spaces are used as a source of knowledge and interpretation either through visualization tools [17, 18], or by using relationships between items from a statistical standpoint or by leveraging information outside the input data [19–22]. We believe that music-RS practitioners and researchers could benefit from a deeper understanding of the behavior of song embedding optimization in relation to important factors such as song popularity. This could assist them in handling some emerging algorithmic biases like the popularity bias [23–25].

1.1 Contributions

In this work, we provide a strong framework within which it is possible to monitor the performance of embedding models and evaluate the potential of the embedding model to adapt to new tasks. Our work presents a general methodology that can be applied to any embedding system and not only to Word2Vec. The main contributions are:

- We define metrics and optimization objectives for three relevant tasks: next-song recommendation, false neighbor rejection, and genre and artist clustering.
- We demonstrate experimentally that single-objective optimization can have negative side effects on the non optimized metrics and propose a multi-objective optimization approach to combine recommendation and clustering objectives effectively.
- We show that next-song recommendation quality and song popularity are anti-correlated and reveal that song embedding optimization can balance performance across different popularity levels.
- We show the potential positive downstream effects on the task of play prediction revealing that the benefits of song embedding optimization extend beyond the tasks considered during optimization.
- Finally, we study embedding optimization at scale on an internal dataset of billions of listening events and show that increasing training dataset size allows for better configurations at the expense of longer optimization times.

2. METHOD

We consider song embeddings based on Word2Vec. This model was originally created to represent words in an

Attribution: M. Quadrana, A. Larreche-Mouly, and M. Mauch, “Multi-objective Hyper-parameter Optimization of Behavioral Song Embeddings”, in *Proc. of the 23rd Int. Society for Music Information Retrieval Conf.,* Bengaluru, India, 2022.

English corpus through a self-supervised shallow neural network trained to learn dense vector representations of the words from sentences based on the surrounding words [14]. The same principle is now commonly used in recommender systems to compute item embeddings from user interaction sequences like site sessions or playlists [25–28].

We study song embedding optimization with respect to four main tasks: next-song prediction, false neighbor rejection, artist clustering, and genre clustering. We define in this section the task and metrics that we used both for embedding optimization and evaluation of its effects on the resulting embedding space.

2.1 Next-Song Prediction

We choose next-song prediction as our primary target task. Next-item prediction is a common recommendation task whose goal is to predict the next item the user will interact with given some context [29]. In music recommenders, this often translates to predicting the next song given the history of songs played by the user [30]. To ensure the contextual relevance of recommendations, the past play history is often limited to the past few played songs or sessions, although more flexible solutions that look beyond a fixed horizon exist [31].

Since our main goal is not to build the most accurate next-song predictor, but rather to measure effects of optimization on the embedding space as directly as possible, we simplified next-song prediction to the extreme case of predicting the next played song based only on the *immediately preceding* song. For each song in the evaluation set (the *target* song henceforth), we consider the song the user played before it as the *query* song. For each (query, target) pair, we simply retrieve the top-100 exact nearest neighbors of the query and compute the average HitRate and Normalized Discounted Cumulative Gain (NDCG) on the top-100 ranked neighbors. HitRate is the fraction of times the target song is contained in the nearest neighbor of the query, while NDCG also accounts for the rank of the correct next-song within the predictions [32].

In order to ensure that improvements are due to true generalisation and not to overfitting, we split the evaluation into in-set and out-of-set. In *in-set evaluation*, we mask the last song in every training sequence and use second-to-last song as the query to compute next-song prediction metrics. In *out-of-set evaluation*, we hold-out the whole play sequences in the validation and test sets, and use every ordered pair of songs therein contained to compute the next-song prediction metrics. We use the average in-set and out-of-set HitRate and NDCG values in our experiments. The precise definition of a sequence varies between our experimental datasets and will be discussed in Section 3.

2.2 False Neighbor Rejection

Since high quality nearest neighbors are essential to providing a good recommendations, we are interested in evaluating the effectiveness of filtering out *spurious* song neighbors, i.e., songs that are in the closest neighborhood of another song merely by chance and not due to real behavioral or metadata similarity.

Query song	Hard Negative Neighbors
Paradise - Coldplay	Snowman - Sia Immigrant Song - Led Zeppelin Basket Case - Green Day
Smells Like Teen Spirit - Nirvana	Power - Kanye West Ob-La-Di, Ob-La-Da - The Beatles Rock Your Body - Justin Timberlake
Carry On Wayward Son - Kansas	Natural - Imagine Dragons Rap God - Eminen It Ain't Me - Kygo & Selena Gomez

Table 1: Examples of hard negative neighbors in Stream.

To this end we define the task of False Neighbor Rejection. We used a chi-squared X^2 test to identify false neighbors in the play sequence dataset [33]. The X^2 test compares the observed co-occurrence frequencies of every pair of songs in the listening sequences against the expected co-occurrence frequencies in case of independence. It is thus suitable to detect pairs of songs that appear in sequence by chance, and not because they are strongly related due to behavioral factors or metadata. In practice, we compute the X^2 coefficient for each unordered bigram of songs in the play sequence dataset. We consider as a *hard negative neighbor* of a song any other song having high co-occurrence but chi-squared statistic below a significance threshold [1] .

One limitation of this approach is that it requires a large number of events to be able to find frequently co-occurring song pairs by chance. We were therefore able to run this analysis only on the Stream dataset, from which we retrieved 5M hard-negative pairs, with an average of 92 hard-negatives per song. Some examples of hard negative neighbors are shown in Table 1. We define the HardNeg metric as the proportion of hard negatives for the top-100 nearest neighbors of every song (the smaller the better). We will use it as a safeguard metric agains undesired side effects of song embedding optimization.

2.3 Artist and Genre Clustering

We are interested in measuring how strongly classes such as artists and genres cluster together in a given embedding space. We introduce here the concept of Local Genre Coherence of the embedding space as the average fraction of songs in nearest neighbors set having the same primary genre as the query song. We similarly define the Local Artist Coherence as the average fraction of nearest neighbors belonging to the same artist as the query song. For example, an embedding space has Local Genre Coherence $= 0.5$ if on average 50% of the nearest neighbors of each song have its same primary genre.

Computing Local Coherence metrics is a costly operation since it requires us to inspect the nearest neighbors of every embedded song. We instead propose to use as *proxy metric* during optimization the much cheaper Caliński-

[1] We empirically chose 10^{-7} times the sum of all song occurrences in the dataset as threshold.