STEM-JEPA: A JOINT-EMBEDDING PREDICTIVE ARCHITECTURE FOR MUSICAL STEM COMPATIBILITY ESTIMATION

Alain Riou[1,2] **Stefan Lattner**[2] **Gaëtan Hadjeres**[3] **Michael Anslow**[2] **Geoffroy Peeters**[1]
[1] LTCI, Télécom-Paris, Institut Polytechnique de Paris, France
[2] Sony Computer Science Laboratories - Paris, France
[3] Sony AI
alain.riou@sony.com

ABSTRACT

This paper explores the automated process of determining stem compatibility by identifying audio recordings of single instruments that blend well with a given musical context. To tackle this challenge, we present Stem-JEPA, a novel Joint-Embedding Predictive Architecture (JEPA) trained on a multi-track dataset using a self-supervised learning approach.

Our model comprises two networks: an encoder and a predictor, which are jointly trained to predict the embeddings of compatible stems from the embeddings of a given context, typically a mix of several instruments.

Training a model in this manner allows its use in estimating stem compatibility—retrieving, aligning, or generating a stem to match a given mix—or for downstream tasks such as genre or key estimation, as the training paradigm requires the model to learn information related to timbre, harmony, and rhythm.

We evaluate our model's performance on a retrieval task on the MUSDB18 dataset, testing its ability to find the missing stem from a mix and through a subjective user study. We also show that the learned embeddings capture temporal alignment information and, finally, evaluate the representations learned by our model on several downstream tasks, highlighting that they effectively capture meaningful musical features.

1. INTRODUCTION

Musical stem compatibility indicates the degree to which a stem (i.e., an audio file of a single instrument) fits a given musical context (an audio file of another instrument or a mix of instruments) when played together. Its automatic estimation can be helpful for stem retrieval, automatic arrangement, or stem generation tasks. The compatibility between stems (or a stem and some musical context) depends on several global factors, such as tonality, tempo, genre, timbre, and singing/playing style. In addition, local features like chords or pitches are crucial to performing temporal alignment between a stem and some musical context.

While initial works have studied musical compatibility between songs based on traditional Music Information Retrieval (MIR) tasks like beat tracking and chord estimation [1, 2], more modern approaches aim to learn compatibility directly from data using deep neural networks [3–5]. Using such learning-based approaches extends the notion of compatibility beyond music-theoretical aspects (like tonality and tempo) toward sound-related and expressive characteristics like timbre and playing style.

Moreover, there are potential applications for musical stem generation [6, 7], where generators usually require musical context conditioning to produce compatible accompaniments. With the proposed system, stem representations can be predicted from context information at inference time. This allows training a stem generation model based solely on stem representations, eliminating the need for context/target pairs.

Paper proposal and organization. In this paper, we introduce Stem-JEPA, a novel Joint-Embedding Predictive Architecture (JEPA) which acts directly on mixtures of stems. It consists of two neural networks, an encoder and a predictor, jointly trained to produce representations of a *context* mix and predict representations of a compatible *target* stem. Unlike previous JEPAs [8, 9], our approach does not rely on masking in the input space but rather on omitting stems within the process of mixing, and it uses the label of the missing stem for conditioning (see section 3).

We assess the performance of Stem-JEPA in a retrieval task and through a subjective evaluation in sections 4.1 and 4.2, respectively. Also, we investigate how well the learned representations encode the temporal alignment of stems and mixes (see section 4.3). We also perform an analysis showing that key and chord annotations of audio snippets close in the embedding space are musically compatible (see section 4.4). Finally, we evaluate the representations produced by our model on various downstream MIR tasks (see section 4.5).

To facilitate further research in this direction, we make our code available. [1]

[1] https://github.com/SonyCSLParis/Stem-JEPA

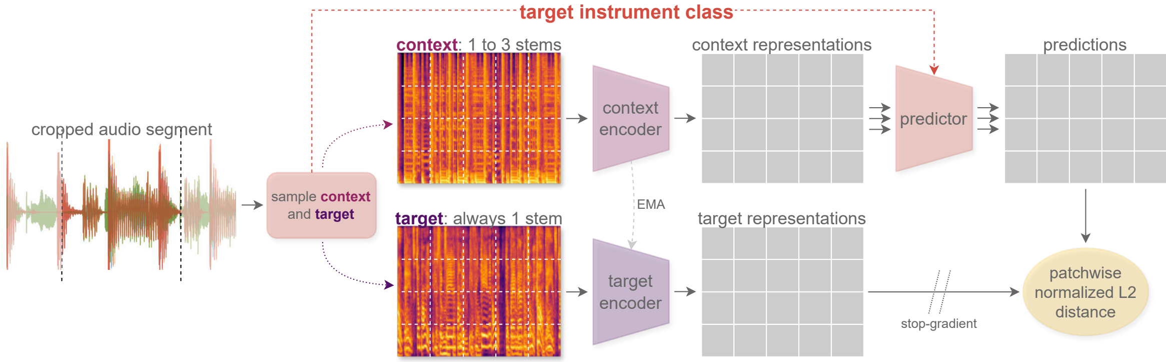

Figure 1. Overview of the Stem-JEPA framework. From an audio clip composed of 4 stems, we first crop a chunk of 8 seconds, then sample the target $\bar{\mathbf{x}}$ (one of the stems) and the context $\mathbf{x}$ (a mix of some of the remaining stems) as described in section 3.2. They are then converted into Log Mel Spectrograms and passed through the *context* and *target* encoders, respectively. Finally, the *predictor* (conditioned on the target instrument label) is trained so that each of its outputs individually predicts each target representation.

2. RELATED WORK

SSL for representation learning. Self-supervised learning (SSL) involves training networks on unlabeled corpora by solving pretext tasks using only the data itself. This paradigm has shown great potential for extracting meaningful representations in various domains [10–13].

A common approach to SSL is contrastive learning [10, 12, 14, 15] or its variants [16, 17]. In autoencoders, an encoder and a decoder are jointly trained to learn latent representations from which the original input can be reconstructed [13, 18]. JEPAs are trained to *predict* some target data from context data directly in the representation space [8, 19].

Joint-Embedding Predictive Architectures. A JEPA is an architecture composed of two trainable networks: an encoder and a predictor. The model receives a context/target pair as input, passes them through the encoder to create latent representations, and then the predictor is trained to predict the target representation from the context representation. Pairs can be generated through various data augmentations [11, 19, 20], or by masking part of the input, as in data2vec [21]. In particular, JEPAs do not require negative samples, unlike contrastive approaches [19], and enable the model to discard uninformative content given that reconstruction is not required.

To prevent model collapse, it is crucial to block gradients in the non-predictor branch [22], treating its output as the target. Moreover, adopting different but tied encoders for each branch as in Eq. (2) helps to stabilize training [19, 22, 23]. Finally, I-JEPA [8] creates pairs through masking and trains the model to predict the representations of small image patches by conditioning the predictor on their positions, allowing the model to grasp local nuances.

Learning from separated sources. Most SSL approaches, often stemming from the vision domain, have been explored and adapted to the audio domain [9, 12, 14, 15, 20, 24–26]. These works are not specific to musical audio, which is typically composed of several stems providing rich compositional potential for SSL. In practice, only a few SSL approaches leverage separated stems for tasks such as audio classification [27], music tagging [28] and beat tracking [29]. Finally, a few works explore modeling the compatibility between stems with applications like automatic mashup creation [1, 3] and sample or loop retrieval for interactive composition [4, 5].

3. STEM-JEPA

3.1 Training pipeline

Our method, depicted in Figure 1, builds upon recent works in JEPAs for image and audio representation learning [8, 9]. Given a music track represented as a set of S stems (roughly corresponding to the separated audio sources) $\mathbf{x}_1, \dots, \mathbf{x}_S$, we crop a chunk of 8 seconds. We then randomly select one of the stems as **target** $\bar{\mathbf{x}} = x_t$ with $t \in \{1, \dots, S\}$ and use the remaining ones to create a **context mix**: $\mathbf{x} = \sum_{c \in C} \mathbf{x}_c$ with $C \subset \{1, \dots, S\} \setminus \{t\}$.

Both $\mathbf{x}$ and $\bar{\mathbf{x}}$ are then converted to Log Mel Spectrograms and divided into a regular grid (over the time and frequency dimensions), leading to K patches. The context patches are then fed to a **context encoder** f_θ to produce patch-wise embeddings $\mathbf{z} = (\mathbf{z}_1, \dots, \mathbf{z}_K)$, where θ are training parameters. Similarly, the target patches are fed to a **target encoder** $f_{\bar{\theta}}$ to produce the patch-wise embeddings $\bar{\mathbf{z}} = (\bar{\mathbf{z}}_1, \dots, \bar{\mathbf{z}}_K)$.

Finally, the context representations $\mathbf{z}$ are independently fed to a **predictor** g_ϕ (with trainable parameters ϕ), which is conditioned on the instrument label l of the missing stem by concatenating a learnable embedding $\mathrm{emb}(l)$ to $\mathbf{z}_k$. The output of the predictor is therefore the prediction $\tilde{\mathbf{z}} = (\tilde{\mathbf{z}}_1, \dots, \tilde{\mathbf{z}}_K)$, with $\tilde{\mathbf{z}}_k = g_\phi(\mathrm{concat}(\mathbf{z}_k, \mathrm{emb}(l)))$.

As in [8, 11, 19], the parameters (θ, ϕ) of the context encoder and predictor are updated through gradient descent by minimizing the mean squared error $\mathcal{L}(\tilde{\mathbf{z}}, \bar{\mathbf{z}})$ between the (normalized) predicted and target representations:

$$\mathcal{L}(\tilde{\mathbf{z}}, \bar{\mathbf{z}}) = \frac{1}{K} \sum_{k=1}^{K} \left\| \frac{\tilde{\mathbf{z}}_k}{\|\tilde{\mathbf{z}}_k\|} - \frac{\bar{\mathbf{z}}_k}{\|\bar{\mathbf{z}}_k\|} \right\|^2, \tag{1}$$

whereas the parameters of the target encoder $\bar{\theta}$ are updated using an Exponential Moving Average (EMA) of the ones of the context encoder, i.e.,

$$\bar{\theta}_i = \tau_i \bar{\theta}_{i-1} + (1 - \tau_i)\theta_i, \quad (2)$$

where the EMA rate τ_i is linearly interpolated between τ_0 and τ_T, T being the total number of training steps.

3.2 Sampling context and target

To avoid training the system on silent target stems or silent context mixes, we first analyze the amplitude content of each of the stems $\mathbf{x}_1, \dots, \mathbf{x}_S$ representing a chunk of a given music track.

Let $\mathcal{A} \subset \{1, \dots, S\}$ be the indices of active (i.e., non-silent) stems among $\mathbf{x}_1, \dots, \mathbf{x}_S$. We first pick a random index $t \in \mathcal{A}$ as target [2]. Then, we randomly select a subset $C \subset \mathcal{A} \setminus \{t\}$ from the remaining non-silent tracks. The number of stems $|C|$ in this subset is uniformly sampled between 1 and the number of other non-silent stems $|\mathcal{A}| - 1$. Most of the time, the prediction task incorporates more stems in the context than in the target ($|C| > 1$), simplifying the predictor's task. However, occasionally, the subset consists of only one stem ($|C| = 1$), allowing the model to process individual stems and learn their representations, which is crucial as these are also used as targets.

3.3 Architecture and training details

We employ a standard ViT-Base model as the encoder [30]. Our predictor is a 6-layer MLP with ReLU activations and 1024 dimensions in each hidden layer. In our ablation studies, the Transformer predictor we use is the same as in [9].

During training, we extract audio chunks of 8 seconds that are converted to log-scaled Mel Spectrograms with 80 mel bins and a window and hop size of 25 and 10 ms, respectively. We use patches of size 16×16, leading to sequences of $\frac{80}{16} \times \frac{800}{16} = 250$ tokens during training.

We train our model during 300k steps using AdamW [31], with a batch size of 256, a base learning rate of 3e-4, and a cosine annealing scheduling after 20k steps of linear warmup. All other hyperparameters are consistent with those used in [9], following their demonstrated effectiveness. Our model is trained for approximately four days on a single A100 GPU with 40 GB of memory.

3.4 Training data

We train the model on a proprietary dataset of 20k multi-track recordings of diverse music genres (e.g., pop/rock, R&B, rap, country) with a total duration of 1350 hours. We use existing instrument annotations to construct four standard categories: Bass, Drums, Vocals, and Other.

[2] If $|\mathcal{A}| < 2$ (a whole chunk is silent or only one active stem), we re-sample another audio chunk from the same track to prevent having silent context or target.

4. EVALUATION

We assess the efficacy of our model to retrieve compatible stems from a given mix through objective and subjective evaluations. We also demonstrate that the learned representations capture local harmonic and rhythmic information. Finally, we show that they also encode high-level features, making them suitable for various MIR tasks.

4.1 Stem retrieval task

Given an input audio, our model predictor has been trained to output a latent representation of a stem such that this stem would fit well with the input audio. To evaluate the performance of our model, we construct a retrieval task in which, given an existing music track, the model should be able to predict the representation of one stem given the mix of the others.

4.1.1 Experimental setup

For evaluation, we used the MUSDB18 dataset [32], which contains $N = 150$ tracks $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}$, each track $\mathbf{x}^{(n)}$ being composed of $S = 4$ stems $\mathbf{x}_1^{(n)}, \dots, \mathbf{x}_S^{(n)}$ (vocals, bass, drums, other). This allows a total of $N \times S = 600$ runs. For any individual stem $\mathbf{x}_s^{(n)}$, define $\mathbf{x}_{\neg s}^{(n)}$ the mix containing all stems from $\mathbf{x}^{(n)}$ except $\mathbf{x}_s^{(n)}$. We aim to predict the embedding of the individual stem $\mathbf{x}_s^{(n)}$ from the one of the mix $\mathbf{x}_{\neg s}^{(n)}$. We compute and average (over time) the patch-wise representations of all stems $\mathbf{x}_s^{(n)}$. It gives us a *reference set* $\mathbf{Z} = \{\mathbf{z}_s^{(n)}\}$, with $\mathbf{z}_s^{(n)}$ being the embedding of $\mathbf{x}_s^{(n)}$. Then, we encode all mixes $\mathbf{x}_{\neg s}^{(n)}$, pass the resulting representations through the predictor conditioned on s, and average (over time and frequency) the result to get a *query* embedding $\mathbf{q}_s^{(n)}$. In other words, $\mathbf{q}_s^{(n)}$ is the prediction of (the embedding of) the missing instrument $\mathbf{x}_s^{(n)}$ from the remaining ones $\mathbf{x}_{\neg s}^{(n)}$. We therefore test if the actual embedding $\mathbf{z}_s^{(n)}$ is among the nearest neighbors of $\mathbf{q}_s^{(n)}$ in the reference set $\mathbf{Z}$.

Metrics. We measure the model performance using two metrics. The *Recall at* K (R@K) measures the proportion of relevant items successfully retrieved among the top K nearest neighbors. We consider here $K \in \{1, 5, 10\}$.

The *Normalized Rank* [5] of a query $\mathbf{q}_s^{(n)}$ is defined as the rank of the ground-truth $\mathbf{z}_s^{(n)}$ in the sorted list of distances $\{d(\mathbf{q}_s^{(n)}, \mathbf{z})\}_{\mathbf{z} \in \mathbf{Z}}$, normalized by the length of the list (here 600) to get a value in $[0, 1)$. For example, a mean Normalized Rank of 5% means that the actual embedding $\mathbf{z}_s^{(n)}$ is, on average, within the 5% nearest neighbors from the prediction $\mathbf{q}_s^{(n)}$. For each model, we report mean and median Normalized Ranks.

4.1.2 Results

The results are shown in Table 1 under the row "MLP w/ cond." Our model achieves a R@1 of 33%, and in half of the cases, the correct stem is within the top 0.5% of nearest neighbors (median Normalized Rank is 0.5%). Moreover, the median rank consistently outperforms the mean rank, indicating the presence of outliers with very high ranks.

Table 1. Influence of the design of the predictor on the retrieval performances. All metrics are in percentages.

Model	Recall ↑ R@1	R@5	R@10	Normalized Rank ↓ mean	median
MLP w/ cond.	**33.0**	**63.2**	**76.2**	**2.0**	**0.5**
MLP w/o cond.	28.2	58.0	69.2	3.3	0.7
Transformer	5.2	17.5	25.7	12.1	6.0
AutoMashupper	1.0	8.8	15.5	29.1	19.5

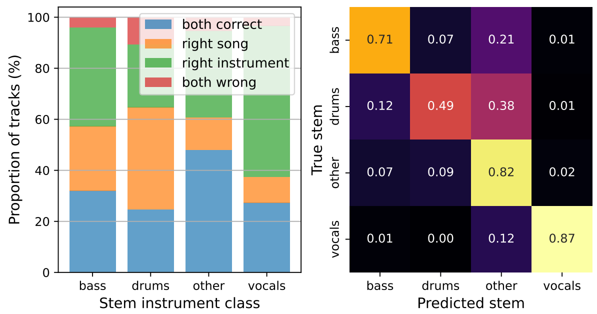

Figure 2. Analysis of the closest embedding $\mathbf{z}^*$ for all queries $\mathbf{q}$ from the MUSDB18 dataset [32]. Left: Categories of failures for each instrument (same song but wrong instrument, the opposite, or both wrong). Right: confusion matrix between conditioning instruments and retrieved instruments.

We also include in Table 1 results for scenarios without predictor conditioning during training (row "MLP w/o cond.") and when using a Transformer instead of an MLP for the predictor. In both cases, the performance drops substantially, emphasizing the importance of conditioning for the retrieval task. When using an MLP instead of a Transformer, the encoder must capture global information because the MLP cannot infer it, which leads to more informative embeddings.

Finally, we compare our model with AutoMashupper [1], which is, to the best of our knowledge, the only openly available work on compatibility estimation. We use their "mashability" measure as a similarity metric to compute the retrieval performances. Note that this metric involves beat tracking and chord detection, making it unsuited for vocals and drum stems, respectively. Therefore, the performance of this method on the retrieval task is relatively weak.

4.1.3 Influence of the instrument class

To get a better understanding of the failure modes of our model and the disparities between the different instruments, we study the nearest neighbor $z^* = \arg\min_{\mathbf{z}\in\mathbf{Z}} d(\mathbf{q}, \mathbf{z})$ for all queries $\mathbf{q}$ from MUSDB18. This analysis, detailed in Figure 2 (left), categorizes z^* into four groups: "both correct" where the model predicts the correct instrument from the correct song, "right instrument" where the correct instrument is predicted but from a different song, "right song" where the model predicts the wrong instrument class but from the correct song, and "both wrong". Additionally, Figure 2 (right) displays a confusion matrix for the instruments.

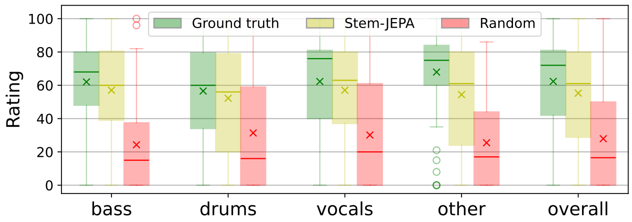

Figure 3. Box plot of the listening test for the different instrument classes. The × represents the mean of the data.

A noticeable result is that the retrieval performances vary a lot between the different instruments, especially between "drums" (R@1 $\simeq$ 25%) and "other" (R@1 $\simeq$ 45%).[3] A plausible explanation is that there are simply more possible candidate drum patterns that actually fit a given mix, resulting in closer neighbors within which it is harder to detect the ground truth.

Additionally, we can see that the "both wrong" scenario is quite uncommon. However, for bass and drums in particular, we predict another instrument (but for the correct song) in more than 25% of the cases. The confusion matrix shows that the category that mostly causes this failure is "other". A reason is probably that "other" is a broad and ill-defined set of instruments that could arguably overlap with bass or drums (e.g., choirs, synth bass, xylophone...).

4.2 User study

In section 4.1, we utilize the compatibility of a mix and a stem from the same song to assess the retrieval performance of our model. However, it is plausible that the dataset also includes compatible stems originally part of different songs. To evaluate our model's ability to retrieve these compatible yet non-original stems, we conduct an online listening test, focusing on retrieving instruments that are not present in the query mix (green segment in Figure 2).

For each trial, the user first listens to a query mix with one missing stem, followed (in random order) by the actual missing stem, the one retrieved by our system, and a random one, but with the same instrument class as $\mathbf{x}_s^{(n)}$. They are then asked to rate (from 0 to 100) the three proposed stems' compatibility with the reference mix.[4]

The mixes and stems are 16-second chunks from the MUSDB18 dataset [32], randomly cropped to 10 seconds during the test to prevent listeners from relying on temporal alignment for rating. We conduct our study on the Go Listen platform [33]. Our test comprises 60 trials, and each user has to answer 12 of them (3 for each instrument class). We had 23 participants, 20 of whom had musical experience (11 for at least 10 years).

Results. The listening test results are depicted in Figure 3. While the ratings for the stems retrieved by our model

[3] The proportion of "both correct" samples is exactly the Recall at 1.
[4] Since we already test temporal alignment and tonality in sections 4.3 and 4.4 respectively, participants are explicitly instructed to rather concentrate on genre, timbre, and playing/singing style in this study.

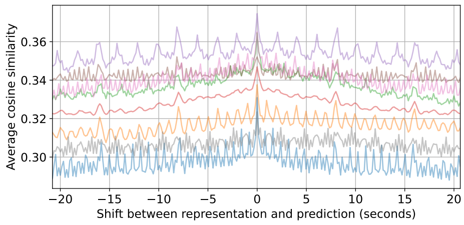

Figure 4. Average pairwise cosine similarity between embeddings and predictions across various temporal shifts. Each curve corresponds to a different track.

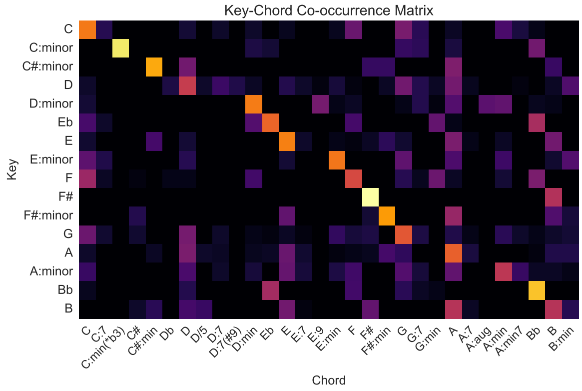

Figure 5. Key/Chord co-occurrence matrix between segments within the same clusters.

are slightly lower than those for the ground truth on average, they are substantially higher (approximately double) than the ratings of random samples. This highlights the ability of Stem-JEPA to retrieve stems compatible with the context mix.

However, we find some disparities between instrument classes. For example, the ratings for drums between the ground truth and our model's suggestions are very close, whereas they are more different for the "other" category. Also, the variance of ratings is higher in "drums," hinting at a generally higher compatibility of drums with any context. Finally, the length of the whiskers and the difference between the mean and median reveal significant disparities between users and samples, indicating the high subjectivity and difficulty of musical compatibility estimation.

4.3 Stem alignment analysis

In this section, we assess the model's ability to evaluate the alignment between stems and mixes by temporally shifting them relative to each other. Our primary metric for this evaluation is the cosine similarity between learned embeddings and their predictions at various offsets, reflecting the local temporal features captured by the model.

Contrary to our previous approach that utilized embeddings averaged over time, here we retain the temporal sequence of the embeddings. We concatenate embeddings in the frequency dimension and stack them in the time dimension, maintaining a resolution of one embedding per 160 milliseconds of audio. We denote the representation of the i-th patch in stem $\mathbf{x}_s^{(n)}$ as $\mathbf{z}_s^{(n)}[i]$ and its corresponding predicted output conditioned on the mix $\mathbf{x}_{\neg s}^{(n)}$ as $\mathbf{q}_s^{(n)}[i]$.

We evaluate the fidelity of these embeddings by examining how the cosine similarity between $\mathbf{z}_s^{(n)}[i]$ and $\mathbf{q}_s^{(n)}[(i+j)\%M]$ evolves with varying j, the temporal offset. The formulation is given by:

$$s(\mathbf{z}, \mathbf{q}, j) = \frac{1}{MS} \sum_{s=1}^{S} \sum_{i=1}^{M} \langle \mathbf{z}_s[i], \mathbf{q}_s[(i+j)\%M] \rangle \quad (3)$$

where M is the total number of embeddings in sequence $\mathbf{z}$, S is the number of stems, and j represents the shift index.

The local nature of the information captured by the embeddings is reflected in how the cosine similarity $s(\mathbf{z}, \mathbf{q}, j)$ changes with different temporal offsets j. Specifically, if the embeddings predominantly contained global information, s would remain relatively constant across shifts. Conversely, a sharp peak in similarity at $j = 0$, followed by a rapid decrease, suggests that the embeddings are rich in local information and less information is shared between adjacent frames.

From our analysis of tracks from the MUSDB18 dataset (8 of them being displayed in Figure 4), we first observe that $s(\mathbf{z}, \mathbf{q}, j)$ always remains relatively high [5], which indicates that the embeddings contain global information. We, however, observe a peak at $j = 0$, underlining the presence of local details that are temporally aligned.

We also observe periodic patterns in the curves, highlighting the model's capacity to capture temporal structures (e.g., beats and bars). Finally, we observe smaller peaks every 8 seconds, the duration of the chunks used for computing the embeddings, which implies that embeddings also capture global position information. This behavior could potentially be avoided by replacing the absolute positional encodings in our encoder with other variants.

An interactive version of Figure 4 with audio examples is provided on the accompaniment website. [6]

4.4 Musical plausibility

We utilize key and chord annotations from Isophonics [7] for 174 Beatles songs to label all patch embeddings. We then conduct k-means clustering on the latent space with $k = 32$ clusters. Within each cluster, we calculate the co-occurrence of all key and chord pairs and aggregate these counts across all clusters. To visualize these relationships, we display in Figure 5 a co-occurrence matrix for the keys and chords that appear in the top 80 most frequent combinations, considering all possible pairs for counting, not just the most common ones.

The matrix reveals that pairs close in the latent space often share significant musical relevance. The highest occurrences typically connect a key with its tonic (e.g., E/E), and prominently with its subdominant and dominant (e.g., C/F, G or D/G, A). Such patterns indicate that the embeddings capture meaningful tonal relationships.

[5] As a reference, the average cosine similarity between random representations and predictions is approximately 0.17.
[6] `https://sonycslparis.github.io/Stem-JEPA`
[7] `http://isophonics.net/`

Table 2. Datasets used for downstream tasks.

Dataset	classes	Task
Giantsteps (GS) [36]	24	Key detection
GTZAN [37]	10	Genre classification
MagnaTagATune (MTT) [38]	50	Tagging
NSynth [39]	11	Instr. classification

4.5 Benchmark on downstream tasks

Lastly, we investigate the musical features encoded in the representations learned by our model. We hypothesize that the encoder captures shared musical information among different stems of the same track, such as rhythm or harmony, to aid the predictor. To verify this, we evaluate it on several downstream classification tasks, a standard protocol for representation learning methods [9, 12, 34, 35].

4.5.1 Experimental setup

Our experimental setup follows the constrained track of the MARBLE benchmark [35]. Each audio sample is processed by the *frozen* encoder, and its patch-wise outputs are concatenated and averaged along frequency and time dimensions to produce a 3840-dimensional global embedding, following [9]. These embeddings are passed through an MLP with 512 hidden units and a softmax layer, which is trained by minimizing the cross-entropy between the predicted distribution and the ground truth labels.

Downstream tasks. To validate our hypothesis, we focus on global musical features that are shared among the different stems of a track, namely tagging, key, and genre estimation. Additionally, we include an instrument classification task to observe whether the encoder preserves stem-specific information. For facilitating comparisons to existing work, we also pick our downstream tasks from the MARBLE benchmark [35]. The full list of datasets and associated tasks is depicted in Table 2.

Baselines. We compare our model to two variants: one trained with a Transformer as predictor, and one without conditioning, as in section 4.1. In addition, we include the two top-performing models from [35] in the considered tasks, namely MULE [12] and Jukebox-5B [40] as references. MULE [12] is an SSL model based on SF-NFNet-F0 [41] trained by contrastive learning on the MusicSet dataset (117k hours), while Jukebox [40] is a huge music generation model trained using codified audio language modeling on 1.2 million songs.

For a more in-depth description of the hyperparameters, datasets, tasks, and corresponding metrics, we refer the reader to [35].

4.5.2 Results

The performances of our model on downstream tasks are provided in Table 3. First, we observe that the choice of the predictor used for training, while extremely influencing for retrieval tasks, has little effect on the downstream performances of our encoder, apart from key detection on Giantsteps, for which the model trained with a Transformer predictor clearly outperforms the others. The performances on NSynth also reveal that our model does not only capture information shared between stems but also stem-specific features. Surprisingly, this holds even without conditioning the predictor during training, and more generally, not conditioning the predictor improves performance on most downstream tasks.

Table 3. Influence of the predictor architecture on the performances of Stem-JEPA on various downstream tasks, and comparison with existing baselines.

	GS	GTZAN	MTT		NSynth
Model	$\mathrm{Acc}^{\mathrm{refined}}$	Acc	ROC	AP	Acc
MLP w/ cond.	40.2	68.6	89.9	42.8	73.5
MLP w/o cond.	36.8	72.5	90.1	**42.9**	**75.0**
Transformer	46.0	68.1	90.0	42.7	73.3
MULE [12]	**64.9**	75.5	91.2	40.1	74.6
Jukebox [42]	63.8	**77.9**	**91.4**	40.6	70.4

We also compare our model to state-of-the-art works in music representation learning. Our performances are on par with baselines for two tasks (MTT and NSynth) but significantly lower on Giantsteps and GTZAN, despite being much better than random guessing. Considering the limited quantity of training data compared to the baselines (about 100 times less), these results suggest that our method is promising for music representation learning but that further efforts have to be made to make it competitive with current state-of-the-art approaches in this field.

5. CONCLUSION

In this study, we introduce a novel SSL paradigm based on stem prediction for musical stem compatibility estimation through the prism of representation learning. Our results show promising performances for retrieval applications and also indicate that the learned representations are localized, suggesting that they could also be valuable for music generation and possibly automatic arrangement. Additionally, these representations are musically meaningful and demonstrate linear separability for various Music Information Retrieval tasks.

Moreover, our model is, to the best of our knowledge, the first use of the predictor component of Joint-Embedding Predictive Architectures (JEPAs) during inference. Employing JEPAs to model compatibility instead of similarity, with appropriate conditioning, may open up possibilities in various fields beyond music.

Nevertheless, our study is not without its limitations. In particular, self-supervised learning usually benefits from very large corpora of training data; however, accessing large datasets of separated stems is challenging, though advancements in source separation technology may alleviate some of these issues. Finally, restricting the analysis to four instruments, while standard in source separation, currently limits the generalizability of our findings. Ideally, extending the predictor to accommodate any instrument would prevent the failure cases illustrated in section 4.1.3 and enhance the model's utility, representing an exciting direction for future research.

6. ACKNOWLEDGMENTS

This work has been funded by the ANRT CIFRE convention n°2021/1537 and Sony France. This work was granted access to the HPC/AI resources of IDRIS under the allocation 2022-AD011013842 made by GENCI. We would like to thank Cyran Aouameur and Marco Comunità for their helpful suggestions. Finally, we would like to thank the reviewers and meta-reviewer for their valuable comments.

7. REFERENCES

[1] M. E. P. Davies, P. Hamel, K. Yoshii, and M. Goto, "Automashupper: automatic creation of multi-song music mashups," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 22, no. 12, p. 1726–1737, dec 2014.

[2] C.-L. Lee, Y.-T. Lin, Z.-R. Yao, F.-Y. Lee, and J.-L. Wu, "Automatic mashup creation by considering both vertical and horizontal mashabilities," in *International Society for Music Information Retrieval Conference*, 2015.

[3] J. Huang, J.-C. Wang, J. B. Smith, X. Song, and Y. Wang, "Modeling the compatibility of stem tracks to generate music mashups," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, 2021, pp. 187–195.

[4] B. Y. Chen, J. B. Smith, and Y. H. Yang, "Neural Loop Combiner: Neural Network Models for Assessing the Compatibility of Loops," in *Proceedings of the 21st International Society for Music Information Retrieval Conference, ISMIR 2020*. International Society for Music Information Retrieval, aug 2020, pp. 424–431.

[5] S. Lattner, "Samplematch: Drum sample retrieval by musical context," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, 2022, pp. 781–788.

[6] J. Nistal, M. Pasini, C. Aouameur, M. Grachten, and S. Lattner, "Diff-a-riff: Musical accompaniment co-creation via latent diffusion models," 2024.

[7] J. D. Parker, J. Spijkervet, K. Kosta, F. Yesiler, B. Kuznetsov, J.-C. Wang, M. Avent, J. Chen, and D. Le, "Stemgen: A music generation model that listens," 2024.

[8] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. G. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*. IEEE, 2023, pp. 15 619–15 629.

[9] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Masked modeling duo: Learning representations by encouraging both networks to model the input," in *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*. IEEE, 2023, pp. 1–5.

[10] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *37th International Conference on Machine Learning, ICML 2020*, vol. PartF16814. International Machine Learning Society (IMLS), feb 2020, pp. 1575–1585.

[11] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "BYOL for Audio: Exploring Pre-Trained General-Purpose Audio Representations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1–15, apr 2022.

[12] M. C. McCallum, F. Korzeniowski, S. Oramas, F. Gouyon, and A. F. Ehmann, "Supervised and Unsupervised Learning of Audio Representations for Music Understanding," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022*, oct 2022.

[13] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 4171–4186, oct 2018.

[14] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2021-June. Institute of Electrical and Electronics Engineers Inc., oct 2021, pp. 3875–3879.

[15] J. Spijkervet and J. A. Burgoyne, "Contrastive Learning of Musical Representations," *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021*, mar 2021.

[16] A. Bardes, J. Ponce, and Y. LeCun, "VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning," in *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, may 2022.

[17] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *37th International Conference on Machine Learning, ICML 2020*, vol. PartF16814. International Machine Learning Society (IMLS), may 2020, pp. 9871–9881.

[18] K. He, X. Chen, S. Xie, Y. Li, P. Dollar, and R. Girshick, "Masked Autoencoders Are Scalable Vision

Learners," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2022-June. IEEE Computer Society, nov 2022, pp. 15 979–15 988.

[19] J. B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko, "Bootstrap your own latent a new approach to self-supervised learning," in *Advances in Neural Information Processing Systems*, vol. 2020-Decem. Neural information processing systems foundation, jun 2020.

[20] X. Li and X. Li, "ATST: Audio Representation Learning with Teacher-Student Transformer," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2022-Septe. International Speech Communication Association, apr 2022, pp. 4172–4176.

[21] A. Baevski, W. Hsu, Q. Xu, A. Babu, J. Gu, and M. Auli, "data2vec: A general framework for self-supervised learning in speech, vision and language," in *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, ser. Proceedings of Machine Learning Research, vol. 162. PMLR, 2022, pp. 1298–1312.

[22] X. Chen and K. He, "Exploring simple Siamese representation learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, nov 2021, pp. 15 745–15 753.

[23] Y. Tian, X. Chen, and S. Ganguli, "Understanding self-supervised learning dynamics without contrastive pairs," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, ser. Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 10 268–10 278.

[24] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[25] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu, "W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 244–250.

[26] P. Huang, H. Xu, J. Li, A. Baevski, M. Auli, W. Galuba, F. Metze, and C. Feichtenhofer, "Masked autoencoders that listen," in *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[27] E. Fonseca, A. Jansen, D. P. Ellis, S. Wisdom, M. Tagliasacchi, J. R. Hershey, M. Plakal, S. Hershey, R. C. Moore, and X. Serra, "Self-supervised learning from automatically separated sound scenes," in *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2021, pp. 251–255.

[28] C. Garoufis, A. Zlatintsi, and P. Maragos, "Multi-Source Contrastive Learning From Musical Audio," in *Proceedings of the Sound and Music Computing Conferences*, vol. 2023-June. Sound and Music Computing Network, feb 2023, pp. 162–169.

[29] D. Desblancs, V. Lostanlen, and R. Hennequin, "Zero-note samba: Self-supervised beat tracking," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2023.

[30] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[31] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[32] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," Dec. 2017.

[33] D. Barry, Q. Zhang, P. W. Sun, and A. Hines, "Go listen: An end-to-end online listening test platform," *Journal of Open Research Software*, 2021.

[34] J. Turian, J. Shier, H. R. Khan, B. Raj, B. W. Schuller, C. J. Steinmetz, C. Malloy, G. Tzanetakis, G. Velarde, K. McNally, M. Henry, N. Pinto, C. Noufi, C. Clough, D. Herremans, E. Fonseca, J. H. Engel, J. Salamon, P. Esling, P. Manocha, S. Watanabe, Z. Jin, and Y. Bisk, "HEAR: holistic evaluation of audio representations," in *NeurIPS 2021 Competitions and Demonstrations Track, 6-14 December 2021, Online*, ser. Proceedings of Machine Learning Research, vol. 176. PMLR, 2021, pp. 125–145.

[35] R. Yuan, Y. Ma, Y. Li, G. Zhang, X. Chen, H. Yin, L. Zhuo, Y. Liu, J. Huang, Z. Tian, B. Deng, N. Wang, C. Lin, E. Benetos, A. Ragni, N. Gyenge, R. B. Dannenberg, W. Chen, G. Xia, W. Xue, S. Liu, S. Wang, R. Liu, Y. Guo, and J. Fu, "MARBLE: music audio representation benchmark for universal evaluation," in *Advances in Neural Information Processing Systems 36:*

Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023, 2023.

[36] P. Knees, Á. Faraldo, P. Herrera, R. Vogl, S. Böck, F. Hörschläger, and M. L. Goff, "Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections," in *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015*, 2015, pp. 364–370.

[37] 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.

[38] 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*. International Society for Music Information Retrieval, 2009, pp. 387–392.

[39] J. H. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with wavenet autoencoders," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, ser. Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 1068–1077.

[40] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *CoRR*, vol. abs/2005.00341, 2020.

[41] L. Wang, P. Luc, Y. Wu, A. Recasens, L. Smaira, A. Brock, A. Jaegle, J. Alayrac, S. Dieleman, J. Carreira, and A. van den Oord, "Towards learning universal audio representations," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2022, Virtual and Singapore, 23-27 May 2022*. IEEE, 2022, pp. 4593–4597.

[42] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," apr 2020.