Using a Joint-Embedding Predictive Architecture for Symbolic Music Understanding

Rafik Hachana
Artificial Intelligence Institute, Innopolis University, Russia
r.hachana@innopolis.university

Bader Rasheed
Laboratory of Innovative Technologies for Processing
Video Content, Innopolis University, Russia
b.rasheed@innopolis.university

Abstract

Despite the growing success of Joint Embedding Predictive Architectures (JEPA) in vision and speech, their potential for symbolic music representation learning remains unexplored. In this work, we adapt JEPA to the symbolic music domain by introducing music-specific masking strategies and combining different regularization techniques. The model achieves competitive results with much less training compute on a few downstream tasks, while falling short on a few others. This is mainly attributed to a bias towards positional information in the learned representation.

1 Introduction

Representation learning for symbolic music is a catalyzer for all music machine learning applications. It aids in both classification and generation downstream tasks, and extends to the interpretability of music machine processing in both symbolic [17] and audio formats. Practical applications range from compositional tools to retrieval and recommendation systems. With existing studies using either hand-crafted [9, 15] or NLP-based methods [27, 26], there has been no application of Joint Embedding Predictive Architecture [14] in symbolic music. JEPA has the potential of learning rich representations with less resources, compared to reconstructive or contrastive methods [1].

We experiment with JEPA's capability of modeling symbolic music and discrete modalities in general, by implementing new masking methods and architectural changes and evaluating on several downstream tasks. Our main contribution lies in domain-specific masking methods and predictor latent variable usage, as well as the addition of regularization and domain-specific input representation to adapt JEPA to symbolic music.

2 Related Work

2.1 Joint Embedding Predictive Architectures

Joint Embedding Predictive Architecture (JEPA) is a self-supervised framework that learns representations by predicting directly in embedding space. It consists of a context encoder, target encoder, and predictor that estimates target representations from context and a latent variable [14]. I-JEPA [1] introduced this approach for images, using image patches as context/targets and positional encodings as latent variables. Extensions followed for videos [3, 4] and audio spectrograms [6, 22]. Studies

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: AI for Music.

show JEPA emphasizes slow, low-frequency features [24, 16], while variance regularization improves performance in images and videos [18, 5, 2]. Beyond graphs [23], JEPA has not been applied to text or other discrete modalities.

2.2 Symbolic Music Processing in Machine Learning

Symbolic music represents notes and attributes as discrete events, commonly stored in MIDI format [17]. For machine learning, it is modeled as token sequences, typically with Transformers [8]. Adaptations include relative position embeddings [11] and content-based encodings [7]. Tokenization methods vary: MIDI-like [19], REMI [12], REMI+ [25], CompoundWord [10], and OctupleMIDI [27], with the latter reducing sequence length by grouping attributes.

Representation learning approaches for symbolic music largely adapt NLP methods. MusicBERT [27] achieves strong classification results via masked token prediction. Autoregressive pretraining [25, 20] enables generation but is computationally expensive, while handcrafted representations risk bias [15]. To date, JEPA or similar predictive embedding frameworks remain unexplored in symbolic music, motivating this work.

3 Methodology

The goal of our experiment is to adapt JEPA to symbolic music in both architecture and training methodology, and evaluate the effect of the learned representation on downstream tasks.

3.1 Training data and input representation

For self-supervised training, we use the LakhMIDI dataset [21] and follow MusicBERT's data pipeline with OctupleMIDI tokenization [27]. Each note is represented by eight tokens—Bar, Onset, Pitch, Velocity, Duration, Instrument, Time Signature, and Tempo—making sequence lengths multiples of eight. We fuse every eight tokens into a single embedding vector $\mathrm{Oct}(x_i)$, reducing sequence length and expanding context. We design seven masking strategies to generate context–target pairs for JEPA pretraining, including segmental, random, and musically informed masks (see Appendix B).

Before encoding the segment, we fuse the input with the Fundamental Music Embedding (FME) based on the Pitch and Duration attributes [7]. The FME is a content-based positional encoding, which differs from the index-based positional encoding $PE(x_i)$ (with $i \in \{1, \ldots, N\}$, N is the training batch size).

$$\mathrm{Oct}(x_i) = \mathbf{W}_O[E[x_{i,1}]; \ldots; E[x_{i,8}]]$$
$$\mathrm{Emb}(x_i) = \mathbf{W}_E[\mathrm{Oct}(x_i); \mathrm{FME}_p(x_i); \mathrm{FME}_d(x_i)] + \mathrm{PE}(x_i)$$

3.2 JEPA architecture

JEPA [14] learns data representations by directly predicting representation, rather than reconstructing data or using contrastive samples. During training, we use two encoders with the same architecture (f) to encode the context music segment, and the target segment. Then we predict the representation of the target from the representation of the context using a predictor model g.

Since the context representation can lack crucial information about the target (e.g. what instrument was masked), we pass a latent variable $\mathbf{z}_i$ to the predictor. Similar to Assran *et al.* [1], we use the sinusoidal positional encoding of the target tokens as a latent variable. For our own Instrument and Transpose masking methods, we also add the information about the masked instrument and the pitch shift to $\mathbf{z}_i$ respectively.

All models use a Transformer architecture with relative key-query attention [13] and 8 attention heads. All latent vectors have dimension $d = 512$. The encoders have 16 layers and the predictor is a Transformer decoder with 2 layers.

3.3 Mitigating representation collapse

JEPA pretraining could lead to representation collapse i.e. the encoders project all input to the same point or to a trivial subspace. We use two techniques to mitigate this.

First, we don't back-propagate the loss gradient through the target encoder $f_{\theta^{\text{EMA}}}$ [1]. Instead we update its weights through an exponential moving average of the context encoder f_θ weights.

$$\mathbf{x}_i = f_\theta\big(\,\mathrm{Emb}(x_{\text{context},\,i})\big)$$
$$\mathbf{y}_i = \mathrm{sg}\Big(f_{\theta^{\text{EMA}}}\big(\,\mathrm{Emb}(x_{\text{target},\,i})\big)\Big) \in \mathbb{R}^d, \quad \mathrm{sg}\text{ is the StopGradient operation.}$$
$$\text{with} \quad \theta^{\text{EMA}} \leftarrow \rho\,\theta^{\text{EMA}} + (1-\rho)\,\theta \quad \text{(EMA update; no gradients through } \mathbf{y}_i\text{).}$$

Second, we use a training loss that combines a cosine regression loss $\mathcal{L}_{\cos}$ on the predictor output $\hat{\mathbf{y}}_i$ and target $\mathbf{y}_i$, and a VICReg-inspired [2] variance and covariance regularization terms.

$$\mathcal{L} = \alpha \underbrace{\frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{\langle \hat{\mathbf{y}}_i, \mathbf{y}_i\rangle}{\|\hat{\mathbf{y}}_i\|\,\|\mathbf{y}_i\|}\right)}_{\mathcal{L}_{\cos}} + \beta \underbrace{\frac{1}{d}\sum_{j=1}^{d}\max\left(0,\,\gamma - \sqrt{\mathrm{Var}(\hat{\mathbf{y}}_{\cdot j})}\right)}_{\mathcal{L}_{\text{var}}} + \sigma \underbrace{\frac{1}{d}\sum_{i\neq j}\big(\mathrm{Cov}(\hat{\mathbf{y}}_{\cdot i}, \hat{\mathbf{y}}_{\cdot j})\big)^2}_{\mathcal{L}_{\text{cov}}}$$

3.4 Downstream tasks

We evaluate the learned representation by training a classification head (comprised of attentive pooling and fully-connected layers) on several classification tasks:

- **Melody Prediction**: Predict if a melody B fits as a continuation of melody A. We construct the melody pairs from the LakhMIDI dataset by splitting input sequences into two halves. We create a training dataset of positive and negative pairs by pairing consecutive and non-consecutive sequences respectively. Even though the task itself is a binary classification task, we evaluate it as a retrieval task, similar to MusicBERT [27]. In evaluation, we create 50 pairs with the same starting melody and different continuations, with only one positive pair. We then use the predicted positive label probability to rank results and evaluate them using Mean Average Precision (MAP) and HITS@k (k=1, 5, 10, 25, showing the rate of correct continuations in the top k candidates).
- **Accompaniment Suggestion**: Similar to melody completion, we compare a melody to possible accompaniments (instrumental tracks that can be played along with the melody). We construct the samples ourselves and use the same evaluation method as Melody Prediction. The only difference is that here, we construct melody-accompaniment pairs from dataset sequences by identifying the melody (the highest pitch instrument track), and separating it from the rest of the instruments in the sequence.
- **Genre and Style Classification**: We use the TOPMAGD and MASD datasets[27, 15] to train a multi-label classifier to predict music genre and style respectively. We evaluate both tasks using the micro-F1 score.

4 Results

We trained the model on the full LakhMIDI dataset using an Nvidia RTX3090 (24GB VRAM) for 5 epochs, taking 2 days, 18 hours, and 53 minutes. Training tracked cosine regression and VICReg losses (Figure 2), which converged with oscillations in opposite phases, reflecting competing objectives.

To inspect learned representations, we projected samples from different songs (Figure 1). Sequence vectors formed string-like curves per song, suggesting positional encodings map into latent space. However, the encoder does not collapse to pure positional encoding, since segments from different songs occupy distinct latent regions.

Fine-tuning yielded state-of-the-art performance on melody completion and accompaniment suggestion, surpassing MusicBERT [27] on several metrics (Table 1). The model excels at approximating good melody continuations but struggles with final candidate selection, while accompaniment shows the opposite trend. Genre and style classification results were weaker, with macro-F1 scores of 0.666 and 0.261, comparable to handcrafted features like pianoroll and tonnetz (Appendix A).

We attribute this task discrepancy to a positional bias: the latent space encodes melodies as continuous curves, aiding melody continuation but limiting generalization in other tasks.

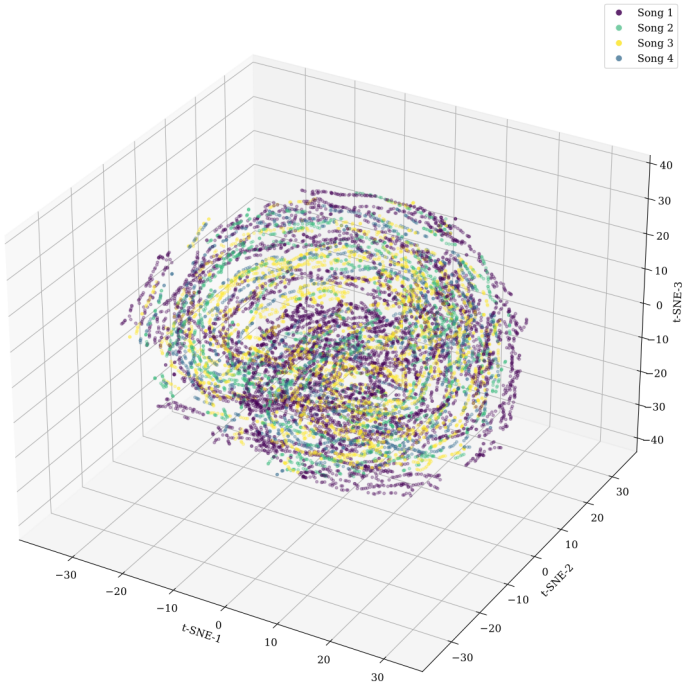


Figure 1: Visualizations of the learned representations of songs (Using t-SNE for dimensionality reduction)

Table 1: Results on melody prediction and accompaniment suggestion.

Model	**Melody prediction**					**Accompaniment suggestion**				
	MAP	H@1	H@5	H@10	H@25	MAP	H@1	H@5	H@20	H@25
`tonnetz`	0.683	0.545	0.865	0.946	0.993	0.423	0.101	0.407	0.628	0.897
`pianoroll`	0.762	0.645	0.916	0.967	0.995	0.567	0.166	0.541	0.720	0.921
$\texttt{PiRhDy}_{\texttt{GH}}$	0.858	0.775	0.966	0.988	0.999	0.651	0.211	0.625	0.812	0.965
$\texttt{MusicBERT}_{\texttt{base}}$	**0.985**	**0.975**	**0.997**	**0.999**	**1.000**	**0.946**	0.333	**0.857**	**0.996**	**0.998**
`JEPA`	0.856	0.743	**0.996**	**1.000**	**1.000**	0.506	**0.361**	0.758	0.895	0.957

5 Current Limitations and Future Work

Our results remain below state-of-the-art music embeddings, with a clear positional bias evident in visualizations and downstream tasks. Future work should incorporate stronger regularization (e.g., InfoNCE, hierarchical JEPA) to enrich representations, and ablation studies are needed to isolate the effects of key interventions (JEPA pretraining, OctupleMIDI tokenization, FME embedding) (Appendix C). A reliable JEPA music representation could also enable generative applications.

6 Conclusion

While achieving competitive results on some metrics with far less compute than MusicBERT, our symbolic music JEPA model underperformed on few downstream tasks due to positional encoding bias. We believe there is more potential to uncover with future experiments, in both performance and resource optimization. Future iterations will focus on latent space regularization and systematic ablations to clarify contributions and improve performance.

Acknowledgement

The research was supported by the Ministry of Economic Development of the Russian Federation (agreement No. 139-10-2025-034 dd. 19.06.2025, IGK 000000C313925P4D0002)

References

[1] Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15619–15629, Vancouver, BC, Canada, June 2023. IEEE. ISBN 979-8-3503-0129-8. doi: 10.1109/CVPR52729.2023.01499. URL https://ieeexplore.ieee.org/document/10205476/.

[2] Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning, January 2022. URL http://arxiv.org/abs/2105.04906. arXiv:2105.04906 [cs].

[3] Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat, Yann LeCun, Mido Assran, and Nicolas Ballas. V-JEPA: Latent Video Prediction for Visual Representation Learning. October 2023. URL https://openreview.net/forum?id=WFYbBOEOtv&referrer=%5Bthe+profile+of+Xinlei+Chen%5D%28%2Fprofile%3Fid%3D~Xinlei_Chen1%29&utm_source=ainews&utm_medium=email&utm_campaign=ainews-companies-liable-for-ai-hallucination-is.

[4] Adrien Bardes, Jean Ponce, and Yann LeCun. MC-JEPA: A Joint-Embedding Predictive Architecture for Self-Supervised Learning of Motion and Content Features, July 2023. URL http://arxiv.org/abs/2307.12698. arXiv:2307.12698 [cs].

[5] Katrina Drozdov, Ravid Shwartz-Ziv, and Yann LeCun. Video Representation Learning with Joint-Embedding Predictive Architectures, December 2024. URL http://arxiv.org/abs/2412.10925. arXiv:2412.10925 [cs].

[6] Zhengcong Fei, Mingyuan Fan, and Junshi Huang. A-JEPA: Joint-Embedding Predictive Architecture Can Listen, January 2024. URL http://arxiv.org/abs/2311.15830. arXiv:2311.15830 [cs].

[7] Zixun Guo, Jaeyong Kang, and Dorien Herremans. A Domain-Knowledge-Inspired Music Embedding Space and a Novel Attention Mechanism for Symbolic Music Modeling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(4):5070–5077, June 2023. ISSN 2374-3468. doi: 10.1609/aaai.v37i4.25635. URL https://ojs.aaai.org/index.php/AAAI/article/view/25635. Number: 4.

[8] Rafik Hachana and Bader Rasheed. Probe-assisted fine-grained control for non-differentiable features in symbolic music generation. *IEEE Access*, 2025.

[9] Tatsunori Hirai and Shun Sawada. Melody2Vec: Distributed Representations of Melodic Phrases based on Melody Segmentation. *Journal of Information Processing*, 27(0):278–286, 2019. ISSN 1882-6652. doi: 10.2197/ipsjjip.27.278. URL https://www.jstage.jst.go.jp/article/ipsjjip/27/0/27_278/_article.

[10] Wen-Yi Hsiao, Jen-Yu Liu, Yin-Cheng Yeh, and Yi-Hsuan Yang. Compound Word Transformer: Learning to Compose Full-Song Music over Dynamic Directed Hypergraphs. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):178–186, May 2021. ISSN 2374-3468. doi: 10.1609/aaai.v35i1.16091. URL https://ojs.aaai.org/index.php/AAAI/article/view/16091. Number: 1.

[11] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. Music Transformer, December 2018. URL http://arxiv.org/abs/1809.04281. arXiv:1809.04281 [cs, eess, stat].

[12] Yu-Siang Huang and Yi-Hsuan Yang. Pop Music Transformer: Beat-based Modeling and Generation of Expressive Pop Piano Compositions, August 2020. URL http://arxiv.org/abs/2002.00212. arXiv:2002.00212 [cs, eess, stat].

[13] Zhiheng Huang, Davis Liang, Peng Xu, and Bing Xiang. Improve Transformer Models with Better Relative Position Embeddings, September 2020. URL http://arxiv.org/abs/2009.13658. arXiv:2009.13658 [cs].

[14] Yann LeCun. A Path Towards Autonomous Machine Intelligence Version 0.9.2, 2022-06-27.

[15] Hongru Liang, Wenqiang Lei, Paul Yaozhu Chan, Zhenglu Yang, Maosong Sun, and Tat-Seng Chua. PiRhDy: Learning Pitch-, Rhythm-, and Dynamics-aware Embeddings for Symbolic Music. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 574–582, October 2020. doi: 10.1145/3394171.3414032. URL http://arxiv.org/abs/2010.08091. arXiv:2010.08091 [cs].

[16] Etai Littwin, Omid Saremi, Madhu Advani, Vimal Thilak, Preetum Nakkiran, Chen Huang, and Joshua Susskind. How JEPA Avoids Noisy Features: The Implicit Bias of Deep Linear Self Distillation Networks, July 2024. URL http://arxiv.org/abs/2407.03475. arXiv:2407.03475 [cs].

[17] Gareth Loy. Musicians Make a Standard: The MIDI Phenomenon. *Computer Music Journal*, 9 (4):8–26, 1985. ISSN 0148-9267. doi: 10.2307/3679619. URL https://www.jstor.org/stable/3679619. Publisher: The MIT Press.

[18] Shentong Mo and Shengbang Tong. Connecting Joint-Embedding Predictive Architecture with Contrastive Self-supervised Learning, October 2024. URL http://arxiv.org/abs/2410.19560. arXiv:2410.19560 [cs].

[19] Sageev Oore, Ian Simon, Sander Dieleman, Douglas Eck, and Karen Simonyan. This time with feeling: learning expressive musical performance. *Neural Computing and Applications*, 32(4):955–967, February 2020. ISSN 1433-3058. doi: 10.1007/s00521-018-3758-9. URL https://doi.org/10.1007/s00521-018-3758-9.

[20] Yang Qin, Huiming Xie, Shuxue Ding, Benying Tan, Yujie Li, Bin Zhao, and Mao Ye. Bar transformer: a hierarchical model for learning long-term structure and generating impressive pop music. *Applied Intelligence*, August 2022. ISSN 1573-7497. doi: 10.1007/s10489-022-04049-3. URL https://doi.org/10.1007/s10489-022-04049-3.

[21] Colin Raffel. Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching.

[22] Alain Riou, Stefan Lattner, Gaëtan Hadjeres, and Geoffroy Peeters. Investigating Design Choices in Joint-Embedding Predictive Architectures for General Audio Representation Learning, May 2024. URL http://arxiv.org/abs/2405.08679. arXiv:2405.08679 [cs].

[23] Geri Skenderi, Hang Li, Jiliang Tang, and Marco Cristani. Graph-level Representation Learning with Joint-Embedding Predictive Architectures, June 2024. URL http://arxiv.org/abs/2309.16014. arXiv:2309.16014 [cs].

[24] Vlad Sobal, Jyothir S V, Siddhartha Jalagam, Nicolas Carion, Kyunghyun Cho, and Yann LeCun. Joint Embedding Predictive Architectures Focus on Slow Features, November 2022. URL http://arxiv.org/abs/2211.10831. arXiv:2211.10831 [cs].

[25] Dimitri von Rütte, Luca Biggio, Yannic Kilcher, and Thomas Hofmann. FIGARO: Generating Symbolic Music with Fine-Grained Artistic Control, March 2022. URL http://arxiv.org/abs/2201.10936. arXiv:2201.10936 [cs, eess, stat].

[26] Ziyu Wang and Gus Xia. MUSEBERT: PRE-TRAINING OF MUSIC REPRESENTATION FOR MUSIC UNDERSTANDING AND CONTROLLABLE GENERATION. page 8, 2021.

[27] Mingliang Zeng, Xu Tan, Rui Wang, Zeqian Ju, Tao Qin, and Tie-Yan Liu. MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training, June 2021. URL http://arxiv.org/abs/2106.05630. arXiv:2106.05630 [cs].

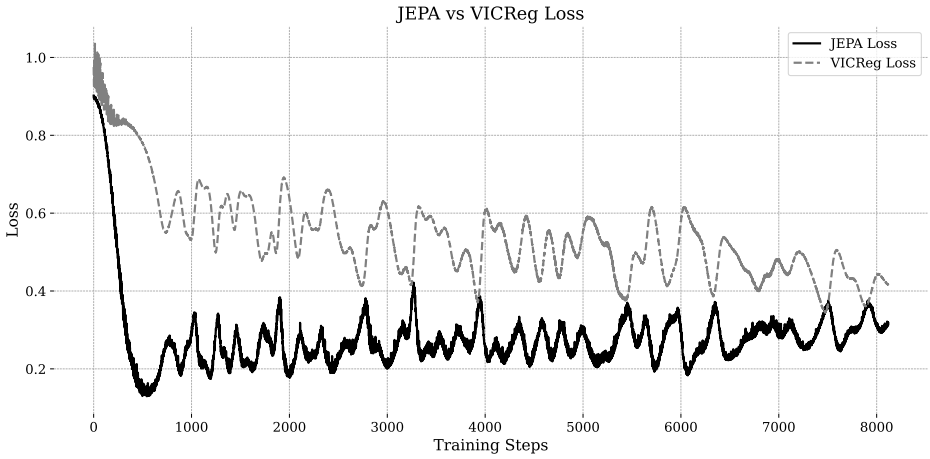


Figure 2: Evolution of the main loss and VICReg loss during training

A More downstream task results

We presented more results from baseline models in Tables 2,3, and 4. The extra data from the baselines is taken from the MusicBERT experiment [27], and we are incorporating it in our work for comparitive purposes.

Table 2: Melody prediction results

Model	MAP	H@1	H@5	H@10	H@25
melody2vec$_{F}$	0.646	0.578	0.717	0.774	0.867
melody2vec$_{B}$	0.641	0.571	0.712	0.772	0.866
tonnetz	0.683	0.545	0.865	0.946	0.993
pianoroll	0.762	0.645	0.916	0.967	0.995
PiRhDy$_{GH}$	0.858	0.775	0.966	0.988	0.999
PiRhDy$_{GM}$	0.971	0.950	0.995	0.998	0.999
MusicBERT$_{small}$	0.982	0.971	0.996	0.999	1.000
MusicBERT$_{base}$	**0.985**	**0.975**	**0.997**	**0.999**	**1.000**
JEPA	0.856	0.743	**0.996**	**1.000**	**1.000**

Table 3: Accompaniment suggestion results

Model	MAP	H@1	H@5	H@20	H@25
tonnetz	0.423	0.101	0.407	0.628	0.897
pianoroll	0.567	0.166	0.541	0.720	0.921
PiRhDy$_{GH}$	0.651	0.211	0.625	0.812	0.965
PiRhDy$_{GM}$	0.567	0.184	0.540	0.718	0.919
MusicBERT$_{small}$	0.930	0.329	0.843	0.993	0.997
MusicBERT$_{base}$	**0.946**	0.333	**0.857**	**0.996**	**0.998**
JEPA	0.506	**0.361**	0.758	0.895	0.957

B Masking methods for training

In order to train JEPA, we need to convert each training sample into a pair of context and target sequences. We use multiple masking methods and augmentations for this purpose. Firstly, we have two domain-agnostic methods:

- Segment masking: Similar to image patch masking in I-JEPA [1], we randomly mask a segment of the input to get the context. We pick a segment contained in the masked part to get the target.
- Random token masking: We simply mask input tokens randomly, with a probability $p = 0.2$. The target is the complement of the masked context.

And five domain-specific masking methods/augmentations:

- Mask an instrument: We mask all tokens belonging to a certain instrument.
- Pitch class: We mask all tokens of a certain pitch class e.g C, $E\flat$.
- Octave: We mask all notes within a certain octave i.e. pitch range.
- Rhythmic Noise: We add noise to the duration value of the notes. The target is the original sequence.
- Transposition: We transpose (shift) the pitch of the sequence. The target is the original sequence.

During training, we randomly pick a masking method for each training sample in the batch, with a uniform probability. Depending on the masking method used, we provide related info in the latent variable of JEPA's predictor.

C Preliminary ablation study

We have some preliminary ablation experiment results. The results are from a model trained on only 20% of the dataset (40000 MIDI files) for 2 epochs. We observe that the final learned embeddings have a different shape in the dimensionality reduction from the model trained on the full dataset (Figure 6). We also show the JEPA Loss, VICReg, and the average variance per vector dimension in figures 3, 5, and 4 respectively.

We train six models for the ablation:

- Baseline model with the VICReg loss term and absolute position encoding, and all the masking methods mentioned in Appendix B.
- Model without VICReg.
- Model without absolute position encoding
- Model without VICReg and with only the Contiguous segment masking method.

Table 4: Genre and style classification results

Model	Genre F1	Style F1
$\texttt{melody2vec}_{\texttt{F}}$	0.649	0.299
$\texttt{melody2vec}_{\texttt{B}}$	0.647	0.293
`tonnetz`	0.627	0.253
`pianoroll`	0.640	0.365
$\texttt{PiRhDy}_{\texttt{GH}}$	0.663	0.448
$\texttt{PiRhDy}_{\texttt{GM}}$	0.668	0.471
$\texttt{MusicBERT}_{\texttt{small}}$	0.761	0.626
$\texttt{MusicBERT}_{\texttt{base}}$	**0.784**	**0.645**
`JEPA`	0.666	0.267

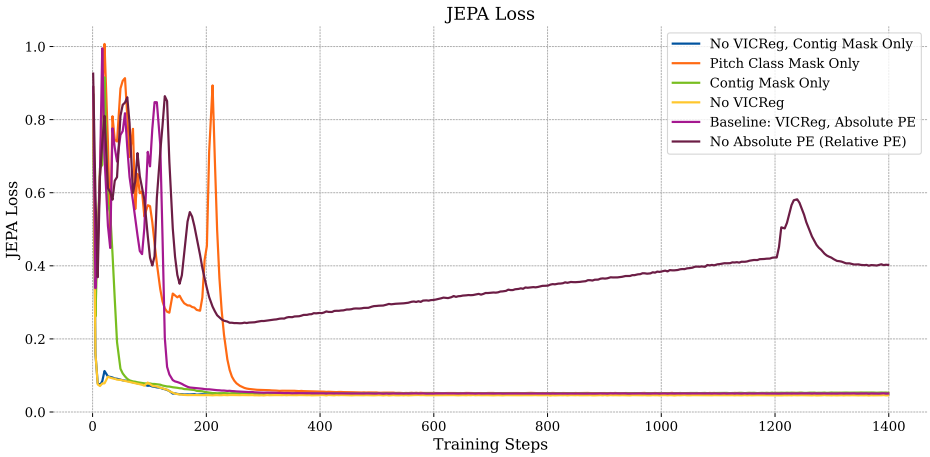


Figure 3: Effect of different ablations on the JEPA loss

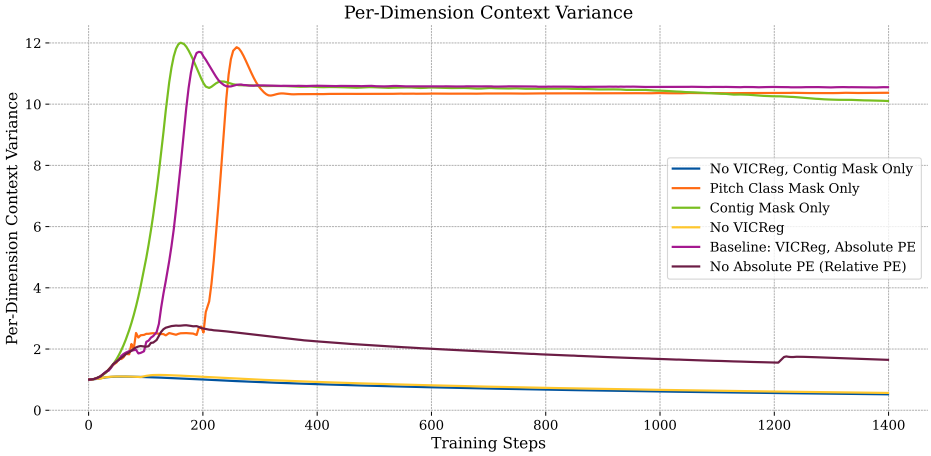


Figure 4: Effect of different ablations on the JEPA loss

- Model with only the Contiguous segment masking method.
- Model with only the Pitch class masking method.

We have the following conclusions from the current ablation study:

- Without VICReg, the VICReg loss stays high, the model converges faster to a lower JEPA loss but with a lower per-dimension variance.
- Without Absolute PE, the JEPA loss struggles to converge, and the per-dimension variance is much lower than usual. This could be explained by the model's tendency to collapse the representation when there is not much useful information present in the latent representation.
- When using only the contiguous masking method, the model converges faster. The latent representation shows a more localized cluster in the dimensionality reduction (Figure 6d), which is different from the multiple clusters or multiple clusters with points in the models trained on all masking methods.
- Eliminating VicReg and/or masking only contiguous segments (Figures 6c, 6f, and 6d) results in more defined clusters in latent space with less scattered points. Meanwhile masking only by pitch (Figure 6e) achieves the opposite effect: less defined clusters and more scattered points.

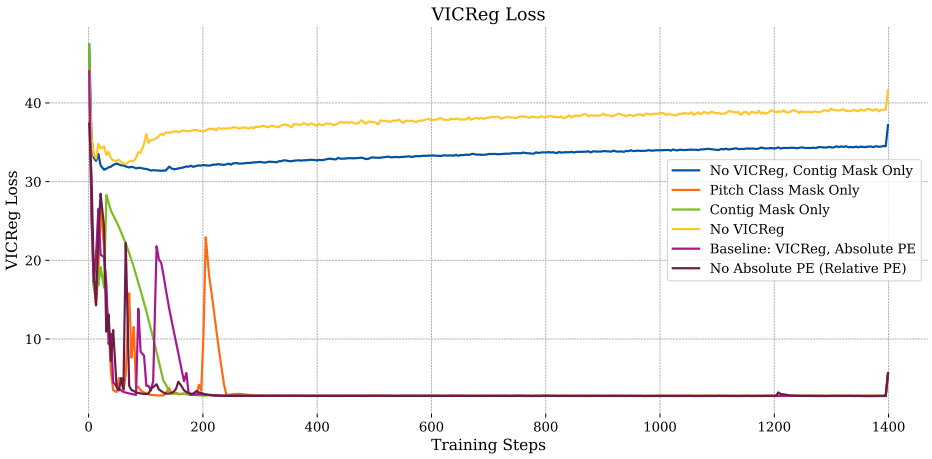
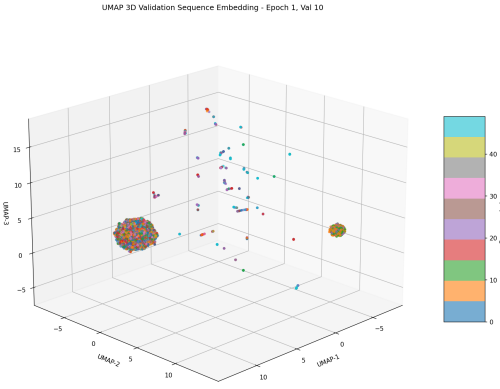
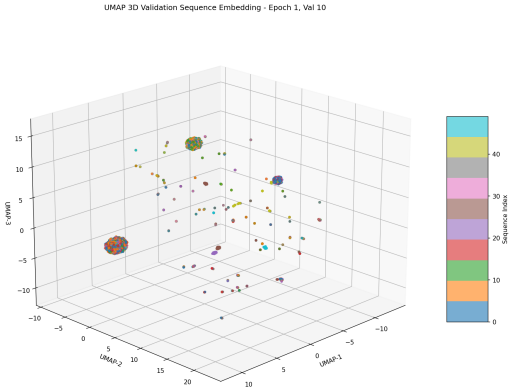


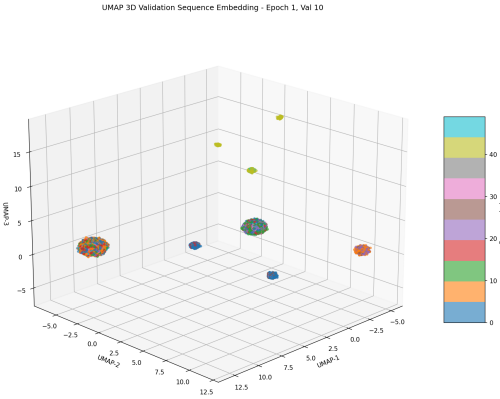
Figure 5: Effect of different ablations on the JEPA loss



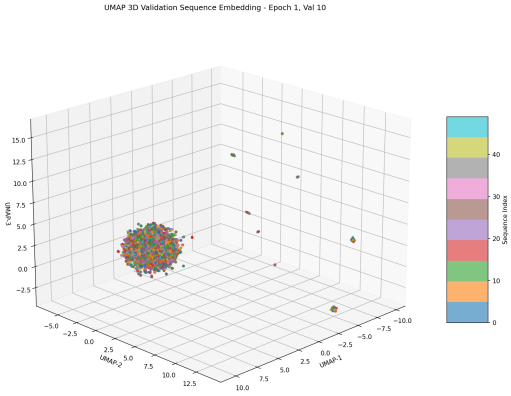
(a) Baseline: VICReg, Absolute PE



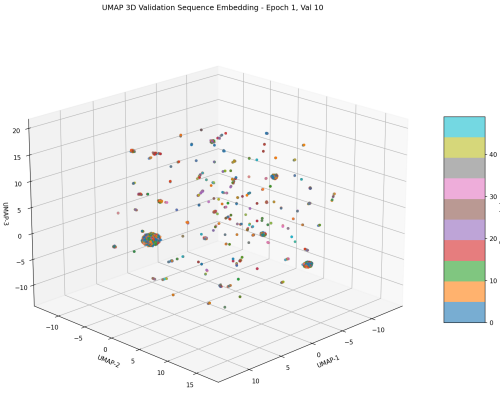
(b) No Absolute PE (Relative PE)



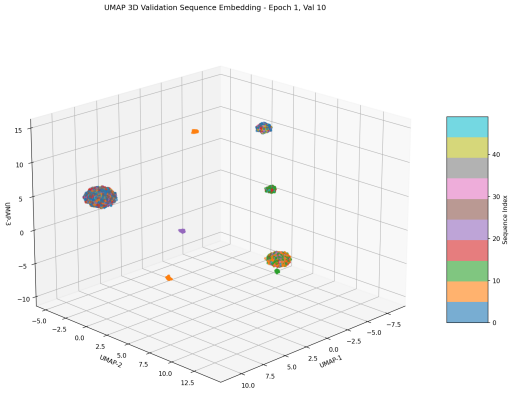
(c) No VICReg



(d) Contig Mask Only



(e) Pitch Class Mask Only



(f) No VICReg, Contig Mask Only

Figure 6: Comparison of latent space projections from different ablations