Self-Supervised Representation Learning with a JEPA Framework for Multi-instrument Music Transcription

Mary Pilataki[1], *Matthias Mauch*[2], *Simon Dixon*[1]

[1]Queen Mary University of London, London, UK [2]Apple, London, UK

***Abstract*—We demonstrate that the Joint-Embedding Predictive Architecture is effective for learning representations suitable for Music Information Retrieval tasks. Specifically, we explore its application to multi-instrument automatic music transcription, focusing on multi-pitch estimation and instrument recognition. We evaluate the learned representations across multiple settings: (1) finetuning a pretrained JEPA model with transcription supervision, (2) end-to-end training with transcription supervision, (3) training an instrument-aware transcriber on frozen JEPA embeddings and (4) training an instrument-agnostic transcriber on frozen JEPA embeddings. To assess the structure of the learned representations, we compute Calinski-Harabasz clustering scores with respect to pitch index, pitch class, instrument, and octave. We find that the representations learned by JEPA and its modified version (2), primarily capture instrument identity and pitch height information, rather than pitch class distinctions. Despite this, our results demonstrate promising transcription performance and highlight the potential of non-generative self-supervised learning for multi-instrument music transcription. Code and model configurations are available on GitHub.**[1]

1. INTRODUCTION

Multi-instrument automatic music transcription (MIAMT) is a core task in Music Information Retrieval (MIR). Its complexity arises from three key challenges: overlapping harmonics, polyphonic ambiguity, and the need to jointly solve several interrelated subtasks, including multi-pitch estimation (MPE), onset and offset detection, instrument recognition (IR), beat tracking, interpretation of expressive dynamics, and score typesetting [1]. Most research in automatic music transcription (AMT) focuses on a small subset of these tasks, often neglecting the broader challenges posed by real-world multi-instrumental music.

MIAMT approaches can be broadly categorized into instrument-agnostic and instrument-aware transcription. Instrument-agnostic models focus on transcribing pitch from multi-instrument audio without explicitly identifying instrument source [2]–[6]. While effective in capturing pitch and temporal information, these models are not designed to distinguish between timbral characteristics, limiting their utility in applications requiring structured representations such as multitrack MIDI or score generation. Instrument-aware models address this limitation by jointly estimating pitch and instrument source [7]–[11], requiring representations that encode both pitch and timbre. Recent models like MT3 [9], YourMT3 [10], and MR-MT3 [11] use transformer architectures to accommodate diverse instrument combinations, but rely heavily on supervised training, which restricts scalability due to the limited availability of annotated datasets.

Large pretrained generative models are increasingly used in MIR tasks, either as feature extractors [6], [12]–[14] or as foundations for transfer learning [15]. These approaches have improved performance across various tasks, including genre classification, key detection, emotion recognition, and melody transcription, often outperforming models trained from scratch with task-specific objectives [12], [13]. These benefits hold even when using simple and shallow downstream architectures, underscoring the value of high-quality representations.

[1]https://github.com/marypilataki/amt-jepa

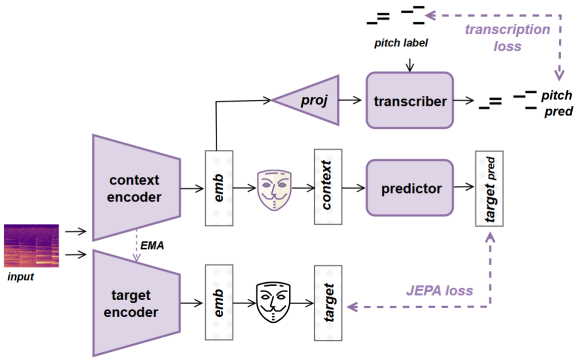

Fig. 1: MIAMT-JEPA framework: Unmasked embeddings pass through a projection layer (proj) to recover the time-frequency axis and then fed into the transcriber. When training with transcription supervision, the transcription loss is added to the JEPA objective. Target encoder weights are updated through an Exponential Moving Average (EMA) of the context encoder weights.

However, generative models typically focus on reconstructing waveform-level details, many of which are irrelevant for transcription. Self-supervised non-generative approaches, such as contrastive learning [16], [17] or Joint Embedding Predictive Architectures (JEPAs) [18], [19], aim to learn task-relevant representations without reconstruction overhead. Building on this idea, Stem-JEPA [19] pioneered JEPA-based architectures for MIR and demonstrated that they can capture timbral, harmonic, and rhythmic structure, suggesting a strong potential for MIAMT applications.

In this paper, we investigate whether JEPA [20] can learn musically meaningful representations suitable for transcription. We evaluate two shallow transcription probes trained on frozen JEPA embeddings: an instrument-agnostic transcriber and an instrument-aware variant that jointly predicts pitch and instrument activity, focusing on the MPE and IR subtasks. Furthermore, we adapt the architecture and training paradigm such that representation learning is aligned with transcription objectives. Our work is in line with previous work in generative pretraining such as [13], Sheet Sage [12] and PA-DAC [6]. However, our proposed representation learning method is not based on generative modeling and, following [6], [19], [20] all predictions are made in the representation space.

Our approach addresses critical challenges in MIAMT by leveraging self-supervised pretraining to enable the use of large-scale unlabeled music collections, mitigating reliance on scarce, costly annotations. Our results demonstrate that even a base JEPA Vision Transformer (ViT), pretrained for a few epochs, learns musically meaningful representations without complex architectures or large labeled datasets. This proof of concept underscores the potential of self-supervised learning for MIAMT, where the scarcity of annotated data hinders progress. It also opens promising directions for self-supervised music representation learning in broader MIR tasks.

2. MIAMT-JEPA FRAMEWORK

2.1. Architecture

The architecture builds on JEPAs [18]–[20], which learn representations by predicting features of masked target regions from visible context regions within the same input, with target features computed by a learned target encoder [20]. As depicted in Fig. 1, the proposed framework consists of four core components: a target encoder, a context encoder, a predictor and a multi-instrument transcriber. Given a spectrogram, the model is trained to predict representations of distinct target blocks from context blocks within the same spectrogram.

Patch embeddings are passed through a projection layer before being fed into the transcriber. This design encourages the encoder to focus on pitch and instrument-specific information while learning high-level music representations. It is inspired by the pitch conditioner of PA-DAC [6]. Masks are applied after creating the patch embeddings of the input spectrogram. Hence, the context and targets are actually masked embeddings that correspond to certain spectrogram regions. We adhere to the original JEPA formulation for the target and context encoders and the predictor; please refer to [20] for further details. The transcriber shares the same architecture as the instrument-aware probe described in Section 2.3.

The target encoder, a Vision Trasformer (ViT) base architecture [20], [21], processes mel spectrograms via a patch-based pipeline. First, the input spectrogram is divided into 256 non-overlapping patches of size $[8 \times 8]$ which are linearly projected into patch-level embeddings. To generate prediction targets *target*, four distinct masked regions are applied to the target embedding. Each mask is a rectangular block with an aspect ratio sampled between 0.75 (wide) and 1.5 (tall), covering 15–20% of the target feature map. This forces the model to learn from sparse visible regions.

The context encoder, identical in architecture to the target encoder, also processes spectrograms by dividing them into 256 non-overlapping $[8 \times 8]$ patches. These patches are embedded into a patch-level representation which is then masked to retain a large contextual region, leaving 85-100% of the feature map visible. This setup encourages the encoder to extract meaningful features for the predictor to infer missing target regions.

The predictor, a narrow Vision Transformer (ViT) [20], takes two inputs: context embeddings and positional mask tokens indicating the target patches to be predicted. For each target block, the predictor is conditioned on corresponding mask tokens representing its spatial location, allowing sequential prediction of multiple masked regions while maintaining consistent global context conditioning.

2.2. Input features

Mel spectrograms are computed from 44.1 kHz mono audio with 128 Mel bands, a 25 ms Hanning window, and 10 ms hop size. These features are generated via *torchaudio.compliance.kaldi.fbank* [22], utilizing HTK-style Mel scaling, no dithering, and no energy term [23]. They are obtained from 1.3-second audio segments with 0.3 s overlap, yielding $[1 \times 128 \times 128]$ spectrograms. All features are normalized by the dataset's mean and standard deviation.

2.3. Multi-instrument transcriber

We employ two shallow transcribers: an instrument-agnostic model and an instrument-aware variant. Both consist of two hidden linear layers and use sigmoid activation to output pitch probabilities. The agnostic model, inspired by PA-DAC's pitch conditioner [6], takes 128 input features, equal to the number of mel-frequency bins, and produces an output layer of 88 neurons, corresponding to pitch range A0–C8. The instrument-aware variant extends this to predict pitch activity per instrument class, with an output layer of $[N \times I] = [88 \times 11]$ neurons, where N is the number of pitches and I the number of instrument classes.

Pitch probabilities are predicted from latent embeddings. As illustrated in Fig. 1, these embeddings are extracted by passing unmasked patch-level representations, derived from 1.3-second audio segments, through the frozen context encoder.

Embeddings have dimensionality $[B \times \frac{W_s \times H_s}{W_p \times H_p} \times D]$, where B denotes the batch size, W_s the spectrogram width (time axis), H_s the spectrogram height (frequency axis) and, similarly, W_p and H_p are the patch width and height respectively. D denotes the embedding dimension. For our configuration, this corresponds to $[B \times (\frac{128}{8})^2 \times 768] = [B \times 256 \times 768]$, since both the spectrograms and patches are square with $W_s = H_s = 128$ and $W_p = H_s = 8$.

Embeddings are projected onto the spectrogram space via a transposed convolutional layer with a kernel size and stride matching the patch size. This yields an output of $[B \times 1 \times W_s \times H_s]$, ensuring a one-to-one correspondence between embedding values and spectrogram regions.

2.4. Instrument vocabulary

Our model employs the MT3_MIDI_PLUS instrument vocabulary [10], but is trained specifically on 11 instrument classes, excluding singing voice and drums: Piano, Chromatic Percussion, Organ, Guitar, Bass, Strings, Brass, Reed, Pipe, Synth Lead, and Synth Pad.

3. EXPERIMENTAL SETUP

3.1. Datasets

For pretraining, we utilize a diverse collection of multi-instrument datasets: GTZAN [24], the Violin Bach dataset [25], Mazurkas[2], and the guitar dataset by Riley et al. [26]. Maintaining identical configurations and dataset splits as described in PA-DAC [6], this results to an approximately 159-hour dataset.

For finetuning, end-to-end training with transcription supervision, and probe training, we utilize the following datasets: Slakh [27], MusicNet's EM version [28], [29], GuitarSet [30] and URMP [31]. We adopt the official train and test splits from Slakh, while for the remaining datasets we follow the standardized configurations established by Chang et al. [10] and Gardner et al. [9], ensuring compatibility with recent state-of-the-art approaches in MIAMT. The MAESTRO dataset [32] is entirely unseen during training, with its test set used exclusively for evaluation.

3.2. Experiments

We first pretrain the baseline JEPA framework for 200 epochs. Then, we use the checkpoint of epoch 200 and we (1) finetune the model for a further 100 epochs with transcription supervision while jointly training the transcription head, (2) train from scratch end-to-end the JEPA framework with transcription supervision, (3) train an instrument-aware transcriber on frozen JEPA embeddings and (4) train an instrument-agnostic transcriber on frozen JEPA embeddings. Pretraining details and details regarding training variants (1-2) are described in Section 3.3 while probing details for (3-4) are described in Section 3.4.

3.3. Training details

The JEPA loss as introduced by Assran et al. [20], computes the average ℓ_2 distance between predicted and target patch-level representations, with the objective of minimizing their discrepancy in the embedding space.

[2]http://www.charm.rhul.ac.uk/index.html

$$\mathcal{L}_{\text{JEPA}} = \frac{1}{M} \sum_{i=1}^{M} \sum_{j \in B_i} \left\| s_{\hat{y}(j)} - s_y(j) \right\|_2^2 \quad (1)$$

As formalized in Equation (1), for M target blocks sampled from the input, the loss aggregates the squared ℓ_2 distances between target encoder's representations $s_y(j)$ and predictor outputs $\hat{s}_y(j)$ for all patches j within each target block B_i. This objective drives the self-supervised pretraining of the JEPA baseline model.

The transcription loss combines two components, a pitch loss and an onset loss. The pitch loss is the Binary Cross-Entropy (BCE) between target and predicted pitch probabilities. For instrument-aware transcription, we employ a weighted BCE loss to address the imbalance between active and silent frames. Frames with at least one active instrument are assigned a weight of 0.7, while silent frames receive a weight of 0.3. Similarly, the onset loss is computed using a weighted BCE between extracted onset labels (derived from piano rolls or multi-instrument labels), with a weighting of 0.9 for onset frames and 0.1 for non-onset frames to emphasize note transitions. The total transcription loss is defined as the sum of the pitch and onset loss components.

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{JEPA}} + \alpha \cdot \mathcal{L}_{\text{transcription}} \quad (2)$$

While in pretraining, only the JEPA loss is optimized (Equation (1)), variants (1-2) minimize a weighted sum of the JEPA and transcription losses, as shown in Equation (2) where α denotes the transcription loss weight. We set $\alpha = 10.0$ to prioritize the downstream task, as empirically, this value balanced transcription performance and gradient stability, allowing the transcription loss to dominate (absolute values differing by approximately 0.1–0.3) without disrupting feature learning

All models are optimized using AdamW [33] with batch size 64. The learning rate follows a warmup schedule with linear increase from 0.0001 to 0.001 over 15 epochs, before decaying to 0.000001 via a cosine scheduler. Target encoder weights are updated via an Exponential Moving Average (EMA) of the context encoder's weights [20], a method proven beneficial for joint embedding architectures [18], [20], [34], [35].

3.4. Probing details

Following established practices in MIR research [6], [14], [36], we evaluate learned representations via downstream probing. A probe is a shallow classifier trained on frozen representations to assess their encoded task-relevant information [37].

We employ the frozen baseline JEPA encoder at epoch 200 as a feature extractor, training both instrument-aware and instrument-agnostic transcription probes for 100 epochs. The models are optimized using AdamW with a batch size of 64, learning rate of 0.0001, and weight decay of 0.0001. We use the the $\mathcal{L}_{\text{transcription}}$ loss from Equation (2), with the same weighting scheme discussed in Section 3.3. This approach allows us to assess the quality of the representations learned via self-supervised learning while isolating the probing task from the pretraining objective.

3.5. Evaluation details

We evaluate our models using three key metrics: note-wise F1-score, multi-instrument F1-score and instrument recognition F1-score. We also include comparisons with recent instrument-aware and instrument-agnostic models where results are available: MT3 [9], YourMT3+ [10], MR-MT3 [11], Basic Pitch [3], and PA-DAC [6]. We note that direct comparisons should be interpreted cautiously due to differences in experimental setups, model architectures, and instrument vocabularies across studies.

Table 1: Evaluation of MIAMT-JEPA variants. *Trainable* indicates which components are updated during training. *F* is the multi-instrument F1-score, except for the instrument-agnostic probe (marked *agn.*), where it reflects note-wise F1. *IR* denotes instrument recognition accuracy.

MIAMT-JEPA setting	*Trainable*		*GuitarSet*		*URMP*		*MusicNet*		*Slakh*	
	JEPA	*Probe*	*F*	*IR*	*F*	*IR*	*F*	*IR*	*F*	*IR*
(1) Finetune baseline	✓	✓	62.7	67.8	72.2	67.8	39.6	49.3	27.9	23.9
(2) End-to-end training	✓	✓	59.2	59.2	60.0	63.4	39.5	45.1	25.1	23.1
(3) Linear probe	✗	✓	63.5	68.0	76.8	64.5	41.2	49.4	45.1	24.0
(4) Linear probe (agn.)	✗	✓	70.1	-	84.3	-	54.2	-	52.3	-

Table 2: Baselines comparison across datasets. Numbers indicate multi-instrument F1-scores except for *Agnostic*, where the note-wise F1-score is reported. MAESTRO dataset is unseen by our models, in contrast to the rest of the baselines where it is included in training.

Dataset	*Instrument*	*YourMT3*	*MT3*	*MR-MT3*	*Basic Pitch*	*PA-DAC*	*JEPA Probe*
GuitarSet	*Guitar*	**91.7**	78.0	62.5	-	-	63.5
	Agnostic	-	**90.0**	-	77.6	49.4	70.1
URMP	*Ensemble*	68.0	-	-	-	-	**78.9**
	All	-	59.0	-	-	-	**76.8**
	Agnostic	81.8	77.0	-	-	-	**84.3**
MusicNet	*Strings*	**91.3**	-	-	-	-	47.9
	Winds	**83.5**	-	-	-	-	35.7
	All	-	31.0	-	-	-	**41.2**
	Agnostic	-	50.0	-	**62.3**	36.8	54.2
Slakh	*All*	**74.8**	57.7	62.5	-	-	45.1
	Agnostic	**84.6**	75.2	67.3	42.0	43.6	52.3
MAESTRO	*Piano*	**97.0**	94.9	-	-	-	48.1
	Agnostic	-	**96.0**	-	71.0	-	54.2

Note-wise and multi-instrument F1-scores are computed using the community-standard mir_eval package [38]. To compute note-level metrics, we convert frame estimates to note events by applying the post-processing method introduced by Bittner et al. [3]. The only modification we make is that we estimate onset times using our model's frame output. In that case, a threshold of 0.3 is used and detected events that are shorter than 10 frames ($\approx$ 100 ms) are removed. For further details on the post-processing method please refer to the original Basic Pitch paper [3]. Notes are considered correct if the pitch is within a quarter tone and the onset is within 50 ms. Regarding instrument-aware evaluation, we extend the note-matching criteria to require correct instrument classification, following the methodology of MT3 [9] and YourMT3+ [10]. This ensures comprehensive evaluation of both pitch and instrument recognition capabilities.

4. RESULTS

4.1. Transcription performance

Table 1 details our framework's transcription and IR performance, reporting multi-instrument F1-scores (except note-wise for the instrument-agnostic Probe (agn.)). The Linear Probe (variant (3)), trained on frozen JEPA features, achieves the strongest transcription performance. This demonstrates the effective transfer of pretrained representations, allowing the transcriber to focus on learning pitch and instrument mappings.

In contrast, finetuning with transcription supervision (variant (1)) shows slightly reduced transcription performance, likely due to representation drift during joint optimization. However, it exhibits comparable or improved IR compared to variant (3), attributed to superior instrument encoding (4.2). End-to-end training (variant (2)) yields the weakest results, as joint optimization from scratch introduces competing objectives.

The superior performance of frozen JEPA features indicates that self-supervised pretraining learns musically meaningful representations with minimal adaptation for transcription. Preserving the pretrained feature space allows the transcriber to focus exclusively on learning instrument and pitch mappings. On the other hand, finetuning seems to introduce instability. While the joint objective in variant (1) is still useful for the transcription task, it gradually degrades the pretrained feature space. End-to-end training, variant (2), struggles further, as neither JEPA nor the transcriber receive stable signals during early training.

The limitations of variant (2) stem partly from its dependence on limited supervised data. In contrast, JEPA pretraining benefits from a much larger unannotated corpus, enabling the encoder to generalize from a broader and more diverse set of musical content. Despite this difference, Table 2 reveals promising generalization capabilities. On the completely unseen MAESTRO dataset, our models perform comparably to in-domain results, suggesting that JEPA representations capture some aspects of music that are transferable across datasets. Notably, on URMP, frozen JEPA *Probe* exceeds the performance of MT3 and YourMT3 despite using simpler architectures and no data augmentation or class balancing.

In addition to overall performance differences across training paradigms, transcription quality is notably lower for underrepresented instruments, due to limited training examples and lack of class-balancing strategies. Addressing this remains an important future work direction.

While performance is limited by factors like class imbalance for underrepresented instruments and optimization constraints requiring longer pretraining times and targeted augmentation, our work highlights self-supervised pretraining as a viable solution to the fundamental challenge of MIAMT research: severe shortage of annotated data. It significantly reduces annotation needs while delivering usable results. This will allow future work to rather focus on the refinement of representations and model architectures.

4.2. Learned embeddings

To understand the structure and semantic content of learned representations, we evaluate latent embeddings using the Calinski–Harabasz Index (CHI) [39], an unsupervised metric for cluster compactness and separation. The higher the score, the more compact and well-separated the clusters are. We compute CHI scores for pitch index, pitch class (chroma), octave (pitch height), and instrument class (timbre), extracting embeddings from the monophonic NSynth dataset [40]. As shown in Figure 2, the encoder develops stronger clustering along timbre and pitch height across all variants, while pitch-related attributes remain less prominent, even with transcription supervision. The T-SNE visualization of variant's (1) latent space, shown in Fig. 3, supports these insights, showing distinct instrument clusters (left) while pitch height is mostly organized from lower to higher on the y axis (darker to brighter colors on the right plot).

This implies that JEPA prioritizes stable macro-level features like timbre and register over transient pitch information. When finetuning the baseline with transcription supervision (variant (1)), instrument clustering scores increase significantly while octave information remains stable, confirming JEPA's focus on high-level representation learning. In contrast, end-to-end training (variant (2)) maintains relatively high but reduced timbre and octave scores compared to other variants (though still higher than pitch attributes), with no significant improvement in pitch or chroma clustering. This implies joint training preserves some macro-feature learning but fails to enhance encoding of pitch information.

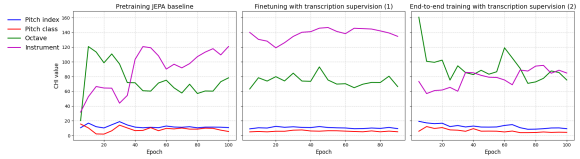

Fig. 2: Calinski–Harabasz index (CHI) of learned embeddings with respect to pitch index, pitch class, instrument and octave. From left to right scores correspond to the latent space of the pretrained JEPA baseline, variant (1) (finetuning with supervision) and variant (2) (end-to-end training with supervision).

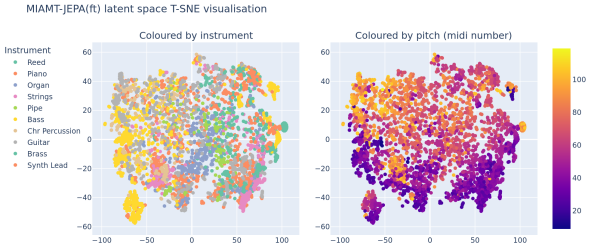

Fig. 3: Latent space t-sne visualisation coloured by instrument (left) and midi number (right).

The consistent strength in timbre and register representations across variants, combined with weaker pitch encoding, likely stems from JEPA's time-frequency masking strategy. Masking both time and frequency regions encourages the model to predict future or past contexts rather than concurrent feature relationships, favoring time-stable attributes (timbre, octave) over transient ones (pitch). Future work could explore alternative masking approaches, such as frequency-only masking, to better balance pitch and timbre encoding by encouraging pitch-harmony learning while maintaining temporal stability.

5. CONCLUSION

Our work demonstrates that self-supervised learning with JEPA is a promising direction for MIAMT, especially under limited annotation. JEPA learns musically meaningful representations, enabling competitive transcription with minimal supervision and directly addressing the field's bottleneck of annotation scarcity.

A key finding is that decoupling representation learning from task-specific objectives improves performance. Frozen JEPA features consistently outperform jointly trained variants, suggesting that task-specific gradients can degrade the quality of learned embeddings.

Latent space analysis shows that JEPA emphasizes stable attributes like timbre and octave over transient pitch details, likely due to its time-frequency masking. Improving temporal precision could help capture finer pitch dynamics and harmony relationships while retaining long-range dependencies.

Future research should explore masking strategies that enhance both temporal accuracy and spectral structure, and further disentangle pitch and instrument representations. Benchmarking against large-scale generative models will also help contextualize JEPA's capabilities. By refining how JEPA encodes musical time and leveraging unlabeled audio, we move closer to transcription systems that are accurate and generalizable, allowing researchers to focus on model design rather than costly data collection and annotation.

6. ACKNOWLEDGMENT

This work is supported by the EPSRC UKRI Centre for Doctoral Training in Artificial Intelligence and Music (Grant no. EP/S022694/1).

REFERENCES

[1] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[2] Y. T. Wu, B. Chen, and L. Su, "Multi-instrument automatic music transcription with self-attention-based instance segmentation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, 2020.

[3] R. M. Bittner, J. J. Bosch, D. Rubinstein, G. Meseguer-Brocal, and S. Ewert, "A lightweight instrument-agnostic model for polyphonic note transcription and multipitch estimation," in *Proc. ICASSP*, 2022.

[4] K. W. Cheuk, D. Herremans, and L. Su, "Reconvat: A semi-supervised automatic music transcription framework for low-resource real-world data," in *Proc. ACM Multimedia*, 2021.

[5] F. Cwitkowitz, K. W. Cheuk, W. Choi, M. A. Martínez-Ramírez, K. Toyama, W.-H. Liao, and Y. Mitsufuji, "Timbre-trap: A low-resource framework for instrument-agnostic music transcription," in *Proc. ICASSP*, 2024.

[6] M. Pilataki, M. Mauch, and S. Dixon, "Pitch-aware generative pretraining improves multi-pitch estimation with scarce data," in *Proc. ACM Multimedia Asia*, 2024.

[7] K. W. Cheuk, K. Choi, Q. Kong, B. Li, M. Won, A. Hung, J.-C. Wang, and D. Herremans, "Jointist: Joint learning for multi-instrument transcription and its applications," 2022.

[8] A. Jaegle, F. Gimeno, A. Brock, A. Zisserman, O. Vinyals, and J. Carreira, "Perceiver: General perception with iterative attention," *ArXiv*, vol. abs/2103.03206, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:232110866

[9] J. Gardner, I. Simon, E. Manilow, C. Hawthorne, and J. Engel, "MT3: Multi-task multitrack music transcription," in *Proc. ICLR*, 2021.

[10] S. Chang, E. Benetos, H. Kirchhoff, and S. Dixon, "YourMT3+: Multi-instrument music transcription with enhanced transformer architectures and cross-dataset stem augmentation," in *Proc. MLSP*, 2024.

[11] H. H. Tan, K. W. Cheuk, T. Cho, W.-H. Liao, and Y. Mitsufuji, "MR-MT3: Memory retaining multi-track music transcription to mitigate instrument leakage," 2024. [Online]. Available: https://arxiv.org/abs/2403.10024

[12] C. Donahue, J. Thickstun, and P. Liang, "Melody transcription via generative pre-training," in *Proc. ISMIR*, 2022.

[13] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," in *Proc. ISMIR*, 2021.

[14] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin, C.-L. Lin, A. Ragni, E. Benetos, N. Gyenge, R. B. Dannenberg, R. Liu, W. Chen, G. G. Xia, Y. Shi, W.-F. Huang, Y.-T. Guo, and J. Fu, "Mert: Acoustic music understanding model with large-scale self-supervised training," in *Proc. ICLR*, 2024.

[15] Z. Wang and G. Xia, "MuseBERT: Pre-training music representation for music understanding and controllable generation," in *Proc. ISMIR*, 2021.

[16] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," in *Proc. ISMIR*, 2021.

[17] J. Guinot, E. Quinton, and G. Fazekas, "Semi-supervised contrastive learning of musical representations," in *Proc. ISMIR*, 2024.

[18] D. Niizumi, D. Takeuchi, Y. Ohishi, N. Harada, and K. Kashino, "Masked modeling duo: Towards a universal audio pre-training framework," *IEEE Trans. Audio, Speech, Lang. Process.*, 2024.

[19] A. Riou, S. Lattner, G. Hadjeres, M. Anslow, and G. Peeters, "Stem-JEPA: A joint-embedding predictive architecture for musical stem compatibility estimation," in *Proc. ISMIR*, 2024.

[20] M. Assran, Q. Duval, I. Misra, P. Bojanowski, P. Vincent, M. Rabbat, Y. LeCun, and N. Ballas, "Self-supervised learning from images with a joint-embedding predictive architecture," in *Proc. CVPR*, 2023.

[21] A. Kolesnikov, A. Dosovitskiy, D. Weissenborn, G. Heigold, J. Uszkoreit, L. Beyer, M. Minderer, M. Dehghani, N. Houlsby, S. Gelly, T. Unterthiner, and X. Zhai, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[22] J. Hwang, M. Hira, C. Chen, X. Zhang, Z. Ni, G. Sun, P. Ma, R. Huang, V. Pratap, Y. Zhang, A. Kumar, C.-Y. Yu, C. Zhu, C. Liu, J. Kahn, M. Ravanelli, P. Sun, S. Watanabe, Y. Shi, and Y. Tao, "Torchaudio 2.1: Advancing speech recognition, self-supervised learning, and audio processing components for pytorch," in *Proc. ASRU*, 2023.

[23] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. K. Goel, M. Hannemann, P. Motlícek, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, 2011. [Online]. Available: https://api.semanticscholar.org/CorpusID:1774023

[24] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Audio, Speech, Lang. Process.*, 2002.

[25] H.-W. Dong, C. Zhou, T. Berg-Kirkpatrick, and J. McAuley, "Deep performer: Score-to-audio music performance synthesis," in *Proc. ICASSP*, 2022.

[26] X. Riley, D. Edwards, and S. Dixon, "High resolution guitar transcription via domain adaptation," in *Proc. ICASSP*, 2024.

[27] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, "Cutting music source separation some slakh: A dataset to study the impact of training data quality and quantity," *Proc. WASPAA*, 2019. [Online]. Available: http://dx.doi.org/10.1109/WASPAA.2019.8937170

[28] J. Thickstun, Z. Harchaoui, and S. Kakade, "Learning features of music from scratch," in *Proc. ICLR*, 2017.

[29] B. Maman and A. H. Bermano, "Unaligned supervision for automatic music transcription in the wild," in *Proc. ICML*, 2022.

[30] Q. Xi, R. M. Bittner, J. Pauwels, X. Ye, and J. P. Bello, "Guitarset: A dataset for guitar transcription," in *Proc. ISMIR*, 2018.

[31] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, "Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications," *IEEE Trans. Multimedia*, vol. 21, no. 2, pp. 522–535, 2019.

[32] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *Proc. ICLR*, 2019.

[33] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2019.

[34] D. Morales-Brotons, T. Vogels, and H. Hendrikx, "Exponential moving average of weights in deep learning: Dynamics and benefits," *arXiv preprint arXiv:2411.18704*, 2024.

[35] M. Caron, H. Touvron, I. Misra, H. Jegou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," in *Proc. ICCV*, 2021.

[36] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[37] G. Alain and Y. Bengio, "Understanding intermediate layers using linear classifier probes," in *Proc. ICLR*, 2016.

[38] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "mir eval: A transparent implementation of common MIR metrics," in *Proc. ISMIR*, 2014.

[39] T. Caliński and J. H. and, "A dendrite method for cluster analysis," *Communications in Statistics*, vol. 3, no. 1, pp. 1–27, 1974. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/03610927408827101

[40] J. 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*, ser. Proceedings of Machine Learning Research, K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, Eds. PMLR, 2017.