Helping Music Co-Creation Agents 'Listen' Well: Hierarchical Self-Supervised World Models for Understanding and Generation

Scott H. Hawley
Department of Chemistry & Physics
Belmont University
Nashville, TN, USA
scott.hawley@belmont.edu

Abstract

Collaborative music agents need internal representations rich enough to support both understanding and generation, yet flexible enough for a workflow where the human retains agency. We present a hierarchical self-supervised "world model" for symbolic music: a 2.55M-parameter Swin V2 encoder trained on MIDI piano-roll images with JEPA-style objectives (pitch- and time-shift equivariance, masked embedding prediction, and a distributional regularizer), using no labels and no music-theory vocabulary. Probing the frozen embeddings shows that the level at which a musical property becomes decodable tracks its musical time scale: phrase boundaries are read off the coarsest levels, note density and harmonic detail off the finest. Temporal and phrase structure emerge from the self-supervised objectives alone, while harmonic content must be asked for; a small chord-supervision head raises joint chord recovery from .18 to .54, and key detection, which is never supervised, from .16 to .70. Following the Representation AutoEncoder paradigm, a conditional flow-matching model stands in for a trained decoder, flowing in pixel space from PCA-reduced conditioning: it reproduces a target window at pixel F1 0.996, and the same per-level conditioning dropout that controls how far variations stray also enables graphical prompting for masked inpainting with no inpainting-specific sampler. The pipeline runs on CPU, producing a suggestion in 2.8 s, or 0.6 s on Apple MPS, which we demonstrate in a live interactive demo. In concert with an LLM-based brain, these capabilities supply the core of a collaborative music creation agent in service of, rather than in place of, human agency.

1 Paradigm: "An AI Rick Rubin"

Preserving human agency in the "age of AI" is a widespread concern particularly in the creative arts. This paper describes a system built to serve that goal, using small models and human domain expertise to offer feedback and suggestions in a collaborative songwriting and production loop. The metaphor of an "AI Rick Rubin" captures the approach, after the producer who described his method in a 2023 interview with Anderson Cooper:

> **Cooper:** Do you play instruments?
> **Rubin:** Barely… I have no technical ability. And I know nothing about music.
> **Cooper:** You must know something.
> **Rubin:** Well, I know what I like and what I don't like. And I'm decisive about what I like and what I don't like… The confidence that I have in my taste and my ability to express what I feel has proven helpful for artists. [32]

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

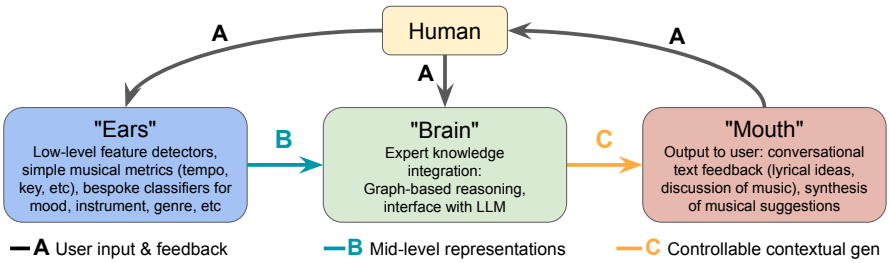

Figure 1: The Ears/Brain/Mouth workflow for a conversational music co-writing agent. This paper develops the Ears and Mouth: perceptual representations (Section 3) rich enough to drive controllable generation of music suggestions (Section 4) without task-specific fine-tuning. The Brain, an expert-knowledge reasoning layer, is beyond the scope of this paper.

Our system listens carefully to the artist's musical ideas, processes them according to its own internal representations, and responds with verbal feedback and limited (generative) musical suggestions. The AI never "does it for" the human: suggestions are left for the human to implement. In short, it's a good verbal articulator, but it "barely" plays any instruments. The structure of the system is illustrated in Figure 1; this paper describes essential components of the "ears" and "mouth." That division of labor is deliberate. Large audio-language models vastly underperform on many simple music tasks compared to smaller, targeted models [22], a finding that persists in the symbolic domain [39, 16]. Systems that close the gap do so by attaching domain-specific perception modules to the LLM [37], the same role the "ears" play in our system.

Generative music models implicitly encode musical structure and style [5, 10, 25], but implicit encoding is not articulation: a co-writing partner must say what structures are present and what changes might improve a composition or arrangement. Machine agents could exchange feedback as mathematical signals, e.g. as gradients toward some endpoint, as in classifier guidance [8], but a human collaborator cannot act on a gradient. Description and reproduction make different demands: representation learning and reconstruction are separate tasks, long regarded as a tradeoff [9]. Representation autoencoders (RAEs) [40] show the tradeoff is an artifact of compression: their latents rearrange the input at nearly its original dimensionality, with reconstruction handled separately rather than by the representation objective. The pairing supports discriminative tasks and high-quality generation alike. We adopt this paradigm for music, with a conditional flow model standing in for the trained decoder: a single encoder, trained with no reconstruction objective and then frozen, serving as the ears and conditioning the mouth.

The internal representations are formed by a joint-embedding predictive architecture (JEPA) [17, 1]. JEPA has been applied to music before: Stem-JEPA [29] predicted musical stem compatibility from audio; Hachana and Rasheed [13] adapted JEPA to tokenized symbolic music with musically motivated masking of instruments, pitch classes and octaves, and found the learned representation dominated by positional information, which limited transfer to content tasks such as genre and style; and, concurrently with this work, Music-JEPA [35] learned an action-conditioned audio world model from paired piano audio and performance data, while ARIMA [38] learned windowed latent-predictive representations of symbolic music. Unlike these, our approach operates purely on symbolic (MIDI) piano-roll images, is trained with a hierarchical, multi-level Swin V2 encoder rather than a single-scale backbone, prevents representational collapse via a LeJEPA-style distributional regularizer (SIGReg) rather than an EMA teacher, and makes shift structure explicit in the objective rather than leaving it to be absorbed by positional encodings. This paper extends our earlier report [15], which details the architecture and training objectives; here we focus on the musical and co-creative capabilities, and report improvements to the model and the generative pipeline together with an assessment of gains from supervised signals and additional datasets.

To maximize accessibility for GPU-poor musicians (who may also prefer not to send works-in-progress to a cloud service) we adopted a fundamental engineering constraint: *timely inference on CPU-only runtimes*, demonstrated by a live demo linked from the upplemental materials.[1]

[1] Supplemental Materials: `drscotthawley.github.io/midi-rae-jepa-son`

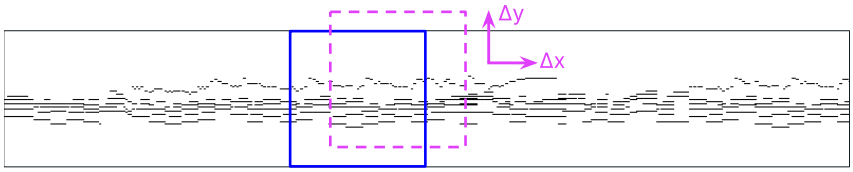

Figure 2: Just as a vision world model pans and tilts a camera to scan a panoramic scene, we take square crops of MIDI piano-roll images and compare crops translated in time by Δx time steps or in pitch by Δy semitones. Given one crop and a shift vector $(\Delta x, \Delta y)$, the model predicts the embedding of the shifted crop. Regions outside the image are filled with blanks. Our design requirement of fast CPU execution fixes the crop at 128×128 pixels. Pixels are binary, so note velocity is discarded, which is acceptable for the songwriting use case. Time is quantized to 32nd notes on a tempo-normalized grid, the finest grid at which a blank frame can separate two repeated notes in a binary image, so a crop spans four bars in 4/4, or eight seconds at 120 BPM.

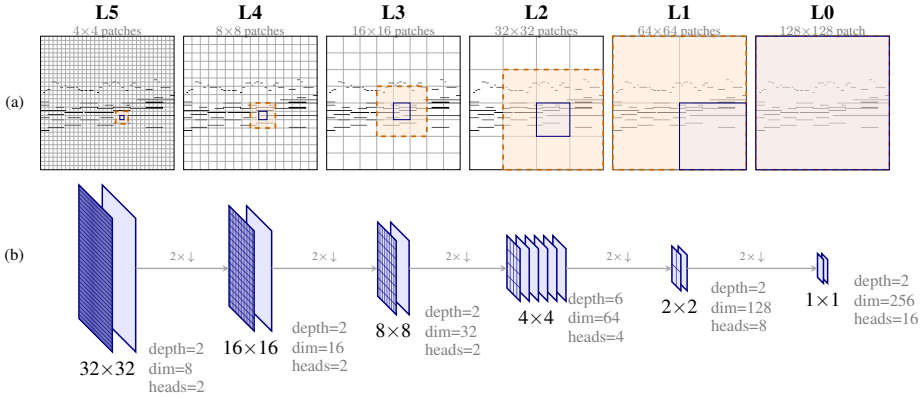

Figure 3: Our 2.55 million-parameter hierarchical Swin V2 Encoder. Model architecture hyperparameters were selected via ablation studies detailed in the Supplemental Materials.

2 How: A Symbolic Music "World Model"

> *"It wasn't made by people who went to the music conservatory. It was made by kids who felt something."* —*Rick Rubin, on early hip-hop [32]*

Rather than relying on supervised labels and the vocabulary of music theory, we train a self-supervised model that builds its own (hierarchical) representations of music, its own 'feel' for musical structure. It does this by exploring the "space" of musical compositions, cf. Figure 2. Human musical representations are hierarchical: notes, chords, phrases, song structure. Yet labeled data for that hierarchy is limited. Chord detectors and song-structure predictors can supply some of it, but we want to learn the hierarchy without labels, the way World Models in computer vision [12, 11] build semantic representations by predicting embeddings of one view of a scene from another. We do this on piano-roll images, partly to take advantage of the Swin V2 transformer [20, 19], a hierarchical representation learner already proven on images, and partly because a piano roll is a semantic proxy for an audio spectrogram, with perceptually aligned axes. Latents trained for reconstruction accuracy rarely become semantically meaningful; they tend instead to resemble spatially downsampled copies of the input. Following the RAE paradigm, ours is a latent model with *no compression*: the latents hold nearly as many numbers as the input (16,384 pixels → 16,128 floats), rearranged into a meaningful geometry rather than a smaller one.

Training objective. The encoder is trained with a combined self-supervised objective,

$$\mathcal{L} = \lambda \mathcal{L}_{\text{equiv}} + (1-\lambda)\mathcal{L}_{\text{SIGReg}} + \lambda_{\text{MEP}}\mathcal{L}_{\text{MEP}} + \lambda_{\text{fact}}\mathcal{L}_{\text{fact}},$$

applied level-wise. $\mathcal{L}_{\text{equiv}}$ is an equivariance loss that pulls together or pushes apart embeddings of shifted crop pairs in proportion to the shift magnitude, so the latent space respects pitch/time translation structure rather than collapsing to shift-invariance. $\mathcal{L}_{\text{SIGReg}}$ is LeJEPA's sketched isotropic Gaussian regularization [2], which applies an Epps–Pulley test along random projections to enforce an isotropic Gaussian prior, preventing representational collapse. (We also tried VISReg [36], which

replaces the Epps–Pulley test with a sliced-Wasserstein distance, and saw no consistent improvement in our metrics.) The equivariance and SIGReg losses are applied only at L0–L3: at L4 and L5, SIGReg washes out the fine note-level structure those levels carry, and patch-level equivariance is less meaningful at those scales. $\mathcal{L}_{\text{MEP}}$ is an intra- and inter-level masked-embedding-prediction loss in the style of I-JEPA [1]: an auxiliary predictor infers an EMA teacher's embeddings from the student's full context. $\mathcal{L}_{\text{fact}}$ is a soft factorization loss which pulls pitch- and time-shift difference vectors toward parallel, anti-parallel, or orthogonal geometry depending on augmentation type. It is applied only at L0–L2, enforcing equivariance out to larger shifts than $\mathcal{L}_{\text{equiv}}$ achieves alone [15]. Full derivations and hyperparameters are given in the Supplemental Materials.

3 Ears: What Does It Actually "Perceive"?

The "ears" of a full system such as that diagrammed in Figure 1 can partially consist of libraries of lightweight audio feature detectors, predicting categories such as genre, instrument type, and mood [4], or metrics such as pitch and tempo [24]. In addition, independent detectors for chords, keys, phrase boundaries and song structure are available. For songwriting, we are concerned less with nuances of performance than with the notes the player intended. Discarding audio detail in favor of a structured MIDI representation therefore costs us little. Audio-to-MIDI conversion [3] recovers that intent in a compact, semantic form already familiar from many music creators' workflows, and recent improvements [31] have made it reliable enough to build on. We treat it as upstream of this work. Our encoder is not meant to replace those small detectors. It learns representations that resist easy labeling, its own sense of "feel," obtained only by exploring the space of musical compositions. To gauge what those representations correspond to musically, we turn to standard probes. Each probe below trains a small linear (or ridge) model on frozen, per-level embeddings, or measures a geometric property of the embedding space directly; none fine-tune the encoder.

We call the model introduced here MIDI-RAE-JEPA-SON (MRJS), a successor of our earlier report's MIDI-RAE-JEPA [15], which appears in Table 1 as MRJ-48∧ ($\Delta t_{\max}$=48, factorization loss active), alongside a DINOv2 [26] baseline and a chance column (🎲). Unlike MRJ-48∧, whose pipeline components were trained against differing train/test splits on a dataset since found to contain corruption, MRJS and every downstream component share a single unified split on corrected data. That column is therefore historical reference, not a controlled comparison. The table reports each probe at the hierarchy level where it scores best, and those levels carry as much information as the scores. Phrase boundaries peak at the coarse end (L0–L2) while note density and harmonic content peak at the fine end (L4–L5), so the level at which a property becomes decodable tracks the musical

Table 1: Probing frozen embeddings for musical structure. Each cell reports a model's best value across its hierarchy levels (minimum for ↓ metrics), with the winning level in small type and a bar whose height is proportional to that level's receptive-field size (L0 tallest); 🎲 = chance level for that metric; bold = best model per row; cells are shaded per row with the plasma colormap (yellow = best). **Cross-song**: whether same-song embeddings sit closer together than cross-song ones. **Time** R^2: how well the distance between two embeddings predicts the temporal offset between their crops. **Joint Chord**: root and chord quality jointly. **Chroma** R^2: recovery of the local 12-bin pitch-class distribution. **Phrase AP/AUC**: linear probe detecting whether a human-annotated phrase boundary falls within half a bar of a crop's center, scored as average precision and ROC-AUC.

Metric	🎲	DINOv2	MRJ-48∧	MRJS	+chords	+phrases	Lakh-1×	Lakh-4×
Note Density R^2 ↑	n/a	.93 L1	.93 L5	.88 L5	.91 L5	.89 L5	.94 L5	**.94** L5
Cross-song ↓	n/a	.68 L1	.69 L5	.80 L3	.73 L1	**.57** L3	.62 L4	.77 L5
Time R^2 ↑	n/a	.23 L1	.24 L2	**.26** L4	.24 L1	.24 L2	.25 L2	.22 L3
Root Note ↑	.08	.24 L0	.25 L5	.24 L5	**.59** L2	.22 L5	.22 L5	.26 L5
Joint Chord ↑	.05	.17 L1	.17 L5	.18 L5	**.54** L3	.18 L5	.18 L5	.20 L5
Key Detection ↑	.05	.19 L0	.18 L5	.16 L5	**.70** L3	.17 L5	.15 L5	.18 L5
Chroma R^2 ↑	.00	.36 L1	.54 L4	.65 L5	**.83** L4	.75 L5	.76 L5	.61 L5
Phrase AP ↑	.17	.21 L0	.28 L1	.27 L0	.25 L2	**.29** L2	.28 L1	.26 L0
Phrase AUC ↑	.50	.57 L0	.61 L2	.61 L0	.58 L0	**.62** L0	.61 L1	.61 L0

time scale on which it operates. Temporal offset is the weakest probe throughout, .22–.26 for every model, which is expected: a piano roll is nearly stationary along the time axis, so sliding a crop sideways yields more of the same, whereas pitch is absolute and a vertical shift moves material into a different register. We read the harmony scores as capacity probes, not as downstream performance; small dedicated chord detectors do that job far better. The DINOv2 baseline is instructive: on properties that reduce to image statistics it is competitive or better, reaching .93 on note density and separating songs more sharply than MRJS, but it falls away on the musically specific probes, recovering chroma at .36 against .65 and phrase boundaries at .21 AP against .27.

To assess the effects of dataset variation and size, we trained encoders on subsets of the Lakh MIDI Dataset [28] containing $1\times$ and $4\times$ as many songs as POP909. Both transfer well: on the POP909 probes they match or exceed MRJS on note density and cross-song separation. At matched training budget, however, the $4\times$ subset gave no consistent gain over $1\times$, and degraded chroma (.76 to .61) and cross-song separation (.62 to .77). We also explored adding auxiliary supervised objectives to the self-supervised recipe: chord recognition, using POP909's chord annotations, and phrase-boundary prediction, using the human-verified annotations of Dai et al. [7], each applied as a small linear head on the coarse levels during training. Chord supervision makes harmonic content linearly decodable where it was barely present: root identification rises from .24 to .59, joint chord from .18 to .54, and key detection, which was never supervised, from .16 to .70. It also moves the seat of that information from L5 (near note-level) to the coarser and more "abstract" L2–L3. The cost is small: phrase AP falls from .27 to .25 and temporal offset from .26 to .24, while note density and cross-song separation both improve. Phrase supervision helps less, but is not without effect: boundary detection rises only slightly (AP .27 to .29, AUC .61 to .62), while cross-song separation improves markedly (.80 to .57). Harmonic content, then, has to be asked for; temporal and phrase structure largely emerge from the self-supervised objectives alone. Once the inputs are encoded, they can be sent to the "brain" as well as used as conditioning signals for the mouth described next.

4 Mouth: Making Suggestions

Verbal feedback and lyrical suggestions can be handled by a properly harnessed LLM. Musical suggestions benefit from their own generative model conditioned on the idea the user already has, as in an inpainting task where we replace a melody or an accompaniment, or suggest a different continuation. Hachana and Rasheed [13] anticipate that a reliable JEPA music representation could enable generative applications; this section realizes one. Autoregressive transformers generate left to right, so filling a hole in the middle of an existing passage is not something they do natively, and must be trained for specifically [33]. Diffusion models over piano-roll images, by contrast, support inpainting at inference time essentially for free [25, 14], and in a form that admits *graphical prompts*: the user draws on the piano roll where they want new notes, and the model generates notes suiting the surrounding material. But the price of this power is inference cost. Diffusion inpainting needs many integration steps, and our own earlier system, Pictures of MIDI [14], was large and slow enough to require a GPU. That is disqualifying under our design constraint. We therefore moved to flow matching [18], which can generate in very few integration steps: the smooth flow admits higher-order integration schemes such as RK4, and optimal-transport pairing [34] during training straightens the trajectories, making even Euler steps sufficient.

Recent work has favored flowing in pixel space rather than in the latent space [6, 21]. We do the same: we flow from noise in pixel space, conditioned on embeddings of the user's existing musical idea. We said earlier that there is no compression in the embeddings; however, we find it sufficient to reduce the conditioning signal via PCA, keeping at least 90% of the variance per level, resulting in a reduction of over $3\times$ (16,128 $\rightarrow$ 4,835 floats). This still reconstructs the input closely (F1=0.996), yet the goal is variations rather than copies. How far those variations stray is controllable: guidance strength sets how tightly a sample follows its conditioning, and dropping conditioning at different levels loosens it further. That dropout also points to how we handle inpainting. To inpaint, we apply spatial dropout to the PCA conditioning at the locations the user painted. The flow was trained with exactly this dropout, so it fills the gap from surrounding context without needing an inpainting-specific sampler, unlike sampling-time schemes for diffusion models [30], and we adopt none of the flow-specific schemes [27, 23]. Varying the dropout probability per level sets how coarse or fine the replacement is: dropping only the fine levels rewrites notes while keeping the harmonic frame, dropping every level rewrites the passage.

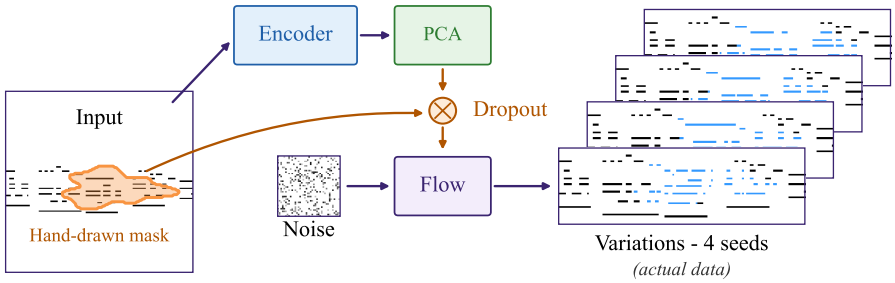

Figure 4: Unified generation and inpainting. The encoder's PCA-reduced conditioning maps pass through a dropout stage that zeroes patches under a hand-drawn mask, at user-chosen strengths per level; a flow-matching model then generates a window from noise, guided only by the surviving conditioning (10 Euler steps, guidance 1). Outside the mask the output is overwritten with the input, preserving untouched music exactly. With no mask, the same pipeline performs ordinary conditioned generation, varied by seed or guidance strength. The piano rolls are real data: one POP909 excerpt and four generated variations (cropped to the occupied pitch range for display).

Table 2: **(a)** Conditioned on a real window's own embeddings, the flow reproduces it nearly pixel-perfectly with no post-hoc alignment: the PCA-reduced conditioning retains enough to rebuild the input. **(b)** Note density restored in the masked region under the dropout pipeline of Fig. 4: the full input is always encoded, and conditioning patches under the mask are dropped at the stated per-level strengths. Restoration falls monotonically with dropout strength; spreads are $\pm$1–23%. Both panels: 10 Euler steps, guidance 1.0, over 6 songs $\times$ 3 seeds, with 3 masks in (b).

(a) Reconstruction, pixel F1	$\mathbf{0.996 \pm 0.004}$
(b) Note density restored in masked region vs. dropout strength	
no dropout (reconstruction)	100%
fine levels (L4–L5) @ 0.85	96%
fine levels (L4–L5) @ 1.0	76%
all levels @ 1.0	53%

Figure 4 shows four inpainted variations of a real excerpt, and Table 2 quantifies both generation and inpainting. Infilled notes generally fit the surrounding music, though they follow it less closely than larger diffusion models [14], and some land off. That manifests as variability in how much material the model commits inside the mask. The paradigm absorbs it: an AI Rick Rubin "barely" plays the instrument, and a suggestion need only convey an idea the user implements themselves.

5 Outro: What To Do With This?

We built a live demo[2] that serves not only as an interactive probe of the model's representational capacity, but as a *super fun* app that shows how this kind of control feels in practice. Encoding a 128×128 window takes 8.6 ms on two CPU threads, faster than Apple's Metal backend (19.9 ms). A suggestion (10 Euler steps at guidance 1.0, one function evaluation per step) takes 3.8 s on two CPU threads and 2.8 s on a full CPU (M1 Max, no MPS), within our "timely" design target; MPS drops this to 0.6 s, a laptop RTX 4090 to 0.10 s. Sampling is over 99% of these times, so latency scales with function evaluations. Because the model is likelier to under-fill a mask than over-fill it, the demo streams several variations ordered by how many new notes each contains, so the next is ready while the user auditions the last. Future work could extend the temporal field by treating the coarsest levels as tokens in a sequence model, or by folding piano-roll images into large squares as in [14]. Adding channels to encode note onsets, velocities, and multiple instruments would lead to richer musical capabilities, and an onset channel would free the separator frames that currently halve the window's span. While this system is intended for interacting with humans, nothing prevents someone from wiring it to a production system and closing the loop entirely. However, we strongly believe in AI for enhancing humans' experience of creativity, not for replacing it.

[2]Best experienced firsthand: `drscotthawley-midi-rae-jepa-son.hf.space`

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 *Conference on Computer Vision and Pattern Recognition*, 2023.

[2] Randall Balestriero and Yann LeCun. LeJEPA: Provable and scalable self-supervised learning without the heuristics, 2025.

[3] Rachel M. Bittner, Juan José Bosch, David Rubinstein, Gabriel Meseguer-Brocal, and Sebastian Ewert. A lightweight instrument-agnostic model for polyphonic note transcription and multipitch estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Singapore, 2022.

[4] Dmitry Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera, Oscar Mayor, Gerard Roma, Justin Salamon, José R. Zapata, and Xavier Serra. Essentia: An audio analysis library for music information retrieval. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, pages 493–498, Curitiba, Brazil, 2013.

[5] Rodrigo Castellon, Chris Donahue, and Percy Liang. Codified audio language modeling learns useful representations for music information retrieval. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[6] Shoufa Chen, Chongjian Ge, Shilong Zhang, Peize Sun, and Ping Luo. Pixelflow: Pixel-space generative models with flow, 2025.

[7] Shuqi Dai, Huan Zhang, and Roger B. Dannenberg. Automatic analysis and influence of hierarchical structure on melody, rhythm and harmony in popular music. In *Proceedings of the Joint Conference on AI Music Creativity (AIMC)*, 2020. arXiv:2010.07518. Human-verified phrase-level structure annotations for POP909 at `https://github.com/Dsqvival/hierarchical-structure-analysis`.

[8] Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, 2021.

[9] Sander Dieleman. Generative modelling in latent space, 2025.

[10] Zach Evans, CJ Carr, Josiah Taylor, Scott H. Hawley, and Jordi Pons. Fast timing-conditioned latent audio diffusion. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024.

[11] Quentin Garrido, Mahmoud Assran, Nicolas Ballas, Adrien Bardes, Laurent Najman, and Yann LeCun. Learning and leveraging world models in visual representation learning. *arXiv preprint arXiv:2403.00504*, 2024.

[12] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[13] Rafik Hachana and Bader Rasheed. Using a joint-embedding predictive architecture for symbolic music understanding. In *NeurIPS 2025 Workshop on AI for Music*, 2025.

[14] Scott H. Hawley. Pictures of midi: Controlled music generation via graphical prompts for image-based diffusion inpainting, 2024.

[15] Scott H. Hawley. MIDI-RAE-JEPA: Hierarchical representation learning and generation for symbolic music, 2026.

[16] Deepak Kumar, Emmanouil Karystinaios, Gerhard Widmer, and Markus Schedl. How far can pretrained LLMs go in symbolic music? controlled comparisons of supervised and preference-based adaptation. *arXiv preprint arXiv:2601.22764*, 2026.

[17] Yann LeCun. A path towards autonomous machine intelligence. *OpenReview preprint*, 2022. Version 0.9.2.

[18] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

[19] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer V2: Scaling up capacity and resolution. In *Conference on Computer Vision and Pattern Recognition*, 2022.

[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *International Conference on Computer Vision*, 2021.

[21] Yiyang Lu, Susie Lu, Qiao Sun, Hanhong Zhao, Zhicheng Jiang, Xianbang Wang, Tianhong Li, Zhengyang Geng, and Kaiming He. One-step latent-free image generation with pixel mean flows, 2026.

[22] Yinghao Ma, Siyou Li, Juntao Yu, Emmanouil Benetos, and Akira Maezawa. CMI-Bench: A comprehensive benchmark for evaluating music instruction following. In *Proceedings of the 26th International Society for Music Information Retrieval Conference (ISMIR)*, 2025.

[23] Ségolène Martin, Anne Gagneux, Paul Hagemann, and Gabriele Steidl. Pnp-flow: Plug-and-play image restoration with flow matching. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu, editors, *International Conference on Representation Learning*, volume 2025, pages 45466–45492, 2025.

[24] Brian McFee, Colin Raffel, Dawen Liang, Daniel P. W. Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in Python. In *Proceedings of the 14th Python in Science Conference (SciPy)*, pages 18–24, 2015.

[25] Lejun Min, Junyan Jiang, Gus Xia, and Jingwei Zhao. Polyffusion: A diffusion model for polyphonic score generation with internal and external controls. In *Proceedings of the 24th International Society for Music Information Retrieval Conference (ISMIR)*, 2023.

[26] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2024.

[27] Ashwini Pokle, Matthew J. Muckley, Ricky T. Q. Chen, and Brian Karrer. Training-free linear image inverses via flows. *Transactions on Machine Learning Research*, 2024.

[28] Colin Raffel. *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*. PhD thesis, Columbia University, 2016.

[29] Alain Riou, Stefan Lattner, Gaëtan Hadjeres, Michael Anslow, and Geoffroy Peeters. Stem-JEPA: A joint-embedding predictive architecture for musical stem compatibility estimation. In *International Society for Music Information Retrieval Conference*, 2024.

[30] Simon Rouard and Gaëtan Hadjeres. Crash: Raw audio score-based generative modeling for controllable high-resolution drum sound synthesis, 2021.

[31] Simon Rouard, Michael Krause, Axel Roebel, Carl-Johann Simon-Gabriel, and Alexandre Défossez. Muscriptor: An open model for multi-instrument music transcription, 2026.

[32] Rick Rubin. Rick Rubin: The 60 minutes interview. *60 Minutes*, CBS News. Interview by Anderson Cooper, January 2023.

[33] Chih-Pin Tan, Alvin W. Y. Su, and Yi-Hsuan Yang. Melody infilling with user-provided structural context. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022.

[34] Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. Expert Certification.

[35] Ziyu Wang, Kun Fang, and Yann LeCun. Music-JEPA: Learning a world model of sound from action, 2026.

[36] Haiyu Wu, Randall Balestriero, and Morgan Levine. Visreg: Variance-invariance-sketching regularization for jepa training, 2026.

[37] Meng Yang, Jon McCormack, Maria Teresa Llano, Wanchao Su, and Chao Lei. MIDI-LLaMA: An instruction-following multimodal LLM for symbolic music understanding. *arXiv preprint arXiv:2601.21740*, 2026.

[38] Mingyang Yao and Zhaoxiang Feng. ARIMA: Reconstruction-grounded predictive representation learning for symbolic music. *arXiv preprint arXiv:2607.10003*, 2026.

[39] Jiahao Zhao, Yunjia Li, Wei Li, and Kazuyoshi Yoshii. ABC-Eval: Benchmarking large language models on symbolic music understanding and instruction following. *arXiv preprint arXiv:2509.23350*, 2025.

[40] Bowen Zheng, Nan Ma, Shengbang Tong, and Saining Xie. Diffusion transformers with representation autoencoders. *arXiv preprint arXiv:2510.11690*, 2025.