PIANOPOET: UNIFYING PIANO TRANSCRIPTION AND RENDERING VIA AUTOREGRESSIVE MODELING

Anonymous authors
Paper under double-blind review

ABSTRACT

Piano transcription and rendering connect the acoustic and symbolic domains of music in opposite directions; however, unified models that can perform both tasks lag behind task-specific methods. One approach to unification is to formulate both tasks as inverse sequence-to-sequence problems which share a token representation space; however, audio input and output formats appropriate for the tasks differ significantly. In this paper, we investigate this challenge and present PianoPoet, a single-checkpoint autoregressive framework that unifies both tasks with a shared transformer backbone. PianoPoet shares a common MIDI tokenizer for both directions while decoupling audio representations: continuous semantic audio frames for fine-grained transcription, and a piano-specific discrete audio tokenizer for realistic piano rendering. PianoPoet outperforms prior unified models, achieves the best results among the evaluated generative transcription methods, and outperforms specialized rendering methods in most evaluated metrics. It supports full-band stereo rendering, enabling stereo-position control through acoustic prompts. Ablations show that the proposed PianoPoet framework allows both directions to share a common backbone without quality degradation, maintaining performance comparable to its single-task variants.

1 INTRODUCTION

Solo-piano transcription and rendering provide a challenging setting for studying the connection between the acoustic and symbolic domains of music. Piano transcription must recover overlapping notes, timing, and dynamics from audio at a very fine temporal resolution (Benetos et al., 2019), whereas piano rendering transform symbolic events into natural performance audio that is acoustically realistic. Although pianists naturally handle both tasks bi-directionally, recent unified models (Jung et al., 2025) lag behind task-specific methods (Hawthorne et al., 2021; Tang et al., 2025b).

A primary challenge in unifying the two tasks into one model lies in the differences in both the audio representations and technical paradigms that are best suited to each application. Piano transcription is dominated by discriminative models that process mel spectrograms directly into pitch activation grids (Hawthorne et al., 2017; Kong et al., 2021; Toyama et al., 2023; Yan & Duan, 2024b). While autoregressive (AR) models have also been used for transcription (Hawthorne et al., 2021; Gardner et al., 2021; Rouard et al., 2026), they remain behind discriminative methods on transcription accuracy benchmarks. Piano rendering models perform the opposite role: predicting audio conditioned on MIDI (MIDI Manufacturers Association, 1996) events. This has been done with a variety of generative frameworks, such as GANs (Dong et al., 2022), diffusion (Jing et al., 2026), or by predicting discrete acoustic tokens autoregressively (Tang et al., 2025b; Jung et al., 2025). Whereas transcription systems require fine-grained audio representations for maximum information extraction, piano rendering systems instead prefer compact yet perceptually realisitc representations.

Despite this, efforts have been made to unify both directions. Wave2MIDI2Wave (Hawthorne et al., 2018) used two models with distinct architectures for transcription and rendering. More recently, U-MusT (Jung et al., 2025) handled both tasks with a shared T5 architecture (Raffel et al., 2020), audio and MIDI representations, and a similar next-token-prediction objective for each direction. However, it still used two different checkpoints, one for each task, and remained substantially behind the state of the art (SOTA) for task-specific methods.

In this work, we introduce *PianoPoet*, a single-checkpoint model that unifies piano transcription and rendering in an autoregressive framework (Fig.1). As the divergent technical approaches to each task

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

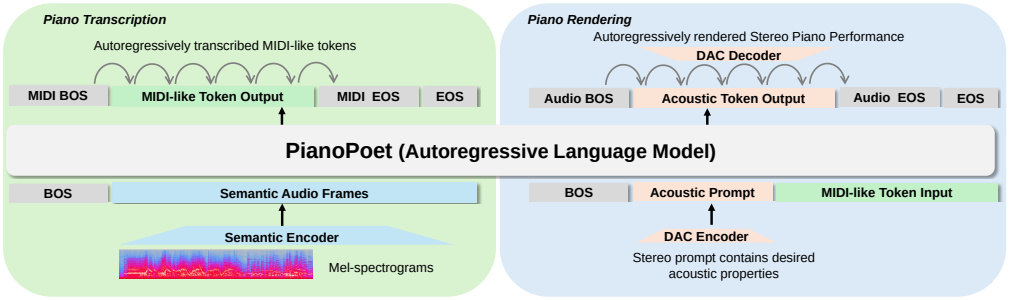


Figure 1: The proposed PianoPoet model. On top of the discrete MIDI-like tokens and acoustic tokens created by corresponding tokenizers, additional continuous semantic audio frames are introduced to enhance the understanding of fine details. The right side presents the sequence format of the piano rendering, while the left side presents the transcription sequence. An autoregressive language model learns both types of sequences during training.

prevent unification, we formulate both as sequence-to-sequence problems, allowing them to be handled by a shared causal transformer backbone (Radford et al., 2019). We identify the construction of training sequences with task-specific representations as a key ingredient for unification, and address this challenge by sharing a common MIDI tokenizer whilst decoupling audio representations: continuous semantic audio frames for fine-grained understanding and a piano-specific discrete audio tokenizer for realistic rendering (Fig.1). Critically, this design allows both tasks to be optimized under a common autoregressive next-token-prediction objective while preserving fine-grained information retrieval and acoustically natural rendering.

Our contributions are: (1) We present PianoPoet, a single-checkpoint AR framework that unifies piano transcription and rendering (Fig.1) and achieves competitive performance in both tasks (Fig.2). (2) Ablation studies show that the unified PianoPoet performs comparably to its single-task variants, indicating little interference between the two tasks when decoupled audio representations are used. This suggests a potential of this framework to unify more tasks in the future. (3) Extensive studies are conducted regarding the impact of audio representations for transcription and rendering, offering practical insights to not only unified but also task-specific models: The semantic audio frames are essential to excellent transcription performance; the piano-specific full-band stereo audio tokenizer enables realistic piano rendering with stereo position control. Audio samples are provided as the supplementary material.

2 Related Work

Piano transcription. Historically, piano transcription has been modeled as a discriminative problem, involving frame-wise prediction of pitch activations (Hawthorne et al., 2017; Kong et al., 2021; Wei et al., 2022; Toyama et al., 2023; Wang et al., 2025a). Building on this framework, the recent work of Yan et al. (2021) aims to learn the intervals between audio notes with Conditional Random Fields (CRFs), and achieved SOTA performance in follow up work (Yan & Duan, 2024b). Recently, Wei et al. (2026) re-formulates the task as a distribution matching problem which can be approached with an optimal transport objective.

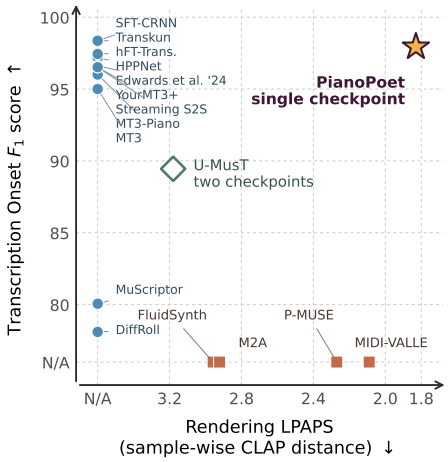


Figure 2: Piano transcription and rendering performance of different methods on MAESTRO test set. PianoPoet achieves competitive performance on both tasks with a single checkpoint.

While diffusion-based methods have been explored for transcription (Cheuk et al., 2023; Yeung et al., 2026) the language modeling (LM) approach has been more popular for generative transcription. Hawthorne et al. (2021) introduces a T5-like encoder-decoder LM for piano transcription, which was later extended to multiple instruments in Gardner et al. (2021); Tan et al. (2024); Chang et al. (2024). Wei et al. (2025) utilizes the autore-

gressive (AR) Transformer to achieve streaming piano transcription. Recently, MuScriptor (Rouard et al., 2026) presents a decoder-only AR-LM with 1.4B parameters and large-scale training data.

Neural Audio Tokenizer. Audio language modeling has shown competitive performance in generative audio tasks (Kreuk et al., 2023; Sheffer & Adi, 2023; Comunità et al., 2024; Yang et al., 2024; Wang et al., 2025b; Ye et al., 2025; Viertola et al., 2025; Zhong et al., 2025; Caillon et al., 2026; Nagashima et al., 2026). Neural audio tokenizers (Défossez et al., 2022; Kumar et al., 2023) are encoder-decoder models that compress audio signals into discrete audio tokens, a compact representation which enables language-modeling approaches to audio generation. Vector quantization (VQ) (Van Den Oord et al., 2017) is often employed residually (RVQ) to extend the codebook size of neural audio tokenizers (Défossez et al., 2022; Kumar et al., 2023; Caillon et al., 2026). Scalar quantization, e.g., LFQ and FSQ (Yu et al., 2024; Mentzer et al., 2024), have enabled the vocabulary size of audio tokenizers to scale to around 16-bits (Parker et al., 2025; Ye et al., 2025). Infinity (Han et al., 2025) further extends the vocabulary size to 64-bit in computer vision.

Piano rendering. The naive concatenation of separate piano note samples can sound mechanical. Dong et al. (2022) pioneered full-track neural MIDI rendering for piano, and M2A (Tang et al., 2025a) improves this methodology; however, both methods suffer from missing acoustic details. MIDI-VALLE (Tang et al., 2025b) introduces a LM-based text-to-speech (TTS) approach (Wang et al., 2023) to piano rendering: Audio signals are tokenized using a version of the EnCodec (Défossez et al., 2022) neural audio tokenizer finetuned for piano acoustics. To predict tokens for the multiple residual codebooks, the authors combine both an autoregressive and bi-directional LM. Recently, P-MUSE (Jing et al., 2026) adapts a diffusion-based TTS method (Eskimez et al., 2024) to MIDI-to-audio rendering.

Unified piano models. Hawthorne et al. (2018) introduced the MAESTRO dataset and trained different neural models on it for piano transcription and rendering, respectively, with different architectures (Hawthorne et al., 2017; Van Den Oord et al., 2016). U-MusT (Jung et al., 2025) revisited this topic, using the T5 architecture (Raffel et al., 2020) to approach the general music-score-to-audio task with piano rendering as a sub-task, and the audio-to-music-score task with piano transcription as a sub-task. Although both directions use the a unified architecture, U-MusT approaches piano transcription and rendering with two separate checkpoints. A unified model for piano transcription and rendering is yet to be explored.

3 METHODOLOGY

We base PianoPoet on a causal transformer backbone without cross-attention (Grattafiori et al., 2024), and formulate both piano transcription and piano rendering as sequence-to-sequence tasks that can be addressed by an autoregressive language model (LM). As shown in Fig.1: the LM either autoregressively samples MIDI-like tokens to transcribe the input audio, or samples acoustic tokens to render the MIDI event contents. During training, we train on a next-token prediction objective, blending transcription and rendering sequences with a ratio of 1:3.

A key challenge in the design of PianoPoet is the representations used to construct the training sequences: although the MIDI tokenizer can be shared across both tasks, in our investigations we find that the audio representations need to be decoupled.

A MIDI tokenizer converts MIDI events into a sequence of MIDI-like tokens that can be processed by an LM, and is therefore a key component for both transcription and rendering. We empirically chose the tokenizer proposed in Bradshaw et al. (2025). It quantizes every 5 levels of the 128-level velocity and then jointly encode the 128 MIDI pitches with its corresponding velocity in pair, resulting in 128 (MIDI pitches) $\times 26$ (quantized velocity levels) $=$ 3328 words. The onset and duration use 500 timestamp tokens respectively, at a 10 ms resolution. The MIDI tokenizer adopts a hybrid absolute-relative timestamps: within 5 s, it uses timestamp tokens; a reset token is used once the timestamp exceeds 5 s. Instead of extending the duration of a note when the pedal is on, the tokenizer adopts pedal on and off tokens to represent pedal activities. Onset timestamp tokens are re-used to indicate the on- and off-timing of the pedal. Figure 3 visualizes an example MIDI tokenization, and a detailed account of the full tokenization scheme can be found in Appendix A.



Figure 3: MIDI-like token examples. "vel": velocity. "Dur": duration. The RESET token implies that 5 s has passed and all succeeding onset timestamps restart from 0 s.

3.1 Semantic Audio Frames for Transcription

As shown in Fig.1, we construct the sequence of piano transcription by concatenating the semantic audio frames with the corresponding MIDI-like tokens. During training, loss is only computed over the MIDI-like tokens. At inference, they are sampled auto-regressively with temperature 0.

We research the audio representations for the piano transcription task under the AR-LM paradigm from the following choices. (1) **Acoustic tokens.** This setting is attractive as it shares the same acoustic tokens used in the rendering sequence, hence simplifying the overall architecture. This design has been shown effective in computer vision (Wu et al., 2026) and piano transcription (Jung et al., 2025). (2) **Encoder-free.** The encoder-free design encourages the AR backbone to learn semantic extraction from low-level audio input such as waveforms (Team et al., 2026) or Mel-spectrograms (Rouard et al., 2026). (3) **Semantic encoder.** Semantic encoders have been shown beneficial for music transcription tasks (Liao et al., 2024; Yeung et al., 2026; Chaturvedi et al., 2026; Rouard et al., 2026). We chose to use features extracted by a frozen pretrained Conformer model (Gulati et al., 2020) based on the results in Sec.5.2. Details regarding the semantic audio frames can be found in Appendix.B.

3.2 Acoustic Tokenization with a 64-bit Binary Quantizer

Neural audio tokenizers decide the upper bound of piano rendering performance. However, existing tokenizers work poorly for real piano performance (Sec.5.3). To address this challenge, we present *PianoPoet DAC* (Fig. 4). More details of PianoPoet DAC can be found in Appendix C.

Architecture modification. Based on the DAC (Kumar et al., 2023) architecture, we set the input and output channels to 2 to support stereo. We set the down- and up-sampling rate to 960 at the sampling rate of 48 kHz to align the latent rate to the same 50 Hz as semantic audio frames described in Sec.3.1. The parameter amount is 76M. **Reconstruction loss.** On top of the original Mel reconstruction losses, we further introduced a mid-side (Parker et al., 2026) Mel loss for stable training of stereo sound stages. **Quantization bottleneck.** We found the adversarial training of the original RVQ bottleneck unstable for stereo music at BF16 precision, hence we replaced it with a binary (1-bit) quantizer that do not require codebook or commitment losses, which is implmented by setting the per-dimension level to 2 in FSQ (Mentzer et al., 2024). We present each audio token with 64 bits in total, enabling 2^{64} vocabularies. **Semantic post-training.** Injecting features of speech recognition is observed to be beneficial to the reverse speech synthesis task (Zhang et al., 2024; Défossez et al., 2024; Parker et al., 2025). We follow this trend to introduce a post-training phase after 1M pretraining steps of the tokenizer. In the post-training, while being optimized by the original losses, a cosine similarity loss is introduced between the pre-quantization embedding and the features from a semantic encoder trained for piano transcription.

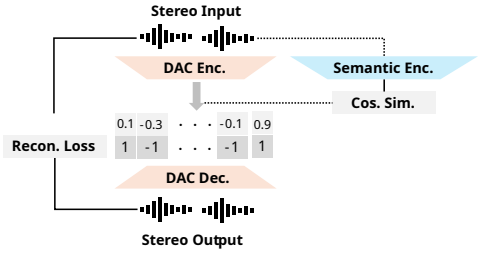


Figure 4: Overview of the PianoPoet DAC. The semantic distillation was only conducted after 1M pretraining steps.

3.3 Predicting Acoustic Tokens for Piano Rendering

Given the 64-bit codebook created by the audio tokenizer described in Sec.3.2, it is impractical to have a linear head to directly estimate the categorical distribution over 2^{64} vocabularies. The original LFQ paper (Yu et al., 2024) showed that we can factorize the binary codebook into multiple smaller codebooks to enable the language modeling. In Tab.1, we demonstrate that, when we use parallel linear heads proposed in LFQ, 4 $\times$ 16bit and 2 $\times$ 32bit demand excessive resources, while the 8 $\times$ 8bit is feasible. Since the 8-bit plan leaves small room for further improvement in efficiency, we choose this factorization plan across the experiments.

Table 1: Model size of parallel linear heads under different codebook factorization.

	8×8-bit (ours)	4×16-bit	2×32-bit
Params.	**1.6M**	201M	6597B

On top of the parallel linear head design, we investigate another type of LM head that uses a small Transformer to predict indices for multiple codebooks. Although the Transformer head was proposed for RVQ designs (Yang et al., 2023; Défossez et al., 2024; Caillon et al., 2026), we will compare a transformer head of 4.6M parameters with parallel linear heads in Sec.5.4.

Although piano rendering does not aim at producing timbres of different instruments, we found that the acoustic environment and production style can vary a lot. In the full-track rendering, the lack of acoustic prompt results in inconsistent acoustic properties. To address the challenge, during training, 5 s acoustic tokens from a different part of the same track are randomly chosen as the prompt and prepended to the rendering sequence as in Fig.1. Since the acoustic tokens contain the same acoustic environment but different note events than the target acoustic tokens, the AR-LM will learn to extract acoustic features from the acoustic prompt. At inference, arbitrary 5 s audio can serve as the acoustic prompt, which can even control stereo positions of the rendered audio, as will be shown in Sec.5.5.

4 EXPERIMENTS

4.1 DATASETS

To train and evaluate PianoPoet, we use paired audio and MIDI datasets chosen based on the different requirements of the transcription and rendering task. First, MAESTRO v3 (Hawthorne et al., 2018) is used to construct the sequences of both tasks, as it presents accurate MIDI-audio alignment and high-quality acoustic recordings of real piano performances. MAESTRO is the predominant dataset for training piano transcription models, and has additionally been used to train rendering models like U-MusT (Jung et al., 2025) and P-MUSE (Jing et al., 2026). Next, auxiliary datasets were introduced. Transcription performance can be improved by synthetic audio (Kusaka & Maezawa, 2025; Rouard et al., 2026). We therefore produced 590 hours synthetic audio based on the Aria-MIDI (Bradshaw & Colton, 2025) dataset. However, it is improper to train a model to predict synthetic audio when the target is to produce realistic piano performance. Following M2A and MIDI-VALLE (Tang et al., 2025a;b), we include a subset of ATEPP (Zhang et al., 2022) to construct the rendering sequences. For evaluation, we use the official MAESTRO v3 test set for both tasks, and additionally use the MAPs test set (ENSTDkAm/MUS and ENSTDkCl/MUS, following (Yan & Duan, 2024a)) as an additional evaluation for transcription.

4.2 METRICS

For transcription, we follow common practices to report note-level onset, onset–offset, and onset–offset–velocity F_1 scores computed by `mir_eval`[1] with its default matching tolerances.

The evaluation of piano MIDI rendering involves multiple aspects. First, we follow P-MUSE (Jing et al., 2026) to render the full-track audio, instead of 10-second short clips. This is because MAESTRO includes samples longer than 20 minutes, which presents unique challenges about the long-form coherence. Next, we choose a wide range of objective metrics from the audio generation and piano rendering community for automatic evaluation. Lastly, subjective tests are conducted.

Distribution matching metrics have been used in the piano rendering task (Tang et al., 2025b; Jung et al., 2025; Jing et al., 2026). These metrics measure the overall distance between sets of embeddings extracted from ground-truth and rendered audio. We extract embeddings using MERT-24kHz (Li et al., 2024), PaSST-32kHz (Koutini et al., 2021), and the music_audioset_epoch_15_esc_90.14.pt of CLAP-48kHz (Wu et al., 2023), and compute the distribution distance with Fréchet Distance (FD) (Kilgour et al., 2018), and Maximum Mean Discrepancy (MMD) (Gretton et al., 2012). All embeddings are computed after slicing the full-track audio data into non-overlapping 10-s clips.

Perceptual distance, LPAPS (Iashin & Rahtu, 2021; Saito et al., 2025) measures the pair-wise embedding distance between rendered and real performance using the aforementioned CLAP model.

Signal-based metrics evaluate reconstruction quality of neural codecs for piano music. We use Mel spectrogram distance following DAC (Kumar et al., 2023), and SDR using the MDX23 implementation (Fabbro et al., 2024). Following MIDI-VALLE, we report Mel distance for piano rendering models to measure the acoustic distance with the reference performance. We compute the log-RMS level envelope correlation (log-RMS Corr.) to measure the similarity of performance dynamics between a rendered and the reference performance, whose details are provided in Appendix E.

Note-level F_1. Similar to U-MusT and P-MUSE (Jung et al., 2025; Jing et al., 2026), we use Transkun (Yan & Duan, 2024b), the SOTA piano transcription model, to transcribe both rendered and ground-truth audio into MIDI events, and then compute their `mir_eval` scores described before.

[1] github.commir-evaluationmir_eval

Pedal distance. To measure the differences of pedaling between a rendered performance and the input MIDI events, we use a pedal depth detection model (Zhang et al., 2026) to extract pedal curves from audio and compute the L1 distance with the input MIDI events.

Subjective quality. We sampled tracks from 10 composers ranging from 18-th to 20-th century for two listening tests: A MUSHRA test (0-100 points) for audio tokenizers, and a 5-point test regarding the overall quality of piano rendering.

4.3 Experimental Designs

Across the evaluations, all variants of PianoPoet use a causal Transformer backbone described in Sec. 3 of 113M parameters. The Conformer encoder described in Sec.3.1 has 26M parameters. Detailed configurations can be found in Appendix D.

Unless otherwise stated, we train for at most 180k steps, and select the checkpoint with the lowest loss on the MAESTRO v3 validation set for either the transcription or rendering ablations. The unified model is picked by the averaged loss after training the same steps as its single-task variants. We optimize using AdamW (Loshchilov & Hutter, 2017) with a learning rate (LR) of 2×10^{-4}, betas of $(0.9, 0.95)$, and weight decay of 10^{-3}. We use a global batch size of 128, a maximum sequence length of 3072, and a 2,000-step linear warmup followed by linear decay to 10% of the peak LR.

Transcription. We design the following ablations to investigate the audio feature choices for transcription, as mentioned in Sec.3.1: (1) acoustic tokens extracted by our acoustic tokenizer described in Sec.3.2; (2) Mel-spectrograms; (3) Semantic features extracted by the pretrained Conformer encoder. All features are at 50 Hz. Details of these features can be found in Appendix.B

We apply two types of data augmentation during transcription training: (a) pitch transposition by an integer number of semitones, uniformly sampled from $\{-2, -1, 0, 1, 2\}$, with the MIDI target modified accordingly; (b) waveform-only augmentations that do not modify the MIDI target, *e.g.*, reverberation. The waveform-augmentation pipeline is applied after (a), with a probability of 75% per mini-batch. Acoustic tokens were precomputed in related experiments as repeatedly encoding stereo $48\,\mathrm{kHz}$ audio was prohibitively expensive, preventing us from applying waveform augmentations to these variants. To quantify the contribution of such data augmentations, we therefore include a Mel-input model trained without waveform augmentations. Details of waveform augmentations can be found in Appendix.F

Audio tokenizers. We evaluate the PianoPoet DAC with existing piano-specific and general-purpose neural codecs to show the value of training our piano-specific tokenizer.

Rendering. All methods receive the same target MIDI for rendering. In the comparison with previous baselines, we categorize methods based on their required prompts: M2A (Tang et al., 2025a) and U-MusT (Jung et al., 2025) do not use acoustic prompts; P-MUSE (Jing et al., 2026) and PianoPoet only use audio as the prompt, and were both prompted with the first 5 s of the corresponding real performance; MIDI-VALLE (Tang et al., 2025b) requires to be prompted by a 3-s audio-MIDI pair. We utilized the last 3 s of the 5-s prompts used by P-MUSE and PianoPoet for fair comparison.

We further ablate the design choices regarding piano rendering described in Sec. 3.2 and Sec 3.3: (1) whether to predict acoustic tokens using parallel linear heads or a Transformer head; (2) whether the semantic distillation of the audio tokenizer is beneficial to rendering.

5 Results

5.1 Comparison with prior work

Table 2 compares PianoPoet, the unified piano transcription and rendering model, with specialized discriminative and generative transcription models. PianoPoet obtains the strongest note-level scores among generative baselines, substantially narrows the gap between the state-of-the-art discriminative and generative methods.

Table 3 reports the corresponding comparison for piano rendering. FluidSynth strongly preserves the MIDI contents such as onsets, offsets, velocity, and pedaling. However, high FD, MMD, LPAPS, and Mel scores indicate that its acoustic features deviate from the real performance recording. The low log-RMS correlation indicates a deviation in dynamics from real performance as well. M2A

[2] www.fluidsynth.org, FluidR3_GM.sf2 ver.3.1 was used.

Table 2: Benchmark with piano transcription methods on the MAESTRO v3 and MAPS test sets. MAPS scores of PianoPoet are reported without finetuning on the MAPS training set. All scores are reported by the original papers, except that MuScriptor is evaluated by us. Bold: best result in each category. –: Result unavailable or not reported.

Model	MAESTRO v3 %			MAPS %		
	Onset F1↑	Onset & offset F1↑	Onset, offset & velocity F1↑	Onset F1↑	Onset & offset F1↑	Onset, offset & velocity F1↑
Discriminative methods						
HPPNet Wei et al. (2022)	97.18	83.80	82.24	86.63	64.39	59.77
hFT-Transformer Toyama et al. (2023)	97.44	90.53	89.48	85.14	66.34	48.20
Edwards et al. (2024)	96.60	–	–	88.40	–	–
Transkun Yan & Duan (2024b)	98.32	**93.48**	**92.94**	**90.38**	–	–
SFT-CRNN Wei et al. (2026)	**98.36**	90.78	–	–	–	–
Generative methods						
MT3-Piano Hawthorne et al. (2021)	96.01	83.94	82.75	–	–	–
MT3 Gardner et al. (2021)	95.00	80.00	–	–	–	–
DiffRoll Cheuk et al. (2023)	78.10	–	–	–	–	–
YourMT3+ Chang et al. (2024)	96.52	–	–	86.25	–	–
Streaming Seq2Seq Wei et al. (2025)	96.52	89.44	–	–	–	–
U-MusT A2I Jung et al. (2025)	89.45	–	–	–	–	–
MuScriptor-1.4B Rouard et al. (2026)	80.06	35.69	–	86.15	46.24	–
PianoPoet (ours, unified)	**97.94**	**91.50**	**90.62**	**89.70**	**67.11**	**53.65**

Table 3: Benchmarking piano rendering methods on the MAESTRO v3 test set with metrics described in Sec.4.2. Note-level F_1 scores are obtained via the default checkpoint in Transkun GitHub repository, which is different from the one in Tab.2. Pedal distance is obtained via Zhang et al. (2026). Bold indicates the best result.

Renderer	FD↓			MMD↓			LPAPS↓	Note-level F1 %↑			Pedal↓	Log-RMS Corr. %↑	Mel↓	Subj. Quality↑
	MERT	PaSST	CLAP	MERT	PaSST	CLAP	CLAP	Onset	Onset, Offset	Onset, Offset & Velocity				
Real recording	0.00	0.00	0.00	0.00	0.00	0.00	0.0	98.33	91.72	91.29	0.129	100	0.0	4.55
Non-neural-network														
FluidSynth[2]	5.13	364	0.843	28.1	149	7.10	2.94	**98.71**	**90.28**	**65.29**	**0.182**	83.6	2.15	1.73
Without audio prompt														
M2A Tang et al. (2025a)	1.29	150	0.464	5.43	75.4	3.56	2.94	17.61	6.44	5.00	0.258	77.1	1.73	–
U-MusT I2A Jung et al. (2025)	0.557	120	0.141	2.04	55.7	0.734	3.18	6.21	2.31	1.37	0.308	9.41	2.09	–
Prompted with 3 s of paired audio and MIDI														
MIDI-VALLE Tang et al. (2025b)	1.45	126	0.136	7.87	53.8	0.689	2.09	81.19	33.79	30.11	0.311	82.9	1.47	2.59
Conditioned on 5 s of audio														
P-MUSE Jing et al. (2026)	0.967	119	0.429	4.84	48.8	3.36	2.27	90.84	32.45	26.24	0.381	84.7	1.42	2.39
PianoPoet (ours, unified)	**0.136**	**64.3**	**0.0631**	**0.417**	**30.4**	**0.282**	**1.83**	70.47	40.79	36.63	0.201	**95.7**	**1.12**	**4.15**

improved acoustic similarities at the cost of missing details, indicated by its weak note-level scores. Although U-MusT uses a strong codec as shown in Tab.5, the volume can change dramatically in the middle of a track, resulting in weak performance in log-RMS correlation. MIDI-VALLE and P-MUSE set strong baselines for piano rendering. Nevertheless, PianoPoet, while being SOTA of generative transcription, still outperforms these baselines on distribution metrics (FD, MMD), pairwise alignment metrics (LPAPS, Mel), offset- and velocity-related scores, pedal distance, log-RMS correlation, and human preference. PianoPoet produced a weaker onset score compared to P-MUSE and MIDI-VALLE, but the score is still much higher than M2A or U-MusT. It is worth noting that FluidSynth achieved a note onset F_1 score better than the reference audio. This implies that the Transkun model (Yan & Duan, 2024b) used to detect note events may prefer mechanical rendering when detecting onsets. We leave the improvement of note-level scores for future work.

5.2 Transcription input ablation

Table 4 compares input audio features and data augmentations for transcription-only variants of PianoPoet, followed by the unified PianoPoet performance. All MAPS results are obtained without finetuning on its training set.

Acoustic tokens are poor inputs for precise note recovery. Directly projecting Mel features instead provides a strong baseline. Adding synthetic data without waveform augmentation improves all metrics on MAESTRO, while thewaveform augmentation described in Sec. 4.3 improves the per-

Table 4: Ablation studies for design choices related to piano transcription described in Sec. 3.1 and Sec. 4.3. MAPS scores are reported zero-shot. Bold: best results.

Model	Aug.	Synth.	MAESTRO v3			MAPS		
			Onset F1↑	Onset & offset F1↑	Onset, offset & velocity F1↑	Onset F1↑	Onset & offset F1↑	Onset, offset & velocity F1↑
Generative piano transcription baseline								
Streaming Seq2Seq Wei et al. (2025)	–	–	96.52	89.44	–	–	–	–
Acoustic-token input								
Acoustic tokens	×	✓	79.00	56.00	53.19	67.14	17.83	13.42
- w/ semantic distillation	×	✓	75.96	52.60	49.99	65.74	18.45	13.72
Mel input and data ablations								
Mel	✓	✓	97.39	87.87	86.40	89.66	67.43	53.30
- no synthetic data	✓	×	95.98	81.15	78.92	88.25	63.63	48.26
- no data augmentation	×	✓	96.87	87.95	86.57	84.08	56.83	43.84
Semantic encoder								
Conformer (frozen)	✓	✓	97.99	91.57	90.69	89.83	67.18	52.32
Conformer	✓	✓	**98.00**	**91.61**	**90.73**	**89.96**	**68.59**	**53.88**
Unified models								
PianoPoet	✓	✓	97.94	91.50	90.62	89.70	67.11	53.65

Table 5: Benchmarking neural tokenizers with metrics described in Sec.4.2. We apply mono-channel codec to the left and right channel of stereo audio respectively in this test. Fs: sampling rate. Bold: best result in each category.

Codec	Bit rate	F_s	Is stereo?	Signal		LPAPS	FD			MMD			MIR	MUSHRA
				Mel↓	SDR↑	CLAP↓	MERT↓	PaSST↓	CLAP↓	MERT↓	PaSST↓	CLAP↓	Onset F1↑	Score↑
Real Performance Audio	–	–	✓	0.00	121	0.00	0.00	0.00	0.00	0.00	0.00	0.00	98.33	84.8
General Neural Codecs														
EnCodec Défossez et al. (2022)	24kbps	48 kHz	✓	**0.741**	**11.6**	1.32	0.334	138	0.152	1.70	47.7	0.870	96.43	47.6
DAC Kumar et al. (2023)	7.8kbps	44.1 kHz	✗	0.836	5.77	**1.15**	**0.124**	170	0.150	**0.608**	56.8	0.979	**96.91**	55.9
SpectroStream Caillon et al. (2026)	16kbps	48 kHz	✓	0.665	2.73	1.29	0.276	**93.6**	**0.0668**	1.20	**39.7**	**0.335**	74.94	57.0
Piano Neural Codecs														
MIDI-VALLE Codec (Tang et al., 2025b)	2.2kbps	32 kHz	✗	0.864	8.11	1.37	0.224	92.6	0.111	1.00	38.2	0.535	84.85	68.2
U-MusT DAC (Jung et al., 2025)	3.4kbps	44.1 kHz	✗	**0.693**	**9.65**	1.29	0.498	92.3	0.115	1.59	42.2	0.713	-	73.0
PianoPoet DAC	3.2kbps	48 kHz	✓	0.766	8.02	1.15	0.0619	50.9	0.0553	0.267	24.4	0.250	87.36	**79.4**
w/o semantic distill.				0.762	8.04	**1.14**	**0.0532**	**47.7**	**0.0525**	**0.228**	**22.4**	**0.234**	**87.56**	79.0

formance on MAPS without synthetic data, for MAESTRO, it improves onset F_1 but reduces the offset- and velocity-sensitive scores. This might have been caused by the fact that reverberation and distortions like overdrive can alter the offsets and dynamics, but it is difficult to accurately reflect such influences in the MIDI event targets. Replacing the projected Mel features with a frozen Conformer encoder improves the performance on MAESTRO. Finetuning this encoder yields marginal performance gains at the cost of a 28% increase in GPU memory. We therefore keep the encoder frozen in the unified PianoPoet model.

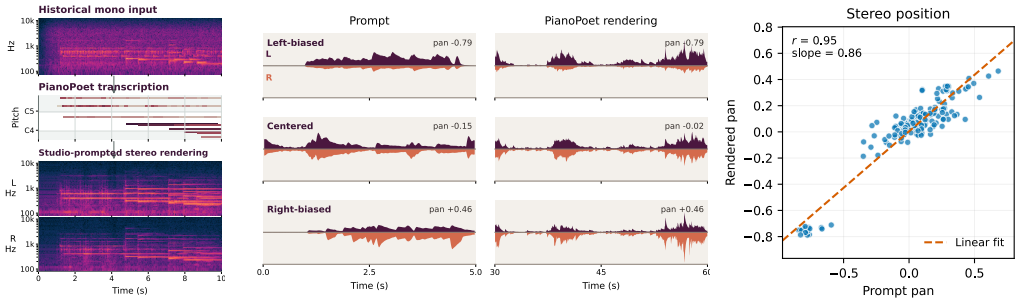
Last but not least, the unified PianoPoet remains very close to its transcription-only variant on both datasets, indicating little interference between the two tasks under the proposed framework in Fig.1.

5.3 Audio Tokenizer Evaluation

Tab.5 presents the comparison of general-purpose and piano-specific neural tokenizers. Although general-purpose codecs are strong in signal and onset metrics, their high FD and MMD indicate that their audio reproduction does not match the distribution of the target real performance, which is justified by the MUSHRA test, too. Previous piano-specific tokenizers showed limited improvement in distribution matching and LPAPS scores. At 3.2 kbps, the PianoPoet DAC described in Sec.3.2 maintained competitive performance for signal and LPAPS metrics, as well as a decent Onset score, but largely improved its FD and MMD. Moreover, PianoPoet DAC is rated the best in the MUSHRA test, indicating its compatibility with human preference regarding classical piano performance. Semantic distillation changes the reconstruction metrics only marginally, suggesting that its value should be assessed through the downstream rendering task.

Table 6: Ablation study regarding rendering choices described in Sec.3.2 and Sec.3.3. Bold: best result.

Configuration	FD↓			MMD↓			LPAPS↓	Note-level F1↑			Pedal↓	Mel↓	Log-RMS Corr. ↑
	MERT	PaSST	CLAP	MERT	PaSST	CLAP	CLAP	Onset	Onset, Offset	Onset, Offset & Velocity			
Parallel linear heads	0.174	68.8	0.0625	0.589	32.9	0.294	1.82	71.71	42.44	37.97	0.200	1.13	0.956
+ Transformer head	**0.110**	60.5	0.0578	**0.293**	28.7	0.273	1.85	60.82	34.00	30.47	**0.199**	**1.11**	**0.959**
+ Semantic Tokenizer	0.139	**59.2**	**0.0563**	0.464	**25.9**	**0.241**	**1.81**	**73.41**	**43.44**	**39.25**	0.200	1.15	0.954
PianoPoet (unified)	0.136	64.3	0.0631	0.417	30.4	0.282	1.83	70.47	40.79	36.63	0.201	1.12	0.957



(a) (b) (c)

Figure 5: Qualitative study of PianoPoet's capability. **(a)** Zero-shot historical piano recording restoration. **(b)** Waveform envelopes show that PIanoPoet reflects the stereo position of the prompt given the same MIDI input. **(c)** The scatter plot of prompt and rendered stereo positions shows strong correlation across the MAESTRO test set.

5.4 Rendering Ablations

Table 6 ablates the design choices for rendering. The parallel linear head design provides a strong baseline. Replacing it with the Transformer head improves FD and MMD at the cost of worse note-level scores. The semantic post-training of PianoPoet DAC helped to improve the note-level scores while maintaining the competitiveness in other metrics. The final unified PianoPoet model remains very close to its render-only variant across the metrics, indicating that being SOTA of generative piano transcription does not compromise PianoPoet's rendering performance.

5.5 Qualitative Results of PianoPoet

Fig.5a demonstrates that PianoPoet can perform zero-shot piano audio restoration. PianoPoet first transcribes an old recording then renders it using a studio-quality acoustic prompt, producing less background noise, clearer harmonic structures, and stereo expansion. Fig.5b shows that, given the transcription result in Fig.5a, the stereo position of restored audio can be controlled by acoustic prompts. The estimation of stereo position (panning) is explained in Appendix.E. The scatter plot in Fig.5c confirms a strong correlation of stereo positions between prompts and rendered audio across the MAESTRO test set.

6 Conclusion

Recent unified models lag behind task-specific methods. We address this challenge with PianoPoet, a unified autoregressive (AR) framework that performs both tasks with a causal Transformer backbone. We formulate both tasks as sequence-to-sequence mapping and identify the construction of sequences with proper representations as a key challenge. PianoPoet shares a MIDI tokenizer for both directions while disentangling audio representations: continuous semantic audio frames for fine-grained transcription, and a piano-specific discrete audio tokenizer for realistic piano rendering, allowing both tasks to be optimized with the cross-entropy loss. PianoPoet outperforms prior unified models, achieves the best results among the evaluated generative transcription methods, and remains competitive with specialized rendering systems. It supports full-band stereo rendering, enabling stereo-position control through acoustic prompts. Ablations show that task-specific representations allow both directions to share an autoregressive backbone while maintaining performance comparable to PianoPoet's single-task variants.

AI USE STATEMENT

We used generative AI tools to proofread the manuscript, correct typographical and grammatical errors, and improve its layout and formatting. AI tools were also used to assist with coding tasks such as general debugging, demo-page development, computing environment setup, and other supporting tasks. Nevertheless, all experimental designs and scientific content were developed by the authors. We reviewed and revised all AI-assisted material and take full responsibility for the scientific claims, reported results, text, and artifacts in the final submission.

REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce PianoPoet throughout the paper and appendices. Sec. 3 describes the unified AR formulation, sequence construction, training objective, and task-specific audio representations. Appendices A, B, and C provide further details on the MIDI tokenizer and full-track reconstruction pipeline, semantic audio features, and the architecture and two-stage training procedure of PianoPoet DAC, respectively. Appendix D reports the configurations and parameter counts of the Transformer and Conformer modules. Sec. 4.1 documents the training and evaluation datasets, while Sec. 4.2 and Appendix E define the evaluation metrics. Sec. 4.3 describes the training protocols, checkpoint-selection procedure, and baseline configurations, and Appendix F specifies the waveform-augmentation pipeline. We will release the source code, model configurations, preprocessing and evaluation pipelines, and other supporting materials to facilitate reproduction of our results.

REFERENCES

Emmanouil Benetos, Simon Dixon, Zhiyao Duan, and Sebastian Ewert. Automatic music transcription: An overview. *IEEE Signal Processing Magazine*, 36(1):20–30, 2019. doi: 10.1109/MSP.2018.2869928.

Louis Bradshaw and Simon Colton. Aria-midi: A dataset of piano midi files for symbolic music modeling. In *International Conference on Learning Representations*, volume 2025, pp. 99894–99910, 2025.

Louis Bradshaw, Honglu Fan, Alexander Spangher, Stella Biderman, and Simon Colton. Scaling self-supervised representation learning for symbolic piano performance, 2025. URL `https://arxiv.org/abs/2506.23869`.

Antoine Caillon, Brian McWilliams, Cassie Tarakajian, Ian Simon, Ilaria Manco, Jesse Engel, Noah Constant, Yunpeng Li, Timo Denk, Alberto Lalama, et al. Live music models. *Advances in Neural Information Processing Systems*, 38, 2026.

Sungkyun Chang, Emmanouil Benetos, Holger Kirchhoff, and Simon Dixon. Yourmt3+: Multi-instrument music transcription with enhanced transformer architectures and cross-dataset stem augmentation. In *2024 IEEE 34th International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6. IEEE, 2024.

Ojas Chaturvedi, Kayshav Bhardwaj, Tanay Gondil, Benjamin Shiue-Hal Chou, Kristen Yeon-Ji Yun, Yung-Hsiang Lu, Yujia Yan, and Sungkyun Chang. Advancing multi-instrument music transcription: Results from the 2025 amt challenge. *arXiv preprint arXiv:2603.27528*, 2026.

Kin Wai Cheuk, Ryosuke Sawata, Toshimitsu Uesaka, Naoki Murata, Naoya Takahashi, Shusuke Takahashi, Dorien Herremans, and Yuki Mitsufuji. Diffroll: Diffusion-based generative music transcription with unsupervised pretraining capability. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Marco Comunità, Zhi Zhong, Akira Takahashi, Shiqi Yang, Mengjie Zhao, Koichi Saito, Yukara Ikemiya, Takashi Shibuya, Shusuke Takahashi, and Yuki Mitsufuji. Specmaskgit: Masked generative modeling of audio spectrograms for efficient audio synthesis and beyond. *ISMIR 2024*, 2024.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *arXiv preprint arXiv:2210.13438*, 2022.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*, 2024.

Hao-Wen Dong, Cong Zhou, Taylor Berg-Kirkpatrick, and Julian McAuley. Deep performer: Score-to-audio music performance synthesis. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 951–955. IEEE, 2022.

Drew Edwards, Simon Dixon, Emmanouil Benetos, Akira Maezawa, and Yuta Kusaka. A data-driven analysis of robust automatic piano transcription. *IEEE Signal Processing Letters*, 31: 681–685, 2024.

Sefik Emre Eskimez, Xiaofei Wang, Manthan Thakker, Canrun Li, Chung-Hsien Tsai, Zhen Xiao, Hemin Yang, Zirun Zhu, Min Tang, Xu Tan, et al. E2 tts: Embarrassingly easy fully non-autoregressive zero-shot tts. In *2024 IEEE spoken language technology workshop (SLT)*, pp. 682–689. IEEE, 2024.

Giorgio Fabbro, Stefan Uhlich, Chieh-Hsin Lai, Woosung Choi, Marco Martínez-Ramírez, Weihsiang Liao, Igor Gadelha, Geraldo Ramos, Eddie Hsu, Hugo Rodrigues, et al. The sound demixing challenge 2023–music demixing track. *Transactions of the International Society for Music Information Retrieval*, 7(1):63–84, 2024.

Josh Gardner, Ian Simon, Ethan Manilow, Curtis Hawthorne, and Jesse Engel. Mt3: Multi-task multitrack music transcription. *arXiv preprint arXiv:2111.03017*, 2021.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012. URL `http://jmlr.org/papers/v13/gretton12a.html`.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

Jian Han, Jinlai Liu, Yi Jiang, Bin Yan, Yuqi Zhang, Zehuan Yuan, Bingyue Peng, and Xiaobing Liu. Infinity∞: Scaling bitwise autoregressive modeling for high-resolution image synthesis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15733–15744. IEEE, 2025.

Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck. Onsets and frames: Dual-objective piano transcription. *arXiv preprint arXiv:1710.11153*, 2017.

Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. Enabling factorized piano music modeling and generation with the maestro dataset. *arXiv preprint arXiv:1810.12247*, 2018.

Curtis Hawthorne, Ian Simon, Rigel Swavely, Ethan Manilow, and Jesse Engel. Sequence-to-sequence piano transcription with transformers. *arXiv preprint arXiv:2107.09142*, 2021.

Vladimir Iashin and Esa Rahtu. Taming visually guided sound generation. *arXiv preprint arXiv:2110.08791*, 2021.

Chong Jing, Junan Zhang, Jing Yang, Yulun Wu, Fan Fan, and Zhizheng Wu. P-muse: Prompt-midi-optional model for unified instrumental music synthesis and editing. *arXiv preprint arXiv:2608.01920*, 2026.

Jongmin Jung, Dongmin Kim, Sihun Lee, Seola Cho, Hyungjoon So, Irmak Bukey, Chris Donahue, and Dasaem Jeong. U-must: A unified framework for cross-modal translation of score images, symbolic music, and performance audio. *IEEE Transactions on Audio, Speech and Language Processing*, 2025.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fr\'echet audio distance: A metric for evaluating music enhancement algorithms. *arXiv preprint arXiv:1812.08466*, 2018.

Nithin Rao Koluguri, Samuel Kriman, Georgy Zelenfroind, Somshubra Majumdar, Dima Rekesh, Vahid Noroozi, Jagadeesh Balam, and Boris Ginsburg. Investigating end-to-end asr architectures for long form audio transcription, 2023. URL https://arxiv.org/abs/2309.09950.

Qiuqiang Kong, Bochen Li, Xuchen Song, Yuan Wan, and Yuxuan Wang. High-resolution piano transcription with pedals by regressing onset and offset times. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3707–3717, 2021.

Khaled Koutini, Jan Schlüter, Hamid Eghbal-Zadeh, and Gerhard Widmer. Efficient training of audio transformers with patchout. *arXiv preprint arXiv:2110.05069*, 2021.

Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. Audiogen: Textually guided audio generation. *ICLR 2023*, 2023.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved rvqgan. *Advances in neural information processing systems*, 36:27980–27993, 2023.

Yuta Kusaka and Akira Maezawa. Learn from virtual guitar: A comparative analysis of automatic guitar transcription using synthetic and real audio. In *2025 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 1–5. IEEE, 2025.

Yizhi Li, Ruibin Yuan, Ge Zhang, Yinghao Ma, Xingran Chen, Hanzhi Yin, Chenghao Xiao, Chenghua Lin, Anton Ragni, Emmanouil Benetos, et al. Mert: Acoustic music understanding model with large-scale self-supervised training. In *International Conference on Learning Representations*, volume 2024, pp. 12181–12204, 2024.

WeiHsiang Liao, Yuhta Takida, Yukara Ikemiya, Zhi Zhong, Chieh-Hsin Lai, Giorgio Fabbro, Kazuki Shimada, Keisuke Toyama, Kinwai Cheuk, Marco A Martínez-Ramírez, et al. Music foundation model as generic booster for music downstream tasks. *arXiv preprint arXiv:2411.01135*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: Vq-vae made simple. *ICLR 2024*, 2024.

MIDI Manufacturers Association. MIDI 1.0 Detailed Specification. Technical Report 96-1-4, MIDI Manufacturers Association, 1996. URL https://midi.org/midi-1-0-detailed-specification. Version 4.2.1.

Chihiro Nagashima, Akira Takahashi, Zhi Zhong, Shusuke Takahashi, and Yuki Mitsufuji. 'studies for': A human-ai co-creative sound artwork using a real-time multi-channel sound generation model. *Advances in Neural Information Processing Systems*, 38, 2026.

Julian Parker, Anton Smirnov, Jordi Pons, CJ Carr, Zack Zukowski, Zach Evans, and Xubo Liu. Scaling transformers for low-bitrate high-quality speech coding. In *International Conference on Learning Representations*, volume 2025, pp. 51997–52021, 2025.

Julian D Parker, Zach Evans, CJ Carr, Zachary Zukowski, Josiah Taylor, Matthew Rice, and Jordi Pons. Same: A semantically-aligned music autoencoder. *arXiv preprint arXiv:2605.18613*, 2026.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision, 2022. URL https://arxiv.org/abs/2212.04356.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Simon Rouard, Michael Krause, Axel Roebel, Carl-Johann Simon-Gabriel, and Alexandre Défossez. Muscriptor: An open model for multi-instrument music transcription. *arXiv preprint arXiv:2607.08168*, 2026.

Koichi Saito, Dongjun Kim, Takashi Shibuya, Chieh-Hsin Lai, Zhi Zhong, Yuhta Takida, and Yuki Mitsufuji. Soundctm: Unifying score-based and consistency models for full-band text-to-sound generation. In *International Conference on Learning Representations*, volume 2025, pp. 57803–57832, 2025.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Roy Sheffer and Yossi Adi. I hear your true colors: Image guided audio generation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Hao Hao Tan, Kin Wai Cheuk, Taemin Cho, Wei-Hsiang Liao, and Yuki Mitsufuji. Mr-mt3: Memory retaining multi-track music transcription to mitigate instrument leakage. *arXiv preprint arXiv:2403.10024*, 2024.

Jingjing Tang, Erica Cooper, Xin Wang, Junichi Yamagishi, and György Fazekas. Towards an integrated approach for expressive piano performance synthesis from music scores. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2025a.

Jingjing Tang, Xin Wang, Zhe Zhang, Junichi Yamagishi, Geraint Wiggins, and George Fazekas. Midi-valle: Improving expressive piano performance synthesis through neural codec language modelling. *arXiv preprint arXiv:2507.08530*, 2025b.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, et al. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Keisuke Toyama, Taketo Akama, Yukara Ikemiya, Yuhta Takida, Wei-Hsiang Liao, and Yuki Mitsufuji. Automatic piano transcription with hierarchical frequency-time transformer. *arXiv preprint arXiv:2307.04305*, 2023.

Aaron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, Koray Kavukcuoglu, et al. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 12(1), 2016.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in NeurIPS*, 30, 2017.

Ilpo Viertola, Vladimir Iashin, and Esa Rahtu. Temporally aligned audio for video with autoregression. In *ICASSP 2025*, pp. 1–5. IEEE, 2025.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al. Neural codec language models are zero-shot text to speech synthesizers. *arXiv preprint arXiv:2301.02111*, 2023.

Qi Wang, Mingkuan Liu, and Maoshen Jia. Multitask frequency-time feature-wise linear modulation for piano transcription with pedal. *IEEE Transactions on Audio, Speech and Language Processing*, 33:1049–1062, 2025a.

Yuancheng Wang, Haoyue Zhan, Liwei Liu, Ruihong Zeng, Haotian Guo, Jiachen Zheng, Qiang Zhang, Xueyao Zhang, Shunsi Zhang, and Zhizheng Wu. Maskgct: Zero-shot text-to-speech with masked generative codec transformer. *ICLR 2025*, 2025b.

Weixing Wei, Peilin Li, Yi Yu, and Wei Li. Hppnet: Modeling the harmonic structure and pitch invariance in piano transcription. *arXiv preprint arXiv:2208.14339*, 2022.

Weixing Wei, Jiahao Zhao, Yulun Wu, and Kazuyoshi Yoshii. Streaming piano transcription based on consistent onset and offset decoding with sustain pedal detection. *arXiv preprint arXiv:2503.01362*, 2025.

Weixing Wei, Raynaldi Lalang, Dichucheng Li, and Kazuyoshi Yoshii. A distribution matching approach to neural piano transcription with optimal transport. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 14897–14901. IEEE, 2026.

Junfeng Wu, Yi Jiang, Chuofan Ma, Yuliang Liu, Hengshuang Zhao, Zehuan Yuan, Song Bai, and Xiang Bai. Liquid: Language models are scalable and unified multi-modal generators. *International Journal of Computer Vision*, 134(1):39, 2026.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP 2023*, pp. 1–5. IEEE, 2023.

Yujia Yan and Zhiyao Duan. Scoring time intervals using non-hierarchical transformer for automatic piano transcription, 2024a. URL https://arxiv.org/abs/2404.09466.

Yujia Yan and Zhiyao Duan. Scoring time intervals using non-hierarchical transformer for automatic piano transcription. *arXiv preprint arXiv:2404.09466*, 2024b.

Yujia Yan, Frank Cwitkowitz, and Zhiyao Duan. Skipping the frame-level: Event-based piano transcription with neural semi-crfs. *Advances in Neural Information Processing Systems*, 34: 20583–20595, 2021.

Dongchao Yang, Jinchuan Tian, Xu Tan, Rongjie Huang, Songxiang Liu, Xuankai Chang, Jiatong Shi, Sheng Zhao, Jiang Bian, Zhou Zhao, et al. Uniaudio: An audio foundation model toward universal audio generation. *arXiv preprint arXiv:2310.00704*, 2023.

Shiqi Yang, Zhi Zhong, Mengjie Zhao, Shusuke Takahashi, Masato Ishii, Takashi Shibuya, and Yuki Mitsufuji. Visual echoes: A simple unified transformer for audio-visual generation. *arXiv preprint arXiv:2405.14598*, 2024.

Zhen Ye, Xinfa Zhu, Chi-Min Chan, Xinsheng Wang, Xu Tan, Jiahe Lei, Yi Peng, Haohe Liu, Yizhu Jin, Zheqi Dai, et al. Llasa: Scaling train-time and inference-time compute for llama-based speech synthesis. *arXiv preprint arXiv:2502.04128*, 2025.

Michael Yeung, Keisuke Toyama, Toya Teramoto, Shusuke Takahashi, and Tamaki Kojima. Noise-to-notes: Diffusion-based generation and refinement for automatic drum transcription. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 14652–14656. IEEE, 2026.

Lijun Yu, José Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G Hauptmann, et al. Language model beats diffusion-tokenizer is key to visual generation. In *International Conference on Learning Representations*, volume 2024, pp. 765–783, 2024.

Hanwen Zhang, Kun Fang, Ziyu Wang, and Ichiro Fujinaga. Evaluating high-resolution piano sustain pedal depth estimation with musically informed metrics. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 16092–16096. IEEE, 2026.

Huan Zhang, Jingjing Tang, Syed RM Rafee, Simon Dixon, George Fazekas, Geraint A Wiggins, et al. Atepp: A dataset of automatically transcribed expressive piano performance. In *ISMIR*, pp. 446–453, 2022.

Xin Zhang, Dong Zhang, Shimin Li, Yaqian Zhou, and Xipeng Qiu. Speechtokenizer: Unified speech tokenizer for speech language models. In *International Conference on Learning Representations*, volume 2024, pp. 31798–31818, 2024.

Zhi Zhong, Akira Takahashi, Shuyang Cui, Keisuke Toyama, Shusuke Takahashi, and Yuki Mitsufuji. Specmaskfoley: Steering pretrained spectral masked generative transformer toward synchronized video-to-audio synthesis via controlnet. *WASPAA 2025*, 2025.

A DETAILS OF THE MIDI TOKENIZER

As mentioned in Sec.3, we use the MIDI tokenizer proposed in Bradshaw et al. (2025)[3]. In this section, we introduce more details of this tokenizer. Only piano-instrument tokens in the MIDI events are enabled.

For the 128 levels of velocity, this tokenizer quantizes every 5 levels resulting 26 effective velocity levels. The tokenizer further jointly encode the 128 MIDI pitches with its corresponding velocity in pair, resulting in 128 (MIDI pitches) $\times 26$ (quantized velocity levels) $= 3328$ words. This design helps to reduce the average token rate on MAESTRO validation set from around 40 Hz to around 30 Hz, at the cost of a larger vocabulary size.

The MIDI tokenizer adopts a hybrid absolute-relative timestamps for onset timestamps: within 5 seconds, onset timestamps are represented by absolute tokens at the 10 ms resolution, resulting in 500 words; A time-reset token is introduced to indicate that the previous onset timestamp has exceeded 5 s, all onset timestamps after the time-reset token need to add 5 s when reading. This design effectively limits the number of onset timestamp tokens to 500. As a comparison, the absolute timestamp in MT3 (Gardner et al., 2021) requires 1500 words for 15 s, and 3000 words for 30 s, growing proportionally to the maximum duration of the audio input.

Regarding durations, the tokenizer represents durations with 500 words at the 10 ms resolution, and clips all notes longer than 5 s to 5 s. This design will only introduce marginal distortions to the MIDI events, because we confirmed that notes longer than 5 s are extremely rare in MAESTRO dataset, The quantitative results in Tab.7 shows that all metrics are beyond 99.6%, suggesting little distortion introduced by the MIDI tokenization procedure.

Instead of extending the duration of a note when the pedal is on, the tokenizer adopts pedal on and off tokens to represent pedal activities. Onset timestamp tokens are re-used to indicate the on- and off-timing of the pedal.

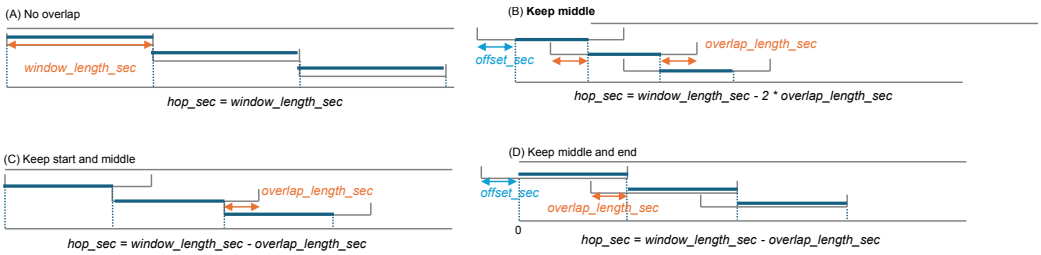


Figure 6: Illustration of different overlap-keep strategies.

During inference, we use an overlap-keep strategy to construct the full-track sequence. Specifically, we divide the 15 s sequence into start (the first 5 s), middle (the middle 5 s), and end (the last 5 s) segment, and quantify the distortions brought by MIDI tokenization and different overlap-keep strategies (Fig. 6). As is shown in Tab.7, by keeping only the middle 5 s of the sequence, the overlap-keep strategy maintains F_1 scores that are better than 99.6% in all three metrics, indicating that MIDI-like tokenization will not be a bottleneck to train PianoPoet.

Table 7: Comparison of chunk overlap-keep configurations. Higher values are better. Bold indicates the best results.

Chunk configuration	**Onset F_1**	**Onset & offset F_1**	**Onset, offset & velocity F_1**
No overlap	98.38	95.97	95.92
Keep start and middle	97.62	97.32	97.27
Keep middle and end	**99.95**	96.02	95.98
Keep middle (ours)	**99.95**	**99.64**	**99.61**

[3] https://github.com/EleutherAI/aria-utils.

B Details of Semantic Audio Frames

This section offers mode details for the semantic audio frames described in Sec.3.1.

Mel representation. For both Mel-based encoders or the encoder-free Mel-input variant of PianoPoet, we empirically adopt a double-stacked log-mel representation. We compute both spectrograms from audio sampled at 16 kHz, using a hop length of 160 samples. The first uses a 768-sample Hann window and 128 mel bins spanning 30 Hz to 4000 Hz, the second uses a 2048-sample window and 384 mel bins spanning 30 Hz to 8000 Hz. We concatenate the two spectrograms along the frequency axis, producing 512-dimensional features at 100 Hz. This provides views at two different time-frequency resolutions.

Conformer-based PianoPoet variant. A Whisper-inspired convolutional frontend (Radford et al., 2022) reduces the frame rate of the input Mel-spectrogram to 50 Hz, which is then sent to the Conformer backbone. We use a standard Conformer architecture (Gulati et al., 2020) based on the Parakeet recipe (Koluguri et al., 2023). This model was pretrained with a light-weight Transformer decoder on piano transcription tasks for 200 epochs on the dataset described in Sec.4.1, aiming at learning features that can be easily used in auto-regressive piano transcription. COnfigurations of the Conformer and the light-weight decoder is shown in Fig.8.

Table 8: Configurations of the Conformer encoder and tiny Transformer decoder used during encoder pre-training.

Component	Blocks	Attention Dim.	Heads	FFN Dim.	Params.
Conformer-based semantic encoder	4	512	8	2048	26 M
Tiny Transformer decoder	4	512	8	2048	15 M

Mel-input PianoPoet variant. In the ablation study, to keep a consistent frame rate of features fed to the AR backbone, we keep the convolutional frontend but remove the Conformer backbone from the encoder described above.

C Details of the Neural Audio Tokenizer

We will open the training and inference libraries for PianoPoet DAC.

Infrastructure. Instead of using the legacy libraries used in the original DAC repository, we from scratch created a repository that solely relies on more mainstream libraries. We conduct the trainining in BF16 precision to reduce memory consumption as well as taking advantage of compute-power acceleration of modern GPUs such as H100.

Pretraining recipe. The learning rate (LR) is set to 1×10^{-4}, with 10k steps of linear warm-up, followed by a cosine annealing 1e-6 after 1M steps. The total batch size was 24. We start from training the tokenizer with 32 frames. Since our latent rate is 50 Hz, which is smaller than the 86 Hz in the original DAC, this makes the duration of each training sample 0.64 s. After 600k steps of training, we extend the duration of each training sample to 1.28 s, this is to encourage the tokenizer to understand beyond a very short time window.

Post-training recipe. After 1M steps of the above pretraining, we introduce the pretrained semantic encoder as the teacher for semantic distillation. The challenge is about duration. The pretrained semantic encoder requires 15 s input, which is prohibitively expensive for high-fidelity adversarial training. We propose a strategy, where the encoder of PianoPoet DAC processes two inputs, one is 1.28 s and the other one is 15 s. The pre-quantization feature of the 15 s input is used to compute the cosine similarity betweeen semantic encoder features. The 1.28 s features are then sent to the PianoPoet DAC decoder for reconstruction. In this way we suppressed the memory usage, and enabled a total batch size of 10. After 200k steps of post-training, the cosine similarity reached 80%, hence we stopped the training.

D TRANSFORMER AND CONFORMER CONFIGURATIONS

On top of our open-source plan, configurations of the following modules are explained in this section: the AR Transformer backbone, the Conformer semantic encoder used for transcription, as well as the Transformer head used for piano rendering.

The AR backbone is based on the LLAMA 3.2 1B model[4], which employs a SwiGLU feed-forward network (FFN) Shazeer (2020) that is widely used in modern LLMs like LLAMA3 series (Grattafiori et al., 2024). With the configurations in Tab.9, the model is around 113-million parameters.

As introduced in Appendix B, we use a standard Conformer architecture (Gulati et al., 2020) based on the Parakeet recipe (Koluguri et al., 2023), Which employs four Conformer blocks with an attention dimension of 512, resulting in 26-million parameters.

The transformer head mentioned in Sec.3.3 deploys a single Transformer block with an attention dimension of 512, which is only marginally larger than the 1.6-M parallel linear heads.

Table 9: Configurations of the Transformer and Conformer modules used in PianoPoet.

Component	Blocks	Attention Dim.	Heads	FFN Dim.	Params.
Shared AR backbone	12	768	12	3072	113 M
Conformer-based semantic encoder	4	512	8	2048	26 M
Transformer head for rendering	1	512	8	2048	4.6 M

E DEFINITION OF LOG-RMS CORRELATION AND THE STEREO POSITIONING ESTIMATOR

Log-RMS level envelop correlation (Log-RMS Corr.). All signals are resampled to $f_s = 24{,}000$ Hz, truncated to their common duration. Frame-level loudness is computed as the log-scale frame-wise Root-Mean-Square (RMS) amplitude

$$l_t = \log \sqrt{\text{mean}\,(\mathbf{x}_t^2)}, \tag{1}$$

where t is the frame index. Each frame contains 9600 samples or 400 ms, with an overlap ratio of 75%, we create the loudness contour made by a series of RMS amplitudes.

We consider the 60 dB below the maximum RMS in the reference audio as a threshold, and frames below this threshold are considered silent and ignored. After this filtering, the Pearson correlation coefficient between the estimated and reference log-RMS sequences.

Stereo distance (Stereo Dist.). We compute an STFT at $f_s = 24{,}000$ Hz using a Hann window, an FFT size of $N_{\text{FFT}} = 2048$ samples, and a hop size of $H_{\text{S}} = 480$ samples (20 ms). The resulting power spectrum is projected onto 64 mel-frequency bands. Mono recordings are duplicated across channels.

Given left and right signals L and R, For each mel time–frequency bin i, the pan is estimated as

$$p_i = \frac{P_{R,i} - P_{L,i}}{P_{R,i} + P_{L,i}}. \tag{2}$$

where P denotes mel-band power.

Here, p_i typically ranges from $[-1, 1]$:

- $p_i = -1$ indicates fully left-panned audio,
- $p_i = 0$ indicates equal left and right energy, i.e., centered audio,
- $p_i = 1$ indicates fully right-panned audio.

[4] https://huggingface.co/meta-llama/Llama-3.2-1B.

F WAVEFORM AUGMENTATION FOR TRANSCRIPTION

During transcription training, we apply waveform augmentation to an entire batch with probability 0.75. Conditional on this outer gate, the effects in Table 10 are sampled independently and may be composed in the order shown. These perturbations do not modify the MIDI target.

Table 10: Waveform augmentation used for transcription. Probabilities are conditional on applying augmentation to the batch.

Effect	**Probability**	**Sampled parameters**
Fine pitch shift	0.02	Shift in $[-15, 15]$ cents
Overdrive	0.05	Gain 0–25; colour 5–95
Reverb	0.75	Random impulse response; wet mix 0–100%
Background noise	0.75	Random recording; SNR $20\,\mathrm{dB}$ to $50\,\mathrm{dB}$
Filtering	0.50	High-pass $20\,\mathrm{Hz}$ to $400\,\mathrm{Hz}$; low-pass $3.5\,\mathrm{kHz}$ to $8\,\mathrm{kHz}$; one or two EQ bands with gain $[-6, 6]$ dB and $Q \in [0.5, 1.5]$
Hard clipping	0.05	Additional gain $3\,\mathrm{dB}$ to $12\,\mathrm{dB}$ before clipping

The effect settings and selected reverb or noise recording are shared across the batch, while reverb wet mix and noise SNR are sampled separately for each example. Both asset pools were sourced from Freesound.com and will be released with their license and attribution metadata. All waveform augmentation is disabled during validation.