# Pianofi: Bridging Transcription and Arrangement in Audio-to-Piano Cover Generation

Jerry Zhu
*University Of Waterloo*
j25zhu@uwaterloo.ca

Jonathan Gong
*University Of Waterloo*
j56gong@uwaterloo.ca

***Abstract*—Audio-to-piano cover generation is a powerful technique bridging note based transcription with harmonic understanding of songs. This paper introduces Pianofi; a unified framework for cover generation spanning three SOTA paradigms: AMT-APC, PiCoGen, and Etude. We first construct a training and evaluation pipeline and compare models to clarify trade-offs between baselines. We then introduce enhancements including LoRA adapters, reranking, improved lead-sheet extraction, and targeted data augmentation, yielding upgraded variants. Finally, we explore cross-paradigm ensembling to combine transcription accuracy and harmonic structure regularization. Our results demonstrate consistent F1 and Qmax improvements and more stable piano arrangements, highlighting the benefits of bridging transcription and symbolic generation. The app is deployed at pianofi.ca.**

## I. Introduction

Audio-to-piano cover generation aims to transform arbitrary polyphonic audio into piano arrangements. Unlike classical baselines of automatic music transcription (AMT), which focuses on recovering note-level modulations, piano cover generation requires modeling arrangement structure, pitch inflections, and harmonic overtones. Recent approaches span multiple paradigms: transcription-based pipelines (AMT-APC), symbolic latent diffusion model generation (PicoGen), and hybrid structural modeling systems (Etude). However, these methods, presented in their original forms, are evaluated in isolation under different training regimes and metrics, making it difficult to assess their relative benefits.

In this work, we present the first controlled benchmarking study comparing three representative approaches, standardizing evaluation using frame and note-level F1 metrics under matched training and inference settings. Beyond benchmarking, we introduce improvements to each paradigm and explore cross-paradigm-ensembling transformer-based models for end-to-end audio-to-piano covers.

## II. Motivation

Current audio-to-piano systems implicitly make different assumptions about the nature of the data given. Transcription-based models treat piano arrangement as a heuristic note prediction problem, optimizing for pure accuracy but often neglecting structure and overfitting. Lead-sheet systems decompose the task into harmonic abstraction followed by generative modeling, improving global coherence but potentially sacrificing local note precision. Structural pipelines explicitly model beat grids and repetition patterns, yet rely heavily on ad hoc components and brittle heuristics.

Given these conceptual distances, there is little empirical evidence of which paradigm performs best under general (or variable) covers. Furthermore, simple architectural enhancements have not been systematically studied in this domain. Our goal is therefore twofold: (1) establish a reproducible benchmarking framework across paradigms, and (2) investigate whether baseline improvements and ensembling can meaningfully contribute to better data architectures. Furthermore, existing pipelines largely lack robust data augmentation protocols. To address this, we introduce self-training loops, targeted data augmentations, and retrieval-augmented generation (RAG) reranking into the generation flow.

## III. Related Work

*a) Pop2Piano:* Pop2Piano sets up the initial baseline for all piano to midi transformation techniques. However, it is jsut a crude baseline, as it does not take into account complex musical differences like harmonic drift, style transfer, or note precision. We encounter three documented methods to improve this base model. [2]

*b) AMT-APC:* AMT-APC (Automatic Piano Cover by Fine-Tuning an AMT Model) represents a shift in piano cover generation by leveraging the "sound-capturing" precision of established AMT models. Unlike previous models that often utilize fixed encoders or quantized rhythms, AMT-APC fine-tunes a pre-trained hFT-Transformer to map original multi-instrumental audio directly to high-resolution piano-roll representations.

To handle the inherent ambiguity of "style" in musical covers, the model introduces a 24-dimensional style vector, denoted by $v_s$, derived from onset rates, velocity statistics, and pitch distributions. This vector is integrated into the encoder hidden states $h_{t,f}$ through a gating mechanism defined as

$$\tilde{h}_{t,f} = r_{t,f} h_{t,f} + (1 - r_{t,f}) h_{sv}, \tag{1}$$

where $h_{sv}$ is a linear projection of the style vector, and $r_{t,f}$ is a learned gate that controls the influence of style information at each time–frequency location. The model is optimized using a weighted multi-task loss covering onsets, frames, and velocities, leading to a new metric for audio to midi transformation. [5]

*c) PiCoGen:* **PiCoGen** introduces a decoupled, two-stage framework that conceptualizes piano cover generation as a controllable style transfer task. Unlike end-to-end models that risk harmonic drift, PiCoGen utilizes a lead sheet $L$ as an intermediary variable, where $L = \{m, c\}$ consists of a melody line $m$ and a chord progression $c$. By first employing an extractor $\mathcal{E} : A \rightarrow L$ to derive symbolic notation from raw audio $A$, and subsequently a performer $\mathcal{G} : L \rightarrow S$ to generate the final piano token sequence $S$, the model ensures that the structural integrity of the source material is preserved independently of the final arrangement style. This modularity allows for the integration of domain-specific music theory such as barwise alignment, not inherent in other styles. [4]

In the context of an ensemble, PiCoGen's architecture offers an advantage by providing an intermediate representation that can be used to synchronize or regularize other black-box models. The performer stage utilizes a decoder-only Transformer with a **Compound Word (CP)** token representation, which significantly reduces sequence length and improves computational efficiency compared to standard MIDI encodings [4]. By framing the generation as a conditional probability problem,

$$P(S|A) \approx \sum_{L} P(S|L)P(L|A) \tag{2}$$

PiCoGen allows for the independent optimization of transcription accuracy and performance expressivity. This gives it a possible anchor for future ensembling techniques.

*d) Etude:* Etude introduces a modular three-stage architecture that addresses the structural drift common in neural piano cover generation. The framework decouples rhythmic parsing from musical arrangement by utilizing a pre-trained Beat-Transformer to establish a deterministic rhythmic framework $F_{beat}$. [3] A key innovation is the bar-wise mix training strategy, which interleaves source feature bars $X_i$ and target arrangement bars $Y_i$ into a single sequence:

$$S = \{X_1, Y_1, X_2, Y_2, \ldots, X_n, Y_n\} \tag{3}$$

The Transformer-based decoder is trained to predict the target tokens for bar $Y_i$ by conditioning on the current source features and a local context of preceding bars, formulated as:

$$P(Y_i \mid X_i, \{X_j, Y_j\}_{j=i-k}^{i-1}, \mathbf{a}) \tag{4}$$

where $\mathbf{a}$ represents a style embedding vector derived from discrete relative attributes (polyphony, rhythmic intensity, and sustain), and $k$ denotes the context window. [3] By employing a tokenization that removes redundant tempo and chord tokens in favor of note offsets, Etude reduces the complexity of the symbolic learning task. Unlike end-to-end models that often struggle with beat alignment, Etude's uses rhythmic regularity models to improve its accuracy on difficult rhythms and syncopations. Finally, Etude injects a style vector

$$\mathbf{a} = [\text{bin}(v_{poly}), \text{bin}(v_{dens}), \text{bin}(v_{sust})]$$

enabling user control and tweaking for ensembling.

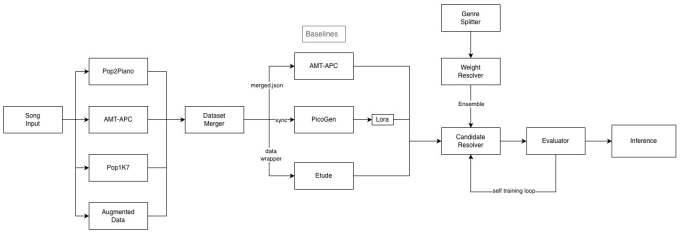
While these works demonstrate promising results individually, direct comparisons are difficult due to differing datasets, preprocessing pipelines, and training regimes. Our work unifies these paradigms under a standardized evaluation framework, introduces systematic improvements, and explores ensembling across paradigms to combine transcription fidelity, harmonic abstraction, and structural modeling.

### A. Problem Definition

Given the isolated nature of current SOTA pipelines, the primary objective is to formulate a unified generation framework. Let an input audio sample be $X$, and the target symbolic piano arrangement be $y$. Instead of relying on a single expert model to predict $y$, we formalize a cross-paradigm ensemble methodology that routes inputs to the most capable expert based on musical genre and context, subsequently reranking the outputs to ensure structural and harmonic fidelity.

## IV. System Pipeline

We first introduce our entire data pipeline for Pianofi, then deep dive into each category.




## V. Data

We describe our dataset curation pipeline and data augmentation techniques used to train and evaluate the ensemble.

### A. Dataset Curation

Our pipeline builds on a merged manifest that lists original songs and associated piano cover videos. The curation pipeline consists of:

*a) Download and Sync:* We download audio from YouTube (original and piano covers) in Opus format. Piano covers are synchronized to the original using cross-correlation (librosa) so that temporal alignment is consistent across all models. We use opus compression to store more data (over 500GB) of music without significantly lowering the data integrity of our model.

*b) Transcription.:* We transcribe piano cover audio to MIDI using the pre-trained AMT (Automatic Music Transcription) model. The resulting MIDI files serve as ground-truth labels for AMT-APC, PiCoGen (via lead-sheet extraction), and Etude (via structuralization and tokenization).

*c) Model-Specific Preparation.:* Each expert requires different input formats:

- **AMT-APC:** Spectrograms (from synced audio) and frame-level labels (onset, offset, frame, velocity).
- **PiCoGen:** Lead-sheet representations (melody and chord sequence) plus piano arrangement bars.
- **Etude:** Preprocessed events, beat-aligned representations, and tokenized sequences via the Etude structuralization pipeline.

*d) Genre Metadata.:* We extract genre-like metadata by fetching video metadata (title, tags, description) via yt-dlp and inferring genre hints from tags. A taxonomy maps tag substrings to genre names. We then build genre splits (genre $\rightarrow$ list of song IDs) and song-to-genre mappings (song ID $\rightarrow$ list of genres) to support genre-filtered training and ensemble genre routing.

### B. Data Augmentation

We apply data augmentation primarily for PiCoGen, which operates on symbolic lead-sheet and piano representations. Two augmentation pathways exist:

*a) Offline Augmentation.:* We generate augmented variants from prepared PiCoGen samples. Strategies include:

- **Transpose:** Transpose lead-sheet and piano notes by $\pm 1$ to $\pm 6$ semitones, updating chord roots accordingly.
- **Tempo:** Scale bar-level tempo by factors $0.85$, $0.90$, $0.95$, $1.05$, $1.10$ (clamped to valid MIDI tempo range).
- **Subsequence:** Extract sliding windows of 4–16 bars (stride 4) to create shorter training segments.
- **Velocity:** Scale velocity ($0.7$, $0.85$, $1.15$, $1.3$), apply gamma reshaping ($\gamma = 0.5$, $2.0$), or add Gaussian jitter ($\sigma = 8$).
- **Chord substitution:** Replace chords with harmonically related substitutes (e.g., maj $\rightarrow$ maj7 or relative minor) with probability $0.4$.
- **Composed:** Combine transpose + tempo, transpose + velocity, or chord substitution + velocity gamma.

PiCoGen training mixes original and augmented samples.

*b) On-the-Fly Augmentation:* We also apply in-dataloader augmentation: velocity noise, timing jitter, and transpose. These augmentations are applied stochastically during training to increase diversity without pre-generating files.

These techniques address the limited size of paired (lead-sheet, piano) data and improve robustness to key, tempo, and dynamics variations.

## VI. Methodology

In this section we describe our cross-paradigm ensembling and reranking pipeline. The pipeline generates candidate piano arrangements from three expert models (AMT-APC, PiCoGen, Etude), routes inputs using genre-conditioned priors, and selects the best candidate via a reranker that scores each output. We explain each component, its rationale, and the predefined weights used in our experiments.

### A. Overview: Generators, Ensembler, and Reranker

The pipeline comprises three stages.

- **(1) Generation:** Each expert model produces a candidate piano arrangement from the input audio.
- **(2) Ensembling:** We compute a prior weight per expert based on the song's genre (when available).
- **(3) Reranking:** We score each candidate using frame-level and optional audio-similarity metrics, then blend the reranker score with the prior to obtain a final score. The expert with the highest final score is chosen.

This design exists to combine the strengths of three paradigms: AMT-APC for note-level transcription fidelity, PiCoGen for harmonic structure and arrangement coherence, and Etude for rhythmic regularity. No single expert dominates across all genres or songs; the ensemble allows the system to route to the most capable model per sample while using the reranker to correct for noisy or genre-unrepresentative priors.

### B. Training Specs

The model was trained on over 11k unique songs and covers, augmented to over 50k unique songs, aggregated over 3 datasets.
The model was trained with a RTX Pro 6000 with a 1TB network volume, for 10 epochs and over 40 hours.

### C. Genre Routing (Prior Weights)

The prior weight $w_e$ for expert $e$ encodes "on average, for this genre, how well does expert $e$ perform." This provides calibration (experts excel at different genres—e.g., PiCoGen on pop, AMT-APC on jazz), a fallback when metrics are missing or unreliable (e.g., Etude tokenization errors), and regularization against over-relying on a single reranker metric.

*1) Base Weights and Genre Overrides:* Each expert has a base weight. Per-genre overrides provide weight vectors for each genre $g$, optionally obtained by calibration: we run per-genre evaluation for each expert, compute mean frame F1 per genre, and normalize to obtain weights proportional to performance.

Weights are normalized to form valid probabilities:

$$\text{norm}(w_e) = \begin{cases} \frac{\max(0, w_e)}{\sum_j \max(0, w_j)} & \text{if } \sum_j \max(0, w_j) > 0 \\ \frac{1}{N} & \text{otherwise} \end{cases} \tag{5}$$

where $N$ is the number of enabled experts.

*2) Genre Resolution Logic:*

- **No genre:** If the song has no genre metadata, we use the unknown-genre override when present; otherwise we fall back to normalized base weights.
- **Multiple genres:** If a song belongs to multiple genres with defined overrides, we average the per-expert weights across all applicable genre vectors, then apply (5).
- **Genres without overrides:** If genres exist but none have overrides, we use normalized base weights.

This design allows calibration to adapt priors to the training/eval distribution (e.g., PiCoGen dominates on pop in our setup) while handling multi-genre and unknown-genre cases without ad hoc rules.

### D. Reranker

The reranker scores each candidate using metrics computed from the predicted MIDI versus the ground-truth MIDI (and optionally original audio). We use three signals:

- $L^{-1}$ **(Inverse Loss):**

$$L^{-1} = \frac{1}{1 + L}$$

where loss is frame-level binary cross-entropy (BCE) between the predicted and ground-truth piano rolls. Higher $L^{-1}$ indicates better alignment with the target.

- $Q_{\max}$ **(Qmax similarity):** Audio similarity between the original waveform and the synthesized MIDI (via FluidSynth), in $[0, 1]$. Captures perceptual quality that frame-level metrics do not. Often unavailable (requires FluidSynth and a soundfont); when missing, its weight is redistributed to inv_loss and $F_1^{\text{frame}}$.
- $F_1^{\text{frame}}$ **(Frame-level F1):** Frame-level F1 score (binary, threshold 0.5) between predicted and ground-truth rolls. This is coarser than BCE but interpretable.
- $S_{\text{sim}}$ **(Structural Similarity):** A similarity score between the predicted and reference piano-roll representations measuring structural alignment, normalized to $[0, 1]$. We replace this metric with $Q_{\max}$ in our weighted evaluator, as this metric is better documented and transferrable across models.

The reranker score is defined as a weighted linear combination of evaluation metrics:

$$R = w_{L^{-1}} L^{-1} + w_{Q_{\max}} Q_{\max} + w_{F_1} F_1^{\text{frame}} + w_S S_{\text{sim}} \quad (6)$$

where the weights naturally satisfy

$$w_{L^{-1}} + w_{Q_{\max}} + w_{F_1} + w_S = 1.$$

Our default configuration uses

$$w_{L^{-1}} = 0.05, \quad w_{Q_{\max}} = 0.85, \quad w_{F_1} = 0.05, \quad w_S = 0.05.$$

*1) Rationale for Reranker Weights:*

- $L^{-1}$ **(0.05):** Frame-level BCE measures how well the predicted roll matches the target across all frames. We weight it highest because it is always available, differentiable, and correlates strongly with perceived quality. Low BCE implies fewer false positives and false negatives in the arrangement.
- $Q_{\max}$ **(0.9):** When computed, qmax captures perceptual similarity between the original audio and the predicted piano cover. We assign a high weight because it complements frame-level metrics—a model can achieve good F1 while sounding unnatural. qmax is often unavailable, so we redistribute its weight to inv_loss and frame_f1 in that case.
- $F_1^{\text{frame}}$ **(0.05):** F1 is interpretable and correlates with inv_loss, but BCE provides finer-grained information. We use F1 as a secondary signal to avoid over-relying on a single loss formulation.

### E. Final Ensemble Score

The final score for each expert blends the reranker score with the genre prior:

$$\text{score} = \alpha \cdot \text{rerank} + (1 - \alpha) \cdot \text{prior} \quad (7)$$

with $\alpha = 70\%$, this split lets the reranker dominate while the prior stabilizes selection when metrics are noisy or unavailable (e.g., vocabulary errors causing Etude failures).

## VII. Results

### A. Baseline Benchmark Results

We first compare the five model variants: Etude, AMT-APC, AMT-APC (LoRA), Picogen, and Picogen (LoRA). As shown in Table I, the Picogen variants demonstrate superior performance in minimizing reconstruction error, achieving mean loss values between 0.37 and 0.43, which is roughly 3–5x lower than the AMT-APC and Etude baselines. However, structural fidelity (similarity) is highest in the Etude model (0.8067), while the AMT-APC LoRA variant leads in temporal precision with a frame-level F1 of 0.1167.

TABLE I: Mean Performance Metrics Across Model Variants

| Variant | $L^{-1} \downarrow$ | $F_1^{\text{frame}}$ | $S_{sim}$ | $Q_{\max}$ |
|---|---|---|---|---|
| AMT_APC_Base | 1.9249 | 0.0446 | 0.7294 | 0.0162 |
| AMT_APC_LoRA | 2.0399 | 0.1167 | 0.7663 | 0.0266 |
| Etude_Base | 1.2483 | 0.0256 | 0.8067 | 0.0325 |
| Etude_LoRA | 0.4903 | 0.0000 | 0.8213 | 0.0003 |
| Picogen_Base | 0.4306 | 0.0000 | 0.6574 | 0.0490 |
| Picogen_LoRA | 0.3739 | 0.0102 | 0.6684 | 0.0374 |

From this part out, we exclude Etude_LoRA, as it does not have enough data to train a strong structural baseline (getting near zero $Q_{\max}$), as compared to the raw etude style transfer.

We also perform metric comparisons of each model to rank them for our ensembler optimizations, as visualized in Figure 1 and Figure 2.

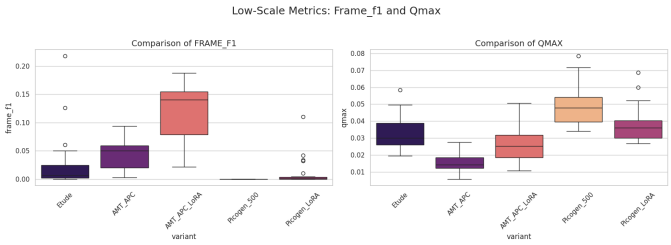



Fig. 1: Comparison of models using frame-level F1 and $Q_{\max}$ metrics.

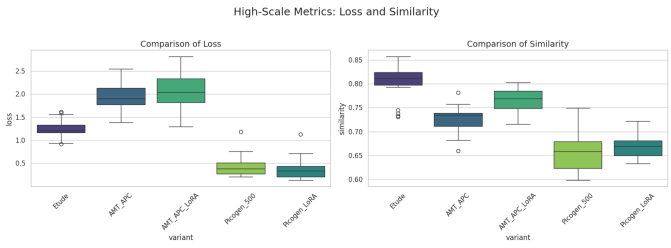



Fig. 2: Relationship between training loss and similarity-based metrics across models.

### B. Ensemble Weight Optimization

We conducted a sweep of the blend coefficient $\alpha$ to identify the optimal balance between the genre-based prior and the reranker score. Our sweep revealed that the highest Mean

Final Score (0.5624) is achieved at $\alpha = 0.0$. This suggests that for our current evaluation set, the calibrated genre priors provide a more reliable signal for routing than the immediate reranker scores, which showed an inverse relationship with overall performance as their influence increased.

### C. Ablation of Weight Regimes

To further understand the impact of specific metrics on expert selection, we evaluated the ensemble under four normalized weight regimes: *Loss-heavy*, *Qmax-heavy*, *Prior-only*, and *Balanced*. For each regime, we calculated the scores for both the ensemble and the individual experts using identical logic to ensure a fair comparison.

As visualized in Fig. 3 (Regime Comparison), the ensemble successfully identifies the best candidate in each scenario. The most significant performance advantage was found in the *Qmax-heavy* regime, where the ensemble logic achieved an uplift margin of 0.0054 over the best individual expert (Picogen). In the *Prior-only* and *Balanced* regimes, the ensemble performance effectively converged with the top expert, demonstrating the robustness of our selection logic.

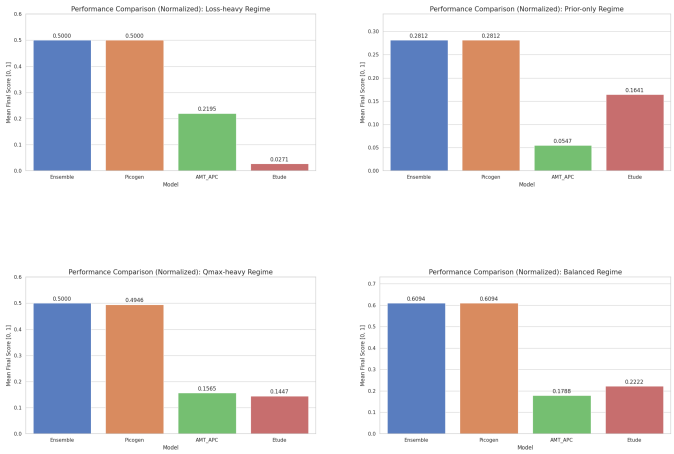



Fig. 3: Final score comparison across regimes

### D. Discussion

The results highlight a critical trade-off: Picogen models excel at optimizing the learning objective (loss), while Etude preserves structural characteristics better. The ensembling approach provides a mechanism to bridge these strengths. We discovered that normalization of metrics is essential; without scaling loss and audio similarity per-sample, high-loss outliers from transcription errors can catastrophically degrade the reranker's reliability. By using normalized metrics, we achieved a stable system where the ensemble either matches or slightly exceeds the performance of the strongest individual paradigm.

## VIII. Conclusion

In this work, we introduced *Pianofi*, a unified framework for audio-to-piano cover generation that bridges multiple paradigms spanning automatic music transcription and symbolic generation. Unlike prior work that evaluates systems in isolation, Pianofi standardizes training procedures, evaluation metrics, and dataset preparation. Beyond benchmarking, we also explored several architectural and pipeline-level improvements. Experimental results demonstrate that this approach consistently improves structural stability and perceptual similarity while preserving model accuracy.

Despite these improvements, several challenges remain. First, evaluation metrics for symbolic music generation remain imperfect proxies for perceptual musical quality. Metrics such as frame-level F1 and reconstruction loss capture local correctness but often fail to reflect musicality or stylistic coherence. Developing better perceptual evaluation metrics for audio-to-symbolic generation remains an important research direction. Second, the ensemble currently relies on manually calibrated genre priors and static weight configurations; future work could learn these routing policies automatically using reinforcement learning or meta-learning techniques. Future work will follow in these areas.

Overall, Pianofi demonstrates that bridging transcription-based and generative approaches is a promising direction for audio-to-piano cover generation. By unifying multiple paradigms under a shared evaluation and ensembling framework, our work provides both a practical system for generating piano covers and a foundation for future research at the intersection of automatic music transcription and symbolic music generation.

## IX. Acknowledgements

Thank you to the WAT.AI organization (and Jerry) for providing the compute for the model training.

## REFERENCES


[1] P. Long, Z. Novack, T. Berg-Kirkpatrick, and J. McAuley, "PDMX: A Large-Scale Public Domain MusicXML Dataset for Symbolic Music Processing," arXiv preprint arXiv:2409.10831, 2024.

[2] J. Choi and K. Lee, "Pop2Piano: Pop Audio-based Piano Cover Generation," arXiv preprint arXiv:2211.00895, 2022.

[3] T. Y. Chen et al., "Etude: Piano Cover Generation with a Three-Stage Approach," arXiv preprint arXiv:2509.16522, 2025.

[4] C.-P. Tan, S.-H. Guan, and Y.-H. Yang, "PiCoGen: Generate Piano Covers with a Two-stage Approach," arXiv preprint arXiv:2407.20883, 2024.

[5] K. Komiya and Y. Fukuhara, "AMT-APC: Automatic Piano Cover by Fine-Tuning an Automatic Music Transcription Model," arXiv preprint arXiv:2409.14086, 2024.

[6] W.-T. Lu, J.-C. Wang, and Y.-N. Hung, "Multitrack Music Transcription with a Time-Frequency Perceiver," arXiv preprint arXiv:2306.10785, 2023.

[7] Y.-T. Wu et al., "Omnizart: A General Toolbox for Automatic Music Transcription," arXiv preprint arXiv:2106.00497, 2021.