AI Harmonizer: Expanding Vocal Expression with a Generative Neurosymbolic Music AI System

Lancelot Blanchard
lancelot@media.mit.edu
MIT Media Lab
Cambridge, MA, USA

Cameron Holt
camholt@mit.edu
Massachusetts Institute of Technology
Cambridge, MA, USA

Joseph A. Paradiso
joep@media.mit.edu
MIT Media Lab
Cambridge, MA, USA

Abstract

Vocals harmonizers are powerful tools to help solo vocalists enrich their melodies with harmonically supportive voices. These tools exist in various forms, from commercially available pedals and software to custom-built systems, each employing different methods to generate harmonies. Traditional harmonizers often require users to manually specify a key or tonal center, while others allow pitch selection via an external keyboard–both approaches demanding some degree of musical expertise. The AI Harmonizer introduces a novel approach by autonomously generating musically coherent four-part harmonies without requiring prior harmonic input from the user. By integrating state-of-the-art generative AI techniques for pitch detection and voice modeling with custom-trained symbolic music models, our system arranges any vocal melody into rich choral textures. In this paper, we present our methods, explore potential applications in performance and composition, and discuss future directions for real-time implementations. While our system currently operates offline, we believe it represents a significant step toward AI-assisted vocal performance and expressive musical augmentation. We release our implementation on GitHub.[1]

Keywords

Vocal Harmonizing, Music, Accompaniment, Machine Learning, Artificial Intelligence, Generative AI

1 Introduction & Previous Work

Vocal harmonizers have long been a valuable tool for vocalists, enabling real-time harmonization and multi-voice effects. Over the years, various hardware solutions have been developed, with TC-Helicon leading the commercial market in harmonization pedals. These pedals generally require the user to set the key in which the system can generate notes, either manually or through another audio input (e.g., with a guitar playing chords). Vocal harmonizers have also been developed as part of research projects. A notable example is Jacob Collier's vocal harmonizer, developed by Ben Bloomberg [4], which allows him to use a keyboard to decide the harmonic texture of the output and provide a vocal melody as input. In parallel, the automation of melodic harmonization, which alleviates the need for a keyboard or a manual setting of a key, has been explored in numerous ways, with methods such as probabilistic modeling [14, 15], dynamic programming [25], and weighted pitch context vectors [11].

In recent years, machine learning has driven significant advancement in generative music. In melody harmonization, projects such as Google's CoCoNet [8] and the Blob Opera [12] have demonstrated the potential of deep learning for polyphonic music generation, alongside many other systems [6, 13, 18, 19, 21, 24]. The application of Transformer-based models–originally designed for natural language processing–to symbolic music generation [9, 20] has opened new possibilities for automatic accompaniment and real-time applications [3, 23, 26]. Additionally, neural networks have shown remarkable success in voice synthesis, with architectures such as VITS [10] and HuBERT [7] enabling the efficient manipulation of vocal timbre and style.

However, and to the best of our knowledge, no end-to-end Machine Learning-based system has been proposed for automatic vocal harmonization. Through this work, we propose our architecture, enabling the harmonization of any input vocal line as a four-part harmony chorale, and discuss our results.

2 Methodology

Our approach for automatic voice harmonization uses a few different existing models, and adds some important logic to connect them together and create a powerful end-to-end framework. We make use of three different model architectures:

- Basic Pitch [2], a model developed and trained by Spotify that can perform automatic music transcription;
- Anticipatory Music Transformer (AMT) [20], a variant of the Music Transformer architecture [9] that enables better compositions through anticipation mechanisms; and
- Retrieval-based Voice Conversion (RVC)[2], a model for singing voice conversion based on the VITS architecture that provides a toolset for fast and accurate singing voice synthesis with pitch and speaker conditioning.

We combine these architectures to accomplish four sequential tasks:

(1) First, we use Basic Pitch to convert our vocal melody to MIDI;
(2) Then, we use a custom-trained AMT model to generate a four-part harmony based on our input melody;
(3) We then extract the fundamental frequency (f_0) information from our vocal melody and shift it to fit our three new parts;
(4) Finally, we use RVC to synthesize these three new vocal lines and add them to our input melody.

An overview of our architecture is displayed on Figure 1. We describe each step of our process in this section.

[1] Our implementation is available at https://github.com/mitmedialab/ai-harmonizer-nime2025.

[2] RVC is available at https://github.com/RVC-Project/Retrieval-based-Voice-Conversion.

NIME '25, June 24–27,2025, Canberra, Australia

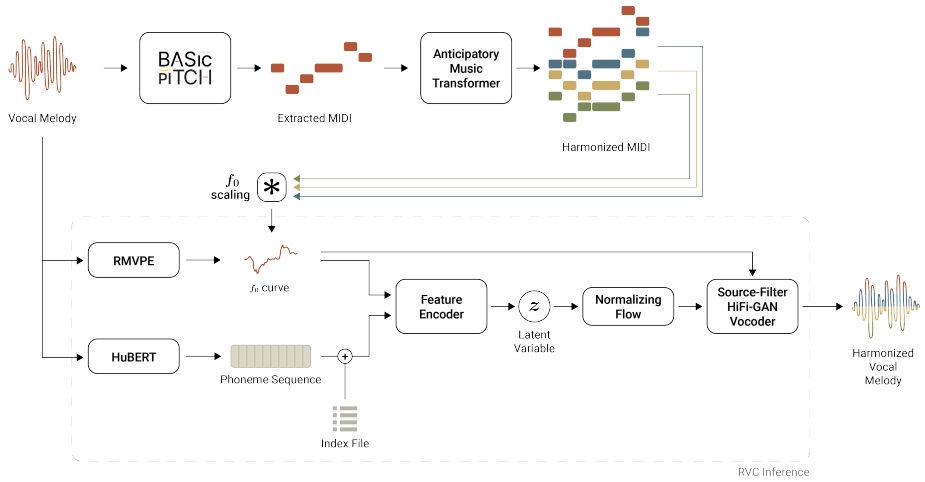


Figure 1: The detailed architecture of our system at inference time.

2.1 Voice-to-MIDI Conversion

To integrate our vocal melody into our custom-trained AMT model, we first need to convert the vocal line into MIDI. This is a relatively straightforward step, accomplished using *Basic Pitch*, an open-source pitch detection tool developed by Spotify. While numerous pitch-tracking tools exist for this purpose, we chose Basic Pitch for its robustness in handling real-world singing inputs and its superior transcription accuracy. Extracting MIDI from vocal melodies is particularly challenging, as human singing often deviates from the strict 12-tone equal temperament required for MIDI representation. In our experiments, Basic Pitch appeared to handle vocal inputs fairly well. It is important to note that this approach comes with inherent limitations, as it restricts harmonization to Western tonal frameworks and favors cleanly sung inputs.

2.2 Harmony Generation with AMT

Once the vocal melody is transcribed into MIDI, we use a custom-trained Anticipatory Music Transformer model (AMT) to generate a four-part harmony that will be used to synthesize new vocal lines. AMT is particularly well-suited for this type of harmonic composition since it can *anticipate* future notes in the melody and generate more thoroughly crafted harmonic lines.

2.2.1 AMT Training. In order to generate accurate harmonies for our vocal melody, we specifically train an AMT model on the task of four-part harmony. For this purpose, we base our training on the pre-trained `music-medium-800k`[3] model trained by the authors of the original paper on the Lakh MIDI Dataset [16] for 800,000 epochs. We then choose to fine-tune our model on the **JSB Chorales** dataset [1, 5], a corpus of 382 four-part harmonized chorales by J.S. Bach. Although this imposes an important genre restriction on the harmonies that our harmonizer can generate, this dataset is particularly interesting since all of the chorales are written with four distinct voices: *Soprano*, *Alto*, *Tenor*, and *Bass* (SATB). For the purpose of our training, we convert the original dataset to distinct MIDI files, and use MIDI instruments `0`, `1`, `2`, and `3` to represent all four voices.[4] Due to the fact that the data of the JSB Chorales dataset is most likely already contained in the Lakh MIDI Dataset, the model quickly overfits and we use early stopping to avoid model degradation.

2.2.2 AMT Inference. Anticipatory Music Transformers introduce the mechanism of *anticipation*, which allows for the conditioning of a temporal point process on the realizations of another correlated process. Following the AMT naming convention, the main temporal point process is called the *event* process, while the conditioning process is called the *control* process. To perform this conditioning, AMT interleaves events $\mathbf{e}_{1:N}$ and controls $\mathbf{u}_{1:K}$ in such a way that a control $\mathbf{u}_k$ on time s_k ends up close to events near time $s_k - \delta$, with δ being the *anticipation interval*. Using the results of the original paper, we use $\delta = 5$ seconds.

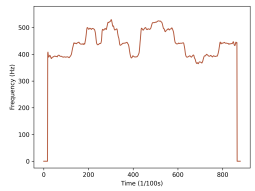
AMT's tokenization of MIDI notes allows us to precisely control the model generation. In particular, the model encodes a MIDI note as a triplet of time, duration, and note $(\mathbf{t}_i, \mathbf{d}_i, \mathbf{n}_i)$, which allows us to guide the model generation to ensure that it can generate a well-founded four-part harmony. To do so, we enforce that each voice (MIDI instruments 1, 2, and 3) can only generate *one* harmony note for each note present in the input melody. We also ensure that the onset time and duration of each note overlap, by forcing the time token $\mathbf{t}_i$ and duration token $\mathbf{d}_i$ to take the value of the time and duration of the corresponding control. We do so by manually selecting the time and duration tokens, and by performing sampling for the note token on a restrained logit distribution, with the logits for notes from other instruments set to $-\infty$.

2.3 MIDI-to-Frequency Conversion

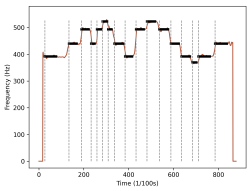
Once we have the MIDI information for our *Alto*, *Tenor*, and *Bass* lines, we can start generating new vocal lines. To do so, we first need to extract the pitch contour (or fundamental frequency f_0) of our input melody. Although this information is also present in the MIDI data, the f_0 is a much more fine-grained measure that also contains the pitch fluctuations within notes, as well as the transition between notes. RVC provides a selection of algorithms and models for the purpose of pitch extraction. Among those, we choose to use *RMVPE* [22] for its robustness and execution speed.

Once the f_0 information is extracted from our original audio, we can start shifting it to match the pitch contour of our three

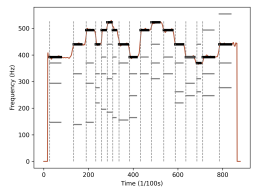
[3]The original checkpoint is available on Huggingface at https://huggingface.co/stanford-crfm/music-medium-800k.
[4]We provide our version of the dataset on GitHub: https://github.com/lancelotblanchard/JSB-Chorales-dataset-midi.



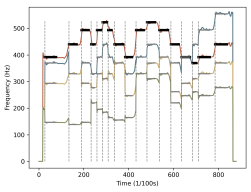
(a) We first extract the f_0 using RVMPE.



(b) Based on our MIDI data computed by Basic Pitch, we segment the f_0 curve by looking at each note onset.



(c) We obtain a four-part harmony from the Anticipatory Music Transformer.



(d) We shift the f_0 curve for each note of every voice, and obtain four distinct curves.

Figure 2: Segmentation and f_0 shifting process for our automatic harmonization.

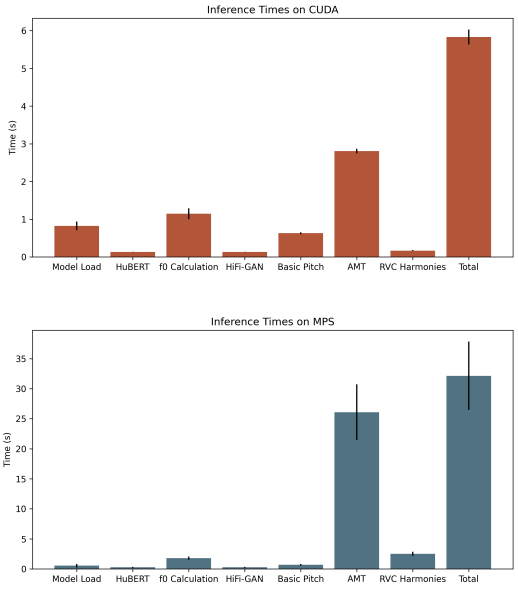


Figure 3: Comparison of inference times on machines running CUDA and MPS, with a 10-second audio input.

harmony voices. Our approach for this task is straightforward: We first detect the onset of each note and delimit our f_0 curve based on those points, then shift the input curve.

Formally, given an input curve f_0^{in} and a sequence of N MIDI notes $\mathbf{e}_{1:N}$ with $\mathbf{e}_i = (\mathbf{t}_i, \mathbf{h}_i)$ (where we simplify the original AMT notation and consider that $\mathbf{t}_i$ and $\mathbf{h}_i$ respectively refer to the onset time of the note and the difference in semitones between the input note and the harmony voice), we have:

$$f_0^{out} = \begin{cases} f_0^{in}(t) \cdot 2^{\mathbf{h}_1/12}, & t_1 \le t < t_2 \\ f_0^{in}(t) \cdot 2^{\mathbf{h}_2/12}, & t_2 \le t < t_3 \\ \vdots & \\ f_0^{in}(t) \cdot 2^{\mathbf{h}_N/12}, & t_N \le t \end{cases}$$

An overview of our pitch shifting approach is presented on Figure 2.

2.4 Voice Synthesis with RVC

We finally perform voice synthesis using the new f_0 curves calculated as described above for each harmonic voice. For this step, RVC requires us to use a pre-trained vocal model that contains an index file of HuBERT embeddings, as well as weights for the feature encoder, normalizing flow, and HiFi-GAN vocoder as shown in Figure 1. This requires us to train the user's voice model beforehand. Once this model is trained, we can use both the modified f_0^{out} curve for each voice as well as the extracted HuBERT embeddings for the input audio to condition the synthesis of each vocal line. With this pitch conditioning, RVC allows us to preserve the formant and timbre of the original audio, while following the provided pitch information, thus producing natural-sounding harmonies that maintain the characteristics of the original singer.

Once the audio is generated for every additional vocal line, we can add them together and retrieve the final harmonized vocal line.

3 Results & Discussion

We tested our system on a variety of audio inputs and obtained highly convincing results. While the system demonstrates significant power, its complexity poses challenges for real-time adaptation as a musical instrument. To facilitate future research on adapting similar systems into real-time performance settings, we provide the results of our experiments, focusing on inference speed across different hardware configurations.

Our tests were conducted on two machines: An RTX 4090 GPU Machine running Ubuntu 22.04 and a M3 Max MacBook Pro running macOS 14.5. Figure 3 presents the inference time of each system component. Notably, our results indicate that inference on the CUDA-powered machine is significantly faster than on the MPS-based MacBook. This discrepancy appears to stem from a known issue in the PyTorch library, where iterative inferences–essential for autoregressive models like AMT–lead to substantial memory leaks, ultimately causing performance slowdowns.[5]

Despite this, our findings are encouraging: on the CUDA system, a full iteration of the model for a 10-second audio input completes in under six seconds on average. Achieving similar performance on the MacBook Pro may be possible by optimizing AMT inference times. The second most time-consuming step is the f_0 calculation, which could likely be accelerated by replacing RMVPE with a more efficient model, like PESTO [17]. Further improvements could be gained by training a unified model capable of simultaneously predicting both the f_0 and a MIDI transcription from the audio input.

Looking ahead, we envision this technology playing a crucial role in real-time musical performance. While further optimizations are necessary, real-time pitch tracking systems and improvements to the Music Transformer architecture could enable harmonization at speeds faster than real-time.

[5]See the open PyTorch issue on GitHub: https://github.com/pytorch/pytorch/issues/91368.

4 Conclusion

In this paper, we introduced the AI Harmonizer, a novel system capable of autonomously generating four-part vocal harmonies without user-provided harmonic input. By leveraging state-of-the-art AI models, our framework successfully arranges input melodies into rich choral textures. Our experimental results demonstrate the effectiveness of our approach in producing musically coherent harmonies that preserve the vocal characteristics of the original singer. Additionally, our performance analysis across different hardware configurations highlights the system's potential for real-time application, particularly with optimizations in pitch tracking and inference processes. Despite the system currently operating offline, these advancements represent a significant step toward AI-assisted vocal performance and expressive musical augmentation. Future work will focus on refining the harmonization process for broader musical contexts beyond SATB chorales and exploring interactive performance applications. We also intend to investigate the potential for machine learning models that further bridge the gap between symbolic and audio representations. By continuing to push the boundaries of AI in music, we hope to empower artists with innovative tools that expand creative possibilities in both composition and live performance.

5 Ethical Standards

There are no observed conflicts of interest. The research was funded using discretionary funding and used lab-owned compute power for the training of the model. Consent by the vocal performer was obtained before training a voice model and distributing the vocal audio recordings.

Acknowledgments

We are grateful to Nancy Rosenberg for providing the vocal samples we used to train and test our system.

References

[1] Moray Allan and Christopher Williams. 2004. Harmonising Chorales by Probabilistic Inference. In *Advances in Neural Information Processing Systems*, L. Saul, Y. Weiss, and L. Bottou (Eds.), Vol. 17. MIT Press. https://proceedings.neurips.cc/paper_files/paper/2004/file/b628386c9b92481fab68fbf284bd6a64-Paper.pdf

[2] Rachel M. Bittner, Juan José Bosch, David Rubinstein, Gabriel Meseguer-Brocal, and Sebastian Ewert. 2022. A Lightweight Instrument-Agnostic Model for Polyphonic Note Transcription and Multipitch Estimation. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Singapore.

[3] Lancelot Blanchard, Perry Naseck, Eran Egozy, and Joseph A. Paradiso. 2024. Developing Symbiotic Virtuosity: AI-Augmented Musical Instruments and Their Use in Live Music Performances. *An MIT Exploration of Generative AI* (Sept. 2024). https://doi.org/10.21428/e4baedd9.69c11de7 Publisher: MIT.

[4] Benjamin Arthur Philips Bloomberg. 2020. *Making Musical Magic Live*. PhD Thesis. MIT.

[5] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. 2012. Modeling temporal dependencies in high-dimensional sequences: application to polyphonic music generation and transcription. In *Proceedings of the 29th International Coference on International Conference on Machine Learning (ICML'12)*. Omnipress, Madison, WI, USA, 1881–1888. event-place: Edinburgh, Scotland.

[6] Yi-Wei Chen, Hung-Shin Lee, Yen-Hsing Chen, and Hsin-Min Wang. 2021. SurpriseNet: Melody Harmonization Conditioning on User-controlled Surprise Contours. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference*. ISMIR, 105–112. https://doi.org/10.5281/zenodo.5624423

[7] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* 29 (Oct. 2021), 3451–3460. https://doi.org/10.1109/TASLP.2021.3122291

[8] Cheng-Zhi Anna Huang, Tim Cooijmans, Adam Roberts, Aaron Courville, and Douglas Eck. 2017. Counterpoint by Convolution. In *International Society for Music Information Retrieval (ISMIR)*.

[9] Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. 2018. Music Transformer. https://doi.org/10.48550/arXiv.1809.04281 arXiv:1809.04281 [cs, eess, stat].

[10] Jaehyeon Kim, Jungil Kong, and Juhee Son. 2021. Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 5530–5540. https://proceedings.mlr.press/v139/kim21f.html

[11] Peter van Kranenburg and Eoin J. Kearns. 2023. Algorithmic Harmonization of Tonal Melodies Using Weighted Pitch Context Vectors. In *Proceedings of the 24th International Society for Music Information Retrieval Conference*. ISMIR, 391–397. https://doi.org/10.5281/zenodo.10265307

[12] Li, David. 2020. Blob Opera. https://experiments.withgoogle.com/blob-opera

[13] Hyungui Lim, Seungyeon Rhyu, and Kyogu Lee. 2017. Chord Generation from Symbolic Melody Using BLSTM Networks.. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*. ISMIR, 621–627. https://doi.org/10.5281/zenodo.1417327

[14] Dimos Makris, Maximos A. Kaliakatsos-Papakostas, and Emilios Cambouropoulos. 2015. Probabilistic Modular Bass Voice Leading in Melodic Harmonisation.. In *Proceedings of the 16th International Society for Music Information Retrieval Conference*. ISMIR, 323–329. https://doi.org/10.5281/zenodo.1416374

[15] Jean-François Paiement, Douglas Eck, and Samy Bengio. 2006. Probabilistic Melodic Harmonization. In *Advances in Artificial Intelligence*, Luc Lamontagne and Mario Marchand (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 218–229.

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

[17] Alain Riou, Stefan Lattner, Gaëtan Hadjeres, and Geoffroy Peeters. 2023. PESTO: Pitch Estimation with Self-supervised Transposition-equivariant Objective. In *Proceedings of the 24th International Society for Music Information Retrieval Conference, ISMIR 2023*. International Society for Music Information Retrieval.

[18] Chung-En Sun, Yi-Wei Chen, Hung-Shin Lee, Yen-Hsing Chen, and Hsin-Min Wang. 2021. Melody Harmonization Using Orderless Nade, Chord Balancing, and Blocked Gibbs Sampling. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 4145–4149. https://doi.org/10.1109/ICASSP39728.2021.9414281

[19] Takuya Takahashi and Mathieu Barthet. 2022. Emotion-driven Harmonisation And Tempo Arrangement of Melodies Using Transfer Learning. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference*. ISMIR, 741–748. https://doi.org/10.5281/zenodo.7316770

[20] John Thickstun, David Hall, Chris Donahue, and Percy Liang. 2023. Anticipatory Music Transformer. https://doi.org/10.48550/arXiv.2306.08620 arXiv:2306.08620 [cs, eess, stat].

[21] Hiroaki Tsushima, Eita Nakamura, Katsutoshi Itoyama, and Kazuyoshi Yoshii. 2017. Function- and Rhythm-Aware Melody Harmonization Based on Tree-Structured Parsing and Split-Merge Sampling of Chord Sequences.. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*. ISMIR, 502–508. https://doi.org/10.5281/zenodo.1416848

[22] Haojie Wei, Xueke Cao, Tangpeng Dan, and Yueguo Chen. 2023. RMVPE: A Robust Model for Vocal Pitch Estimation in Polyphonic Music. In *Interspeech 2023 (interspeech_2023)*. ISCA, 5421–5425. https://doi.org/10.21437/interspeech.2023-528

[23] Yusong Wu, Tim Cooijmans, Kyle Kastner, Adam Roberts, Ian Simon, Alexander Scarlatos, Chris Donahue, Cassie Tarakajian, Shayegan Omidshafiei, Aaron Courville, Pablo Samuel Castro, Natasha Jaques, and Cheng-Zhi Anna Huang. 2024. Adaptive accompaniment with ReaLchords. In *Proceedings of the 41st International Conference on Machine Learning (ICML'24)*. JMLR.org. Place: Vienna, Austria.

[24] Yujia Yan, Ethan Lustig, Joseph VanderStel, and Zhiyao Duan. 2018. Part-invariant Model for Music Generation and Harmonization. In *Proceedings of the 19th International Society for Music Information Retrieval Conference*. ISMIR, 204–210. https://doi.org/10.5281/zenodo.1492383

[25] Li Yi, Haochen Hu, Jingwei Zhao, and Gus Xia. 2022. AccoMontage2: A Complete Harmonization and Accompaniment Arrangement System. In *Proceedings of the 23rd International Society for Music Information Retrieval Conference*. ISMIR, 248–255. https://doi.org/10.5281/zenodo.7316642

[26] Xun Zhou, Charlie Ruan, Zihe Zhao, Tianqi Chen, and Chris Donahue. 2024. Local deployment of large-scale music AI models on commodity hardware. https://doi.org/10.48550/arXiv.2411.09625 arXiv:2411.09625 [cs].