HARP: Bringing Deep Learning to the DAW with Hosted, Asynchronous, Remote Processing

Hugo Flores Garcia[*], **Patrick O'Reilly, Aldo Aguilar, Bryan Pardo**
Department of Computer Science
Northwestern University
Evanston, IL, USA

Christodoulos Benetatos, Zhiyao Duan
University of Rochester
Rochester, NY, USA

1 Introduction

Audio deep learning models can serve as powerful creative tools in a wide variety of applications including singing voice conversion [6], audio source separation [7], automatic mixing [8], and music generation [4]. However, such tools are often developed in research contexts and thus exist in formats (e.g., experimental Python repositories) inaccessible to many members of the creative artist community. By contrast, musicians and other audio creators typically work within Digital Audio Workstation (DAW) software environments, such as Logic or ProTools, because DAWs support idiomatic audio recording and editing workflows. While model developers can use web frameworks, such as Gradio [1], to build interactive interfaces for deep learning models, this interaction paradigm requires users to switch between DAW and browser contexts as they upload, download, and import audio manually, introducing significant friction into the creative process.

The Audacitorch project [3] attempts to bridge this gap by providing a Python framework to incorporate deep learning models into a custom build of the Audacity DAW [2]. This work is DAW-specific and requires models to run locally on the CPU, limiting the ranges of both potential models and users. An alternative to the DAW-specific approach is to develop "plug-in" software that allows running a locally-hosted model in any DAW (e.g., NeuTone [3] and NeuralMidiFx [5]). Implementing cutting-edge deep-learning models as plug-in software within DAWs has proven challenging, as the dominant plug-in formats (e.g., VST) process audio in small chunks (e.g. 15-100 miliseconds of audio) under real-time performance constraints using only the local CPU. This presents significant overhead for model developers to refactor and retrain models to run under these constraints, discouraging adoption and limiting the range of applications to tasks suitable for low-resource real-time processing.

To overcome these limits, we propose HARP[4], a free plug-in that allows for **h**osted, **a**synchronous, **r**emote **p**rocessing with deep learning models by routing audio from the DAW through Gradio endpoints. We provide a lightweight API for building compatible Gradio audio-processing apps with optional interactive controls, enabling model developers to create user interfaces for virtually any audio processing model with only a few lines of Python code. Because Gradio apps can be hosted locally or in the cloud (e.g., HuggingFace Spaces), HARP lets DAW users access large state-of-the-art models with GPU compute from the cloud without breaking their within-DAW workflow, as shown in Figure 1.

[*]Corresponding author: hugofloresgarcia2025@u.northwestern.edu

[2]`https://interactiveaudiolab.github.io/project/audacity.html`

[3]`https://neutone.space`

[4]`https://github.com/audacitorch/pyharp`

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

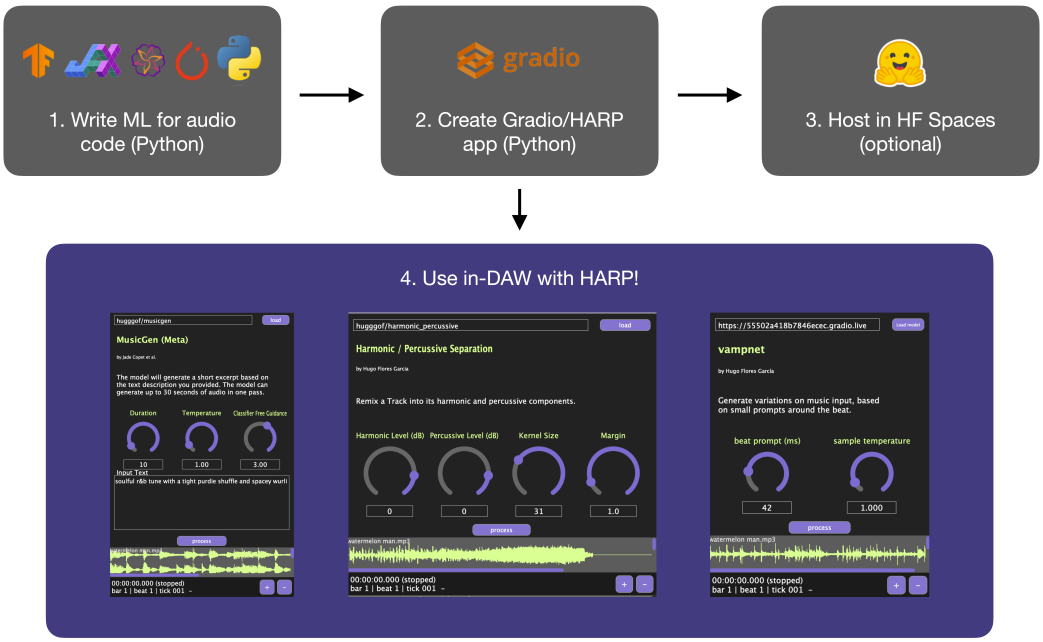

Figure 1: TOP: The steps to make a python deep model HARP compatible. BOTTOM: three example models as seen in the HARP plugin: the Vampnet [4] generative model, the MusicGen [2] generative model, and an HPSS source separator. The end-user just enters the URL of the model's Gradio endpoint or HuggingFace Space in the HARP plug-in. Model controls are automatically populated in the plug-in window. One can then use the model as part of a standard DAW-centric workflow.

2 Method

HARP uses the Audio Random Access (ARA)[5] extension of the popular VST[6] plug-in format. Whereas VST requires realtime-compatible processing of audio in short blocks (e.g., 100 miliseconds), the ARA format allows for processing of entire audio regions (e.g., 10 seconds) from the DAW timeline. Processed regions are stored alongside the original regions and rendered to the timeline in their place, allowing for latency-free playback after the initial processing pass.

To facilitate the development of HARP-compatible Gradio endpoints, we release a simple python library to add HARP capabilities to any Gradio application for processing audio. Virtually any Python function that consumes and produces audio files can be wrapped to serve as a HARP backend. Additional processing inputs such as sliders, text boxes, and conditioning audio can be encapsulated in pre-defined control objects that propagate from the Gradio app to the HARP plug-in interface automatically, allowing for easy interface design. We also include model cards for documentation and search, facilitating the development of an open-source HARP ecosystem. Our plug-in code, API library, and downloadable HARP binaries are available at `https://github.com/audacitorch/pyharp`. We include extensive documentation, and show that popular Gradio audio apps can be converted to HARP endpoints with only a few lines of code.

3 Conclusions

HARP provides a framework to empower musicians and sound artists with cutting-edge deep models for audio. This will allow the creation of music in ways that are not possible in existing DAWs. HARP also provides a path for researchers and developers of deep models to reach end-users without requiring a complete refactor of their models and code to work on a local machine's CPU in real-time. This can potentially transform the relationship between AI researchers and working musicians, bringing them into direct dialog through sharing and using cutting-edge tools for sound creation.

[5] `https://en.wikipedia.org/wiki/Audio_Random_Access`
[6] `https://en.wikipedia.org/wiki/Virtual_Studio_Technology`

Ethical Implications of this Work

Providing an easy way for artists to use cutting-edge deep learning models democratizes deep learning tools for audio manipulation in end-user devices. This work enables access to such tools for users who would otherwise have difficulty accessing them. As such, the work described in this paper could foster an interactive ecosystem of artists and deep learning practitioners. Those with privacy concerns about source material (e.g., artists working on unreleased works) may not be comfortable uploading their audio to a third party (e.g., a cloud-hosted model with Gradio). This can be alleviated by providing a warning to users that web-based model processing requires sending their audio to a remote server. This work also has the potential side effect of facilitating misuse by bad actors (e.g., putting tools that facilitate manipulating recorded speech in the hands of non-technical people that may wish to falsify evidence). Some may argue that access to deep audio manipulation tools should, therefore, be tightly controlled. We argue that the right solution is not to restrict artist access to any new editing tools that may arise. Instead, we encourage the exploration of solutions for detecting and addressing the unethical use of audio manipulation tools (e.g., through watermarking generated audio) while preserving the open, unrestricted access to these systems for creative expression.

Acknowledgments

The authors would like to thank Ryan Devens for the many meaningful conversations, example code and thoughts on audio plugin development and Audio Random Access.

References

[1] A. Abid, A. Abdalla, A. Abid, D. Khan, A. Alfozan, and J. Zou. Gradio: Hassle-free sharing and testing of ml models in the wild. *arXiv preprint arXiv:1906.02569*, 2019.

[2] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez. Simple and controllable music generation. *arXiv preprint arXiv:2306.05284*, 2023.

[3] H. Flores Garcia, A. Aguilar, E. Manilow, D. Vedenko, and B. Pardo. Deep learning tools for audacity: Helping researchers expand the artist's toolkit. In *5th Workshop on Machine Learning for Creativity and Design at NeurIPS 2021*, 2021.

[4] H. Flores Garcia, P. Seetharaman, R. Kumar, and B. Pardo. Vampnet: Music generation via masked acoustic token modeling. In *Conference of the International Society for Music Information Retrieval (ISMIR)*, 2023.

[5] B. Haki, J. Lenz, and S. Jorda. NeuralMidiFx: A Wrapper Template for Deploying Neural Networks as VST3 Plugins. In *Proceedings of the 4th International Conference on on AI and Musical Creativity*, Sept. 2023.

[6] W.-C. Huang, L. P. Violeta, S. Liu, J. Shi, Y. Yasuda, and T. Toda. The singing voice conversion challenge 2023. *arXiv preprint arXiv:2306.14422*, 2023.

[7] N. Schaffer, B. Cogan, E. Manilow, M. Morrison, P. Seetharaman, and B. Pardo. Music separation enhancement with generative modeling. In *Conference of the International Society for Music Information Retrieval (ISMIR)*, 2022.

[8] C. J. Steinmetz, N. J. Bryan, and J. D. Reiss. Style transfer of audio effects with differentiable signal processing. 2022.