# AI-DRIVEN, MOBILE-FIRST WEB UI FOR CONTROLLABLE EXPRESSIVE PIANO PERFORMANCE COMPOSITION

**Théis Bazin**[1,2] **Gaëtan Hadjeres**[1] **Mikhail Malt**[2]
[1] Sony CSL Paris [2] UMR9912 STMS Ircam – CNRS – Sorbonne Université – Ministère de la Culture
theis.bazin@ircam.fr

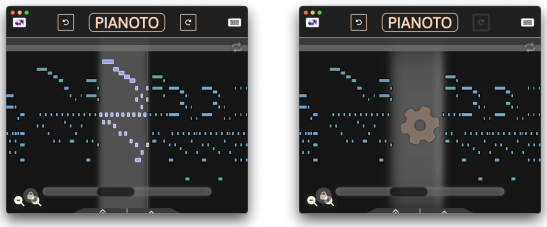
## ABSTRACT

The piano-roll has been the *de facto* standard representation for melodic and harmonic content in DAWs for decades, yet direct manipulation of those requires expert knowledge of music theory to begin with, and additionally becomes physically impractical when switching to the smaller, touch-based screens of modern mobile devices, too coarse for the precision required by micro-timings and the unforgiving discrete placement of pitches. Imbuing these interfaces with machine learning and offloading their precise and error-prone aspects to style-adaptive AI assistants may allow the design of more intuitive interactions whilst maintaining a high level of control, helping lower the cost of entry to composition for novices and offer stimulating new creative tools for professional musicians. We introduce PIANOTO, a touch-ready, responsive web interface for creating expressive piano performances through AI-assisted inpainting, all via simple swipe operations. This open-source, model-agnostic prototype is designed with both novice and expert users in mind, for usage either as a standalone tool or in conjunction with existing DAWs, on desktop or mobile.

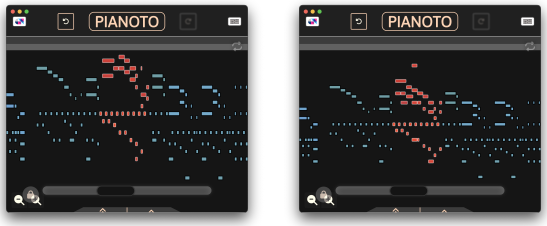
## 1. INTRODUCTION

Computer-assisted music composition, *even without accounting for any sound related aspects*, is an expert task: it brings together music theory, for the construction of melody and harmony, and a practical sense of musicality and rhythm to bring these sequences to life. In addition, most of the current music creation software were designed in the 90s or early 2000s for mouse-based interactions on desktop computers, translating poorly to the current, predominantly mobile-based ecosystem. Indeed, appropriately placing a note on a piano-roll becomes surprisingly error-prone on small touchscreens, inappropriate for balancing the precision required by musical micro-timing and the necessity for large-scale views encompassing the long-term dependencies typically found in music. A solution to these limitations has been the development of *"automatic"* music creation systems such as AIVA, Amper, OrbComposer or the now retired JukeDeck. These tools remove access to the low-level musical representations and only provide high-level, e.g. dropdown-based, controls. As such, they move the interactivity away from the music itself, arguably reducing the user's grip on the musical material.






(a) Selection (b) Selected region cleared




(c) Inpainting result #1 (d) Inpainting result #2

**Figure 1**: Using PIANOTO to generate several variations of a sequence through repeated inpainting of the same zone. The music is from the 3rd Movement of Mozart's Symphony No. 41 in C major. The model successfully maintains the left hand motif and offers melodic variations on the right hand.

## 2. PIANOTO

We propose **PIANOTO** [1], a *model-agnostic web interface* that replaces the precise note-level interactions of traditional piano-rolls with simple temporal selection, as shown on Figure 1. Users can select time slices on the piano-roll that are then regenerated by an *inpainting*-enabled generative model, that is, a stochastic model of the form $p_\theta(\texttt{note}_t|\texttt{notes}_{<t}, \texttt{notes}_{>t})$, able to generate new, *meaningful* notes, accounting for the past and future notes.

[1] Interface code, demo videos and model Docker image (PIA) available at github.com/SonyCSLParis/music-inpainting-ts/.

Late-Breaking / Demo Session Extended Abstract, ISMIR 2022 Conference

### 2.1 Use cases: from novices to professionals

Inpainting enables a wide range of use cases, from millisecond-scale edits to macro-scale (re)generations. Unconditional generation, crucial for novices to obtain valuable initial content, can be achieved by repeatedly generating material from scratch until obtaining a satisfactory starting point. Conversely, expert users can import their own performances as MIDI files or directly record themselves via MIDI-IN and transform those. Users can also drag-and-drop audio files to create personal, expressive variations of existing performances, via built-in transcription [1, 2]. Since every generation step then accounts for the evolving piano-roll content, the users can progressively shape it to their taste by keeping zones which they find valuable and regenerating others. The context ultimately reflects the user's style, accounted for by the model. Inpainting thus enables a form of *on-the-fly user-adaptation* of large models, without requiring expensive retraining.

### 2.2 Design principles: the piano-roll as an instrument

Our aim with PIANOTO is to turn the piano-roll from a static, expert-only *interface* into a playful yet rich *instrument*, as embodied by the "Low entry fee with no ceiling on virtuosity" philosophy of music interface design pioneers Wessel and Wright [3]. The interface is reactive, in that selecting a temporal zone by click-and-drag or swipe on the piano-roll *immediately* triggers an inpainting. Crucially, all interactions happen directly *on the musical representation*, rather than through external controls (e.g., there is no "generate" button). This is in line with recent work, e.g. in the field of 3D modelling: Michel and Boubekeur argue for moving interactions "back into the viewport" [4], for instance by visually dragging parts of a 3D model to move or resize them instead of relying on external sliders.

### 2.3 Backend: model-agnostic, but batteries included

The PIANOTO interface is *model-agnostic* and we encourage researchers and practitioners to connect it to their own models, making it easy to turn any static model into an interactive instrument. PIANOTO was nevertheless designed around a new version [2] of the PIA model by Hadjeres and Crestel [5]. This Transformer-based model was trained for inpainting on GIANTMIDI-PIANO [6], a large corpus of solo classical piano performances obtained by automatic transcription of piano videos. PIA is able to predict pitches, velocities as well as note onsets and durations at a millisecond scale, enabling expressive timing. Interestingly, this new version of PIA also performs well on out-of-domain data and can adapt to more modern playing styles such as jazz, minimalism or pop music, ensuring that PIANOTO can be beneficial for a wide range of users. We note here a limitation: the free-form, solo-piano training of PIA can become impractical when users expect it to adhere to a fixed tempo, e.g. when composing a piano part to include into an existing piece. Imbuing this model with metronomic timing could therefore be the object of future work.

[2] To be published. Model available in the interface's repository.

## 3. DISCUSSION AND RELATED WORKS

Generative AI models of images [7, 8] have become viral in the last months. A significant step has been the open-sourcing of the Stable Diffusion architecture [9], prompting the release of custom forks [10] and dedicated user interfaces [11], including an integration with Photoshop [12]. Inspite of impressive advances in AI-assisted music composition, similar levels of public enthusiasm are not yet to be seen regarding these techniques. This might be attributed to the fact that integrating these techniques into consumer music production processes is arguably more challenging. Indeed, even though these tools crucially require interoperability for integration within musicians' workflows, there is no open standard – as opposed to the availability of the VST standard for e.g. neural sound synthesis [13–18] – for controlling the content of MIDI tracks in DAWs, calling for custom interfaces. Solutions have included either standalone interfaces [19–21] or ad-hoc systems limited to one specific host. The latter have included Max4Live-based Ableton Live plugins [5, 19] or an integration into MuseScore [22]. These integrations can be poorly accessible for research purposes when the host software in question requires users to own a commercial license, as is the case for Ableton Live. Existing standalone interfaces such as the Magenta Studio suite [19] and Calliope [20], target measure-wide generation, thus lacking the ability for short-term generation at the note-level. Closest to ours is the CoCoCo interface [23], which enables note-level editing, yet does not aim for reactivity, instead offering to precisely parameterize the desired output via sliders prior to triggering generations. These interfaces are furthermore essentially button-based, with no reactive interactivity. Web UIs such as Piano Genie [24] and the A.I. Duet [25, 26] system offer intuitive, mobile-ready usage but discard offline, long-term composition in favour of playful, real-time improvisation. PIANOTO seeks to strike a balance between playful, reactive interaction and long-term editing capability for structured music composition.

## 4. TECHNICAL BACKGROUND

Even without considering its inpainting-enhanced generative capabilities, PIANOTO offers a ready-to-use solution for single-instrument, polyphonic MIDI playback and recording in the browser, equipped with seek/loop, velocity-scaled display for improved readability and built-in MIDI-IN and OUT. PIANOTO builds upon previous work by Bazin et al. on inpainting-based music creation interfaces, as part of the `music-inpainting.ts` library [27–29]. PIANOTO uses the same audio engine as these, based on the `Tone.js` Web Audio API framework [30, 31] and the `webmidi.js` library [32, 33]. PIANOTO also uses the Onsets and Frames transcription model and some UI elements from `Magenta.js` [2]. These UI elements, along with the associated `html-midi-player` [34] library, form the basis for the piano-roll visualization. PIANOTO expands on those and turns the static piano-roll of `Magenta.js` into an interactive instrument.

## 5. ACKNOWLEDGEMENTS

This work was supported by the French ANRT through the CIFRE Industrial PhD grant Nr. 2019/0472. The authors thank Martin Leblancs, Matthieu Queru and Sandro Sgro for their help on building a preliminary version of a web interface for the PIA model during a development project hosted at Sony CSL Paris, as part of their curriculum at EPITECH.

Théis Bazin is now with MWM[3], Boulogne-Billancourt, France. Gaëtan Hadjeres is now with SONY AI[4], Zürich, Switzerland.

## 6. REFERENCES

[1] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. H. Engel, S. Oore, and D. Eck, "Onsets and Frames: Dual-Objective Piano Transcription," in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, E. Gómez, X. Hu, E. Humphrey, and E. Benetos, Eds., 2018, pp. 50–57. [Online]. Available: http://ismir2018.ircam.fr/doc/pdfs/19_Paper.pdf

[2] Magenta, "@magenta/music," 2021, published under an Apache-2.0 license. [Online]. Available: https://github.com/magenta/magenta-js/tree/master/music

[3] D. Wessel and M. Wright, "Problems and Prospects for Intimate Musical Control of Computers," *Computer Music Journal*, vol. 26, no. 3, pp. 11–22, Sep. 2002. [Online]. Available: https://doi.org/10.1162/014892602320582945

[4] E. Michel and T. Boubekeur, "DAG amendment for inverse control of parametric shapes," *ACM Transactions on Graphics*, vol. 40, no. 4, pp. 173:1–173:14, Jul. 2021. [Online]. Available: https://doi.org/10.1145/3450626.3459823

[5] G. Hadjeres and L. Crestel, "The Piano Inpainting Application," Jul. 2021, arXiv:2107.05944 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2107.05944

[6] Q. Kong, B. Li, J. Chen, and Y. Wang, "GiantMIDI-Piano: A Large-Scale MIDI Dataset for Classical Piano Music," *Transactions of the International Society for Music Information Retrieval*, vol. 5, no. 1, pp. 87–98, May 2022, number: 1 Publisher: Ubiquity Press. [Online]. Available: http://transactions.ismir.net/articles/10.5334/tismir.80/

[7] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen, "Hierarchical Text-Conditional Image Generation with CLIP Latents," Apr. 2022, arXiv:2204.06125 [cs]. [Online]. Available: http://arxiv.org/abs/2204.06125

[8] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes, T. Salimans, J. Ho, D. J. Fleet, and M. Norouzi, "Photorealistic Text-to-Image Diffusion Models with Deep Language Understanding," May 2022, arXiv:2205.11487 [cs]. [Online]. Available: http://arxiv.org/abs/2205.11487

[9] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-Resolution Image Synthesis with Latent Diffusion Models," Apr. 2022, arXiv:2112.10752 [cs]. [Online]. Available: http://arxiv.org/abs/2112.10752

[10] T. Moore, "A fork of the Stable Diffusion Cog model that outputs tileable images for use in 3D applications such as Monaverse," Sep. 2022, original-date: 2022-09-06T22:48:42Z. [Online]. Available: https://github.com/TomMoore515/material_stable_diffusion

[11] AUTOMATIC1111, "Stable Diffusion web UI," Sep. 2022, original-date: 2022-08-22T14:05:26Z. [Online]. Available: https://github.com/AUTOMATIC1111/stable-diffusion-webui

[12] C. Cantrell, "The Stable Diffusion Photoshop Plugin," 2022. [Online]. Available: https://christiancantrell.com/#ai-ml

[13] A. Caillon and P. Esling, "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis," Dec. 2021, arXiv:2111.05011 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2111.05011

[14] Bytedance, "Mawf," 2022, closed-source. [Online]. Available: https://mawf.io/

[15] J. H. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable Digital Signal Processing," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. [Online]. Available: https://openreview.net/forum?id=B1x1ma4tDr

[16] "DDSP VST," Sep. 2022, published under an Apache-2.0 license. [Online]. Available: https://github.com/magenta/ddsp-vst

[17] Qosmo, "Neutone by Qosmo," 2022. [Online]. Available: https://neutone.space/

[18] ——, "Neutone SDK," Sep. 2022, published under an LGPL-2.1 license. [Online]. Available: https://github.com/QosmoInc/neutone_sdk

[19] A. Roberts, J. Engel, Y. Mann, J. Gillick, C. Kayacik, S. Nørly, M. Dinculescu, C. Radebaugh, C. Hawthorne, and D. Eck, "Magenta Studio: Augmenting Creativity with Deep Learning in Ableton Live," in *Proceedings of the International Workshop on Musical Metacreation (MUME)*, 2019. [Online]. Available: http://musicalmetacreation.org/buddydrive/file/mume_2019_paper_2/

[3] https://mwm.io/

[4] https://ai.sony/

[20] R. Bougueng Tchemeube, J. J. Ens, and P. Pasquier, "Calliope: A Co-creative Interface for Multi-Track Music Generation," in *Creativity and Cognition*, ser. C&C '22. New York, NY, USA: Association for Computing Machinery, Jun. 2022, pp. 608–611. [Online]. Available: https://doi.org/10.1145/3527927.3535200

[21] C.-Z. A. Huang, C. Hawthorne, A. Roberts, M. Dinculescu, J. Wexler, L. Hong, and J. Howcroft, "Approachable music composition with machine learning at scale," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, the Netherlands, November 4-8, 2019*, A. Flexer, G. Peeters, J. Urbano, and A. Volk, Eds., 2019, pp. 793–800. [Online]. Available: http://archives.ismir.net/ismir2019/paper/000097.pdf

[22] G. Hadjeres, F. Pachet, and F. Nielsen, "DeepBach: a Steerable Model for Bach Chorales Generation," in *Proc. of the 34th International Conference on Machine Learning (ICML)*, 2017, pp. 1362–1371, _eprint: 1612.01010.

[23] R. Louie, A. Coenen, C. Z. Huang, M. Terry, and C. J. Cai, "Novice-AI Music Co-Creation via AI-Steering Tools for Deep Generative Models," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20. New York, NY, USA: Association for Computing Machinery, Apr. 2020, pp. 1–13. [Online]. Available: https://doi.org/10.1145/3313831.3376739

[24] C. Donahue, I. Simon, and S. Dieleman, "Piano Genie," in *Proceedings of the 24th International Conference on Intelligent User Interfaces*, Mar. 2019, pp. 160–164, arXiv:1810.05246 [cs, eess, stat]. [Online]. Available: http://arxiv.org/abs/1810.05246

[25] Y. Mann, "A.I. Duet," 2020. [Online]. Available: https://experiments.withgoogle.com/ai-duet

[26] ——, "googlecreativelab/aiexperiments-ai-duet," Sep. 2022, published under an Apache-2.0 license. [Online]. Available: https://github.com/googlecreativelab/aiexperiments-ai-duet

[27] T. Bazin and G. Hadjeres, "NONOTO: A Model-agnostic Web Interface for Interactive Music Composition by Inpainting," in *Proceedings of the Tenth International Conference on Computational Creativity, ICCC 2019, Charlotte, North Carolina, USA, June 17-21, 2019*, K. Grace, M. Cook, D. Ventura, and M. L. Maher, Eds. Association for Computational Creativity (ACC), 2019, pp. 89–91. [Online]. Available: http://computationalcreativity.net/iccc2019/papers/iccc19-paper-27.pdf

[28] T. Bazin, G. Hadjeres, P. Esling, and M. Malt, "Spectrogram Inpainting for Interactive Generation of Instrument Sounds," in *Proceedings of the 2020 Joint Conference on AI Music Creativity*. Stockholm, Norway: KTH Royal Institute of Technology, Jul. 2020. [Online]. Available: https://hal.archives-ouvertes.fr/hal-03207968

[29] T. Bazin and G. Hadjeres, "SonyCSLParis/music-inpainting.ts," Oct. 2022. [Online]. Available: https://github.com/SonyCSLParis/music-inpainting-ts

[30] "Tone.js," Sep. 2022, published under an MIT license. [Online]. Available: https://github.com/Tonejs/Tone.js

[31] Y. Mann, "Interactive Music with Tone.js," in *First Web Audio Conference*, Paris, France, Jan. 2015, p. 5. [Online]. Available: https://wac.ircam.fr/pdf/wac15_submission_40.pdf

[32] J. P. Côté, "User-Friendly MIDI in the Web Browser," in *International Conference on New Interfaces for Musical Expression*, Jun. 2022. [Online]. Available: https://nime.pubpub.org/pub/user-friendly-midi-in-the-web-browser/release/1

[33] ——, "WebMidi.js," 2022, published under an Apache-2.0 license. [Online]. Available: https://github.com/djipco/webmidi

[34] O. Cífka, "html-midi-player," Jul. 2022, published under a BSD-2-Clause License. [Online]. Available: https://github.com/cifkao/html-midi-player/