

FXplorer: A Map-Based Interface for Exploratory Audio Effect Design

Annie Chu
anniechu@u.northwestern.edu
Northwestern University
Chicago, IL, USA

Jason Brent Smith
jason.smith1@northwestern.edu
Northwestern University
Chicago, IL, USA

Bryan Pardo
pardo@northwestern.edu
Northwestern University
Chicago, IL, USA

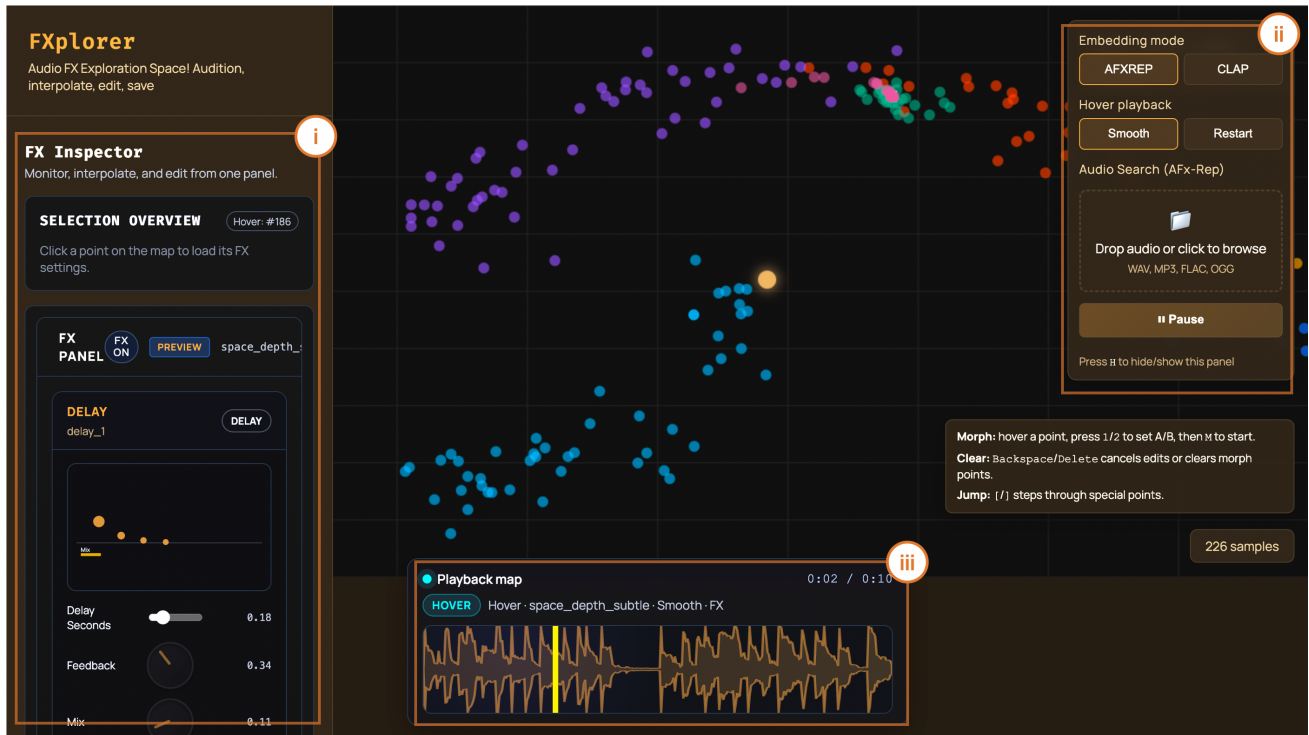


Figure 1: *FXplorer*'s Explore Mode: Users primarily navigate via mouse (hover, click, pan, zoom) and keyboard shortcuts. Inspector panel (i) for parameter editing, assigning interpolation endpoints, and viewing exact parameters for the currently selected effect variant. Floating control panel (ii) for embedding mode switching (CLAP/AFX-Rep), hover playback settings, and semantic search inputs. Waveform monitor (iii) displays the currently playing audio with a moving playhead.

Abstract

Audio effects (FX) shape sound in contemporary music practice. However, most interfaces present them as discrete modules and parameters that favor targeted adjustment over exploratory listening. This separation can make it difficult to build intuition about the broader space of possible transformations or to move fluidly between searching and refinement. We present *FXplorer*, an interface that organizes audio effects within a perceptually informed 2D space, allowing sound transformations to be browsed as a continuous landscape rather than as isolated presets. By combining established spatial interaction approaches and interpretable DAW-style controls with recent embedding-based machine learning methods for similarity and semantic search, the system brings exploration and parameter refinement into a single workspace. *FXplorer* supports composition, production, or

performance by allowing users to edit and interpolate between effect presets interactively.

Keywords

audio effects, interface, exploration, audio embeddings

1 Introduction

Audio effects (FX) shape the timbral identity of recorded and performed sound, from subtle enhancement to radical transformation. The interfaces through which musicians access these possibilities often constrain creative exploration. For example, a producer seeking “something warm and spacious” must choose among discrete effect types (e.g., Reverb, EQ) and modify low-level parameters describing the underlying signal processing of a single effect (e.g., changing the “Q” of a filter) without clear guidance about what sonic possibilities exist and how they relate to one another. This reflects a broader gap between the expressive potential of audio FX and the limited exploratory affordances of current tools [22].



This work is licensed under a Creative Commons Attribution 4.0 International License.

NIME '26, June 23–26, 2026, London, UK

© 2026 Copyright held by the owner/author(s).

Creative sound design involves both divergent exploration (discovering what is possible) and convergent refinement (achieving a specific sonic goal) [26]. While musicians frequently move fluidly between these modes, most digital audio workstations (DAWs) privilege refinement [18]: users select effects from taxonomic menus and manipulate isolated parameters. The prior knowledge assumed by this workflow (e.g., which parameters matter) fails both newcomers building sonic intuitions and experienced practitioners in exploratory contexts.

Creators often begin with a vague intent [5, 12] –perhaps a verbal description like “something muddy” or another audio sample that is “muddy” in their opinion– and need a way to reach the space of “muddy” possibilities without necessarily knowing which FX chains or parameters to choose. After identifying a few promising transformations, they may explore the space between them, listening to how changes in mixing and effect parameters shape the sound.

The key challenge in this workflow lies in the multidimensionality and nonlinearity of FX parameter spaces. Even simple effects expose vast combinatorial possibilities, and FX chains introduce order-dependent, perceptually-uneven interactions [23]. Auditioning alternatives typically requires repeated cycles of parameter adjustment and rollback, discouraging rapid, low-commitment exploration [5]. Preset libraries provide entry points but obscure the continuous sonic space between configurations [22]. Recent deep learning approaches enable semantic access to effects through text queries [3, 6, 13] or audio examples [24], but typically return isolated results rather than ongoing exploration.

Map-based interfaces have long been used in NIMEs to organize discrete sound corpora in perceptually meaningful 2D layouts [4, 8–10, 20]. Deep learning has further enabled synthesis-focused instruments such as RAVE [2], NSynth [7], and related latent-space systems supporting continuous, perceptually organized exploration within similarly navigable 2D control spaces [1, 7, 11, 17, 21, 25, 28], including investigations of their gestural affordances [14, 29]. These systems support low-commitment auditioning and reveal structure beyond categorical menus, yet typically operate on fixed-sound corpora or on abstract generative parameters.

We present *FXplorer*, a prototype interface for musical expression (NIME) for **exploratory audio FX design**. *FXplorer* explores the question, “what if audio effects could be navigated spatially, continuously and by ear, rather than through lists of modules and parameters?” *FXplorer* organizes hundreds of effect variants of a source sound into a perceptually organized two-dimensional space, where nearby points have timbral similarity. Musicians and performers can search this space using text or audio examples and explore how their changes reposition them within the sonic landscape. *FXplorer* is designed to **support both divergent exploration and convergent refinement**, making FX parameter space navigable without sacrificing interpretability.

2 Design Requirements for Audio Effect Exploration

Based on the challenges outlined above—navigating high-dimensional effect spaces, supporting rapid auditioning, connecting parameter changes to perceived sound, and enabling search through listening-based goals—we derive four design requirements for exploratory audio FX interfaces.

DR1: Perceptually-organized exploration space. Effect variants should be organized into a 2D layout where proximity

Keyboard Shortcut	Function
1	Set Interpolation Endpoint A
2	Set Interpolation Endpoint B
[and]	Cycle through Saved Points
B	Toggle FX Bypass
M	Enter Interpolation Mode
← / →	Adjust Interpolation Blend
H	Toggle Floating Controls
Ctrl/Cmd & +/-	Zoom In/Out

Table 1: *FXplorer*’s Keyboard Shortcuts

reflects timbral similarity of transformations, revealing relationships beyond categorical FX hierarchies.

DR2: Semantic entry points into navigable space. Text and audio queries should situate users within a continuous map rather than isolated retrieval results, enabling “ballpark” entry and subsequent exploration of adjacent sonic territories.

DR3: Low-commitment auditioning of transformations. The interface should enable rapid auditioning of dozens of configurations without explicit selection or commitment, facilitating divergent exploration while preserving parameter interpretability for convergent refinement. When a user finds two points they like, they should be able to “slingshot,” or navigate between them quickly and continuously.

DR4: Connect real-time parameter edits to perceptual context. The system should scale parameters perceptually (logarithmic for frequency, exponential for time constants, linear for gain/mix) and visualize how changes map to perceptual space.

FXplorer integrates these requirements through a hybrid architecture: pre-rendered variants organized via audio embeddings (DR1), dual-embedding search supporting text & audio queries (DR2), rapid in-situ auditioning (DR3), and module-aware editing with synchronized spatial/parametric/auditory feedback (DR4).

3 System Design & Implementation

FXplorer is a locally-hosted web interface prototype for composition or real-time musicking, designed with novices and low setup in mind. It displays audio effect variants on a fluid, interactive 2D canvas (see Fig. 1).

FXplorer follows a hybrid architecture: computationally expensive operations such as variant generation are performed offline, while audio playback and interaction are handled in the browser for low-latency exploration. It’s contained within a Svelte¹ web application that loads precomputed coordinates and audio at startup. Tone.js² enables real-time FX processing in the browser, reconstructing audio on-the-fly by applying parameter configurations to the dry source. A Flask API handles on-demand operations in the backend: projecting new points, embedding text queries, matching audio examples.

3.1 Sample Generation

FXplorer’s backend operates in three stages: generation, embedding, and reduction.

Generation. The system first randomly samples parameter configurations for user-selected effect modules and renders the variants using Pedalboard³ (fast, bit-accurate DSP).

¹<https://svelte.dev/>

²<http://tonejs.github.io/>

³<https://github.com/spotify/pedalboard>

Embedding. Each variant is encoded with two complementary embedding models capturing different notions of similarity. AFx-Rep [24], an audio-only model, embeds and organizes sounds by perceived timbral characteristics, while CLAP (audio-text) [27] aligns sounds with language descriptors (e.g., “warm,” “metallic”). Each embedding kind is projected separately into its own 2D layout. Users can toggle between these views, switching the notion of similarity while exploring the same set of variants. This enables neighborhood browsing and text- or audio-driven search within a consistent interface.

Reduction. These high-dimensional embeddings (512) are projected to 2D via PCA [19] (or UMAP [16]) for visualization. Unlike methods like t-SNE [15], PCA/UMAP supports out-of-sample projection, enabling runtime mapping of new points (edits, searches) without refitting the entire space.

4 Interaction Design

FXplorer supports a creative workflow (Fig. 2) where users generate a perceptually organized variant map and perform all browsing, search, interpolation, and refinement directly within this shared space before saving.

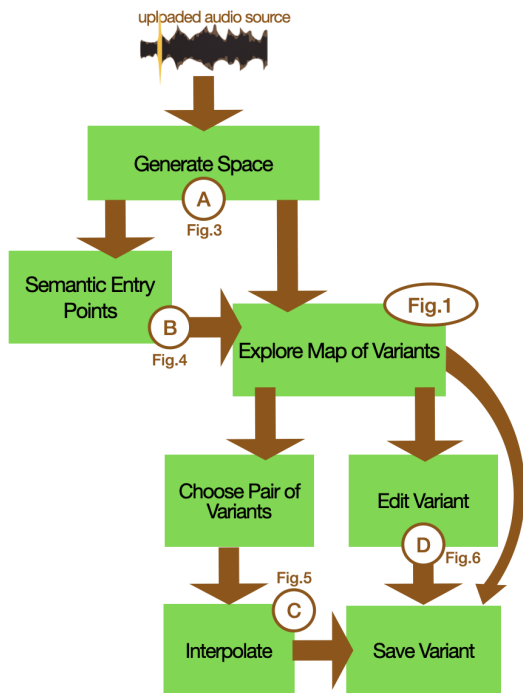


Figure 2: Overview of interaction workflows in *FXplorer*. From an input audio source, the system constructs a semantic variant space that users navigate to explore alternatives, select and interpolate between variants, iteratively edit outcomes, and save results, supporting both exploratory (divergent) and refinement (convergent) processes.

4.1 Preparing the Exploration Space: Generating Variants (DR1)

Users may begin without a parameter plan, asking what transformations are possible for a source sound and generating a broad set of variations (Fig. 3).

Input Dur.	Generate	Embed	Reduce	Total	Device
2s	3.74	26.18	6.97	36.95	CPU
2s	3.79	20.10	6.98	30.89	GPU
4s	4.33	28.49	7.03	39.90	CPU
4s	4.33	20.49	7.06	31.92	GPU
10s	6.10	35.54	7.25	49.00	CPU
10s	6.11	21.72	7.26	35.18	GPU

Table 2: Mean end-to-end wall-clock time (seconds) for the user-facing pipeline with a new dry input upload, target_samples=100. Device affects only the *Embed* stage; *Embed* sums AFx-Rep and CLAP embedding time, *Reduce* sums the two PCA reductions, and each row reports the mean over 3 runs.

Users begin by uploading dry audio, choose effect modules, and *FXplorer* generates the variants that populate the map. *FXplorer* was designed with short samples in mind (2-4s); Table 2 provides a benchmark on end-to-end wall-clock time. Available FX modules include 6-band parametric equalization (EQ), Compression, Reverb, Delay, Distortion, Chorus, and Phasing. Users can choose individual effects or combine up to three modules to explore interactions between effects.

FXplorer samples parameter configurations for the selected modules and displays the resulting variants (≈ 100 , adjustable) as points on a 2D canvas. Each point represents a distinct effect configuration colored by FX chain type. The layout is organized by timbral similarity, so variants that sound alike cluster together regardless of the specific effects used. For example, a “warm” sound produced by low-pass filtering appears similar to a sound shaped with reverb, despite different underlying processes.

Once generated, this space serves as the canvas for subsequent browsing, searching, and editing.

4.2 Entering the Space: Semantic Entry Points (DR2)

Rather than scrolling through lists of isolated presets, users can enter the exploration space through semantic entry points. Starting from a rough sonic intention like “warm and fuzzy,” users can provide text queries or audio examples to search and highlight the most similar variants *in-situ* within the existing map, situating results within their local neighborhood for immediate exploration (Fig. 4).

This search is computed by two complementary embeddings and cosine similarity: CLAP [27] aligns sounds with descriptive language for text queries; AFx-Rep [24] measures timbral similarity for audio-to-audio matching. Users toggle between these semantic and timbral similarity modes while exploring the same variant set.

4.3 Exploring Variants by Rapid Audition & Interpolation (DR3)

Users explore the space by browsing and auditioning nearby variants. Moving the cursor over any point immediately plays the corresponding audio, with FX applied in real-time to the dry source. This enables rapid sampling of dozens of FX configurations. Clicking a point selects it for focused audition, begins audio looping, and exposes its parameters for editing in the Inspector.

When users identify two variants of interest, they can set them as endpoints via keyboard commands and continuously move between them to explore intermediate effects. Rather than

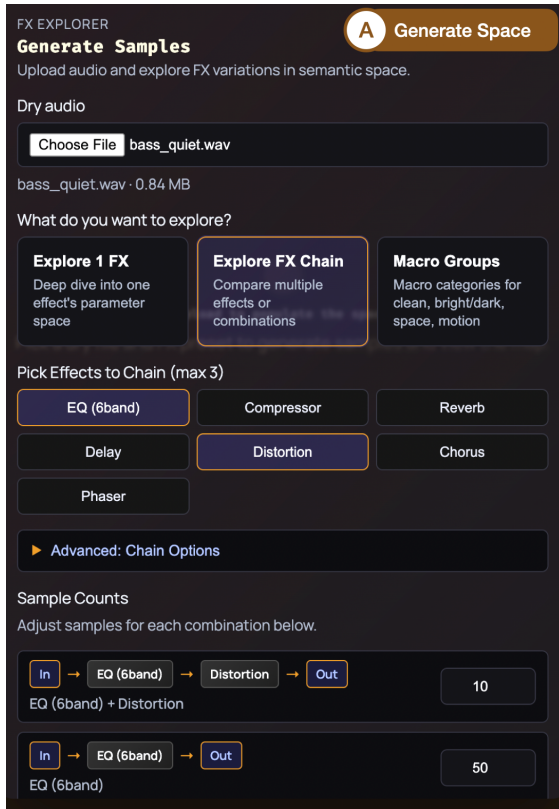


Figure 3: Sample generation in *FXplorer*. Users upload dry audio, select an exploration mode (single, effect chain, or macro groups), choose effects to include and specify n samples to generate for each configuration.

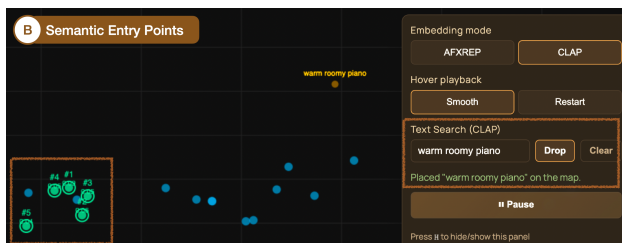


Figure 4: Semantic Entry Point Example (text): Results for “warm roomy piano”. Embeds the actual query and highlights the top-5 matches.

crossfading audio, *FXplorer* interpolates in parameter space: it blends the endpoint effect settings and applies the resulting configuration to the dry source in real time. When endpoints share the same effect chain, their parameters blend directly; different chains require manual endpoint reselection. The view zooms to focus on the endpoints, and the arrow keys slide across the interpolation line between the two endpoints (see Fig. 5). As users navigate, audio updates immediately; parameter values and their FX visuals (e.g., EQ curve, delay lines) adjust in the inspector panel; a ghost marker moves along the trajectory in 2D space, providing synchronized feedback. To support perceptually smooth transitions, parameters are scaled in effect-specific ways (e.g., logarithmic for frequency, exponential for time constants, linear for wet/dry).

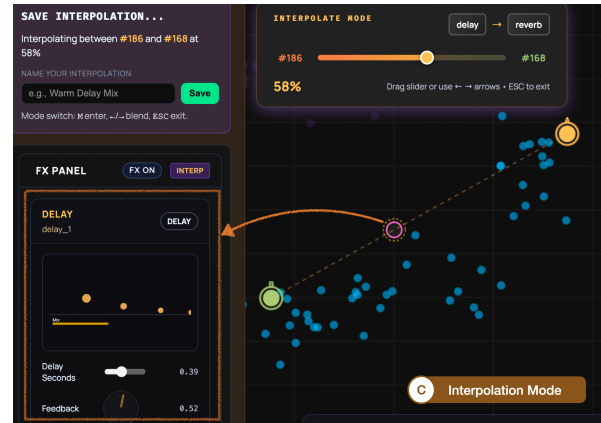


Figure 5: Interpolation Mode interface. Two samples from the exploration space are selected as points “A” and “B”. The user can quickly interpolate continuously between endpoint parameters closest to point A (0%) or B (100%).

4.4 Refining Ideas: Real-time Parameter Editing Feedback (DR4)

After finding a promising region or variant, users can refine a sound directly by adjusting its effect parameters in the Inspector. Edits are applied to the dry source in real time, allowing immediate auditory feedback while the sound loops. To maintain context within the exploration space, *FXplorer* projects edited configurations back into the 2D layout, enabled by re-embedding and PCA’s out-of-sample projection. A “ghost” point marker shows where modified settings would lie on the map, linking parameter changes to their spatial position. Large movements indicate substantial timbral change, while small shifts reflect subtle adjustments (Fig 6). This feedback aims to help users develop intuition about how parameter edits relate to sonic transformations.



Figure 6: Variant editing selected point in the map (right) is opened in the FX panel (left), where users directly manipulate effect parameters (e.g., delay time, feedback, mix) to iteratively refine the sound and save an updated variant.

5 Conclusions

We present *FXplorer*, a prototype interface that reframes audio effect design from a sequence of discrete parameter adjustments to navigation through a continuous landscape of transformations. By unifying rapid auditioning, interpolation, semantic search, and direct parameter editing within a single audio-visual space, the system enables fluid movement between exploration and refinement.

This work extends the NIME tradition of map-based navigational interfaces to audio FX. Unlike systems that operate over fixed corpora or abstract parameters, *FXplorer* maintains explicit, DAW-compatible controls, aiming to balance exploratory interaction with practical production needs. Future work could evaluate how these spatial and semantic affordances influence sound design strategies, exploring larger or adaptive variant sets.

6 Ethical Standards

The system uses machine learning only for similarity estimation and retrieval through discriminative embeddings; it does not generate audio or train models during use. All sounds derive from user-provided material and transparent effect parameters. Processing occurs locally; embedding and projection introduce some computational overhead, though the footprint is substantially lower than generative approaches. We acknowledge the potential for some artists to feel embedding-based similarity suggestions stifle their creativity. Our goal is to assist artists in this exploratory process; we believe designing *FXplorer* with creators in mind overcomes potential adverse impact on human creativity.

Acknowledgments

This work was supported by NSF Award Number 2300633.

References

- [1] 2021. XO - XLN Audio.
- [2] Antoine Caillon and Philippe Esling. 2021. RAVE: A variational autoencoder for fast and high-quality neural audio synthesis. *arXiv preprint arXiv:2111.05011* (2021).
- [3] Annie Chu, Patrick O'Reilly, Julia Barnett, and Bryan Pardo. 2025. Text2fx: Harnessing clap embeddings for text-guided audio effects. In *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.
- [4] Francesco Ardan Dal Ri, Domenico Stefani, Luca Turchet, and Nicola Conci. 2025. MorphDrive: Latent Conditioning for Cross-Circuit Effect Modeling and a Parametric Audio Dataset of Analog Overdrive Pedals. In *Proceedings of the 28-th Int. Conf. on Digital Audio Effects (DAFx25)* (Ancona, Italy), L. Gabrielli and S. Cecchi (Eds.).
- [5] Stefano Delle Monache, Nicolas Misdariis, and Elif Özcan. 2022. Semantic models of sound-driven design: Designing with listening in mind. *Design Studies* 83 (2022), 101134.
- [6] Seunghoon Doh, Junghyun Koo, Marco A Martínez-Ramírez, Wei-Hsiang Liao, Juhan Nam, and Yuki Mitsufuji. 2025. Can Large Language Models Predict Audio Effects Parameters from Natural Language? *arXiv preprint arXiv:2505.20770* (2025).
- [7] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. 2017. Neural audio synthesis of musical notes with wavenet autoencoders. In *International conference on machine learning*. PMLR, 1068–1077.
- [8] Frederic Font and Giuseppe Bandiera. 2017. Freesound explorer: make music while discovering freesound!. In *Proceedings of the 3rd Web Audio Conference*.
- [9] Ohad Fried, Zeyu Jin, Reid Oda, and Adam Finkelstein. 2014. AudioQuilt: 2D Arrangements of Audio Samples using Metric Learning and Kernelized Sorting. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Goldsmiths, University of London, London, United Kingdom, 281–286. <https://doi.org/10.5281/zenodo.1178766>
- [10] Leandro Garber, Tomás Ciccola, and Juan Cruz Amusatogui. 2020. AudioStellar, an open source corpus-based musical instrument for latent sound structure discovery and sonic experimentation. In *Proceedings of ICMC*.
- [11] L. H. Hantrakul. 2017. lamtharnhantrakul/klustr. [Online] <https://github.com/lamtharnhantrakul/klustr>.
- [12] Aaron Hertzmann. 2022. Toward Modeling Creative Processes for Algorithmic Painting. In *ICCC*. <https://api.semanticscholar.org/CorpusId:248506151>
- [13] Hojoon Ki, Jongsuk Kim, Minchan Kwon, and Junmo Kim. 2026. FxSearcher: gradient-free text-driven audio transformation. In *ICASSP 2026-2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 15462–15466.
- [14] Alexander Lunt and Sebastian Trump. 2023. Latent Space Explorer. *AIMC 2023* (aug 29 2023). <https://aimc2023.pubpub.org/pub/zgc5j7ha>.
- [15] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9, Nov (2008), 2579–2605.
- [16] Leland McInnes, John Healy, and James Melville. 2018. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).
- [17] Jason Naradowsky. 2021. Amp-space: A large-scale dataset for fine-grained timbre transformation. In *2021 24th International Conference on Digital Audio Effects (DAFx)*. IEEE, 57–64.
- [18] Bryan Pardo, Mark Cartwright, Prem Seetharaman, and Bongjun Kim. 2019. Learning to build natural audio production interfaces. In *Arts*, Vol. 8. MDPI, 110.
- [19] Karl Pearson. 1901. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science* 2, 11 (1901), 559–572.
- [20] Gerard Roma, Owen Green, and Pierre Alexandre Tremblay. 2019. Adaptive Mapping of Sound Collections for Data-driven Musical Interfaces. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Marcelo Queiroz and Anna Xambó Sedó (Eds.). UFRGS, Porto Alegre, Brazil, 313–318. <https://doi.org/10.5281/zenodo.3672976>
- [21] Diemo Schwarz, Grégory Beller, Bruno Verbrugge, and Sam Britton. 2006. Real-time corpus-based concatenative synthesis with catart. In *9th International Conference on Digital Audio Effects (DAFx)*. 279–282.
- [22] Prem Seetharaman and Bryan Pardo. 2016. Audelize: Crowdsourced audio production tools. *Journal of the Audio Engineering Society* 64, 9 (2016), 683–695.
- [23] Spyridon Stasis, Nicholas Jillings, Sean Enderby, and Ryan Stables. 2017. Audio processing chain recommendation. In *Proceedings of the 20th International Conference on Digital Audio Effects, (Edinburgh, UK)*.
- [24] Christian J Steinmetz, Shubhr Singh, Marco Comunità, Ilias Ibnayahya, Shanxin Yuan, Emmanouil Benetos, and Joshua D Reiss. 2024. St-ito: Controlling audio effects for style transfer with inference-time optimization. *arXiv preprint arXiv:2410.21233* (2024).
- [25] Manny Tan and Kyle McDonald. 2017. Infinite Drum Machine. [Online] <https://experiments.withgoogle.com/ai/drum-machine/>.
- [26] Robert Tubb and Simon Dixon. 2014. The Divergent Interface: Supporting Creative Exploration of Parameter Spaces. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Goldsmiths, University of London, London, United Kingdom, 227–232. <https://doi.org/10.5281/zenodo.1178967>
- [27] Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. 2023. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 1–5.
- [28] Ruihan Yang, Tianyao Chen, Yiyi Zhang, and Gus Xia. 2019. Inspecting and interacting with meaningful music representations using VAE. *arXiv preprint arXiv:1904.08842* (2019).
- [29] Shuoyang Jasper Zheng, Anna Xambó Sedó, and Nick Bryan-Kinns. 2025. Exploring gestural affordances in audio latent space navigation. *Frontiers in Computer Science* 7 (2025), 1575202.