Organised Sound (2026), 1–10
doi:10.1017/S1355771826101666

CAMBRIDGE UNIVERSITY PRESS

Article

The Tensed Computer Improviser and the workmanship of risk: Small-dataset machine learning and real-time interactive musical improvisation

Taylor Brook
Music, The University of British Columbia Faculty of Arts, Canada

Abstract

Generative artificial intelligence (AI) has entered music overwhelmingly by way of large industrial models trained on vast corpora, in which the outcome is fixed in advance by an apparatus that the musician neither controls nor understands. This article argues that the essential machine-learning operations of dataset assembly, feature extraction and model fitting can instead be utilised transparently as part of an artist's creative practice. Looking to David Pye's concepts of *workmanship of certainty and workmanship of risk*, this article examines the design of the Tensed Computer Improviser (TCI), an open-source machine improviser built in Max with the Fluid Corpus Manipulation (FluCoMa) toolkit, which trains models from a small audio corpus of the user's own choosing, with no pretrained component and no external training stage. Situating the TCI within a genealogy of machine improvisation using small data, the article analyses its design through the vocabulary of craft where the corpus curation and training are integrated into the creative act and the characteristic errors of small data are welcomed as tool-marks.

Keywords: AI music; computer improvisation; computer music aesthetics; craft; creative coding

(Received 6 July 2026 UTC; revised 29 August 2026 UTC; accepted 28 September 2026 UTC)

1. Introduction: craft and automation

In *The Nature and Art of Workmanship*, David Pye distinguishes the *workmanship of risk*, where 'the quality of the result is not predetermined, but depends on the judgment, dexterity and care which the maker exercises as he works', from the *workmanship of certainty*, in which the outcome is fixed in advance by the design of the productive apparatus (Pye 1968: 4). Machine learning and artificial intelligence (AI) have entered music overwhelmingly in the latter mode where generative AI systems are trained on vast corpora (Agostinelli et al. 2023; Copet et al. 2023), and the commercial services built upon this approach, such as Suno and Udio, are able to produce entire songs from text prompts. The values of scale, stylistic fidelity and commercial viability that are articulated by generative music systems reflect Pye's workmanship of certainty, where the musical result is determined by a system in which each output is sampled within a controlled band of randomness from statistical regularities fixed during a training process. The workings of large-scale industrial musical generative AI remain substantially opaque, both by technical constitution and by corporate design (Burrell 2016; Morreale 2021), and the copyright questions identified by Sturm et al. (2019) have crystallised into major litigation, with the principal record labels suing the generative-music services Suno and Udio over their training corpora (UMG Recordings, Inc. v. Suno, Inc. 2024; UMG Recordings, Inc. v. Uncharted Labs, Inc. 2024). This article keeps three properties of such systems distinct: *scale* names the size of a system's data and computation; *industrial* names a mode of production in which training is infrastructure built by an organisation apart from use; and *generative* names a model's function of synthesising new output from a learned representation. The prevailing systems are all the three at once, but the properties are separable; the TCI is itself generative, while being neither large-scale nor industrial.

The underlying operations of machine learning at the base of generative AI do not demand a large-scale mode of implementation; a dataset may be assembled, features extracted and models fitted at any scale, including the scale of a single musician's practice or as part of the performance itself. Similarly, the way in which a corpus of audio is used to train a generative model can be articulated with transparency through open source software development alongside clear and precise language. At a manageable scale and with a transparent ethos, training ceases to be industrial infrastructure and becomes a *workmanship of risk*. The Tensed Computer Improviser (TCI) was developed to practise machine learning in this way, in the comparatively unoccupied niche of software through which an improvising musician can train, inspect and perform with their own small-scale machine-learning models.

The TCI is an open-source machine improviser written in Max, built using the Fluid Corpus Manipulation (FluCoMa; Tremblay et al. 2021) toolkit that listens to a live audio input and responds from an audio corpus of the user's own choosing. The corpus is segmented, analysed and modelled within the Max patch; there is no external training stage and no pretrained component. From its

Email: taylor.brook@ubc.ca

Cite this article: Brook T. (2026). The Tensed Computer Improviser and the workmanship of risk: Small-dataset machine learning and real-time interactive musical improvisation. *Organised Sound*, 1–10, https://doi.org/10.1017/S1355771826101666

© The Author(s), 2026. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (https://creativecommons.org/licenses/by/4.0/), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 17 Nov 2026 at 23:36:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1355771826101666

title, Tensed refers first to the tensor, the data structure at the heart of contemporary machine learning; where its predecessor, the Scuffed Computer Improviser (Brook 2023b), assembled its generative behaviours from simple machine listening and algorithmic approaches, the TCI is built on machine-learning architecture proper where unsupervised models must be trained before the software will sound. The TCI's models are the classical, inspectable kind, including dimensionality reduction, clustering and nearest-neighbour search. Using *Tensed* in the title is also a play on *scuffed*, retaining the earlier name's self-deprecating register. The Scuffed Computer Improviser was imperfect, and the tensed TCI is nervous and strung tight, serving as a deliberate counterweight to the mystifying rhetoric that surrounds AI (Elish and boyd 2018). The TCI does not represent a technical leap forward in machine learning or AI, but it serves as an example of how creative coders are using the machine-learning algorithms introduced into Max by the FluCoMa toolkit and building upon a lineage of computer improvisation; a lively area of activity without substantial scholarship describing the approach or theorising the aesthetic stance and implications of such tools.

This article extends an argument developed across two of my earlier texts: the first critiqued large-scale machine learning as a force for stylistic standardisation and economic consolidation in music and art (Brook 2023a); the second proposed small, custom-made datasets and models tailored to specific artistic intentions as an aesthetic alternative to big data (Brook 2025). The claim advanced here is that small-dataset machine learning is not a degraded miniature of the industrial kind but a distinct practice with its own values, and that those values are well articulated through the vocabulary of craft: curation of the corpus as a compositional act; training as part of the creative act rather than infrastructure; the legibility of one's tools as a form of transparency; and the characteristic errors of small-data models welcomed rather than discarded as defects.

2. A genealogy of small-data machine improvisation

Interactive machine improvisation has, from its beginnings, favoured a small-data practice: its systems derive their behaviour from the individual performer (Chadabe 1984; Rowe 1993; Pachet 2003), or learn from corpora curated by their makers (Schwarz 2006, 2007). For much of the tradition this smallness was a technical condition, a consequence of the computational limitations of real-time operation and of building systems that listen to a particular musician, rather than as an aesthetic commitment. The rise of large-scale generative models has thrown that condition into relief, and this retrospective view has revealed a difference in kind between models trained on material the maker has selected and heard versus models trained on corpora too vast for any individual to fully know. The first keeps the musician's judgement in the loop at every stage; it is, in Pye's terms, hospitable to a workmanship of risk.

George Lewis's *Voyager*, whose first version was programmed between 1986 and 1988, remains a foundational example of interactive machine improvisation (Lewis 2000: 34), preceded by earlier real-time interactive systems such as Joel Chadabe's Coordinated Electronic Music Studio and the League of Automatic Music Composers (Chadabe 1984; Rowe 1993). *Voyager* did not learn in the statistical sense, as its 'virtual improvising orchestra' was rule-based, but it exemplified the two commitments the tradition has carried since: that an improvising machine performs with musicians interactively, and that its programme is a cultural text, representing 'the particular ideas of their creators' rather than any 'objective' or 'universal' musicality (Lewis 2000: 33). Statistical models of a performer's own playing also entered this tradition in the mid-1980s, when Zicarelli's Jam Factory (issued alongside the related programme M, by Chadabe's company Intelligent Music) built Markov transition tables from a musician's MIDI input and improvised upon them in real time (Zicarelli 1987). Pachet's Continuator brought this approach to maturity, constructing variable-order Markov models of a musician's input in real time and continuing their phrases in a stylistically consistent manner (Pachet 2003). In the factor-oracle lineage at Institute for Research and Coordination in Acoustics/Music, the MIDI-based OMax (Assayag et al. 2006) was extended first to live audio through pitch-tracking into a symbolic representation, then to continuous audio descriptors by the Audio Oracle (Dubnov et al. 2007), and developed through a family of agents, the scenario-guided ImproteK (Nika et al. 2016) and the reactive SoMax2 (Borg et al. 2022) among them. What these systems train on typically amounts to minutes of music, such as a single session's playing or a handful of recordings, and frequently the present performance itself; ImproteK's memory, for instance, could be constituted 'by recording the music played by the human co-improvisers during a live performance' (Nika et al. 2016: 5).

In parallel, corpus-based concatenative synthesis established the audio corpus as a navigable space, organised by descriptor analysis and traversed by similarity (Schwarz 2006, 2007), and CatOracle joined the two strands, coupling CataRT's corpus navigation to Audio Oracle pattern learning for improvisation in which 'the user's creative aural judgements are favored over theoretical criteria' (Einbond et al. 2016: 141).

The TCI descends from both currents at once. From the interactive systems lineage it takes the premise of a machine that performs interactively alongside a musician and that embodies its maker's particular ideas rather than a neutral musicality; from concatenative synthesis it takes the audio corpus as a navigable space, traversed by descriptor similarity and sounded by replaying actual recorded segments of audio. What allows it to join the two is the arrival of machine-learning algorithms inside Max through the FluCoMa toolkit (Tremblay et al. 2021). The TCI thus brings together the improvising partner of the *Voyager* lineage and the corpus-as-instrument of concatenative synthesis: an interactive improviser whose entire musical world is a small, chosen body of recorded sound.

A second current in interactive machine improvisation makes the small-data commitment explicit: interactive machine learning as instrument building. The Wekinator demonstrated that models trained 'on the fly' from a handful of examples supplied by the musician could function as a meta-instrument (Fiebrink et al. 2009), extending a use of machine learning in instrument design that Fiebrink and Sonami (2020: 237) trace back more than 25 years. The practice that followed, most extensively documented in Fiebrink's eight-year collaboration with Laetitia Sonami on the Spring Spyre (Fiebrink and Sonami 2020), has developed a working aesthetics in which the smallness and intimacy of training data are treated as creative affordances rather than statistical deficiencies (Fiebrink 2019; Vigliensoni and Fiebrink 2023). Young and Blackwell's account of live algorithms supplies the tradition's shared horizon: such research 'is not concerned with systems that imitate human behavior; genuinely novel outcomes are sought, a product of renewed forms of human-computer interaction' (Young and Blackwell 2016: 508). In this way, partnership and interaction become a site for expanding the boundaries of a musician's aesthetic sensibility and expression (Brook 2025).

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 17 Nov 2026 at 23:36:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1355771826101666

The values stated by generative music research at scale are of a different kind, and the systems that state them deserve the same precision as the improvising lineage. The Music Transformer is presented as an advance in 'generating music with long-term structure', evaluated by perplexity and listener preference against a corpus of competition piano performances (Huang et al. 2019a); the Bach Doodle, which harmonised user melodies in the style of Bach for a mass audience, announces its achievement in its title as 'approachable music composition with machine learning at scale' and reckons that achievement in scale: more than 55 million harmonisation requests over three days, of which 21.6 million compositions, voluntarily contributed by their users, were released as a public dataset (Huang et al. 2019b). The audio-domain systems that followed extend the trajectory from style modelling to end-to-end song production. A foundational model is trained once, by a research organisation, on a corpus assembled at a scale only such an organisation can hold, Jukebox on 1.2 million songs (Dhariwal et al. 2020), MusicLM on 280,000 hours of audio (Agostinelli et al. 2023), MusicGen on 20,000 hours of licensed music (Copet et al. 2023), at a computational cost that places training beyond any individual practitioner. What a user then meets is the finished model's conditioning interface: a text prompt, but increasingly also musical material, whether MusicGen's melodic conditioning or MusicLM's transformation of a hummed or whistled tune (Agostinelli et al. 2023; Copet et al. 2023). The commercial song services, like Suno, built on the same foundation carry this interface to a mass public. Training determines the model's sonic world and is complete before any musician arrives. These are substantial achievements on their own terms, but fidelity to a style distribution, coherence at scale and deployment to millions measure a workmanship of certainty, and they are values orthogonal to those that experimental traditions of improvised music have most prized. Practices as heterogeneous as those of the Association for the Advancement of Creative Musicians (AACM) and free jazz (Lewis 1996) and Pauline Oliveros's deep listening (Oliveros 2005) are affiliated less by a shared history than by shared values: not the faithful reproduction of a style but its creative disruption through the cultivation of difference in an improvisation that is an encounter with an unpredictable partner that Born, Lewis and Straw describe as 'not only of social commentary but, potentially, of social experimentation' (Born et al. 2017: 9–10).

3. TCI technical overview

The TCI's design follows the workflow its user encounters: a corpus of audio is imported, segmented and analysed; a bank of models is trained from those analyses; and in performance the software assembles its response by sampling short segments of the corpus with concatenative synthesis, steered by real-time analysis of the live input (Figure 1).

The interface presents the preparation of these models as a numbered sequence; its panels are labelled STEP 1 through STEP 4, and the improviser does not sound until the steps have been completed to train the models (Figure 2). Until a corpus is analysed, the descriptor maps are blank cells and its waveform display is empty. Importing and analysing an audio corpus is the act that allows the software to sound and function.

3.1. Building and segmenting the corpus

A TCI corpus may be a single audio file, a folder of files concatenated into one buffer, or audio recorded live from the input. The documentation advises keeping the source audio short to keep analysis and real-time matching within a workable latency budget.

The corpus is divided in two complementary ways. First, onset detection slices the audio buffer at points of attack or abrupt spectral change so that each slice is a sounding event of the kind a listener might parse as a single gesture. In parallel, the corpus is cut into grains of fixed duration, which capture continuous texture rather than discrete gesture. The two segmentations provide a gestural and a textural reading of the same material respectively. The threshold that governs onset detection is left in the user's hands as a musical and interpretive decision, and the manual embedded in the software states that 'the more active the onset detection is, the more saturated the resulting sound of the improviser will be'. Once detection has run, the interface displays the corpus waveform overlaid with its slice boundaries and reports the number of onsets found (Figure 3), making the consequences of the threshold immediately legible to the user.

3.2. Analysis and the model bank

For the TCI training, every slice and grain of audio is analysed along five families of audio descriptors as implemented by the FluCoMa library: pitch as frequency and confidence; loudness across mel-frequency bands; spectral shape as a collection of spectral descriptors; chroma, the distribution of energy across the 12 pitch classes; and mel-frequency cepstral coefficients. Applied across the segmentations and grains, these five descriptors yield ten parallel analyses of the corpus.

From these measurements the TCI builds, for each analysis, a small bank of machine-learning models. The descriptor values are first standardised so that no single measure dominates by accident of its units. They are then projected from many dimensions down to two by a dimensionality-reduction algorithm (Uniform Manifold Approximation and Projection (UMAP)), which arranges the segments on a plane so that sounds measured as similar fall close together. A grid algorithm then redistributes the projected points onto a regular lattice of evenly spaced cells, trading the projection's faithful clustering for a surface that can be navigated evenly. The two representations serve complementary ends; the ungridded projection preserves the corpus's own density and gives a faithful picture of how the corpus is organised, whereas the gridded version trades that fidelity in favour of variation so that the dense regions do not overpower the sparse. These maps are displayed in the interface, one for each analysis (Figure 3), and the user may also move a cursor across any of them and hear the region beneath it to gain an understanding of the models.

Lastly, a clustering algorithm (k-means) sorts the segments into a chosen number of groups by similarity, labelling each corpus segment by cluster, and a nearest-neighbour index (a k-d tree) makes the maps searchable in real time so that any incoming sound can be compared with the corpus segments for similarity or difference. Once these steps are complete the TCI is fully trained and ready to perform.

3.3. Listening, behaving, sounding

Once trained, the TCI listens to the live input with the same descriptors it used on the corpus, tracking the pitch, loudness, spectral profile and attacks of the audio input in real time, following the range of dynamics and estimating beat and metre. What it hears becomes a stream of queries: a live sound is matched, through the k-d trees, to its nearest neighbours in the corpus, so

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 17 Nov 2026 at 23:36:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1355771826101666

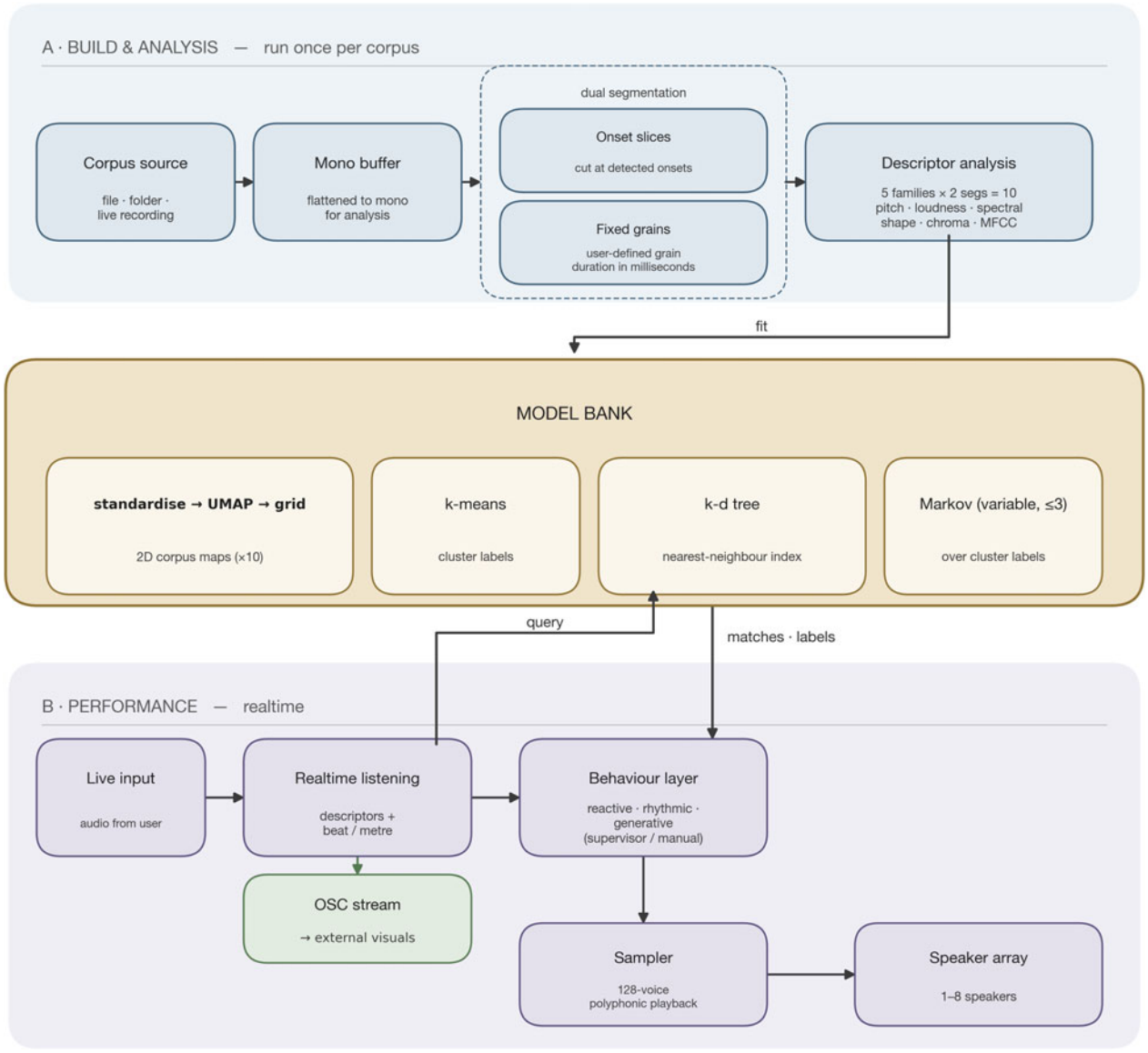


Figure 1. Signal-flow diagram of the TCI. Build and analysis section: the corpus is reduced to a mono buffer, segmented in parallel into onset slices and grains of fixed duration, and measured along five descriptor families, yielding ten analyses that fit the model bank. Model bank section: for each analysis, a standardised–UMAP–grid pipeline produces 2D corpus maps, k-means yields cluster labels, a k-d tree provides a nearest-neighbour index and a variable-order Markov chain (maximum order three) is built upon the cluster labels and inter-onset durations. Performance section: live input is tracked with the five descriptors and queried against the k-d tree, whose matches and labels drive a layer of behaviours (see Figure 4); the segments they select are sounded by a 128-voice polyphonic sampler across a variable loudspeaker array, when the analysis is streamed over OSC to external processes such as live visuals as desired.

that the machine answers from its own material in terms set by the user.

How the TCI selects what samples or grains of audio it plays back is governed by a bank of 16 behaviours, grouped in three families. Many are offered as slices-or-grains or long-or-short pairs, and the reactive and generative behaviours can each be run on any of the five descriptor analyses, so that a single behaviour may sound in as many as five variants. *Reactive* behaviours track the live input, doubling its attacks with matched material and optionally sustaining them, echoing it granularly, following, inverting or detuning its amplitude envelope, harmonising to its chroma or deliberately contrasting its spectrum. *Rhythmic* behaviours impose or follow a pulse, locking corpus playback to the detected tempo and metre, matching pulse, or generating Euclidean patterns, ratchets, and accelerating 'bouncing-ball' and decelerating 'slowing-down' figures (built from the gen~ abstractions of Wakefield and Taylor 2022). *Generative* behaviours draw on the corpus's own statistics or on autonomous processes. The first is a variable-order Markov chain (ml.star; Smith and Garnett 2012), of maximum order three, trained on the sequence of cluster labels, plays out new labels in the order that this particular material tends to follow. The second generative behaviour is a Lorenz chaotic attractor that steers parameters along continuously unfolding trajectories. The third imposes a pulsation of grains or segments within a randomly selected label in from the model. This three-family grouping is analytic, organised by how each behaviour responds to the live input; in the patch itself the behaviours are laid out differently, first by segmentation (those acting on onset slices and those acting on grains) and then in rhythmic, textural, doubling and generative

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 17 Nov 2026 at 23:36:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1355771826101666

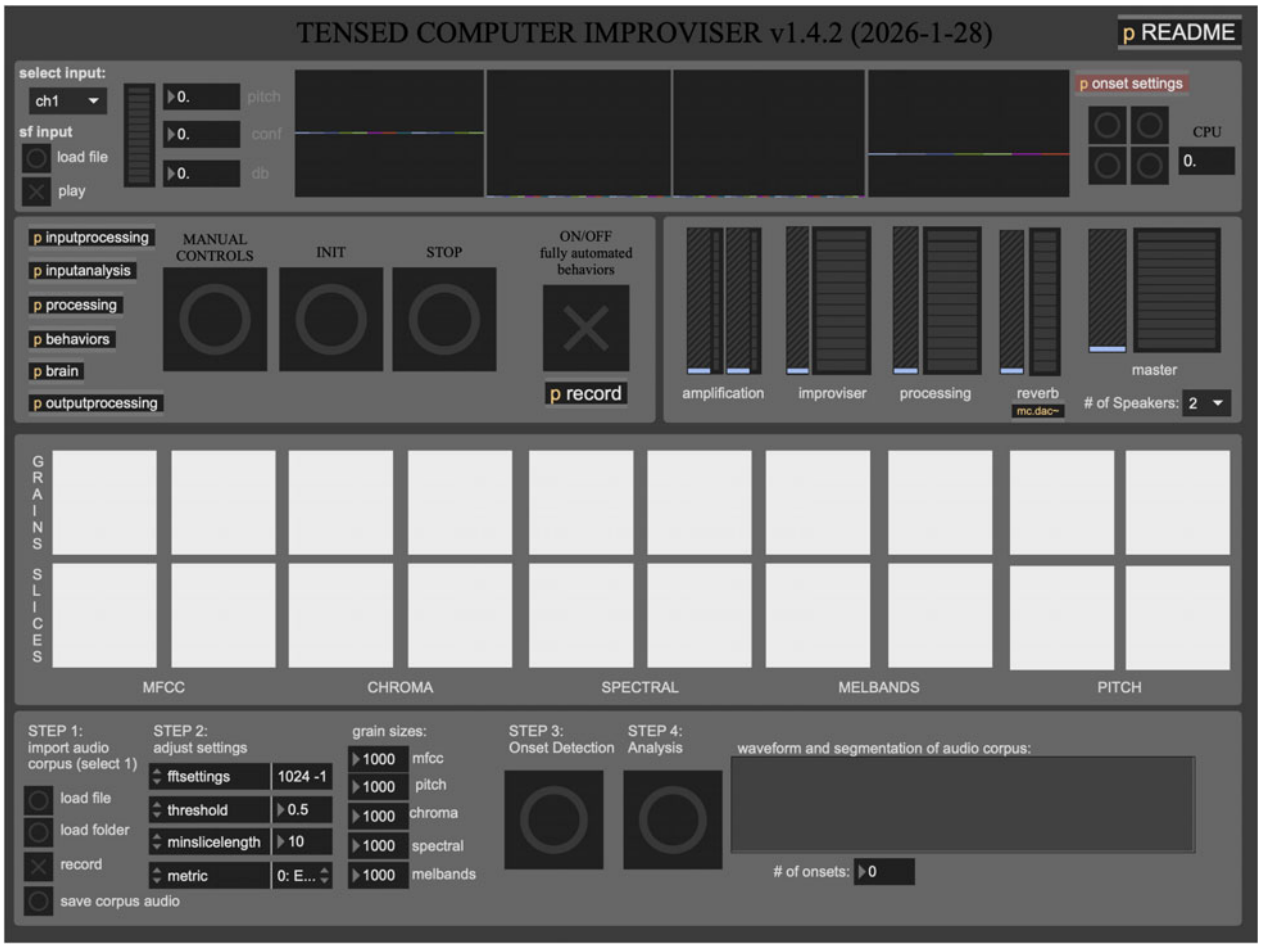


Figure 2. The TCI's main interface as it appears on launch, before a corpus has been analysed. The preparation of the models is laid out as a numbered sequence in the lower panel: STEP 1 (import audio corpus), STEP 2 (adjust settings), STEP 3 (onset detection) and STEP 4 (analysis). Until those steps are run, the descriptor maps remain as empty cells, the waveform display is empty and the TCI will produce no sound.

columns, alongside a separate set of one-shot gestures (the bouncing-ball and slowing-down figures among them) and the global effects (Figure 4). Table 1 regroups them by how they respond to the live input.

In its fully automated mode the TCI supervises the behaviours itself, holding one or a few active at a time, switching among them by weighted chance, sensing whether a pulse is present, and forcing a change after five seconds of the performer's silence. The behaviours are saturated with controlled randomness, so that no two passages through the same corpus are alike; the result is an improviser consistent in character but unpredictable in detail. A manual mode sets the supervisor aside and lets a performer audition or play individual behaviours directly.

As the TCI performs, it streams its analysis in the MAX console and via Open Sound Control (OSC) to local network ports so that the same listening that drives its playing can be made to drive external processes such as live visuals. The instrument that results is wholly contained in the patch and wholly the product of the material fed to it.

4. Small datasets as a workmanship of risk

Pye's concepts provide an understanding about where in a process the quality of the outcome comes to be decided, and he observed that most real work mixes the two modes of risk and certainty. In industrial generative music the deciding act of the assembly and modelling of the training corpus happens once, before the system is ever played and on infrastructure the musician has no role in. A vast body of audio is gathered and a model trained on it at great computational expense by the developing organisation. The musician/user meets only the finished model, and the meeting is not confined to the text prompt; as Section 2 noted, contemporary systems accept melodic or audio conditioning that steers generation (Agostinelli et al. 2023; Copet et al. 2023). But conditioning operates within a sonic world already fixed. The practitioner can neither intervene in the foundational training nor determine the corpus from which the model was built, and it is there, in training, that the aesthetic boundaries of the outcome are decided: the locus of Pye's 'risk' has been engineered out of reach. The algorithms the TCI uses are deterministic procedures: given the same corpus and the same settings (with random seeds held fixed), UMAP, k-means and the rest return the same models every time. In Pye's terms the apparatus fixes the result, which is the definition of the workmanship of certainty. However, the TCI puts the certainty of its algorithms back into the service of a workmanship of risk by returning control over what material the models are built from, how it is segmented, when and how often they are fitted.

The TCI's models are not pretrained, but are built entirely from whatever audio is provided by the user. Choosing and recording the corpus are part of the creative act, reflecting the understanding

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 17 Nov 2026 at 23:36:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1355771826101666

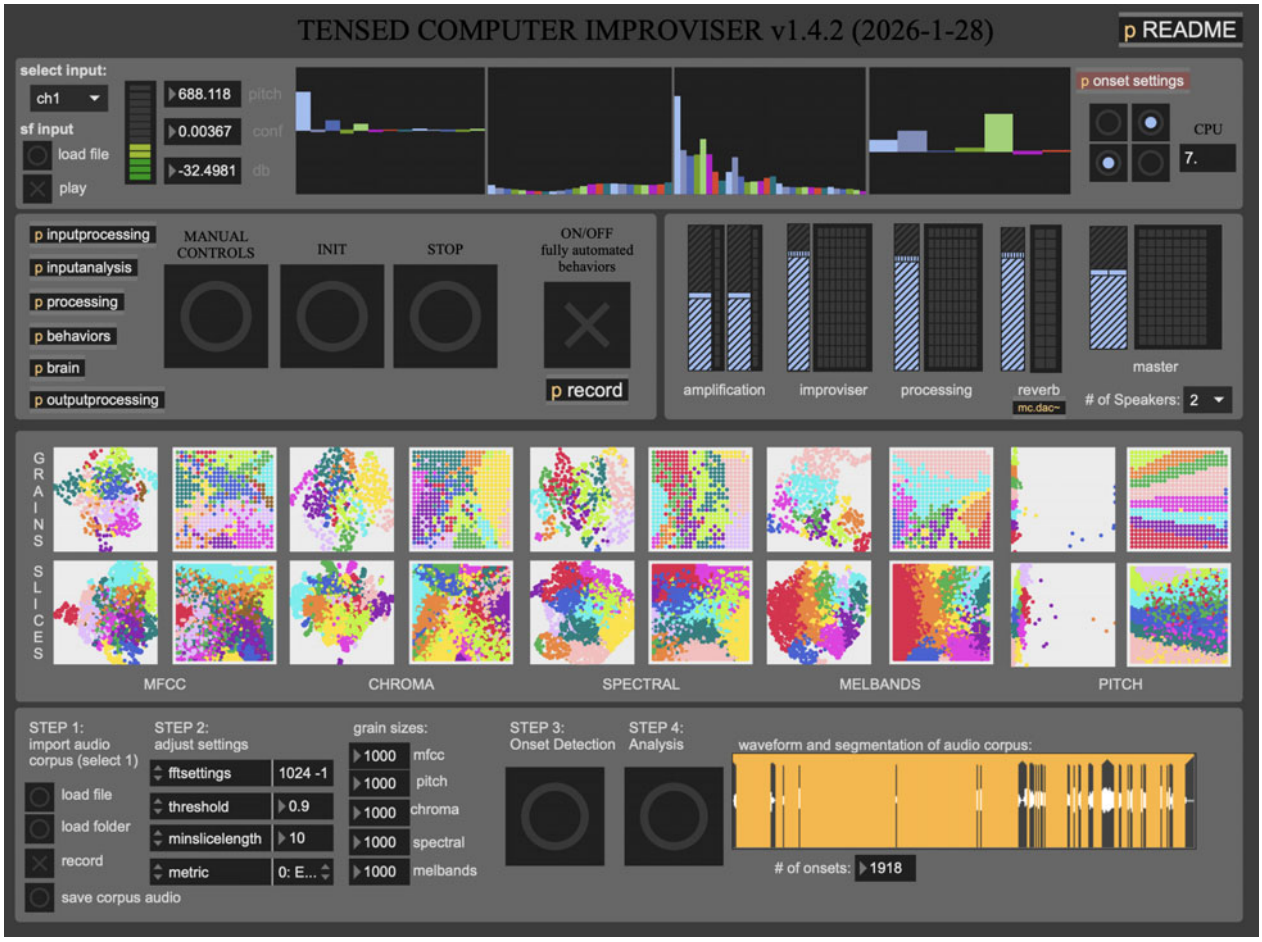


Figure 3. The TCI interface after a corpus has been imported and analysed. Each analysis is shown as a 2D corpus map aligning with the five descriptor families (MFCC, chroma, spectral, melbands and pitch) across the two segmentations (grains, top row; slices, bottom row). The points of the corpus maps are spread by the high minimum-distance UMAP setting and normalised to fit the plotters; the colours of the points represent the results of the unsupervised k-means clustering; the performer may click on a point on any map and hears the slice or grain of audio it represents. The corpus waveform and its slice boundaries appear at lower right, with the onset count reported beneath (here 1,918). Corpus shown: loadbang's recording of the author's *Tarantism* (loadbang 2021).

that deciding which sounds are admitted to the corpus, and whose, is an aesthetic and social decision rather than a technical one (Born et al. 2017). It is here, too, that the smallness of the corpus contributes to the aesthetic. The standardising tendency I have elsewhere identified as central to large-scale machine learning, the regression towards the statistical centre of a vast dataset (Brook 2023a; Doshi and Hauser 2024), finds insufficient data to act on when that dataset is ten minutes of a single musician's playing. The distinction between training and conditioning is relevant; ten minutes of audio could not train a large generative model; and it could serve a pretrained model as conditioning or adaptation material, but material used that way steers a representation already derived from a much larger dataset, and the result remains answerable to that dataset's statistical centre. In the TCI the same ten minutes is not a query put to a larger learned world; they constitute the model's entire sonic world, and the boundary of the corpus is the boundary of what the TCI can sound. This bespoke fit of a corpus tailored to a single musician and invites a recognised objection; McCormack and colleagues observe, of real-time creative-AI systems generally, that they 'tend towards being bespoke to their designer(s), making general or sustained creative development difficult' (McCormack et al. 2020: 45). Read through Pye, the close fit of instrument to maker is not a deficiency to be engineered out but the condition of any craft tool, and the source of whatever identity the instrument comes to have.

Training a model from a small corpus of audio in the TCI fits in seconds to a few minutes, and inside the same patch that will perform, the division that structures industrial machine learning does not arise. Training is not infrastructure laid down before the work begins but folded into the work itself. Setting the onset threshold, choosing how finely the corpus is clustered, hearing the resulting map and re-analysing with a different setting are interpretive decisions that are reversible, repeatable and audible in their effects. This is the condition Fiebrink and Sonami describe from eight years of building and refining the Spring Spyre with machine learning: models shaped iteratively by a musician's ear rather than optimised against a benchmark (Fiebrink and Sonami 2020). It is what Pye names a workmanship of risk: a result whose quality depends on the maker's judgement and care as the work proceeds. The effort this asks of the player is not friction to be designed away: as Michalakos, after Joel Ryan, observes, the effort an interface demands is closely bound to the expression it affords (Michalakos 2021: 80). Effort alone does not make a craft; however, it earns its place where it exposes consequential decisions that become legible through practice. The labour of slicing, clustering and auditioning is of this kind, where

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 17 Nov 2026 at 23:36:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1355771826101666

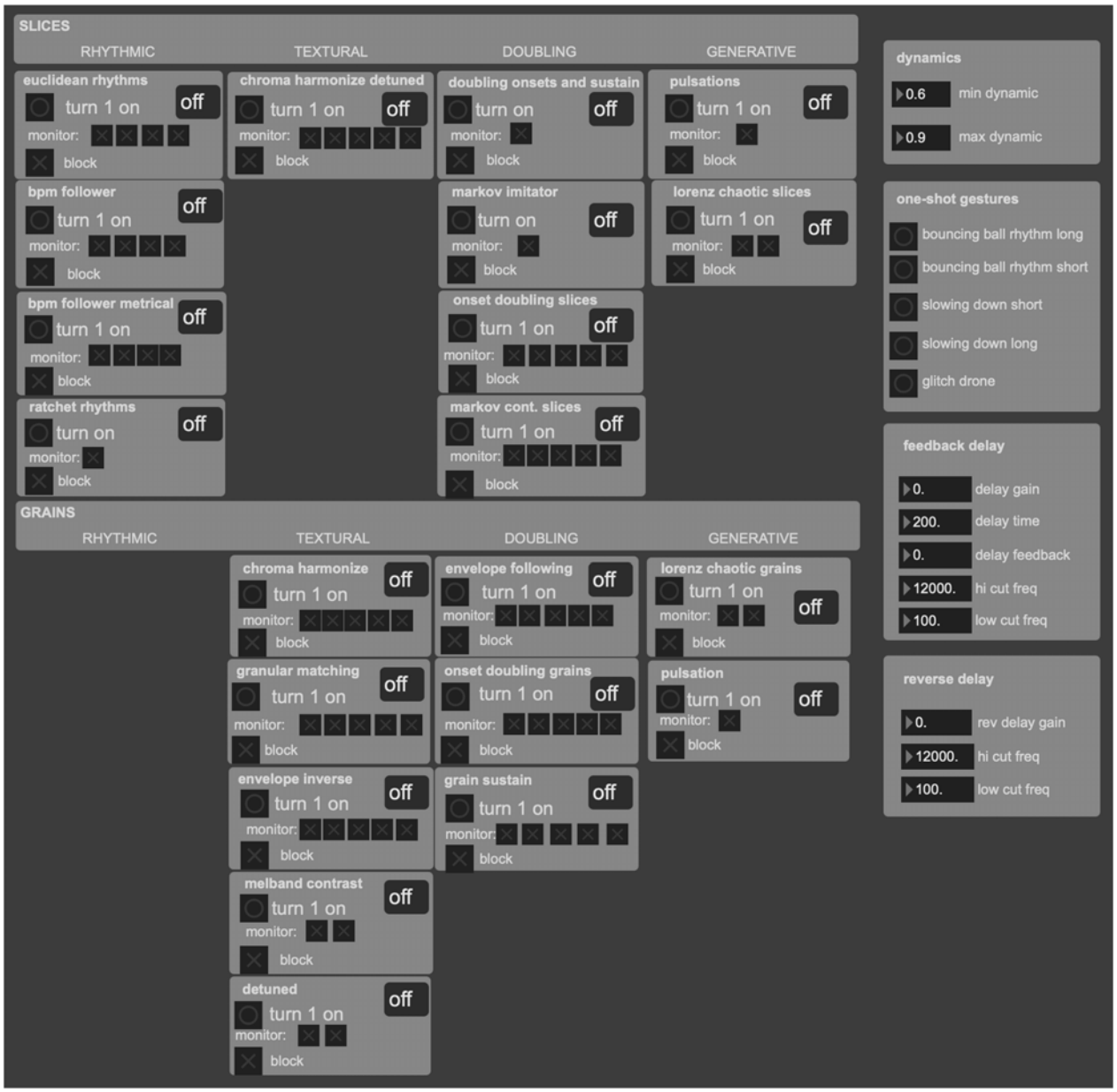


Figure 4. The behaviour panel. The TCI's behaviours are grouped first by segmentation (SLICES, top; GRAINS, below) and then into RHYTHMIC, TEXTURAL, DOUBLING and GENERATIVE columns, with a separate set of one-shot gestures (the bouncing-ball and slowing-down figures and a glitch drone) and the global effect parameters (dynamics, feedback delay, reverse delay) on the right. Behaviours change automatically during operation in reaction to the audio input. Additionally, behaviours can be turned on, monitored or blocked manually by the user. The article slightly regroups these modules analytically by how they respond to the live input as reactive/rhythmic/generative (Table 1).

every adjustment is audible in the maps and in the playing, and it is where the playing of this instrument begins.

A craftsperson knows their tools and materials intimately, and that knowledge is itself part of the skill (Sennett 2008). Understood this way, craft is less a category of objects than, in Adamson's terms, a way of doing things, a process and a disposition rather than a class of artefact (Adamson 2019: 3–4). Radford gives that disposition its electroacoustic setting, describing the studio as an ecology of implements through which a practitioner's skill and knowledge are mediated (Radford 2015). The TCI reflects these values of craftsmanship by making the corpus maps the principal display of the user interface. The user sees how the machine has organised the corpus, can move through any map and audition its regions, and can therefore form an understanding of what the TCI is capable of. Where the prevailing rhetoric of AI trades on opacity, the 'magic' that Elish and boyd identify as obscuring how these systems work (Elish and boyd 2018), the TCI instead makes its hearing clear and comprehensible.

Three further features follow from the corpus being audio rather than symbol. First, recorded sound carries the trace of the body that made it: in Iyer's account, to hear music is to perceive the action of another agent (Iyer 2016: 87). What reaches the listener from the TCI can be heard as someone's playing, not as a rendering of playing in general. Second is the technical condition on which the first depends; the TCI's concatenative playback cuts, indexes and reassembles the corpus itself so that its output is always fragments of the source recordings rather than an approximation generated from a learned stylistic distribution. The embodied trace

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 17 Nov 2026 at 23:36:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1355771826101666

Table 1. Behaviour modules, grouped analytically by how they respond to the live input (Figure 4)

Group	Behaviour	Description
Reactive *(track the live input)*	Onset doubling (slices/grains)	Doubles the audio input attacks at detected onsets with matched corpus material
	Doubling onsets and sustain	Doubles audio input attacks at detected onsets, then sustains the matched material
	Granular matching	Continuous matching of the audio input from nearest-neighbour grains
	Envelope follower (following/ inverse/detuned)	Follows, inverts or detunes against the amplitude of the audio input
	Chroma harmonise (plain/ detuned)	Continuous transposition of the chroma of the audio input
	Melband contrast	Selects material spectrally contrasting the audio input
Rhythmic *(impose or follow a pulse)*	BPM follower (by label/by match)	Locks corpus playback to the detected tempo, sampling slices by matching the audio input or drawing randomly from a labeled cluster
	BPM follower, metrical	Locks playback to the detected tempo and metre
	Euclidean rhythms	Samples slices in Euclidean patterns, drawing upon the labelled clusters of the corpus
	Ratchet rhythms	Subdivided, retriggering ratchet figures, drawing upon the labelled clusters of the corpus
	Pulsation (slices/ grains)	Samples slices from the audio corpus with a steady pulse
	Bouncing ball (short/long)	Accelerating 'bouncing-ball' rhythmic figures
	Slowing down (long/short)	Decelerating figures
Generative *(corpus statistics/ autonomous processes)*	Markov continuous slices	Plays new cluster-label sequences from the variable-order Markov chain
	Markov imitator	Markov-driven imitation of the corpus's temporal syntax
	Lorenz chaotic (slices/grains)	Rhythm of slice playback derived from a Lorenz-attractor algorithm

Variants are shown in parentheses. Reactive and generative behaviours run on any of the five descriptor maps (pitch, melbands, spectral shape, chroma and MFCC) – the 'up to five variants' of each. The rhythmic generators (bouncing-ball, ratchet, Euclidean, pulsation) are built from the gen~ abstractions of Wakefield and Taylor (2022); the Markov behaviours use the ml.star MAX package (Smith and Garnett 2012). Grain sustain and glitch drone also appear in the behaviours window as toggles, but these are one-shot effects, not behaviours, and are excluded from the table.

survives in the output because the TCI plays fragments of the corpus source audio, rather than newly generated. Third, a small corpus yields a sparse and uneven map; its clusters do not divide cleanly, and its nearest-neighbour matches are often approximate or relatively distant. These are the errors of data-poverty that can be creatively stimulating, and they are not engineered away in the TCI. These errors are where the instrument may exceed its maker's intention, the surprise that McCormack and colleagues identify as a central value of generative systems, those able to 'surprise and delight their authors' (McCormack et al. 2019: 39). The Scuffed Computer Improviser pursued such imprecision deliberately (Brook 2023b); in the TCI, it arises of its own accord, from the smallness of the data, and is welcomed in the same spirit, as the tool-marks of a workmanship of risk. Unpredictability is not in itself a workmanship of risk; Pye's term names the dependence of the result on judgement, dexterity and care, not the mere presence of uncertainty. As a performer develops the skill to anticipate the TCI behaviours, to provoke or avoid them, and to turn them to musical account, the workmanship of risk materialises.

Keeping machine learning small, then, moves the decisions that determine the music into the hands and ears of a musician working with chosen material in real time. The small-data approach also facilitates legibility regarding how the software is functioning and is a practice that can be shown and taught.

5. Making in public

Craft is never fully reducible to written instruction but passes from maker to maker by showing, watching and imitation. Indeed, Sennett places apprenticeship at the centre of craftsmanship (Sennett 2008). The TCI enters that tradition in two ways. First, it is open-source; the Max patch and its abstractions are published in full so that any user may open the instrument, read its construction and alter it. Second, the TCI was built publicly through a series of video streams documenting its making from scratch (Brook 2023c). The video series exists as a record of the development of the TCI, in which the design decisions set out in Section 3 are shown being taken and worked through. What my previous arguments urged regarding the demystification of machine learning (Brook 2023a, 2025), the build series attempts to enact, making the instrument's construction explained and reproducible.

The TCI is built on FluCoMa, a toolkit conceived to put corpus analysis and machine learning into musicians' hands as a shared and documented practice rather than a proprietary service (Tremblay et al. 2021), and carried by an explicit pedagogy of teaching that accessibility (Moore et al. 2025), reflecting a culture of transmission and open-source access. This culture of transmission is seen within the patch itself as the TCI credits its sources where they are borrowed: the beat-and metre-detection derives from work contributed by Philippe Salembier via their Youtube comment; the chaotic and rhythmic generators are adapted from the gen~ abstractions of Wakefield and Taylor (2022); the learning objects are those of FluCoMa and of Smith's ml.star library. These acknowledgements embody a model of authorship sharply opposed to the one prevailing in industrial generative AI, where the labour a system depends on is characteristically effaced; recordings absorbed into a training set return only as statistical tendency, a condition that has resulted in both ethical and legal difficulties (Morreale 2021). Two distinct failures are at work here, and they should be separated. The first is a lack of acknowledgement, and it is not universal; MusicGen was trained on licensed catalogues (Copet et al. 2023), and Stable Audio Open is trained entirely on Creative Commons audio with its sources documented (Evans et al. 2024). The second is a lack of output-level traceability, and it survives even such disclosure; a licensed and documented corpus still returns as statistical tendency, and neither listener, developer nor contributing musician can trace a given output back to the recordings that shaped it. Nor does the effacement stop at

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 17 Nov 2026 at 23:36:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1355771826101666

musicians; the much-publicised sale of *Edmond de Belamy* as the work of an algorithm, when the network that produced it was in fact another developer's, showed the same opacity operating on upstream technical labour: who built a system that can disappear as readily as whose recordings trained it (McCormack et al. 2019: 35–7). The TCI's chain of authorship is legible at both levels and at a finer grain: the corpus a musician has chosen and recorded, the thresholds they have set, the behaviours they have selected, and the borrowed components named in the patch are all acknowledged, open to inspection and audible in the output they shape.

For a demonstration of the TCI, I trained a corpus from Architek Percussion's recording of my percussion quartet, *Incantation* (Architek Percussion 2018), whose bowed metal bars, brake drums and prepared hi-hats become the vocabulary from which the TCI improvises, its responses audibly made of that percussion writing even as they recombine it into figures the recording never held (Video example 1). The corpus in the video example is a single 8-minute recording of a percussion quartet being excited by a live performance using a mallet on a metal pipe. The mismatch of the live performance with corpus means that the TCI's nearest-neighbour matches are frequently inexact, and one hears the consequence directly; a bright metallic attack in the live input is answered not by an identical or nearly identical sound but by its nearest available neighbour of a brake-drum stroke or a bowed bar of roughly similar brightness. With this interaction, the response shadows the input at a timbral distance, suggesting perceptual connections and variations rather than replica. Furthermore, the clusters drawn from so little material do not divide cleanly, and the variable-order Markov chain that plays out their labels strings the percussion into successions that are similar to the recording but never exactly the same, making audible a syntax that has been developed in the model. Where the corpus thins, the maps grow sparse and a handful of segments recur. A performer comes to recognise and, with familiarity, to solicit or sidestep these recurrences in the moment. These are the errors of data-poverty heard as tool-marks; the bowed bars, brake drums and prepared hi-hats remain recognisably themselves, their recorded bodies intact, even as the instrument recombines them into gestures that exceed what was ever played into it. A performer may accumulate, over time, a library of such corpora, each a particular sound-world to which the TCI can be returned.

6. Conclusion

The TCI learns from a small audio corpus with legible algorithms on a single user's computer. In the operation of the TCI, the corpus is chosen and recorded rather than scraped; the models are fitted as a form of rehearsal rather than installed as infrastructure; the instrument's organisation is shown to the performer rather than hidden; its borrowings are communal and credited rather than effaced; and its sound is the slowly accumulated identity of one musician's practice rather than the statistical average of a genre. None of this requires the renunciation of machine learning, only that machine learning be employed in a way where the musician and developer can remain answerable for it. I have argued that this smallness is not a diminished version of what large generative models do, but an alternate practice that prioritises the workmanship of risk where the result stays in the control of the maker as the work proceeds.

An instrument built this way forgoes nearly everything the dominant systems of machine learning and AI are praised for. It does not generalise: a corpus is one player's sound-world, and what is learned from it transfers to no other. It does not scale, and it will not produce a finished, style-faithful artefact from a prompt or accompany a live performer in a genre-faithful way. These are real limitations, and from the standpoint of a field that measures progress by generality and scale they may look like a refusal of progress. They are better understood as the ordinary condition of craft, which values the particular over the general.

Supplementary material. The supplementary material for this article can be found at https://doi.org/10.1017/S1355771826101666.

References

Adamson, G. 2019. *Thinking Through Craft*. London: Bloomsbury.

Agostinelli, A., Denk, T. I., Borsos, Z., Engel, J., Verzetti, M., Caillon, A., *et al.* 2023. MusicLM: Generating Music from Text. *arXiv:2301.11325*. https://doi.org/10.48550/arXiv.2301.11325.

Architek Percussion. 2018. *The Privacy of Domestic Life*. Toronto: Centrediscs.

Assayag, G., Bloch, G., Chemillier, M., Cont, A. and Dubnov, S. 2006. OMax Brothers: A Dynamic Topology of Agents for Improvization Learning. Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia. New York: ACM, 125–32. https://doi.org/10.1145/1178723.1178742.

Borg, J., Assayag, G. and Malt, M. 2022. Somax 2: A Reactive Multi-Agent Environment for Co-Improvisation. Proceedings of the 19th Sound and Music Computing Conference. Saint-Étienne, 673–4. https://doi.org/10.5281/zenodo.6576141.

Born, G., Lewis, E. and Straw, W. (eds.) 2017. *Improvisation and Social Aesthetics*. Durham, NC: Duke University Press. https://doi.org/10.1215/9780822374015.

Brook, T. 2023a. Music, Art, Machine Learning, and Standardization. *Leonardo* **56**(1): 81–6. https://doi.org/10.1162/leon_a_02135.

Brook, T. 2023b. The Scuffed Computer Improviser: The Aesthetics of Imprecision in Artificial Intelligence Musical Improvisation. *Journal SEAMUS* **29**(1–2): 20–5.

Brook, T. 2023c. Building an Improvising AI Musician in Max. YouTube video series. https://www.youtube.com/playlist?list=PLvzWMw8M7A_0RBmVjJYVp71k4NswZLPYX (accessed 20 August 2026).

Brook, T. 2025. Small Datasets for Musical Creativity Using AI. *Journal SEAMUS* **33**(1–2): 19–21.

Burrell, J. 2016. How the Machine 'Thinks': Understanding Opacity in Machine Learning Algorithms. *Big Data & Society* **3**(1): 1–12. https://doi.org/10.1177/2053951715622512.

Chadabe, J. 1984. Interactive Composing: An Overview. *Computer Music Journal* **8**(1): 22–7. https://doi.org/10.2307/3679894.

Copet, J., Kreuk, F., Gat, I., Remez, T., Kant, D., Synnaeve, G., Adi, Y. and Défossez, A. 2023. Simple and Controllable Music Generation. Advances in Neural Information Processing Systems 36. Red Hook, NY: Curran Associates, 47704–20.

Dhariwal, P., Jun, H., Payne, C., Kim, J. W., Radford, A. and Sutskever, I. 2020. Jukebox: A Generative Model for Music. *arXiv:2005.00341*. https://doi.org/10.48550/arXiv.2005.00341.

Doshi, A. R. and Hauser, O. P. 2024. Generative AI Enhances Individual Creativity but Reduces the Collective Diversity of Novel Content. *Science Advances* **10**(28): eadn5290. https://doi.org/10.1126/sciadv.adn5290.

Dubnov, S., Assayag, G. and Cont, A. 2007. Audio Oracle: A New Algorithm for Fast Learning of Audio Structures. Proceedings of the International Computer Music Conference. San Francisco: ICMA, 224–7.

Einbond, A., Schwarz, D., Borghesi, R. and Schnell, N. 2016. Introducing CatOracle: Corpus-Based Concatenative Improvisation with the Audio Oracle Algorithm. Proceedings of the International Computer Music Conference. San Francisco: ICMA, 140–6.

Elish, M. C. and boyd, d. 2018. Situating Methods in the Magic of Big Data and AI. *Communication Monographs* **85**(1): 57–80. https://doi.org/10.1080/03637751.2017.1375130.

Evans, Z., Parker, J. D., Carr, CJ, Zukowski, Z., Taylor, J. and Pons, J. 2024. Stable Audio Open. *arXiv:2407.14358*. https://doi.org/10.48550/arXiv.2407.14358.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 17 Nov 2026 at 23:36:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1355771826101666

Fiebrink, R. 2019. Machine Learning Education for Artists, Musicians, and Other Creative Practitioners. *ACM Transactions on Computing Education* **19**(4): 1–32. https://doi.org/10.1145/3294008.

Fiebrink, R. and Sonami, L. 2020. Reflections on Eight Years of Instrument Creation with Machine Learning. Proceedings of the International Conference on New Interfaces for Musical Expression. Birmingham, 237–42. https://doi.org/10.5281/zenodo.4813334.

Fiebrink, R., Trueman, D. and Cook, P. R. 2009. A Meta-Instrument for Interactive, On-the-Fly Machine Learning. Proceedings of the International Conference on New Interfaces for Musical Expression. Pittsburgh, 280–5. https://doi.org/10.5281/zenodo.1177513.

Huang, C.-Z. A., Vaswani, A., Uszkoreit, J., Simon, I., Hawthorne, C., Shazeer, N., *et al.* 2019a. Music Transformer: Generating Music with Long-Term Structure. Proceedings of the 7th International Conference on Learning Representations. arXiv:1809.04281. https://doi.org/10.48550/arXiv.1809.04281.

Huang, C.-Z. A., Hawthorne, C., Roberts, A., Dinculescu, M., Wexler, J., Hong, L. and Howcroft, J. 2019b. The Bach Doodle: Approachable Music Composition with Machine Learning at Scale. Proceedings of the 20th International Society for Music Information Retrieval Conference. Delft, 793–800. https://doi.org/10.5281/zenodo.3527930.

Iyer, V. 2016. Improvisation, Action Understanding, and Music Cognition with and without Bodies. In G. E. Lewis and B. Piekut (eds.) *The Oxford Handbook of Critical Improvisation Studies*, Vol. **1**. New York: Oxford University Press, 74–90. https://doi.org/10.1093/oxfordhb/9780195370935.013.014.

Lewis, G. E. 1996. Improvised Music after 1950: Afrological and Eurological Perspectives. *Black Music Research Journal* **16**(1): 91–122. https://doi.org/10.2307/779379.

Lewis, G. E. 2000. Too Many Notes: Computers, Complexity and Culture in Voyager. *Leonardo Music Journal* **10**: 33–9. https://doi.org/10.1162/096112100570585.

loadbang. 2021. *Plays Well with Others*. New York: New Focus Recordings.

McCormack, J., Gifford, T. and Hutchings, P. 2019. Autonomy, Authenticity, Authorship and Intention in Computer Generated Art. In *Computational Intelligence in Music, Sound, Art and Design (EvoMUSART 2019), LNCS 11453*. Cham: Springer, 35–50. https://doi.org/10.1007/978-3-030-16667-0_3.

McCormack, J., Hutchings, P., Gifford, T., Yee-King, M., Llano, M. T. and d'Inverno, M. 2020. Design Considerations for Real-Time Collaboration with Creative Artificial Intelligence. *Organised Sound* **25**(1): 41–52. https://doi.org/10.1017/S1355771819000451.

Michalakos, C. 2021. Designing Musical Games for Electroacoustic Improvisation. *Organised Sound* **26**(1): 78–88. https://doi.org/10.1017/S1355771821000078.

Moore, T., Bradbury, J., Tremblay, P. A. and Green, O. 2025. Making Machine Learning Musical: Reflections on a Year of Teaching FluCoMa. *Journal SEAMUS* **32**(1–2): 24–38.

Morreale, F. 2021. Where Does the Buck Stop? Ethical and Political Issues with AI in Music Creation. *Transactions of the International Society for Music Information Retrieval* **4**(1): 105–13. https://doi.org/10.5334/tismir.86.

Nika, J., Chemillier, M. and Assayag, G. 2016. ImproteK: Introducing Scenarios into Human-Computer Music Improvisation. *Computers in Entertainment* **14**(2): Article 4, 1–27. https://doi.org/10.1145/3022635.

Oliveros, P. 2005. *Deep Listening: A Composer's Sound Practice*. New York: iUniverse.

Pachet, F. 2003. The Continuator: Musical Interaction with Style. *Journal of New Music Research* **32**(3): 333–41. https://doi.org/10.1076/jnmr.32.3.333.16861.

Pye, D. 1968. *The Nature and Art of Workmanship*. London: Cambridge University Press.

Radford, L. 2015. Implements and Artisans: The Craft of Electroacoustic Composition. Paper presented at the Electroacoustic Music Studies Network Conference, Sheffield.

Rowe, R. 1993. *Interactive Music Systems: Machine Listening and Composing*. Cambridge, MA: MIT Press.

Schwarz, D. 2006. Concatenative Sound Synthesis: The Early Years. *Journal of New Music Research* **35**(1): 3–22. https://doi.org/10.1080/09298210600696857.

Schwarz, D. 2007. Corpus-Based Concatenative Synthesis. *IEEE Signal Processing Magazine* **24**(2): 92–104. https://doi.org/10.1109/MSP.2007.323274.

Sennett, R. 2008. *The Craftsman*. New Haven, CT: Yale University Press.

Smith, B. D. and Garnett, G. E. 2012. Unsupervised Play: Machine Learning Toolkit for Max. Proceedings of the International Conference on New Interfaces for Musical Expression. Ann Arbor, 248–51. https://doi.org/10.5281/zenodo.1178419.

Sturm, B. L. T., Iglesias, M., Ben-Tal, O., Miron, M. and Gómez, E. 2019. Artificial Intelligence and Music: Open Questions of Copyright Law and Engineering Praxis. *Arts* **8**(3): 115. https://doi.org/10.3390/arts8030115.

Tremblay, P. A., Roma, G. and Green, O. 2021. Enabling Programmatic Data Mining as Musicking: The Fluid Corpus Manipulation Toolkit. *Computer Music Journal* **45**(2): 9–23. https://doi.org/10.1162/comj_a_00600.

UMG Recordings, Inc. v. Suno, Inc. 2024. No. 1:24-cv-11611 (D. Mass., filed 24 June 2024).

UMG Recordings, Inc. v. Uncharted Labs, Inc. 2024. No. 1:24-cv-04777 (S.D.N.Y., filed 24 June 2024).

Vigliensoni, G. and Fiebrink, R. 2023. Steering Latent Audio Models through Interactive Machine Learning. Proceedings of the 14th International Conference on Computational Creativity. Waterloo, ON, 393–6. https://doi.org/10.5281/zenodo.8087978.

Wakefield, G. and Taylor, G. 2022. *Generating Sound and Organizing Time: Thinking with gen~*. San Francisco: Cycling '74.

Young, M. and Blackwell, T. 2016. Live Algorithms for Music: Can Computers Be Improvisers? In G. E. Lewis and B. Piekut (eds.) *The Oxford Handbook of Critical Improvisation Studies*, Vol. **2**. New York: Oxford University Press, 507–28. https://doi.org/10.1093/oxfordhb/9780199892921.013.002.

Zicarelli, D. 1987. M and Jam Factory. *Computer Music Journal* **11**(4): 13–29. https://doi.org/10.2307/3680237.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 17 Nov 2026 at 23:36:13, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S1355771826101666