Architecture and Affordances of PLAUD: Performative Latents and Unsupervised DDSP

Błażej Kotowski*
Music Technology Group
Universitat Pompeu Fabra
Barcelona
blazej.kotowski@upf.edu

Frederic Font
Music Technology Group
Universitat Pompeu Fabra
Barcelona
frederic.font@upf.edu

Abstract

PLAUD (Performative Latents and Unsupervised DDSP) is a neural synthesizer and Max for Live instrument for live electronic music, built on NoiseBandNet and trained on small personal sound corpora. We present its architecture, combining a variational DDSP synthesis model, latent smoothing, multi-scale spectral and adversarial losses, and an optional transformer prior, alongside a set of bending operations that intervene directly in the synthesis chain: component limiting, waveshaping, and prior feedback. The Max for Live interface exposes control generation, trajectory sampling, and modulation as primary modes of interaction. Throughout, we thread an affordance analysis arguing that the system's performative character follows from architectural decisions rather than being designed on top of them. The paper contributes both a technical account of the system and a situated affordance analysis of its role in live electronic music performance.

1 INTRODUCTION

PLAUD — standing for Performative Latents And Unsupervised DDSP — is a real-time generative audio system, which, on the backend, consists of two loosely coupled neural networks trained on shared data: the DDSP-based synthesis engine responsible for the sonic vocabulary, and the Transformer-based prior module, providing its temporal evolution. On the frontend, the interaction layer is implemented through the Max for Live instrument.

PLAUD was developed with electronic music performance in mind and tested on stage throughout the engineering process. Design decisions were taken through first-person experimentation with the system by the first author, and are motivated by an interest in material engagement with machine learning artefacts. AI music generation is now widely available in various forms, from generative platforms offering mostly prompt-based audio generation, to small, DAW-compatible instruments (Caillon and Esling, 2021; Engel et al., 2020; Mitcheltree et al., 2025b). Such systems are often developed with imitation in mind: imitating qualities of music present in data, or the timbral characteristics of a target acoustic instrument, with control schemes based on well-established semantic and compositional patterns. Creative experimentation, by contrast, points to qualities specific to ML techniques, often contingent on the use of specific architectures, which seem to hold potential for direct engagement in the context of music performance. Previous work exploring these facets of ML engineering in the music domain includes experimentation with latent spaces (Tatar et al., 2023) and our own work on network bending (Kotowski and Font, 2026), among others. In this article, we subscribe to such approaches, exploring what more direct forms of material engagement with ML might offer in terms of musical expression.

This paper details PLAUD's technical architecture and reflects on the engineering motivations behind it, threading an affordance analysis across the levels of the model. We follow Davis' account of affordances (Davis and Chouinard, 2016; Davis, 2023), which complicates the binary understanding of artefacts either possessing or not possessing specific qualities, proposing instead that "artifacts *request*, *demand*, *allow*, *encourage*, *discourage*, and *refuse*" specific interactions. Affordances so understood are situated, in line with Gibson's formulation that they "have to be measured relative to the animal" (Gibson, 1979, p. 127). Here the animal is singular and specific: throughout the paper, affordances are measured relative to the first author, an engineer and performer of experimental electronic music who built the system and is, to date, its only performer. The remarks appear as set-apart *affordance notes*, threading the analysis alongside the technical exposition.

2 RELATED WORK

This work builds on both technical research in neural audio synthesis and analytical work on ML materiality and affordances. This section reviews these areas in turn.

2.1 Neural Audio Synthesis

Differentiable Digital Signal Processing (DDSP) embedded classical signal processing components within neural network training loops, producing synthesis models whose internal structure remains interpretable and open to intervention (Engel et al., 2020). The paradigm has since expanded into a diverse landscape of instrument-oriented, sound-design, and vocoder applications, surveyed comprehensively by Hayes et al. (2024). NoiseBandNet (Barahona-Ríos and Collins, 2024) departs from DDSP's harmonic assumptions by substituting oscillators with a filterbank of narrow noise bands, making the architecture suitable for noisy, textural, and unpitched material while retaining the interpretability of the synthesis chain. In parallel, variational approaches to audio representation have shown that structured latent spaces can organise timbral variation along perceptually meaningful axes (Esling et al.,

*Website of the author: https://blazejkotowski.com

2018). On the training side, adversarial losses such as MelGAN's multi-scale discriminator (Kumar et al., 2019) have proven effective at recovering spectral detail that averaging-prone reconstruction objectives suppress.

RAVE (Caillon and Esling, 2021) represents the most widely adopted instrument-oriented neural synthesizer to date, combining a variational latent space with an optional autoregressive prior and real-time inference through the nn~ external. Its streaming architecture and small training footprint have made it a common starting point for instrument builders working with neural audio. The present work shares deployment infrastructure with RAVE but departs from it at the synthesis level, operating on a DDSP chain rather than a learned waveform decoder.

2.2 Latent Space Navigation and Active Divergence

A growing body of work investigates how latent audio spaces can be navigated and performed in practice. Tatar et al. (2023) propose traversal strategies for sound design contexts that foreground artistic agency over technical optimisation. Tahiroglu and Wyse (2024) examine GAN latent spaces as platforms for sonic creativity, arguing that their relationship to real-time performance remains problematic. Privato et al. (2024)'s Stacco investigates how different models afford distinct modes of gestural control through a dedicated physical interface, while Zheng et al. (2025) contribute empirical observations on how performers develop intuitions for abstract parameter spaces. Across these projects a common theme emerges: the structure and mode of access to the latent space determine what the resulting instrument makes possible. Much of the analysis in the present paper is motivated by this question: *how do affordances propagate from architectural decisions through interface design into performance practice?*

A complementary strand engages directly with the breakdown and misuse of generative models. Broad et al. (2021) formalise active divergence as a framework for deliberately pushing generative models beyond their training distributions, identifying a taxonomy of strategies including data-level, network-level, and training-level interventions. The framework provides useful vocabulary for practices that treat out-of-distribution behaviour not as failure but as compositional material — a perspective with roots in the aesthetics of failure articulated by Cascone (2000). In the audio domain, network bending — real-time modification of neural network parameters during sound generation — has been explored as a form of creative misuse grounded in the ethos of circuit bending, treating instability and breakdown as sites of musical invention (Kotowski and Font, 2026). A related strategy can be applied to autoregressive generation: feeding a prior's own output back into its context produces controlled departures from learned temporal structure (Kotowski, 2025). These approaches share a commitment to engaging with the specific material behaviours of neural systems rather than abstracting them away behind conventional control paradigms.

2.3 Materiality and Affordances

The affordance analysis threaded through this paper draws primarily on Davis and Chouinard (2016)'s framework, which moves beyond binary accounts of what artefacts do or do not afford, proposing instead that artefacts *request*, *demand*, *allow*, *encourage*, *discourage*, and *refuse* specific interactions depending on the user and context. Davis (2023) extends this framework to machine learning systems, treating ML applications as objects of design whose features invite situated affordance analysis. The present work applies this lens to ask how the architectural and training-time decisions shape a neural synthesiser's affordances in performance. This situated understanding aligns with Gibson's ecological formulation, in which affordances are always relative to the organism that encounters them (Gibson, 1979).

Several perspectives inform the approach taken here. Technological mediation theory grounds how artefacts actively shape the practices they participate in (Verbeek, 2005), a stance extended in postphenomenological studies of computational systems and in analyses of how specific architectural decisions yield distinct situated affordances across contexts (Gerlek and Weydner-Volkmann, 2025; Kotowski et al., 2025). Benjamin et al. (2021) reframe ML uncertainty as a design material rather than an obstacle. In sound, Fell (2022) argues from direct practice that tool-making and creative work are inseparable, and Scurto et al. (2021) that small-data regimes and first-person experimentation are legitimate modes of inquiry into ML — the methodological stance adopted here.

3 SYSTEM ARCHITECTURE

PLAUD's architecture consists of two loosely coupled components: a DDSP-based neural synthesizer with a variational latent space, and an optional autoregressive prior. Together they define the instrument's sonic vocabulary and temporal tendencies.

3.1 NoiseBandNet-Based Synthesis Model

The synthesis model is based on NoiseBandNet (Barahona-Ríos and Collins, 2024), a DDSP architecture replacing harmonic oscillators with a filterbank of narrow noise components. The output is a weighted sum of M pre-rendered noise bands:

$$y(t) = \sum_{m=1}^{M} a_m(t) \cdot n_m(t)$$

where $n_m(t)$ is the m-th band and $a_m(t)$ its predicted amplitude. The bands are pre-computed by filtering white noise through M adjacent FIR filters tiling $[0, F_s/2]$ and stored as loopable wavetables. This amounts to additive synthesis with noise bands as base functions, preserving interpretability while avoiding inductive bias toward harmonicity.

> *Affordance note.* The transparency of the synthesis chain — fixed noise bands, predicted amplitudes — *allows* direct intervention in the signal path. This interpretability *encourages* creative manipulation that would be opaque in less structured approaches, such as neural codec-based synthesis. Several such interventions are described in section 4.4.

NoiseBandNet was conceived for sound effects, where pitch-based assumptions do not hold. This makes it a great candidate for a universal audio synthesis architecture. The model predicts at an internal rate F_s/W and upsamples through linear interpolation. W defines temporal resolution, while M affects reconstruction quality and model size. In its original formulation, NoiseBandNet uses pre-engineered features (loudness, spectral centroid) rather than a learned latent space — departing from DDSP's encoder-derived z. This suits extremely compact

datasets (∼4 to ∼95 seconds in the original paper), where sonic variance is adequately captured by a few acoustically grounded features.

Affordance note. This small-data regime *encouraged* engagement with compact, hand-curated collections, steering early development toward treating corpus curation as a compositional act.

Experiments with fitting more complex material (i.e. 30 minutes of drone music composed by the first author) revealed that loudness and centroid constrain the model to dominant timbral states: overlapping regions in the feature space force the network to average across perceptually distinct sounds. We substituted these inputs with a 4-dimensional variational latent space, letting the encoder discover parameters describing the data distribution on its own. Although in the original DDSP formulation, the encoder is preserved for style-transfer, we use it exclusively to learn the latent space during the training; at the inference time, only the decoder is used.

Affordance note. Latent navigation *allows* a more exploratory control approach, shifting the performer's relationship from interpretive adjustment of known quantities toward discovery within an abstract space whose meaning is learned through practice.

Live performance later motivated reintroducing loudness and centroid alongside the latents, now reduced to two dimensions. The resulting hybrid control space provides minimal perceptual anchoring while retaining the exploratory character of the latent space.

3.2 Latent Space Regularisation

The latent space in the original DDSP architecture is unregularised, allowing the encoder to organise representations arbitrarily. In practice, this produces latent topologies with narrow, concentrated clusters separated by unpopulated gaps. In such a topology, spectrally similar timbres may end up distant in Euclidean terms, and small parameter changes can cross empty regions to produce disproportionate sonic jumps. We address this by training the encoder as a variational autoencoder (Kingma and Welling, 2019), adding a KL divergence term to the loss, weighted by a factor β (Burgess et al., 2018):

$$\mathcal{L} = \mathcal{L}_{\text{reconstruction}} + \beta \cdot D_{\text{KL}}(q(z|x)\|p(z)) \qquad (1)$$

Higher β values push the encoder toward a denser, more uniformly populated latent manifold (fig. 2). The effect is directly relevant to performance: a space with fewer gaps means that continuous movement in any direction is more likely to produce continuous timbral change, better satisfying established criteria for digital musical instrument design where parameter changes should yield proportional sonic responses (Jordà, 2004). Conversely, lower β values preserve more of the data's intrinsic structure at the cost of navigability — the resulting space is more expressive in principle, but harder to traverse smoothly.

Affordance note. The regularised, continuous latent space *encourages* exploratory navigation by making the space more uniformly responsive to control input. A space with fewer gaps means that small changes in control are more likely to produce correspondingly small changes in the output, a property Jordà (2004) identifies as necessary for the development of expressive control. Conversely, lower β values *discourage* casual traversal by concentrating representational density in specific regions, demanding more precise, deliberate control from the performer.

In training, the β factor is increased from 0 to its target value across epochs 100–200 in a warmup schedule, avoiding premature posterior collapse that would flatten the latent space before the decoder has learned useful reconstructions.

3.3 Adversarial Training

Multi-scale spectral (MSS) losses tend to produce oversmoothed synthesis parameters, suppressing fine spectral detail — an effect demonstrated in the DDSP context by Wu et al. (2022). We observed the same training PLAUD on noisy, detail-rich textures. Additionally, spectral losses fall short in optimising for pitch correctly (Turian and Henry, 2020). Adding a MelGAN-style adversarial feature loss (Kumar et al., 2019) helps recovering spectral detail, especially in higher parts of the spectrum. The discriminator consists of three sub-discriminators, each operating at a progressively $4\times$ downsampled scale, with three layers each, producing a multi-scale feature matching loss. The DDSP model serves as the generator.

Notably, adversarial training can worsen the MSS reconstruction metric while improving perceptual quality — the discriminator pushes toward spectral detail that averaging-prone reconstruction losses actively suppress (see fig. 3). To avoid instability, adversarial training activates after 200 epochs of MSS+KLD training, functioning as perceptual fine-tuning rather than a primary objective.

Affordance note. Adversarial training allows inclusion of more texturally rich, noisy training material that MSS-only regimes discourage, broadening the range of sound corpora the system can meaningfully learn from.

3.4 Latent Smoothing and Short-Term Temporal Context

At PLAUD's default resampling rate of $W = 32$ at 44.1kHz, the control signal operates at 1378 Hz — fine-grained enough that the encoder produces highly dynamic latent trajectories. Reproducing such sequences manually is unfeasible, making it difficult to reproduce the temporal structures embedded in the data through direct latent navigation.

Inspired by the smoothing strategy in Sketch2Sound (García et al., 2025), we smooth the latent trajectories during training using a moving average filter:

$$\tilde{z}_l(t) = \frac{1}{K} \sum_{k=0}^{K-1} z_l(t+k), \quad l = 1, \ldots, L$$

where L is the number of latent dimensions and K the kernel size. This removes high-frequency content from the trajectories, encouraging the GRU of the decoder to internalise more of the short-term temporal structure. Where in standard DDSP the GRU encodes minimal context, smoothing pushes it to preserve longer continuity — the same latent point will sound differently depending on the trajectory leading to it. This frees the optional prior network

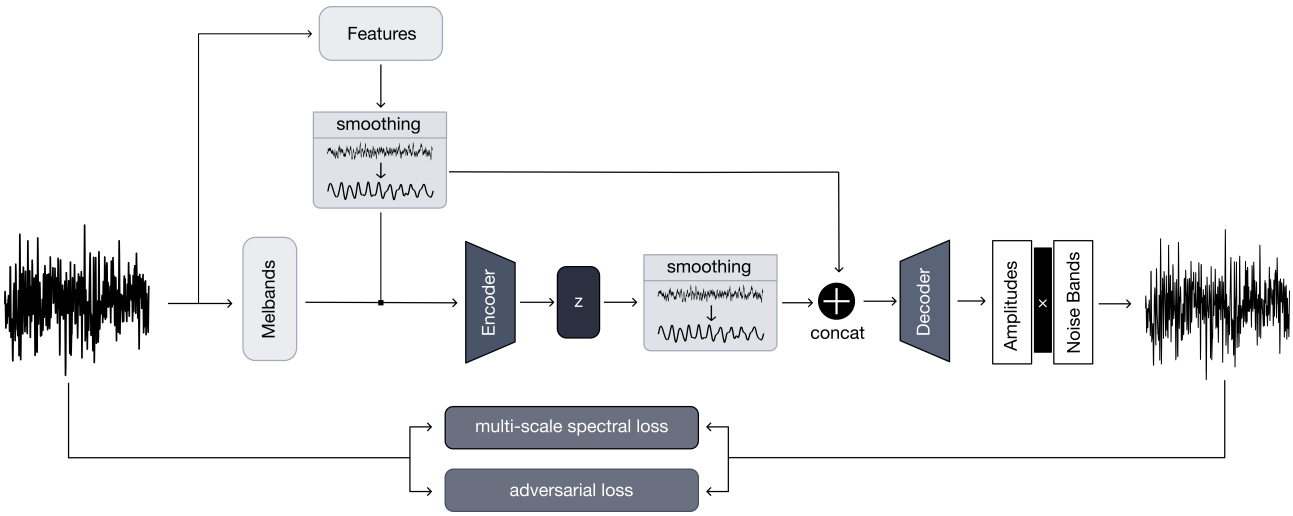

Figure 1: PLAUD synthesis model architecture. During training, mel-band features feed the encoder (producing latent z), while raw audio feed a parallel feature extraction path; both are smoothed before concatenation and decoding into per-band amplitudes, which are multiplied with pre-rendered noise bands. Training combines multi-scale spectral and adversarial losses. In inference, only the decoder is used. The prior network is not shown.

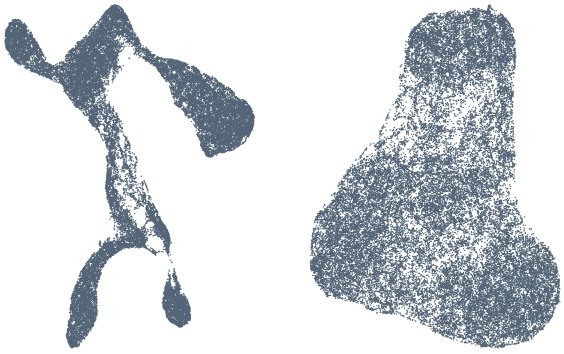

Figure 2: Latent space projections from models trained on the same corpus with $\beta = 10^{-3}$ (left) and $\beta = 10^{-2}$ (right). Stronger regularisation yields a denser, more uniformly populated manifold, reducing the unpopulated gaps that make continuous navigation unpredictable.

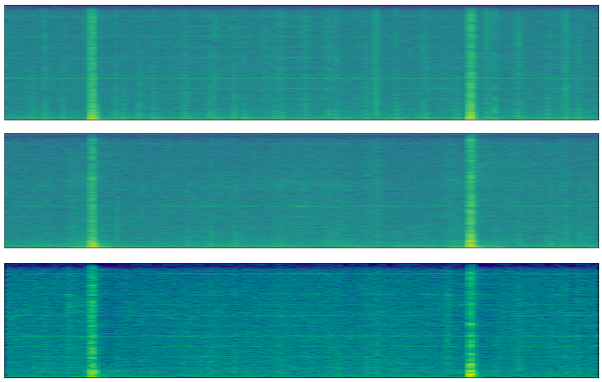

Figure 3: Log-magnitude spectrograms of training audio (top), reconstruction without adversarial training (middle), and reconstruction with adversarial training (bottom). The adversarially trained model recovers finer spectral detail, visible as sharper horizontal structures in the upper frequency range and increased contrast in transient regions, compared to the smoother, more diffuse reconstruction produced by MSS loss alone.

(section 3.5) to operate at a higher temporal level (section 3.6).

Affordance note. Smoothing *allows* synthesis of temporally complex structures from relatively simple latent modulation (see fig. 4). The GRU enacts movement once a position in the latent space is established. This inspired the modulation-based control mode described in section 4.2.1. A related idea of leveraging modulation patterns within DDSP has been explored independently by Mitcheltree et al. (2025a).

3.5 Transformer-Based Autoregressive Prior

The prior is an optional component, meaning that the synthesizer – controlled directly through parameter modulation – is fully functional without it. In training, the prior learns temporal structure from the corpus: rhythmic patterns, envelopes, and longer-range sequential dependencies that the GRU-based decoder does not capture.

The architecture is an encoder-only transformer operating autoregressively over the synthesizer's discretised control space. Following a similar approach to the RAVE prior (Caillon and Esling, 2021), the continuous control signals are quantized into Q classes per latent dimension through mu-law companding:

$$c_q = \text{round}\left(\frac{Q-1}{2} \cdot \frac{\ln(1+\mu|z|)}{\ln(1+\mu)} \cdot \text{sgn}(z) + \frac{Q-1}{2}\right)$$

The algorithm nonlinearity allocates finer resolution near zero, matching the concentration of values around the VAE's approximately normal distribution. We train with $Q = 32$ classes, which provides sufficient resolution to reconstruct meaningful control trajectories while keeping the vocabulary compact. The transformer receives quantized codes as token embeddings with positional encoding and predicts the next token per dimension. It is trained via cross-entropy loss. At inference, logits pass through softmax and are dequantized back to continuous values. Since models share the same control parameter space, any

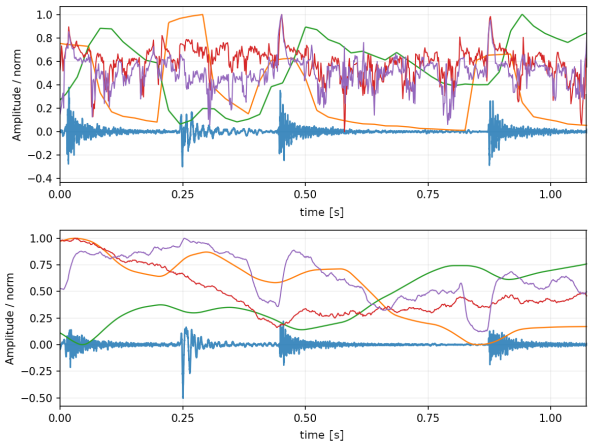

Figure 4: Control trajectories over a 1 second excerpt, without smoothing (top) and with smoothing at kernel size $K = 257$ (bottom). Each coloured line represents one control dimension; the blue waveform shows the corresponding audio. Without smoothing, the trajectories are highly dynamic and tightly coupled to the audio's micro-temporal structure. After smoothing, the same dimensions trace slower, more gestural contours — delegating short-term continuity to the GRU and making the latent space navigable through simple modulation.

prior can drive any synthesizer. Pairing synthesisers with priors trained on distinct corpora produces a loosely interpreted form of style transfer: the temporal structure of one dataset — its phrasing, rates of change — is imposed onto the timbral vocabulary of another, or the other way around.

Affordance note. Combining priors and synthesizers from distinct corpora *allows* cross-corpus composition. The softmax temperature allows balancing between closer reproduction of learned patterns and increasingly stochastic generation.

Affordance note. The autoregressive nature of the prior — generating one token at a time conditioned on its own recent output — allows for feedback interventions where generated sequences are folded back into the transformer's context. The mechanism is detailed in Kotowski (2025), where destabilising the prior's self-conditioning produced patterns that depart from the learned temporal structure in controlled ways. The technique was subsequently integrated into PLAUD's interface as a performance control (section 4.4).

3.6 Resampling Rate Trade-off

The resampling rate W (the number of audio samples per control frame) determines how often the model predicts, and consequently, how much temporal structure must be delegated to the prior versus resolved directly by the synthesizer. At audio rate F_s, the internal control rate is $f_{int} = F_s/W$, meaning each predicted amplitude frame spans $\Delta t = W/F_s$ seconds. For a prior with a fixed sequence length of L tokens, the duration of audio the prior can condition on equals:

$$T_{RF} = \frac{L \cdot W}{F_s}$$

The control rate and the prior's temporal context move in opposite directions. Lower W yields a higher f_{int}, resulting in sharper transients, and less envelope smearing. At the same time it produces more tokens per unit time, making the prior's sequence modelling task harder and more compute-intensive. Higher W extends T_{RF}, but coarsens the synthesis envelope, blurring rapid timbral changes. The two quantities are bound by a fixed product: $T_{RF} \cdot f_{int} = L$, and cannot be independently optimised (see fig. 6).

There is no universally optimal setting. The choice is tuned to the corpus and the intended use. Texturally dense material demands lower W, while slowly evolving textures tolerate higher values that grant the prior a wider temporal view.

4 INTERFACE AND INTERACTION

The Max for Live instrument exposes PLAUD's latent space, autoregressive prior, and synthesis chain through a set of controls and interaction modes, detailed in the following subsections[1]. The device is designed to engage the material affordances of the underlying networks described in the previous sections, and can in fact be seen as emerging from these affordances as much as from the performance intentions of the first author.

4.1 The Max for Live Instrument

PLAUD is implemented as a Max for Live device, built around the nn~ external[2], originally developed for deploying IRCAM's RAVE models to Pure Data and Max. The synthesizer and prior networks are bundled and exported into an nn~-compatible format.

Affordance note. The reliance on nn~ *encourages* adoption among users who have already experimented with RAVE-based instruments, as no additional externals are needed. The choice of Max for Live *allows* seamless extension through Live's modulation routing, automation, and effect chaining, and *discourages* viewing PLAUD as a closed, self-contained system.

The interface is organised in functional blocks from left to right (See fig. 7). The leftmost section handles model and preset management: up to sixteen parameter snapshots can be saved and interpolated between. Below, a volume control sets the output level and a meter provides visual monitoring. Four control knobs are mapped to the synthesis model's input parameters: loudness, spectral centroid, and two latent variables (section 3.1). Default ranges are $[0, 1]$ for audio features and $[-1, 1]$ for latents, reflecting training data distributions. These ranges are adjustable, allowing the performer to push the model beyond its training manifold into regions where it must interpret unseen control signals. A bypass toggle disables the knobs temporarily.

Affordance note. Adjustable control ranges *allow* the performer to probe the latent space beyond its training distribution and *encourage* experimentation with out-of-distribution behaviour.

The prior control section features an "auto" toggle for engaging the autoregressive prior, temperature and priming

[1] A companion page with video demonstrations of the interactions discussed in this section is available at https://plaudaimc2026.github.io/.

[2] See https://github.com/acids-ircam/nn_tilde.

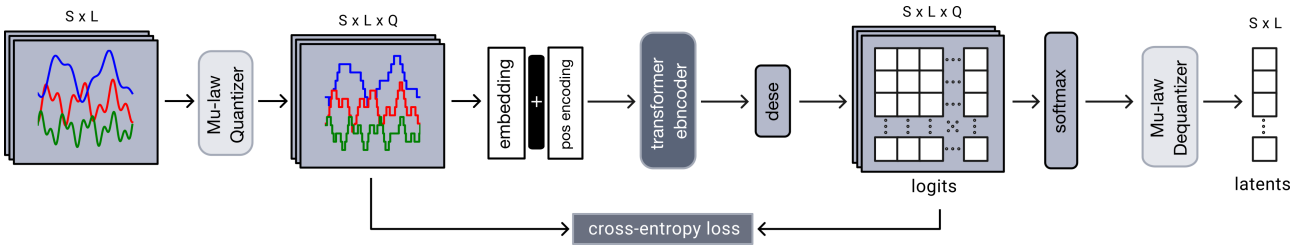

Figure 5: Prior network architecture. Continuous control signals are mu-law quantized into discrete tokens, embedded with positional encoding, and processed by an encoder-only transformer. A dense layer produces logits over the quantization classes, trained with cross-entropy loss; at inference, softmax sampling and mu-law dequantization recover continuous control trajectories.

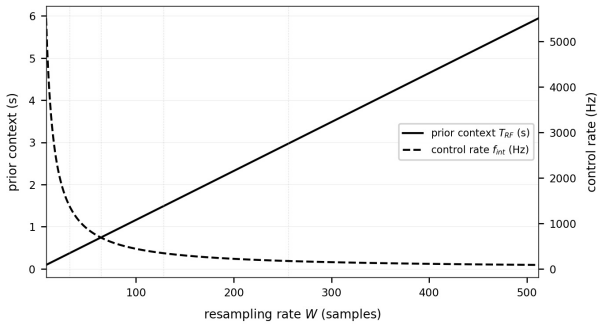

Figure 6: Prior temporal context T_{RF} (solid) and internal control rate f_{int} (dashed) as functions of resampling rate W, for a fixed prior sequence length $L = 512$ at $F_s = 44100$ Hz. Their product is constant at L.

controls (section 4.2.2), and feedback parameters (section 4.4). Adjacent is the trajectory sampling module (section 4.3). The posterior modifiers block allows offsetting and scaling the control streams — whether from the looper or the knobs — alongside two bending controls for waveshaping the synthesis basis functions and adjusting the number of active noise bands (section 4.4). Finally, an abstract visualisation renders a sphere of spokes encoding waveshaping value, partial count, and playback speed.

4.2 Two Sources of Control Signal

The synthesizer's input parameters can be driven by two complementary sources: external modulation and control generation by the prior.

4.2.1 Modulation

As a Max for Live device, PLAUD's parameters are available to any modulation source within Ableton Live: LFOs, sequencers, envelopes, or manual control. These sources produce control trajectories the model never encountered during training.

Affordance note. Modulation is a familiar gesture in synthesizer practice, making this interaction mode immediately accessible to electronic musicians. It allows integration of PLAUD into existing modulation workflows.

4.2.2 Control Generation by the Prior

When the autoregressive prior is engaged, it generates control trajectories in real time, automating the control parameters according to temporal patterns learned from the training data: learned expressive patterns and longer-term temporal dynamics that would be difficult to reproduce manually. The prior can be primed by manual modulation: a short segment of trajectory is fed as context, after which the network continues from where the input left off. This allows the performer to seed generation with a specific starting condition and let the prior develop it further. Temperature controls the softmax distribution over the quantised control vocabulary, controlling predictability of generation. A feedback parameter routes the prior's own output back into its context, described further in section 4.4.

4.3 Trajectory Sampling

Trajectory sampling mirrors audio sampling techniques such as looping, slicing, and transposition, but operates in the control space rather than on audio directly. The module records control trajectories into a buffer, capturing the sum of knob positions and prior generation (if enabled). Playback supports forward and reverse directions with adjustable speed and buffer length. Start and end points define a loop region within the buffer. Two heads — playback and recording — can either move in sync or independently: the record head loops around the entire buffer while playback cycles between the selected start and end points. Operations can be quantised to a grid at selectable resolutions. A real-time preview displays the generated parameter modulation across up to four channels, and the functionality can be applied to a single selected channel or all at once.

Although the interaction vocabulary resembles audio sampling, the nature of neural synthesis augments the generation. Because the smoothed decoder makes the synthesised output depend on the trajectory leading to a given latent point and not only on the point itself (section 3.4), playback speed and direction shape the emerging temporal structures rather than simply compressing or stretching them. The posterior modifiers shift timbral properties of the synthesised output or relocate it within the training data manifold. Quantisation locks operations to the session grid, integrating trajectory sampling into the broader project context.

Affordance note. The familiarity of sampling controls among electronic music producers *encourages* transferring existing intuitions about looping, slicing, and speed adjustment to the control domain. At the same time, the differences arising from neural synthesis *allows* exploration beyond what audio sampling affords.

4.4 Bending Operations

Three operations intervene directly in the synthesis chain and prior generation, collectively referred to as bending op-

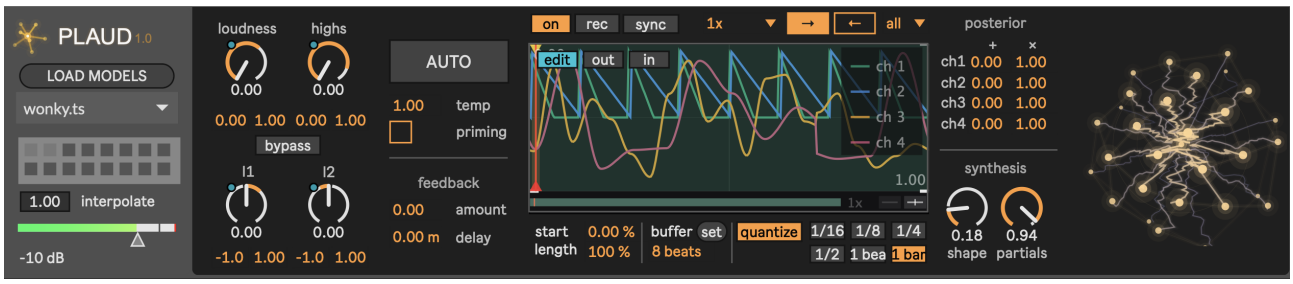

Figure 7: The PLAUD Max for Live interface. From left: model loading, presets and interpolation controls; loudness, highs, and latent dimension control knobs; autoregressive prior controls, latent trajectory and sampling; posterior trajectory modifiers and synthesis bending parameters; bending visualizer.

erations. The term draws from circuit and network bending (Kotowski and Font, 2026).[3]

Component limiting reduces the number of active noise bands used for synthesis. At each frame, the bands with the highest predicted amplitudes are retained and the rest silenced. This produces abrupt spectral discontinuities that occasionally result in clicks, an artefact that proved musically interesting in practice and was therefore kept.

Waveshaping substitutes noise bands with frequency-matched sinusoids and progressively shapes them toward square waves, producing a continuum from noise through sine to square. Both operations exploit the interpretability of the DDSP synthesis chain.

Prior feedback routes the generated latent codes back into the prior's own predictions, with adjustable amount and delay, destabilising generation implementing the mechanism detailed in Kotowski (2025).

> *Affordance note.* Bending operations *allow* moving sound beyond the training data space in a controlled, musically legible way. They *encourage* direct engagement with the materiality of the network.

5 PERFORMANCE REFLECTION

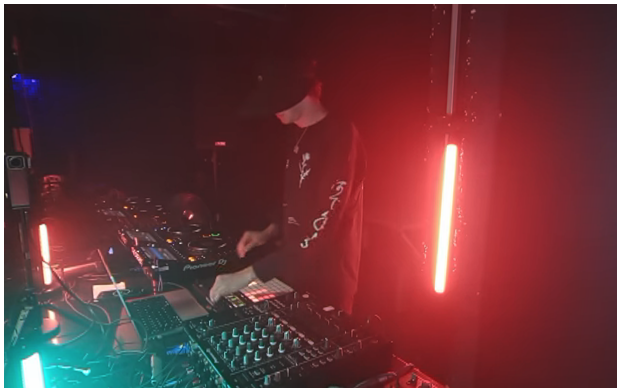

Figure 8: The first author performing with PLAUD in a live club setting. The setup includes a laptop running Ableton Live with the PLAUD Max for Live device, alongside external Push controller.

[3]Whether these interventions constitute network bending in a strict sense is debatable, as they operate on the synthesis chain rather than on weights or activations. However, we believe the term fits broader interpretations and contributes to situating these operations within a lineage of creative misuse.

PLAUD has been used in a range of live electronic music contexts, from open improvisation to more structured sets with prepared presets and loose scores, across four public performances by the first author, who remains its only performer to date. This section reflects on what the system makes possible in performance, and what it resists.

Working with self-curated datasets is central to the practice. Quick training times and small data requirements mean that a different set of models, trained on new material, can be prepared for each performance. This produces performance-specific instruments that differ in reactivity, timbral vocabulary, and overall character. No two are the same, which keeps the process surprising and generative as a creative practice.

The bending operations have proven to be expressive performance tools. They allow dramatic shifts: limiting the number of active components to the minimum produces minimal compositions of unstable sine tones and clicks. Waveshaping these minimal signals introduces synthetic precision, and the full spectral detail of the original model can be restored with a single knob movement.

A recurring observation concerns the relationship between reconstruction accuracy and expressivity. Neither the synthesizer nor the prior achieves high-fidelity reproduction of the training data, and in many cases the reconstruction quality would not satisfy conventional evaluation criteria. This did not prove to be an obstacle. For the synthesizer, the lo-fi character of the output became part of the instrument's identity. For the prior, stable long-term generation was never achieved, because of accumulating prediction errors leading to drifts. In a creative context where faithful reproduction is not the goal, this instability can be seen as an aesthetic affordance. The resulting sound has a character that can be valued in performance as a distinct quality of the neural synthesis, much as certain iconic synthesizers are sought out for their sonic feel.

The trajectory sampling module was the most recently introduced element, and partly serves to integrate PLAUD's generation into more conventional grid-based setups. In practice it has worked well, providing a way to impose structure over the prior's more chaotic tendencies. One effective technique is to load a model trained on rhythmic data, sample its prior generation into the buffer, then switch to a model trained on different material (e.g. metal impacts), and let the pre-sampled trajectories steer the new synthesis. This produces a form of style transfer performed live, combining the temporal structure of one corpus with the timbral vocabulary of another.

One significant limitation is computational performance. Without the prior, up to three model instances can run simultaneously on a MacBook M1 Pro from 2021 with relative stability, but engaging the prior reduces this to one, leaving limited headroom for additional instruments and effects in a Live session. On several occasions this has required compromises in the performance setup, and in a workshop context some participants' hardware could not run the models at all. Reducing this computational footprint remains the most pressing area for future development.

6 CONCLUSION, LIMITATIONS AND FUTURE WORK

PLAUD demonstrates that the expressive affordances of a neural instrument can emerge directly from its architecture: from the constraints of small data training, the structure of the variational latent space, the interpretability of the DDSP synthesis chain, and the temporal tendencies encoded by the prior.

The paper presented a system combining a VAE-regularised NoiseBandNet synthesis model, latent smoothing, adversarial training, and an optional transformer prior, alongside bending operations that intervene directly in the synthesis chain. The Max for Live instrument exposes these components through trajectory sampling, modulation-based control, and prior's automatic steering mode. Throughout, we have argued that the instrument's performative character follows from its architectural decisions rather than being designed on top of them.

The system's audience and aesthetic commitments are both narrow by design. Building on nn$\sim$ situates PLAUD among setups already prepared for RAVE, and packaging it as a Max for Live device rather than a standalone application addresses producers and performers rather than researchers. Sonically, the model was developed for noisy, textural, and unpitched material, and its lo-fi reconstruction, spectral discontinuities, and drifting prior are treated as instrument character rather than as defects to be engineered away, which suits some practices considerably better than others.

Several directions remain open. Computational cost is the most pressing limitation, as discussed in Section 5. On the synthesizer side, the autoregressive GRU is likely a bottleneck. Replacing it with causal convolutions could improve throughput while also offering more explicit control over short-term context length, with potentially interesting architectural affordances of its own. On the prior side, the resampling rate trade-off (section 3.6) makes it unfeasible to learn structure at low resampling rates while keeping compute manageable. An intermediary rate reduction block, in the style of neural audio codecs (Zeghidour et al., 2022; Kumar et al., 2023) applied to the latent signal, could decouple the synthesizer's temporal resolution from the prior's context length, enabling finer detail without proportionally increasing the prior's computational load.

Alternative DDSP decoders such as sinusoidal-plus-stochastic modelling would broaden the timbral range. Preliminary experiments have shown some promise, but unsupervised learning of frequencies present in the signal remains a hard problem (Turian and Henry, 2020), and although promising research exists (Hayes et al., 2023), more work is needed before such decoders can be integrated into the full pipeline.

Finally, the bending vocabulary could be expanded with operations such as spectral rolling, alternative modes of component subset selection, and harmonisation.

AUTHOR DECLARATIONS

The authors used Claude for language editing to improve grammar and clarity. All scientific content, interpretations, and conclusions were developed by the authors.

REFERENCES

Barahona-Ríos, A. and Collins, T. (2024). Noisebandnet: Controllable time-varying neural synthesis of sound effects using filterbanks. *IEEE ACM Trans. Audio Speech Lang. Process.*, 32:1573–1585.

Benjamin, J. J., Berger, A., Merrill, N., and Pierce, J. (2021). Machine learning uncertainty as a design material: A post-phenomenological inquiry. In Kitamura, Y., Quigley, A., Isbister, K., Igarashi, T., Bjørn, P., and Drucker, S. M., editors, *CHI '21: CHI Conference on Human Factors in Computing Systems, Virtual Event / Yokohama, Japan, May 8-13, 2021*, pages 171:1–171:14. ACM.

Broad, T., Berns, S., Colton, S., and Grierson, M. (2021). Active divergence with generative deep learning - A survey and taxonomy. In de Silva Garza, A. G., Veale, T., Aguilar, W., and y Pérez, R. P., editors, *Proceedings of the Twelfth International Conference on Computational Creativity, ICCC 2021, México City, México (Virtual), September 14-18, 2021*, pages 227–236. Association for Computational Creativity (ACC).

Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A. (2018). Understanding disentangling in β-vae. *CoRR*, abs/1804.03599.

Caillon, A. and Esling, P. (2021). RAVE: A variational autoencoder for fast and high-quality neural audio synthesis. *CoRR*, abs/2111.05011.

Cascone, K. (2000). The aesthetics of failure: "post-digital" tendencies in contemporary computer music. *Comput. Music. J.*, 24(4):12–18.

Davis, J. L. (2023). 'affordances' for machine learning. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '23, page 324–332, New York, NY, USA. Association for Computing Machinery.

Davis, J. L. and Chouinard, J. B. (2016). Theorizing affordances: From request to refuse. *Bulletin of Science, Technology & Society*, 36(4):241–248.

Engel, J. H., Hantrakul, L., Gu, C., and Roberts, A. (2020). DDSP: differentiable digital signal processing. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Esling, P., Chemla-Romeu-Santos, A., and Bitton, A. (2018). Generative timbre spaces with variational audio synthesis. *CoRR*, abs/1805.08501.

Fell, M. (2022). *Structure and Synthesis: The Anatomy of Practice*. Urbanomic/MIT Press.

García, H. F., Nieto, O., Salamon, J., Pardo, B., and Seetharaman, P. (2025). Sketch2sound: Controllable audio generation via time-varying signals and sonic imitations. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2025, Hyderabad, India, April 6-11, 2025*, pages 1–5. IEEE.

Gerlek, S. and Weydner-Volkmann, S. (2025). Materiality and machinic embodiment: A postphenomenological inquiry into chatgpt's active user interface. *Journal of Human-Technology Relations*, 3:1–15.

Gibson, J. J. (1979). *The Ecological Approach to Visual Perception: Classic Edition*. Houghton Mifflin.

Hayes, B., Saitis, C., and Fazekas, G. (2023). Sinusoidal frequency estimation by gradient descent. In *IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 2023, Rhodes Island, Greece, June 4-10, 2023*, pages 1–5. IEEE.

Hayes, B., Shier, J., Fazekas, G., McPherson, A., and Saitis, C. (2024). A review of differentiable digital signal processing for music and speech synthesis. *Frontiers in Signal Processing*, Volume 3 - 2023.

Jordà, S. (2004). Digital instruments and players: Part ii-diversity, freedom and control. In *Proceedings of the 2004 International Computer Music Conference, ICMC 2004, Miami, Florida, USA, November 1-6, 2004*. Michigan Publishing.

Kingma, D. P. and Welling, M. (2019). An introduction to variational autoencoders. *Found. Trends Mach. Learn.*, 12(4):307–392.

Kotowski, B. (2025). Broken forecasts: Feedbacking latent generators for sonic instability. In *Proceedings of the International Conference on AI Music Creativity*.

Kotowski, B., Evans, N., Haki, B., Font, F., and Jordà, S. (2025). Exploring situated stabilities of a rhythm generation system through variational cross-examination. In *Proceedings of the 6th AI Music Creativity Conference (AIMC 2025), Brussels, Belgium, September 10-12, 2025*.

Kotowski, B. and Font, F. (2026). Network bending as circuit-bending inspired live neural synthesis hacking. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 419–424. Zenodo.

Kumar, K., Kumar, R., de Boissiere, T., Gestin, L., Teoh, W. Z., Sotelo, J., de Brébisson, A., Bengio, Y., and Courville, A. C. (2019). Melgan: Generative adversarial networks for conditional waveform synthesis. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R., editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14881–14892.

Kumar, R., Seetharaman, P., Luebs, A., Kumar, I., and Kumar, K. (2023). High-fidelity audio compression with improved RVQGAN. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S., editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Mitcheltree, C., Tan, H. H., and Reiss, J. D. (2025a). Modulation discovery with differentiable digital signal processing. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, WASPAA 2025, Tahoe City, CA, USA, October 12-15, 2025*, pages 1–5. IEEE.

Mitcheltree, C., Teleaga, B., Fyfe, A., Masuda, N., Schäfer, M., Bradic, A., and Tokui, N. (2025b). Neutone sdk: An open source framework for neural audio processing.

Privato, N., Shepardson, V., Lepri, G., and Magnusson, T. (2024). Stacco: exploring the embodied perception of latent representations in neural synthesis. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. International Conference on New Interfaces for Musical Expression.

Scurto, H., Caramiaux, B., and Bevilacqua, F. (2021). Prototyping machine learning through diffractive art practice. In Ju, W., Oehlberg, L., Follmer, S., Fox, S. E., and Kuznetsov, S., editors, *DIS '21: Designing Interactive Systems Conference 2021, Virtual Event, USA, 28 June, July 2, 2021*, pages 2013–2025. ACM.

Tahiroglu, K. and Wyse, L. (2024). Latent spaces as platforms for sonic creativity. In Grace, K., Llano, M. T., Martins, P., and Hedblom, M. M., editors, *Proceedings of the 15th International Conference on Computational Creativity, ICCC 2024, Jönköping, Sweden, June 17-21, 2024*, pages 359–363. Association for Computational Creativity (ACC).

Tatar, K., Cotton, K., and Bisig, D. (2023). Sound Design Strategies for Latent Audio Space Explorations Using Deep Learning Architectures. In *Proceedings of Sound and Music Computing 2023*.

Turian, J. and Henry, M. (2020). I'm sorry for your loss: Spectrally-based audio distances are bad at pitch. *CoRR*, abs/2012.04572.

Verbeek, P.-P. (2005). *What things do: Philosophical reflections on technology, agency, and design*. Penn State Press.

Wu, Y., Manilow, E., Deng, Y., Swavely, R., Kastner, K., Cooijmans, T., Courville, A. C., Huang, C. A., and Engel, J. H. (2022). MIDI-DDSP: detailed control of musical performance via hierarchical modeling. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Zeghidour, N., Luebs, A., Omran, A., Skoglund, J., and Tagliasacchi, M. (2022). Soundstream: An end-to-end neural audio codec. *IEEE ACM Trans. Audio Speech Lang. Process.*, 30:495–507.

Zheng, S. J., Xambó Sedó, A., and Bryan-Kinns, N. (2025). Exploring gestural affordances in audio latent space navigation. *Frontiers in Computer Science*, 7:1575202.