Real-Time Conditioning Strategies for Shared Agency in AI-Mediated Piano Improvisation

Àlex Barrachina*
Music Technology Group
Universitat Pompeu Fabra
Barcelona, Spain
abarrachina@esmuc.cat

Lonce Wyse
Music Technology Group
Universitat Pompeu Fabra
lonce.wyse@upf.edu

Sergi Jordà
Music Technology Group
Universitat Pompeu Fabra
sergi.jorda@upf.edu

Abstract

AI-mediated instruments can make pianistic improvisation more accessible by delegating pitch decisions to generative models, but they also raise questions about how agency is shared in real time. This paper explores real-time conditioning strategies for a low-latency Transformer-based symbolic piano generation system in which performers steer pitch generation through a key-based interface. We examine two conditioning approaches: compression-based mappings that reduce the pitch-control vocabulary, and relative-pitch tendency controls that guide melodic motion through interval categories, including buttons that temporarily release constraints. Through prototype-based exploration and a small participant study, we examine perceived control, musical coherence, and performer–model agency tension. We argue this tension need not be fixed at design time. Instead, interfaces can make the trade-off legible and allow performers to negotiate between control and autonomy during play—turning constraint into an expressive resource. We identify design principles for such hybrid-control instruments, including the "joker" mechanism that lets performers temporarily release constraints.

1 INTRODUCTION

1.1 Background on pianistic expressiveness and AI-mediated instruments

Good musical instruments must negotiate a balance between accessibility and depth: interfaces that are too simple may plateau quickly, whereas overly complex systems can impose a learning burden that prevents users from reaching their expressive potential. The piano exemplifies this balance: its direct one-to-one mapping between keys and pitches allows even non-musicians to improvise music that can be immediately pleasing or "beautiful," yet expressive mastery and reliable control over nuanced musical outcomes typically require long-term technical training and sustained practice (Jordà, 2005).

AI-mediated instruments promise to lower the "entry fee" to a musically convincing output by delegating parts of the performance to an algorithmic system. However, such systems often trade away "feel" or playability. In practice, many AI music systems prioritize "correctness" (plausible continuations) while exposing only coarse or high-level controls, making intentional shaping difficult during live play.

This motivates a central tension in AI-mediated instruments: how musical agency is distributed between performer and generative system in real time. This distribution involves two complementary dimensions. Performer agency refers to the ability to intentionally shape musical outcomes during performance, while system agency refers to the system's capacity to produce stylistically plausible output under limited guidance. In piano-oriented systems, this becomes a trade-off between performer agency and what we term pianistic coherence: the model's ability to sustain idiomatic piano texture, harmony, and phrasing over time. We use "pianistic coherence" pragmatically, referring to whether generated pitch sequences remain consistent with the statistical and stylistic properties of the training corpus (e.g., voice-leading continuity, register usage, and local harmonic plausibility). Rather than offering a formal music-theoretic definition, we treat coherence as a combination of model-prior consistency and perceived musical plausibility, reflected in participant responses.

Shared-control interfaces often force a compromise: increasing low-level control constrains the model's generative competence, while increasing autonomy can weaken the performer's sense of agency and real-time steerability.

1.2 Research focus and objectives

This paper presents an exploratory study of real-time conditioning strategies in AI-mediated instruments for pianistic improvisation. Our objective is to examine how different conditioning strategies reallocate musical decision-making between performer and model, and how these choices affect perceived performer guidance and idiomatic output.

We describe two preliminary families of real-time conditioning strategies in order to explore this performer–model agency tension. To keep conclusions interpretable, we study a constrained setting where the model generates only pitch tokens while timing, velocity, and duration do not influence model inference. Within this setting, we ask how changes in pitch conditioning alter (i) the performer's ability to intentionally steer outcomes, and (ii) the model's role in realizing specific pitch continuations, and (iii) the performer's perception of control, coherence, and shared agency during interaction.

1.3 Contributions of the study

This work makes three main contributions:

*Video demonstration and source code available

- **Real-time conditioning strategies for shared control.** We present and analyze two pitch-conditioning paradigms—compression-based mappings and relative-pitch tendency control—that enable low-dimensional, real-time interaction with a symbolic generative piano model.
- **Design-oriented characterization of performer-model agency negotiation.** Through prototype exploration and a small participant study, we examine how different conditioning schemes redistribute musical decision-making between performer and model, and how this affects perceived agency and musical coherence.
- **Interface design implications for AI-mediated improvisation.** We identify design principles for shared-control instruments, including making control trade-offs legible and supporting hybrid interaction strategies that allow performers to negotiate between autonomy and control during performance.

2 BACKGROUND AND RELATED WORK

2.1 Human–machine agency in AI-mediated instruments

A useful lens for AI-mediated instruments is to treat musical outcomes as emerging from a distribution of agency across performer, system, and context, rather than as the product of a single controlling subject. Within this perspective, agency is graded and relational: interactive musical practice can be understood as an "agency network" in which different components exert different degrees of influence on what is played and what is perceived as intentional (Brown, 2016). Recent conceptualizations of human–AI creativity further clarify how such distributions are experienced in practice. The Extended Creativity framework characterizes human–AI relations along two dimensions—system autonomy (capacity to operate independently) and perceived agency (the extent to which the system is experienced as an influential participant)—and proposes multiple relational modes (e.g., tool-like support vs. collaborative partnership) (Gaggioli et al., 2025). Applied to musical interaction, this suggests that engaging AI as a co-creative partner often requires performers to relinquish the expectation of complete control and instead develop situated strategies for sharing influential agency with the algorithmic process (Dahlstedt, 2021). In parallel, work on negotiating autonomy in performance highlights that such collaborations benefit from journeys through control: performers progressively explore, calibrate, and sometimes intentionally loosen control to establish trust and musical flow with autonomous behavior (Benford et al., 2024, 2021). Taken together, these perspectives help to articulate the central tension of this paper: in musical instruments, AI systems can increase musical plausibility while simultaneously reducing instrument-like controllability. The same mechanism that supports coherence can also weaken the performer's ability to intentionally shape outcomes at performance timescales.

2.2 Controllable and conditioned symbolic generation

The idea of distributing musical control between performer and system long predates neural models, dating back to the early 1980s in "interactive" and "intelligent" instrument traditions described by Chadabe Chadabe (1984) and Spiegel Spiegel (1987), respectively. Within contemporary neural architectures, systems such as MuseNet (Payne, 2019) and Music Transformer (Huang et al., 2018) introduced global conditioning mechanisms that allow outputs to be guided by fixed attributes (e.g., melody-conditioned sequence-to-sequence). More recent models, such as FIGARO (von Rütte et al., 2023) or Anticipatory Music Transformer (Thickstun et al., 2023), provide advanced finer-grained control by integrating explicit control signals within seq2seq pipelines or interleaving event and control sequences. These methods demonstrate that symbolic generation can, in principle, be made steerable, but they are typically designed for offline or semi-offline use, where control signals are prepared in advance and interaction does not need to remain legible at performance rates.

2.3 Real-time generative systems with performer control

Most real-time generative systems that expose control operate as fully or semi-autonomous agents that react to performers rather than function as instruments affording continuous, instrument-like manipulation. Representative examples include Realchords (Wu et al., 2024), which uses reinforcement learning to improvise real-time chord accompaniment to live melodic input, and GrooveTransformer (Evans et al., 2024), a Eurorack generative drum sequencer that leverages a variational autoencoder to enable real-time user interaction while serving as an accompaniment generator.

Only a smaller subset of systems aim for token-level or event-level steerability in real time, where performer actions directly sculpt generation as it unfolds—most notably Piano Genie (Donahue et al., 2019), which employs a latent space quantization strategy to map simplified control inputs to complex outputs, enabling novice-friendly improvisation, and Notochord (Shepardson et al., 2022), a low-latency model enabling fine-grained, interactive manipulation of the generative process that has been proposed as a general-purpose backbone for tasks such as steerable generation or harmonization. Taken together, these systems highlight a central design issue: real-time control can be achieved either by granting the system substantial autonomy (agent-like behavior) or by increasing the bandwidth and immediacy of human input (instrument-like behavior). The trade-offs between these choices are not yet well characterized, particularly for piano-like interaction where stylistic coherence and fine control are both salient.

2.4 Prior explorations of shared control in pianistic interactive systems

Long before current generative models, pianistic interactive systems explored shared control by dividing musical responsibility between human and machine. Early systems such as Mathews' *Conductor Program* (1991) separated pitch structure from expressive execution, allowing performers to shape timing and dynamics while pitch was algorithmically specified . Malinowski (2007) similarly examined expressive performance under programmatically determined pitch constraints, while Gurevich and colleagues' one-button instrument (2010) showed that recognizable performance strategies can emerge even when note-level choice is severely limited.

A related line of work approaches shared control through interactive continuation, where the system maintains stylistic continuity while the performer shapes the unfolding

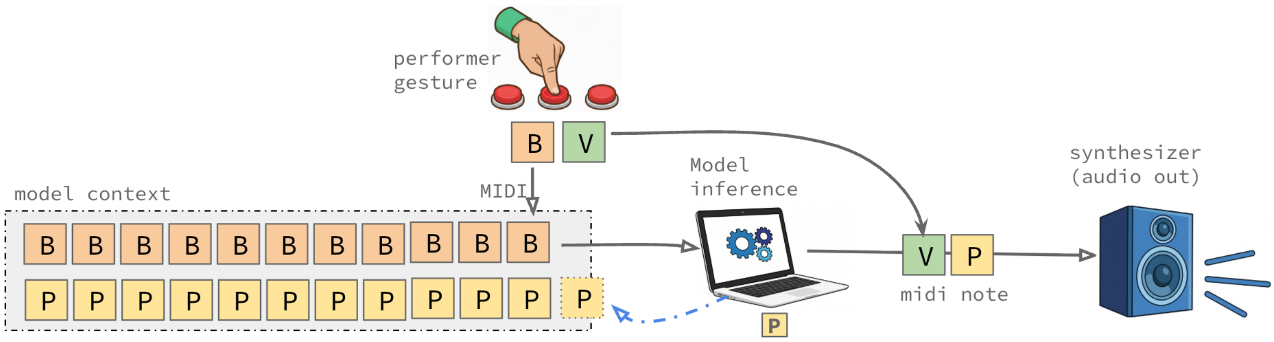

Figure 1: *Dataflow of the AI-mediated instrument. Performer gestures are encoded as conditioning tokens, appended to the model context, and used to bias next-token pitch inference before MIDI/audio rendering.*

performance through interaction. Pachet's *Continuator* (2003) frames improvisation as a dialogue with a style-learning system, and more recent work by Bradshaw et al. (2025), in Disklavier-based human–AI duet studies, examines whether generative models can produce coherent phrasal responses to human pianists.

These systems show that shared control, constraint, and distributed musical responsibility are longstanding concerns in interactive music. However, there remains a need to examine the agency–coherence tension in AI-mediated instruments where performer actions sculpt generation event by event under interactive latency constraints.

3 SYSTEM OVERVIEW

3.1 Technical foundation

The prototypes presented here share a low-latency autoregressive Transformer foundation, trained on symbolic piano performance data from the GiantMIDI-Piano dataset (Kong et al., 2020) and optimized for real-time performance. While this dataset reflects a bias toward notated classical repertoire, it provides a large corpus of symbolic piano performance data, making it suitable for studying pitch-level control under controlled stylistic conditions.

The prototypes are conceived as real-time improvisational instruments in which the performer remains continuously "in the loop" of generation. They share a common real-time conditioning architecture: performer gestures are converted into discrete conditioning tokens that bias next-token pitch generation, while the specific mapping between gesture and conditioning token changes across prototypes. Interaction is performed through a custom key-based controller whose layout varies across prototypes: the compression interfaces use reduced button vocabularies such as 1, 8, or 19 buttons, while the relative-pitch interface uses arrow-labeled buttons for interval tendencies plus joker buttons for constraint release.

Generation proceeds token-by-token. Each user action on the controller is captured, encoded as a conditioning signal, and injected into the model's input stream. At each step, the performer's most recent gesture modulates the next-token probability distribution, enabling the user to steer the evolving pitch sequence in real time instead of specifying a complete target sequence offline.

All systems are framed as improvisational continuations in which the instrument aims to preserve the stylistic and harmonic context of an initial prompt. To initialize this context, we seed the model with a short excerpt of MIDI extracted from a human piano performance, selected by the performer from a dataset comprising thousands of performances, which serves as the starting window of musical material that will influence the style of the improvisation.

3.2 Interaction paradigm

To preserve a familiar sense of pianistic playability, interaction is mediated through a key-based interface that retains the core "one key press corresponds to one sounded event" metaphor. However, unlike a conventional piano, the pressed key does not necessarily specify an exact pitch. Instead, each keystroke provides a conditioning token that shapes the model's next pitch prediction.

In this paper, shared control is limited to pitch. On each keystroke, the performer's current gesture is appended to the model context—the running sequence of past gestures and previously generated pitches—thereby biasing the next pitch-token prediction. Timing, velocity, and duration are not used for model inference and remain under direct performer control in the audio/MIDI rendering layer: the performer freely determines onset time, key velocity, and note length for each event, while the model determines, and the performer steers, the pitch sequence.

This restriction allows us to isolate pitch conditioning as the primary site where performer-model agency is redistributed. This simplification necessarily limits the scope of our conclusions, as timing and dynamics are known to contribute strongly to pianistic expressiveness; our results should therefore be interpreted as specific to pitch-mediated interaction. The following sections therefore compare different ways of encoding performer intention as pitch-conditioning information: compressed pitch mappings, relative-pitch tendency controls, and release mechanisms that temporarily reduce conditioning pressure.

4 EXPLORING REAL-TIME CONDITIONING STRATEGIES

This section examines two design variables for real-time pitch conditioning: **pitch-control resolution**, explored through compression-based mappings, and **control representation**, explored through relative-pitch tendencies. The joker mechanism extends the relative-pitch interface by adding a way to temporarily release interval constraints, allowing performers to alternate between guiding, constraining, and releasing model generation.

4.1 Compression-based pitch conditioning

Our first conditioning strategy reduces the dimensionality of pitch input by learning a compressed control vocabulary that the performer can play in real time. The core idea is to replace the full 88-key pitch input with a smaller set of discrete control symbols, or "buttons," so that performers can guide pitch generation through a low-dimensional input while the model realizes the final pitch sequence. We implement this compression using a variational autoencoder (VAE) trained to encode pitch events into a limited set of categories and to decode those categories back into pitches in a musically structured manner. In performance, the model is conditioned on the performer's button stream rather than on raw pitch values.

This strategy is conceptually inspired by Piano Genie (Donahue et al., 2019), which maps the 88-key pitch space to a reduced real-time control vocabulary. Our aim, however, is not to replicate Piano Genie or provide a like-for-like comparison. Instead, we use pitch-space compression as a design probe for studying how dimensionality reduction redistributes musical decision-making between performer and model in an autoregressive, real-time Transformer setting. Unlike Piano Genie, our system conditions only on pitch-control tokens, while timing, velocity, and duration remain outside model inference. The model is optimized for low-latency real-time inference (see Appendix A).

4.1.1 Compression level and pitch-control resolution

To examine how pitch compression affects performer–model interaction, we trained several mappings at different compression levels. These range from an uncompressed 88-button condition, where the performer effectively specifies pitch directly, to a one-button condition, where the conditioning signal contains no pitch-specific information and the model largely determines the continuation.

Between these extremes, reduced vocabularies such as 8- and 19-button mappings provide different degrees of pitch-control resolution. The 8-button interface gives the performer a compact melodic-shape vocabulary while leaving substantial freedom to the model. As compression is relaxed, as in the 19-button interface, the performer has a finer-grained vocabulary for conditioning the generated pitch trajectory: the button sequence can transmit more information about the intended melodic shape, while still not functioning as direct pitch selection. In this sense, compression level determines how pitch decision-making is distributed between performer guidance and model realization.

Table 1: Mapping strategies

Buttons	Mapping strategy	Performer ctrl.	Autonomy
88	1:1 Absolute	Maximal	Zero
19	Low compression	High	Reduced
8	Reduced vocabulary	Reduced	High
1	Total compression	Zero	Total

4.2 Relative pitch control

In the compression-based approach, producing musically convincing continuations with higher-resolution compressed mappings requires a degree of dexterity approaching conventional keyboard performance. To explore an alternative representation of control, we investigate a paradigm based on relative pitch tendencies rather than absolute pitch targets. Relative-interval controllers can be considered information-efficient for melodic guidance: with a small set of discrete actions, performers can generate a wide range of trajectories by steering transitions between notes rather than selecting notes directly (see Appendix B).

This strategy is conceptually inspired by interval-relative instruments such as the Samchillian Tip Tip Tip Cheeepeeeee (Gruenbaum, 2007), where performer input specifies motion rather than absolute pitch. Our aim, however, is not to reproduce the Samchillian or provide a direct comparison, but to adapt the broader idea of relative control to a symbolic generative setting in which exact pitch realization remains model-dependent. For this purpose, we define a deterministic mapping from pitch intervals to a small set of categorical control symbols, termed arrows, which represent coarse-grained interval classes:

- Upward large leap: $\geq$ +8 semitones
- Upward medium leap: +3 to +7 semitones
- Upward small step: +1 to +2 semitones
- Unison: 0 semitones (repeat / stay)
- Downward small step: -1 to -2 semitones
- Downward medium leap: -3 to -7 semitones
- Downward large leap: $\leq$ -8 semitones

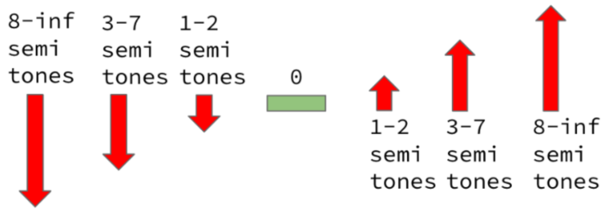

Figure 2: Relative-pitch tendency controls used in the arrows interface.

Unlike the Samchillian, where cumulative interval choices directly determine the pitch trajectory and can produce tonal or registral drift unless the performer continuously compensates, our system delegates exact pitch realization to the model. The performer specifies a local interval tendency, biasing the model toward a pitch compatible with that tendency and the model's recent context.

Because this mapping is deterministic, it does not require a learned encoder: given a target melodic interval, the corresponding arrow token is computed directly. During training, the arrow sequence is provided as an additional conditioning feature aligned with the pitch-token stream. We implement this in a Transformer decoder by embedding arrow tokens, fusing them with pitch-token embeddings, and adding auxiliary losses that encourage generated pitches to respect the specified tendency class.

4.2.1 Relative-interval conditioning and constraint profiles

The arrows interface changes the representation of control from compressed pitch categories to relative interval classes. Rather than conditioning the model with a learned button category, each arrow specifies a local interval tendency. These arrow categories should not be understood as a simple linear scale of pitch-control resolution. Each arrow defines a different interval-class constraint with its own statistical and musical profile. Small-step arrows restrict the model to one or two semitones of movement. Medium-leap arrows constrain the model to a different interval region. Large-leap arrows cover a wider numerical range, but they also exclude all intervals smaller than eight semitones. Thus, they do not necessarily provide more model freedom; they impose a different constraint profile. For this reason, the arrows interface is better described as a non-uniform interval-conditioning space rather than as a uniformly low- or high-resolution controller.

4.2.2 Joker controls as constraint release

To avoid forcing an interval tendency at every event, we add a "joker arrow" representing “no tendency constraint.” We train it by pairing the joker token with random pitch differences so the model learns that it carries no informative interval signal. At inference time, joker inputs remove the current interval-class constraint: the performer stops specifying a direction or interval range, and the model predicts the next pitch from the recent musical context without an explicit arrow bias. The joker token therefore extends the arrow vocabulary with a no-constraint condition. It differs from the other arrow classes because it does not specify a target interval region. Instead, it allows the model to use the accumulated autoregressive context as the primary basis for the next pitch prediction. In this sense, the joker functions as a local release from relative-interval conditioning.

5 PARTICIPANT STUDY

To support the design reflections developed through the prototype explorations, we conducted a small participant study focused on how performers experienced the conditioning strategies described above. The study examined perceived agency, musical coherence, playability, and the felt distribution of control between performer and model.

The study was designed as a lightweight qualitative probe rather than a statistically powered evaluation. Given its small sample size, brief exposure time, and reliance on self-report interviews, the results are treated as design-oriented evidence rather than as a definitive comparison of interfaces. Representative participant comments and supplementary observations are provided in Appendix C.

5.1 Participants, interfaces, and procedure

Eleven participants took part in the study: four non-musicians, three musicians who were not pianists, and four pianists. Participants tested four interfaces: a 1-button compression interface, an 8-button compression interface, a 19-button compression interface, and an arrows+jokers interface combining relative-interval conditioning with temporary constraint release. After a short introduction to each interface, participants improvised briefly and first answered open-ended questions about their overall impressions, followed by specific questions on perceived control and musical coherence, in a semi-structured format. Responses were analyzed using a hybrid deductive–inductive thematic analysis (Fereday and Muir-Cochrane, 2006; Braun and Clarke, 2006): control and coherence served as a priori themes, while sub-themes and additional themes were coded inductively.

5.2 Main observations from the participant responses

Participant’s responses suggest several recurring observations:

1. **The 1-button interface was perceived as almost uncontrollable, but musically coherent.** Participants generally felt that they had no influence over pitch, and that the model was carrying the musical generation. At the same time, the output was often perceived as coherent or musically convincing.

2. **The 19-button interface was the strongest compressed-control condition.** Across non-musicians, musicians, and pianists, participants generally preferred the 19-button interface among the compression-based conditions. They described it as giving them more ability to shape the music, with a clearer relation between physical layout and pitch or register movement.

3. **The 8-button interface reduced effort but limited precision because of its smaller control space.** Participants frequently reported that it produced reasonable musical results with little effort, but that it was harder to guide specific notes, melodic direction, or local detail because of the limited button vocabulary.

4. **Musical coherence did not appear to depend on compression level alone.** The responses do not strongly support a simple claim that the 8-button interface is necessarily more coherent than the 19-button interface. Instead, participants’ comments suggest that coherence also depends on how effectively the performer can guide the model within the available control vocabulary.

5. **Accepting a guiding role reduced confusion.** Prior keyboard expectations sometimes made the compressed mappings confusing, especially when participants expected piano-like pitch behavior. Participants seemed more comfortable when they accepted their role as guiding or collaborating with the model rather than trying to fully specify every note.

6. **The arrows interface was widely seen as interesting and expressive, but cognitively demanding.** Several participants felt that arrows gave a more explicit or direct sense of influence than the 8-button interface, despite having a similar number of controls. However, the interface was also described as harder to learn and less immediately intuitive.

7. **The joker controls were one of the strongest positive findings.** Participants repeatedly described jokers as useful for reset, rescue, and reorientation, especially in the arrows interface, where it was easy to lose track of the resulting pitch direction.

8. **A recurrent weakness was local coherence without long-range direction.** Participants often found the output convincing at the level of short phrases, but harder to guide toward larger musical development. The issue was less simply model context length than the lack of interface mechanisms for steering harmonic movement or global register, especially in reduced button mappings.

6 DISCUSSION: SHARED AGENCY IN REAL-TIME PITCH CONDITIONING

6.1 Performer–model agency tension in compressed pitch conditioning

The compression-based prototypes expose a structural tension between performer and model agencies over pitch realization. As compression decreases, the resolution of the performer's pitch-conditioning signal increases: with more buttons, performers can transmit more detailed information about intended melodic shape, but the musical result becomes more dependent on their ability to use that vocabulary effectively. Conversely, stronger compression leaves more of the concrete pitch sequence to the model, potentially supporting more immediate musical plausibility while reducing intentional control.

From this design structure, one might expect a parallel tension between performer agency and pianistic coherence: if the performer constrains the model more specifically, the model may have less freedom to maintain idiomatic continuations. The participant study supports the perceived-controllability side of this reasoning, but not a simple version of the coherence side. The 1-button interface was experienced as almost entirely model-led; the 8-button interface was easy to make sound convincing but limited in fine control; and the 19-button interface as the clearest compressed condition for intentional control. However, responses do not support the claim that the 8-button interface was necessarily more coherent than the 19-button interface. Instead, performer guidance appears to have a strong influence on coherence even in compressed melodic-shape interfaces. Additional observations support this interpretation and are described in Appendix D. A related limitation is that the system more readily supports local coherence than longer-range musical direction: participants could often generate convincing short phrases, but found it harder to guide phrase development, register trajectory, or harmonic evolution.

6.2 Relative-pitch conditioning as explicit guidance

The relative-pitch interface reveals a different aspect of shared agency. Arrows do not simply provide more or less control than compressed button mappings; by encoding interval tendencies, they change the representation of control.

Participants often reported that the arrows interface gave a stronger sense of influence than the 8-button interface, despite using a similar number of controls. This can be explained by the fact that arrows constrain the model's possible paths in a more explicit way: a small-step arrow forces movement within a narrow interval class, while a large-leap arrow excludes all smaller intervals and pushes the model toward a different region of the learned interval distribution. In both cases, the performer's action has a more legible musical meaning than in a compressed button mapping.

At the same time, this representation required significant mental effort. Several participants found the arrows interesting and expressive, but harder to learn, especially when prior keyboard expectations interfered with the relative-control paradigm. Some participants seemed to approach the 8-button mapping as a small absolute-pitch keyboard, which made its indirect pitch behavior difficult to interpret. By contrast, the arrows made the guiding role of the performer more explicit: although cognitively demanding, it often produced a stronger sensation of direction or guidance than the 8-button interface. This suggests that changing the representation of control can increase perceived influence and directionality, but may also increase learning demands and the risk of over-constraining the model if the interval classes are used too rigidly. This suggests that future shared-control interfaces should therefore not only reduce the number of controls within an absolute-pitch paradigm, but also explore control representations designed specifically for guiding the model.

6.3 Hybrid strategies: making negotiation performable

The most promising insight from the prototype explorations and participant study is that the performer–model agency tension does not need to be fixed at design time. Interfaces can allow performers to move between different degrees and kinds of conditioning during performance.

The joker mechanism provides the clearest example of this performable negotiation. The performer can specify interval tendencies for some events and then use a joker to remove the current interval constraint. Participants repeatedly described jokers as useful for reset, rescue, and reorientation, especially in the arrows interface, where it was easy to lose track of the resulting pitch direction. We also observe that performers can exploit the autoregressive dependence on recent context: performers can impose contour constraints to seed a pattern, then insert joker arrows to let the model elaborate it idiomatically, leveraging the model's strong dependence on recent context. These observations suggest that joker inputs were not experienced simply as giving up control, but as a way to recover orientation or let the model continue from context in musically interesting ways.

More broadly, they point toward hybrid designs where the interface offers different control–coherence profiles, and where moving between them is itself a meaningful gesture. The participant study reinforces the relevance of this idea: several participants expressed particular interest in interfaces that do not simply maximize either control or autonomy, but allow alternating between direct steering and model-led continuation.

The design implication is therefore not simply to maximize agency or maximize model autonomy, but to design interfaces where (1) the trade-off is legible, (2) switching costs are low enough for real-time use, and (3) the resulting interaction supports a plausible learning trajectory—from "instant music" toward progressively finer control.

This design direction is illustrated by preliminary variants we implemented: a non-uniform 19-button mapping where central buttons provide finer pitch conditioning and extremes leave more model freedom; a register-sensitive mapping where repeated extreme-button use cues upward or downward phrase movement; and a compressed-button release mechanism where rapid repetition of a small control subset temporarily relaxes pitch constraints.

7 CONCLUSION AND FUTURE DIRECTIONS

7.1 Summary of key findings

This paper explored real-time conditioning strategies for shared agency in AI-mediated piano improvisation. The prototype explorations and participant study suggest three main findings.

First, compressed pitch conditioning creates a performer–model agency tension in which increasing pitch-control resolution gives performers a stronger sense of control, while also making the musical result more dependent on the performer's ability to provide useful conditioning gestures. However, this dependence is not limited to higher-resolution mappings. Even in compressed melodic-shape interfaces such as the 8-button condition, pianists appeared better able to guide the model toward coherent continuations. This suggests that performer guidance and dexterity may influence musical coherence more strongly than compression ratio alone.

Second, perceived control depends not only on the amount of control, but also on its representation. The relative-pitch interface suggests that performers may experience a stronger sense of direction when controls encode musical tendencies, such as interval motion, rather than compressed absolute-pitch categories.

Third, we found that the tension between performer and model agency is not merely a limitation but a design material: hybrid mechanisms can allow performers to negotiate control and autonomy as an expressive resource. The contribution of this paper is therefore not a definitive evaluation of a single interface, but a design-led account of how shared control can be made legible and performable in AI-mediated piano improvisation.

7.2 Design recommendations

From these explorations, we propose four practical recommendations for AI-mediated instruments targeting non-expert pianistic interaction:

- **Treat shared control as a negotiable parameter**. Interfaces should enable performers to dynamically move along the agency–coherence gradient during play, rather than locking the instrument into a single compromise.
- **Make agency shifts legible.** Systems should help performers understand when the model is taking more responsibility for producing idiomatic output, and when the performer is taking more responsibility through finer or stronger control.
- **Design interfaces for shared control as guidance.** Rather than tying interaction to fixed absolute pitches or simply reducing the number of piano-like controls, interfaces should function as ways of guiding the model. Controls may encode tendencies, directions, harmony, register movement, or other musically meaningful cues that allow performers to steer generation without specifying every pitch.
- **Design for a learning trajectory, with expertise differences in mind**. Novice-friendly mappings should provide "instant music," but also a clear path toward increased nuance, repeatability, and intentionality as performers practice. More compressed mappings may be especially valuable for non-musicians seeking immediate coherence, whereas higher-agency settings may better serve musicians able to exploit finer pitch control.

In the process of re-mediating pianistic expressiveness from technique-based mastery to interaction-based guidance, AI-mediated instruments should support shared control strategies that help performers guide the model, release control when useful, and progressively develop more intentional ways of shaping the musical result.

7.3 Future work

First, we will develop conditioning mechanisms for longer-range musical direction, targeting higher-level attributes such as harmonic movement, style, register, and phrase-level tendencies while preserving real-time playability.

Second, we will investigate hybrid conditioning strategies, including context-tuning methods that inject additional musical cues into the autoregressive stream, and cross-attention mechanisms that sustain stylistic or structural influence over longer spans. These approaches could complement the current local conditioning strategies by supporting more persistent and flexible forms of performer guidance.

Third, we will study how performer strategies develop over time. Longer-term studies comparing pianists, non-pianists, and non-musicians could show whether compressed and relative-control interfaces support a learning trajectory from immediate playability toward more intentional musical control.

8 ACKNOWLEDGMENTS

This work is supported by Cátedra IA y Música project (TSI-100929-2023-1), funded by the Secretaría de Estado de Digitalización e Inteligencia Artificial and European Union-Next Generation EU funds and by BMAT Music Innovators and IMPA project PID2023-152250OB-I00 funded by MCIU/AEI/10.13039/501100011033/FEDER, UE.

AUTHOR DECLARATIONS

To the best of the authors' knowledge, all code and data utilized in this work are used under appropriate rights and licenses and are not subject to copyright infringement. An AI-assisted language editing tool was employed to improve grammar and readability. This assistance did not modify the content, interpretation, or intent of the authors' ideas. AI tools were used to produce segments of code under the direction and validation of the authors, who are solely responsible for the ideas, their integration, and their operation. The icon images included in this work were generated using the Nano Banana AI image generation tool.

REFERENCES

Steve Benford, Marco Amerotti, Bob L. T. Sturm, and Juan Martinez Avila. 2024. Negotiating Autonomy and Trust when Performing with an AI Musician. In *Proceedings of the Second International Symposium on Trustworthy Autonomous Systems (TAS '24)*. 1–10. https://doi.org/10.1145/3686038.3686040

Steve Benford, Richard Ramchurn, Joe Marshall, Max L. Wilson, Matthew Pike, Sarah Martindale, Adrian Hazzard, Chris Greenhalgh, Maria Kallionpää, Paul Tennent, and Brendan Walker. 2021. Contesting control: journeys through surrender, self-awareness and looseness of control in embodied interaction. *Human–Computer Interaction* 36, 5-6 (2021), 361–389. https://doi.org/10.1080/07370024.2020.1754214

Louis Bradshaw, Alexander Spangher, Stella Biderman, and Simon Colton. 2025. The Ghost in the Keys: A

Disklavier Demo for Human-AI Musical Co-Creativity. arXiv preprint arXiv:2511.01663. https://doi.org/10.48550/arXiv.2511.01663

Virginia Braun and Victoria Clarke. 2006. Using Thematic Analysis in Psychology. *Qualitative Research in Psychology* 3, 2 (2006), 77–101. https://doi.org/10.1191/1478088706qp063oa

Andrew R. Brown. 2016. Understanding Musical Practices as Agency Networks. In *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016)*. Paris, France. https://www.computationalcreativity.net/iccc2016/wp-content/uploads/2016/01/Understanding-Musical-Practices-as-Agency-Networks.pdf

Joel Chadabe. 1984. Interactive Composing: An Overview. *Computer Music Journal* 8, 1 (1984), 22–28. https://www.jstor.org/stable/3679894

Palle Dahlstedt. 2021. Musicking with Algorithms: Thoughts on Artificial Intelligence, Creativity, and Agency. In *Handbook of Artificial Intelligence for Music*, Eduardo R. Miranda (Ed.). Springer, Cham, 873–914. https://doi.org/10.1007/978-3-030-72116-9_31

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness. arXiv preprint arXiv:2205.14135. https://doi.org/10.48550/arXiv.2205.14135

Chris Donahue, Ian Simon, and Sander Dieleman. 2019. Piano Genie. In *Proceedings of the 24th International Conference on Intelligent User Interfaces (IUI 2019)*. ACM, 160–164. https://doi.org/10.1145/3301275.3302288

Nicholas Evans, Behzad Haki, and Sergi Jordà. 2024. GrooveTransformer: A Generative Drum Sequencer Eurorack Module. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME 2024)*. 261–265. https://doi.org/10.5281/zenodo.13904848

Jennifer Fereday and Eimear Muir-Cochrane. 2006. Demonstrating Rigor Using Thematic Analysis: A Hybrid Approach of Inductive and Deductive Coding and Theme Development. *International Journal of Qualitative Methods* 5, 1 (2006), 80–92. https://doi.org/10.1177/160940690600500107

Andrea Gaggioli, Sabrina Bartolotta, Andrea Ubaldi, Katusha Gerardini, Eleonora Diletta Sarcinella, and Alice Chirico. 2025. Extended Creativity: A Conceptual Framework for Understanding Human-AI Creative Relations. arXiv preprint arXiv:2506.10249. https://doi.org/10.48550/arXiv.2506.10249

Leon Gruenbaum. 2007. The Samchillian Tip Tip Tip Cheeepeeeee: A Relativistic Keyboard Instrument. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME '07)*. ACM, 256–259. https://doi.org/10.1145/1279740.1279793

Michael Gurevich, Adnan Marquez-Borbon, and Paul Stapleton. 2010. Style and Constraint in Electronic Musical Instruments. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME 2010)*. Sydney, Australia, 106–111. https://www.nime.org/proceedings/2010/nime2010_106.pdf

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. 2018. Music Transformer. arXiv preprint arXiv:1809.04281. https://doi.org/10.48550/arXiv.1809.04281

Sergi Jordà. 2005. *Digital Lutherie: Crafting musical computers for new musics' performance and improvisation*. Ph. D. Dissertation. Universitat Pompeu Fabra. https://mtg.upf.edu/files/publications/PhD2005-sjorda.pdf

Qiuqiang Kong, Bochen Li, Jitong Chen, and Yuxuan Wang. 2020. GiantMIDI-Piano: A large-scale MIDI dataset for classical piano music. arXiv preprint arXiv:2010.07061. https://doi.org/10.48550/arXiv.2010.07061

Haoyang Li, Yiming Li, Anxin Tian, Tianhao Tang, Zhanchao Xu, Xuejia Chen, Nicole Hu, Wei Dong, Qing Li, and Lei Chen. 2024. A Survey on Large Language Model Acceleration based on KV Cache Management. arXiv preprint arXiv:2412.19442. https://doi.org/10.48550/arXiv.2412.19442

Stephen Malinowski. 2007. The Conductor Program – computer-mediated musical performance. Music Animation Machine (web page). https://www.musanim.com/Tapper/ConductorProgram.html

M. V. Mathews. 1991. The Radio Baton and the Conductor Program, or: Pitch, the Most Important and Least Expressive Part of Music. *Computer Music Journal* 15, 4 (1991), 37–46.

François Pachet. 2003. The Continuator: Musical Interaction with Style. *Journal of New Music Research* 32, 3 (2003), 333–341. https://doi.org/10.1076/jnmr.32.3.333.16861

Christine Payne. 2019. MuseNet. OpenAI blog post. https://openai.com/index/musenet/

Victor Shepardson, Jack Armitage, and Thor Magnusson. 2022. Notochord: A Flexible Probabilistic Model for Embodied MIDI Performance. In *Proceedings of the 3rd Conference on AI Music Creativity (AIMC)*. https://iil.is/pdf/2022_aimc_shepardson_et_al_notochord.pdf

Laurie Spiegel. 1987. A Short History of Intelligent Instruments. *Computer Music Journal* 11, 3 (1987), 7–9. https://www.jstor.org/stable/3679729

John Thickstun, David Hall, Chris Donahue, and Percy Liang. 2023. Anticipatory Music Transformer. arXiv preprint arXiv:2306.08620. https://doi.org/10.48550/arXiv.2306.08620

Dimitri von Rütte, Luca Biggio, Yannic Kilcher, and Thomas Hofmann. 2023. FIGARO: Controllable Music Generation using Learned and Expert Features. In

International Conference on Learning Representations (ICLR). https://openreview.net/forum?id=NyR8OZFHw6i

Yusong Wu, Tim Cooijmans, Kyle Kastner, Adam Roberts, Ian Simon, Alexander Scarlatos, Chris Donahue, Cassie Tarakajian, Shayegan Omidshafiei, Aaron Courville, Pablo Samuel Castro, Natasha Jaques, and Cheng-Zhi Anna Huang. 2024. Adaptive Accompaniment with ReaLchords. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*. https://dl.acm.org/doi/10.5555/3692070.3694255

A MODEL ARCHITECTURE AND RUNTIME CONFIGURATION

Models were trained on the GiantMIDI-Piano dataset (Kong et al., 2020), which contains 10,779 classical piano performances corresponding to approximately 1,200 hours of music, split into training, validation, and test partitions using an 80/10/10% ratio. For the arrow-based model we use monophonic melodic lines, whereas for the button-based models we use the full polyphonic note stream. Sequences were segmented into non-overlapping windows of 1024 tokens and randomly transposed within a range of 6 to +6 semitones for data augmentation. Pitch events are encoded as a vocabulary of 88 pitch tokens (MIDI notes 21–108), plus one special token (PAD). Because timing, velocity, and duration are not used for model inference (Section 3.2), no time-shift or velocity tokens are included; the model operates on the pitch sequence only. In both architectures, conditioning tokens form a separate vocabulary. All models share a Transformer backbone with 4 layers, 32 attention heads, an embedding dimension of 2048, and a maximum sequence length of 1024 tokens. Rotary positional encodings are used, and the system is optimized for low-latency real-time inference with KV caching (Li et al., 2024) and FlashAttention (Dao et al., 2022).

In the button-based models, the compressed control vocabularies were learned with a variational autoencoder that encodes pitch events into 1, 8, 19, or 88 discrete categories, using the same Transformer configuration for the encoder. The encoder maps each pitch token to a scalar latent value, and the conditioning stream is the resulting sequence of discrete button indices, which are embedded and fused with the pitch-token embeddings. At inference time, the performer's button stream replaces the encoder output as the conditioning signal.

In the arrow-based model, the mapping from interval class to arrow token is deterministic and therefore requires no learned encoder. Arrow tokens are embedded as discrete symbols and added to the pitch-token embeddings. During training, arrow tokens are computed from ground-truth consecutive pitch intervals and provided as an aligned conditioning stream; 25% of the arrow tokens are randomly replaced with joker tokens so that the model learns they carry no interval information.

All models were trained on a single NVIDIA GeForce RTX 4090 GPU using the Adam optimizer with a learning rate of 1e-4, a batch size of 20, and dropout 0.1. The training objective is next-pitch-token cross-entropy, complemented by a term that encourages the encoder to produce button contours matching the shape of the input melodic contours; for the arrow model, auxiliary losses encourage generated pitches to fall within the specified interval class (Section 4.2).

Runtime latency was measured under different hardware and context-length settings. Inference on an NVIDIA GeForce RTX 4090 took approximately 3 ms per token with a 300-token context and 6 ms with a 512-token context, low enough for real-time performance. On a MacBook Pro M1, inference took approximately 47 ms with a 128-token context, which was sufficient for basic testing. Sampling uses a temperature of 1.

B EXPRESSIVE REACH OF RELATIVE VS. ABSOLUTE PITCH MAPPINGS

When the available pitch space greatly exceeds the number of physical keys ($M \gg N$), an absolute N-key mapping can only ever trigger at most N fixed pitches, regardless of how long the performer plays. By contrast, a relative (interval-based) mapping uses the same N keys to apply pitch transitions to a running pitch state; as long as the interval set allows movement in both directions and is sufficiently connected (e.g., includes a 1-semitone step), repeated transitions can access essentially the full M-pitch space, making the control vocabulary strictly more expressive over time.

C REPRESENTATIVE PARTICIPANT COMMENTS

This appendix provides representative comments supporting the observations summarized in Section 5.2. Comments were selected to illustrate recurring themes rather than to provide an exhaustive transcript.

1-button interface: model-led but coherent. Participants described the 1-button interface as offering almost no pitch control, while still often producing coherent musical results. One participant said, "It sounds perfect, but I have zero influence over the notes I play." Another described it as "zero feeling; I only control the tempo." A third noted that "the possibilities of feeling that you are really doing something run out very quickly," because the performer's role is "extremely limited."

19-button interface: strongest compressed-control condition. The 19-button interface was repeatedly associated with a clearer sense of agency and intentional shaping. One participant said, "With this one I can shape better where I want it to go," while another reported, "Suddenly I felt I could draw melodic lines." A pianist described it as "a very good balance" between letting the AI generate harmonies and retaining performer control. Another participant attributed this to the larger control vocabulary: "Since you have more buttons, the correlation between the physical space on the interface and the pitch of the sound is much clearer."

8-button interface: low effort but limited precision. The 8-button interface was often described as easy to make musically convincing, but frustrating when participants wanted more specific control. One participant said, "I feel like I have no control over the harmony or the melody" while another said, "There isn't enough room to play the melody and accompaniment with both hands." Another described a "very little sense of making mistakes," because "whatever you play, it's easy to make it sound more or less in place."

Coherence and performer guidance. Several comments suggest that coherence depended not only on compression level, but also on the performer's ability to discover effective gestures. A pianist said, "The music sounded coherent, but that's because I understood the structure and provided the appropriate gestures." Another participant noted, "If I learn the gestures, I'll be able to direct the thing."

Accepting a guiding role. Some participants were confused when they approached the interfaces as conventional keyboards. One said, "I can't help thinking in piano keys; it feels strange when you press a key and it isn't the note you expect." Others seemed more comfortable when they treated the system as something to guide rather than fully control. One participant described the interaction as "like giving instructions to someone who plays for you," while another said, "I felt that it was correcting me, instead of me correcting it." Another appreciated that the system could "make interesting decisions for me."

Arrows interface: expressive but cognitively demanding. The arrows interface was often described as more explicitly directional than the 8-button interface, but also harder to learn. One participant said, "This one is more interesting. Harder to understand, but easier to get an interesting result with." A pianist called it "a puzzle for a pianist's brain," while another participant described it as "like having to learn to read again, but after a while, I feel like I can communicate more directly." Another participant valued the departure from keyboard expectations, saying, "You'll be able to provide better guidance, since you're not constrained by the traditional keyboard paradigm."

Joker controls: reset, rescue, and reorientation. Joker controls were repeatedly described as useful when participants lost orientation or reached a dead end. One participant said, "You press the button and it saves you. It resolves things by itself and gets the music back on track." Others described the joker as "a point of reference" and "an emergency exit," helping them "recover the feeling of knowing where you are." Another participant explained: "It's good for when you run out of ideas or realize you've hit a dead end. You tap there and it gives you a path again."

Local coherence without long-range direction. Participants often found the system convincing locally, but harder to guide over longer spans. One participant said, "I am able to create sentences that are coherent but do not reach a final argument." Others said the music "doesn't really end up going anywhere" or lacks "a larger musical argument." Similar issues appeared for register control: "Locally yes you can guide the note up and down, but globally no."

D SUPPLEMENTARY OBSERVATIONS ON COHERENCE

The following supplementary observations clarify why coherence should not be interpreted as a direct function of compression level alone.

From the structure of the compression mappings, one might expect differences in musical coherence between expert and non-expert performers to become stronger in less compressed interfaces, where the performer has more influence, and weaker in highly compressed interfaces, where the model carries more of the musical generation. This expectation is clearest in the 1-button condition: because participants have virtually no influence over pitch, the model almost entirely determines the musical output.

However, the responses do not support a simple claim that the 8-button interface necessarily produces more musical coherence than the 19-button interface. Pianists appeared better able to guide the model toward musically plausible continuations, despite not directly controlling exact pitches: skilled performers seem better able to supply gestures that keep the model within musically plausible paths. For non-musicians using the 8-button interface, the higher degree of model agency made it easier to find locally coherent results, but wrong or unclear guidance could still quickly lead to less coherent outcomes.

This interpretation was reinforced by an additional oracle conditioning test. We seeded the model with a short excerpt of a professional piano performance, extracted the corresponding button sequence from the subsequent notes using the trained encoder, and used this button stream to drive generation. Under this condition, the system produced highly convincing pitch continuations. This suggests that reduced musical quality in some real-time interactions should not be attributed only to compression or to a loss of model competence, but also to the difficulty of supplying effective conditioning gestures during live performance.