IRIDyOM: Exposing Musical Expectation as an Interactive Creative Space

Lluc Bono Rosselló, Julien Baudru and **Hugues Bersini**
Institute for Interdisciplinary Studies on Artificial Intelligence
Université Libre de Bruxelles
Bruxelles, Belgium
Lluc.Bono.Rossello@ulb.be

Abstract

Computational models of musical expectation such as IDyOM estimate listener-relative surprise and uncertainty, providing a formal basis for analyzing how musical events are predicted and experienced. This paper presents IRIDyOM, a real-time Max for Live interface that brings IDyOM-style estimates into Ableton Live through a Python backend and a no-code front end. Rather than using expectation only as a retrospective descriptor, the system makes it available as an interactive creative space: users can configure a stylistic prior from a corpus, inspect predictive profiles, apply guided transformations, and generate constrained continuations. By exposing predictive structure directly within a DAW workflow, IRIDyOM supports exploratory composition, interpretable expectation-guided interaction, and research workflows requiring direct manipulation of musical predictability.

Introduction

Musical listening and composition are shaped by expectations about what may happen next. Through exposure to musical styles, listeners acquire statistical regularities that influence perception, aesthetic response, and pleasure (Huron 2006; Gold et al. 2019; Mas-Herrero and Marco-Pallarés 2025). Computational models of musical expectation make these processes explicit by estimating probability distributions over possible continuations. Among these, IDyOM (Information Dynamics of Music) is a widely used model that derives information-theoretic measures such as entropy, quantifying uncertainty about future events, and Information Content, quantifying the surprise of an observed event (Pearce and Wiggins 2012; Pearce 2018). These predictors have become important in music-cognition research, where they have been used to relate predictive structure to behavioural and neural responses during music listening (Sauvé and Pearce 2019; Cheung et al. 2019; Di Liberto et al. 2020; Daikoku 2018).

This expectation-based perspective is also relevant for creativity. In creativity, novelty and value are not absolute properties of an artifact, but depend on the conceptual and evaluative frame within which that artifact is encountered (Boden 2004; Wiggins 2006). In music, that frame is shaped in part by learned expectations: what counts as familiar, complex or surprising depends on the listener's prior exposure to style and on the predictive context established by the unfolding piece. Expectation models therefore matter not only because they describe what a listener may predict next, but because they provide formal proxies for one dimension of how musical ideas are evaluated. This is especially relevant in light of work linking predictive uncertainty and surprise to perceived complexity and musical pleasure (Sauvé and Pearce 2019; Gold et al. 2019; Cheung et al. 2019; Mas-Herrero and Marco-Pallarés 2025). Recent work has likewise argued that musical creativity depends in part on statistical learning, hierarchical organization, and the regulation of uncertainty, rather than on unpredictability alone (Daikoku 2018; Daikoku, Wiggins, and Nagai 2021). Expectation models are therefore relevant not only as descriptive tools, but also as formalizations of one dimension of creative evaluation and control.

Despite this relevance, expectation models are still used mainly as offline analytical tools. They can describe how surprising a melody is after it has been composed, but they are rarely available as interactive resources during composition. This also limits their use in experimental design, where predictive variables are often measured after stimulus generation rather than manipulated during it (Mas-Herrero and Marco-Pallarés 2025). A tool that makes predictive distributions and information-content profiles inspectable and controllable in real time could therefore support both compositional practice and behavioural or neuroscience studies requiring controlled manipulation of musical expectation.

Existing AI music tools increasingly support interactive generation, from digital audio workstation (DAW)-oriented systems such as Magenta Studio to corpus-based improvisation systems such as Somax (Roberts et al. 2019; Borg 2019). These systems demonstrate the creative potential of machine learning in music, but they typically focus on producing or recombining musical material rather than exposing how that material may be experienced relative to a learned expectation model. By contrast, IDyOM provides interpretable listener-relative signal, but has historically been less accessible for real-time, no-code interaction within a DAW. Recent Python implementations have improved access to IDyOM-style analysis (Guan, Ren, and Pelofi 2022; Marion et al. 2025), yet expectation modelling remains used primarily in offline analytical workflows rather than as an interactive resource within compositional workflows.

This paper presents *IRIDyOM* (Interface for Real-time

IDyOM), a Max for Live device—that is, a custom interactive module running inside the Ableton Live digital audio workstation (DAW)—that integrates a new Python reimplementation of IDyOM into a real-time compositional workflow. The backend is designed for low-latency inference while preserving the expectation-modelling logic of the original Lisp implementation. Rather than presenting another autonomous music generator, *IRIDyOM* makes predictive musical expectation directly available as an interactive compositional resource. Users can define a stylistic prior, inspect next-event probabilities and information-content profiles, and guide continuation, variation, and generation through explicit control of expectedness. The contribution is therefore an interface framework that brings listener-relative predictive structure into a DAW-native workflow for composition, co-creative interaction, and experimental stimulus design.

System Overview

Information Dynamics of Music

IDyOM (Information Dynamics of Music) is a computational model of musical expectation that predicts the next event in a musical sequence from statistical regularities acquired through exposure (Pearce and Wiggins 2012; Pearce 2018). Following the multiple-viewpoint approach to music modelling, musical events can be represented through alternative feature descriptions such as pitch class, pitch with octave, interval, duration, or linked viewpoint combinations (Conklin and Witten 1995). In the current implementation, predictions target a pitch-related representation, expressed over pitch–octave events.

Given a musical history $x_{1:t-1}$, IDyOM estimates a probability distribution $P(x_t \mid x_{1:t-1})$ over possible next events in the chosen target viewpoint. This distribution combines evidence from two complementary sources. The long-term model (LTM) represents prior exposure to a corpus, style, repertoire, or user collection, and is estimated by accumulating context-dependent transition statistics from that material. The short-term model (STM) is updated online from the current piece or interaction, capturing regularities that emerge locally as the melody unfolds. Within each model, predictions are made using variable-order contexts: longer histories provide more specific predictions, while shorter histories provide more robust estimates when evidence for the longer context is sparse. In standard IDyOM, this variable-order combination is handled using Prediction by Partial Matching (PPM) back-off (Cleary and Witten 1984); in the present implementation, order-specific predictions may also be merged using entropy-weighted merging, so that lower-entropy, more confident distributions receive greater weight. The same principle is used to combine LTM and STM predictions, allowing the final distribution to reflect both corpus-level regularities and the immediate melodic context.

From this distribution, IDyOM derives Information Content, which measures how surprising the observed event is under the current expectation model:

$$IC(x_t) = -\log_2 P(x_t \mid x_{1:t-1}).$$

High-probability events therefore have low Information Content and are interpreted as expected, while low-probability events have high Information Content and are interpreted as surprising. Importantly, surprise is relative to the selected viewpoint and to the model's prior exposure: the same note may be expected or unexpected depending on the corpus-level statistics and the current melodic context. This makes IDyOM well suited for an interactive interface in which an LTM estimated from a loaded or user-provided corpus defines a listener-like prior, the STM adapts to the ongoing musical interaction, and the resulting probabilities and information-content values can be analyzed, steered, or used for generation.

Architecture

IRIDyOM is designed to make IDyOM-based expectation modelling usable within creative and research workflows. Following recommendations for deployable music-AI tools, the aim is not only to expose model behaviour, but to do so through an interface that is accessible, flexible, and compatible with established compositional environments (Roberts et al. 2019; Kayacik et al. 2019; Pasquier et al. 2017). As in Magenta Studio's integration of generative models with a DAW (Roberts et al. 2019), *IRIDyOM* separates predictive inference, communication, and user interaction (Figure 1).

The predictive layer uses a Python reimplementation of IDyOM based on an explicit graph formulation of variable-order Markov models. It loads long-term models estimated from user-provided corpora, maintains short-term context from the current musical sequence, and returns next-event probability distributions used for Information Content calculation. A local Python server wraps this functionality and acts as the interaction controller. It manages the active model, prediction sessions, short-term melodic history, file-based analysis requests, and the exchange of control and result messages between the modelling backend and the DAW environment. The user-facing layer is a Max for Live device running inside Ableton Live. Because the Python model is kept outside the Max device, communication is handled by a Node for Max bridge using the `node.script` object and `max-api`. The bridge exchanges newline-delimited JSON messages with the Python server over a local TCP connection. This design keeps the system synchronized with the DAW while allowing the modelling code, server logic, and Max interface to be developed and deployed independently. The software can be run directly from the repository or installed through a no-code Windows installer.

Interaction Modes

The interface is organized around five interaction modes, each corresponding to a different way of working with the same expectation model rather than to an independent function. *Train* defines the stylistic prior by estimating the long-term model from user-provided material. *Assistant* exposes the model's predictive structure during composition by showing both the Information Content profile of the current melody, that is, its evolving surprise pattern, and the probabilities of candidate next notes, which function as

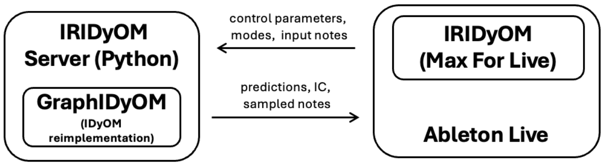

Figure 1: Runtime architecture of *IRIDyOM*. A Max for Live device inside Ableton Live communicates with a local Python server through a Node for Max bridge. The server manages interaction state and calls GraphIDYOM, a Python reimplementation of IDyOM, to return probability distributions, Information Content values, and generated MIDI events.

ranked suggestions for likely continuations. *Continue* and *Variate* allow users to manipulate musical material under explicit control of expectedness, either by extending an existing sequence or by rewriting a selected region. *Sequencer* supports more continuous constrained generation from the same learned model. Across these modes, *IRIDyOM* treats predictive expectation not only as an analytical description of music, but also as a compositional parameter that can be inspected and shaped in real time.

Train Mode

Training operates at the meta-level of the system. *Train* mode lets the user define the stylistic prior by selecting MIDI files and naming the resulting model (see Figure 2). The resulting model determines what the system considers probable or stylistically expected. If the corpus consists of the user's previous compositions, the system approximates that user's own stylistic tendencies; if it is drawn from an external repertoire, it instead reflects a different stylistic frame of reference. *Training* therefore determines whose expectations are being modelled, and against which prior later creative decisions are evaluated in terms of predictability and surprise.

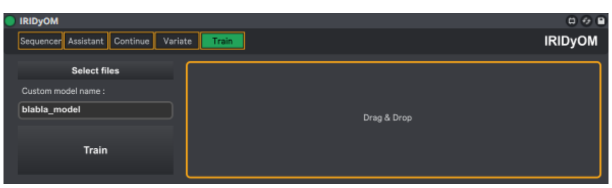

Figure 2: *Train* Mode interface, showing corpus selection and model naming in the left panel and the drag-and-drop MIDI area in the right.

Assistant Mode

Assistant mode exposes the expectation model as an analytical listener during composition. The user can play notes live, analyse a melody within the session, or import a MIDI file; in each case, the system evaluates the unfolding sequence against the long-term loaded model and the local short-term context (See Figure 3). Returns information content values for the notes of the sequence, displays likely next-note candidates, and visualizes the resulting IC profile, which can also be exported for later inspection (see Fig. 4). *Assistant* mode therefore lets the user inspect how a melody is positioned relative to the learned style, showing which events are more or less expected.

Figure 3: *Assistant* Mode interface, with live note input and next-note inspection on the top of keyboard display, a drag and drop area in the middle and a real-time information-content trace shown at right.

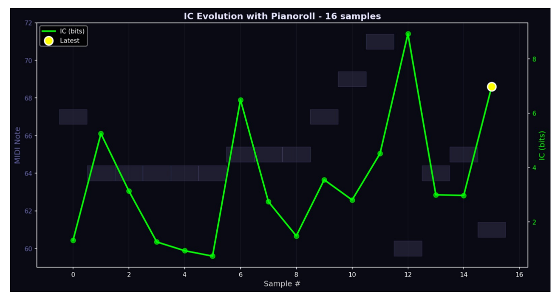

Figure 4: **Export Example.** Exported analysis view showing the MIDI piano roll overlaid with the corresponding information-content (IC) curve.

Continue Mode

Continue mode turns the expectation model into a co-creative partner for extending existing material. After analyzing a MIDI file, the user selects an output location and can specify a sequence of expectedness choices for the generated continuation (See Figure 5). Each step can favour the most probable candidate, an intermediate-probability candidate, or a low-probability candidate. The resulting MIDI file appends new notes to the source material, using the analyzed melody as context while allowing the user to shape the expectedness profile of the continuation without choosing exact pitches.

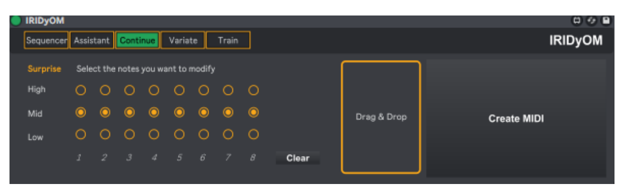

Figure 5: *Continue* Mode interface, organized around a discrete surprise-pattern control panel on the left and a drag-and-drop MIDI input/output workflow.

Variate Mode

Variate mode applies the same expectation logic to local transformation rather than extension. The user selects an analyzed MIDI file, chooses one of four regions of the note sequence, and sets a surprise value (See Figure 6). The system preserves the original timing and durations but replaces pitches in the selected region according to the current prediction distribution. Low surprise favours probable notes, medium surprise favours intermediate candidates, and high surprise favours lower-probability alternatives. The result is exported as a new MIDI file, allowing the user to explore nearby versions of a phrase while controlling how strongly it conforms to or departs from learned expectations.

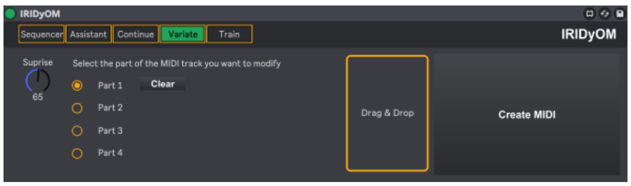

Figure 6: *Variate* Mode interface, showing region-selection controls and a surprise-level knob for local pitch rewriting of an analysed MIDI sequence.

Sequencer Mode

Sequencer mode is the most autonomous use of the expectation model, but it remains constrained by user-defined predictive and temporal parameters. The system continuously generates MIDI notes by sampling from the next-event probability distribution conditioned on the current history. The user controls this stream through parameters including model selection, temperature, probabilistic sampling, pitch range, note duration, inter-onset interval, and context length (See Figure 7). This supports real-time brainstorming or performance-oriented generation, where material can be recorded, inspected, and further edited inside the DAW.

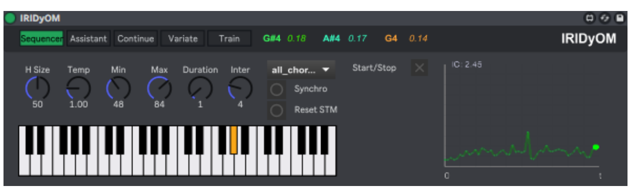

Figure 7: *Sequencer* interface, exposing real-time sampling controls for model selection, sampling temperature, temporal behaviour, and pitch-range constraints. Real-time surprise is plotted on the right panel.

Discussion and Conclusion

IRIDyOM proposes a different role for expectation models in creative practice. Rather than treating surprise and uncertainty only as post-hoc analytical descriptors, the system makes them directly usable during composition. By integrating IDyOM-derived probabilities, Information Content, and constrained generation into a DAW-native workflow, *IRIDyOM* supports analysis, controlled transformation, and exploratory continuation while leaving musical evaluation with the user. In this sense, the contribution is not a new autonomous generator, but an interpretable co-creative interface in which listener-relative predictive structure becomes a compositional resource.

More broadly, this shifts expectation from a descriptive model of listening to a usable dimension of creative evaluation. What counts as familiar, surprising, or meaningfully novel depends on the learned expectations brought to the musical material. Exposing such expectations interactively therefore opens a way to work not only with musical artifacts, but with the social evaluative frames through which those artifacts are explored and shaped. While the present demo validates the real-time integration and model-faithfulness of IRIDyOM, assessing its compositional value and the effects of different LTM/STM configurations requires future case studies and user feedback.

This perspective also suggests broader directions for future work at multiple levels of the framework. At the backend level, the current implementation remains limited to melodic expectation and a restricted target representation, but broader viewpoint support from the original IDyOM framework could be added, including rhythmic dimensions and richer interactions between pitch- and duration-based prediction. At the interface level, future work could expose these additional predictive variables through more flexible controls and visualizations while preserving the accessibility of the current workflow. At the interaction level, because the server exposes expectation as a real-time resource rather than only as an analytical output, the same predictive machinery could support richer studies of compositional decision-making, co-creative interaction, and experimental design beyond the present Max for Live interface. In particular, the current single-user setting could be extended toward more explicitly social and distributed forms of creativity (Saunders and Bown 2015; Linkola and Kantosalo 2019), including collective systems in which musical structure emerges through the interaction of multiple autonomous agents (Bono Rosselló et al. 2024). This would make it possible to study creativity not only as the exploration of a single predictive space, but as the interaction between multiple listener-relative spaces with different settings.

Code and demonstration. *IRIDyOM* is released as an open-source project, including the Max for Live device, Python backend, installation instructions, and example materials.[1] A short video demonstration documents the interface workflow, showing the main interaction modes and the real-time integration with Ableton Live.[2]

Acknowledgments

L. B. R acknowledges support from the ARIAC project (No. 2010235), funded by the Service Public de Wallonie (SPW Recherche).

[1] Project repository.
[2] Video demonstration.

References

[Boden 2004] Boden, M. A. 2004. *The creative mind: Myths and mechanisms*. Routledge.

[Bono Rosselló et al. 2024] Bono Rosselló, L.; Alkilabi, M.; Tuci, E.; Bersini, H.; and Reina, A. 2024. Emergent orchestras: A modular framework for musical robot swarms. In *Proceedings of the 2024 Conference on Artificial Life*. MIT Press.

[Borg 2019] Borg, J. 2019. Somax 2: A real-time framework for human-machine improvisation. *STMS IRCAM & Aalborg University, Internal Student Report*.

[Cheung et al. 2019] Cheung, V. K. M.; Harrison, P. M. C.; Meyer, L.; Pearce, M. T.; Haynes, J.-D.; and Koelsch, S. 2019. Uncertainty and surprise jointly predict musical pleasure and neural responses. *Current Biology* 29(23):4084–4092.

[Cleary and Witten 1984] Cleary, J. G., and Witten, I. H. 1984. Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications* 32(4):396–402.

[Conklin and Witten 1995] Conklin, D., and Witten, I. H. 1995. Multiple viewpoint systems for music prediction. *Journal of New Music Research* 24(1):51–73.

[Daikoku, Wiggins, and Nagai 2021] Daikoku, T.; Wiggins, G. A.; and Nagai, Y. 2021. Statistical properties of musical creativity: Roles of hierarchy and uncertainty in statistical learning. *Frontiers in Neuroscience* 15:640412.

[Daikoku 2018] Daikoku, T. 2018. Entropy, uncertainty, and the depth of implicit knowledge on musical creativity: Computational study of improvisation in melody and rhythm. *Frontiers in Computational Neuroscience* 12:97.

[Di Liberto et al. 2020] Di Liberto, G. M.; Pelofi, C.; Bianco, R.; Patel, P.; Mehta, A. D.; Herrero, J. L.; de Cheveigné, A.; Shamma, S.; and Mesgarani, N. 2020. Cortical encoding of melodic expectations in human temporal cortex. *eLife* 9:e51784.

[Gold et al. 2019] Gold, B. P.; Pearce, M. T.; Mas-Herrero, E.; Dagher, A.; and Zatorre, R. J. 2019. Predictability and uncertainty in the pleasure of music: A reward for learning? *Journal of Neuroscience* 39:9397–9409.

[Guan, Ren, and Pelofi 2022] Guan, X.; Ren, Z.; and Pelofi, C. 2022. py2lispidyom: A python package for the information dynamics of music (idyom) model. *Journal of Open Source Software* 7(79):4738.

[Huron 2006] Huron, D. 2006. *Sweet Anticipation: Music and the Psychology of Expectation*. The MIT Press. CogNet special collection.

[Kayacik et al. 2019] Kayacik, C.; Chen, S.; Noerly, S.; Holbrook, J.; Roberts, A.; and Eck, D. 2019. Identifying the intersections: User experience+ research scientist collaboration in a generative machine learning interface. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, 1–8.

[Linkola and Kantosalo 2019] Linkola, S., and Kantosalo, A. 2019. Extending the creative systems framework for the analysis of creative agent societies. In *Proceedings of the 10th International Conference on Computational Creativity*, 204–211. Association for Computational Creativity.

[Marion et al. 2025] Marion, G.; Gao, F.; Gold, B. P.; Di Liberto, G. M.; and Shamma, S. 2025. IDyOMpy: A new Python-based model for the statistical analysis of musical expectations. *Journal of Neuroscience Methods* 415:110347.

[Mas-Herrero and Marco-Pallarés 2025] Mas-Herrero, E., and Marco-Pallarés, J. 2025. Predictive processes shape individual musical preferences. *Proceedings of the National Academy of Sciences* 122(29):e2500494122.

[Pasquier et al. 2017] Pasquier, P.; Eigenfeldt, A.; Bown, O.; and Dubnov, S. 2017. An introduction to musical metacreation. *Computers in Entertainment (CIE)* 14(2):1–14.

[Pearce and Wiggins 2012] Pearce, M. T., and Wiggins, G. A. 2012. Auditory expectation: The information dynamics of music perception and cognition. *Topics in Cognitive Science* 4(4):625–652.

[Pearce 2018] Pearce, M. 2018. Idyom: A computational model of musical expectation. *Journal of New Music Research*.

[Roberts et al. 2019] Roberts, A.; Engel, J.; Mann, Y.; Gillick, J.; Kayacik, C.; Nørly, S.; Dinculescu, M.; Radebaugh, C.; Hawthorne, C.; and Eck, D. 2019. Magenta studio: Augmenting creativity with deep learning in ableton live. In *Proceedings of the international workshop on musical metacreation (mume)*, 11.

[Saunders and Bown 2015] Saunders, R., and Bown, O. 2015. Computational social creativity. *Artificial Life* 21(3):366–378.

[Sauvé and Pearce 2019] Sauvé, S. A., and Pearce, M. T. 2019. Information-theoretic Modeling of Perceived Musical Complexity. *Music Perception* 37(2):165–178.

[Wiggins 2006] Wiggins, G. A. 2006. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems* 19(7):449–458.

Appendix

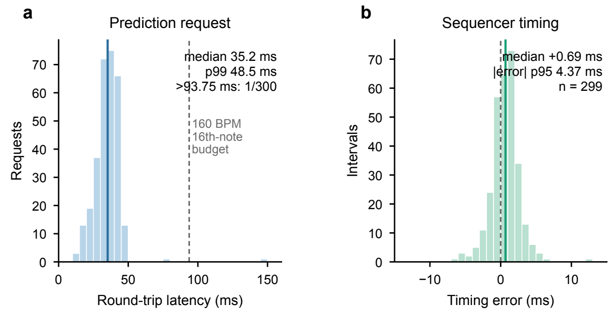

Figure 8: Server latency benchmark at 160 BPM sixteenth-note rate (93.75 ms inter-onset interval). (a) Round-trip latency for paced note-prediction requests over the local JSON/TCP interface with the interactive musical delay minimized. (b) Timing error of autonomous sequencer note-on messages relative to the 93.75 ms target interval. The server returned predictions within the musical time budget for 299/300 requests (median 35.2 ms; p99 48.5 ms), while autonomous sequencer output remained close to the target grid (median timing error +0.69 ms; 95th percentile absolute error 4.37 ms).

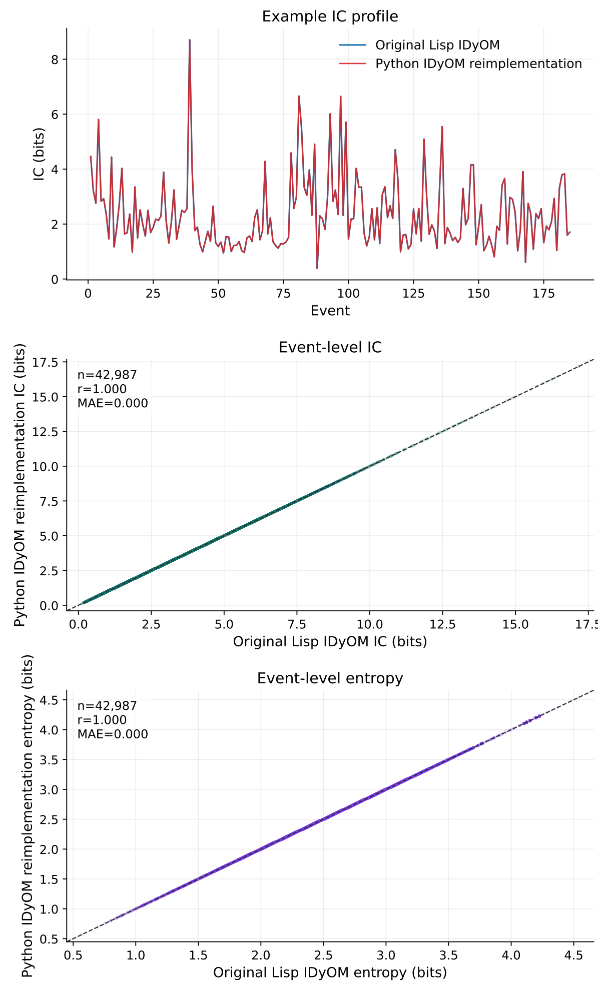

Figure 9: **Faithfulness backend check against the original IDyOM implementation.** Event-level information content and entropy from the Python IDyOM reimplementation are compared with the original Lisp IDyOM on a 900 files pitch-only cross-validation benchmark, using matched fold splits and memory-store settings. The near-identity IC profile and scatter plots show numerical agreement between implementations under these settings.