AI Framework for Dynamic Robotic Instrument Calibration

Colton Arnold
California Institute of the Arts
coltonarnold@students.calarts.edu

Zhaohan Cheng
California Institute of the Arts
zcheng@calarts.edu

Ajay Kapur
California Institute of the Arts
NYU Shanghai
akapur@calarts.edu

ABSTRACT

This paper presents a data-driven calibration framework for robotic musical instruments based on a hybrid ensemble model that combines K-nearest neighbors (KNN) and a multi-layer perceptron (MLP). KNN anchors predictions to recorded acoustic measurements, while the MLP enables nonlinear generalization and smooth interpolation across the instrument's playable range. A distance-dependent blending strategy integrates the two models, improving consistency across sparse and dense data.

The proposed approach produces stable and repeatable calibration estimates for both pitched and non-pitched instruments, outperforming standalone models across a range of sampling conditions. This work establishes a scalable foundation for automated calibration in robotic musical systems.

1. INTRODUCTION

The California Institute of the Arts (CalArts) maintains a collection of robotic instruments that require regular calibration to preserve consistent timbre, dynamic response, and overall performance quality. Over time, mechanical drift and component wear introduce nonlinear changes that alter each instrument's acoustic behavior, a well-documented issue in mechatronic systems. As these changes accumulate, the instruments no longer respond as originally intended by composers or performers.

To address this issue, we introduce a machine learning calibration framework that combines a K-nearest neighbors (KNN) with a multi-layer perceptron (MLP) to predict calibration adjustments across the instrument's playable range. By combining these complementary strengths, the system produces consistent predictions across different data densities. The remainder of this paper is organized as follows: Section 2 reviews related work on mechanical drift and data-driven calibration; Section 3 describes the proposed system and modeling framework; Section 4 presents the evaluation; and Section 5 concludes with discussion and future directions.

2. BACKGROUND AND RELATED WORK

2.1 CalArts Machine Lab

The CalArts Machine Lab [1] hosts a collection of custom-designed robotic musical instruments [2] developed over the past fifteen years, including BreakBot, GanaPati, Lydia, MahaDeviBot [3, 4], MalletOTon [5], RattleTron, and Tammy. Most recently, an 8x8 grid of Modulets [5] has been added to the lab. As shown in Figure 1, these instruments vary significantly in construction, actuation, and acoustic behavior, requiring individualized calibration procedures to maintain consistent performance.

Figure 1. From bottom left clockwise: BreakBot, MalletOTon, Lydia, and GanaPati.

2.2 Mechanical Drift

Mechanical drift is a persistent challenge in robotic musical instruments. Over time, actuators loosen, mallets shift, and hardware components wear, leading to changes in timbre and dynamic output. As a result, an instrument's response can diverge from its calibrated state, introducing inconsistencies in both composition and performance. Traditional calibration methods rely on manual adjustment and subjective listening, which are time-consuming and difficult to reproduce [6, 7]. Because drift progresses slowly, these changes may remain unnoticed until performance becomes unreliable. This motivates the need for systematic, repeatable, data-driven calibration methods.

2.3 Data-Driven Calibration Methods

Sensor-driven and machine learning methods have been increasingly used to characterize and correct the acoustic behavior of robotic instruments. Spectral features such as MFCCs, combined with dimensionality reduction, have been applied to model loudness variation in robotic percussion, improving dynamic stability [8].

Subsequent work extended these ideas through closed-loop control, using audio feedback to automatically correct deviations in pitch, dynamics, and timing [6, 9]. These systems demonstrate that algorithmic calibration can maintain stable performance over extended periods while reducing reliance on manual adjustment.

More recent work focuses on learning richer, instrument-specific response models. Self-supervised exploration enables robots to infer geometric and acoustic properties directly from interaction data [10], while large-scale learning frameworks for robotic piano performance show that modern models can compensate for nonlinear mechanical variation through data-driven policy learning [11, 12].

Collectively, these approaches demonstrate that data-driven modeling can capture complex acoustic behavior, detect drift earlier than human perception, and improve calibration consistency over time. In this work, we introduce a hybrid KNN–MLP framework that integrates data-driven and nonlinear modeling to achieve stable and repeatable calibration.

3. SYSTEM OVERVIEW

3.1 System Architecture

The calibration framework uses an ensemble learning approach [13] that integrates a KNN model with an MLP. A single model often exhibits higher prediction variance, particularly when datasets are sparse. By combining the complementary strengths of both models, the system produces more stable and accurate calibration predictions. The workflow consists of three stages:

1. Constructing an instrument-specific dataset,
2. Training the MLP on the recorded samples, and
3. Generating and blending predictions from both the KNN and MLP.

The system is implemented using ChucK's [14] artificial intelligence library, ChAi [15], enabling direct integration with the Machine Lab code-base and real-time execution.

3.2 Dataset Construction

In the Machine Lab, each instrument requires its own dataset due to differences in actuator design, mechanical layout, and acoustic behavior. Training samples consist of $(\text{note}, \text{velocity})$ pairs with corresponding RMS values that capture the dynamic response of each strike. For each note, strikes are recorded across the full playable velocity range $(60 - 127)$, with a 1.5-second delay to avoid acoustic overlap. Each strike is stored as a JSON record containing the note, velocity, and RMS value. The overall data collection workflow is summarized in Figure 2.

3.2.1 Pitched Instruments

For pitched instruments such as the MalletOTon [5], notes are sampled across the playable range at configurable intervals, along with all playable velocities. Sampling at different intervals controls dataset density: wider spacing reduces collection time and avoiding redundancy from acoustically similar notes, such as those with similar overtone structure and microphone response, while denser sampling provides more detailed coverage of the instrument's

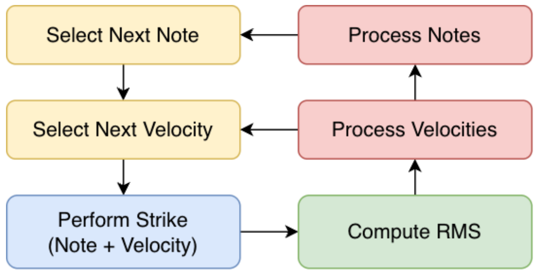

Figure 2. For each note and velocity, the system performs strikes, records RMS values, and iterates until all notes and velocities are sampled.

response. The interval is chosen per instrument to balance collection efficiency and prediction accuracy.

3.2.2 Non-pitched Instruments

For non-pitched instruments, such as GanaPati, each drum is treated as an independent acoustic source and requires a separate dataset. Individual drum differ in tuning, strike position, and timbral response, making a shared dataset across drums insufficient to capture their distinct characteristics. For each drum, a single mallet or beater is used to maintain consistency, and strikes are recorded using the same procedure as for pitched instruments.

3.3 K-Nearest Neighbors

The KNN [16] model provides predictions grounded in recorded data. By identifying nearest neighbors and interpolating between them, it preserves the instrument's measured acoustic behavior and avoids overfitting in well-sampled regions. However, its utility for full-range calibration is limited by its locality. When data is sparse, linear interpolation cannot capture the underlying nonlinear acoustic variation between notes or velocities. Without smoothing, KNN predictions reflect dataset noise, such as room acoustics and microphone response. This limits its ability to generalize beyond the observed samples.

3.4 Multilayer Perceptron

The MLP [17] complements the KNN model by learning nonlinear relationships beyond local interpolation. By modeling a continuous function, it captures smooth acoustic behavior across the instrument's playable range and reduces the impact of noise present in the dataset. However, these advantages come with limitations. When trained on sparse datasets, an MLP can extrapolate poorly, producing unrealistic RMS estimates near the boundaries of the training set. Its performance also depends on appropriate normalization and sufficient training data, with smaller datasets yielding less reliable behavior.

3.5 Hybrid Ensemble Model

The hybrid model combines the complementary strengths of KNN and MLP by blending their RMS predictions into a single output, as illustrated in Figure 3. The final calibration value is computed as:

$$\text{final\_volume} = \alpha \cdot v_{\text{mlp}} + (1 - \alpha) \cdot v_{\text{knn}} \qquad (1)$$

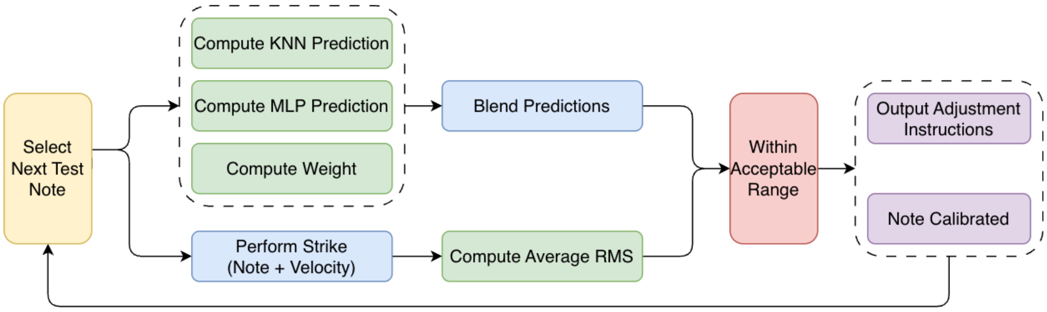

Figure 3. The system computes KNN and MLP predictions, measures the average RMS response, evaluates deviation from the predicted value, and determines whether mechanical adjustment is required.

where, v_{mlp} and v_{knn} denote the RMS predictions of the MLP and KNN models, respectively. The blending weight α is defined as a function of the distance between test sample and nearby training samples, allowing the model to balance local interpolation and nonlinear generalization. When a test sample coincides with a training sample, $\alpha = 0$, relying entirely on KNN. As the distance increases, α increases, progressively incorporating the MLP prediction. This weighting enables a smooth transition between data-driven interpolation and learned generalization.

4. EXPERIMENT

4.1 Experimental Setup

To evaluate the hybrid model, we use four dataset configurations across the playable range of MalletOTon. Specifically, 9, 13, 22, and 26 notes are sampled, corresponding to intervals of 5, 3, 2 semitones, and full coverage, respectively. These configurations simulate different levels of data availability, from sparse to dense sampling of the instrument. All experiments use 5-fold GroupKFold cross-validation, where folds are partitioned by note such that all velocity samples of a given note are assigned to the same fold. This ensures that no $(\text{note}, \text{velocity})$ samples from the same note appear in both training and validation sets, and that all predictions are made on unseen notes, requiring interpolation across notes.

4.2 Ensemble Weight Evaluation

We first examine the peak value of α, which controls the maximum contribution of the MLP. Peak values are varied from 0.15 to 0.85, with the optimal value found at $\alpha_{\max} = 0.60$, yielding the lowest average Mean Absolute Error (MAE) of 1.3853 across all dataset configurations.

Building on this result, we evaluate 61 weighting functions across five function families under the same cross-validation protocol. Table 1 presents the best-performing function from each family. The top-performing functions cluster within 0.006 MAE and consistently share the same peak value, indicating that performance is primarily governed by the maximum weighting rather than the specific functional form. Among the evaluated functions, the flat-top (trapezoidal) function achieves the lowest overall MAE by maintaining the peak across a wider portion of the interpolation range, resulting in a more consistent balance between MLP and KNN contributions.

Model	Parameters	Avg MAE
KNN	–	1.5251
MLP	–	1.4706
Hybrid (Flat)	$\alpha_{\max} = 0.60$	**1.3798**
Hybrid (Para)	$\alpha_{\max} = 0.60$	1.3853
Hybrid (Gauss)	$\alpha_{\max} = 0.60,\ \sigma = 0.4$	1.3880
Hybrid (Tri)	$\alpha_{\max} = 0.60$	1.4003
Hybrid (Sine)	$\alpha_{\max} = 1.00$	1.4664

Table 1. Average MAE across all dataset configurations.

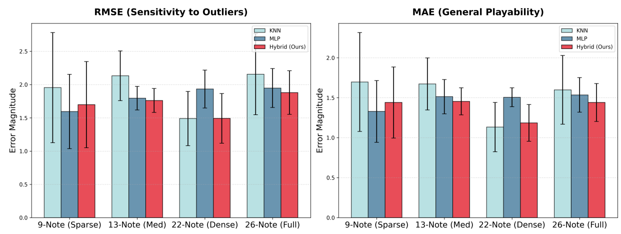

Figure 4. Performance of the hybrid model (flat-top weighting, $\alpha_{\max} = 0.60$) across dataset densities, compared with KNN and MLP using RMSE (left) and MAE (right).

4.3 Model Evaluation

Averaged across all dataset configurations, the hybrid model with flat-top weighting achieves the lowest MAE, outperforming both KNN and MLP (see Table 1). Individually, KNN performs best on densely sampled data (22-note), while MLP performs best under sparse sampling (9-note). However, neither model generalizes consistently across configurations. The hybrid model maintains stable performance without prior knowledge of dataset density. By preserving a KNN contribution in the weighting, it anchors predictions to recorded acoustic measurements while leveraging the MLP for smooth generalization and noise reduction. A tradeoff is observed in variance: sinusoidal weighting yields lower fold-to-fold variance in some cases, while capped functions achieve 5–6% lower MAE. In this context, lower error is prioritized for calibration accuracy.

5. CONCLUSIONS AND FUTURE WORK

This paper introduces a data-driven calibration framework for robotic instruments using a hybrid of KNN and MLP. By combining data-grounded interpolation with nonlinear approximation, the system produces accurate and repeatable predictions for pitched and non-pitched instruments.

Evaluation across multiple dataset densities shows that the hybrid model consistently outperforms standalone approaches, achieving the lowest average MAE. The results further indicate that calibration performance is primarily governed by the maximum blending weight rather than the functional form, providing a simple and robust design principle for balancing interpolation and generalization.

Beyond performance, the system provides a modular calibration pipeline. Its dataset design supports diverse instruments, while adaptive weighting enables stable predictions without requiring prior knowledge of data density. These properties make the framework suitable for long-term deployment where mechanical drift is unavoidable.

Future work will extend the model in several directions. Incorporating additional acoustic features may improve calibration fidelity beyond RMS. Integrating closed-loop feedback could enable continuous adjustment during performance, rather than relying on offline calibration. Finally, exploring more expressive models may improve behavior in sparse or rapidly changing conditions.

Acknowledgments

We thank Michael Darling, Ethan Brewer, Eric Browning, Samantha Cheng, Valentin Correa, Reese Downes, Amogh Dwivedi, Nathaniel Gladstone-Lyon, Ilai Gilbert, Daniel Gonzalez, Jake Morgan, Alexandra Pasquale, Will Rinkoff, Sol Rosenthal, and Ruiqi Wang for their contributions to the Modulets and the CalArts Machine Lab. We also thank Ge Wang, Yikai Li, and the Stanford CCRMA team for developing ChAI, which supported this research.

6. REFERENCES

[1] N. Villicaña-Shaw, S. Salazar, and A. Kapur, “The Machine Lab: A Modern Classroom to Teach Mechatronic Music,” in *Proceedings of the 2017 International Computer Music Conference*, Shanghai, China, 2017, pp. 482–487.

[2] A. Kapur, M. Darling, D. Diakopoulos, J. W. Murphy, J. Hochenbaum, O. Vallis, and C. Bahn, “The machine orchestra: An ensemble of human laptop performers and robotic musical instruments,” *Computer Music Journal*, vol. 35, no. 4, pp. 49–63, 2011.

[3] A. Kapur, E. Trimpin, A. Singer, G. Suleman, and G. Tzanetakis, “A comparison of solenoid-based strategies for robotic drumming,” in *Proceedings of the 2007 International Computer Music Conference*, Copenhagen, Denmark, 2007.

[4] A. Kapur, E. Singer, M. S. Benning, G. Tzanetakis, and Trimpin, “Integrating hyperinstruments, musical robots & machine musicianship for North Indian classical music,” in *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, New York, USA, 2007, p. 238–241.

[5] A. Kapur, J. Murphy, M. Darling, E. Heep, B. Lott, and N. Morris, “Malletoton and the modulets: Modular and extensible musical robots,” in *Proceedings of the 16th International Conference on New Interfaces for Musical Expression*, Brisbane, Australia, 2016, pp. 69–72.

[6] J. Murphy, M. Mathews, A. Kapur, and D. Carnegie, “Robot: Tune Yourself! Automatic Tuning for Musical Robotics,” in *Proceedings of the 14th International Conference on New Interfaces for Musical Expression*, London, UK, 2014, pp. 565–568.

[7] J. Murphy, D. Carnegie, and A. Kapur, “Using Expressive Musical Robots: Working with an Ensemble of New Mechatronic Instruments,” in *Proceedings of International Symposium for Electronic Arts*, 2016.

[8] S. R. Ness, S. Trail, P. F. Driessen, W. A. Schloss, and G. Tzanetakis, “Music Information Robotics: Coping Strategies for Musically Challenged Robots.” in *Proc. of the 12th Int. Society for Music Information Retrieval Conf.*, Miami, USA, 2011, pp. 567–572.

[9] J. Long, A. Kapur, and D. A. Carnegie, “The closed-loop robotic glockenspiel: Improving musical robots using embedded musical information retrieval,” in *Proceedings of the 16th International Conference on New Interfaces for Musical Expression*, Brisbane, Australia, 2016, pp. 2–7.

[10] M. Görner, N. Hendrich, and J. Zhang, “Pluck and Play: Self-supervised Exploration of Chordophones for Robotic Playing,” in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 18 286–18 293.

[11] Y. Huang, R. Li, and Z. Tu, “PANDORA: Diffusion Policy Learning for Dexterous Robotic Piano Playing,” *arXiv preprint arXiv:2503.14545*, 2025.

[12] L. Chen, Y. Zhao, J. Schneider, Q. Gao, S. Guist, C. Qian, J. Kannala, B. Schölkopf, J. Pajarinen, and D. Büchler, “Dexterous Robotic Piano Playing at Scale,” *arXiv preprint arXiv:2511.02504*, 2025.

[13] T. G. Dietterich, “Ensemble methods in machine learning,” in *International workshop on multiple classifier systems*, 2000, pp. 1–15.

[14] G. Wang, P. R. Cook, and S. Salazar, “Chuck: A strongly timed computer music language,” *Computer Music Journal*, vol. 39, no. 4, pp. 10–29, 2015.

[15] Y. Li and G. Wang, “Chai: Interactive ai tools in chuck,” in *Proceedings of the International Conference on New Interfaces for Musical Expression*, Utrecht, Netherlands, 2024, pp. 553–559.

[16] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967.

[17] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.