PIANNOTATE: A WEB ANNOTATION TOOL FOR PIANO FINGERING, WITH A DIAGNOSTIC PROBE

Joonhyung Bae[1], Kirak Kim[1], Hyeyoon Cho[1], Sein Lee[1]
Yoon-Seok Choi[2], Hyeon Hur[2], Gyubin Lee[1], Akira Maezawa[3]
Jonghwa Park[2], Jaebum Park[2], Juhan Nam[1,*]

[1]Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea
[2]Seoul National University, Seoul, South Korea
[3]YAMAHA, Hamamatsu, Japan

{jh.bae,kirak,hyeyooncho,seinlee,gbstorm81,juhan.nam}@kaist.ac.kr
{dallas71,sunny000927,parkpe95}@snu.ac.kr
pianistpark@gmail.com
akira.maezawa@music.yamaha.com
*Corresponding author

ABSTRACT

Piano fingering shapes how a passage can be played, yet it is difficult to label after a performance. An annotator must decide which finger produced each note while reconciling the score, timing, video, and hand motion. We present **PiAnnotate**, a web-based pipeline for adding expert fingering annotations to the FürElise performance dataset [1]. The tool brings together a piano-roll view, performance video, and a 3D MANO hand mesh so that reviewers can inspect each assignment in musical and physical context. Rather than storing only the final answer, PiAnnotate keeps paired rule-based and human-edited fingering tracks. These paired tracks make the annotation history auditable by showing where a geometric rule was sufficient, where experts intervened, and how labels changed across review passes. As a final diagnostic, we train a small Transformer probe on the paired tracks. The probe improves on the rule baseline on held-out pieces while remaining conservative about changing labels that were already correct, suggesting that the edited labels contain learnable structure rather than only isolated fixes.

1. INTRODUCTION

Many methods have tried to predict piano fingering, from ergonomic and physical rules [2, 3] to HMMs, statistical learning, and neural models [4–6]. These methods all depend on trustworthy labels for which finger played each note, a simple resource that is expensive to create. Existing fingering corpora such as PIG [5] are valuable, but they cover only part of the annotation need. Recent multimodal datasets and toolkits capture richer performance evidence, including video and MIDI workflows [7, 8]. FürElise [1] goes further by providing synchronized audio, video, MIDI, and 3D hand motion [9, 10], but it still does not say which finger played each note.

Turning motion data into fingering labels is still an expert annotation problem. Reviewers need to compare note events with hand position and decide whether an assignment is physically plausible. PiAnnotate is designed for this setting. It does not replace expert review; it makes review easier to audit by placing motion evidence, rule labels, edited labels, and review-stage metadata on the same timeline. Time-aligned tools exist in adjacent domains, including Praat [11], ELAN [12], and Sonic Visualiser [13], but they lack 3D hand-mesh rendering and finger-specific piano workflows.

This paper describes **PiAnnotate**, the pipeline we used to add expert fingering annotations to the full FürElise collection. Our contributions are as follows.

1. A web annotation tool with frame-level editing, 3D MANO hand mesh [14] visualization, and an explicit multi-stage review workflow (Sec. 3).
2. An expert-verified fingering layer for FürElise, organized as paired rule-based and human-edited tracks so that rule errors can be inspected directly (Sec. 4).
3. A small Transformer-style diagnostic probe [15] that checks whether the edited labels contain learnable structure beyond isolated fixes and surfaces a timestamp-related annotation artifact (Sec. 5 – 6).

We borrow the term *diagnostic probe* from NLP [16]; to our knowledge, this is the first application to a music annotation corpus. The model is not a deployable corrector because the full collection is used in training and the inference gate is a composite confidence filter. The setup is based on human-in-the-loop annotation [17],

© 2026 Korean Society for Music Informatics. **Citation:** Joonhyung Bae, Kirak Kim, Hyeyoon Cho, Sein Lee, Yoon-Seok Choi, Hyeon Hur, Gyubin Lee, Akira Maezawa, Jonghwa Park, Jaebum Park, and Juhan Nam, "PiAnnotate: A Web Annotation Tool for Piano Fingering, with a Diagnostic Probe", *The First Korean Society for Music Informatics Conference*, Seoul, Republic of Korea, May 30, 2026.

model-assisted relabeling [18], and learning from noisy labels [19].

2. PIPELINE OVERVIEW

Figure 1 sketches the loop. A rule-based annotator extracts MIDI onset-offset intervals on the motion-frame grid. For each pressed key at onset, it scans both hands' five MANO fingertips, keeps candidates inside the key's pitch and front-back bounds and near the key surface, and chooses the minimum score combining surface-height distance with normalized front-back distance. If no fingertip passes the test, the rule label is missing. The assigned label is propagated until the note offset to produce f_{rule}. A human reviewer opens the piece in the PiAnnotate web tool, edits frames where the rule output is wrong, and marks the piece as R_1-checked in a per-piece status JSON. Once a non-trivial R_1 pool exists, a probe Transformer is trained on $(f_{\text{rule}}, f_{\text{edited}})$ pairs and run on every piece; its output is written to a separate directory and optionally surfaced to the reviewer in subsequent passes (R_2, R_3).

3. ANNOTATION TOOL

The tool is a browser application: a Vite/React frontend talks to a Flask backend over a small REST API. Each piece bundles the synced video, audio, MIDI, keypress timing, and per-frame 3D hand mesh provided by FürElise [1]; PiAnnotate adds the fingering layer on top (Figure 2). The annotator sees three coupled views: a piano roll with overlaid finger labels colored by hand; the original performance video; and a 3D MANO hand mesh rendered for the current frame, which makes physically implausible assignments (e.g., a thumb on a key the thumb cannot reach) immediately visible. Editing is keyboard-driven (Table 1): $\leftarrow$ / $\rightarrow$ steps through fingering events, digit keys 1–5 assign a finger, and `Space` toggles playback.

Key	Action
`Space`	Play / Pause
$\leftarrow$ $\rightarrow$	Previous / Next fingering event
`1–5`	Assign finger number
`F`	Toggle fingering overlay
`ESC`	Deselect current fingering

Table 1. Keyboard shortcuts in the annotation tool.

Edits are committed to an edited file per-piece. Each piece carries a JSON status tracking three review stages (R_1, R_2, R_3) with timestamps; only R_1-checked pieces are admitted to probe training. The tool also provides an in-browser backup/recovery path for unsaved edits, which has proven necessary in practice during multi-hour annotation sessions. All three views update synchronously as the annotator moves through the timeline.

4. CORPUS

The annotated corpus consists of all 153 piano performances of the FürElise dataset [1], totaling ∼5.3 M note events with fingering labels. All 153 pieces have been through a first expert review pass (R_1); 62 have a second pass (R_2) and 91 have a third pass (R_3). R_1 passes were produced by trained research annotators; R_2 and R_3 were performed individually by a different music specialist who independently reviewed all assigned pieces. Where the two specialists disagreed, the final label was resolved by consensus. This two-specialist review covers 152 of 153 pieces and improves label quality, but it does not provide a formal inter-annotator agreement coefficient. For every piece, we store *paired* fingering tracks: a rule-based prediction f_{rule} and a human-edited version f_{edited} on the same motion-frame grid. This pairing is the structural feature that distinguishes the corpus from prior fingering datasets: each note is labelled with both what an automatic annotator predicted and what the human ultimately accepted, so the rule-error population is a first-class object that downstream models can train on or audit against. Aggregate statistics are in Table 2. The 91.82% rule–edited agreement is computed over all 5,301,309 edited notes. This aggregate combines wrong-finger assignments with notes that the rule annotator missed entirely, both counted as rule errors. Per-piece disagreement has median 7.4% (IQR [4.9, 10.6]%, max 28.4%), leaving ∼433 k rule-error notes as the natural correction target. Unlike PIG [5], whose labels are score-level and monolithic, PiAnnotate is frame-aligned to performance motion and stores rule and human tracks in parallel, enabling the held-out diagnostics in Sec. 6.

Data card. *Source recordings* (153 pieces, ∼10 h, 15 elite pianists, Baroque through 20C) are reused from FürElise [1] and *not* redistributed; users obtain them from the FürElise release. *What we add*: (i) the rule-based fingering track f_{rule} from a geometric distance assignment on the FürElise hand meshes; (ii) the human-edited track f_{edited} produced through the PiAnnotate tool (R_1 by research annotators, R_2/R_3 by two independent music specialists with consensus resolution); (iii) per-piece R_1/R_2/R_3 status JSONs. The annotation tool, probe code, and analysis scripts, but not the edited fingering labels, are released under MIT at `https://github.com/joonhyungbae/PiAnnotate`.

Property	Value
Pieces	153
Pieces R_1-verified	153
Pieces with R_2 review	62
Pieces with R_3 review	91
Total notes (rule)	5,089,395
Total notes (edited)	5,301,309
Rule–edited agreement	91.82%
Rule-error population	8.18% (433 410 notes)

Table 2. The PiAnnotate corpus at the time of writing. Edited contains ∼200 k more notes than rule because human reviewers also *add* fingerings that the rule-based annotator missed entirely.

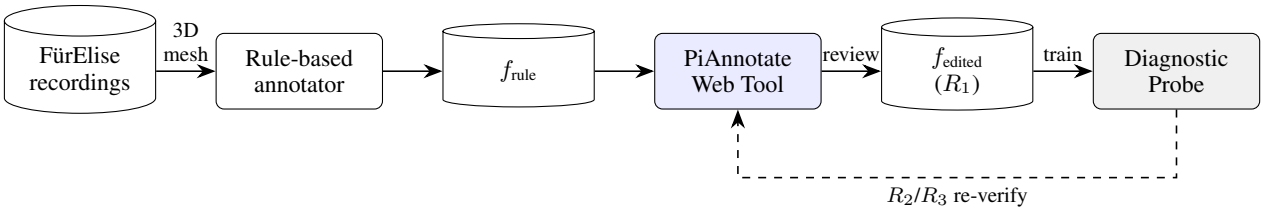

Figure 1. PiAnnotate pipeline. FürElise 3D hand-mesh recordings are processed by a geometric rule-based annotator to produce f_{rule}, which the annotator reviews and corrects in the web tool to produce f_{edited}. Edited pairs train a diagnostic probe whose output is optionally surfaced in subsequent review passes (dashed).

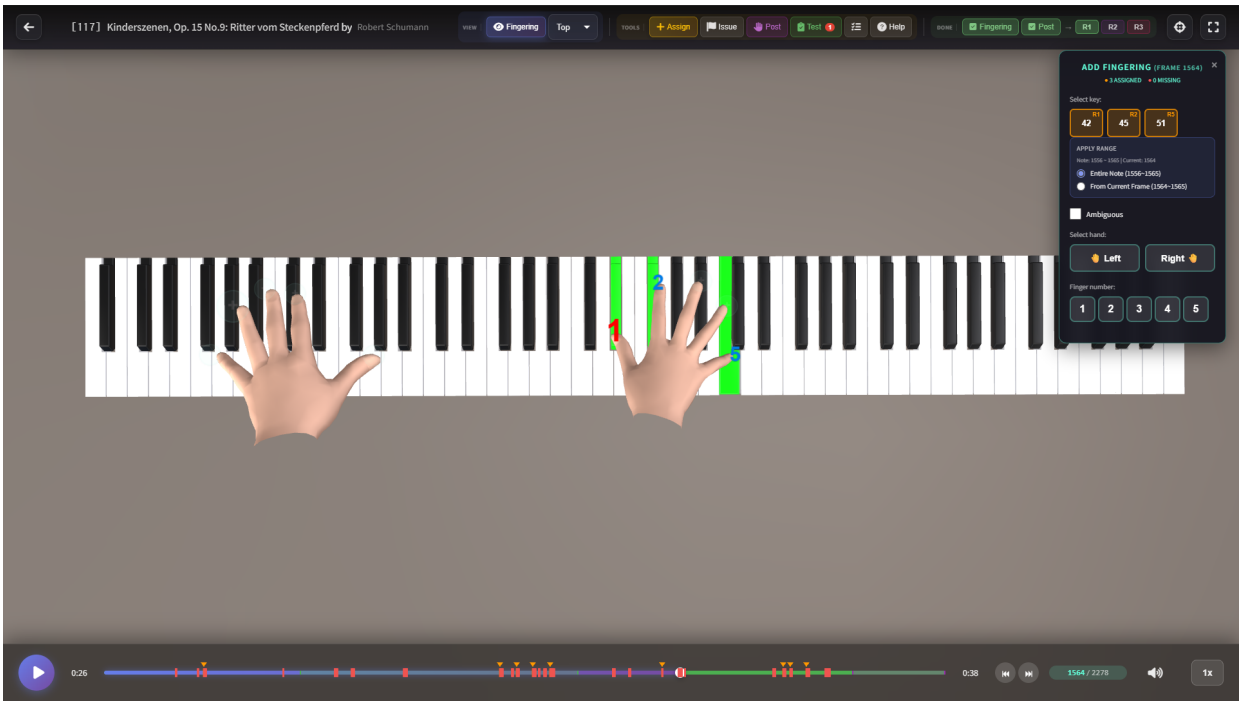

Figure 2. The PiAnnotate web annotation tool. A top-down piano view renders the FürElise hand-mesh tracks in 3D for the current frame; pressed keys are highlighted in green and overlaid with their currently assigned finger numbers (red = left hand, blue = right). The right-hand panel adds a fingering at the playhead frame: the key, hand, and finger are selected with single clicks, and the assignment can be extended to the entire sustained note or applied only from the current frame onward. The bottom timeline shows the audio waveform with click-onset markers and the per-piece review state ($R_1/R_2/R_3$) is tracked in the top-right corner.

5. DIAGNOSTIC PROBE

We train a small causal Transformer encoder [20] in pairs $(f_{\text{rule}}, f_{\text{edited}})$ from the corpus. Let $f_i^{\text{rule}} \in \{0, 1, \ldots, 10\}$ be the rule label for the onset of the i -th note (0 = missing) and $f_i^\star$ the edited label. Each note is encoded by a 77-d feature vector. It contains five key-geometry terms (normalized key index, black-key flag, key-center x/y, and surface height), 60 fingertip-geometry terms (10 fingertips $\times$ $x/y/z$ offsets to the current key, absolute height, and two in-range flags), and 12 rule-label descriptors (hand/finger one-hots and missing/match flags). A NoteEncoder projects these to d and embeds f_i^{rule} with a learnable per-class vector that is added to the note representation. Note embeddings are pooled per onset group and fed to a causal Transformer encoder. The main configuration used in Sec. 6 is intentionally small, with 1 encoder layer and d=64. Its learned rule-label embedding is zeroed and frozen at training time, while fixed rule descriptors still remain in the 77-d note vector and the inference gate. A 4-layer, d=256 configuration with the rule-label embedding active is reported as a capacity ablation. Two heads are trained jointly: an 11-way class head and a binary correction head, with loss

$$\mathcal{L} = \text{CE}(f_i^{\text{cls}}, f_i^\star) + \text{BCE}(c_i, c_i^\star),$$

where $c_i^\star = \mathbf{1}[f_i^{\text{rule}} \neq f_i^\star]$. We deliberately take rule-based fingering rather than raw MIDI as input, because we observed that MIDI–motion frame alignment drifts on long pieces (the first $\sim$5 000 frames match at 99.6% but later sections fall to 10–30%); the rule-based fingering is already on the motion frame grid.

Inference gate. The probe overrides the rule label for

the note i when

$$\left(\hat{f}_i^{\text{cls}} \neq f_i^{\text{rule}}\right) \wedge \left(p_i^{\text{cls}} > 0.9\right) \wedge \left(p_i^{\text{cls}}/p_i^{\text{rule}} > 2\right), \quad (1)$$

where p_i^{cls} and p_i^{rule} are the probabilities of the classifier head for its prediction top-1 and for the rule label, respectively. The thresholds 0.9 and $2\times$ are implementation defaults and were not tuned on the slices reported in Sec. 6. A sweep of sensitivity on seed 0 (no-rule embossed) over the top-1 threshold $\tau \in \{0.70, 0.80, 0.90, 0.95\}$ gives hold-out margins of $\{-2.14, +1.28, +2.78, +2.83\}$ pp, respectively, so 0.90 sits on the flat end of the curve and 0.95 is nearly indistinguishable from the default. The correction-head logit is an auxiliary training objective and does not enter Eqn (1); sweeping on its probability alone yields $P{\leq}0.13$. All numbers in Sec. 6 are produced by Eqn (1).

6. FINDINGS

We score the probe against the current edited labels in the full corpus and in the R_2/R_3 subsets. All 153 pieces are in training; the subsets differ in how many human passes their labels have received, so we read them as label-quality slices rather than as held-outs.

Finding 1. Precision-first triage. In the whole corpus, the gate of Eqn (1) indicates 1.91% of the notes as suspect. Of those flags, 79.7% land on notes where the rule actually disagrees with the human – about 10× above the 8.18% chance rate. The cost of this precision is the recall: only 28.0% of rule errors are flagged. The two paired numbers, in this order, are the operating point at which we recommend reading the probe.

Finding 2. Near-zero break rate. In the same corpus, the probe overwrites only 14,011 of 4,867,897 originally-correct rule labels (0.29%). In the R_2 cut this is 6,876 of 2,297,010 (0.30%). The gate of Eqn (1) is structurally conservative: when in doubt it leaves the rule alone. This is the property that makes the probe usable as a triage filter rather than as a re-annotator.

Held-out generalization. To separate memorization from generalizable signal, we retrained the main probe (1L, d=64, no-rule embedded) on the 91 pieces that are not in R_2, held out the 62 R_2 pieces entirely, and repeated with five independent random seeds. We report Δ as probe accuracy minus rule accuracy on the same held-out pieces, measured in percentage points. The per-seed margins are $\Delta \in [+2.54, +2.92]$ pp; every paired piece-level cluster bootstrap (5 000 resamples of whole pieces) places its 95% lower bound above zero. The combination gives $\Delta = +\mathbf{2.83}$ pp (Student t CI $[+2.63, +3.03]$, df=4), with recall $42.6 \pm 2.1\%$, precision $88.4 \pm 5.4\%$, and break rate $0.19 \pm 0.33\%$. Thus the same 62 pieces that show -5.22 pp train-fit underperformance show positive held-out margins in every seed.

Robustness. All four perturbations leave the conclusion intact (Table 3, bottom blocks). *Capacity*: scaling to 4 layers, d=256 gives $+2.14$ pp ($[+1.43, +2.86]$) – the same mean, $3.6\times$ wider σ, so extra capacity only adds calibration noise. *Rule label as input*: reinforcing the embedding of the rule label in the larger model gives $+2.10$ pp ($[+1.19, +3.01]$), indistinguishable in mean but with higher break rate, ruling out a rule-as-input shortcut. *Partition*: an independent random 91/62 split gives $+3.10$ pp ($[+2.54, +3.65]$, recall 52%), so the R_2 partition is not a suspiciously easy slice. *Architecture family*: a gradient-boosted decision tree (GBDT) [21], implemented as a per-note classifier on the same 77-d features without sequence context, reaches only 70.4% on the R_2 holdout under Eqn (1); even with τ=0.99, its margin is $\Delta = -2.94$ pp. The signal therefore requires the per-note features plus minimal causal attention.

Setting	Flag P/R	Break	Rule	Probe
Train-fit, single seed (all 153 in training)				
All R_1 (153)	79.7/28.0	0.29%	91.82	89.98
R_3 (91)	81.8/33.3	0.28%	92.39	**93.57**
R_2 (62)	76.9/22.9	0.30%	91.39	86.17
Held-out 5-seed (62 R_2 pieces excluded)				
Main (1L,64)	88.4/42.6	0.19%	91.39	**94.22**
Abl. 4L,256	81.5/43.1	0.58%	91.39	93.54
Abl. +rule	78.6/43.1	0.85%	91.39	93.49
Held-out 5-seed (random 62/91 split, 4L)				
Random	85.8/52.2	0.47%	91.28	**94.37**
Non-sequence baseline (GBDT, 1 seed, R_2 split)				
GBDT 77-d feats	12.9/79.8	26.5%	91.39	70.41

Table 3. Probe diagnostics. The top block is train-fit on label-quality slices, not a generalisation test. The middle block is the 5-seed R_2-excluded holdout; the main probe freezes the learned rule-label embedding, while fixed rule descriptors and the gate remain. Bottom blocks report a random split and GBDT baseline.

Label vintage. The train-fit agreement is $+1.18$ pp on R_3 but -5.22 pp on R_2 (Table 3). A timestamp audit explains this gap. For 50/59 R_2 pieces, the probe output predates the R_2 completion timestamp, i.e. the labels against which it is scored did not yet exist when it was trained. The lesson is operational: paired-track corpora under active editing need explicit label-vintage tracking, which we now record in the per-piece status JSON.

7. DISCUSSION AND CONCLUSION

PiAnnotate does not remove expert review, and the probe is not a deployable corrector. Its role is diagnostic: paired rule/edited annotations support a conservative triage point, a near-zero break rate, and a $+2.83$ pp held-out gain. The capacity, random-split, and GBDT checks argue against a simple learned rule-label shortcut.

Limitations. The corpus uses one capture setup and 15 performers; no formal inter-annotator agreement coefficient is reported. The held-out split follows review stage rather than composer or difficulty, reviewer-time impact is unmeasured, and the no-rule-embedding ablation still leaves fixed rule descriptors and the gate. The released artifacts are the tool and probe code, not the edited labels. Together, these results position PiAnnotate as an inspectable expert-annotation workflow; future work should measure reviewer time and evaluate beyond FürElise.

8. References

[1] R. Wang, P. Xu, H. Shi, E. Schumann, and C. K. Liu, "Fürelise: Capturing and physically synthesizing hand motion of piano performance," in *SIGGRAPH Asia 2024 Conference Papers*, 2024, pp. 1–11.

[2] R. Parncutt, J. A. Sloboda, E. F. Clarke, M. Raekallio, and P. Desain, "An ergonomic model of keyboard fingering for melodic fragments," *Music Perception*, vol. 14, no. 4, pp. 341–382, 1997.

[3] A. Al Kasimi, E. Nichols, and C. Raphael, "A simple algorithm for automatic generation of polyphonic piano fingerings," in *Proc. 8th Int. Society for Music Information Retrieval Conf. (ISMIR)*, Vienna, Austria, 2007, pp. 355–356.

[4] Y. Yonebayashi, H. Kameoka, and S. Sagayama, "Automatic decision of piano fingering based on a hidden markov models." in *IJCAI*, vol. 7, 2007, pp. 2915–2921.

[5] E. Nakamura, Y. Saito, and K. Yoshii, "Statistical learning and estimation of piano fingering," *Information Sciences*, vol. 517, pp. 68–85, 2020.

[6] P. Ramoneda, D. Jeong, E. Nakamura, X. Serra, and M. Miron, "Automatic piano fingering from partially annotated scores using autoregressive neural networks," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 6502–6510.

[7] Y. Kim, J. Park, J. Bae, K. Kim, T. Kwon, A. Lerch, and J. Nam, "PianoVAM: A multimodal piano performance dataset," in *Proc. Int. Soc. Music Information Retrieval Conf. (ISMIR)*, 2025.

[8] J. Park, Y. Kim, J. Bae, K. Kim, T. Kwon, A. Lerch, and J. Nam, "Two web toolkits for multimodal piano performance dataset acquisition and fingering annotation," 2025.

[9] T. Simon, H. Joo, I. Matthews, and Y. Sheikh, "Hand keypoint detection in single images using multiview bootstrapping," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 1145–1153.

[10] W. Goebl and C. Palmer, "Temporal control and hand movement efficiency in skilled music performance," *PloS one*, vol. 8, no. 1, p. e50901, 2013.

[11] P. Boersma, "Praat, a system for doing phonetics by computer," *Glot. Int.*, vol. 5, no. 9, pp. 341–345, 2001.

[12] H. Brugman and A. Russel, "Annotating multimedia/multi-modal resources with elan," in *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, 2004.

[13] C. Cannam, C. Landone, and M. Sandler, "Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files," in *Proceedings of the 18th ACM international conference on Multimedia*, 2010, pp. 1467–1468.

[14] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies," *ACM Transactions on Graphics*, vol. 36, no. 6, pp. 245:1–245:17, 2017.

[15] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, I. Simon, C. Hawthorne, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer," in *International Conference on Learning Representations*, 2019.

[16] Y. Belinkov and J. Glass, "Analysis methods in neural language processing: A survey," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 49–72, 2019.

[17] B. Settles, "Active learning literature survey," University of Wisconsin-Madison, Tech. Rep. Computer Sciences Technical Report 1648, 2009.

[18] C. Northcutt, L. Jiang, and I. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *Journal of Artificial Intelligence Research*, vol. 70, pp. 1373–1411, 2021.

[19] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, "Learning from noisy labels with deep neural networks: A survey," *IEEE transactions on neural networks and learning systems*, vol. 34, no. 11, pp. 8135–8153, 2022.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[21] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.