method	n_{unique}	n_{player}	n_{Σ}	t_{Σ}
Suzuki, Vol. 1-5	40	4	158	248.8
Dancla, Op. 84	27	2	59	131.9
Wohlfahrt, Op. 45	41	6	357	458.1
Sitt, Op. 32 Vol. 1-3	34	2	60	140.1
Kayser, Op. 20	5	8	40	81.9
Mazas, Op.36	12	4	35	100.7
Dont, Op. 37	10	3	30	68.1
Kreutzer, Études	24	4	95	229.8
Fiorillo, Op. 3	13	3	34	72.1
Rode, Op.22	7	5	35	82.5
Dont, Op. 35	7	2	14	31.7
Gavinies, Matinées	6	2	8	24.6
Total	**226**	**21**	**925**	**1670.2**

Table 1. Pedagogical methods ordered in approximate difficulty. n_{unique}: number of unique monophonic studies, n_{player}: number of distinct players, n_{Σ}: recording count per method, t_{Σ}: recording duration in minutes per method.

The most effort was spent on curating a monophonic [2] subset of these works by manually going through the scores on IMSLP and discarding all the works that include even a single double stop or chord, which corresponds to manually removing 264 of 490 studies [3]. The remaining 226 monophonic pedagogical works are summarized in Table 1 in their usual order of use in the violin curriculum.

The videos of the selected works are collected from YouTube by querying method/etude names followed by manually identifying the most reliable content creators, and then searching with queries including their names. Some videos, especially in the beginner repertoire, include different studies as multiple chapters within the same recording. These chapters are split and the audio recordings are extracted from the source videos with the highest possible quality, resulting in 925 performances.

3.2 Metadata

For each recording, we provide metadata with player ID, study No., which method/etude book it belongs to, and piece-wise difficulty rankings from multiple sources. Since difficulty is a subjective term, sources sometimes disagree on the ranking of these materials, but there are some common patterns: Suzuki Vol. 1-2 are always considered the easiest of all, with Dancla, Wohlfahrt, Suzuki Vol. 3-5 being the next. Rode, Gavinies, Fiorillo, and Dont Op. 35 form the hardest cluster. Whilst difficulty grades change from source to source, the progressive ordering within each book is universally accepted, e.g. study No. 30 from a method is always considered as harder than No. 2. Hence, *Violin Etudes* can be considered as a performance difficulty analysis dataset similar to [37].

The performers in the dataset are manually confirmed to be highly-skilled violinists, mostly violin teachers creating content for their students. As shown in Figure 1, our dataset includes performances from 21 players and is also quite skewed in favor of Players 01 and 02. Although this is a limit of the dataset from one aspect, with more than 5 hours of monophonic recordings for Player02 and 7 hours for Player01, these subsets include enough data to train violin synthesizers. In Table 1, we can see that we have a total of 925 reference performances for 226 monophonic violin studies. With some exceptions, each study has multiple renditions which allow high-level music research such as expressive performance analysis (exemplified briefly in Section 5.1), or low-level audio representation learning as exemplified in the f_0 estimation experiments in Section 5.2.

Despite its many strengths, the current state of *Violin Etudes* has two main flaws. Although the scores are included, only Mazas and Kayser etudes are in a machine-readable format, and we do not have their aligned scores. As repertoire gets harder, it gets harder to find non-commercial reference recordings. Thus, recording distribution in the dataset is limited by method popularity and difficulty, e.g., Matinées in Table 1 is only 24 minutes.

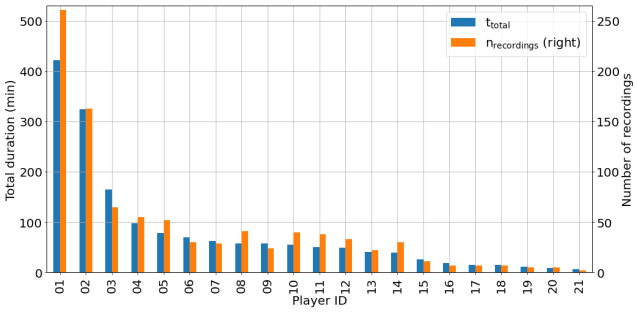

Figure 1. Total performance duration (left) and number of recordings (right) per PlayerID.

4. AUTOMATIC F_0 LABELING

The f_0 labels in the Violin Etudes dataset are automatically generated through a novel iterative f_0-labeling strategy based on two assumptions: (i) the recordings are monophonic, i.e. we can apply harmonic analysis with respect to an initial $\hat{f}_0$ estimate, and (ii) all the frames have a similar harmonic structure, i.e. violin is the sole instrument. After obtaining raw $\hat{f}_0$ estimates through a data-driven f_0 estimator, we search for harmonic peaks around the $\hat{\mathbf{f}}_{0:N}$ multiples of the $\hat{f}_0$ estimate for each frame through a modified version of Spectral Modeling Synthesis (SMS) [38]. We then force the harmonics to follow the $\hat{f}_0$ label similar to the analysis/synthesis framework of Salamon et al. [11], with additional novel constraints on instrument modeling and harmonic consistency to silence low-confidence segments. The remaining data is smaller in amount, yet it is more reliable to be used in the training of a new f_0 estimator. We then retrain the f_0 estimator using these resynthesized versions and extract new f_0' estimates, and repeat the process. Thus, as exemplified in Figure 2, both the f_0 estimator's performance and the dataset's f_0 label quality are enhanced simultaneously by interacting with one another.

[2] Here monophonic simply refers to *single note at a time*, i.e. removal of superposed notes, rather than the musicological definition.

[3] Unlike the use of violin in popular music, the classical and pedagogical violin repertoire exhaustively include double stops and chords.

4.1 Constrained Harmonic Resynthesis

First introduced by Serra et al. [38], Spectral Modeling Synthesis (SMS) has had widespread adoption in audio signal processing, including a recent revival with differential Digital Signal Processing (DDSP) [39]. We adopt the harmonic model from SMS to create constrained f_0 labels where we force the synthesis to follow the label or simply silence the frame if it does not satisfy the constraints.

4.1.1 Harmonic Analysis

Waveforms in 44.1kHz are analyzed with 1025-sample Blackman-Harris windows and hop size of 128 using `harmonicModelAnal` function from the SMS-tools [38]. The DSP-based f_0 detection algorithm of the `harmonicModelAnal` is replaced with our external $\hat{f}_0$ estimates; and harmonic amplitudes, frequencies, and phases are searched around the $\hat{\mathbf{f}}_{0:39}$ with a harmonic deviation slope of 0.001. We refer to [38] for further details.

4.1.2 Instrument-modeling Constraint (IC)

Instrument timbre defines the harmonic amplitudes $\mathbf{A}_{0:39}$ we see on top of an $\hat{f}_0$ estimate. Thus, if we know the instrument model, i.e. $P(\mathbf{A}_{0:39}|\hat{\mathbf{f}}_{0:39})$, we can assess the correctness of an estimate from harmonics. We use this idea as an additional layer of regularization of the label quality and removal of automatic-labeling artifacts. We learn an approximate model for violin resonance structure by linearly dividing the violin frequency range into N regions and fitting elliptic envelopes to the first 12 harmonic amplitudes for each of these N regions, where we experimented with $N = 50$ and 100. If this elliptic envelope model detects an anomaly in the harmonic amplitudes, we silence the frame.

4.1.3 Harmonic Consistency Constraint (HCC)

As we will show in Figure 4, an f_0 estimator is prone to overfitting problems associated with its training set distribution. To remedy this and ensure a reliable spectral distribution for our labels, we employed the Two-Way-Mismatch (TWM) procedure [19] between harmonic peaks $\hat{\mathbf{f}}_{0:39}$ and $\hat{f}_0$ candidates around the initial $\hat{f}_0$ estimate. For each voiced segment, 33 pitch candidates are selected in the range $(\hat{f}_0 - 16c, \hat{f}_0 + 16c)$ with 1 cent intervals. The candidate with the lowest Two-Way-Mismatch-error is selected to be the new $\hat{f}'_0$ estimate if its TWM-error is smaller than 5.0, and if this Harmonic Consistency Constraint is not satisfied, the frame is silenced. Finally, the constrained harmonic frequencies $\hat{\mathbf{f}}'_{\mathbf{0:39}}$ are set to the *exact multiples* of this final $\hat{f}'_0$ estimate before resynthesis.

4.1.4 Sinusoidal Synthesis

The constrained harmonics are resynthesized using the sinusoidal model from SMS [38]. Any segments shorter than 50ms are muted before synthesis to reduce artifacts. Furthermore, for each resynthesized recording, we also resynthesize its replicate with pitch shifts: both harmonics and f_0 labels are shifted with random microtonal pitch shifts in the range of 5-55 cents to ensure the statistical diversity of our labels. We found that training f_0 estimator with shifted versions of the recordings increases the stability of the estimator against equal temperament deviation.

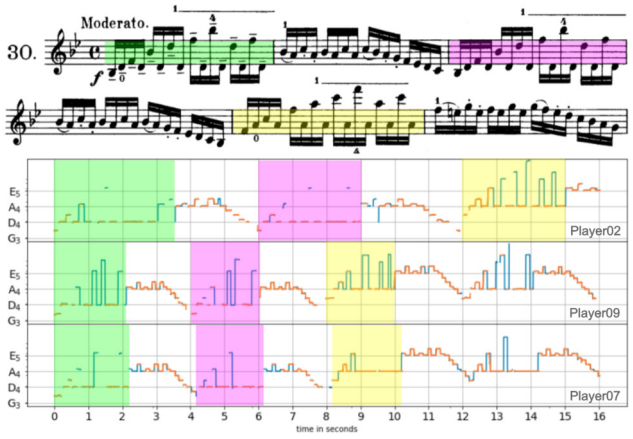

Figure 2. Iterative f_0 labeling exemplified in Kreutzer Etude No.30 performed by Players 02, 09, and 07. Constrained harmonic resynthesis acts as a barrier to wrong f_0 estimates and creates discrepancies in the initial f_0 contours (orange). Notice that most of these discrepancies are filled after the first iteration of finetuning (blue), especially in the highlighted string crossings.

4.2 Iterative Refinement of f_0 Labels

Initial f_0 tracks of the dataset are generated via the CREPE [8] convolutional f_0 estimator with 1 ms intervals and a custom Viterbi decoding to incorporate some heuristics we know about the violin repertoire. CREPE has 360 bins in its final layer with 20 cents between consecutive bin centers, i.e., states. In their implementation, they apply Viterbi with constant state observation probabilities without utilizing the confidences given by the algorithm. We replaced this by decoding with CREPE confidences as posteriors, similar to standard ANN-HMM posteriorgram decoding [40]. Using our prior knowledge of violin repertoire, we decided the Viterbi transition probabilities empirically as a weighted sum of two Gaussians to allow for fast string jumps while encouraging continuous f_0 contours: Gaussian with $\sigma_1 = 6$ semitones in Equation 1 enables fast string jumps, whilst continuous contours are encouraged by weighting the other Gaussian ($\sigma_2 = 40$ cents) with 9.

$$Pr(s_j(t+1)|s_i(t)) = \frac{1}{30\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{s_j(t+1) - s_i(t)}{30}\right)^2\right) + \frac{9}{2\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{s_j(t+1) - s_i(t)}{2}\right)^2\right) \quad (1)$$

After the initial f_0 estimates are obtained, (audio, f_0) pairs go through the Constrained Harmonic Resynthesis which silences out the wrong estimates as described in Section 4.1. The remaining (constrained audio, constrained f_0) pairs are used for finetuning of the f_0 estimator, which produces new estimates as exemplified in Figure 2. This iterative labeling process can be thought of as an Expectation-Maximization hybrid akin to ANN-HMMs [40].

	Violin		Other Inst.	
	RPA50	RPA5	RPA50	RPA5
Pretrained [8]	96.4	68.3	**96.2**	68.1
HCC	96.3	83.8	95.2	76.4
HCC + IC50	**96.7**	84.0	94.6	**76.5**
HCC + IC100	**96.7**	**84.2**	94.8	75.7
Sawtooth	68.3	49.5	56.8	37.2

Table 2. The pre-trained CREPE compared with training-from-scratch on different versions of *Violin Etudes*. Tests are conducted on the unseen URMP dataset. HCC: Harmonic Consistency Constraint. IC: Instrument-modeling Constraint. Sawtooth: Single-timbre control group.

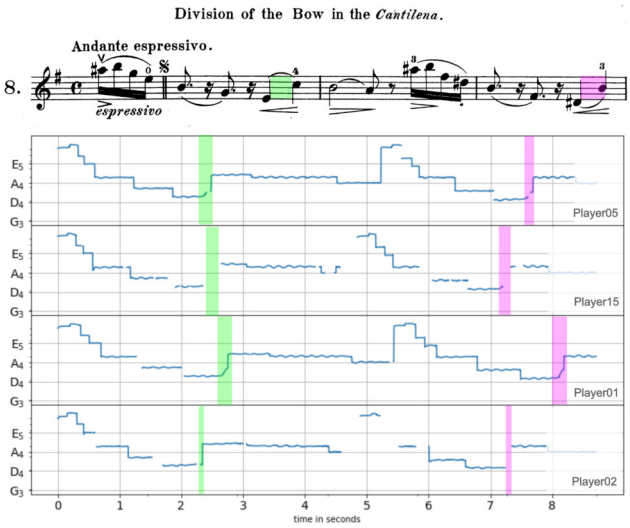

Figure 3. First line of Mazas Etude No. 8 with constrained f_0 labels of 4 performances. From the composer's words, we see that this etude targets teaching *division of the bow in the Cantilena*. f_0 tracks are automatically extracted with the method described in Section 4. Highlighted hand position changes are discussed in Section 5.1

5. EXPERIMENTS

5.1 Qualitative Performance Analysis

In this section, we give two brief examples of how *Violin Etudes* enable research on the pedagogically-motivated analysis of reference performances. In Figure 3, Mazas Op.36 No.8 with f_0 tracks of 4 different renditions can be seen: We can observe that both Players 05 and 01 have very expressive slides at the position changes indicated with green (E_4-C_5) and pink ($D_4^\sharp$-B_4), while Player02 landed into these notes directly -note that such analyses are not possible for corpora annotated with pYIN [7] or CREPE [8] since their Viterbi implementations smooth every jump. But, our method, too, has flaws: analysis of Player15 for the same section is inconclusive due to our design choice of removing unreliable automatic f_0 estimates.

Having multiple reference recordings also enable the analysis of performance tempo. In Figure 2, we see that Players 09 and 07 agree on a considerably faster pace for *Moderato* on Kreutzer Etude No.30, while Player02 play the same study a lot slower and sticks to this choice throughout the performance. Although we did not include it here, studying bowing technique analysis is also possible in *Violin Etudes*. Following the textual descriptions, players record some etudes with bowing variations. The most extreme case of this is the Player01 recording Wohlfahrt etudes with all the bowing variations, resulting in a subset of 171 recordings where some studies were repeated with up to 15 bowing variations. Moreover, the dataset allows for analyzing intonation and tuning patterns of professional violin teachers thanks to the high frequency precision of our f_0 estimates, which we will discuss in the next section.

5.2 Monophonic f_0 Estimation

Here we study the label quality of *Violin Etudes* in the context of f_0 estimator training. We train the commonly-used CREPE [8] f_0 estimator in our data, and compare it with the pre-trained version [4] trained on the following six manually-labeled and synthetic datasets: MIR-1k [10], MedleyDB [43], MDB-STEM-Synth [11, 43], Bach10-mf0-synth [11, 42], RWC-Synth [7], and NSynth [44].

We use *Violin Etudes* only for training and validation with 80/20 split, and train with a batch size of 32 and early stopping on validation accuracy. We conduct the tests on two unseen datasets where we evaluate the raw pitch accuracy (RPA, in %) with two thresholds: the conventional RPA50 quarter-tone accuracy, and our benchmark RPA5 fine-grained accuracy. Owing to the high-frequency precision requirements for performance analysis, this second metric considers the estimate accurate only if it is within 5 cents of the performance. We report frame-level accuracies without Viterbi, and separately evaluate on violin and other instruments for discussion. Since the models trained solely on violin range do not see other pitch classes, we calculate accuracies over violin range for the unseen instruments, i.e., we compute a combined accuracy from all the other instruments except violin, where the ground $f_0 \geq 190$Hz.

5.2.1 *Effect of Instrument-modeling Constraints*

In these experiments, we compare models trained on different variants of *Violin Etudes* in order to study the effectiveness of the constraints introduced in Section 4.1. Tests on the unseen URMP dataset are provided in Table 2. We see that all the three versions of *Violin Etudes* surpass the pre-trained model decisively on fine-grained accuracy RPA5 by more than 15% improvements, and increasing the number of filters from 50 to 100 in Instrument-modeling Constraint (IC) leads to marginally better results. IC also reduces the RPA50 on unseen instruments, implying that the estimator focuses excessively on the target instrument. Interestingly, one of the most conclusive results from these experiments was in the Sawtooth synthesis of the *Violin Etudes* f_0 tracks: model trained with sawtooth performs significantly better in violin compared to others, which may be due to the similarity of violin timbre to sawtooth or the effect of repertoire pitch distribution.

[4] Full CREPE model weights are taken from `https://github.com/marl/crepe/raw/models/model-full.h5.bz2`

	URMP Dataset				Bach10-mf0-synth (from train set of [8])			
	Violin		Others		Violin		Others	
	RPA50	RPA5	RPA50	RPA5	RPA50	RPA5	RPA50	RPA5
Pretrained [8]	96.4	68.3	**96.2**	68.1	98.9	89.5	99.1	**87.4**
Trained only on Violin Etudes	96.7	**84.2**	94.8	75.7	99.1	95.1	98.3	77.6
[8] finetuned on Violin Etudes	**97.0**	83.0	96.0	**77.3**	**99.2**	**95.7**	**99.2**	84.7

Table 3. Comparison of finetuning and training-from-scratch on *Violin Etudes*, tested on two instrumental datasets.

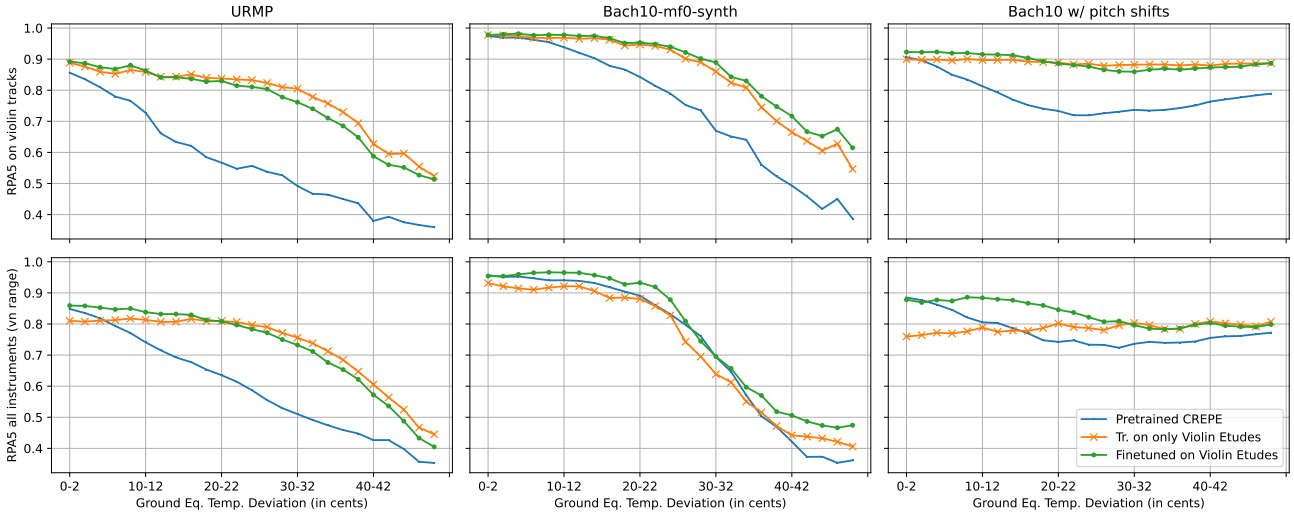

Figure 4. Fine-grained f_0 accuracy (RPA5) summarized on a grid. Music performance analysis requires precise f_0 estimates even if the player deviates from the equal temperament. However, training data-driven f_0 estimators on standard MIR datasets may induce an *auto-tuning effect*. Top: Violin-only tracks. Bottom: Over all the instruments. Left: URMP [41] dataset unseen to all the models. Center: Bach-10-mf0-synth [11, 42] from train set of [8], yet unseen to ours. Right: The conclusive experiment where we ensure the same statistics by applying microtonal pitch shifts to the Bach10 dataset.

5.2.2 *Finetuning and Generalization*

In Table 3, we study finetuning on *Violin Etudes* by testing on two datasets: alongside URMP, this time including Bach10-mf0-synth [11, 42], one of the 6 train sets of the pre-trained CREPE. We tested the estimators on the entirety of the datasets since they do not come with inherent train/test splits. We see that the finetuned model performs the best overall and remains reliable in all scenarios, including other instruments. However, interestingly, finetuning was generally more unstable and took longer to train: while the models converge after seeing around 20% of our dataset in training-from-scratch experiments, finetuning took twice as long on average. As the most interesting result of these experiments, Table 3 illustrates that our model trained on only *Violin Etudes* outperforms the pre-trained CREPE by 5.6% RPA5 on the violin tracks of its own train set.

5.2.3 *Equal Temperament and the Auto-Tuning Effect*

In Figure 4, we show that the drastic gap between the RPA5 and RPA50 performances of pre-trained CREPE can be explained by equal-temperament deviation, even on its own train data. On the other hand, our models are more robust, especially for violin-only evaluation in Figure 4. To show that this dependency is not caused by the annotation quality of the test datasets, we also experimented with microtonal pitch-shifted versions using RubberBand [5]. Yet, all our results show the importance of microtonal label distribution while training data-driven f_0 estimators. We name this phenomenon *dataset-induced auto-tuning effect*.

6. CONCLUSION

In this paper, we present *Violin Etudes*, a large-scale, comprehensive dataset for f_0 estimation and performance analysis. Our dataset curation method is easily extendable to other instruments thanks to two main novelties: 1) Reliance on pedagogy and community-driven platforms ensure abundance of material and metadata suitable for many high-level music research directions. 2) Our novel automatic f_0-labeling paradigm allows iterative refinement of labels while significantly improving the data-driven f_0 estimator as a by-product. Observing that the CREPE model converges after seeing just 20% of the *Violin Etudes* and still outperforms its pre-trained version, we argue that our dataset curation method would allow training more complex f_0 estimators that can specialize better for the needs of performance analysis. In the future, we are going to use this dataset for learning accompaniment-aware f_0 estimators and synthesizers, while further extending the *Etudes* dataset family with woodwind and brass instruments.

[5] `https://breakfastquay.com/rubberband/`

7. ACKNOWLEDGEMENTS

We would like to thank Esteban Maestre for his valuable insights on the violin resonance structure. This research was carried out under the project Musical AI - PID2019-111403GB-I00/AEI/10.13039/501100011033, funded by the Spanish Ministerio de Ciencia e Innovación and the Agencia Estatal de Investigación.

8. REFERENCES

[1] J. M. Geringer and C. K. Madsen, "Musicians' ratings of good versus bad vocal and string performances," *Journal of Research in Music Education*, vol. 46, no. 4, pp. 522–534, 1998.

[2] P. C. Greene, *Violin performance with reference to tempered, natural, and Pythagorean intonation*. University of Iowa Press, 1937.

[3] F. Loosen, "Intonation of solo violin performance with reference to equally tempered, pythagorean, and just intonations," *The Journal of the Acoustical Society of America*, vol. 93, no. 1, pp. 525–539, 1993.

[4] J. M. Geringer, "Eight artist-level violinists performing unaccompanied bach: Are there consistent tuning patterns?" *String Research Journal*, vol. 8, no. 1, pp. 51–61, 2018.

[5] J. M. Geringer, R. B. MacLeod, and J. C. Ellis, "A descriptive analysis of performance models' intonation in a recorded excerpt from suzuki violin school volume i," *String Research Journal*, vol. 4, no. 1, pp. 71–88, 2013.

[6] P. Boersma *et al.*, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," in *Proceedings of the institute of phonetic sciences*, vol. 17, no. 1193. Citeseer, 1993, pp. 97–110.

[7] M. Mauch and S. Dixon, "pyin: A fundamental frequency estimator using probabilistic threshold distributions," in *2014 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2014, pp. 659–663.

[8] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "Crepe: A convolutional representation for pitch estimation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 161–165.

[9] J. Salamon, E. Gómez, D. P. Ellis, and G. Richard, "Melody extraction from polyphonic music signals: Approaches, applications, and challenges," *IEEE Signal Processing Magazine*, vol. 31, no. 2, pp. 118–134, 2014.

[10] C.-L. Hsu and J.-S. R. Jang, "On the improvement of singing voice separation for monaural recordings using the mir-1k dataset," *IEEE transactions on audio, speech, and language processing*, vol. 18, no. 2, pp. 310–319, 2009.

[11] J. Salamon, R. M. Bittner, J. Bonada, J. J. Bosch, E. Gómez Gutiérrez, and J. P. Bello, "An analysis/synthesis framework for automatic f0 annotation of multitrack datasets," in *Hu X, Cunningham SJ, Turnbull D, Duan Z. ISMIR 2017 Proceedings of the 18th International Society for Music Information Retrieval Conference; 2017 Oct 23-27; Suzhou, China.[Suzhou]: ISMIR; 2017.* International Society for Music Information Retrieval (ISMIR), 2017.

[12] J. Dubnowski, R. Schafer, and L. Rabiner, "Real-time digital hardware pitch detector," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 1, pp. 2–8, 1976.

[13] D. Talkin and W. B. Kleijn, "A robust algorithm for pitch tracking (rapt)," *Speech coding and synthesis*, vol. 495, p. 518, 1995.

[14] M. Ross, H. Shaffer, A. Cohen, R. Freudberg, and H. Manley, "Average magnitude difference function pitch extractor," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 22, no. 5, pp. 353–362, 1974.

[15] A. De Cheveigné and H. Kawahara, "Yin, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002.

[16] A. M. Noll, "Cepstrum pitch determination," *The journal of the acoustical society of America*, vol. 41, no. 2, pp. 293–309, 1967.

[17] P. Martin, "Comparison of pitch detection by cepstrum and spectral comb analysis," in *ICASSP'82. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 7. IEEE, 1982, pp. 180–183.

[18] A. Camacho and J. G. Harris, "A sawtooth waveform inspired pitch estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 124, no. 3, pp. 1638–1652, 2008.

[19] R. C. Maher and J. W. Beauchamp, "Fundamental frequency estimation of musical signals using a two-way mismatch procedure," *The Journal of the Acoustical Society of America*, vol. 95, no. 4, pp. 2254–2263, 1994.

[20] S. Singh, R. Wang, and Y. Qiu, "Deepf0: End-to-end fundamental frequency estimation for music and speech signals," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 61–65.

[21] K. Han and D. Wang, "Neural network based pitch tracking in very noisy speech," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 12, pp. 2158–2168, 2014.

[22] B. Gfeller, C. Frank, D. Roblek, M. Sharifi, M. Tagliasacchi, and M. Velimirović, "Spice: Self-supervised pitch estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1118–1128, 2020.

[23] W. Ma, Y. Hu, and H. Huang, "Dual attention network for pitch estimation of monophonic music," *Symmetry*, vol. 13, no. 7, p. 1296, 2021.

[24] J. Engel, R. Swavely, L. H. Hantrakul, A. Roberts, and C. Hawthorne, "Self-supervised pitch detection by inverse audio synthesis," 2020.

[25] H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba, "The sound of pixels," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 570–586.

[26] H. Zhou, Z. Liu, X. Xu, P. Luo, and X. Wang, "Vision-infused deep audio inpainting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[27] J. F. Montesinos, O. Slizovskaia, and G. Haro, "Solos: A dataset for audio-visual music analysis," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2020, pp. 1–6.

[28] Q. Kong, B. Li, J. Chen, and Y. Wang, "Giantmidi-piano: A large-scale midi dataset for classical piano music," *arXiv preprint arXiv:2010.07061*, 2020.

[29] M. K. Buckles, *A structured content analysis of five contemporary etude books for the violin*. Louisiana State University and Agricultural & Mechanical College, 2003.

[30] A. R. Cutler, *Rodolphe Kreutzer's Forty-two Studies or Caprices for the Violin: Detailed analyses according to Ivan Galamian's philosophies of violin playing*. Northwestern University, 2003.

[31] M. H. Tung, *The pedagogical contributions of Rode's Caprices to violin mastery*. The University of Texas at Austin, 2001.

[32] D. Kaplunas, "Pedagogical analysis of jakob dont's 24 etudes and caprices, op. 35," Ph.D. dissertation, University of Georgia, 2008.

[33] "Violin wiki - violin etudes," Mar 2022. [Online]. Available: https://www.violinwiki.org/wiki/Violin_etudes

[34] "Violin Methods and Etudes | Violin Masterclass.com." [Online]. Available: https://www.violinmasterclass.com/p/violin-methods-and-etudes

[35] G. T. Keat, "A guide to violin etudes and studies." [Online]. Available: https://spinditty.com/learning/A-Guide-to-Violin-Etudes-and-Studies

[36] "Ecsm violin curriculum." [Online]. Available: https://www.esm.rochester.edu/community/faq/student-curriculum/violin/

[37] P. Ramoneda, N. C. Tamer, V. Eremenko, X. Serra, and M. Miron, "Score difficulty analysis for piano performance education based on fingering," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 201–205.

[38] X. Serra *et al.*, "Musical sound modeling with sinusoids plus noise," *Musical signal processing*, pp. 91–122, 1997.

[39] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, "Ddsp: Differentiable digital signal processing," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=B1x1ma4tDr

[40] E. Trentin and M. Gori, "A survey of hybrid ann/hmm models for automatic speech recognition," *Neurocomputing*, vol. 37, no. 1-4, pp. 91–126, 2001.

[41] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, "Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 522–535, 2018.

[42] Z. Duan, B. Pardo, and C. Zhang, "Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2121–2133, 2010.

[43] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "Medleydb: A multitrack dataset for annotation-intensive mir research." in *ISMIR*, vol. 14, 2014, pp. 155–160.

[44] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with wavenet autoencoders," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1068–1077.

CHECKLIST MODELS FOR IMPROVED OUTPUT FLUENCY IN PIANO FINGERING PREDICTION

Nikita Srivatsan
Carnegie Mellon University
nsrivats@cmu.edu

Taylor Berg-Kirkpatrick
UC San Diego
tberg@eng.ucsd.edu

ABSTRACT

In this work we present a new approach for the task of predicting fingerings for piano music. While prior neural approaches have often treated this as a sequence tagging problem with independent predictions, we put forward a checklist system, trained via reinforcement learning, that maintains a representation of recent predictions in addition to a hidden state, allowing it to learn soft constraints on output structure. We also demonstrate that by modifying input representations — which in prior work using neural models have often taken the form of one-hot encodings over individual keys on the piano — to encode *relative* position on the keyboard to the prior note instead, we can achieve much better performance. Additionally, we reassess the use of raw per-note labeling precision as an evaluation metric, noting that it does not adequately measure the *fluency*, i.e. human playability, of a model's output. To this end, we compare methods across several statistics which track the frequency of adjacent finger predictions that while independently reasonable would be physically challenging to perform in sequence, and implement a reinforcement learning strategy to minimize these as part of our training loss. Finally through human expert evaluation, we demonstrate significant gains in performability directly attributable to improvements with respect to these metrics.

1. INTRODUCTION

While sheet music is often very specific as to *what* notes a musician must play, it can also be vague about *how* to play them. Composers generally write notation that focuses on describing the desired musical output, but leave the specifics of how to achieve this with their instrument up to the performer, as these details are outside the scope of their role in the creative process or in some cases beyond their expertise. The piano is an instrument which requires an extensive amount of decision-making on the performer's part. In order to perform a piece of music, a pianist must either consciously or intuitively select which finger to use to play every note in the song. Notes may overlap in duration or even have simultaneous onsets. Since any key on the piano can potentially be played by any finger, there are an exponential number of ways one could perform any given piece; of these however, the majority would be uncomfortable, difficult to play at full tempo, or even physically impossible [1]. The challenge of deciding the most ergonomic fingering to use can be nontrivial for less experienced pianists, who would not yet have the ability to quickly map common musical patterns to conventional fingering strategies, and might produce more consistent and accurate performances if provided with them [2].

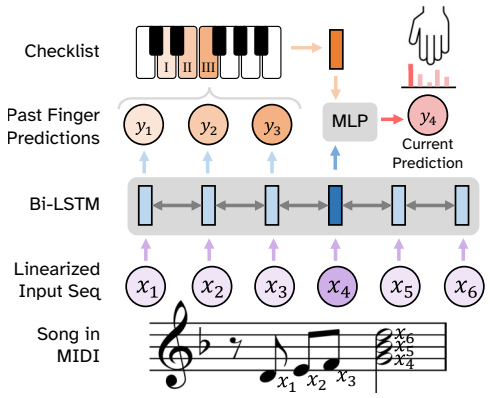

Figure 1. Visualization of the checklist model. Notes are embedded and then passed to a Bi-LSTM which outputs contextualized embeddings at each timestep. A checklist encodes where the fingers that have recently been used are located based on prior predictions. These are both fed into an MLP which predicts the next finger.

However recent work has shown that the use of machine learning techniques may be able to assist in this regard [3–6]. Automatic piano fingering prediction is the task of inferring finger assignments for each note in a symbolic representation of a piano song, given knowledge of which hand is meant to play each note. Human players generally prefer fingerings that balance physical comfort and efficiency [2], yet these criteria are difficult to quantify and highly subjective [7]. Decisions are simultaneously constrained by the current placement of the musician's fingers, and by how that decision may in turn affect possible options for the notes that follow. And while the spatial location of the notes on the keyboard is arguably the most salient factor, timing is also crucial; certain fingerings might be easy for a set of consecutive notes, but impossible if the same notes are in a chord.

Prior approaches to this task have largely treated it as a sequence tagging problem, where the input at each step is simply the pitch of the note and the output is a softmax over indices corresponding to each finger, and have employed methods analogous to those used for part of speech tagging such as LSTMs and HMMs. However, these techniques are limited in their capacity to model output dependencies, and also disregard the amount of elapsed time between notes which can greatly affect a prediction's performability.

We propose an autoregressive approach, where prior predictions are fed back into the network in order to encourage the model to produce fingerings that are not just accurate on average, but also locally *fluent* and therefore playable. We do this using a checklist which indicates which fingers have either recently been or are currently in use, and where they are on the keyboard relative to the note at the current timestep. By directly exposing this information, the model can more easily learn to restrict its predictions to fingers that are actually free, and also make decisions more directly influenced by the hand's physical placement. We also experiment with an approach that only feeds back in the most recent prediction, as well as an autoregressive tagger which maintains a neural representation of prior predictions using an encoder network.

Furthermore, the evaluation metrics used in prior work do not always correlate with what a pianist might intuitively perceive as "good" predictions. In fact there are many types of desirable (or undesirable) patterns that may emerge in a model's output, which a simplistic metric such as labeling precision may not actually reflect. We demonstrate that it is possible for two models with nearly identical labeling precision to differ dramatically in the frequency of specific types of output subsequences that are physically challenging to play. Therefore, we evaluate on a battery of metrics that collectively convey a more holistic and interpretable overview of fingering quality.

Since these metrics are not always correlated with the labeling precision that a cross entropy loss optimizes, we also investigate incorporating them explicitly into our loss function at train time. While these scores are non-differentiable, we show that reinforcement learning techniques—which have previously seen success in sequence generation tasks outside of music—can successfully optimize them. This, in conjunction with our checklist approach, ensures that the model not only has direct access to the information it would need to avoid undesirable output patterns, but is also encouraged to do so.

Finally, we conduct human evaluations and qualitative analysis which confirm that our approach improves predicted fingerings in ways consistent with our modeling choices, and also suggest directions for future work.

In summary our contributions are as follows: (1) We provide a comparison of various input representations for LSTM models (2) We put forward checklist based approaches which directly incorporate information from previous decisions (3) We introduce several additional metrics which track fluency of output, and demonstrate how to train directly on them using reinforcement learning.

2. RELATED WORK

While constraint-based models for automatic piano fingering have long existed [1], the standardization as a machine learning task which we follow was formalized by [3], which introduced the PIano fingerinG (PIG) dataset, put forward LSTM and HMM baselines (following prior work using HMMs [8, 9]), and has since been followed up on by others. [4] demonstrated the value of pretraining on even a noisy automatically annotated dataset, although they focused on basic LSTM models and evaluated exclusively on labeling precision. [5] put forward the idea of representing inputs based on relative difference in pitch rather than absolute pitch, and also showed improvements from the use of a constrained transition matrix, albeit one that was built off of prior knowledge of the task. [6] recast fingering as an information retrieval problem, although they focused on the Czerny corpus instead of the PIG dataset.

Our work also draws from research into implicitly learning output structure by feeding prior decisions into the computation of future predictions. There are classic examples of graphical models that condition predictions both on inputs at the current timestep and outputs at the previous [10]. Recent work on explicit checklists in neural settings, such as by [11] found success using a structured agenda that tracked ingredient usage in conditional cooking recipe generation. Our checklists are however more transient; outputs can be removed from the checklist if enough time has passed. We also track not just the presence of outputs, but information about how they were used.

There is also much work on the use of REINFORCE [12] for sequence generation tasks. While [13] performed several tasks including summarization, translation, and image captioning, but required a critic model for stabilization, [14] developed a self-critical method for captioning that did not require this additional network. [15] applied a similar technique to abstractive news summarization. Reinforcement learning has also been applied in a musical setting [16], including the REINFORCE algorithm specifically [17, 18], but most of these have focused on music generation rather than downstream prediction of performance-oriented attributes such as fingerings.

3. MODEL

We now describe the basic layout of the models we compare, and detail the specific variations between them. First, we will formalize the basic model contract and explain the basic architecture components that most of them share.

At a high level, each model takes in a MIDI-derived representation of either the left or right hand of a complete piano song, and outputs a predicted finger for each note. More formally, we are given an input sequence of notes $\mathbf{x} = \{x_1, ..., x_N\}$, where each $x_i = (p_i, t_i^-, t_i^+)$ is a tuple consisting of a pitch $p_i \in \{1, ..., 88\}$ as well as a corresponding onset time $t_i^- \in \mathbb{R}_{\geq 0}$ and offset time $t_i^+ > t_i^-$. From $\mathbf{x}$ we must predict a corresponding sequence $\mathbf{y} = \{y_1, ..., y_N\}$ where each $y_i \in \{1, 2, 3, 4, 5\}$ is an index representing the finger used to play x_i from thumb to pinky respectively.