MUSICALLY AWARE AUTOMATIC PIANO TRANSCRIPTION USING SYNTHETIC PRETRAINING

Louis Bradshaw, Simon Colton
Queen's College London

Alexander Spangher
Stanford University

Stella Biderman
EleutherAI

ABSTRACT

Advances in automatic music transcription systems have been driven by high-quality datasets of matched audio and MIDI. In this work, we study techniques to expand and best utilize limited training data in the context of seq-to-seq transcription models. When incorporating unsupervised data appropriately, we observe notable improvements in robustness, reflecting realistic use cases. We propose and analyze an adapted metric, useful for pruning datasets of MIDI files transcribed from audio and for evaluating music transcription models. We release transcription models for various instruments, optimized inference code, re-transcriptions of well-known expressive MIDI datasets, and a new diverse dataset comprising over 5,000 piano transcriptions.

1. INTRODUCTION

Converting sounds in audio recordings to symbolic representations is a fundamental part of interpreting, learning from and ultimately preserving the intentions of composers and performers [1]. Musicians have traditionally performed transcription manually; however the process can be tedious, and requires extensive musical training. Automatic Musical Transcription (AMT) is a way to address transcription by learning models to convert audio input to musical representations like MIDI [2].

A primary use-case of AMT systems is to build MIDI datasets from vast collections of audio recordings [3, 4]. Existing models for piano transcription generally perform well when assessed against held-out training data, typically studio recordings [5,6] where noise, reverb and other kinds of sound distortions kept to a minimum.

Although AMT models [3, 7] have achieved shown impressive performance on standardized benchmarks, these benchmarks are typically based on studio recordings [8] where noise, reverb and other kinds of sound distortions kept to a minimum. In practice, live performances and historical recordings might be of artistic interest, but differ from these pristine environments. In this paper, we develop a novel approach to AMT by drawing inspiration from human cognitive processing.

When humans transcribe music, having a working musical knowledge is incredibly helpful [9]. If, for instance, one note in an arpeggio is distorted in a recording, a human-transcriber can adhere to surrounding musical context and understand which notes are most likely in the arpeggio. However, explicitly incorporating prior *musical awareness* into models, to our knowledge, has not previously attempted in AMT. We seek to obtain music knowledge a-priori in two ways. (1) we take noise-based augmentation to the limit, forcing our models to jointly learn music *and* transcription models. (2) We take a *bootstrapping* approach to greatly expand paired transcription data available for training.

We train a new AMT model, which we call Aria-AMT, based on an encoder-decoder architecture inspired by OpenAI's Whisper [10]. Shown in Figure 1, our bootstrapping approach is as follows: we train an initial model on the MAESTRO dataset [5], and use this model to transcribe a large dataset of unlabelled data. To insure the quality of our pretraining dataset we adapt Dynamic Time Warping (DTW) [11] to assess each transcription and discard poorly transcribed pieces. DTW allows us to assess how divergent a MIDI file is from it's original audio; to our knowledge, it has not been used in piano transcription before. Then, we retrain. In theory, one could continue this process many times, greatly scaling up the dataset of transcription music available.

Our contributions are the following:

- **State-of-the-art performance** Our model performs well across the board, achieving state-of-the-art by a wide margin on standard benchmarks (MAESTRO [5], MAPS [6]). We achieve this top performance through extensive data augmentation and bootstrapped dataset expansion.

- **Novel Metrics for OOD Evaluation** We introduce DTW, as a novel transcription evaluation metric to better assess AMT performance in OOD settings and show it can be practically used in bootstrapping. We provide extensive analysis of DTW in the AMT setting, showing it correlates strongly with [8] scores and transcription quality, as judged by humans.

- **Open-Sourced, Optimized Models, New Tasks and Datasets**: we implement our models with optimzed CUDA-backends for high-throughout, achieving 50x real-time transcription speed. We use these optimized models to provide a new dataset of over 5,000 piano transcriptions and re-transcriptions of existing expressive MIDI datasets.

2. RELATED WORK

Automatic piano transcription has seen much progress since the advent of deep-learning based approaches. Seminal works include that of Sigtia et al. [12, 13], later expanded on by Hawthorne et al. [14], who demonstrated strong results using an RRN and LSTM architectures. More recently, convolutional neural networks have seen seen use. Kong et al. [15] became the defacto benchmark for accuracy, formulating a regression algorithm which enabled higher-resolution note/pedal onsets and offsets. This model was subsequently used to create datasets [3, 4] of MIDI files, which have seen use in symbolic generative modelling.

Transformers have been successful in both music transcription and speech recognition. Whisper [10], demonstrated the power of dataset scale in speech recognition, which we use as a guiding principle in this work. Hawthorne et al. [16] applied a sequence-to-sequence Transformer-based approach to piano transcription, which was extended to the multi-track AMT model MT3 [17]. hFT, a hierarchical (frequency-time) Transformer proposed by Toyama et al. [7], reports state-of-the-art results on mir_eval benchmarks.

Two main datasets have been used to train AMT models. The MAPS dataset [6] is a collection of MIDI-annotated piano recordings and was used extensively prior to the release of MAESTRO. By collaborating with the organizers of the International Piano-e-Competition, Magenta [18] released the MAESTRO dataset [5] which now serves as the predominant training/evaluation data for automatic piano transcription. Recent work [19] has investigated using alternative data sources such as synthetic data.

3. APPROACH

In this section we discuss and motivate our model architecture, training, and inference procedures as well as shedding light on techniques we have adapted for dataset curation.

3.1 Model Architecture

We choose a simple, minimalist architecture based on OpenAI's speech recognition model Whisper [10]. This architecture, shown in Figure 2. is a Transformer-based encoder-decoder [20] in which log-mel spectrograms are first embedded using two Conv1d layers before being processed by the encoder. Like in standard seq2seq transformer-based architectures [20], the encoded input-sequence embeddings (i.e. audio embeddings) act the targets for cross-attention, which are combined with representations generated via causal self-attention in each decoder blocks. We train our model on a next token prediction objective using Cross Entropy loss. This formulation of the AMT problem necessitates an autoregressive tokenization scheme representing note/pedal onsets, offsets and velocity values, which is decoded at inference time. We describe this in detail in Section 2.4.

In an effort to keep cohesion with Whisper runtimes, we generate log-mel spectrograms in a nearly identical manner: After down-sampling to 16kHz, we chunk the audio files into 30 second segments using a sliding window. We use a hop-length of 160, FFT-size of 2048, and 256 mel-bins. The resulting tensor with dimensions (3000, 256) is used as the input to the audio encoder.

3.1.1 Data Representation

Our choice of tokenization scheme follows two primary considerations. Firstly, in an attempt to reduce the internal arithmetic required to process the token sequences, we use absolute time representations for the onsets and offsets. Note-on and note-off events are represented as sequences of tokens of the form:

$$(\text{on: } n), (\text{onset: } m), (\text{velocity: } k);\ (\text{off: } n), (\text{onset: } m)$$

where n, m, k is the MIDI pitch, absolute onset (in milliseconds), and MIDI velocity respectively. In the case where the sustain-pedal is active during a note-off event, we extend the corresponding note-offset token. We order the total collection of subsequences according to their onsets/offsets. Notably, this tokenizer closely resembles that used in other encoder-decoder based AMT models [16]. An important difference is our use of separate tokens for note-on, note-off, and pedal messages. This factorisation incorporates additional information that can be used when predicting the onsets/offsets, and is slightly more compact. A secondary consideration is how amenable the tokenizer is to inference-time optimizations. Our choice of tokenizer enables a natural inference optimization, enhancing the precision of the onset and velocity values. Similar to previous approaches [15], we include tokens representing pedal-on and pedal-off events, as well as tokens which represent active note-on signals occurring prior to the current context.

3.2 Training

We design our training procedure to jointly learn two related objectives. Adopting a Bayesian perspective, one may expect that a strong prior distribution for note onsets/offsets only conditioned on context present in the symbolic music (i.e. not on the audio) may be beneficial to incorporate into a transcription model. A key observation here is that when using our chosen architecture, if the audio embeddings are constant, cross-attention is also constant and is therefore absorbed into a bias term in each decoder block. Training in our setup with constant audio embeddings is therefore mathematically equivalent to training a decoder-only model to predict the next MIDI token, identical to that of MIDI based auto-regressive generative transformer models [21, 22]. Motivated by this observation, and in an effort to simplify model inference, instead of training a separate prior model unconditioned on audio, we incorporate this idea directly into our training objective. During pretraining, we aggressively mask sections of the log-mel spectrogram before providing it as input to the audio-encoder. In sections where the spectrogram is masked, the model is forced to make its next-token predictions based primarily on the previous tokens in its context window. In practice, this encourages the model to learn to predict note and pedal onsets/offsets, both with and without a clear view of the audio. An important point is the

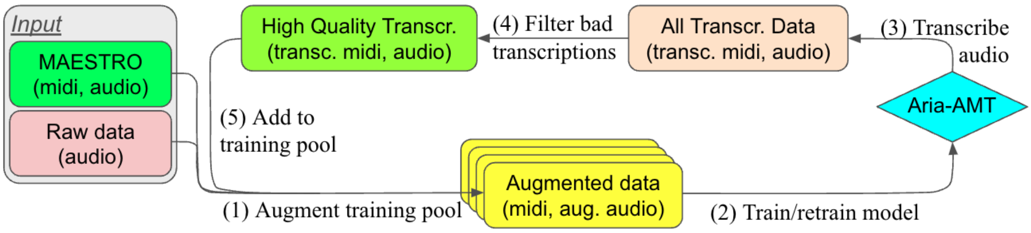
Input
MAESTRO (midi, audio)
Raw data (audio)
High Quality Transcr. (transc. midi, audio)
(4) Filter bad transcriptions
All Transcr. Data (transc. midi, audio)
(3) Transcribe audio
Aria-AMT
(5) Add to training pool
Augmented data (midi, aug. audio)
(1) Augment training pool
(2) Train/retrain model

Figure 1. **Overview of our training protocol. Step 1-2**: We start with a small transcribed corpus with (midi, audio) pairs (i.e. MAESTRO), and a large unlabeled audio corpus. We apply augmentations and train an initial model. **Step 3-4**: We use this model to transcribe the raw data, giving us (transcribed midi, audio) pairs and use dynamic time warping (DTW) to filter out low-quality transcriptions. **Step 5**: We add remaining high-quality transcriptions to the training pool and repeat.

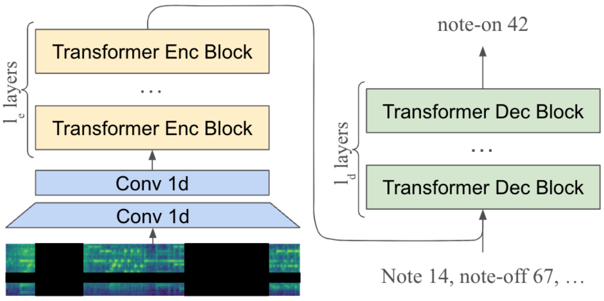

Figure 2. **Aria-AMT model architecture**. We take as input log-mel spectrograms of audio waves (bottom left) and predict MIDI (top right). Shown is Spectrogram Augmentation, one of our core data augmentations done during training. We encode with 2 conv. layers and l_e layers of Transformer encoder blocks. We decode MIDI tokens with l_d layers of Transformer decoder blocks. We train different model sizes.

function of synthetic data in such a training setup. Although of considerably less acoustic value than real audio, including additional MIDI data paired with synthesized audio supplements MAESTRO, providing additional tokens for the model to learn from when the spectrogram is masked.

We train a model with hyper-parameters corresponding to the **base** configuration of Whisper [10]. As our vocabulary size is smaller, this model has only 49M parameters. We train in bfloat16 using AdamW [23] with $\beta_1, \beta_2 = 0.9, 0.98$, $\epsilon = 1e\text{-}6$ and an L2 weight decay of 0.1. We use a linear learning rate scheduler decaying the 10% of the initial learning rate after a warmup over the first 500 optimizer steps. To help with training stability we use gradient norm clipping [24] with a cutoff of 1.0.

3.2.1 Pretraining

We split model training into two distinct stages. Initially we built a dataset of MIDI files for pretraining purposes using a bootstrapped model trained on the MAESTRO dataset [5] for 50 epochs. We describe the full data pipeline in Section 2.3. Using the virtual instrument software Pianoteq [25] we rendered the MIDI files to produce matching audio. The result is a large synthetic dataset containing over 1,000 hours of paired audio and MIDI. During pretraining, we aggressively apply spectrogram augmentation [26], randomly masking between 0 and 25 seconds of the input spectrogram. We pretrain for 50 epochs on the combined corpus consisting of our synthetic dataset as well as the MAESTRO dataset [5]. For pretraining we used an initial learning rate of 5e-4 and a batch size of 512. In preliminary experiments we observed that even a model pretrained on purely synthetic data generalized surprisingly well to some real transcription tasks

3.2.2 Finetuning

Following pretraining, we align the model for transcription by finetuning on the MAESTRO dataset alone. During this we apply light spectrogram augmentation: on average masking only 2.5 seconds of the spectrogram. We finetune for 5 epochs, with an initial learning rate of 1e-4 and a batch size of 512.

3.2.3 Data Augmentation

Not only has data augmentation been shown to improve robustness in out-of-distribution tasks [27], but also has the potential to improve standard evaluation metrics. We employ a variety of data augmentation techniques, mostly targeting common recording environments used for piano. These include adding room impulse responses (RIR) (e.g. different reverb signals), adding noise and distortion (e.g. clapping or record static), adjusting frequencies using a bandpass filter, applying pitch augmentation, and occasionally de-tuning the audio up to 15 cents. We compiled 20 different RIRs to simulate a range of different recording environments, including cylinder recorders, vintage microphones and live concert halls. We apply all data augmentation in an online manner using the torchaudio library [28].

3.3 Dataset

As well as being a popular use for transcription models [3], the ability to assemble clean datasets of transcribed MIDI files efficiently is a vital aspect of our pretraining process. To this end, we employ a technique called Dynamic Time Warping (DTW) [29] with two goals in mind. Most large corpora of audio files do not uniquely include solo-piano performances, when cleaning such datasets we therefore need a way of removing out-of-genre recordings (such as piano concertos). Ideally we would also like a measure of the *quality* of a transcription, allowing us to prune a dataset by removing outliers. DTW can be used to address both of these concerns, assigning a transcription score to an audio-MIDI pair presumed to have come from an arbitrary transcription algorithm. Concisely, we follow the following procedure given an audio-MIDI pair

1. Synthetize the transcribed MIDI into audio using a piano virtual instrument.

2. Align this synthetic audio with the original and use DTW to compute a similarity score.

DTW has been used previously to assess the similarity between pairs of recordings in large databases [29]. However, it has never been evaluated as an unsupervised transcription accuracy metric. Given that the scale of difference between two *different* songs is far greater than the scale of difference between a good/bad transcription in an audio wave, we need to evaluate, first, whether it will work on this scale (we show, in Section **??**, that it does).

Using DTW as a proxy for a transcription score, we prune our MIDI datasets to produce clean versions. During preliminary experimentation we found that DTW is especially effective at flagging previously missed multi-instrument recordings in popular MIDI datasets [1] . To build the pretraining dataset, we used a conservative DTW cutoff of 0.42 when pruning.

3.4 Inference

The standard method for supervised encoder-decoder inference in a setting such as ours [16] is to use argmax decoding: chunking the audio into segments which are decoded independently and stitched together post factum. We slightly modify this procedure as illustrated in Figure 3, using a sliding window with a stride of ten seconds to generate overlapping thirty second chunks. During inference we decode tokens in the range $10\text{s} < t < 20\text{s}$, thus providing ten seconds of audio context on either side of the transcribed region.

A key consideration is the ability to regress on note-onsets, note-offsets, and velocity values during inference. Other work [15] has shown that regressing on onsets and offsets results in higher-resolution and more accurate transcriptions. One advantage of our tokenizer design is a mathematically motivated formulation of this process. When autoregressively decoding onset tokens during inference, instead of choosing the onset with the maximal probability o_{argmax} as is standard, we instead use the formula

$$o = q\left(\mathbb{E}_P\left[X \middle| |X - o_{\text{argmax}}| \leq 30\text{ms}\right]\right)$$

where q is our tokenizer quantization function (in our case 10ms), and $P := \mathrm{S}(l)$ is the probability distribution derived from the onset's logits. Similarly for velocity values, we use the formula $v = q\left(\mathbb{E}_P\left[X\right]\right)$, omitting the tolerance condition on $|x - v_{\text{argmax}}|$. For the onsets such a condition is vital as in this case P can include onset probabilities corresponding to temporally different activations of the same pitch.

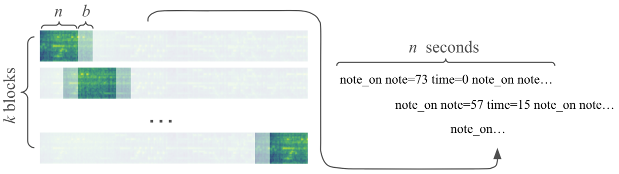
n
b
k blocks
...
n seconds
note_on note=73 time=0 note_on note…
note_on note=57 time=15 note_on note…
note_on…

Figure 3. **Buffered encoder-decoder scoring**. On the left is the audio input and on the right is the MIDI output. We take audio input blocks of length $n+2b$ seconds and output MIDI segments of length n seconds. The b-sized buffer on either size of the input is used to provide proper context to prevent the MIDI transcription from stopping or starting notes abruptly.

Using standard inference optimizations targeting CUDA backends including CUDA graphs, JIT compilation, and $\leq$8bit weight-only quantization, we achieved an inference speed of roughly 50x real-time on an RTX 4090 when transcribing a single piano recording, and up to 150x real-time (per GPU) when doing batched inference. This represents an improvement over inference implementations in other work [7, 15, 16].

4. EXPERIMENTS

We describe first our training procedure and the baselines we compared against, and then we describe the experiments we ran to evaluate them. To test the robustness of our methods, we evaluate on MAESTRO test-set accuracy. We also devise two new evaluations to test a range of different out-of-distribution recording settings, and finally, we introduce one special challenge test.

4.1 Experimental Comparisons

Aria-AMT We train Aria-AMT using bootstrapping for 1 round. We show the results of the baseline model, which was trained only with MAESTRO, and the results with *+bootstrapping*, which were trained with the round of bootstrapping we did.

Kong et. al. [15] researchers implemented an simple neural network architecture based off of convolutional layers [30] and gated recurrent units [31]. This architecture is trained on MAESTRO v2.0.0 dataset [5], which consists of 1,282 audio files, or 200 hours of music. This architecture represents a lower bound of complexity, as the neural layers are outperformed by more current layers [20]. However, the model is still commonly used in research [27]. Additionally, it is the first architecture to learn continuous note onsets and offsets with a regression model, rather than quantizing them in frames and calculating a probability per frame. We compare this method as a standard baseline in AMT, with a lower bound on model complexity and dataset size.

[1] We provide (re-)transcriptions and corresponding DTW scores for datasets including GiantMIDI [3], as well as making public a new diverse dataset compiling transcriptions of over 5,000 piano recordings obtained using spotify-dl.

hFT-Transformer [7] Hierarchical Frequency-Time Transformer represents the upper bound of model complexity. Authors implement a two-level transformer architecture where the first level applies self-attention in the frequency axis: using self attention, it compresses a log-mel spectrogram with F frequency channels into P pitches. The second level encodes information in the time domain. Authors test on the MAPS [32] and the MAESTRO v3.0.0 datasets. We compare this model as the current state-of-the-art method, with an upper bound on model complexity.
Google We include the many-to-many transcription model released by Google [].

4.2 Evaluations

MAESTRO (M): We test transcription error rates on a standard evaluation dataset, the MAESTRO dataset, which contains high-quality MIDI transcriptions and pristine audio recordings. This evaluation is a widely used test, studied by prior work [3, 7]. We evaluate the performance of models using the `mir_eval` package [8] to test precision, recall and F1 for note onsets, note offsets and pitch estimation. We additionally test generalization using two augmentations, (M-Aug-1, M-Aug-2). We generate these augmented versions by randomly applying a series of augmentations to the audio files designed to degrade the quality of recording, including: reverb, noise, static and EQ compression. **M-Aug-1, M-Aug-2** are differentiated by the likelihood of applying augmentation: M-Aug-2 has a higher level of augmentation.
MAPS: MAPS is another commonly used dataset in evaluated AMT [6]. Like MAESTRO, MAPS is also generated from disklavier pianos, which automatically transcribe performance music while it is being played.
Historic Recordings: We take a sample of notable piano music recordings from the past 50 years (1974-2024) including: Glenn Gould, Yuja Wang and Vladimir Horowitz. These recordings include a mixture of live and studio recordings. We capture a range of different recording styles, audience noise and acoustics, creating a more representative sample of the kinds of transcription that would preserve relevant moments in musical history. We evaluate the performance of our models using average DTW score.
Historic Recordings #2 (H2): We create a special challenge set to lay the groundwork for future work, and to highlight the importance of accurate transcriptions. H2 includes some of the earliest piano music recordings ever made, from 1888-1920. This set includes the earliest-known piano recording in history (Arthur Sullivan's "The Lost Chord"), as well as recordings from famous composers (e.g. Sergei Rachmaninoff and George Gershwin). We additionally evaluate the performance of our models using average DTW score.

5. RESULTS

5.1 Dynamic Time Warping (DTW) Evaluation

Our method to perform dataset expansion relies on DTW as unsupervised distance method. DTW, to our knowledge, has not yet been used in that has not yet been tested in AMT. In Figure 4, we show a comparison of DTW against human evaluations. We recruited 1 former professional musician to provide these annotations. The musician labeled a 5 if the transcription was entirely correct, a 4 if it was mostly correct, down to a 1 if it was entirely incorrect. Additionally, we use M-Aug-1 and M-Aug-2 to calculate the correlation between mir_eval scores and DTW. Table2 shows these correlations. As can be seen, DTW is highly correlated to transcription accuracy.

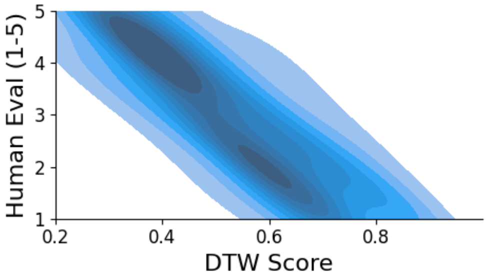

Figure 4. DTW scores and how they relate to human judgements. Human ratings rate how many audible transcription mistakes are made between 1 and 5, 1 being "unintelligible" and 5=perfect transcription. A former professional musician rated 88 transcriptions for us.

In general, we find that a score of .4 or below indicates near-perfect accuracy, while a score of .55 or above indicates serious errors. We also note that, surprisingly DTW shows robustness to recording quality. Our annotator reports that, even for recordings with high reverb, overall high correlation was observed. This could be due to the event-based nature of piano-playing, which focuses on the onsets. As can be seen in Table 2, the highest correlations to MIR_eval metrics are observed in onsets.

5.2 MIR_Eval Results

As shown in Table 1, Aria-AMI scores state-of-the-art on both MAESTRO and MAPS, outperforming previous methods by a large margin and scoring near-perfect accuracy for onset prediction. Additionally, we see that 1 round of bootstrapping greatly increasese the performance of this model.

6. DISCUSSION

The primary insight from [10] was the role of massive dataset scale in training robust, performant ASR systems. [10] built a large dataset

One common use of AMT models is in creating large-scale symbolic datasets from a corpus of audio files [**?**]. Above a certain scale, efficiency of model inference becomes increasingly important [2]. Encoder-decoder transformer models such as Whisper benefit from having relatively fast inference algorithms available, as well as a variety of open source libraries for efficient implementations.

[2] Transcribing the unpruned collection of 60,000 audio files reported in the data pipeline for GiantMIDI [**?**] could take over 1,000 GPU hours using the numbers they report for their chosen transcription algorithm

	MAESTRO			MAPS		
	F1	Onset F1	Offset F1	F1	Onset F1	Offset F1
Kong [15]	0.38 *(0.17, 0.16)*	0.97 *(0.67, 0.66)*	0.56 *(0.53, 0.55)*	0.47	0.93	0.67
Google	0.39 *(0.16, 0.14)*	0.96 *(0.63, 0.57)*	0.55 *(0.54, 0.53)*	0.23	0.67	0.54
hFT [7]	0.41 *(0.2, 0.17)*	0.96 *(0.67, 0.65)*	0.57 *(0.56, 0.56)*	0.52	0.95	0.68
Aria-AMT	0.84	0.97	0.90	0.74	0.97	0.81
+ *bootstrapping*	0.86	0.97	0.91	0.78	0.97	0.85

Table 1. **Transcription Error Rates**, calculated on the MAESTRO and MAPS datasets. F1 scores for overall, note onsets and note offsets are calculated per song, and averaged together across each dataset. Shown in parenthesis are *aug-1*, *aug-2* variations of the dataset: each variation is a different levels of augmentation we apply to test the generalization abilities of these models.

	Correlation
Human Eval.	-0.88
Onset F1	-0.66
Comb. F1, no offset	-0.65
Comb. F1	-0.55
Offset F1	-0.33

Table 2. **Usefulness of DTW as a MIDI-evaluation metric**: Spearman-f between DTW scores, human evaluations, and MIR_Eval scores on MAESTRO, M-Aug-1, M-Aug-2 and MAPS. All values significant at $p < .001$.

By keeping architectural cohesion with Whisper, it is relatively straightforward to integrate Aria-AMT into pre-existing Whisper run-times, opening up possibilities like real-time cpu inference. The only inference motivated architectural deviation from Whisper was to remove the bias term from all linear layers. This maintimes runtime compatibility whilst simplifying code needed for <=8bit weight-only quantization.

7. CONCLUSIONS

In conclusion, Aria-AMT shows impressive performance across a range of different tasks, achieving domain generalizability in a way that previous AMT models did not. We ascribe this impressive performance boost to several innovations we have introduced in this paper. First, we introduce bootstrapping as an approach to greatly increase the dataset available for transcription. This requires the use of a novel unsupervised metric, DTW, which allowed us to cut out data that threatened to degrade the quality of our training dataset. Finally, we introduced a novel and expanding data augmentation paradigm, which forced the model to learn a musical awareness along with transcription skills. Taken together, this approach gives us state-of-the-art performance across a range of evaluations.

8. REFERENCES

[1] B. Hinz, "Transcribing for greater musicality: Bob hinz explains how transcribing can be an effective tool for music students and gives some helpful suggestions for teaching it," *Music Educators Journal*, vol. 82, no. 1, pp. 25–63, 1995.

[2] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[3] 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.

[4] D. Edwards, S. Dixon, and E. Benetos, "Pijama: Piano jazz with automatic midi annotations," *Transactions of the International Society for Music Information Retrieval*, 2023.

[5] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C.-Z. A. Huang, S. Dieleman, E. Elsen, J. Engel, and D. Eck, "Enabling factorized piano music modeling and generation with the maestro dataset," *arXiv preprint arXiv:1810.12247*, 2018.

[6] V. Emiya, N. Bertin, B. David, and R. Badeau, "Maps-a piano database for multipitch estimation and automatic transcription of music," 2010.

[7] K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W.-H. Liao, and Y. Mitsufuji, "Automatic piano transcription with hierarchical frequency-time transformer," *arXiv preprint arXiv:2307.04305*, 2023.

[8] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, "Mir_eval: A transparent implementation of common mir metrics." in *ISMIR*, vol. 10, 2014, p. 2014.

[9] G. List, "The musical significance of transcription (comments on hood," musical significance")," *Ethnomusicology*, vol. 7, no. 3, pp. 193–197, 1963.

[10] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.

[11] *Dynamic Time Warping*. Springer, 2007, pp. 69–84.

[12] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 5, pp. 927–939, 2016.

[13] S. Sigtia, E. Benetos, N. Boulanger-Lewandowski, T. Weyde, A. S. d'Avila Garcez, and S. Dixon, "A hybrid recurrent neural network for music transcription," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, April 2015, pp. 2061–2065.

[14] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," *arXiv preprint arXiv:1710.11153*, 2017.

[15] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-resolution piano transcription with pedals by regressing onset and offset times," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3707–3717, 2021.

[16] C. Hawthorne, I. Simon, R. Swavely, E. Manilow, and J. Engel, "Sequence-to-sequence piano transcription with transformers," in *Proc. of the 22nd Int. Society for Music Information Retrieval Conf.*, 2021.

[17] J. Gardner, I. Simon, E. Manilow, C. G.-M. Hawthorne, and J. Engel, "Mt3: Multi-task multitrack music transcription," in *ICLR 2022*, 2022. [Online]. Available: https://arxiv.org/pdf/2111.03017.pdf

[18] TensorFlow Magenta, "Magenta: A research project exploring the role of machine learning in the process of creating art and music," 2024, accessed: 2024-04-13. [Online]. Available: https://magenta.tensorflow.org/

[19] G. Sato and T. Akama, "Annotation-free automatic music transcription with scalable synthetic data and adversarial domain confusion," *arXiv preprint arXiv:2312.10402*, 2023.

[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] 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," 2018.

[22] Y.-S. Huang and Y.-H. Yang, "Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions," in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 1180–1188.

[23] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[24] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*. Pmlr, 2013, pp. 1310–1318.

[25] Modartt, "Pianoteq - overview," https://www.modartt.com/pianoteq_overview, 2023, accessed: April 13, 2023.

[26] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[27] D. Edwards, S. Dixon, E. Benetos, A. Maezawa, and Y. Kusaka, "A data-driven analysis of robust automatic piano transcription," *IEEE Signal Processing Letters*, 2024.

[28] PyTorch, "Pytorch audio documentation," https://pytorch.org/audio/stable/index.html, 2023, accessed: April 13, 2023.

[29] C. Raffel, *Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching*. Columbia University, 2016.

[30] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[31] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[32] V. Emiya, R. Badeau, and B. David, "Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1643–1654, 2009.