Extended Abstract

Motivation Automated Piano Transcription (APT) converts raw piano audio into MIDI notes, offering broad accessibility to music transcription without manual effort. While monophonic AMT is considered solved, piano transcription remains challenging due to polyphony. Prior work shows current models often overfit to training acoustics, limiting generalization. We explore Diffusion Q-Learning (Diffusion-QL) for APT, leveraging its sequence-level sampling and robustness potential. We hypothesize that combining diffusion models with RL fine-tuning can improve performance on out-of-distribution piano recordings.

Method We adapt our approach from the Diffusion-QL framework introduced by Wang et al. (2023). First, we preprocess audio from the MAESTRO dataset into mel spectrograms, which are perceptually motivated and storage-efficient, and upload them to Amazon S3. Our model treats transcription as an offline reinforcement learning problem using a Markov Decision Process. The Diffusion-QL agent comprises an MLP, a diffusion-based actor, and a critic. The actor generates MIDI predictions by gradually denoising Gaussian noise into action sequences conditioned on the audio input. Training optimizes a hybrid loss of behavior cloning and Q-learning for improved robustness. Final outputs are piano roll arrays, which are evaluated via F1 score and converted to playable MIDI files for listening tests.

Implementation We preprocess audio from the MAESTRO dataset into mel spectrograms, reducing storage size and enabling model input. Our Diffusion-QL agent, based on Wang et al. (2023), combines an MLP, diffusion actor, and critic, modeling transcription as a conditional denoising process over diffusion timesteps. To preserve temporal information, we adapted the MLP to handle time dimensions explicitly. Training combines behavior cloning and Q-learning losses. Model outputs are postprocessed into MIDI piano rolls for evaluation.

Results We evaluated our model against the SOTA Transkun V2 baseline using standard transcription metrics (Precision, Recall, F1) via the mir_eval library. Transkun performed well on piano-like instruments but struggled with robustness on out-of-distribution audio like guitar or vocals. Toy experiments on short clips showed promising trends: reward increased and losses decreased, validating training behavior. On the full-scale run (100 songs, 10s each), Diffusion-QL underperformed quantitatively compared to baselines on all subtasks, with low (0.47) note onset F1 scores. However, training logs showed denoising progress (e.g., fewer total notes in later epochs). Qualitative tests found participants could audibly distinguish improvements in denoised outputs, suggesting perceptual gains despite weak numeric performance. This highlights the inherent potential and weaknesses of RL-driven diffusion models, though more training and tuning would be needed to understand the upper limits of their performance.

Discussion Training Diffusion-QL for audio transcription posed several challenges, including high computational demands, unreliable AWS Spot Instances, and difficulty evaluating early-stage performance. Handling the temporal dimension required adapting the model to 2D inputs, and data had to be chunked into 30-second clips, limiting full-song context. These constraints made the project ambitious for a single quarter. Future work could include exploring alternative temporal encodings, more extensive hyperparameter tuning, and longer training runs. Furthermore, finetuning models like Transkun V2 using PPO and incorporating negative or noisy training samples could lead to better generalization on non-piano instruments. Finally, refining the reward function to emphasize performance extremes may better guide learning.

Conclusion Overall, this project set out to explore Diffusion Q-Learning (Diffusion-QL) for Automated Piano Transcription, combining RL with expressive diffusion models. Despite ambitious goals, we faced significant challenges in compute, dimensionality, and data chunking that limited performance. These obstacles revealed key limitations of diffusion models in this setting and opened up new avenues for future research.

Automatic Piano Transcription with Diffusion Q-Learning

Alex Hodges
Department of Computer Science
Stanford University
alexh555@stanford.edu

Dante Danelian
Department of Computer Science
Stanford University
danelian@stanford.edu

Ramya Ayyagari
Department of Computer Science
Stanford University
ayyagari@stanford.edu

Abstract

In this work, we seek to tackle the Automatic Piano Transcription (APT) problem using a novel approach based on Diffusion Q-Learning. Specifically, we adapt this framework to a reinforcement learning environment by conditioning a diffusion model on mel spectrogram input states to generate corresponding MIDI outputs as actions. Although our results are heavily limited by computational resources and training time, preliminary findings indicate a noticeable reduction in output noise over the course of training. At the same time, these results reveal the limitations of applying diffusion models to this setting.

1 Introduction

1.1 Motivation

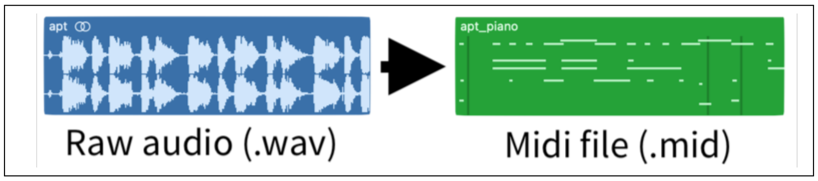

Figure 1: Automatic Piano Transcription

Automatic Music Transcription (AMT) is a longstanding task in the music field, in which a model transcribes music notes from raw audio waves. Successful AMT eliminates the need for human transcription by ear, which presents a high time and skill cost. This ultimately allows for more democratic access to high-quality transcriptions for recording and educational purposes. This domain has attracted a great body of ever-improving work, leading Benetos et al. (2019) to conclude in 2018 that monophonic (single note melodies without chords or harmonies) AMT is a solved task. Despite this, piano transcriptions remain an especially complex task, due to the highly polyphonic nature of the instrument. In this project, we focus on the subdomain of **Automated Piano Transcription (APT)**, whereby a model transcribes music notes (MIDI) from raw piano audio waves. Although there has not been extensive application of RL methods to this task, previous methods have employed a wide variety of other techniques to great success (see Section 2). However, work from Edwards et al. (2024) has demonstrated a robustness issue, whereby current models have the tendency to severely overfit to acoustic properties of the training data. This leads to under-performance on out-of-distribution piano data.

Our work aims to produce novel findings by exploring novel RL methods on the task of APT. In particular, we plan to apply Diffusion Q-Learning (Diffusion-QL), a process which combines the highly expressive nature of diffusion models with Q-learning to guide the "denoising sampling towards the optimal region in its exploration area." (Wang et al., 2023) We hypothesize that diffusion models provide a more robust approach to this problem because they sample entire audio sequences and continuously refine their output given the full audio context. In line with this, we also hypothesize that an added layer of RL fine-tuning on already well-performing can mitigate some of the robustness concerns that currently exist. Given time and compute constraints, our final project consists of incrementally training and evaluating a small-scale diffusion model on the MAESTRO dataset (further described in Related Work). Our project is novel in that we are the first to employ ensemble diffusion-RL strategies to audio transcription, and we discuss the challenges that we faced during this process.

2 Related Work

2.1 Early Work

Early work in the field has made use of a wide variety of state representations and model architectures to address this task. However, the first deep learning approaches came in 2016, when convolutional (CNN) and recurrent (RNN) neural networks were employed for estimating the probability of pitches in a given frame of audio. This approach came to be dominant in the field for several years, along with bi-directional Long Short-Term Memory (BiLSTM) techniques. (Böck and Schedl, 2012), (Sigtia et al., 2016), (Kelz et al., 2016) Then, in 2017, Hawthorne et al. (2018) introduced the onsets and frames architecture, which split the task into two stacks of BiLSTMs-based neural networks: one to detect where notes began (onsets), and another to detect every frame where a note is active (frames). This approach was able to achieve a 2X improvement over previous SOTA methods in note-onset F1 score (the common metric for comparing similarity between the generated MIDI outputs and the ground truth), raising the benchmark score from 23.14 to 50.22 on the MAPS dataset. (Hawthorne et al., 2018)

2.2 MAESTRO Dataset

Up until 2018, the predominant dataset for APT had been the MAPS dataset, made up of 18 hours of MIDI-annotated piano recordings. However, in 2018, work by Hawthorne et al. (2019) introduced the MAESTRO dataset, comprising over 200 hours of piano audio recordings and the corresponding note labels, including key strike velocities and pedal inputs. The data is segmented by piece and aligned note-for-note with $\sim$3ms accuracy. This rich collection of audio-MIDI pairs has been the used to train SOTA methods in piano transcription, and since its creation, overall performance on APT has greatly increased. For instance, more recent work has employed a musical language model (MLM) based on a bidirectional long short-term memory (BiLSTM) to achieve a note F1 score of 96.52 on the MAESTRO test set. (Wei et al., 2025)

2.3 Transformer Architectures

Like in many other tasks within the domain of machine learning, transformers have since come to be the dominant architecture in APT. Toyama et al. (2023) first introduced the *hFT-Transformer*, a two-level hierarchical frequency-time Transformer which capitalized on the self-attention mechanism of transformers in order to capture "long term dependencies in the frequency and time axes". This model was shown to achieve a near perfect F1 score of 97.43 when evaluated on the MAESTRO test set. Yan and Duan (2024) were able to approve upon this performance yet, by implementing a neural semi-Markov Conditional Random Field (*Semi-CRF*) framework to achieve a SOTA performance across all subtasks (activation, note onset, note w/ offset, notes w/offset and velocity) in terms of the F1 score. In particular, this architecture has demonstrated a note onset F1 score of 98.32 on the MAESTRO test set.(Yan and Duan, 2024)

2.4 RL Methods and Diffusion-QL

While APT has been widely explored across different methods, there has not been extensive application of RL methods to this task. As such, there are many different frameworks which warrant further experimentation with regard to the task. One novel approach is Diffusion Q-Learning (Diffusion-QL), a process introduced in Wang et al. (2023) which combines the highly expressive nature of diffusion models with Q-learning to guide the "denoising sampling towards the optimal region in its exploration area." This approach has been shown to outperform many other behavior cloning and Q-value constraint-based methods when evaluated on several different environments in the D4RL benchmark (Wang et al., 2023). Given the highly expressive capabilities of diffusion models, we aim to combine and build on the above body of work, applying Diffusion-QL methods to APT in a completely novel way.

3 Method

Note: Our code and general model approach is adapted from that of Wang et al. (2023).

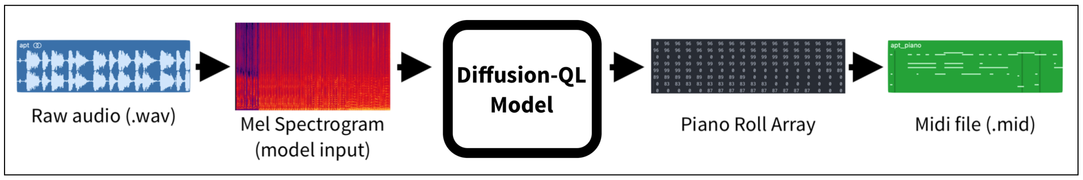

Figure 2: Full Transcription Pipeline

3.1 Storage and Preprocessing

In the broad field of audio-related work, it is commonplace to initially pre-process raw audio into a spectrogram format, so that it can be represented visually and fed into the model. In particular, the mel spectrogram is a popular format, as it employs the mel scale for frequency, which aligns more closely with human hearing perception. As such, our first step was to convert each audio (.WAV) file in the MAESTRO dataset into a Mel spectrogram to be uploaded to Amazon S3 storage. This process not only allowed for easy manipulation (truncating, padding, etc.) of our data, but it also significantly decreased the storage parameters of our massive dataset (originally >120GB).

3.2 Diffusion-QL Agent

Our Diffusion-QL class is composed of an MLP model, a Diffusion actor, and a Critic, simulating the policy class of the same name introduced in Wang et al. (2023). We model the offline environment as a Markov Decision Process (MDP), where transitions (s_t, a_t, r_t, s_{t+1}) are sampled from a preprocessed version of the MAESTRO dataset. Specifically, we convert raw audio files into mel spectrograms, which can then be fed into our model. The policy $\pi_\theta(a|s)$ is implemented as the reverse process of a conditional diffusion model where the generation of each intermediate action step $a_i, i \in [0, N]$ is conditioned on the current state s (which represents the current audio file being transcribed). In other words, the model gradually denoises Gaussian noise into a final action over N diffusion timesteps until it reaches the final action a_0, which serves as the MIDI output used for RL evaluation. During training, π_θ is optimized using both a behavior cloning loss, which encourages imitation of the dataset, and a Q-learning term that guides the model toward high-value actions, providing policy regularization.

To preserve the temporal aspect of the data, we modified the dimensional inputs of the MLP model to add an extra time dimension. In the MLP, we first flatten and stack our input vectors, which include noise, timestep of the diffusion process, and the current state. These are passed through affine linear transformations and interspersed with non-monotonic neural activation functions, which introduce non-linearity. In the future, we hope to explore how methods such as global average pooling can preserve the temporal aspects of vectors better than simply flattening them completely. Alternatively, we could explore the use of learned embeddings (via transformers) as another technique to preserve this information.

3.3 Postprocessing

On the post-processing side, the model outputs piano roll arrays of dimension $M \times T$, where M is the MIDI pitch range and T is time measured in discrete timesteps (10 ms per timestep). We use this array to compute F1 scores and convert into a playable MID file for human evaluation.

4 Experimental Setup

4.1 Dataset

MAESTRO v3.0.0. We test using the standard train/validation/test splits of the MAESTRO dataset, which contains over 200 hours of piano audio recordings and their corresponding note labels, aligned note-for-note with $\sim$3ms accuracy.

4.2 Model Specification

The key model specifications are summarized in Table 1. We trained for 5 hours on an Amazon EC2 G5.4 large instance with an NVIDIA A10G Tensor Core GPU.

Table 1: Diffusion-QL Model Specification

Hyperparameter	**Value**
Learning Rate	3×10^{-4}
ETA	1.0
Diffusion Steps	10
Batch Size	25
Files	100
Duration per File	10s
Epochs	10
Training Steps per Epoch	20
Optimizer	Adam
Input Mel Spectrogram	sr: 22050 Hz, hop: 512, window size: 1024, mels: 96

4.3 Evaluation Metrics

We calculate average precision, recall, and F1 scores for various subtasks, including activation (frame level with infinitesimal hop size), note onset, note w/ offset, and note w/ offset and velocity, all using the `mir_eval` library.(Raffel et al., 2014) As observed in the results section, some tasks are easier than others and have higher benchmarks as a result. For example, accurately transcribing just the note onsets (i.e. where each note begins) is less complex than accurately capturing the onsets, offsets, and key-strike velocities of each note.

5 Results

5.1 Transkun Baseline

To calculate our baseline, we ran Transkun V2, on the MAESTRO test set, calculating Precision, Recall, and F1 scores for various subtasks (activation, note onset, note w/ offset, note w/ offset and velocity) using the `mir_eval` library. To test for robustness, we designed a custom dataset to test the performance of Transkun on out-of-distribution single-instrument data samples, including guitar, xylophone, and glockenspiel, and vocals. Empirically, we found that Transkun performed best on keyed instruments, such as the xylophone. However, the model struggled to capture out-of-distribution effects, such as sliding notes, vibrato, and twang (guitar). These findings can guide future finetuning efforts since they demonstrate that current state-of-the-art models can't generalize robustly to non piano-based samples. While this could be a limitation of MIDI itself (due to its discrete storage of note information), training on string instrument MIDI data could help current models transfer information better from a continuous to a discrete audio space.

5.2 Toy Experiments

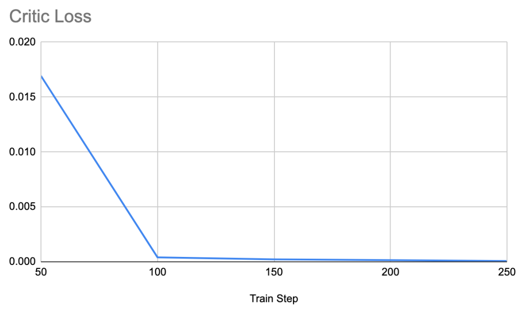

Figure 3: Toy Experiment | Critic Loss

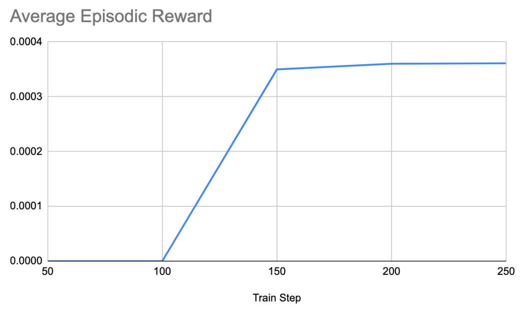

Figure 4: Toy Experiment | Average Episode Reward

Train Step	BC Loss	QL Loss	Actor Loss	Critic Loss	Avg Episodic Reward
50	1.12518	-1.5365	-0.411319	0.0169255	0
250	1.00581	-1.31069	-0.304874	6.78E-05	3.6105E-04

Table 2: Training metrics in first and last steps

To test the validity of our model on a small scale, we trained on toy experiments for a batch size of 5 on 10 songs (1 second each). Each epoch took ∼30 minutes on a Google Colab NVIDIA T4 GPU. Even at a small scale, noticed that our average episodic reward increased and our critic and BC loss decreased. Specifically, as noted in Figures 3 and 4, performance jumped around timestep 100. This toy experiment validated that our model was training as expected and paved the way for more extensive testing later. However, given diffusion models take millions of timesteps to train fully, we do recognize that this increase could be a simple fluctuation and thus further testing is required. In the next section, we explore the statistics of training our model for longer and comparing this to qualitative performance checks.

5.3 Quantitative Evaluation

For our final experiment, we trained our model using a batch size of 25 on 100 songs (10 seconds each) on an NVIDIA A10G GPU. The end results are summarized in Table 3.

During training, we tracked several metrics to ensure that the model was learning, such as BC loss, QL loss, Actor loss, Critic Loss, and average reward (F1 score). In this final experiment, we note that all losses decrease over time, as seen Figure 5. These losses decrease more significantly than those in the toy experiment, indicating that more data can decrease the loss more significantly at

Method	# Param	Note Onset			Note w/ Offset			Note w/ Offset & Vel.		
		P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)	P(%)	R(%)	F_1(%)
hFT (Toyama et al., 2023)	5.5M	**99.64**	95.44	97.44	92.52	88.69	90.53	91.43	87.67	89.48
Transkun V2 (Yan and Duan, 2024)	12.9M	99.53	**97.16**	**98.32**	**94.61**	**92.39**	**93.48**	**94.07**	**91.87**	**92.94**
Diffusion-QL	6.7M	0.27	2.77	0.47	0.26	2.71	0.46	0.10	0.97	0.17

Table 3: Transcription Result on Maestro v3.0.0 Dataset Test Set

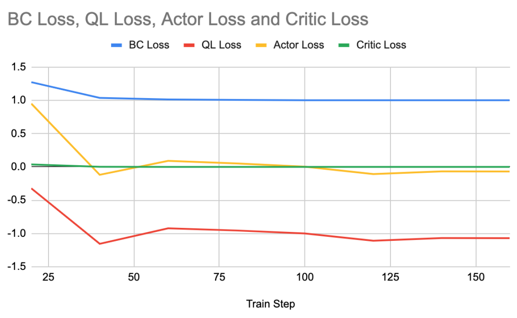

Figure 5: Final Experiment | Losses

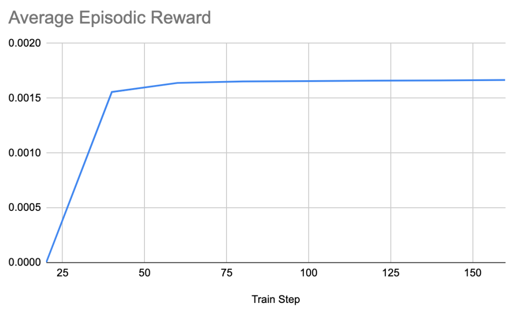

Figure 6: Final Experiment | Average Episode Reward

earlier timesteps. Because the timesteps and epochs we train on are roughly the same between the toy experiment and the final experiment, the primary change is the amount of data we were able to include (both in duration of songs and number of songs), which means this directly affects the model's training capabilities and early metrics.

In addition to these metrics, we compare and contrast the MIDI files at earlier and later train timesteps and note a pattern of decrease in the number of notes, suggesting progress in the denoising process. For instance, in Hayden's Piano Sonata in G Major, we observe the total number of notes in transcription decrease from 26,913 notes in Epoch 0 to 24,652 notes in Epoch 10. Given this significant decrease in notes, we are optimistic that running this process for a longer period of time will lead to a greater note reduction and a larger F1 reward.

5.4 Qualitative Analysis.

User Testing Experiments. While originally brainstorming this project, we designed a simple experiment to provide qualitative insights into the performance of our model as compared to the baseline. This experiment consisted of gathering participants to listen to a piece of piano audio. Afterwards, the participants were asked to listen to two MIDI transcriptions—one from our baseline, and one from our model—and rate the performance of each on a scale of one to five. Once we had trained our model, we ran this experiment on peers. However, due to the vast performance gap between the two models, the findings from these experiments did not yield novel insights.

Evaluating the Denoising Process. Still, in pursuit of insight into the model's performance, we decided to modify our experimental design. We had each participant compare the output from our model at training step 1 to the output on the final training step, to measure whether or not there was any audible improvement in transcription quality on the generated MIDI files for a randomly-chosen songs. We ran this brief qualitative study with six participants, all of whom were able to identify the denoised epoch from an initial noisy MIDI. Visually inspecting the denoised MIDI revealed that the higher frequency pitches were removed in the training process, which might explain how the melody became more clear to participants.

6 Discussion

6.1 Challenges

In implementing our Diffusion-QL model, we encountered numerous challenges which brought us to reflect on and re-evaluate the theoretical and practical advantages of our approach. We ultimately realized our project, although very interesting, was very ambitious for the quarter. Here is an overview of the challenges we encountered:

Extensive Compute. For one, we observed that diffusion models require a lot of computation to properly train, especially for tasks involving a rich medium such as audio transcription. As a result, despite using Amazon EC2 G5 Instances, we were unable to train for the significant period of time that would have been needed to yield definitive results, especially since we were training models from scratch. Given the nature of AWS Spot Instances, our training was often interrupted, so we were unable to complete runs even if we ran them as background tasks overnight.

Moreover, it was difficult evaluating model performance in intermediate steps, because qualitative improvements in MIDI files as well as quantitative improvements in F1-scores were negligible, especially considering only the first few epochs.

Dimensionality. One of the challenges of this project was determining how to adapt the state and action space to audio. Since it was important to conserve the temporal dimension, we had to adapt the Diffusion-QL model to handle 2D instead of 1D input, which required making significant changes across the model architecture.

Chunking Data. Due to compute and storage restraints, as well as our model's requirement to have a fixed state and action dimension, we had to chunk our data in intervals of thirty seconds. While this allowed us to run the model, it meant we could no longer condition our diffusion process on the entire song, sacrificing one of the advantages of our model over auto-regressive approaches.

6.2 Future Work

In the future, we could expand on this work by investigating different Diffusion QL model architecture dimensionality choices. In our current implementation, our models process explicit temporal data. We could investigate implicit ways of encoding time into our states. Furthermore, we could run extensive sweeps over hyperparameter choices (batch size, learning rate, etc.) during training and train for longer. This will require more time and compute resources, but would likely significantly boost performance. In our original project proposal, we also discussed finetuning the Transkun model. We had difficulty repurposing the gymnasium environment for audio transcription, but with more time, would hope to explore using PPO to finetune Transkun for non-piano transcription tasks. Finally, we could conduct additional tests for robustness using non-MAESTRO test sets to ensure that our model does not overfit to the acoustic properties of the training data.

Currently, we train our Diffusion-QL model on positive samples from the MAESTRO dataset. In the future, we might look into incorporating negative samples, such as audio and MIDI files that do not completely align, instead of ones that align perfectly and result in an F1 score of 1. We could augment the existing dataset this way by introducing noise into existing ground truth MIDIs and then comparing them with the ground truth to get non-perfect data for training.

Instead of the reward function being based on a vanilla F1 score, we could introduce further penalties on scores that are lower and boost scores that are higher to make incentives more extreme for the agent.

7 Conclusion

In this project, we set out to produce novel findings by exploring novel RL methods on the task of APT. We aimed to present an RL-finetuned model, as well as to build on the work of Wang et al. (2023) and experiment with a Diffusion Q-Learning (Diffusion-QL), a blend of RL-methods with highly expressive diffusion models. However, we encountered a great deal of challenges on the way, from extensive compute requirements in the training process, to difficult design decisions with regard to the dimensionality of our state/action space and chunking of our dataset that sacrificed some of the key advantages of employing a diffusion model. In the end, these challenges highlighted some of the weaknesses of diffusion models for this task, leading us to new research directions which could lead to more promising results in the future.

8 Team Contributions

- **Alex Hodges**: Preprocessing (converting wav to mel specs), model training
- **Dante Danelian**: Postproccessing (converting arrays to outputted MIDI file), report/poster
- **Ramya Ayyagari**: AWS, model training

Changes from Proposal In the original project proposal, Alex was expected to work on evaluation metrics and Dante was expected to handle data preprocessing. However, during the early stages of the project, we realized that these responsibilities were unevenly distributed over the timeline. Therefore, we reassigned preprocessing to Alex and postprocessing to Dante, allowing parallel development. Furthermore, we had originally planned to train two models: a Diffusion-QL agent and a RL-finetuned model. However, implementation and training of the Diffusion-QL model was more time and compute-intensive than originally expected, leading us to drop the second model. Finally, our original contributions did not include training the model and creating the final report and poster, so we've added them here. Overall, we believe that all team members contributed equally, whether it be debugging or writing up the final report. We did a lot of pair programming and in-person collaborative work on this project.

References

Emmanouil Benetos, Simon Dixon, Zhiyao Duan, and Sebastian Ewert. 2019. Automatic Music Transcription: An Overview. , 20-30 pages. https://doi.org/10.1109/MSP.2018.2869928

Sebastian Böck and Markus Schedl. 2012. Polyphonic piano note transcription with recurrent neural networks. , 121-124 pages. https://doi.org/10.1109/ICASSP.2012.6287832

Drew Edwards, Simon Dixon, Emmanouil Benetos, Akira Maezawa, and Yuta Kusaka. 2024. A Data-Driven Analysis of Robust Automatic Piano Transcription. , 681-685 pages. https://doi.org/10.1109/LSP.2024.3363646

Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck. 2018. Onsets and Frames: Dual-Objective Piano Transcription. arXiv:1710.11153 [cs.SD] https://arxiv.org/abs/1710.11153

Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck. 2019. Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset. arXiv:1810.12247 [cs.SD] https://arxiv.org/abs/1810.12247

Rainer Kelz, Matthias Dorfer, Filip Korzeniowski, Sebastian Böck, Andreas Arzt, and Gerhard Widmer. 2016. On the Potential of Simple Framewise Approaches to Piano Transcription. arXiv:1612.05153 [cs.SD] https://arxiv.org/abs/1612.05153

Colin Raffel, Brian McFee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel P. W. Ellis. 2014. mir_eval: A Transparent Implementation of Common MIR Metrics. https://github.com/craffel/mir_eval

Siddharth Sigtia, Emmanouil Benetos, and Simon Dixon. 2016. An End-to-End Neural Network for Polyphonic Piano Music Transcription. arXiv:1508.01774 [stat.ML] `https://arxiv.org/abs/1508.01774`

Keisuke Toyama, Taketo Akama, Yukara Ikemiya, Yuhta Takida, Wei-Hsiang Liao, and Yuki Mitsufuji. 2023. Automatic piano transcription with hierarchical frequency-time transformer.

Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. 2023. Diffusion Policies as an Expressive Policy Class for Offline Reinforcement Learning. arXiv:2208.06193 [cs.LG] `https://arxiv.org/abs/2208.06193`

Weixing Wei, Jiahao Zhao, Yulun Wu, and Kazuyoshi Yoshii. 2025. Streaming Piano Transcription Based on Consistent Onset and Offset Decoding with Sustain Pedal Detection. arXiv:2503.01362 [cs.SD] `https://arxiv.org/abs/2503.01362`

Yujia Yan and Zhiyao Duan. 2024. Scoring Time Intervals using Non-Hierarchical Transformer For Automatic Piano Transcription. arXiv:2404.09466 [cs.SD] `https://arxiv.org/abs/2404.09466`