No Data Required: Zero-Shot Domain Adaptation for Automatic Music Transcription

Andrew McLeod
Semantic Music Technologies
Fraunhofer IDMT
Ilmenau, Germany
andrew.mcleod@idmt.fraunhofer.de

***Abstract*—Automatic music transcription (AMT) takes a music recording and outputs a transcription of the underlying music. Deep learning models trained for AMT rely on large amounts of annotated training data, which are available only for some domains such as Western classical piano music. Using pre-trained models on out-of-domain inputs can lead to significantly lower performance. Fine-tuning or retraining on new target domains is expensive and relies on the presence of labeled data. In this work, we propose a method for taking a pre-trained transcription model and improving its performance on out-of-domain data without the need for any training data, requiring no fine-tuning or retraining of the original model. Our method uses the model to transcribe pitch-shifted versions of an input, aggregating the output across these versions where the original model is unsure. We take a model originally trained for piano transcription and present experiments under two domain shift scenarios: recording condition mismatch (piano with different recording setups) and instrument mismatch (guitar and choral data). We show that our method consistently improves note- and frame-based performance.**

***Index Terms*—Automatic music transcription, Zero-shot learning, Domain adaptation**

I. Introduction

Automatic music transcription (AMT) is one of the fundamental tasks in the field of Music Information Retrieval, which involves taking an input music recording and outputting a transcription of the underlying music. In this work, we imagine a simple, real-world use case. A user has access to an AMT model and would like to use it to transcribe a music collection. The user has no access to labeled data and has no way to further train or fine-tune the model—the model is treated as a black box that takes as input an audio file and outputs a transcription in some form. However, the user would like to produce the highest quality transcriptions possible. Our proposed method targets exactly this use case.

The output of AMT models comes in many forms. The most common is a piano roll, a 2D matrix where the vertical axis represents pitch and the horizontal time. The pitch axis is typically in semitone resolution while the time axis has some small resolution like 40 ms. The model outputs are floating values between 0 and 1, and are typically thresholded at 0.5 as a post-processing step to produce a binary piano roll. The Onsets & Frames [1] model extended this strategy to two piano rolls: one indicating the presence of an onset at a given pitch and time, and the other representing the presence of a note in general. They achieved state-of-the-art performance and became the de facto standard model to compare against (which we do as well). Its multiple piano roll strategy has become commonplace and was even extended to include additional offset or note velocity piano rolls [2], [3]. Post-processing involves first thresholding the outputs and then converting the resulting binary piano rolls into note events, each with an onset and an offset. Recently, a few models have been proposed that output other formats, typically to allow for token-based language models such as Transformers to be used. For example, [4] outputs a musical score directly in a text-based score format, and [5], [6] define and use a token-based output similar to MIDI, where each token represents something like a MIDI message. Still, such models remain the clear minority.

Regardless of the exact model, as with nearly all deep learning approaches, there is a reliance on large amounts of labeled training data from the same domain (instrument and recording conditions) as the test data. For example, the Onsets & Frames model trained on a large dataset of piano music suffers from a loss in performance, even when just tested on another piano dataset with a slightly different recording setup [1]. This data mismatch problem, called domain shift, is found in many fields of audio processing, and existing strategies for dealing with it in AMT are described in the Section II. For our example use case above, domain shift represents a real problem, since no training data in the target domain is available to our user, and the user has no access to retrain the model in any case.

In this paper, we describe Transcription Adaptation via Pitch Shifting (TAPS), an approach that is indeed able to improve performance given our use case above. It involves no additional data, and requires neither model retraining nor access to the model code itself to work. TAPS treats a model simply as a black box that takes as input an audio file and outputs one or more float-valued piano rolls. It works by transcribing pitch-shifted versions of the input audio, aggregating outputs where the model is unsure.

II. Related Work

Domain shift represents a real problem for AMT, especially because large datasets are only available for piano and very few other instruments. As such it has been the focus of a number of recent works. Nonetheless, we were unable to find any that would work in our example use case above. None are able to take an existing pre-trained model and improve its performance under domain shift without access to the model itself (e.g., for retraining) or additional data, although some similar work exists in Computer Vision (e.g., [7]).

Some approaches rely on a manipulation of the model's train or test data to improve generalization performance. For example, [8] thoroughly investigates the application of a variety of data augmentation techniques at train time, showing that their use can improve performance under domain shift. They point specifically to "timbral diversity" in the training set as the cause for this improvement. Meanwhile, [9] looks into various normalization and data manipulation techniques, showing that certain methods of normalization and feature projection on the training and test data can reduce the domain shift itself by making the test data more similar to the training data.

Other approaches for AMT rely on fine-tuning a model on the target domain, though on comparatively little—or at least more easily-

This research was supported by the German Research Foundation (Grant No. 350953655)

labeled—data. For example, [10] proposes pre-training a general (not for a specific instrument) model on large amounts of easy-to-produce synthesized data, then fine-tuning it on the target domain (instrument) with relatively small amounts of labeled data. [11] proposes a method—later also used by [12]—to train or fine-tune a model without relying on hand-labeled data (or at least hand-aligned labels) at all. They instead rely on coarse-grained musical scores as labels in the target domain, using the Expectation-Maximization algorithm to improve model performance in a self-supervised fashion.

Some approaches rely on no labels in the target domain, instead applying self-supervised learning to fine-tune or train a model given only unlabeled data from the target domain. For example, [13] use Virtual Adversarial Training, transcribing a noisy and a clean version of each input spectrogram and using a loss function to try to make the outputs more similar to each other. Similarly, [14] relies on cross-version consistency, using a teacher model (trained on the source domain) to label pairs of data points in the target domain that are known to have the same underlying representation (e.g., multiple performances of the same musical piece), and training a new model using matched labels. These methods still rely on additional training of the model which is not always possible, and our method avoids.

Pitch Estimation with Self-Supervised Transposition-Equivariant Objective (PESTO) [15] does not address domain adaptation explicitly, but it deserves mention nonetheless for being an inspiration for our approach. It is able to train a model from scratch using only unlabeled data. At train time, the (unlabeled) inputs are transposed by a known number of semitones, relying on the fact that this should result in a known shift in the model's output (by the same number of semitones). The loss function is designed to enforce this property, and they show that it is able to perform monophonic transcription, but do not extend the work to tests on polyphonic input.

III. Proposed Method

In this section we describe our method, Transcription Adaptation via Pitch Shifting (TAPS). Given a collection A of audio files and a transcription model that takes as input an audio file and outputs piano-roll-shaped activation matrices with values between 0 and 1, TAPS generates improved output activations in the same form. It does that through transcribing pitch-shifted versions of the input, aggregating outputs across those versions where the model is uncertain.

Note that we assume the model outputs somewhat reasonable values. If a model outputs completely (or mostly) random values, we expect TAPS to perform poorly, since it relies on those outputs. We also require that the piano roll's frequency resolution be semitones, though we intend to investigate similar methods on finer-grained outputs in future work. For simplicity, we initially describe the case where the transcription model outputs only one piano roll for an input, extending it to multiple outputs in Section III-B.

TAPS's design is similar to a mixture of experts model [16] where the "experts" are the model itself run on pitch shifted input. We pitch shifting rather than other audio data augmentations such as EQ curves and time stretching because pitch shifting forces the model to use a different output path for the same data after the shift. That is, the model's probability of a C4 being present is calculated using the same neural network nodes before and after EQ or time stretching. However, after pitch shifting, when the note becomes a C#4, a completely different calculation is performed. Nonetheless, an investigation into additional data augmentations is intended for future work.

A. Single Piano Roll

We first generate $2S$ pitch-shifted versions of each input, where S is a hyperparameter denoting the maximum pitch shift amount in semitones (it is doubled because we pitch shift each input both up and down). In our implementation, we use the Pyrubberband package[1]. Specifically, for each integer s where $-S \leq s \leq S$, we pitch shift the original audio by s semitones. After this process, we have $2(S+1)$ input files per original input, which we denote as $a^{i,s}$ (i denoting the file's index in the original collection A, and s denoting the pitch shift amount in semitones). The transcription model is run on each of these $a^{i,s}$, generating a piano roll-like probability matrix $P^{i,s}$ for each.

We then find each element of each $P^{i,0}$ where the model is unsure (only looking at the non-shifted outputs here), defined as those greater than ϵ and less than $1-\epsilon$. The set of all such indexes i, t, and f (where t and f index along the time and frequency dimensions of each piano roll, respectively) is denoted as U as in Equation 1.

$$U := \{(i,t,f) : \epsilon < P^{i,0}_{t,f} < 1-\epsilon\} \tag{1}$$

The global average unsure output across all (shifted and non-shifted) inputs is then calculated as μ_G in Equation 2[2]. Note that we sum across only those shifted and non-shifted outputs corresponding to unsure *non-shifted* outputs. The pitch shifting adds some noise, so when the non-shifted output is already sure, we simply trust it and ignore the shifted outputs entirely.

$$\mu_G = \frac{\sum_{(i,t,f)\in U} \sum_{s=-S}^{S} P^{i,s}_{t,f+s}}{2|U|(S+1)} \tag{2}$$

Finally, we calculate the final output P^i for each input audio. For all $(i,t,f) \notin U$, the value is simply copied over from $P^{i,0}$. However, unsure elements ($(i,t,f) \in U$) are changed. First, for each unsure output, the average value across all corresponding pitch-shifted outputs is calculated as $\mu(i,t,f)$ in Equation 3[2]). Then, $P^i_{t,f}$ is set as in Equation 4. If $\mu(i,t,f) < \mu_G$ (and thus TAPS estimates that outputs to be a 0), the equation linearly scales $\mu(i,t,f)$ from the range $(0, \mu_G)$ to the range $(0, 0.5)$. Likewise, values from the range $(\mu_G, 1)$ are scaled to the range $(0.5, 1)$. This allows transcription models that use 0.5 as a threshold in post-processing (as is most common) to operate that way, while downstream tasks that use the output as probabilities can also do so.

$$\mu(i,t,f) = \frac{\sum_{s=-S}^{S} P^{i,s}_{t,f+s}}{2(S+1)} \tag{3}$$

$$P^i_{t,f} := \begin{cases} \frac{\mu(i,t,f)}{2\mu_G} & \text{if } \mu(i,t,f) < \mu_G \\ \frac{1}{2} + \frac{\mu(i,t,f)-\mu_G}{2(1-\mu_G)} & \text{if } \mu(i,t,f) \geq \mu_G \end{cases} \tag{4}$$

Our decision to compare the model's outputs to μ_G rather than some simple threshold like 0.5 is designed for cases where its unsure outputs might be biased in one direction or the other. This choice is investigated further in the experiments in Section IV.

[1] https://github.com/bmcfee/pyrubberband

[2] The frequency bin $f+s$ is sometimes outside of the output range of the model. For simplicity, this is left out of our equations, but such outputs are skipped and the denominator is reduced by one for each occurrence.

B. Extending to Multiple Piano Roll Outputs

The extension of TAPS to models that produce multiple output piano rolls is straightforward. We simply treat each output piano roll type completely independently. For example, if a model outputs both onset and a frame activation piano rolls, we perform the above process twice, calculating two unsure index sets U, two global averages μ_G, and two final output piano rolls P^i—one for each type of piano roll. This allows TAPS to handle cases where the model's output piano roll of each type might be biased differently.

IV. Experiments

A. Setup

As mentioned, our method (TAPS) could be applied to any transcription model that outputs piano-roll-shaped activations of any kind, be they frame, onset, or offset. In our experiments, we have chosen to use the Onsets & Frames model [1], a CNN which outputs both frame and onset activations. To generate note-based outputs, we follow the same decoding process as in the original work. It is no longer the state-of-the-art method, but it is relatively small and simple to train compared to more recent methods, and serves our purposes well for initial experimentation. The model we used was trained on the MIDI and Audio Edited for Synchronous TRacks and Organization (MAESTRO) dataset [17], a large dataset of Western classical piano music with aligned MIDI files and (non-synthesized) audio recordings from a Yamaha disklavier. We tested TAPS on five datasets across three domain-shift scenarios:

No Domain Shift: the test split from MAESTRO [17], 178 recordings from the Yamaha Piano e-Competition identical to the transcription model's training data in terms of recording condition.

Recording Condition Shift: Midi-Aligned Piano Sounds (MAPS) [18] (the non-synthesized, music subsets) and Saarland Music Data (SMD) [19], containing recordings (60 and 50 respectively) of Western classical music from a Yamaha disklavier, but with a different recording condition as MAESTRO.

Instrument Shift: GuitarSet (GS) [20] and Daghstuhl ChoirSet (DCS) [21]. For GS, the microphone subset containing 360 recordings of acoustic guitar from a microphone from 5 genres (rock, singer-songwriter, bossa nova, jazz, and funk), including both soloing and comping. DCS contains recordings of a cappella choral music. We use the 20 Full Choir and Quartet recordings of Locus Iste and Tebe Poem taken from the stereo microphone.

As in [1], in the presence of a sustain pedal on piano, we extend the offset position of each note to either the next onset at that pitch or the end of the sustain pedal. We report note- and frame-based Precision, Recall, and F1 values from mir_eval [22] with default parameters. The note-based values are reported with an onset tolerance of 50 ms and an offset tolerance of 20% of the note's duration or 50 ms, whichever is larger. Frame-based metrics are pre-notes, where the frame activations are taken directly from the model's output.

TAPS has only two hyperparameters: the uncertainty hyperparameter ϵ and the maximum pitch shift S. We set these based on performance on the synthesized data from the SptkBGAm subset of MAPS which are not included in our evaluation. We tried ϵ values of 0.1, 0.2, 0.3, and 0.4 and found only a minimal effect. 0.2 showed the best performance by a slight margin, so we use it for all of our experiments. We tested values of S from 2 to 10, and found it to have a large effect on runtime, since the model must produce $2(S+1)$ transcriptions per input, but again only a small effect on performance, with larger values generally performing slightly better. We use $S = 8$ for our experiments, which exhibited the best performance on the subset. The effect of S is investigated in Figure 1, where we compare values from 0 to 10 on our test sets.

		Notes			Frames		
Dataset	Method	P	R	F1	P	R	F1
MAPS	O&F	81.0	76.0	78.3	85.1	75.4	79.8
	TAPS(0.5)	**86.3**	73.2	79.0	**89.3**	73.1	80.1
	TAPS(μ_G)	85.5	**78.5**	**81.7**	88.0	**77.0**	**81.9**
SMD	O&F	97.7	88.0	92.5	51.9	90.2	63.3
	TAPS(0.5)	**99.0**	85.2	91.4	**53.3**	89.7	**64.3**
	TAPS(μ_G)	98.8	**88.5**	**93.2**	52.6	**91.4**	64.2
GS	O&F	65.5	70.9	66.9	72.1	63.6	66.5
	TAPS(0.5)	**74.9**	67.6	69.4	**76.7**	61.6	67.2
	TAPS(μ_G)	69.3	**73.7**	**70.2**	75.3	**65.1**	**68.9**
DCS	O&F	14.1	22.3	17.1	69.3	37.2	48.3
	TAPS(0.5)	**21.7**	9.9	13.6	**76.0**	31.0	43.8
	TAPS(μ_G)	14.2	**31.0**	**19.3**	74.5	**41.2**	**53.0**
MAESTRO	O&F	99.3	**91.5**	**95.1**	93.8	**93.7**	**93.7**
	TAPS(0.5)	**99.7**	86.9	92.7	**94.7**	92.8	**93.7**
	TAPS(μ_G)	99.6	90.1	94.5	93.8	**93.7**	**93.7**

TABLE I

THE RESULTS OF OUR PROPOSED METHOD (TAPS) COMPARED TO THE ONSETS & FRAMES MODEL ACROSS THE TESTED DATASETS. "TAPS(0.5)" INDICATES WHERE A THRESHOLD OF 0.5 WAS USED FOR TAPS INSTEAD OF μ_G.

B. Results

Table I shows the performance of TAPS compared to the Onsets & Frames model which it takes as input. It is clear that TAPS is effective at improving transcription performance in the presence of domain shift without using any additional training data. In every test with domain shift, TAPS leads to an increase in performance across all metrics: precision, recall, and F1 for both notes and frames.

The first thing to notice is that the use of a 0.5 threshold with TAPS instead of the adaptive μ_G indeed leads to worse performance. Precision increases but at the cost of a substantial loss in recall, and consistently worse F1 (the only exception being SMD, where the extremely low starting precision plays a large role). This is a clear sign of a non-optimal threshold—in this case one that is too high—whereas TAPS using μ_G tends to see a more consistent improvement across precision, recall, and F1. This supports our hypothesis that μ_G helps to overcome any bias of the model.

We also applied the learned μ_G directly to the Onsets & Frames outputs (without the full TAPS procedure). We found that this actually lead to a significant performance decrease compared to the original 0.5 threshold, suggesting that the aggregated pitch shifted outputs are also integral the process, not just the learned threshold.

For the recording condition shift (MAPS and SMD), the original model's surprisingly low frame-based precision (and F1) on SMD stands out initially. This appears to be due to a lack of pedal information in the SMD Midi files: As mentioned, the model is trained to extend note offsets while the sustain pedal is being being held. In SMD, although the use of the sustain pedal can be heard in many of the wav files, no pedal information is present in the Midi. This leads to the model extending the note offsets much further than the ground truth annotations, and therefore to low precision.

Disregarding SMD's frame-based metrics, TAPS's performance increase under a recording condition shift appears to be roughly inversely proportional to the initial model's performance. This makes sense, because there are fewer errors to be corrected given high performance (as on SMD's note-based metrics) compared to lower performance (as on MAPS). Still, TAPS leads to an increase of roughly one point in F1 on SMD and just over three points on MAPS.

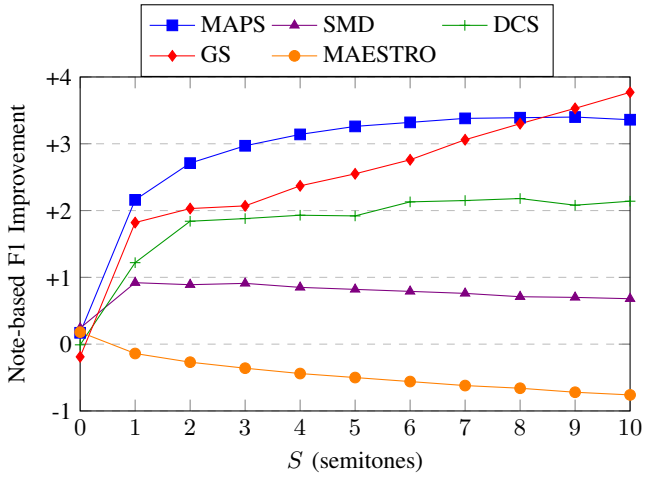

Fig. 1. TAPS's improvement in note-based F1 over the Onsets & Frames model on each dataset, varying the maximum pitch shift S.

We expected the instrument shift condition to be more difficult for the Onsets & Frames model, and that does appear to be the case. Its performance on GS and DCS are significantly lower, but TAPS is still able to increase its performance. On GS, the original model's performance is still decent since the training data's piano music (with percussive note onsets) are more similar to guitar music than DCS's choral music, which lacks any percussive onsets. On GS, where the original model's performance is slightly below that of MAPS, but not terrible, TAPS leads to an increase slightly less than that seen on MAPS, suggesting that there might be some kind of performance cutoff below which the inverse relationship seen on SMD and MAPS no longer holds. Meanwhile, on DCS, where the original model's performance is quite bad, TAPS still somewhat surprisingly leads to a modest increase performance (and a large one on the frame-based metrics). We mentioned earlier that we expected decent model performance to be a prerequisite for TAPS to work. However, it seems that decent might be too high of a bar—it may be that only better-than-random performance is required.

For the baseline without domain shift (MAESTRO), TAPS leads to roughly equivalent performance as the original model, with a slight decrease on some of the metrics. This is important, as it shows that TAPS can be applied in any test situation with only minimal decrease in performance in the worst case. For real-world applications, there will almost always be some domain shift—even if it as small as a different recording setup as in MAPS and SMD.

Finally, Figure 1 shows TAPS's increase in note-based F1 over the Onsets & Frames model with varying levels of S (the maximum pitch shift hyperparameter). The plot shows that, while most of the performance gain occurs already at quite small values of S, there is a long tail for most of the domain-shift datasets: The curves seem to indicate a roughly logarithmic growth to F1 as S increases. This makes sense, because as S grows, each additional pitch shift contributes proportionally less to the model's average output. Furthermore, as the pitch shift increases, the inputs become noisier. The main exception to this trend is GS, which exhibits a near linear improvement from $S = 3$ to $S = 10$. We were unable to find any clear reason for this property of GS by examining the outputs. In all other data sets, the value of S seems to have only a small effect on performance once it is greater than 3.

Looking into TAPS's outputs to see how it is able to improve model performance makes things more clear. First, for non-pitch-shifted inputs across our four domain-shifted datasets, an overwhelming majority of the Onsets & Frames model's outputs are within 0.2 (our ϵ) of 0 or 1 (99.4%), and an overwhelming majority of those (98.9%) are correct. TAPS will never change those outputs, which helps to avoid a large decrease in performance. In the other cases, where the model's output is instead between 0.2 and 0.8 (of which only 57.9% would be correct using a threshold of 0.5), the corresponding pitch-shifted outputs are rarely unsure themselves. Rather, 77.6% of them are within 0.2 of 0 or 1, and 74.7% of those are correct. Since these extreme values contribute more to the average output than outputs between 0.2 and 0.8, this leads to a more clear and accurate estimate in the aggregate.

The above statistics rely upon the good performance of the Onsets & Frames model. Any explanation for the model's reasoning is a question best left for the field of Explainable AI, but roughly stated: It seems that by some quirk in the model, it is sometimes unsure for particular outputs which are otherwise "obvious" (presumably these are cases which were not quite covered in the training data due to the domain shift). The pitch shifting tweaks the input enough, or passes the input through slightly different nodes in the neural network (shifted by S), that the outputs again become clear.

V. Conclusion

In this paper, we have presented Transcription Adaptation via Pitch Shifting (TAPS)[3], a method to improve automatic transcription quality in the presence of domain shift without requiring any additional model training or labeled data. TAPS works by using a transcription model to transcribe pitch-shifted versions of each input, aggregating outputs across each version where the model is uncertain. Using a transcription model originally trained on piano music, we presented experiments on four datasets with domain shift: two with a recording condition shift and two with an instrument shift (guitar and a cappella music). We showed that TAPS consistently improved performance in all datasets with domain shift across all metrics. We also showed that its decrease in performance in the absence of domain shift is minimal (even non-existent on some metrics). Given this and the rarity of encountering no domain shift in real-world scenarios, TAPS represents a promising addition to all transcription pipelines.

In future work, we intend to investigate TAPS further with a wider variety of transcription models, including some which we purposefully train non-optimally in order to investigate the effect of model quality. We will also extend TAPS to methods which output transcriptions at a more fine-grained level than the semitone, and test different settings of the uncertainty hyperparameter ϵ. We also intend to see if other data augmentations besides pitch shifting might bring a similar effect. Finally, we would like to investigate other methods of dealing with domain shift, for example different methods of data normalization which can be applied at model train time to try to reduce the performance drop of domain shift. Such methods' effect on TAPS should also be investigated.

References

[1] Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck, "Onsets and frames: Dual-objective piano transcription," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Paris, France, 2018, pp. 50–57.

[3] www.github.com/apmcleod/taps_music

[2] Qiuqiang Kong, Bochen Li, Xuchen Song, Yuan Wan, and Yuxuan 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.

[3] Andres Fernandez, "Onsets and velocities: Affordable real-time piano transcription using convolutional neural networks," in *Proceedings of the European Signal Processing Conference (EUSIPCO)*, Helsinki, Finland, 2023, pp. 151–155.

[4] María Alfaro-Contreras, Antonio Ríos-Vila, Jose J. Valero-Mas, and Jorge Calvo-Zaragoza, "A transformer approach for polyphonic audio-to-score transcription," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Seoul, Korea, 2024, pp. 706–710.

[5] Curtis Hawthorne, Ian Simon, Rigel Swavely, Ethan Manilow, and Jesse Engel, "Sequence-to-sequence piano transcription with transformers," in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, Virtual Event, 2021, pp. 246–253.

[6] Josh Gardner, Ian Simon, Ethan Manilow, Curtis Hawthorne, and Jesse Engel, "MT3: Multi-task multitrack music transcription," in *Proceedings of the International Conference on Learning Representations (ICLR)*, Virtual Event, 2022.

[7] Marvin Zhang, Sergey Levine, and Chelsea Finn, "MEMO: Test time robustness via adaptation and augmentation," in *Proceedings of the Advances in Neural Information Processing Systems Conference (NeurIPS)*, 2022, pp. 38629–38642.

[8] Drew Edwards, Simon Dixon, Emmanouil Benetos, Akira Maezawa, and Yuta Kusaka, "A data-driven analysis of robust automatic piano transcription," *IEEE Signal Processing Letters*, vol. 31, pp. 681–685, 2024.

[9] Franca Bittner, Marcel Gonzalez, Maike Richter, Hanna Lukashevich, and Jakob Abeßer, "Multi-pitch Estimation meets Microphone Mismatch: Applicability of Domain Adaptation," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, Bengaluru, India, 2022, pp. 477–484.

[10] Michał Leś and Michał Woźniak, "Transfer of knowledge among instruments in automatic music transcription," in *Proceedings of the International Conference on Artificial Intelligence and Soft Computing (ICAISC)*, Zakopane, Poland, 2023, pp. 122–133.

[11] Ben Maman and Amit H. Bermano, "Unaligned supervision for automatic music transcription in the wild," in *Proceedings of the International Conference on Machine Learning (ICML)*, Baltimore, USA, 2022, pp. 14918–14934.

[12] Xavier Riley, Drew Edwards, and Simon Dixon, "High resolution guitar transcription via domain adaptation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP)*, Seoul, Korea, 2024, pp. 1051–1055.

[13] Kin Wai Cheuk, Dorien Herremans, and Li Su, "ReconVAT: A semi-supervised automatic music transcription framework for low-resource real-world data," in *Proceedings of the ACM International Conference on Multimedia*, Virtual Event, 2021, pp. 3918–3926.

[14] Lele Liu and Christof Weiss, "Utilizing cross-version consistency for domain adaptation: A case study on music audio," in *Proceedings of the Tiny Papers Workshop @ the International Conference on Learning Representations (ICLR)*, Vienna, Austria, 2024.

[15] Alain Riou, Stefan Lattner, Gaëtan Hadjeres, and Geoffroy Peeters, "Pesto: Pitch estimation with self-supervised transposition-equivariant objective," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Milan, Italy, 2023, pp. 535–544.

[16] Saeed Masoudnia and Reza Ebrahimpour, "Mixture of experts: a literature survey," *Artificial Intelligence Review*, vol. 42, pp. 275–293, 2014.

[17] Curtis Hawthorne, Andriy Stasyuk, Adam Roberts, Ian Simon, Cheng-Zhi Anna Huang, Sander Dieleman, Erich Elsen, Jesse Engel, and Douglas Eck, "Enabling factorized piano music modeling and generation with the MAESTRO dataset," in *Proceedings of the International Conference on Learning Representations (ICLR)*, New Orleans, USA, 2019.

[18] Valentin Emiya, Nancy Bertin, Bertrand David, and Roland Badeau, "MAPS - A piano database for multipitch estimation and automatic transcription of music," Research Report inria-00544155, Telecom ParisTech, Paris, France, 2010.

[19] Meinard Müller, Verena Konz, Wolfgang Bogler, and Vlora Arifi-Müller, "Saarland music data (SMD)," in *Late-Breaking and Demo Session of the 12th International Conference on Music Information Retrieval (ISMIR)*, Miami, United States, 2011.

[20] Qingyang Xi, Rachel M. Bittner, Johan Pauwels, Xuzhou Ye, and Juan P. Bello, "GuitarSet: A dataset for guitar transcription," in *Proceedings of the International Society for Music Information Retrieval (ISMIR)*, Paris, France, 2018, pp. 453–460.

[21] Sebastian Rosenzweig, Helena Cuesta, Christof Weiß, Frank Scherbaum, Emilia Gómez, and Meinard Müller, "Dagstuhl choirset: A multitrack dataset for MIR research on choral singing," *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 3, pp. 98–110, 7 2020.

[22] Colin Raffel, Brian Mcfee, Eric J. Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, Daniel P. W. Ellis, C Colin Raffel, Brian Mcfee, and Eric J. Humphrey, "mir_eval: A transparent implementation of common mir metrics," in *Proceedings of the International Society for Information Retrieval (ISMIR)*, Taipei, Taiwan, 2014, pp. 367–372.