TISMIR

STAR Drums: A Dataset for Automatic Drum Transcription

DATASET ARTICLE

PHILIPP WEBER
CHRISTIAN UHLE
MEINARD MÜLLER
MATTHIAS LANG

*Author affiliations can be found in the back matter of this article

ubiquity press

ABSTRACT

Current state-of-the-art automatic drum transcription (ADT) algorithms make use of neural networks. To train such models, large amounts of annotated data are needed. We introduce the Separate–Tracks–Annotate–Resynthesize Drums (STAR Drums) dataset, derived from full audio recordings that include mixtures of drum instruments, melodic instruments, and vocals. First, we separate the music recordings into a drum stem and a non-drum stem by applying a music source separation algorithm, then automatically annotate the drum stem with an ADT algorithm. The annotations are used for the re-synthesis of the drum stem using sample-based virtual drum instruments. Finally, we mix the re-synthesized drum stem with the original non-drum stem to obtain the final mix. In summary, STAR Drums includes annotated synthesized drum sounds mixed with real recordings of melodic instruments and vocals, offering several benefits: high temporal accuracy of annotations; training data that include recordings of instruments played by musicians, rather than solely relying on MIDI-rendered audio; a large number of supported drum classes; the possibility to customize the final mix by, for instance, applying additional processing to the drum stem, as both drum and non-drum stems are provided; and suitable licenses of audio files for making the dataset fully available to the research community. We demonstrate that, in the context of ADT, training with STAR Drums achieves superior performance compared to training with datasets solely relying on MIDI-rendered data and that the synthesized nature of the drum stem does not diminish performance.

CORRESPONDING AUTHOR:
Philipp Weber
Fraunhofer Institute for Integrated Circuits (IIS), Erlangen, Germany
philipp.weber@iis.fraunhofer.de

KEYWORDS:
Automatic drum transcription, automatic music transcription, dataset, audio

TO CITE THIS ARTICLE:
Weber, P., Uhle, C., Müller, M., & Lang, M. (2025). STAR Drums: A Dataset for Automatic Drum Transcription. *Transactions of the International Society for Music Information Retrieval, 8*(1), 248–264.
DOI: https://doi.org/10.5334/tismir.244

1 INTRODUCTION

Automatic drum transcription (ADT) as a sub-field of automatic music transcription (AMT) aims to identify and classify drum sounds. The most challenging task within ADT is drum transcription in the presence of melodic instruments (DTM) in contrast to drum transcription of drum-only recordings (DTD), where no interfering instruments, including guitar, piano, and vocals, are present. In recent years, DNN-based algorithms have outperformed other traditional techniques such as non-negative matrix factorization, hidden Markov model, or support vector machines, particularly in addressing the complexities of DTM (Wu et al., 2018).

For training neural networks, large amounts of data are essential to obtain models that generalize well to unseen data, whereas a smaller quantity of high-quality, representative data may suffice for testing. For ADT, audio data with reference annotations of drum sounds are required. Additionally, for training models for DTM, the presence of melodic instruments during training is desirable to obtain a similar data distribution during training and testing. While manual labeling is possible for small test datasets, it is hardly feasible when creating large ADT training datasets. Annotations have to include all drum instruments, which are to be transcribed with high temporal accuracy.

As our main contribution, we present the Separate–Tracks–Annotate–Resynthesize Drums (STAR Drums) dataset, created from full audio recordings that include drum recordings along with recordings of melodic instruments and vocals. An overview of the dataset creation is shown in Figure 1 and explained in detail in Section 3. We split the full mixture recording into the drum stem and the non-drum stem by the use of a music source separation (MSS) algorithm. Then, an estimated annotation is obtained from the separated drum stem by applying an ADT algorithm. Finally, we create a re-synthesized drum stem by rendering the estimated annotation with virtual drum instruments and mix this stem with the original non-drum stem. In other words, we replace the original drum stem with a synthesized version for which the annotations are known.

With STAR Drums, we address the existing gap in ADT datasets by providing a solution for improved performance across a high number of classes. For more than five classes, this dataset offers enhanced results compared to existing datasets that rely entirely on MIDI-rendered audio. Even with fewer classes, STAR Drums achieves comparable performance to existing datasets without MIDI-rendered sounds and provides more flexibility by offering both drum and non-drum stems. By re-synthesizing the drum stems, we eliminate the need for manual labeling, which carries the risk of annotation errors and is challenging to scale for a large number of classes. This serves as a central motivation for our approach. Furthermore, STAR Drums uses music with licenses that permit the distribution of raw audio data. We demonstrate that the use of re-synthesized drum stems can be omitted, if necessary, by training with

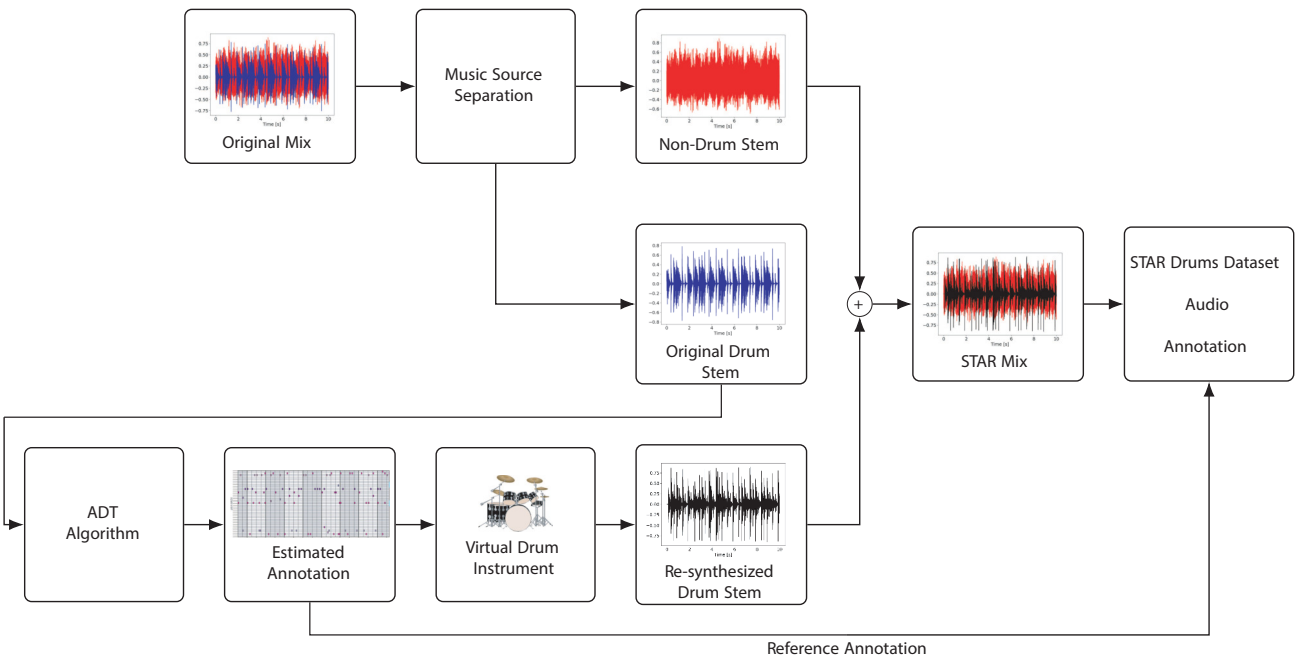

Figure 1 Overview of the STAR Drums creation process. The non-drum stem and the original drum stem are obtained from the original mix by using an MSS algorithm. An ADT algorithm creates an estimated annotation from the original drum stem, which is then used to render the re-synthesized drum stem by using virtual drum instruments. The re-synthesized drum stem is mixed with the non-drum stem, resulting in the STAR mix, which forms the STAR Drums dataset together with the estimated annotation, which is now regarded as the reference annotation.

original drum stems using pseudo-labeling, while still maintaining high performance.

The remainder of the paper is structured as follows. Section 2 reviews related datasets. In Section 3, we describe the dataset creation, its characteristics, and limitations in detail. Section 4 outlines how the proposed dataset can be applied to ADT and compares the transcription performance of models trained with STAR Drums to models trained with other datasets. Finally, we conclude the paper in Section 5.

2 RELATED DATASETS

In the recent years, different approaches have been taken to address the challenge of generating large amounts of realistic training data. Table 1 provides an overview of commonly used datasets for DTM, all of which include drum sounds along with sounds of melodic instruments. For the sake of brevity, we do not discuss datasets intended for DTD, such as the Groove MIDI Dataset (Gillick et al., 2019), the StemGMD dataset (Mezza et al., 2024), or IDMT-SMT-Drums (Dittmar and Gärtner, 2014).

From here on, we will refer to audio rendered from MIDI files as *synthesized* audio. In contrast, we will denote audio files consisting of recordings of instruments as *recorded* audio.

ADT datasets commonly used for testing are MDB Drums (Southall et al., 2017), ENST Drums (Gillet and Richard, 2006), and RBMA13 (Vogl et al., 2017). All three datasets are manually annotated, support a large number of classes, and consist of drum recordings and recordings of melodic instruments. The authors of MDB Drums and RBMA13 used already-existing recordings, which were manually annotated. In the case of MDB Drums, the audio files were taken from the MedleyDB dataset (Bittner et al., 2014). The authors of RBMA13 used tracks created during the Red Bull Music Academy 2013.[1] In contrast, for ENST Drums, three different drummers were specifically recorded while playing both solo and along with backing tracks. Some backing tracks contain synthesized audio. The length of the datasets ranges from 0.37–1.9 hours. While MDB Drums and RBMA13 include recordings of vocals, this is not the case for ENST Drums. Even though those three datasets can be used for training deep neural networks (DNNs), more data are needed to obtain well-generalizing models.

Vogl et al. (2018) created an ADT dataset by rendering a large number of freely accessible MIDI files, resulting in 257 hours of synthesized data.[2] This dataset is commonly known as Towards Multi-Instrumental Drum Transcription (TMIDT), derived from the title of the corresponding publication. We will also use this terminology throughout our paper. The MIDI files were split into a non-drum track and a drum track. The authors used a single set of MIDI samples covering all included instruments for rendering the non-drum tracks and 57 different drum sample sets for rendering the drum tracks. TMIDT includes complete audio mixtures, the isolated drum stems, and annotations for 18 drum classes.

The Slakh dataset by Manilow et al. (2019) is created similarly to TMIDT and relies solely on synthesized audio for drums and melodic instruments but uses more realistic-sounding virtual instruments than TMIDT. The MIDI files were taken from Lakh (Raffel, 2016). Slakh is intended for MSS and is not explicitly aimed at ADT. Consequently, the dataset provides mixtures of all instruments, single instrument stems, and MIDI files per stem but no ready-to-use drum annotations. Annotations can be derived from the MIDI files, as the MIDI pitch information determines which drum sample is triggered.

Dataset	Non-drum Instr.	Drums	Vocals	Melodic Instr.	# Drum Classes	Len. [h]
RWC Music Database (Goto et al., 2002)	Rec.	Rec.	Yes	Yes	29	18.1
ENST Drums (Gillet and Richard, 2006)	Rec.	Rec. & Synth.	No	Yes	20	1.0
MDB Drums (Southall et al., 2017)	Rec.	Rec.	Yes	Yes	20	0.4
RBMA13 (Vogl et al., 2017)	Rec.	Rec.	Yes	Yes	23	1.9
TMIDT (Vogl et al., 2018)	Synth.	Synth.	No	Yes	18	257.1
Slakh (Manilow et al., 2019)	Synth.	Synth.	No	Yes	—	118.3
A2MD (Wei et al., 2021)	Rec.	Rec.	Yes	Yes	3	34.5
ADTOF-RGW (Zehren et al., 2021)	Rec.	Rec.	Yes	Yes	5	89.2
ADTOF-YT (Zehren et al., 2023)	Rec.	Rec.	Yes	Yes	5	202.2
Proposed STAR Drums	Rec.	Synth.	Yes	Yes	18	124.5

Table 1 Overview of available ADT datasets.

In Slakh, no mapping from MIDI notes to drum classes is provided. Therefore, the number of supported classes depends on the mapping created by the user.

However, the MIDI notes associated with specific drum classes are inconsistent across the virtual drum kits used. Therefore, to use Slakh for ADT, users must create individual mapping tables from MIDI notes to drum classes for each drum kit. The granularity of this mapping limits the number of supported classes.

Zehren et al. (2024) experimented with a synthetic dataset creation approach aimed at minimizing the transfer gap from synthetic training data to real test data. In contrast to Slakh and TMIDT, the authors used MIDI files recorded by musicians on electronic instruments, which therefore contain small variations and are not fully quantized. All audio was synthesized using virtual instruments. Using the same test data, their method demonstrated a smaller transfer gap compared to training with Slakh. However, the resulting dataset is not available online.

Wei et al. (2021) created the A2MD dataset by using an audio-to-MIDI alignment technique to align MIDI files from the Lakh MIDI dataset with real-world audio data followed by an automated data inspection step. This results in a dataset with a total duration of 35 hours. The dataset supports three drum classes: kick drum, snare drum, and hi-hat. Splits with different alignment levels (low, medium, and high), reflecting the quality of the time alignment, are offered. The authors obtained the splits by categorizing the tracks according to a penalty value obtained during the audio-to-MIDI alignment. Lower penalty values lead to a higher estimated annotation quality.

In Zehren et al. (2021) and Zehren et al. (2023), crowd-sourced annotations from rhythm video games are used to generate the Automatic Drums Transcription On Fire (ADTOF) dataset. A two-stage cleansing procedure improves temporal accuracy of the annotations and maps them to the used drum classes: kick drum, snare drum, hi-hat, cymbals, and tom. The two versions, ADTOF-RGW and ADTOF-YT, use data from different sources but follow the same creation pipeline. For both ADTOF datasets, only pre-computed spectrograms are published instead of the raw audio files, limiting their usage to scenarios where the same short-time Fourier-transform (STFT) parameters and number of frequency bands are applicable.

The approaches used in A2MD and ADTOF rely exclusively on recordings without the use of virtual instruments. This results in training data with a distribution that closely matches the data expected at test time, as the test data typically consist of recordings of real instruments rather than synthetic data. The drawback is that the annotations are not guaranteed to be error-free and have limited temporal accuracy. For ADTOF, temporal inaccuracies of around 50 ms were reported in the raw data, which are then minimized during the alignment steps. In A2MD, the authors do not provide absolute expected time deviations for the three alignment levels. Furthermore, the number of supported classes in A2MD and ADTOF (three and five, respectively) is low compared to other datasets.

For completeness, we mention the Real-World Computing (RWC) Music Database (Goto et al., 2002), which contains recorded music with manual annotations across up to 29 drum classes. RWC consists of various datasets featuring recordings of the genres pop, classic, jazz, and datasets with mixed genres. For calculating the duration in Table 1, we excluded the dataset containing classical items. Ishizuka et al. (2020) utilized the dataset containing popular music for ADT and trained a model transcribing three drum classes: bass drum, snare drum, and hi-hat. The authors identified 89 tracks containing drum sounds and claim that 65 tracks have correct annotations, which were used in their experiments. Notably, the dataset cannot be downloaded and is only available via physical distribution media.

Lastly, Jacques and Roebel (2019) used the dataset from the MIREX 2018 ADT challenge, which includes MDB Drums, RBMA13, two datasets from the MIREX 2005 ADT challenge, and synthesized drum sounds without melodic instruments. Jacques and Roebel (2019) utilized a total of three hours from the dataset and expanded the data volume through data augmentation to transcribe the classes: bass drum, snare drum, and hi-hat. This dataset was provided to participants in the challenge but is not accessible online.

A similar method to that used in this paper was applied for f_0 estimation by Salamon et al. (2017). To create large amounts of training data for an f_0 estimator, the authors analyzed isolated stems of a multitrack recording using a monophonic pitch tracker, followed by a wide-band sinusoidal modeling algorithm to estimate the harmonic parameters. The estimated information is then used to re-synthesize the single stems, which are mixed in a final step. Similar to the approach described here, replacing the data to be annotated with re-synthesized data ensures error-free and perfectly time-aligned annotations and eliminates the need for manual labeling. However, unlike the method used by Salamon et al. (2017), we only re-synthesize a portion of the data, as annotations for instruments other than drums are not required for ADT. Additionally, by using MSS, we can extend our approach to data which are not available as multitrack recordings.

The dataset creation approach for STAR Drums, which involves replacing the original drum stem with a re-synthesized version based on labels generated by an ADT algorithm, was initially sketched by Weber et al. (2024). In this work, we refine the approach by implementing a loudness-based velocity computation, using 20 instead of 15 virtual drum kits, and providing a detailed description of the creation pipeline for reproducibility. Additionally, we leverage copyright-free music to ensure that the STAR Drums dataset remains accessible to the research

community, and we offer code for further modifications of the dataset. We also investigate and discuss the limitations of the STAR Drums approach and conduct experiments to compare the performance of models trained on STAR Drums with those trained on other available ADT datasets using various class vocabularies.

3 STAR DATASET

3.1 CREATION

We now describe the STAR Drums creation method in detail, as shown in Figure 1. The approach was already sketched by Weber et al. (2024) and is similar in spirit to that of Salamon et al. (2017). Our approach combines the benefits of eliminating manual labeling effort, thereby avoiding annotation errors, while retaining the use of real recordings and vocals. This is achieved by using synthetic audio exclusively for the drum stem.

3.1.1 Separation of tracks

The input data must be provided as two stems: the drum stem, containing only drum sounds, and the non-drum stem, containing the recordings of all melodic instruments and vocals. The data originate either from the MUSDB18 dataset (Rafii et al., 2017), which already provides separate instrument stems, or by applying an MSS algorithm to complete mixture recordings.

We selected the hybrid transformer Demucs MSS algorithm (Rouard et al., 2023). The algorithm separates a mixture recording into four stems: drums, bass, vocals, and other. We use the implementation provided in the python package `demucs` and the model identified as `htdemucs_ft`. Among the models available in the `demucs` package, `htdemucs_ft` is the best-performing one. For the test split of MUSDB18, the authors report a signal-to-distortion ratio of 10.08 dB on drums.

According to Fabbro et al. (2024), who summarize the Sound Demixing Challenge 2023, several top-performing teams used models from Rouard et al. (2023), for instance as part of a network ensemble or as basis for further training, underscoring its promising performance.

3.1.2 Annotation

In the next step, we create an estimated annotation by analyzing the separated drum stem with an ADT algorithm. We use the convolutional recurrent neural network model for 18 classes provided by Vogl et al. (2018), as it is, to our knowledge, the only well-documented published algorithm supporting 18 classes. We apply a threshold value of 0.1, as recommended by Vogl et al. (2018), who found that threshold values between 0.1 and 0.2 yield the best results. Using values at the lower end of this range results in a higher number of detected onsets, providing more training examples.

We perform the annotation on source-separated drum stems, which simplifies the ADT task. Therefore, we expect the transcription performance to be higher than the values reported by Vogl et al. (2018) for DTM. The influence of source separation artifacts on transcription performance is investigated in Section 3.8. In Section 4.2.4, we conduct additional experiments using the estimated annotations directly as ground truth for the original mix, employing the original mix instead of the mix with re-synthesized drums during training. The positive results further indicate the effective performance of the chosen algorithm on source-separated drum stems.

3.1.3 Re-Synthesis

The obtained annotations provide onset times for the individual drum instruments. Additionally, velocity information is required to convert the annotations into MIDI files. The MIDI velocity parameter ranges from 1–127 and determines the dynamic level of the current note. As discussed by Dannenberg (2006), there is neither a specification in the MIDI standard nor a common consensus on which velocity value corresponds to a specific amplitude or loudness. Consequently, the mapping from a technical parameter to a perceptual value lies at the intersection of technical work and artistic interpretation. We apply heuristics to perform this mapping in a way that produces plausible musical data.

We compute a short-term loudness of the original drum stem similarly as described in the recommendation ITU-R BS1770-4 (2015). Using the proposed frequency weighting, we calculate the signal energy and loudness for blocks of 100 ms with a step-size of 10 ms, corresponding to the time resolution of the ADT algorithm used. This provides a distinct loudness value for every possible onset position. We normalize the loudness within a track to a range between 0 and 1. We then apply an exponential mapping so that most onsets have a velocity around 105, clipping the velocity values at a lower bound of 40. Figure 2 shows the obtained velocity distribution of all notes in all tracks.

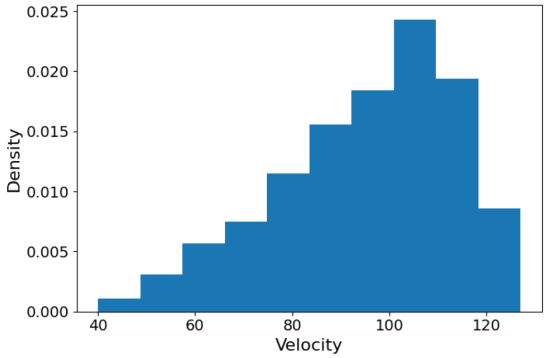

Figure 2 MIDI note velocity distribution of all drum classes and all tracks.

Informal listening tests showed that the obtained velocity values result in a musical outcome. However, the described heuristic should be regarded merely as a tool for obtaining musically plausible data; while it may potentially increase the performance of trained models, our experience indicates that it is not a critical step in the dataset-creation process.

Using the obtained velocity values, we convert the estimated annotation into a MIDI file and render it using virtual drum kits to create a re-synthesized drum stem. The estimated annotation is now treated as the reference annotation as it perfectly matches the re-synthesized drum stem, which is included in the STAR mix. To account for the fact that the input files were produced using different drum kits, we use different virtual drum kits for the re-synthesis of the drum tracks. To preserve the overall sound impression, we use five different electronic drum kits for the tracks of genre electronic—see Section 3.5. For all other tracks, 15 acoustic virtual drum kits are used. All utilized drum kits originate from the software bundle *Komplete 13 Ultimate* by *Native Instruments*.[3] The names of the used drum kits are included in the filenames of the re-synthesized drum stems.

The chosen drum kits offer a wide variety of drum sounds from different genres and reflect the sound characteristics of popular music of different decades. While more diversity is always desirable, we restrict the number of drum kits to 20 to limit the overall complexity of the dataset-creation process.

The MIDI pitch information determines which sample is triggered, necessitating the creation of mapping tables from MIDI notes to drum classes. The virtual drum instruments used vary in the number of toms and cymbals. Furthermore, depending on their sound characteristics, they may offer a different range of playing techniques for a certain class, such as snare drum sounds produced with brushes in addition to sticks in the *AR50s* and *AR60s* kits. Consequently, the MIDI notes associated with a particular class differ between virtual drum kits, requiring a separate mapping table for each kit.

If the number of drum instruments of the virtual instrument does not match the class vocabulary described in Section 3.6, we summarize multiple instruments into one class or assign a single instrument to several classes. For each drum sound onset to be rendered, we randomly choose one of the MIDI notes associated with the class, ensuring slight variations in sound nuances.

To make the final mix more similar to commercially produced music, we apply dynamic range compression to the re-synthesized drum stem using a multiband compressor and a single preset for all files. The re-synthesized drum stems are peak normalized before compression to prevent clipping and to obtain a similar amount of compression for every track. From our experience, the dynamic range compression parameters are not crucial, but applying no dynamic range compression results in mixes with noticeably higher dynamics compared to the original mix. By providing the non-drum and re-synthesized drum stems separately, we encourage users to customize the mix to their needs and to experiment further with augmenting the drum stem if desired. We scale the new drum stem such that its loudness matches the one of the original drum stem to obtain a mix with similar loudness ratios in comparison to the original mix. Finally, we mix the new drum stem and the non-drum stem, resulting in the STAR mix—see Figure 1.

3.2 INPUT DATA

We use audio data from three sources:

1. MUSDB18 (Rafii et al., 2017)
2. ISMIR2004 Genre dataset (ISMIR04) (Cano et al., 2006)
3. MTG-Jamendo dataset (MTG-Jamendo) (Bogdanov et al., 2019)

Table 2 lists the datasets and summarizes their data usage. We remove all classical tracks from ISMIR04 since it does not contain drum sounds typically used in popular music. The audio in MTG-Jamendo is provided with metadata, such as genre and instrument activity, which allows us to filter the tracks. We select all tracks including acoustic or electronic drum sounds and exclude any tracks where the license does not permit the publication of remixed versions.

To limit the size of the whole dataset while still offering diverse training data, we randomly select a 60-second excerpt from each ISMIR04 and MTG-Jamendo track.

Dataset	Instrument stems provided	# Total Tracks	# Used Tracks	Len. Used Tracks [h]
MUSDB18 (Rafii et al., 2017)	Yes	150	150	9.8
ISMIR04 (Cano et al., 2006)	No	2000	1228	98.4
MTG-Jamendo (Bogdanov et al., 2019)	No	55000	4807	302.9

Table 2 Input data for STAR Drums.

In our experiments, this approach results in a similar performance compared to training with full tracks but reduces the total duration of the training split from 401.3 hours to 114.7 hours. We utilize the full length of all MUSDB18 tracks to achieve reasonable time duration ratios across the different splits.

3.3 SPLITS

The tracks from MUSDB18 are already provided as stems, so no MSS needs to be applied. Consequently, there is no risk of audible drum sound residuals in the non-drum stems caused by imperfections in the MSS algorithm. Therefore, we use this data for validation and testing. The remaining ISMIR04 and MTG-Jamendo tracks form the training split. Table 3 provides an overview of the defined splits.

The licenses of 48 of the 150 tracks of MUSDB18 allow for the redistribution of remixed versions. For the remaining 102 tracks, we publish the new drum stem along with a Python script that downloads MUSDB18 and mixes the new drum stems with the other instrument stems. The extent of these data is indicated with values in brackets in Table 3. We emphasize that the STAR Drums dataset can be used without executing the mixing script, as the entire training set and a portion of the validation and test splits are directly provided.

3.4 DATA ORGANIZATION AND STRUCTURE

In the STAR Drums dataset, all audio files are sampled at 48 kHz and provided in lossless FLAC format with two channels. For each track, we supply a text file containing the annotations, where each onset is described in a single line. Each line begins with the onset time in seconds, followed by the class according to the 18-class mapping in Table 4, and the MIDI velocity value.

The folder structure is organized into directories for the training, validation, and test split, each containing sub-directories for audio, annotations, and MIDI files used to create the re-synthesized drum stems. Within the audio folders, sub-directories are available for the individual stems, as described in Figure 1. For the training split, additional sub-directories are provided for items originating from ISMIR04 and MTG-Jamendo.

While we retain the original filenames for MUSDB18 and MTG-Jamendo, we replaced the filenames of the ISMIR04 tracks with numbers due to excessively long file names in the original dataset. A lookup table is provided in the ISMIR04 folder to map these numbers back to the original filenames.

3.5 GENRE DISTRIBUTION

To gain a better understanding of the characteristics of the proposed STAR Drums dataset, we analyze its content with respect to the distribution of genres.

In ISMIR04, genre tags are provided as this dataset is intended for genre-classification tasks. The genres are organized into six categories, some of which combine several genres: classical, electronic, jazz and blues, pop and rock, metal and punk, and world. We exclude all classical music as our focus is on the transcription of drum sounds used in popular music.

The majority of MTG-Jamendo tracks include genre tags with finer class divisions. We integrate several sub-categories from MTG-Jamendo into the broader categories established in ISMIR04:

- Electronic: Techno, triphop, trance, house, hiphop, dance, rap, downtempo
- Jazz and blues: Lounge
- Pop and rock: Popfolk, easy listening, alternative, indie, reggae, folk, singer/songwriter, country
- Metal and punk: Heavy metal
- World: New age

A total 139 MTG-Jamendo tracks do not have a genre tag and are assigned to the category 'unknown.' We additionally introduce the category 'others,' which includes genres such as ambient, experimental, soundtrack, chillout, and atmospheric that do not fit into any of the existing categories.

Split	Origin of data	MSS algorithm applied	Full tracks	Len. [h]
Training	ISMIR04	Yes	No	20.6
Training	MTG-Jamendo	Yes	No	94.1
Training (total)	ISMIR04 + MTG-Jamendo	Yes	No	114.7
Validation	MUSDB18	No	Yes	8.3 (6.7)
Test	MUSDB18	No	Yes	1.6 (0.3)

Table 3 Splits of STAR Drums.

Values in brackets indicate the duration of audio files that users must create by executing a mixing script. This is necessary because some track licenses of MUSDB18 do not permit the redistribution of remixed versions.

Class name	# Classes			
	18	8	5	3
Bass drum	BD	BD	BD	BD
Snare drum	SD	SD	SD	SD
Side stick	SS			
Hand clap	CLP			
Closed hi-hat	CHH	HH	HH	HH
Pedal hi-hat	PHH			
Open hi-hat	OHH			
Tambourine	TB			
Low tom	LT	TT	TT	
Mid tom	MT			
High tom	HT			
Splash cymbal	SPC	CY	CY	
Chinese cymbal	CHC			
Crash cymbal	CRC			
Ride cymbal	RD	RD		
Ride bell	RB	BE		
Cowbell	CB			
Clave/sticks	CL	CL		

Table 4 Drum classes used with mapping to eight-, five-, and three-class vocabulary, based on Vogl et al. (2018) and Zehren et al. (2023).

Genres of MUSDB18 tracks are provided on the *SigSep* home page and align with the genre categories established by ISMIR04.[4]

Figure 3 shows the genre distribution for the training split and the combined validation and test split. Both splits show a strong emphasis on pop and rock. Additionally, the genre distribution is more uneven in the validation and test split, with some genres absent. This discrepancy arises from using MUSDB18 for validation and testing, coupled with the general scarcity of freely available music recordings provided as single instrument stems. We chose not to add source-separated tracks to the validation and test split to maintain this data free from potential source separation artifacts.

As the original drum sounds are not present in STAR Drums, the genre distribution is primarily relevant for providing diverse examples of melodic instruments and vocals. This diversity ensures that a model can learn to distinguish sounds that do not originate from drum kits.

3.6 CLASS DISTRIBUTION

STAR Drums supports the 18-class vocabulary defined by Vogl et al. (2018) and listed in Table 4.

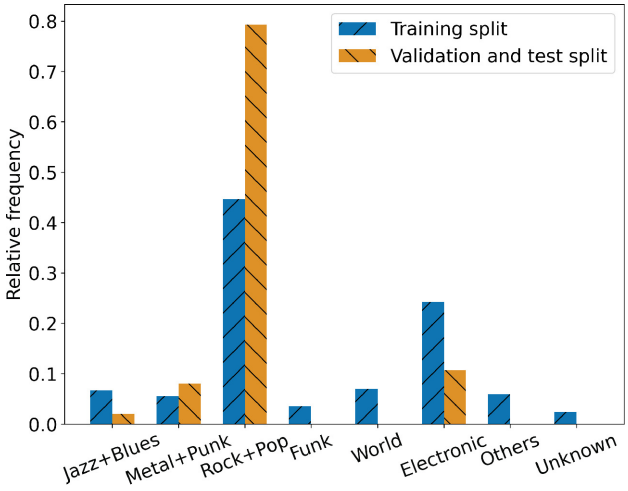

Figure 3 Genre distribution of the STAR Drums dataset.

In addition to visualizing the class frequencies of STAR Drums, we compare them to the class distributions of three manually annotated datasets: MDB Drums, ENST Drums, and RBMA13 (see Section 2). The ADT algorithm used to annotate the original drum stem will inevitably cause annotation errors. Consequently, the class distribution of the re-synthesized drum stem will differ from that of the original drum stem. Here, we investigate if the resulting class distribution strongly deviates from that of datasets which do not re-synthesize the drum stem.

Beyond altering class frequencies, the dataset-creation method inherently modifies drum beats, as individual onsets may be classified incorrectly. For instance, bass drum onsets in the original drum stem might be mistakenly classified as low tom and subsequently rendered as low tom in the re-synthesized drum stem. While such class replacements can occur naturally in human-made drum beats as part of a creative process, they might not consistently adhere to musical conventions in popular music. This is particularly relevant when incorporating language models into ADT models, as done by Ishizuka et al. (2020). The effects of such class confusions merit further investigation.

Figure 4 shows the relative class frequencies of STAR Drums, MDB Drums, ENST Drums, and RBMA13 on a logarithmic scale. We observe that the frequencies are in a similar range for most classes, with the largest differences seen in the tom and cymbals classes. Overall, the class distribution produced by the ADT algorithm used aligns well with those of manually annotated recordings.

Vogl et al. (2018) investigated the effects of artificially balancing the class distribution during training in experiments conducted with TMIDT. Their findings indicated that, while a balanced distribution improved performance for rare classes, it decreased overall performance. Additionally, altering drum classes may lead to musically implausible results. For these reasons, we chose not to manipulate the class distribution and instead recommend applying class weights during training, as outlined

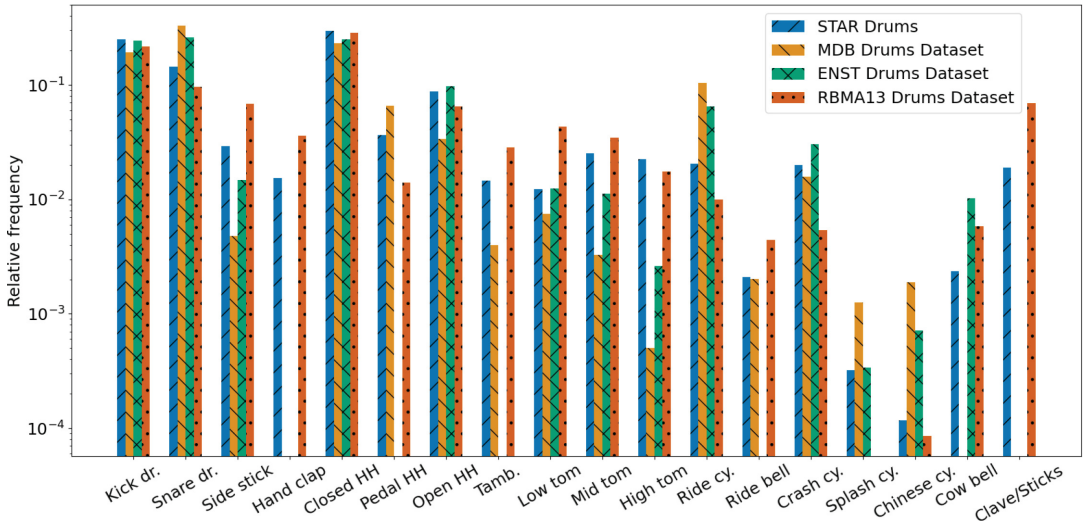

Figure 4 Relative class frequencies of STAR Drums, MDB Drums, ENST Drums, and RBMA13.

in Section 4.1.4, to adjust the influence of rare classes according to specific needs.

3.7 BENEFITS AND LIMITATIONS

In this section, we discuss the benefits and limitations resulting from the STAR Drums creation pipeline. STAR Drums supports 18 classes, a level of granularity previously only achieved by fully synthetic datasets like TMIDT. At the same time, it provides diverse recordings of non-drum data, making it attractive for training DNNs for DTM. Due to the re-synthesis of the drum-stem, the annotations are error-free and perfectly time-aligned.

However, the variety of drum sounds of STAR Drums is lower than that of recorded drum sounds produced by human drummers. First, STAR Drums uses 20 different drum kits to render all tracks, whereas the original audio tracks were recorded using a much higher number of physical drum kits. Second, the diversity of drum sounds within a single virtual drum kit is lower compared to a drum kit played by a human musician, where each produced drum sound differs slightly from the previous one. Nevertheless, professional-grade virtual drum kits offer several samples for each instrument, covering different playing techniques and finely graduated velocity, to closely replicate the sounds of a human-played drum kit.

Another potential issue with STAR Drums are residuals of the original drum sounds in the non-drum stems due to limitations in the MSS algorithm. These residuals would also appear in the final mix provided in STAR Drums and could hinder successful training, as only the re-synthesized drum sounds are correctly transcribed. We investigate this potential problem further in Section 3.8. However, the validation and test splits are created without applying MSS. If drum residuals were causing wrongly labeled training data, we would expect to see a lack of performance improvement on the validation and test splits compared to the training split, which is not the case. Moreover, the consistently high performance across all test datasets suggests that residuals are not a significant issue, as they would likely cause a lower overall performance.

Lastly, the MSS algorithm introduces artifacts in the separated instrument recordings. While no obvious artifacts are audible in the non-drum stem, this is not the case for the original drum stem, where the decay phases of the sound envelope are particularly affected. Consequently, the ADT algorithm used to create the transcription, which serves as the basis for the re-synthesis of the drum stem, may perform lower on these files compared to unprocessed drum recordings. This may lead to fewer onsets, a higher number of incorrectly classified drum sounds and a greater number of false detections. Such annotation errors affect the STAR Drums class distribution but are consistent with the STAR mix. Consequently, this does not lead to contradictory training data. Similarly to the previously mentioned issue, the validation and test split are unaffected as MSS is not applied to them.

3.8 INFLUENCE OF MSS ARTIFACTS

In this section, we address potential risks resulting from the use of MSS, as outlined in Section 3.7. In particular, we asses whether residuals of the original drum sounds in the non-drum stem lead to an increase in false positives and investigate how artifacts introduced in the drum stem affect the transcription.

First, we transcribe the drum and non-drum stems of the MUSDB18 tracks using the provided instrument stems and the same ADT algorithm used in the STAR Drums creation pipeline. As no MSS is applied in this scenario, we refer to these stems as *ideal*. In a second step, we create the drum and non-drum stems from complete mixtures by applying MSS and transcribe them as well. These tracks may contain residuals of the original drum sounds and artifacts introduced by the MSS, and are referred to as *non-ideal*.

We compare the transcriptions of ideal and non-ideal non-drum stems in terms of the total number of detected drum sounds. Since the ideal non-drum stems do not contain drum sounds, all detections are false positives. If strong residuals of the original drum sounds are present in the non-ideal non-drum stem, this should result in a higher number of false positives compared to the transcription of the ideal non-drum stem.

Interestingly, Figure 5 reveals a slightly lower number of false positives for the non-ideal non-drum stem. This could be attributed to the MSS algorithm altering the transients of non-drum instruments, especially when drum and non-drum onsets occur simultaneously. Nevertheless, we conclude from this that no strong residuals are present in the non-ideal non-drum stem. Informal listening tests further support this observation.

When comparing the number of onsets of the ideal and non-ideal drum stems, Figure 5 shows a slightly lower number for the non-ideal drum stem. To assess whether the onset times or detected classes vary significantly, we treat the annotation of the ideal drum stem as a reference. We then calculate an F-measure by comparing the transcription of the non-ideal drum stem to the one of the ideal drum stem. Similar to the approach in Section 4, we use a tolerance window of ± 50 ms and aggregate all true positives, false negatives, and false positives across all classes and tracks. We obtain an F-measure of 0.92, indicating that the transcription of the drum stem is not significantly affected by the MSS artifacts.

From these experiments, we conclude that the performance of the utilized MSS algorithm is sufficiently high for the use case in the STAR Drums creation pipeline.

4 APPLICATION TO ADT

This section compares the performance of models trained with different ADT datasets using various class vocabularies.

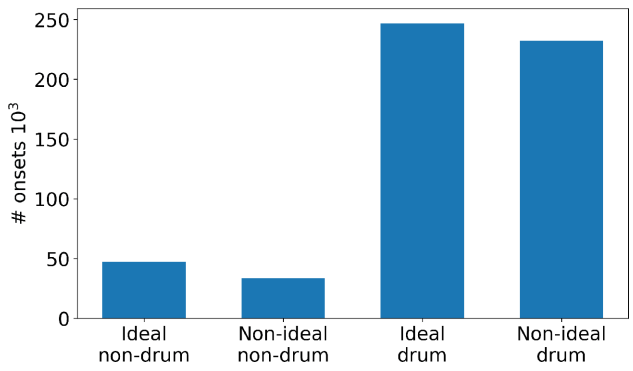

Figure 5 Total number of detected drum sounds when transcribing ideal and non-ideal non-drum stems and ideal and non-ideal drum stems of MUSDB18.

4.1 DESCRIPTION OF THE EXPERIMENTS

4.1.1 Class vocabularies

Table 4 provides an overview of all class vocabularies used, along with abbreviations for all classes and combined classes referenced in the following plots that show the performance per instrument class.

In addition to using the 18 classes as provided in STAR Drums, we also reduce the number of classes to eight, five, and three for training and testing, following the experiments conducted by Vogl et al. (2018) and Zehren et al. (2023).

4.1.2 Evaluation

We use the datasets MDB Drums, ENST Drums, RBMA13 for testing, as described in Section 2. Additionally, we evaluate the performance of all models on the test split of STAR Drums. Our evaluation is conducted solely in the DTM scenario, where we transcribe the drum instruments within the full mixtures of the test datasets.

For consistency, we perform the evaluation similar in fashion to the experiments conducted by Zehren et al. (2023): We allow a tolerance of ± 50 ms and compute global F-measures by aggregating all true positives, false negatives, and false positives across all classes and tracks. True and false positives are identified by comparing the positions of detected and reference onsets using `mir_eval` (Raffel et al., 2014).

4.1.3 Model architecture

We use a convolutional recurrent neural network model architecture, consisting of a four-layer Convolutional neural network (CNN) with 128 convolutional filters in each layer, followed by three unidirectional Gated recurrent unit (GRU) layers with 60 hidden states. The dimensionality is reduced to the number of classes through two dense layers with tanh activation function, followed by one linear layer. Onset probabilities are obtained from the output of a sigmoid function at the linear layer.

4.1.4 Training

We use magnitude spectra as input data. For the ADTOF datasets by Zehren et al. (2023), only pre-computed spectra and no raw audio data are provided. The spectra are computed using a 2048-point STFT with hop length of 441 samples, derived from audio signals sampled at 44.1 kHz using the `Madmom` library (Böck et al., 2016). This results in a frame rate of 100 Hz. The frequency bins are reduced to 84 logarithmically spaced frequency bands. The same input spectrum computation is used across all experiments.

To train the networks, we partition the training split of each dataset $D = \{(x_1, y_{1,j}), ..., (x_N, y_{N,j})\}$, where $(x_i, y_{i,j})$ represents the i-th sample and its corresponding ground truth label for class j. Here, N is the number of samples and C is the total number of classes. We apply frame-wise processing with a context of 400 frames, to provide

sufficient temporal information for accurate classification. The ground truth label $y_{i,j}$ is set to 1 if the class j is active at the center of the sample's audio block (at frame 200) and to 0 otherwise. Consequently, some training samples only contain inactive-target labels with a value of 0 for all classes.

While it is generally beneficial to include samples without class activity to help the model learn when to predict a low-onset probability, we balance the number of samples with and without class activity by randomly removing some samples that exclusively contain label values of 0 for all classes. This process is part of undersampling (Buda et al., 2018), a technique used to balance class frequencies. In this context, it helps balance precision and recall during training across all datasets, which naturally have varying onset densities and ratios of samples with and without class activity.

To address class imbalance by increasing the weight of the loss for rare classes and balancing precision and recall during training, we apply class weights to the binary cross-entropy loss for each class in the current sample. This involves computing class weights for active classes $w_{\text{active},j}$ for each class separately and using a global inactive-class weight w_{inactive} tuned as a hyperparameter.

$w_{\text{active},j}$ are calculated as follows:

$$w_{\text{active},j} = \frac{N}{N_{\text{active},j}} \tag{1}$$

where $N_{\text{active},j}$ is the number of active-class labels (value 1) of the j-th class in the training split. To ensure fast and reliable convergence across all experiments, we limit $w_{\text{active},j}$ to a value tuned as hyperparameter to prevent excessively high weight values for rare classes.

The weighted loss for the current batch containing B samples is computed using the weight vector w, which combines $w_{\text{active},j}$ and w_{inactive}:

$$\mathcal{L}_{\text{weighted}} = \frac{1}{C}\sum_{j=1}^{C}\frac{1}{B}\sum_{i=1}^{B} w_{i,j} \cdot \text{BCE}(y_{i,j}, p_{i,j}) \tag{2}$$

where

$$w_{i,j} = w_{\text{active},j} \cdot y_{i,j} + w_{\text{inactive}} \cdot (1 - y_{i,j}). \tag{3}$$

Here, $w_{i,j}$ is the class weight for the i-th sample of the j-th class. This weight can be either the active-class weight $w_{\text{active},j}$ or w_{inactive}, depending on the ground truth label $y_{i,j}$. The binary cross-entropy loss function $\text{BCE}(y_{i,j}, p_{i,j})$ is applied to the ground truth label $y_{i,j}$ and the predicted probability $p_{i,j}$.

The training process minimizes the binary cross-entropy loss using the Adam optimizer. We implement early stopping based on validation loss and select the best model according to the F-measure on the validation split.

We employ label smoothing (Zhang et al., 2021) to create soft labels. This involves adding a scaled value drawn from a normal distribution to the inactive-class labels (value 0) and subtracting it from the active-class labels (value 1). The scaling factor is tuned as a hyperparameter. Furthermore, we apply target widening by labeling the two adjacent frames of a drum sound with lower weights of 0.6 and 0.3, respectively, similar to the strategy used by Böck and Davies (2020). As a result, a label sequence of 0, 0, 0, 1, 0, 0, 0 would be transformed to, for instance, 0.03, 0.3, 0.6, 0.97, 0.6, 0.3, 0.03.

Lastly, we augment the frequency spectra by applying spectral and temporal masking as described by Park et al. (2019). All models are trained using identical hyperparameter settings.

In our experiments, the performance of models trained with ADTOF did not improve when training on both versions, ADTOF-RGW and ADTOF-YT, simultaneously. Consequently, we only report the performance of these versions separately.

4.1.5 Peak picking

A peak picking algorithm is used to determine onset times from the onset probabilities by identifying local maxima and applying a threshold. We use a threshold of 0.5 for all models and all classes. Based on our experience, performance can be slightly improved by fitting a threshold independently for each class using the validation data of the respective training dataset. While this approach works well for datasets containing recorded audio as STAR Drums or ADTOF, it results in thresholds that are too high for fully synthetic datasets, leading to poor performance on real test data.

4.2 RESULTS

Table 5 provides a performance overview of all trained models. In this section, we highlight general trends and provide detailed comments on performance across different class vocabularies.

4.2.1 General observations

For all models, performance decreases as the number of classes increases. All models achieve the lowest performance on RBMA13 and the highest on either MDB Drums or ENST Drums.

On ENST Drums, Slakh consistently achieves performance levels close to those of STAR Drums and ADTOF, whereas the performance difference is more noticeable on MDB Drums and RBMA13. For example, in the five-class scenario, the performance difference between Slakh and STAR Drums is 0.05 on ENST Drums (Slakh: 0.72, STAR Drums: 0.77), while it is 0.11 on MDB Drums (Slakh: 0.68, STAR Drums: 0.79) and 0.14 on RBMA13 (Slakh: 0.48, STAR Drums: 0.62). The trend of particularly high performance on ENST Drums compared to RBMA13 was also observed by Wu and Lerch (2018). The authors hypothesize that the presence of synthetic melodic instruments in some

Model		Test Datasets			
CL		MDB Drums	ENST Drums	RBMA13	STAR Drums Test
3	TMIDT	0.78	0.71	0.62	0.72
	Slakh	0.76	0.77	0.55	0.73
	STAR Drums	0.81	0.78	**0.67**	**0.85**
	ADTOF-RGW	0.80	**0.80**	**0.67**	0.77
	ADTOF-YT	**0.83**	0.79	0.62	0.72
5	TMIDT	0.65	0.69	0.55	0.61
	Slakh	0.68	0.72	0.48	0.59
	STAR Drums	**0.79**	**0.77**	**0.62**	**0.82**
	ADTOF-RGW	0.78	0.75	0.60	0.72
	ADTOF-YT	**0.79**	0.76	0.59	0.66
8	TMIDT	0.63	0.66	0.52	0.63
	Slakh	0.66	0.71	0.47	0.61
	STAR Drums	**0.75**	**0.74**	**0.61**	**0.80**
18	TMIDT	0.58	0.61	0.41	0.55
	Slakh	0.59	0.63	0.39	0.58
	STAR Drums	**0.67**	**0.66**	**0.50**	**0.78**

Table 5 Global F-measure when training with TMIDT, Slakh, ADTOF, or STAR Drums and testing with MDB Drums, ENST Drums, RBMA13, or STAR Drums for models transcribing 3, 5, 8, and 18 classes.

tracks and the complete absence of vocals might make ENST Drums a relatively simple dataset. These characteristics, shared with TMIDT and Slakh, may contribute to their high performance on ENST Drums.

4.2.2 Performance across class vocabularies

When transcribing three and five classes, ADTOF and STAR Drums achieve the highest performance across all test datasets. While, in the three-class scenario, ADTOF-YT and ADTOF-RGW outperform STAR Drums on MDB Drums and ENST Drums, STAR Drums matches this performance when transcribing five classes on MDB Drums and slightly exceeds it on ENST Drums and RBMA13 with F-measures of 0.77 and 0.62, respectively. The performance of ADTOF-YT and ADTOF-RGW is similar to the values reported by Zehren et al. (2023).

ADTOF supports only up to five classes and is therefore excluded from the 8- and 18-class results. STAR Drums outperforms the two synthetic datasets across all three test datasets for both vocabularies. The highest F-measures achieved are 0.75 in the 8-class scenario and 0.67 in the 18-class scenario, both on MDB Drums.

Figure 6 illustrates performance per instrument on MDB Drums for the five-class vocabulary. For bass drum and snare drum, we observe similar trends as with the global F-measure: the fully synthetic datasets Slakh and TMIDT achieve the lowest performance. With the exception of TMIDT, all models achieve similar performance on hi-hat. The presence of real recordings seems to be particularly beneficial for tom and cymbals, with notable performance improvements observed when comparing fully synthetic datasets to ADTOF and STAR Drums.

Figure 7 displays the performance per instrument on MDB Drums for the 18-class vocabulary, excluding classes without annotations (hand clap, cowbell, and clave/sticks). We observe consistently low performance across all models for mid tom, high tom, ride bell, splash cymbal, and Chinese cymbal. However, these classes have absolute class counts ranging from 4–26 in MDB Drums, making a representative evaluation difficult. This underscores a fundamental challenge when using a fine-grained class vocabulary in training, validation, and testing, as strongly varying class frequencies are typical in drum recordings, as discussed in Section 3.6.

4.2.3 Performance on STAR drums test split

Using the test split of STAR Drums in the evaluation, STAR Drums naturally achieves the highest performance across all vocabularies, as models encounter the same drum sounds during both training and testing. A small performance decrease from F-measure of 0.85 to 0.78

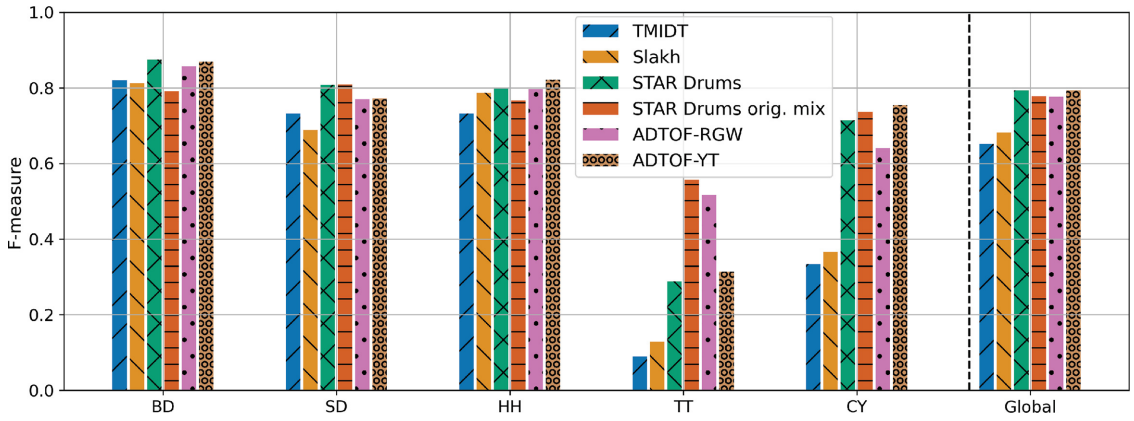

Figure 6 Global F-measure and F-measure per instrument on MDB Drums for five classes when training with TMIDT, Slakh, ADTOF-RGW, ADTOF-YT, STAR Drums, and the original mix of STAR Drums (see Section 4.2.4). Class abbreviations are explained in Table 4.

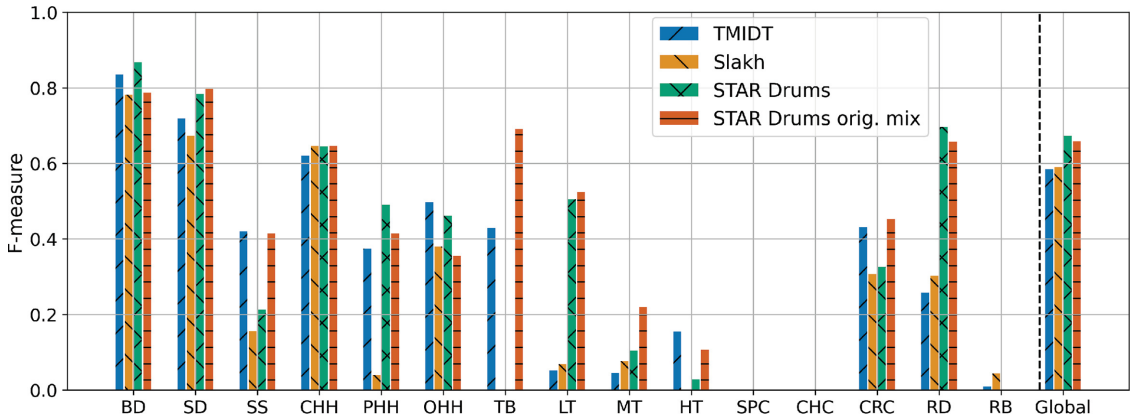

Figure 7 Global F-measure and F-measure per instrument on MDB Drums for 18-class vocabulary when training with TMIDT, Slakh, STAR Drums, and the original mix of STAR Drums (see Section 4.2.4). The classes hand clap, cowbell, and clave/sticks are excluded as MUSDB18 does not contain annotations for these classes. Class abbreviations are explained in Table 4.

with increasing class numbers suggests effective training even with 18 classes. Trends on the STAR Drums test split reflect those observed on other datasets, with ADTOF outperforming synthetic datasets when transcribing three or five classes and a similar performance for TMIDT and Slakh. This indicates that STAR Drums can also be effectively used for testing algorithms trained with other datasets. Absolute values on the STAR Drums test split fall between those for MDB Drums and RBMA13, suggesting that the difficulty of the STAR Drums test split is intermediate between MDB Drums and RBMA13.

4.2.4 Training with pseudo-labels and random mixes

In this section, we present final experiments that explore whether training with pseudo-labels and random mixes can yield competitive results, even when the input is noisy or musically implausible.

The use of pseudo-labels is intended to test their ability to reduce the need for manual annotation or precisely aligned synthesis while still supporting effective model training. Pseudo-labeling is common in self-supervised learning (Yang et al., 2023) and has been successfully applied to AMT by Strahl and Müller (2024).

Randomly combining stems from different tracks is a well-established method in training MSS algorithms to expand the amount of limited training data (Jeon et al., 2024; Özer and Müller, 2024). We aim to investigate whether similar augmentation strategies can be beneficial for ADT as well.

Table 6 lists the results in terms of global F-measure for three training approaches:

- **STAR mix:** Training based on the STAR dataset using the re-synthesized, perfectly aligned drum tracks (as in previous experiments).
- **Original mix:** Training with pseudo-labels by directly using the estimated annotations obtained from an ADT algorithm applied to the source-separated original drum stem, as described in Section 3.1.2, alongside the original input audio data.
- **STAR random mix:** Training data created by randomly combining unrelated drum and non-drum stems from different tracks to augment the dataset.

Additionally, we provide results for combinations of these approaches: STAR mix with original mix, STAR mix with STAR random mix, and original mix with STAR

Model		Test Datasets			
CL	STAR DrumsTraining Data	MDB Drums	ENST Drums	RBMA13	STAR Drums Test
5	STAR mix	**0.79**	0.77	**0.62**	**0.82**
	Original mix	0.78	**0.80**	0.60	0.73
	STAR random mix	0.78	0.77	0.61	0.81
	STAR mix + original mix	0.78	0.78	0.61	0.77
	STAR mix + STAR random mix	0.77	0.74	0.60	0.80
	Original mix + STAR random mix	0.78	0.78	0.60	0.77
18	STAR mix	0.67	0.66	0.51	**0.78**
	Original mix	0.66	**0.72**	0.53	0.65
	STAR random mix	0.65	0.67	0.51	0.77
	STAR mix + original mix	**0.68**	0.70	**0.54**	0.73
	STAR mix + STAR random mix	0.67	0.66	0.51	**0.78**
	Original mix + STAR random mix	0.67	0.69	0.53	0.73

Table 6 Global F-measure results for training with the original mix and estimated annotations (pseudo-labels), STAR random mix that combines re-synthesized drum stems and non-drum stems from different tracks, and a combination of both methods when transcribing 5 and 18 classes.

For comparison, results using the STAR mix are also provided.

random mix. By examining these combinations, we aim to explore potential synergistic effects between using the STAR mix and original mix and the effectiveness of using the random mix as additional data augmentation.

For five classes, Table 6 reveals similar results for all training variants on MDB Drums and RBMA13. On ENST Drums, a small performance increase from F-measure 0.77 to 0.80 is observed when training with the original mix instead of the STAR mix. When transcribing 18 classes, performance again increases on ENST Drums when training with the original mix, resulting in an F-measure of 0.72. On RBMA13 and MDB Drums, the combination of STAR mix and original mix leads to a slight performance increase in comparison to training on STAR mix alone with F-measures of 0.68 and 0.54, respectively.

Figures 6 and 7 show that, despite similar global F-measures, the performance of individual instrument classes varies when comparing the STAR mix and original mix. Training with the original mix results in lower performance for bass drum and hi-hat but leads to higher performance in rare classes such as side stick, tambourine, and toms. The generally good performance using pseudo-labels may be attributed to the unexpectedly high effectiveness of the ADT algorithm by Vogl et al. (2018), especially when applied to source-separated drum stems. Furthermore, labeling errors may not critically impact the used DNN architecture.

Despite the similar performance observed when using pseudo-labels in training, inaccurate annotations may not be tolerable in the test split, as even a small error rate can significantly skew results. This is especially true for rare classes, where performance is calculated from a limited number of samples. Therefore, it is advisable to use the accurately annotated STAR mix exclusively for testing purposes, while considering pseudo-labels as an additional approach for training.

According to Table 6, training with random mixes mostly results in slightly lower but still comparable performance. This may be due to the DNN architecture analyzing relatively short segments of four seconds, potentially limiting its ability to learn or rely on musical conventions. Consequently, training with musically implausible data does not significantly degrade performance. This finding aligns with experiments conducted for source separation (Jeon et al., 2024), where random mixes even enhanced overall performance when used for data augmentation. However, the combination of the STAR mix and STAR random mix does not increase performance in this investigation, as shown in Table 6, suggesting limited potential for data augmentation, at least with the current DNN architecture.

4.2.5 Summary

The STAR Drums dataset, which combines synthesized drum stems with original recordings of melodic instruments and vocals, delivers performance comparable to the semi-automatically annotated ADTOF. Additionally, STAR Drums offers greater flexibility by providing raw audio data instead of pre-computed spectra and a higher number of supported classes, and it enhances data-augmentation possibilities by providing both drum and non-drum stems. Although differences in song collection

and dataset length exist, and potential annotation errors in ADTOF remain unquantified, this experiment suggests that synthesized drum stems do not significantly diminish performance compared to original drum recordings. Moreover, the use of original recordings for non-drum sounds contributes to the superior performance of STAR Drums compared to fully synthetic data. Additionally, we demonstrated that using the annotations provided in STAR Drums as pseudo-labels for training, alongside the original mix, can achieve promising performance. This offers an additional training approach without relying on re-synthesized drum sounds.

5 CONCLUSION

The number of existing ADT datasets offering a duration suitable for training deep neural networks and offering a large number of classes is limited. We address this gap by introducing STAR Drums, a dataset containing drum signals synthesized from MIDI files alongside recordings of melodic instruments and vocals. STAR Drums is created from music licensed under Creative Commons, allowing for the redistribution of raw audio data, and supports 18 classes. We demonstrated that training with STAR Drums generally achieves higher performance compared to datasets that rely entirely on MIDI-rendered audio. Despite the synthesized nature of the drum sounds, we obtain a performance comparable to that of datasets without synthesized data, such as ADTOF. Furthermore, we showed that the annotations provided in STAR Drums can also successfully be used as pseudo-labels for the original mix, further enhancing the flexibility of STAR Drums.

When transcribing more than five classes, STAR Drums is the only large ADT dataset offering the presence of real recordings. The only other available ADT dataset in the five-class scenario containing real recordings is ADTOF which, unlike STAR Drums, offers only pre-computed spectra instead of raw audio signals. Additionally, STAR Drums provides both the drum and non-drum stems, facilitating data augmentation.

Future work may leverage the flexibility of STAR Drums to gain deeper insights into the still-limited performance of state-of-the-art ADT algorithms. One direction is to examine the effect of a reduced variety of drum sounds by using a subset of virtual drum kits for rendering the re-synthesized drum stems. Additionally, modified versions of STAR Drums could be created that exclude soft or simultaneous onsets to assess their influence on transcription performance. By generating multiple test splits with different virtual instruments, it becomes possible to analyze how performance varies depending on the similarity between test and training drum sounds. All experiments can be carried out with or without the presence of melodic instruments and vocals, using the separately provided drum and non-drum stems.

ACKNOWLEDGMENTS

The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander–Universität Erlangen–Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits (IIS).

NOTES

1. https://rbma.bandcamp.com/album/various-assets-not-for-sale-red-bull-music-academy-new-york-2013
2. http://www.midiworld.com
3. https://www.native-instruments.com
4. https://github.com/sigsep/website/blob/master/content/datasets/assets/tracklist.csv

COMPETING INTERESTS

Meinard Müller is a Co–Editor-in-Chief of the *Transactions of the International Society for Music Information Retrieval*. He was removed completely from all editorial decisions. The authors have no other competing interests to declare.

AUTHORS' CONTRIBUTIONS

Philipp Weber was the main contributor to writing the article, creating and designing the dataset, and running the experiments. Christian Uhle, Meinard Müller, and Matthias Lang shared their expertise and helped with designing the dataset and writing the article.

DATA ACCESSIBILITY

The STAR Drums dataset, including audio, annotations, accompanying code, and license information, is available on Zenodo: https://doi.org/10.5281/zenodo.15690078

AUTHOR AFFILIATIONS

Philipp Weber
Fraunhofer Institute for Integrated Circuits (IIS), Erlangen, Germany

Christian Uhle
Fraunhofer Institute for Integrated Circuits (IIS), Erlangen, Germany; International Audio Laboratories Erlangen, Germany

Meinard Müller
Fraunhofer Institute for Integrated Circuits (IIS), Erlangen, Germany; International Audio Laboratories Erlangen, Germany

Matthias Lang
Fraunhofer Institute for Integrated Circuits (IIS), Erlangen, Germany

REFERENCES

Bittner, R. M., Salamon, J., Tierney, M., Mauch, M., Cannam, C., and Bello, J. P. (2014). Medleydb: A multitrack dataset for annotation-intensive MIR research. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 155–160.

Böck, S., and Davies, M. E. P. (2020). Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation. In *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, pp. 574–582.

Böck, S., Korzeniowski, F., Schlüter, J., Krebs, F., and Widmer, G. (2016). madmom: A new python audio and music signal processing library. In *Proceedings of the ACM Conference on Multimedia Conference*, pp. 1174–1178. ACM.

Bogdanov, D., Won, M., Tovstogan, P., Porter, A., and Serra, X. (2019). The MTG-Jamendo dataset for automatic music tagging. In *Machine Learning for Music Discovery Workshop, International Conference on Machine Learning*, Long Beach, CA, United States.

Buda, M., Maki, A., and Mazurowski, M. A. (2018). A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, *106*, 249–259.

Cano, P., Gómez, E., Gouyon, F., Herrera, P., Koppenberger, M., Ong, B., Serra, X., Streich, S., and Wack, N. (2006). *ISMIR 2004 Audio Description Contest*. Music Technology Group of the Universitat Pompeu Fabra.

Dannenberg, R. B. (2006). The interpretation of MIDI velocity. In *Proceedings of the International Computer Music Conference (ICMC)*. Michigan Publishing.

Dittmar, C., and Gärtner, D. (2014). Real-time transcription and separation of drum recordings based on NMF decomposition. In *Proceedings of the 17th International Conference on Digital Audio Effects (DAFx-14)* (pp. 187–194).

Fabbro, G., Uhlich, S., Lai, C., Choi, W., Ramírez, M. A. M., Liao, W., Gadelha, I., Ramos, G., Hsu, E., Rodrigues, H., Stöter, F., Défossez, A., Luo, Y., Yu, J., Chakraborty, D., Mohanty, S. P., Solovyev, R. A., Stempkovskiy, A. L., Habruseva, T., . . . Mitsufuji, Y. (2024). The sound demixing challenge 2023 - music demixing track. *Transactions of the International Society for Music Information Retrieval*, *7*(1), 63–84.

Gillet, O., and Richard, G. (2006). ENST-drums: An extensive audio-visual database for drum signals processing. In *Proceedings of the 7th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 156–159.

Gillick, J., Roberts, A., Engel, J., Eck, D., and Bamman, D. (2019). Learning to groove with inverse sequence transformations. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Goto, M., Hashiguchi, H., Nishimura, T., and Oka, R. (2002). RWC music database: Popular, classical and jazz music databases. In *Proceedings of the 3rd International Society for Music Information Retrieval Conference (ISMIR)*.

Ishizuka, R., Nishikimi, R., Nakamura, E., and Yoshii, K. (2020). Tatum-level drum transcription based on a convolutional recurrent neural network with language model-based regularized training. In *Proceedings of the Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pp. 359–364. IEEE.

Jacques, C., and Roebel, A. (2019). Data augmentation for drum transcription with convolutional neural networks. In *Proceedings of the 27th European Signal Processing Conference (EUSIPCO)*, pp. 1–5. IEEE.

Jeon, C., Wichern, G., Germain, F. G., and Roux, J. L. (2024). Why does music source separation benefit from cacophony? In *Proceedings of the 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 873–877. IEEE.

Manilow, E., Wichern, G., Seetharaman, P., and Le Roux, J. (2019). Cutting music source separation some Slakh: A dataset to study the impact of training data quality and quantity. In *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 45–49. IEEE.

Mezza, A. I., Giampiccolo, R., Bernardini, A., and Sarti, A. (2024). Toward deep drum source separation. *Pattern Recognition Letters*, *183*, 86–91.

Özer, Y., and Müller, M. (2024). Source separation of piano concertos using musically motivated augmentation techniques. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *32*, 1214–1225.

Park, D. S., Chan, W., Zhang, Y., Chiu, C., Zoph, B., Cubuk, E. D., and Le, Q. V. (2019). SpecAugment: A simple data augmentation method for automatic speech recognition. In *Proceedings of the 20th Annual Conference of the International Speech Communication Association (Interspeech)*, pp. 2613–2617. ISCA.

Raffel, C. (2016). *Learning-Based Methods for Comparing Sequences, With Applications to Audio-to-MIDI Alignment and Matching* [PhD thesis]. Columbia University.

Raffel, C., McFee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., and Ellis, D. P. W. (2014). MIR_EVAL: A transparent implementation of common MIR metrics. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 367–372.

Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I., and Bittner, R. (2017). The MUSDB18 corpus for music separation. *arXiv:1703.04178*.

Rouard, S., Massa, F., and Défossez, A. (2023). Hybrid transformers for music source separation. In *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE.

Salamon, J., Bittner, R. M., Bonada, J., Bosch, J. J., Gómez, E., and Bello, J. P. (2017). An analysis/synthesis framework for automatic F0 annotation of multitrack datasets. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 71–78.

Southall, C., Wu, C.-W., Lerch, A., and Hockman, J. (2017). MDB drums: An annotated subset of MedleyDB for automatic drum transcription. *arXiv:1710.01813*.

Strahl, S., and Müller, M. (2024). Semi-supervised piano transcription using pseudo-labeling techniques. In *Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 173–181.

Vogl, R., Dorfer, M., Widmer, G., and Knees, P. (2017). Drum transcription via joint beat and drum modeling using convolutional recurrent neural networks. In *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 150–157.

Vogl, R., Widmer, G., and Knees, P. (2018). Towards multi-instrument drum transcription. In *Proceedings of the 21st International Conference on Digital Audio Effects (DAFx-18)*.

Weber, P., Uhle, C., Müller, M., and Lang, M. (2024). Real-time automatic drum transcription using dynamic few-shot learning. In *Proceedings of the 5th International Symposium on the Internet of Sounds (IS2)*. IEEE.

Wei, I., Wu, C., and Su, L. (2021). Improving automatic drum transcription using large-scale audio-to-midi aligned data. In *Proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 246–250. IEEE.

Wu, C., Dittmar, C., Southall, C., Vogl, R., Widmer, G., Hockman, J., Müller, M., and Lerch, A. (2018). A review of automatic drum transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(9), 1457–1483.

Wu, C., and Lerch, A. (2018). From labeled to unlabeled data - on the data challenge in automatic drum transcription. In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 445–452.

Yang, X., Song, Z., King, I., and Xu, Z. (2023). A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, *35*(9), 8934–8954.

Zehren, M., Alunno, M., and Bientinesi, P. (2021). ADTOF: A large dataset of non-synthetic music for automatic drum transcription. In *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, pp. 818–824.

Zehren, M., Alunno, M., and Bientinesi, P. (2023). High-quality and reproducible automatic drum transcription from crowdsourced data. *Signals*, 4(4), 768–787.

Zehren, M., Alunno, M., and Bientinesi, P. (2024). Analyzing and reducing the synthetic-to-real transfer gap in music information retrieval: The task of automatic drum transcription. *CoRR*, abs/2407.19823.

Zhang, C.-B., Jiang, P.-T., Hou, Q., Wei, Y., Han, Q., Li, Z., and Cheng, M.-M. (2021). Delving deep into label smoothing. *IEEE Transactions on Image Processing*, *30*, 5984–5996.

TO CITE THIS ARTICLE:
Weber, P., Uhle, C., Müller, M., & Lang, M. (2025). STAR Drums: A Dataset for Automatic Drum Transcription. *Transactions of the International Society for Music Information Retrieval*, *8*(1), 248–264. DOI: https://doi.org/10.5334/tismir.244

Submitted: 11 December 2024 **Accepted:** 16 June 2025 **Published:** 29 July 2025

COPYRIGHT:
© 2025 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See https://creativecommons.org/licenses/by/4.0/.

Transactions of the International Society for Music Information Retrieval is a peer-reviewed open access journal published by Ubiquity Press.