Pattern Recognition Letters 183 (2024) 86–91

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Toward deep drum source separation

Alessandro Ilic Mezza *, Riccardo Giampiccolo, Alberto Bernardini, Augusto Sarti

Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Piazza Leonardo da Vinci 32, Milan, 20133, Italy

ARTICLE INFO

Editor: Xiaoning Qian

MSC:
68T07

Keywords:
Deep learning
Drums
Music decomposition
Source separation
U-Net

ABSTRACT

In the past, the field of drum source separation faced significant challenges due to limited data availability, hindering the adoption of cutting-edge deep learning methods that have found success in other related audio applications. In this letter, we introduce StemGMD, a large-scale audio dataset of isolated single-instrument drum stems. Each audio clip is synthesized from MIDI recordings of expressive drum performances using ten real-sounding acoustic drum kits. Totaling 1224 h, StemGMD is the largest audio dataset of drums to date and the first to comprise isolated audio clips for every instrument in a canonical nine-piece drum kit. We leverage StemGMD to develop LarsNet, a novel deep drum source separation model. Through a bank of dedicated U-Nets, LarsNet can separate five stems from a stereo drum mixture faster than real-time and is shown to considerably outperform state-of-the-art nonnegative spectro-temporal factorization methods.

Deep learning has been a breakthrough in the field of audio source separation, with applications ranging from speech [1] to a variety of environmental sounds [2]. In recent years, neural-network-aided music demixing models (MDX) have shown tremendous results. Open-Unmix [3], Spleeter [4], Meta-TasNet [5], and HT-Demucs [6], to name a few, have indeed proved very capable at extracting various stems from a mixed music track. MDX models are typically trained to isolate four stems [7,8], i.e., "vocals", "bass", "drums", and "other", while sometimes including other instruments, such as piano and guitar. In practice, however, the drum kit is conventionally treated as a single instrument despite being an ensemble of various percussion instruments in itself. Unfortunately, this prevents the full inversion of the mixing process performed by the sound engineer when producing a song. High-quality Drum Source Separation (DSS) may have a far-reaching impact on many creative applications, as it may enhance remixing, remastering, and audio production by providing precise control over individual drum elements, allowing artists and sound engineers to fine-tune their tracks beyond the capabilities of existing MDX software. Additionally, it can be valuable in music analysis, facilitating detailed studies of drum patterns and rhythms, as well as in music education to isolate drum parts for instructional purposes. Furthermore, DSS may be instrumental in advancing Automatic Drum Transcription (ADT) [9].

Deep learning methods are data-hungry. However, as shown in Table 1, available datasets of isolated drum stems are few and of limited size. This substantial lack of data is possibly one of the main reasons why, so far, deep DSS has not been investigated in the literature, unlike other drum-related tasks such as, e.g., ADT, for which deep learning approaches have been studied extensively [10–13]. By contrast, the latest advances in DSS to date are based on variants of nonnegative matrix factorization (NMF) and nonnegative matrix factor deconvolution (NMFD), as they prove effective even when only a small amount of data is available [14–17].

With this work, our goal is to address the longstanding data scarcity problem that held back research into more advanced data-driven DSS methods in the past few decades, and demonstrate that deep neural networks are not only a viable solution but also an efficient tool when it comes to DDS. In this letter, we present StemGMD, a new large-scale audio dataset of isolated drum stems synthesized from several hours of expressive performances. We leverage StemGMD to develop LarsNet, a deep DSS model that separates five stems from a stereo drum mixture. Each stem is extracted by a dedicated U-Net yielding a spectro-temporal soft mask that is applied to the short-time Fourier transform (STFT) of the mixture signal. In our experiments, LarsNet consistently outperforms state-of-the-art DSS methods based on NMF and NMFD, while greatly reducing inter-channel cross-talk artifacts[1] and computational time.

1. Dataset

In 2019, Magenta released GMD, a large corpus comprising 13.6 h of human-performed drum tracks recorded by ten drummers playing

* Corresponding author.
E-mail addresses: alessandroilic.mezza@polimi.it (A.I. Mezza), riccardo.giampiccolo@polimi.it (R. Giampiccolo), alberto.bernardini@polimi.it (A. Bernardini), augusto.sarti@polimi.it (A. Sarti).

[1] Audio examples are available online: https://polimi-ispl.github.io/larsnet.

https://doi.org/10.1016/j.patrec.2024.04.026
Received 26 September 2023; Received in revised form 21 March 2024; Accepted 29 April 2024
Available online 4 May 2024

Table 1
Overview of existing drums datasets.

	Clips	Duration (h)	Classes/Mics	Drum kits	Real/Synth	Human	Mixture	Transcription	Isolated stems
MDB-Drums [18]	23	0.345	21	≤ 23	R	✓	Mono	TXT	×
IDMT-SMT-Drums [14]	608	2.1	3	N/A	R/S	×	Mono	XML	64 mixtures
ENST-Drums [19]	456	3.75	20/8	3	R	✓	Stereo	TXT	✓
GMD [20]	1150	13.6	22	1	S	✓	Mono	MIDI	×
TMIDT [21]	4197	259	18	57	S	×	Mono	TXT	×
E-GMD [12]	45 537	444.5	22	43	S	✓	Mono	MIDI	×
StemGMD (ours)	103 500	1224	9	10	S	✓	Stereo	MIDI	✓

on a Roland TD-11 electronic drum kit [20]. GMD contains MIDI files for each performance, along with the corresponding full-kit audio mixtures. Later on, [12] introduced E-GMD, an ADT dataset containing about 444 h of audio data gathered by re-recoding all GMD sequences using 43 drum kits. Despite being valuable for many tasks, such as beat generation, drum infilling, ADT, and groove transfer [20], these datasets do not contain isolated stems and are thus unsuited for training deep source separation models. In this letter, we expand the GMD family by introducing StemGMD, a new large-scale corpus of isolated drum stems.

First, we applied the note mapping proposed in [20] to the raw MIDI data provided with GMD. Such mapping, while preserving the velocity and timing of each note, reduces the 22 original MIDI channels to nine canonical instruments (whose General MIDI note numbers are reported in brackets), i.e., Kick Drum (36), Snare Drum (38), High Tom (50), Low-Mid Tom (47), High Floor Tom (43), Open Hi-Hat (46), Closed Hi-Hat (42), Crash Cymbal (49), and Ride Cymbal (51).

Then, we synthesized each of the nine channels of the resulting MIDI clips as 16-bit/44.1 kHz stereo WAV files. We used high-fidelity Logic Pro X sample libraries, selecting ten different acoustic drum kits in order to cover a wide range of timbres. To respect the superposition principle that underlies most source separation techniques, we obtained the mixture signals simply by summing the nine synthesized stems. This procedure preserves a one-to-one correspondence between the audio tracks and the original MIDI files. Hence, GMD metadata were kept for each clip, e.g., duration, genre, tempo, and drummer ID, which we augmented with the drum kit ID. The resulting audio dataset, which we call StemGMD, is made freely available online.[2]

StemGMD comprises more than 136 h of drum mixtures and totals approximately 1224 h of audio.

With 103 500 clips, to the best of our knowledge, StemGMD is the largest publicly available dataset of drums. Moreover, it is the first large-size dataset of isolated stems to account for all pieces in a standard drum kit, including toms and cymbals.

The train, test, and validation folds from Magenta's GMD are retained in StemGMD. However, the recommended test fold amounts to more than ten hours. Therefore, in this work, we use a more manageable test split named *Eval Session*. StemGMD *Eval Session* comprises 400 drum mixtures (40 for each drum kit) and spans several genres, as it was collected by tasking four drummers to record the same set of ten beats in their own style [20].

2. Deep DSS model

This section presents a new deep DSS model dubbed LarsNet,[3] which is designed to separate five drum stems from a stereo drum mixture, i.e., kick drum (KD), snare drum (SD), tom-toms (TT), hi-hat (HH), and cymbals (CY). More specifically, TT includes High Tom, Low-Mid Tom, and High Floor Tom; HH includes Open and Closed Hi-Hat; and CY includes Crash and Ride cymbals.

[2] StemGMD is publicly available online: https://zenodo.org/records/7860223.

[3] Pretrained LarsNet models are available online: https://github.com/polimi-ispl/larsnet.

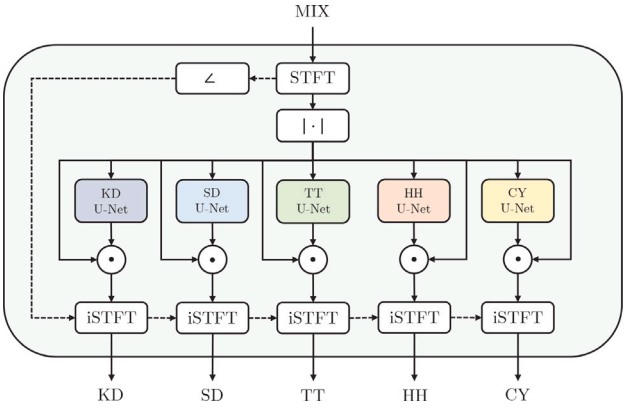

Fig. 1. LarsNet architecture.

2.1. Model architecture

Inspired by prior work on MDX [4,6,22], LarsNet comprises five parallel U-Nets [23], i.e., one for each target stem. As shown in Fig. 1, the U-Nets operate in the time-frequency domain. Each is fed a (portion of a) two-channel magnitude STFT computed from the stereo waveform of a drum mixture and outputs a stem-specific soft mask of the same size. The time-domain stem signal is estimated by taking the inverse STFT (iSTFT) of the Hadamard product between the complex-valued STFT of the mixture and the real-valued mask thus obtained. The input magnitude STFT is first cropped to retain only the lowest F frequency bands. Then, zero-padding is applied along the temporal dimension before segmenting it into an integer number of chunks of T time frames. The resulting spectro-temporal patches $\mathbf{X} \in \mathbb{R}_{\geq 0}^{2\times F\times T}$ are normalized using Batch Norm computed across frequency bands (FreqBN), instead of across channels as it is usual in computer vision applications [24]. As shown in Fig. 2, each U-Net comprises 13 convolutional layers. Such layers have 5×5 kernels, 2×2 stride, and 2×2 padding, except for the last decoder layer which has 4×4 kernels, 2×2 dilation, and 3×3 padding. The decoder yields a soft mask $\mathbf{M} \in [0, 1]^{2\times F\times T}$ through a Sigmoid nonlinearity. The mask is zero-padded back to the original STFT size, and the temporal dimension is reinstated by concatenating all T-sized chunks. Finally, the time-domain signal is reconstructed by taking the iSTFT of the masked magnitude after padding any frequency band above F with zeros. Albeit more sophisticated phase estimation techniques exist, e.g., [25,26], we opt to simply use the original mixture phase when transforming the signal back into the time domain for efficiency's sake.

2.2. Model training

We train the U-Nets in parallel on five NVIDIA Titan V GPUs for 100 000 iterations using Adam, a learning rate of 0.0001, and a batch size of 24. Each mixture-stem training pair consists of aligned stereo waveform segments of 11.85 s extracted from StemGMD with a stride of 2 s, resulting in 110 000 segments per epoch. We compute the STFT using a periodic Hann window of length 4096 and a hop size of 1024 (ca.

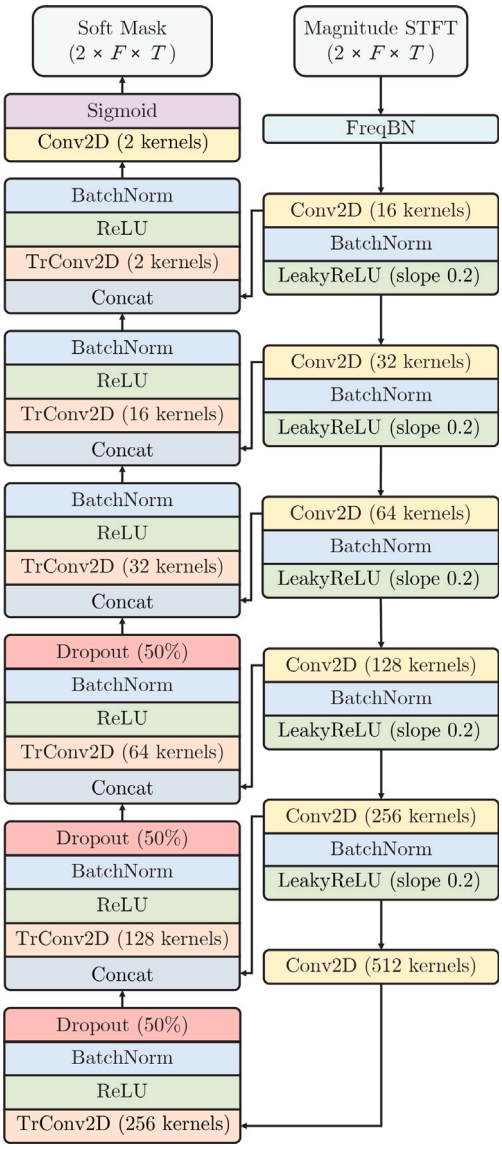

Fig. 2. U-Net architecture.

23 ms). Since we are dealing with percussion instruments whose onsets have impulsive spectral characteristics, we set $F = 2048$ to ensure a broadband estimate of every stem. Additionally, to avoid padding the STFT patches at training time, we fix $T = 512$. The loss function is computed as the L^1-norm of the error between the magnitude STFT patch of the ground truth stem $\mathbf{X}_i \in \mathbb{R}_{\geq 0}^{2\times F\times T}$ and that of the masked mixture, i.e.,

$$\mathcal{L} = \|\mathbf{X}_i - \mathbf{M}_i \odot \mathbf{X}\|_1, \tag{1}$$

where $\odot$ denotes the Hadamard product. Throughout the training, we apply data augmentation as described in Section 3. We use six of the ten drum kits included in StemGMD to train LarsNet, whereas the remaining four are held out for evaluation purposes on unseen drum sounds (see Table 2).

2.3. Wiener filtering

Once the training is over, the magnitude STFT of each stem can be inferred independently of the others as

$$\hat{\mathbf{X}}_i = \mathbf{M}_i \odot \mathbf{X}. \tag{2}$$

Additionally, LarsNet implements α-Wiener filtering [27] as in [15]. Namely, the ith soft mask is computed as

$$\tilde{\mathbf{M}}_i = \hat{\mathbf{X}}_i^{\alpha} \oslash \left(\sum_{j=1}^{N} \hat{\mathbf{X}}_j^{\alpha} + \epsilon \right) \tag{3}$$

and the magnitude STFT is estimated as $\tilde{\mathbf{X}}_i = \tilde{\mathbf{M}}_i \odot \mathbf{X}$, where ϵ avoids division by zero, $\oslash$ indicates the Hadamard division, and the exponent $\alpha \in (0, 2]$ is applied in an element-wise fashion. In the following, we report the results of this latter LarsNet variant by setting $\alpha = 1$. Notice that, differently from the forward inference mechanism, (3) combines the estimates of all $N = 5$ stems, preventing full parallelization.

2.4. Target stems

In this work, we separate five stems (KD, SD, TT, HH, CY) even if StemGMD contains isolated tracks for nine different instruments (see Section 1). From an application standpoint, we argue that there are limited practical scenarios in which one might want to isolate, e.g., the High Tom from the Low-Mid Tom, as such a distinction mainly depends on the diameter and tuning of the drums that may vary greatly across drum kits. Furthermore, some drummers may use two tom-toms; others may have more than three. Having a single TT class allows one to isolate that entire family of drums regardless of the drum kit composition. Similarly, the CY class may be considered a stereo "overhead" track where all cymbals are recorded at once. In fact, the ride and crash cymbals are rarely recorded using dedicated microphones unlike, e.g., the hi-hat, which is typically close-miked. Finally, the hi-hat is a percussion instrument that can be played in two different ways (open or closed) depending on whether the pedal is pressed or not. In having a single HH class, we favored the consistency with respect to the acoustic source rather than the sounds it may produce, akin to having a microphone capturing the instrument.

3. Data augmentation

Having access to isolated audio stems enables countless data augmentation strategies and allows one to draw inspiration from common music production practices. We use the following six methods; the order in which they are presented below corresponds to the order in which they are applied in our implementation.

1. **Kit-swap augmentation (KS)**: for each drum pattern, we create a novel mixture by adding together stems from randomly selected drum kits.
2. **Doubling augmentation (DB)**: in modern-day music production, layering multiple drum hits in a Digital Audio Workstation (DAW) is common practice. Inspired by this, for each stem, we compute a new track by averaging the same pattern from different drum kits. To save on I/O operations, this method is limited to two drum kits at a time.
3. **Pitch-shift augmentation (PS)**: to simulate a wider range of drums and cymbals, each stem is pitch shifted by a random amount of semitones using SoX as backend. Specifically, we randomly sample integer-valued shifts in the range of ± 3 semitones.
4. **Saturation augmentation (ST)**: we apply nonlinear processing to individual stems to simulate compression and saturation, which are common when mixing drums in a DAW. Namely, for each stem, we compute the hyperbolic tangent (tanh) after scaling the input waveform by a uniformly distributed random variable $\beta_i \sim \mathcal{U}_{[1,5]}$.
5. **Channel-swap augmentation (CS)**: left and right channels of an audio track are randomly swapped.
6. **Remix augmentation (RX)**: Each stem is multiplied by a scalar $\gamma_i \sim \mathcal{U}_{[0.1,1]}$, corresponding to a gain variation in the range of -20 dB to 0 dB. Albeit increasing the gain above 0 dB would have been possible, we opted against it to avoid potential clipping.

Table 2
nSDR computed on StemGMD Eval Session. The six drum kits denoted by the superscript ⋆ were included in the training dataset, whereas the four drum kits denoted by † were held out for evaluation.

	Method	Stem	No.	Drum kit										
				Brooklyn⋆	East Bay⋆	Heavy⋆	Portland ⋆	Retro Rock⋆	SoCal⋆	Bluebird†	Detroit Garage†	Motown Rev.†	Roots†	All
Nonzero-energy stems	**SAB-NMF** [14]	KD	400 (100%)	4.63	3.48	6.14	9.57	4.81	5.49	5.82	5.30	7.20	15.55	6.80
		SD	400 (100%)	13.03	16.10	17.56	11.99	18.32	20.73	16.86	9.14	10.27	14.58	14.86
		TT	40 (10%)	−11.66	−11.81	−12.13	−10.36	−14.78	−10.06	−11.45	−16.96	−11.06	−5.07	−11.53
		HH	390 (97.5%)	1.97	6.50	3.88	−0.21	6.61	6.47	1.14	−3.34	5.57	4.26	3.29
		CY	50 (12.5%)	−5.42	−2.98	−1.04	−9.22	−0.97	−2.11	−1.67	−7.41	−7.03	−5.27	−4.31
		All	1280	5.54	7.61	8.17	5.99	8.74	9.77	7.01	2.67	6.54	10.35	7.24
	NMFD [15]	KD	400 (100%)	21.98	24.29	12.33	25.12	24.67	24.14	28.26	27.19	23.08	21.62	23.27
		SD	400 (100%)	11.07	10.83	11.83	10.27	4.86	12.75	10.02	8.75	7.17	11.71	9.93
		TT	40 (10%)	−2.64	−2.73	−12.62	−0.48	−8.80	−1.67	0.23	−1.85	−5.55	−5.70	−4.18
		HH	390 (97.5%)	4.51	3.35	2.75	−1.43	4.21	3.47	3.72	2.95	3.93	3.56	3.10
		CY	50 (12.5%)	−6.49	−5.34	−5.02	−7.37	−3.81	−4.08	−6.96	−7.16	−3.40	−7.30	−5.69
		All	1280	11.37	11.70	7.80	10.32	10.09	12.38	12.83	11.79	10.35	11.04	10.97
	LarsNet (Ours)	KD	400 (100%)	27.07	26.91	26.22	31.17	27.10	25.99	29.29	27.54	25.56	25.07	**27.19**
		SD	400 (100%)	21.48	20.91	20.84	22.61	23.07	24.61	23.01	21.49	17.42	22.23	**21.77**
		TT	40 (10%)	9.22	10.25	9.49	8.37	10.43	11.92	9.50	7.65	8.37	5.84	**9.10**
		HH	390 (97.5%)	6.80	9.04	5.69	4.52	7.70	8.35	6.03	4.54	5.59	6.03	**6.43**
		CY	50 (12.5%)	4.46	5.75	5.08	4.39	4.82	5.46	3.91	4.08	3.32	−0.35	**4.09**
		All	1280	**17.71**	**18.24**	**16.93**	**18.62**	**18.54**	**18.94**	**18.63**	**17.10**	**15.53**	**16.79**	**17.70**
Zero-energy stems	**SAB-NMF** [14]	TT	360 (90%)	−37.33	−37.09	−36.77	−36.38	−33.89	−30.49	−35.29	−33.49	−32.85	−26.06	−33.96
		HH	10 (2.5%)	−27.91	−30.16	−28.13	−20.68	−20.11	−25.25	−25.13	−27.85	−24.16	−27.55	−25.69
		CY	350 (87.5%)	−20.60	−14.13	−7.98	−25.97	−12.57	−11.15	−10.92	−23.23	−25.06	−17.44	−16.91
		All	720	−29.07	−25.83	−22.66	−31.10	−23.34	−21.01	−23.30	−28.42	−28.95	−21.89	−25.56
	NMFD [15]	TT	360 (90%)	−27.19	−24.14	−33.91	−20.97	−35.82	−23.25	−24.46	−23.34	−27.14	−30.62	−27.08
		HH	10 (2.5%)	−24.67	−28.37	−27.26	−25.54	−22.82	−23.34	−21.59	−23.88	−26.43	−26.67	−25.06
		CY	350 (87.5%)	−24.61	−23.53	−21.61	−28.85	−21.13	−22.20	−24.15	−24.44	−25.51	−24.12	−24.01
		All	720	−25.90	−23.90	−27.84	−24.87	−28.50	−22.74	−24.27	−23.88	−26.33	−27.41	−25.56
	LarsNet (Ours)	TT	360 (90%)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	−8.99	**−0.90**
		HH	10 (2.5%)	0.00	−4.34	−3.78	0.00	0.00	0.00	0.00	−7.24	−5.48	−23.65	**−4.45**
		CY	350 (87.5%)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	−2.46	−4.33	**−0.68**
		All	720	**0.00**	**−0.06**	**−0.05**	**0.00**	**0.00**	**0.00**	**0.00**	**−0.10**	**−1.27**	**−6.93**	**−0.84**

If we apply all augmentations at once, the mixture signal

$$\mathbf{x}[n] = \sum_{i=1}^{N} \mathbf{x}_i[n]$$

becomes

$$\mathbf{x}_{\mathrm{a}}[n] = \sum_{i=1}^{N} \gamma_i \tanh\left(\beta_i \ \mathrm{shift}_{\pm 3}\left(\frac{\mathbf{C}\mathbf{x}_i^{(k)}[n] + \mathbf{C}\mathbf{x}_i^{(k')}[n]}{2}\right)\right),$$

where i denotes the stem index out of N stems, k indicates the index of a randomly selected drum kit, $k \neq k'$, $\mathbf{C} = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$, $\beta_i \sim \mathcal{U}_{[1,5]}$, $\gamma_i \sim \mathcal{U}_{[0.1,1]}$, and $\mathrm{shift}_{\pm 3}(\cdot)$ is an operator shifting the pitch of the input signal by a random integer amount in the range of $[-3, 3]$ semitones. In practice, each augmentation method is applied stochastically and independently of the others. If applied, KS and RX involve all stems in the mixture, whereas applying DB, PS, ST, and CS depends on the outcome of a Bernoulli trial run for each voice. The probability of every augmentation is as follows: Pr(KS) = Pr(CS) = 0.5, and Pr(DB) = Pr(PS) = Pr(ST) = Pr(RX) = 0.3. On top of that, at training time, we disable data augmentation altogether with a probability of 50%.

4. Evaluation

To assess the performance of our deep DSS model, we consider the signal-to-distortion ratio (SDR). In particular, we adopt the definition used in previous MDX challenges, which [6] refers to as nSDR. Namely, for the ith stem, the metric is given by

$$\mathrm{nSDR}_i = 10 \log_{10} \frac{\sum_n \|\mathbf{x}_i[n]\|^2 + \epsilon}{\sum_n \|\mathbf{x}_i[n] - \hat{\mathbf{x}}_i[n]\|^2 + \epsilon}, \tag{4}$$

where n is the time index, $\epsilon = 10^{-7}$ is a small scalar introduced to avoid numerical problems, $\mathbf{x}_i[n]$ is the ith ground truth source, and $\hat{\mathbf{x}}_i[n]$ is the estimated stereo waveform. Then, the overall nSDR score is given by the average over every stem of every clip in the test set. Different from the more widespread formulation by Vincent et al. [28] (later referred to as oSDR for clarity's sake), (4) is simpler and faster to evaluate, which is convenient given the considerable size of large-scale deep learning datasets. The SDR is well-defined as long as the discrete-time energy of the target signal $\mathbf{x}_i[n]$ is greater than zero. If not, the optimal ratio in (4), i.e., assuming no distortion, is nonpositive and caps at 0 dB. Hence, we divide the test set into two parts by annotating each stem according to the corresponding MIDI file. Specifically, we analyze the results for all those stems where at least one MIDI note is present (nonzero-energy stems) separately from those with no drum hits (zero-energy stems). We test LarsNet against the baseline methods described in Section 4.1. The three methods are evaluated on the *Eval Session* fold of StemGMD, comprising mixtures from ten drum kits, four of which were held out during training.

4.1. Baseline methods

In [14], Dittmar and Gärtner presented a low-latency method for drums transcription and separation of KD, SD, and HH based on frame-wise nonnegative matrix factorization with semi-adaptive bases (SAB-NMF). The method aims at decomposing the magnitude STFT of the drum mixture into two nonnegative matrices, one comprising a set of spectral templates (bases) and the other containing the corresponding temporal activations. In particular, SAB-NMF achieves this factorization by processing each short-time spectrum independently of the others. In [15], Dittmar and Müller proposed an alternative approach based on nonnegative matrix factor deconvolution (NMFD) followed by α-Wiener filtering. Differently from other methods, [15] assumes one has prior knowledge about the drum score and presents an *informed* NMFD variant. However, such a prior is rarely available when it comes to real-life source separation problems. Hence, we implement the baseline decomposition from [15], which closely resembles the classic NMFD formulation by Smaragdis [29].

Compared to [14,15], we apply the multiplicative update rules [30] for $K = 200$ iterations instead of 25 and 30, as it remarkably improved the nSDR.

In our implementation, the spectral basis functions are pre-computed using a partition of StemGMD containing isolated drum and cymbal

hits, each synthesized at ten different velocities ranging from 30 to 127 using the same ten drum kits as in the main dataset. Moreover, whereas the original references only considered mono files, we process the two stereo channels independently of one another. As for the remaining hyperparameters, we refer the readers to the original papers.

4.2. Results

Table 2 reports the nSDR for every drum kit and isolated stem in StemGMD *Eval Session*. The "All" rows show the average nSDR over all stems in a drum kit, while the "All" column lists the average across all drum kits for a given stem.

First, none of the methods exhibits a noticeable drop in performance when evaluated on the four held-out drum kits (marked with † in Table 2), suggesting that all three methods can generalize in the face of unseen timbral characteristics.

By looking at the nSDR of nonzero-energy stems, we notice that LarsNet provides, on average, a performance increment of +20.39 dB for KD, +6.91 dB for SD, +20.63 dB for TT, +3.14 dB for HH, and +8.4 dB for CY compared to SAB-NMF. Similarly, we report an increment of +3.92 dB for KD, +11.84 dB for SD, +13.28 dB for TT, +3.33 dB for HH, and +9.78 dB for CY with respect to NMFD. When accounting for all drum kits and stems, LarsNet yields an nSDR of 17.7 dB, against the 10.97 dB of NMFD and 7.24 dB of SAB-NMF. Respectively, the average oSDR scores [28] obtained using `mir_eval` [31] are 17.91 dB for LarsNet, 11.02 dB for NMFD, and 7.23 dB for SAB-NMF, differing at most by 0.21 dB from the respective nSDR scores.

As for the results pertaining to zero-energy stems, both SAB-NMF and NMFD have an average nSDR of −25.56 dB. This indicates a great deal of cross-talk between drum channels, i.e., sound components from other stems tend to leak into the TT, HH, and CY tracks. Conversely, LarsNet scores a perfect 0 dB for many of the ten drum kits, meaning that the proposed method correctly outputs silence when no drum hits are present in the ground truth stem. Overall, LarsNet average nSDR is −0.84 dB, corresponding to a remarkable +24.7 dB improvement upon the baselines.

4.3. Real-time performance

LarsNet is implemented in Python using PyTorch. Despite not being optimized for speed, the model achieves an average Real-Time Ratio (RTR) of 0.016 on a single NVIDIA Titan V GPU (62.5 times faster than real-time) and 0.15 on an Intel Xeon E5-2687 W (6.6 times faster than real-time). On the same CPU, SAB-NMF and NMFD achieve an RTR of 4.85 (slower than real-time) and 0.4 (only 2.5 times faster than real-time), respectively.

5. Conclusions

In this letter, we presented StemGMD, the first large-scale audio dataset of isolated drum stems encompassing all the percussion instruments of a canonical nine-piece drum kit, and LarsNet, a first-ever deep drum source separation model. Based on a parallel arrangement of dedicated U-Nets, LarsNet can extract five stems from a stereo drum mixture up to 62.5 times faster than real-time. At the same time, LarsNet is shown to substantially outperform state-of-the-art methods based on nonnegative spectro-temporal decomposition in terms of signal-to-distortion ratio and unwanted cross-talk artifacts.

CRediT authorship contribution statement

Alessandro Ilic Mezza: Writing – original draft, Methodology, Conceptualization, Investigation, Data curation, Writing – review & editing, Software. **Riccardo Giampiccolo:** Writing – review & editing, Investigation, Validation, Data curation. **Alberto Bernardini:** Supervision, Writing – review & editing. **Augusto Sarti:** Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data and code have been made publicly available online.

Acknowledgments

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 — program "RESTART.")

References

[1] D. Wang, J. Chen, Supervised speech separation based on deep learning: An overview, IEEE/ACM Trans. Audio Speech Lang. Process. 26 (10) (2018) 1702–1726.
[2] I. Kavalerov, S. Wisdom, H. Erdogan, B. Patton, K. Wilson, J. Le Roux, J.R. Hershey, Universal sound separation, in: 2019 IEEE Workshop Appl. Signal Process. Audio Acoust., 2019, pp. 175–179.
[3] F.-R. Stöter, S. Uhlich, A. Liutkus, Y. Mitsufuji, Open-Unmix – A reference implementation for music source separation, J. Open Source Softw. 4 (41) (2019) 1667.
[4] R. Hennequin, A. Khlif, F. Voituret, M. Moussallam, Spleeter: A fast and efficient music source separation tool with pre-trained models, J. Open Source Softw. 5 (50) (2020) 2154.
[5] D. Samuel, A. Ganeshan, J. Naradowsky, Meta-learning extractors for music source separation, in: 2020 IEEE Int. Conf. Acoust. Speech Signal Process., 2020, pp. 816–820.
[6] A. Défossez, Hybrid spectrogram and waveform source separation, in: Proc. MDX Workshop, 2021, pp. 1–13.
[7] F.-R. Stöter, A. Liutkus, N. Ito, The 2018 signal separation evaluation campaign, in: Latent Variable Analysis and Signal Separation, 2018, pp. 293–305.
[8] Y. Mitsufuji, G. Fabbro, S. Uhlich, F.-R. Stöter, A. Défossez, M. Kim, W. Choi, C.-Y. Yu, K.-W. Cheuk, Music demixing challenge 2021, Front. Signal Process. 1 (2022) 1–14.
[9] C.-W. Wu, C. Dittmar, C. Southall, R. Vogl, G. Widmer, J. Hockman, M. Müller, A. Lerch, A review of automatic drum transcription, IEEE/ACM Trans. Audio Speech Lang. Process. 26 (9) (2018) 1457–1483.
[10] R. Vogl, M. Dorfer, P. Knees, Drum transcription from polyphonic music with recurrent neural networks, in: 2017 IEEE Int. Conf. Acoust. Speech Signal Process., 2017, pp. 201–205.
[11] K. Choi, K. Cho, Deep unsupervised drum transcription, in: Proc. 20th Int. Soc. Music Inf. Retrieval Conf., 2019, pp. 183–191.
[12] L. Callender, C. Hawthorne, J. Engel, Improving perceptual quality of drum transcription with the Expanded Groove MIDI dataset, 2020, arXiv preprint arXiv:2004.00188.
[13] R. Ishizuka, R. Nishikimi, K. Yoshii, Global structure-aware drum transcription based on self-attention mechanisms, Signals 2 (3) (2021) 508–526.
[14] C. Dittmar, D. Gärtner, Real-time transcription and separation of drum recordings based on NMF decomposition., in: Int. Conf. Digital Audio Effects, 2014, pp. 187–194.
[15] C. Dittmar, M. Müller, Reverse engineering the Amen break — Score-informed separation and restoration applied to drum recordings, IEEE/ACM Trans. Audio Speech Lang. Process 24 (9) (2016) 1535–1547.
[16] L. Vande Veire, C. De Boom, T. De Bie, Sigmoidal NMFD: Convolutional NMF with saturating activations for drum mixture decomposition, Electronics 10 (3) (2021) 284.
[17] C.-Y. Cai, Y.-H. Su, L. Su, Dual-channel drum separation for low-cost drum recording using non-negative matrix factorization, in: 2021 Asia-Pacific Signal Inf. Process. Association Annu. Summit Conf., 2021, pp. 17–22.
[18] C. Southall, C. Wu, A. Lerch, J. Hockman, MDB Drums – An annotated subset of MedleyDB for automatic drum transcription, in: Late-Breaking/Demos Session, 18th Int. Soc. Music Inf. Retrieval Conf., 2017.
[19] O. Gillet, G. Richard, ENST-drums: An extensive audio-visual database for drum signals processing, in: Proc. 7th Int. Soc. Music Inf. Retrieval Conf., 2006, pp. 156–159.
[20] J. Gillick, A. Roberts, J. Engel, D. Eck, D. Bamman, Learning to groove with inverse sequence transformations, in: Int. Conf. Mach. Learning, Vol. 97, 2019, pp. 2269–2279.

[21] R. Vogl, G. Widmer, P. Knees, Towards multi-instrument drum transcription, in: Int. Conf. Digital Audio Effects, 2018, pp. 57–64.
[22] A. Jansson, E.J. Humphrey, N. Montecchio, R.M. Bittner, A. Kumar, T. Weyde, Singing voice separation with deep U-net convolutional networks, in: Proc. 18th Int. Soc. Music Inf. Retrieval Conf., 2017, pp. 745–751.
[23] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Proc. 18th Int. Conf. Med. Image Comput. Comput. Assisted Intervention, 2015, pp. 234–241.
[24] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: Int. Conf. Mach. Learning, 2015, pp. 448–456.
[25] D. Griffin, J. Lim, Signal estimation from modified short-time Fourier transform, IEEE Trans. Acoust. Speech Signal Process. 32 (2) (1984) 236–243.
[26] T. Kobayashi, T. Tanaka, K. Yatabe, Y. Oikawa, Acoustic application of phase reconstruction algorithms in optics, in: 2022 IEEE Int. Conf. Acoust. Speech Signal Process., 2022, pp. 6212–6216.
[27] A. Liutkus, R. Badeau, Generalized Wiener filtering with fractional power spectrograms, in: 2015 IEEE Int. Conf. Acoust. Speech Signal Process., 2015, pp. 266–270.
[28] E. Vincent, R. Gribonval, C. Févotte, Performance measurement in blind audio source separation, IEEE Trans. Audio Speech Lang. Process. 14 (4) (2006) 1462–1469.
[29] P. Smaragdis, Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs, in: Ind. Compon. Anal. Blind Signal Separation, 2004, pp. 494–499.
[30] D. Lee, H.S. Seung, Algorithms for non-negative matrix factorization, Adv. Neural Inf. Process. Syst. 13 (2000).
[31] C. Raffel, B. McFee, E.J. Humphrey, J. Salamon, O. Nieto, D. Liang, D.P.W. Ellis, mir_eval: A transparent implementation of common MIR metrics, in: Proc. 15th Int. Soc. Music Inf. Retrieval Conf., Vol. 10, 10, 2014, p. 2014.