Music Source Separation With Generative Flow

Ge Zhu, *Graduate Student Member, IEEE*, Jordan Darefsky, Fei Jiang, Anton Selitskiy, *Graduate Student Member, IEEE*, and Zhiyao Duan, *Member, IEEE*

***Abstract*—Fully-supervised models for source separation are trained on parallel mixture-source data and are currently state-of-the-art. However, such parallel data is often difficult to obtain, and it is cumbersome to adapt trained models to mixtures with new sources. Source-only supervised models, in contrast, only require individual source data for training. In this paper, we first leverage flow-based generators to train individual music source priors and then use these models, along with likelihood-based objectives, to separate music mixtures. We show that in singing voice separation and music separation tasks, our proposed method is competitive with a fully-supervised approach. We also demonstrate that we can flexibly add new types of sources, whereas fully-supervised approaches would require retraining of the entire model.**

***Index Terms*—Generative source separation, glow, singing voice separation, music source separation.**

I. Introduction

MUSIC source separation involves separating a music mixture into multiple source signals. It plays an important role in many downstream tasks [1] in music signal processing including melody extraction, lyric recognition and music search. Consequently, many algorithms have been proposed for various problem settings of music source separation in the past decades.

We categorize existing approaches as either supervised or unsupervised based on the availability of separated clean training data. An approach is supervised when any clean sources are available during training; an approach is unsupervised when no such data is available. Further, we define supervised approaches including the following two settings during training:

1) *Fully-supervised approaches:* Both mixtures and their corresponding individual sources are available.
2) *Source-only supervised approaches:* Only clean individual sources are available. This approach is sometimes referred to as unsupervised [2], though we feel that learning to model source data is a form of supervision.

Manuscript received 17 July 2022; revised 15 October 2022; accepted 16 October 2022. Date of publication 3 November 2022; date of current version 18 November 2022. This work was supported by National Science Foundation under Grants 1741472 and 1846184, and in part by the New York State Center of Excellence in Data Science Award. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Huseyin Hacihabiboglu.

Ge Zhu, Jordan Darefsky, Anton Selitskiy, and Zhiyao Duan are with the University of Rochester, Rochester, NY 14627 USA (e-mail: ge.zhu@rochester.edu; jdarefsk@u.rochester.edu; aselitsk@ur.rochester.edu; zhiyao.duan@rochester.edu).

Fei Jiang was with the University of Rochester, Rochester, NY 14627 USA. He is now with the Tencent Technology (Shanghai) Company, Ltd., Shanghai 518054, China (e-mail: flyjiang92@gmail.com).

Digital Object Identifier 10.1109/LSP.2022.3219355

In recent literature, fully-supervised approaches achieve the state-of-the-art performance in source separation tasks. These approaches often involve training a model (e.g., a deep neural network) on parallel mixture-source data to map mixtures to their underlying sources (or their corresponding masks) [3]. Such data can be naturally recorded or crafted to follow the real distribution of mixtures and their underlying sources. For example, the MUSDB18 dataset [4] consists of songs along with their corresponding individual sources. Such datasets, however, are difficult to construct or obtain. An alternative approach involves synthesizing training mixtures by randomly mixing clean training sources. This fully-supervised approach is referred to specifically as *synthetic full-supervision*.

In the source-only supervised setting, a common approach involves first using the individual sources to learn models of the source domains and then using these models to perform separation of mixtures during inference. Notably, the non-negative matrix factorization (NMF) [5], [6] based source separation is built upon the concept of a signal dictionary. Other approaches learn a probabilistic model [7], [8], [9] for each source, and then separate the mixture with a signal reconstruction objective. More recently, the generative source separation framework [10] has gained much attention with the emergence of expressive deep generative models and various optimization techniques; for example, implicit generative models such as generative adversarial networks (GANs) have been used to train source priors [10], [11], [12] which can then be used to separate sources with gradient-based methods [2], [10]. Jayaram and Thickstun [13] train an explicit prior and sample with Langevin dynamics to perform source separation in the image domain; however, such sampling methods can be slow even with parallel sampling [14].

In this paper, we focus on a source-only supervised, generative approach to music source separation. More specifically, we 1) train flow-based generators to model the spectrograms of various instruments; and 2) apply gradient-based optimization to separate sources at inference. Compared to fully-supervised methods, our approach only needs access to clean individual sources at train time; practically, it is easier to obtain individual source data than paired mixture-source data. Although synthetic full-supervision approach is shown to outperform traditional data augmentation [15], [16] techniques, it requires a large amount of combinations of the sources [17], [18]. Compared to existing source-only supervised, generative methods, we find that using flow-based models provides two advantages in particular. First, flow-based models are invertible and thus have zero representation error; this is not the case for GAN-based

generative priors [19]. This representation capability is beneficial for optimizing a reconstruction objective during separation. Second, we find empirically that the separation process converges quickly and that our approach is faster than current sampling-based methods.

On singing voice separation and music source separation tasks, we show that our proposed method outperforms current source-only separation approaches and achieves competitive performance with one of the fully-supervised methods. Furthermore, we demonstrate that we are able to flexibly add a new source. In contrast, in fully-supervised systems, to separate new sources, it is required to either alter the entire network architectures or prepare paired target source tracks and accompaniment tracks following one-versus-all training paradigm [20]. We make the code[1] publicly available.

II. Background

A. Generative Source Separation

Source separation involves separating a mixture $\mathbf{x}$ into n individual sources $\mathbf{s}_i$. Under an instantaneous mixing setting [21], we have:

$$\mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{s}_i, \tag{1}$$

where α_i is the mixing coefficient. For simplicity, we assume $\alpha_i = 1$. In probabilistic modeling framework [9], [10], [13], we can assume that different sources have different statistical behavior. Specifically,

$$\mathbf{s}_i \sim p_G(\mathbf{s}_i), \tag{2}$$

$$\mathbf{x}|\mathbf{s}_1, \ldots, \mathbf{s}_n \sim p_{mix}\left(\mathbf{x} \middle| \sum_{i=1}^{n} \mathbf{s}_i\right), \tag{3}$$

where p_G models sources $\mathbf{s}_i$, and p_{mix} models the mixture conditioned on the sum of sources. p_{mix} is often fixed and chosen explicitly to model the noise between the sum of clean sources and the final mixture. To perform source separation, one could first train models p_G to model the source distributions. Then, to perform separation, sources $\mathbf{s}_i$ could be found to maximize the above maximum a posteriori (MAP) objective, using a preferred method of optimization. This method would be considered source-only supervised, as only individual sources are seen during training.

B. Flow Models

Normalizing flow is a generative model that transforms a random variable $\mathbf{z}$ with a simple distribution $p_{\mathbf{z}}(\mathbf{z})$ (Gaussian in our case) into a target random variable $\mathbf{y} \sim p_{\mathbf{y}}(\mathbf{y})$ through an invertible function f_θ. Using the change of variables formula, the log probability of $\mathbf{y}$ can be written as:

$$\log p_{\mathbf{y}}(\mathbf{y}) = \log p_{\mathbf{z}}(\mathbf{z}) - \log \left| \det \frac{\partial f_\theta(\mathbf{z})}{\partial \mathbf{z}} \right|. \tag{4}$$

Often, f_θ is a composition of neural invertible flow layers, for which the Jacobians are efficient to compute. This allows for efficient computation of the total log-determinant term in (4). Since $\mathbf{z}$ is Gaussian, the $\log p_{\mathbf{z}}(\mathbf{z})$ term can directly be computed; thus, f_θ can be trained to maximize the log probability of data, $\log p_{\mathbf{y}}(\mathbf{y})$. In the case of source separation, we can use flow models as prior distributions p_G.

[1]Open source code: github.com/gzhu06/GenerativeSourceSeparation.

III. Proposed Method

A. Glow Priors

Having witnessed the success in solving inverse problems with flow-based 2D image priors [19], we use Glow [22] as our generative model backbone to learn source priors from 2D audio magnitude spectrograms. One motivation for modeling music priors in the spectral domain is that the magnitude spectrograms of singing voice and background music have different structures, which may facilitate separation; singing voice spectrograms tend to be sparse while background music spectrograms tend to be low rank and change more slowly [23]. Note that the magnitude spectrogram of the mixture is not the exact sum of those of the sources due to phase differences; however, the sum is a good approximation as shown in NMF-based methods [24].

Fig. 1 illustrates the training and inference (i.e., separation) process of our proposed flow-based model. For the task of music source separation, we train a set of independent Glow priors (named *InstGlow*), one for each source.

We adapt the Glow [22] as the flow-based generator backbone and use $\mathbf{z}_i \sim \mathcal{N}(0, \mathbf{I})$ as the latent prior. Our glow generator consists of a squeeze layer, 12 flow blocks, and an unsqueeze layer. The squeeze and unsqueeze operation follows the design in [25]. In each step of flow, we use an activation normalization layer, an invertible 1x1 convolution layer [25], and an affine coupling layer in [26] without local conditioning.

B. Inference

In the separation stage, we assume that we have knowledge of sources presented in the mixture and apply *all* of the predefined source priors to separate corresponding components. As mentioned in Section II-A, we use MAP [8] as the separation objective and apply an iterative optimization to separate predefined sources:

$$\hat{\mathbf{s}}_1, \ldots, \hat{\mathbf{s}}_n = \underset{\mathbf{s}_1, \ldots, \mathbf{s}_n}{\operatorname{argmax}} \log p(\mathbf{s}_1, \ldots, \mathbf{s}_n|\mathbf{x}) \tag{5}$$

$$= \underset{\mathbf{s}_1, \ldots, \mathbf{s}_n}{\operatorname{argmax}} \log p(\mathbf{x}|\mathbf{s}_1, \ldots, \mathbf{s}_n) + \sum_{i=1}^{n} \log p(\mathbf{s}_i), \tag{6}$$

where $\mathbf{x}$ is the observed mixture. In (6) we assume statistical independence among all source tracks. In the above MAP formulation, we can also optimize over latent variable $\mathbf{z}_i$ rather than $\mathbf{s}_i$, as there is a bijection between them from the Glow model, $\mathbf{s}_i = f_\theta^i(\mathbf{z}_i)$.

To model $p(\mathbf{x}|\mathbf{s}_1, \ldots, \mathbf{s}_n)$ in the instantaneous mixing, we assume an independent additive residual noise $\mathbf{n}$ over the sum of the sources [8], i.e., $p(\mathbf{x}|\mathbf{s}_1, \ldots, \mathbf{s}_n) = p(\mathbf{x} - \sum_i^n \mathbf{s}_i) = p(\mathbf{n})$. We assume that the spectrogram magnitude of the residual noise follows a Poisson distribution, then the log-likelihood of the

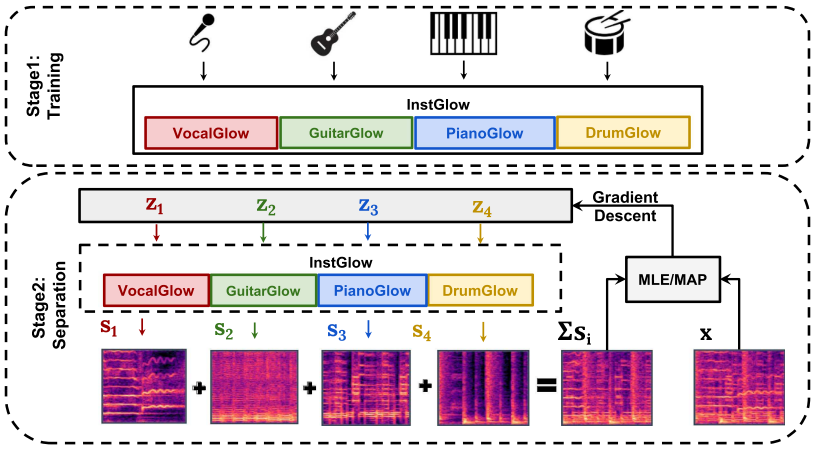

Fig. 1. Diagram for proposed flow-based generative source separation. Stage 1: training source prior models with instrument-specific unconditional models (*InstGlow*), one for each source. Stage 2: separating sources by searching the optimal latent code $\{z_i\}$ to optimize an MLE or an MAP objective.

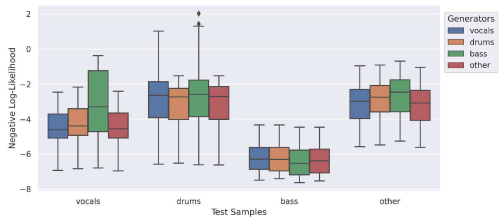

Fig. 2. Boxplots of NLLs of pre-trained 'vocals,' 'drums,' 'bass' and 'other' Glow generators (different colors) on a total of 100 one-minute test audio pieces from the four source categories (different columns). Each data point is the NLL of one generator on one test audio piece.

mixture (first term in (6)) becomes equivalent to the negative generalized KL-divergence. Because we are using flow-based models, the second term in (6) (the exact log-likelihood of the source priors) can be computed directly. We optimize the objective (6) by searching the latent space within the support of pre-trained generators [19], [27]. Since both terms in (6) are differentiable with respect to $\mathbf{z}_i$, we perform this optimization with gradient descent. During optimization, we initialize $\mathbf{z}_i = 0$ to bias the latent codes towards zero in order to align with the target $\mathbf{z}_i$ after the source specific priors are trained. This bias could be viewed as a simpler prior, with the benefit of being more robust to high out-of-distribution likelihoods. After finding the optimal latent codes $\mathbf{z}_i$, we can compute the spectrograms of sources using the Glow models with $\mathbf{s}_i = f_\theta^i(\mathbf{z}_i)$. Eventually, we synthesize the source waveforms using inverse-STFT with the recovered source spectrogram and the mixture phase.

C. Prior Reweighting

Previous works on speech enhancement [28], audio source separation [29], and image inpainting tasks [19], [27] have found that flow-based models tend to assign high probability density to some out-of-distribution data while assigning low density to some in-distribution data. We find similar phenomena in our experiments. We computed the negative log-likelihoods (NLLs) of the four pretrained Glow priors on the 100 one-minute source tracks from the test partition in MUSDB18 shown in Fig. 2. We observe that the estimated NLLs are highly correlated and overlapping with each other for the same samples, suggesting that the pre-trained instrument generators are not discriminative enough in differentiating unseen instruments at inference.

To address this concern, we empirically re-weigh the prior term in (6) with coefficient $\gamma \in [0, 1]$, initially proposed in [19]. We can discard the prior term in (6) by choosing $\gamma = 0$ and arrive at a maximum likelihood estimation (MLE) objective. Notice that, we keep the zero initialization of $\mathbf{z}_i$ in the MLE approach to avoid trivial solutions for (6) without the prior term constraints. Also note that in this MLE objective we effectively treat our Glow model as an implicit generator [30], though in our case sources are deterministically related to the latents.

IV. Experiments

A. Dataset

We train the source priors for `vocals`, `bass`, `drums` and `other` using the train subset of the MUSDB18 and `guitar` and `piano` from the train subset of Slakh2100 (i.e. we do not use `bass`, `drums` source tracks from Slakh2100 to train the source priors). For preprocessing, we use the mono channel and downsample the tracks into 22.05 kHz and split them into 5-second non-silent segments. We use spectrograms with 1024-point FFT size and 256-point hop size as input features.

For MUSDB18 evaluation, we test both multi-instrument separation and singing-accompaniment separation. To construct accompaniment tracks, we sum the separated non-vocal sources. For Slakh2100-submix evaluation, we select and remix the subtracks of the top four instrument categories (`piano`, `bass`, `guitar` and `drums`) from the original Slakh2100. We split the test portion into one-minute segments to fit into memory. We measure the global signal-to-distortion ratio (SDR) defined in Music Demixing Challenge [36] of each segment with museval toolbox [37] to evaluate separation performance. Following [36], we remove silent segments in the test data, where SDR is undefined.

B. Baselines and Training

We use Conv-TasNet, Demucs(v2) [16], open-unmix [20] and Wave-U-Net [35] as fully-supervised baseline systems.

TABLE I
COMPARISON OF SOURCE SEPARATION SYSTEMS WITH MEDIAN SDR (DB) ACROSS TRACKS ON THREE SETTINGS OF TWO TEST SETS

Method	Neural Networks	Supervision	MUSDB18-22.05kHz [4]					Slakh2100-submix [31]			
			Vocals	Bass	Drums	Other	Acc.	Bass	Drums	Guitar	Piano
INSTGLOW-MLE (OURS)	Glow	Source-only	**3.92**	**2.58**	**3.85**	**2.37**	**9.82**	**1.54**	**6.14**	**1.85**	**0.80**
INSTGLOW-MAP (OURS)	Glow	Source-only	3.66	2.51	3.70	1.99	9.52	1.39	5.95	1.51	0.51
LQ-VAE [32]	VQ-VAE	Source-only	0.16	–	–	–	4.47	–	–	–	–
GAN-PRIOR [2]	SpecGAN	Source-only	-0.44	0.48	-0.40	0.32	4.29	0.09	0.85	-0.01	-0.42
Conv-TasNet [33]	TCN	Full	7.00	4.19	5.25	3.94	12.84	4.97	9.95	–	–
Demucs (v2) [34]	U-Net	Full	**7.14**	**5.50**	**6.74**	**4.16**	**12.94**	**5.48**	**10.21**	–	–
Open Un-mix [20]	BiLSTM	Full	6.86	4.88	6.35	3.86	12.75	4.66	8.64	–	–
Wave-U-Net [35]	Wave-U-Net	Full	5.06	2.63	3.74	1.95	7.02	0.01	3.91	–	–

Grey cells indicate that the system is unable to separate that source type unless it is retrained from scratch using one-versus-all paradigm and on the same kind of sources as the test set.

We directly use the authors' pre-trained models trained on MUSDB18. Since MUSDB18 does not contain guitar and piano source tracks, these models cannot separate such sources in Slakh2100-submix without preparing paired source-mixture data and retraining from scratch.

We also compare our model to source-only supervision systems, using GAN-prior [2] and LQ-VAE [32] as baselines. During separation, we apply projected gradient descent (PGD) on the reconstruction objective to search for the latent codes. For LQ-VAE, we apply the authors' method as-is.

For source priors training, we use the Adam [38] optimizer at a learning rate of 1e-4 for 1000 epochs. During separation, we use the Adam optimizer at a learning rate of 0.01 for 150 iterations. Each iteration takes 0.3 seconds during evaluation on NVIDIA 2080Ti GPU. We also conduct an ablation study on MLE and MAP optimization objectives.

C. Results

We start by comparing different variants of our proposed method, shown in the first two rows in Table I. We observe that the InstGlow with the MLE objective achieves the best results in terms of SDR across all of the tasks. For the InstGlow models, the MLE objective that only uses KL-divergence achieves better performance than MAP estimation; this differs from results in image domain [27], where MAP estimation shows better performance. One reason may be due to the independence assumption of instrument sources in MAP estimation in (6). Another reason may result from the fact that deep generative models lack of discriminative abilities to distinguish data of other classes [39], especially for the high dimensional data [39].

When comparing to source-only supervision systems shown in the middle rows of Table I, our proposed InstGlow significantly outperforms the other systems. While the results for the GAN-prior [2] are perhaps surprisingly poor, our findings are consistent with the authors of LQ-VAE, who report that the GAN-prior performs poorly on the drum-piano toy dataset. We suspect the LQ-VAE baseline performs poorly due to the relatively small dataset (MUSDB18) on which we trained it. Note that in a similar method to LQ-VAE, [40], [41] uses a pre-trained Jukebox VQ-VAE-model (on 1.2 million songs) [42], and El Amri et al. [40] achieved comparable performance to fully-supervised methods. However, for a fair comparison with our method and the GAN-prior baseline, we only pre-train LQ-VAE on the MUSDB dataset. Also note that LQ-VAE is only able to separate two sources, due to its training paradigm.

We compare our best performing system, InstGlow-MLE, with the fully-supervised baselines. On the MUSDB18 test set, a statistical test shows that InstGlow-MLE significantly outperforms Wave-U-Net in other, and achieves comparable results in bass and drums, although there is still a large gap behind Wave-U-Net in vocals. By listening to the separated vocal samples from InstGlow-MLE, we found that they contain more interference from other sources, which could be eased by adding regularization such as coherence loss [43].

On the Slakh2100-submix test set, we can only compare with fully-supervised models on the bass and drums sources, as their models are trained on MUSDB18 which does not contain guitar and piano tracks. We observe that InstGlow-MLE achieves better results than Wave-U-Net without retraining on bass and drums sources from Slakh2100 dataset, showing its generalization ability to new datasets. Finally, we additionally train guitar and piano priors using *only* the source data from the Slakh2100 training subset and apply them to separate the corresponding tracks in the Slakh2100-submix test subset. We find that the separation performance of guitar and piano is similar to that of the bass source but lower than that of the drum source; this relatively weak performance may be due to the significant pitch range overlap of guitar and piano [31]. We find this training paradigm promising given the above observations, as InstGlow only requires instrument data and can be used to separate sources undefined in MUSDB18, which is not easily feasible in fully-supervised models.

V. CONCLUSION

In this paper, we employed flow-based generators for music source separation in the source-only supervision setting. To the best of our knowledge, we are the first to report successful separation results in this setting on benchmark separation tasks, achieving significantly better results than other source-only supervised methods. Future work is to bridge the performance gap between our method and fully-supervised approaches by potentially scaling up with more instrument data in the wild as well as to extend it to more general settings.

REFERENCES

[1] E. Manilow, P. Seetharman, and J. Salamon, "Open source tools & data for music source separation," 2020. [Online]. Available: https://source-separation.github.io/tutorial

[2] V. Narayanaswamy, J. J. Thiagarajan, R. Anirudh, and A. Spanias, "Unsupervised audio source separation using generative priors," in *Proc. Annu. Conf. Int. Speech Commun. Assoc.*, INTERSPEECH, 2020, vol. 2020, pp. 2657–2661.

[3] N. Mohammadiha, P. Smaragdis, and A. Leijon, "Supervised and unsupervised speech enhancement using nonnegative matrix factorization," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 10, pp. 2140–2151, Oct. 2013.

[4] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," Dec. 2017.

[5] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[6] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 3, pp. 1066–1074, Mar. 2007.

[7] S. Roweis, "One microphone source separation," *in Proc. Adv. Neural Inf. Process. Syst.*, 2000, vol. 13.

[8] L. Benaroya, F. Bimbot, and R. Gribonval, "Audio source separation with a single sensor," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 1, pp. 191–199, Jan. 2006.

[9] A. Ozerov, P. Philippe, F. Bimbot, and R. Gribonval, "Adaptation of Bayesian models for single-channel source separation and its application to voice/music separation in popular songs," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 5, pp. 1564–1578, Jul. 2007.

[10] Y. C. Subakan and P. Smaragdis, "Generative adversarial source separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 26–30.

[11] D. Stoller, S. Ewert, and S. Dixon, "Adversarial semi-supervised audio source separation applied to singing voice extraction," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 2391–2395.

[12] Q. Kong, Y. Xu, P. J. B. Jackson, W. Wang, and M. D. Plumbley, "Single-channel signal separation and deconvolution with generative adversarial networks," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 2747–2753.

[13] V. Jayaram and J. Thickstun, "Source separation with deep generative priors," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2020, pp. 4724–4735.

[14] V. Jayaram and J. Thickstun, "Parallel and flexible sampling from autoregressive models via langevin dynamics," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2021, pp. 4807–4818.

[15] S. Uhlich et al., "Improving music source separation based on deep neural networks through data augmentation and network blending," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2017, pp. 261–265.

[16] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," 2019, *arXiv:1911.13254*.

[17] X. Song, Q. Kong, X. Du, and Y. Wang, "CatNet: Music source separation system with mix-audio augmentation," 2021, *arXiv:2102.09966*.

[18] Q. Kong, Y. Cao, H. Liu, K. Choi, and Y. Wang, "Decoupling magnitude and phase estimation with deep ResUNet for music source separation," 2021, *arXiv:2109.05418*.

[19] M. Asim, M. Daniels, O. Leong, A. Ahmed, and P. Hand, "Invertible generative models for inverse problems: Mitigating representation error and dataset bias," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2020, pp. 399–409.

[20] F.-R. Stoter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-unmix-a reference implementation for music source separation," *J. Open Source Softw.*, vol. 4, 2019, Art. no. 1667.

[21] A. Ozerov and C. Févotte, "Multichannel nonnegative matrix factorization in convolutive mixtures for audio source separation," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 3, pp. 550–563, Mar. 2010.

[22] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," *in Proc. Adv. Neural Inf. Process. Syst.*, 2018, vol. 31.

[23] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson, "Singing-voice separation from monaural recordings using robust principal component analysis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2012, pp. 57–60.

[24] T. Virtanen, A. T. Cemgil, and S. Godsill, "Bayesian extensions to non-negative matrix factorisation for audio signal modelling," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2008, pp. 1825–1828.

[25] J. Kim, S. Kim, J. Kong, and S. Yoon, "Glow-TTS: A generative flow for text-to-speech via monotonic alignment search," *in Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 8067–8077.

[26] R. Prenger, R. Valle, and B. Catanzaro, "Waveglow: A flow-based generative network for speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2019, pp. 3617–3621.

[27] J. Whang, Q. Lei, and A. Dimakis, "Solving inverse problems with a flow-based noise model," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2021, pp. 11146–11157.

[28] Y. Shi, H. Chen, Z. Tang, L. Li, D. Wang, and J. Han, "Can we trust deep speech prior," in *Proc. IEEE Spoken Lang. Technol. Workshop*, 2021, pp. 742–749.

[29] M. Frank and M. Ilse, "Problems using deep generative models for probabilistic audio source separation," in *Proc. "I Can't Believe It's Not Better!" NeurIPS 2020 Workshop*, 2020, pp. 53–59.

[30] S. Mohamed and B. Lakshminarayanan, "Learning in implicit generative models," 2016, *arXiv:1610.03483*.

[31] E. Manilow, G. Wichern, P. Seetharaman, and J. Le Roux, "Cutting music source separation some Slakh: A dataset to study the impact of training data quality and quantity," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust.*, 2019, pp. 45–49.

[32] M. Mancusi, E. Postolache, M. Fumero, A. Santilli, L. Cosmo, and E. Rodolá, "Unsupervised source separation via Bayesian inference in the latent domain," 2021, *arXiv:2110.05313*.

[33] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, Aug. 2019.

[34] A. Défossez, "Hybrid spectrogram and waveform source separation," in *Proc. ISMIR Workshop Music Source Separation*, 2021.

[35] S. E. Daniel Stoller and S. Dixon, "Wave-U-Net: A multi-scale neural network for end-to-end audio source separation," in *Proc. 19th Int. Soc. Music Inf. Retrieval Conf. (ISMIR)*, 2018, pp. 334–340.

[36] Y. Mitsufuji et al., "Music demixing challenge 2021," *Front. Signal Process.*, vol. 1, 2022.

[37] F.-R. Stöter, A. Liutkus, and N. Ito, "The 2018 signal separation evaluation campaign," in *Proc. Int. Conf. Latent Variable Anal. Signal Separation*, Springer, 2018, pp. 293–305.

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[39] E. Nalisnick, A. Matsukawa, Y. W. Teh, D. Gorur, and B. Lakshminarayanan, "Do deep generative models know what they don't know?" 2018, *arXiv:1810.09136*.

[40] W. Z. El Amri, O. Tautz, H. Ritter, and A. Melnik, "Transfer learning with jukebox for music source separation," in *Proc. IFIP Int. Conf. Artif. Intell. Appl. Innovations*, 2022, pp. 426–433.

[41] E. Manilow, P. O'Reilly, P. Seetharaman, and B. Pardo, "Source separation by steering pretrained music models," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2022, pp. 126–130.

[42] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," 2020, *arXiv:2005.00341*.

[43] Y. Tian, C. Xu, and D. Li, "Deep audio prior: Learning sound source separation from a single audio mixture," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020.