

Unsupervised Source Separation via Bayesian Inference in the Latent Domain

Michele Mancusi^{*1}, Emilian Postolache^{*1}, Giorgio Mariani¹, Marco Fumero¹, Andrea Santilli¹, Luca Cosmo^{2,3}, Emanuele Rodolà¹

¹Sapienza University of Rome, ²Ca' Foscari University of Venice, ³University of Lugano

mancusi@di.uniroma1.it

Abstract

State of the art audio source separation models rely on supervised data-driven approaches, which can be expensive in terms of labeling resources. On the other hand, approaches for training these models without any direct supervision are typically high-demanding in terms of memory and time requirements, and remain impractical to be used at inference time. We aim to tackle these limitations by proposing a simple yet effective unsupervised separation algorithm, which operates directly on a latent representation of time-domain signals. Our algorithm relies on deep Bayesian priors in the form of pre-trained autoregressive networks to model the probability distributions of each source. We leverage the low cardinality of the discrete latent space, trained with a novel loss term imposing a precise arithmetic structure on it, to perform exact Bayesian inference without relying on an approximation strategy. We validate our approach on the Slakh dataset [1], demonstrating results in line with state of the art supervised approaches while requiring fewer resources with respect to other unsupervised methods.

Index Terms: Signal separation, Autoregressive generative models, Bayesian inference, Unsupervised learning

1. Introduction

Generative models have reached promising results in a wide range of domains, including audio, and can be used to solve different tasks in unsupervised learning. A relevant problem in the musical domain is the task of source separation of different instruments. Given the sequential nature of music and the high variability of rhythm, timbre and melody, autoregressive models [2] represent a popular and effective choice to process data on such domain, showcasing high multi-modality in the modeled probability distributions. The widely adopted WaveNet autoregressive architecture [3] works in the temporal domain. Given that audio signals are typically sampled at high frequencies (e.g. 44 kHz) for music, the choice of modeling the data distribution directly in the time domain leads to short contexts being captured by neural computations and quick saturation of memory. Nevertheless, existing unsupervised approaches for source separation operate in the time domain [4]. In order to capture longer contexts and to reduce memory burden, different quantization schemes have been introduced for autoregressive models [5, 6], where chunks in time are mapped to sequences of latent tokens belonging to a small vocabulary. OpenAI's Jukebox [7] follows this approach and excels as an architecture that can capture very long contexts, generating highly consistent tracks. Leveraging the useful properties of this architecture, we propose a novel approach to unsupervised source separation that works directly on quantized latent domains.

Our contributions can be summarized as follows:

1. We perform source separation applying exact Bayesian inference directly in the latent domain, exploiting the relative small size of the latent dictionary. We do not rely on any approximation strategy, such as variational inference or Langevin dynamics.
2. We introduce LQ-VAE: a quantized autoencoder trained with a novel loss that imposes an algebraic structure on the discrete latent space. This allows us to alleviate noisy and distorted samples which arise from a vanilla quantization approach.

2. Related work

The problem of source separation has been classically tackled in an unsupervised fashion [8], where the sources to be separated from a mixture signal are unknown [9]. With the advent of deep learning, most source separation tasks applied to musical data started relying on supervised learning, training models on data with known correspondence between sources. Recently, following the success of deep generative models, there has been a renewed interest in unsupervised methods.

2.1. Supervised source separation

Supervised source separation aims to map high dimensional observations of audio mixtures to a smaller dimensional space and apply, explicitly or implicitly, a mask to filter out the sources from the latent representation of the mixtures in a supervised way. Most of these works can be divided into *frequency-domain* or *waveform-domain* approaches. The former [10] operate on the spectral representation of the input mixtures. This line of works has highly benefited from the incoming of deep learning techniques from simple fully connected networks [11], LSTM [12], and CNN coupled with recurrent approaches [13, 14]. Recent approaches such as [15] and [16] hold the state of the art in music source separation over the dataset MUSDB18 [17], by respectively extending the conditional U-net architecture of [18] to multi-source separation, and by exploiting multi-dilated convolution that applies different dilation factors in each layer to model different resolutions simultaneously. In contrast, waveform domain approaches process the mixtures directly in the time domain to overcome phase estimation, which is necessary when converting the signal from the frequency domain. The method of [19] performs in line with the state of the art by extending a WaveNet-like architecture, coupled with an LSTM in the latent space.

The main limitation of these state-of-the-art methods for audio source separation is that they require large amounts of fully separated, labeled data to perform the training.

* Equal contribution

2.2. Unsupervised source separation

Recent approaches in unsupervised source separation leverage self-supervised learning. A prominent baseline is MixIt [20], which trains a model by trying to separate sources from a mixture of mixtures. Although promising, such model suffers from the *over-separation* problem, where at test time a number of sources that is greater than those present in the mixture are estimated. As such, stems can be split across different output tracks. Generative approaches instead overcome this problem by imposing that a model should output an individual stem.

Closer to our work, [21] proposes to leverage generative priors in the form of GANs trained on individual sources. They use projected gradient descent optimization to search in the source-specific latent spaces and effectively recover the constituent sources in the time domain. Although promising, GANs suffer from modal collapse, so their performance is limited in the musical domain, where variability is abundant. [4] proposes to use Langevin dynamics on the global log-likelihood of the audio sequences to parallelize the sampling procedure of autoregressive models used as Bayesian priors. This approach produces good results but with a high computational cost due to the need of training distinct models for each noise level, and due to the costly optimization procedure in the time domain.

Differently, our inference procedure has much lower computational and memory requirements, allowing us to efficiently run the model on a single GPU. In addition, we can perform exact Bayesian inference without relying on an approximation scheme of the posterior (e.g., its score).

3. Background

In this section we briefly introduce the background concepts necessary to understand our architecture, which builds upon [7]. The overall architecture can be split into two parts: (i) a quantization module mapping the input sequences to a discrete latent space, and (ii) an autoregressive prior (one per source) which models the distribution of a given source in the discrete latent space. We point the reader to [7] for a deeper understanding.

3.1. Quantization module

Let us consider an input sequence $\mathbf{x} = x_1, \dots, x_T \in [-1, 1]^T$ of length T , which represents a normalized waveform in the time domain. In order to be representative of an expressive portion of the audio sequence, T should be large. However, due to the complexity of modern neural architectures, choosing a large enough value of T is not always feasible. To reduce the dimensionality of the space one can leverage the VQ-VAE architecture [5] to map large continuous sequences in the time domain to smaller sequences in a discrete latent domain. A VQ-VAE is composed of three blocks:

- A convolutional encoder $E : [-1, 1]^T \rightarrow \mathbb{R}^{S \times D}$, with $S \ll T$, where S is the length of the latent sequence and D denotes the number of channels;
- A bottleneck block $B = B_I \circ B_Q$ where $B_Q : \mathbb{R}^{S \times D} \rightarrow \mathcal{C}^S \subseteq \mathbb{R}^{S \times D}$ is a vector quantizer, mapping the sequence of latent vectors $\mathbf{h} = \mathbf{h}_1, \dots, \mathbf{h}_S = E(\mathbf{x})$ into the sequence of nearest neighbors contained in a codebook $\mathcal{C} = \{\mathbf{e}_k\}_{k=1}^K$ of learned latent codes, and $B_I : \mathcal{C}^S \rightarrow [K]^S$ is an indexer mapping the codes $\mathbf{e}_{k_1}, \dots, \mathbf{e}_{k_S}$ into the associated codebook indices $z_1 = k_1, \dots, z_S = k_S$. Note that since B_I is bijective, the codes \mathbf{e}_k and their indices k are semantically equivalent, but we shall use the

term ‘codes’ for the vectors in \mathcal{C} and ‘latent indices’ for the associated integers;

- A decoder $D : [K]^S \rightarrow [-1, 1]^T$ mapping the discrete sequence back into the time domain.

The VQ-VAE is trained by minimizing the composite loss:

$$\mathcal{L}_{\text{VQ-VAE}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{codebook}} + \beta \mathcal{L}_{\text{commit}}, \quad (1)$$

where:

$$\mathcal{L}_{\text{rec}} = \frac{1}{T} \sum_t \|x_t - D(z_t)\|_2^2 \quad (2)$$

$$\mathcal{L}_{\text{codebook}} = \frac{1}{S} \sum_s \|\text{sg}[\mathbf{h}_s] - \mathbf{e}_{z_s}\|_2^2 \quad (3)$$

$$\mathcal{L}_{\text{commit}} = \frac{1}{S} \sum_s \|\mathbf{h}_s - \text{sg}[\mathbf{e}_{z_s}]\|_2^2, \quad (4)$$

where sg is the stop-gradient operator and β is the commitment loss weight. The losses $\mathcal{L}_{\text{codebook}}$ and $\mathcal{L}_{\text{commit}}$ update the entries of the codebook \mathcal{C} during the training procedure. In addition, we introduce a novel loss term \mathcal{L}_{lin} , described in Section 4.2, which imposes a precise algebraic structure on the latent space, facilitating the task of source separation.

3.2. Latent autoregressive priors

Once the VQ-VAE is trained, time domain data $\mathbf{x} \sim p^{\text{data}}$ can be mapped to latent sequences \mathbf{z} . Autoregressive priors $p(\mathbf{z}) = p(z_1)p(z_2|z_1) \dots p(z_S|z_{S-1}, \dots, z_1)$ can then be learned over the discrete domain. In this work, the autoregressive models are based on a deep scalable Transformer architecture as in [7]. In order to generate new time-domain examples, sequences of latent indices are sampled from $p(\mathbf{z})$ via ancestral sampling and then mapped back to the time domain via the decoder of the VQ-VAE.

4. Method

The proposed algorithm is composed of two parts. A first *separation phase* in the latent domain, in which we sequentially sample from an exact posterior on discrete indices. A following *rejection sampling procedure* based on a (scaled) global posterior conditioned on the separation results, which we use to sort the proposed solutions and select the most promising one.

4.1. Latent Bayesian source separation

Our task is to separate a mixture signal $\mathbf{m} = \frac{1}{2}\mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2$ into $\mathbf{x}_1 \sim p_1^{\text{data}}$ and $\mathbf{x}_2 \sim p_2^{\text{data}}$, where p_1^{data} and p_2^{data} represent the distributions of each instrument class in the time domain. In a Bayesian framework, a candidate solution $\mathbf{x} = \mathbf{x}_1, \mathbf{x}_2$ is distributed according to the posterior $p(\mathbf{x}_1, \mathbf{x}_2 | \mathbf{m}) \propto p_1^{\text{model}}(\mathbf{x}_1)p_2^{\text{model}}(\mathbf{x}_2)p(\mathbf{m} | \mathbf{x}_1, \mathbf{x}_2)$, where the priors $p_1^{\text{model}}, p_2^{\text{model}}$ are typically deep generative models and the likelihood $p(\mathbf{m} | \mathbf{x}_1, \mathbf{x}_2)$ is parameterized as $p(\mathbf{m} | \frac{1}{2}\mathbf{x}_1 + \frac{1}{2}\mathbf{x}_2)$.

In this work, we follow the Bayesian approach but we work in the latent domain. After training the VQ-VAE on an *arbitrary* audio dataset (with samples lying also outside p_1^{data} and p_2^{data}), we learn two latent autoregressive priors $p_1(\mathbf{z}_1)$ and $p_2(\mathbf{z}_2)$ over the two instrument classes. The priors do not require any correspondence between the sources, being trained in a completely unsupervised setting. We assume the two priors to be independent, i.e. $p(\mathbf{z}) = p(\mathbf{z}_1, \mathbf{z}_2) = p_1(\mathbf{z}_1)p_2(\mathbf{z}_2)$. Therefore, for each step $s \in [S]$, we can

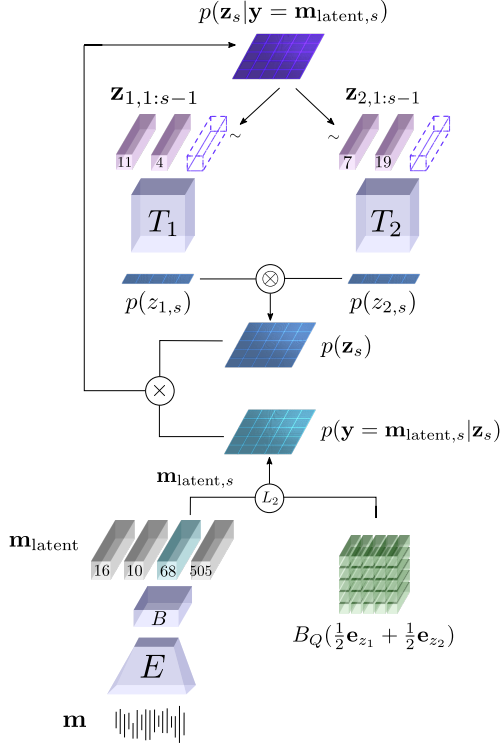


Figure 1: In our method, two autoregressive priors T_1 and T_2 are trained on different instrument sources in the latent domain. At each step s they provide the joint prior $p(\mathbf{z}_s)$. The prior is combined with a σ -isotropic Gaussian likelihood $p(y = \mathbf{m}_{\text{latent},s} | \mathbf{z}_s) = \mathcal{N}(\mathbf{m}_{\text{latent},s} | B_Q(\frac{1}{2}\mathbf{e}_{z_1} + \frac{1}{2}\mathbf{e}_{z_2}), \sigma^2 \mathbf{I})$ in order to compute the posterior $p(\mathbf{z}_s | y = \mathbf{m}_{\text{latent},s})$ from which new samples are drawn.

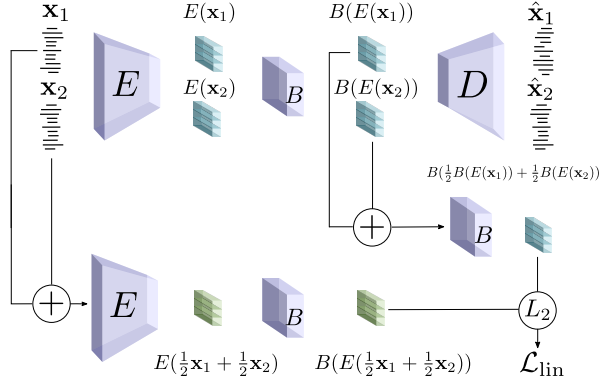


Figure 2: Training scheme of the LQ-VAE: reconstructions $\hat{\mathbf{x}}_1, \hat{\mathbf{x}}_2$ are obtained from input pairs $\mathbf{x}_1, \mathbf{x}_2$ as in the VQ-VAE, leading to the loss $\mathcal{L}_{\text{VQ-VAE}}$ (Eq. (1)). To this loss we add the post-quantization linearization loss \mathcal{L}_{lin} (Eq. (8)), that is computed by matching time-domain sums with latent vector sums.

compute the posterior distribution $p(z_{1,s}, z_{2,s} | \mathbf{z}_{1:s-1}, \mathbf{y}) \propto p_1(z_{1,s} | \mathbf{z}_{1,1:s-1}) p_2(z_{2,s} | \mathbf{z}_{2,1:s-1}) p(\mathbf{y} | z_{1,s}, z_{2,s}, \mathbf{z}_{1:s-1})$.

The random variable $\mathbf{y} = f(\mathbf{m})$ is a function of the mixture \mathbf{m} . One can choose to model \mathbf{y} in multiple ways; a naive

approach is to choose f as the identity and set $\mathbf{y} = \mathbf{m}$, thus computing the likelihood function directly in the time domain. This approach, however, requires the decoding of at least $2K$ possible latent indices in order to locally compare the mixture \mathbf{m} with the hypotheses $z_{1,s}$ and $z_{2,s}$. Note that this corresponds to a lower bound, given that the convolutional nature of the decoder requires a larger past context to produce meaningful results. Differently, we propose to define \mathbf{y} in the latent domain, setting $\mathbf{y} = B_Q(E(\mathbf{m})) := \mathbf{m}_{\text{latent}}$. This approach is preferable since it does not require decoding the hypotheses at each step s , resulting in lower memory usage and computation time. Our method benefits from the choice of operating in the latent space, thanks to the relatively small size of the priors and the likelihood function domain (we choose $K = 2048$, as in [7]). In addition, by exploiting the Transformer architecture, the prior distributions can be computed in parallel. For these reasons, evaluating and sampling from $p(z_{1,s}, z_{2,s} | \mathbf{z}_{1:s-1}, \mathbf{y})$ at each s is computationally feasible and has $O(K^2)$ memory complexity. See Figure 1 for a visual description of the inference algorithm.

4.2. Latent likelihood via LQ-VAE

In this section we describe how we model the likelihood function and introduce the LQ-VAE model. Following [22] we chose a σ -isotropic Gaussian likelihood, setting:

$$\begin{aligned} p(\mathbf{m}_{\text{latent}} | z_{1,s}, z_{2,s}, \mathbf{z}_{1:s-1}) &= \\ &= p(\mathbf{m}_{\text{latent},s} | z_1, z_2) \\ &= \mathcal{N}(\mathbf{m}_{\text{latent},s} | B_Q(\frac{1}{2}\mathbf{e}_{z_1} + \frac{1}{2}\mathbf{e}_{z_2}), \sigma^2 \mathbf{I}). \end{aligned} \quad (5)$$

The hyper-parameter σ balances the trade-off between the likelihood and the priors. Lower values promote the likelihood: the separated tracks combine perfectly with \mathbf{m} , but may not sound like the instrument of the class they belong to. Instead, higher values of σ give importance to the priors: the separated tracks contain only sounds from the corresponding source distribution, but may not mix back to \mathbf{m} (not resembling the sources). The logarithm of the likelihood is:

$$-\frac{1}{2\sigma^2} \|\mathbf{m}_{\text{latent},s} - B_Q(\frac{1}{2}\mathbf{e}_{z_1} + \frac{1}{2}\mathbf{e}_{z_2})\|_2^2. \quad (6)$$

At each step s , we compare a variable term $\mathbf{m}_{\text{latent},s}$ with a constant matrix $B_Q(\frac{1}{2}\mathbf{e}_{z_1} + \frac{1}{2}\mathbf{e}_{z_2})$ representing all possible (scaled) sums over all codes in \mathcal{C} . This term can be precomputed once and then reused during inference, saving additional computational resources.

We observed that performing separation with the likelihood in Eq. (5) using a VQ-VAE trained with the loss in Eq. (1), results in disturbed and noisy outcomes. Such behavior is expected because the standard VQ-VAE does not impose any algebraic structure on the discrete domain; therefore, summing codes as in Eq. (5) does not lead to meaningful results. This problem can be lifted by enforcing a post-quantization linearization loss on the VQ-VAE:

$$\mathcal{L} = \mathcal{L}_{\text{VQ-VAE}} + \mathcal{L}_{\text{lin}}, \quad (7)$$

where $\mathcal{L}_{\text{VQ-VAE}}$ is defined as in Eq. (1) and

$$\mathcal{L}_{\text{lin}} = \frac{1}{T} \sum_t \|LQ_t - QL_t\|_2^2 \quad (8)$$

$$QL_t = B_Q(\frac{1}{2}B_Q(E(\mathbf{x}_{1,t})) + \frac{1}{2}B_Q(E(\mathbf{x}_{2,t}))) \quad (9)$$

$$LQ_t = B_Q(E(\frac{1}{2}\mathbf{x}_{1,t} + \frac{1}{2}\mathbf{x}_{2,t})). \quad (10)$$

Method	Drums	Bass	Drums	Guitar	Guitar	Bass
Ours (best)	5.83	7.42	8.33	3.80	3.75	8.65
Ours (rej)	4.08	5.31	6.93	2.48	1.95	6.35
Demucs [†]	5.42	5.36	5.80	5.36	6.42	7.68
TasNet [†]	5.51	5.43	5.87	5.47	7.80	8.46
rPCA[23]	0.60	1.05	2.27	-0.42	0.52	-1.12
ICA[24]	-0.99	-1.53	-0.53	-3.23	-0.73	-2.79
HPSS [25]	-0.56	-0.33	0.31	-2.72	0.15	-0.38
REPET[26]	0.53	1.54	2.91	0.11	0.40	-1.09
FT2D [27]	0.59	1.31	2.63	-0.15	0.65	-1.02

Table 1: SDR scores evaluated on Slakh2100 test set. All methods are unsupervised except those marked with †. The rej attribute indicates that the solutions were obtained by the rejection sampling procedure with $\alpha = 0$. The scores are computed according to the implementation in [28]

Rejection α	Drums	Bass	Drums	Guitar	Guitar	Bass
0	4.08	5.31	6.93	2.48	1.95	6.35
0.5	3.61	4.78	6.69	2.17	1.68	6.00
1	2.94	4.03	6.44	1.95	1.15	5.35

Table 2: Ablation study for rejection parameter α .

Method	Drums	Piano
Ours	0.68	3.66
Ours (rejection $\alpha = 0$)	0.08	2.75
GAN [20]	-3.16	-2.26

Table 3: SDR table evaluated on the test set of [21].

Minimizing this loss pushes the quantized latent code representing a mixture of two arbitrary source signals (LQ_t term) to be equal to the sum of the quantized latent codes, corresponding to the single sources (QL_t term), therefore enforcing the discrete codes to behave in an approximately linear way. We shall refer to the VQ-VAE trained as above, as a *Linearly Quantized Variational Autoencoder* (LQ-VAE). See Figure 2 for a visual illustration of the LQ-VAE training procedure.

4.3. Rejection sampling

Given the low memory requirements of our method, at inference time we can sample in parallel multiple solutions $\{\mathbf{z}^{(b)}\}_{b=1}^B$ in the same batch. Autoregressive models tend to accumulate errors over the course of ancestral sampling, therefore the quality of the solutions varies across the batch. In order to select a solution, we look at the posterior $p_{\text{rej}}(\mathbf{z}|\mathbf{m}) \propto p_{\text{rej},1}(\mathbf{z}_1)p_{\text{rej},2}(\mathbf{z}_2)p_{\text{rej}}(\mathbf{m}|\mathbf{z})$, conditioned by the sampling event. We obtain the priors $p_{\text{rej},1}$ and $p_{\text{rej},2}$ by normalizing p_1 and p_2 over the batch (computed by integrating over s during the inference). For numerical stability, we scale their logits by the length of the latent sequences S . The likelihood function $p_{\text{rej}}(\mathbf{z}|\mathbf{m}) = \mathcal{N}(\mathbf{m}|\frac{1}{2}D(\mathbf{z}_1) + \frac{1}{2}D(\mathbf{z}_2), \sigma_{\text{rej}}^2\mathbf{I})$ is computed directly in the time domain, with the decoding pass being executed only once at the end of the sampling procedure. The hyper-parameter σ_{rej} plays a similar role to the σ used in Eq. (5). We can balance the likelihood and the priors by setting:

$$\mathbb{E}_b\left[\log p_{\text{rej}}(\mathbf{z}^{(b)})\right] = -\frac{1}{2\sigma_{\text{rej}}^2}\mathbb{E}_b\left[\left\|\mathbf{m} - \frac{1}{2}(D(\mathbf{z}_1^{(b)}) + D(\mathbf{z}_2^{(b)}))\right\|_2^2\right]$$

and solving for σ_{rej} . Albeit natural, this framework does not lead to the best selection. We performed an ablation study by

weighting the contribution of the global likelihood with a scalar $\alpha \in [0, 1]$ (using $\sigma_{\text{rej}}'^2 = \alpha\sigma_{\text{rej}}^2$) and the best empirical results are obtained when the global likelihood is not taken into account ($\alpha = 0$), see Table 2. We call this selection criterion *prior-based rejection sampling*.

5. Results

We validate our approach on *Slakh2100* [1]: a large musical source dataset containing mixed tracks separated into 34 instrument categories. We select tracks from the classes ‘drum’, ‘bass’ and ‘guitar’ coming from the training and test splits, subsampled at a frequency of 22kHz. We train the convolutional LQ-VAE over mixtures obtained by randomly mixing sources from the individual tracks of the training set. The LQ-VAE has a downsampling factor of $\frac{T}{S} = 64$ and uses a dictionary of $K = 2048$ latent codes. After training the LQ-VAE, we train two autoregressive models, one per source, on latent codes extracted from ~ 1200 tracks each. In all our separation experiments we fixed $\sigma = 0.1$ in Eq. (6). In Table 1 we compare our method with two state-of-the-art supervised approaches and different non-learning based unsupervised methods. To this end, we iterate on the test split of [1] made up of about 150 different songs, and for each we extract 450 random chunks each of 3 seconds.

In order to strengthen our empirical evaluation, we show in Table 3 results of our model applied to a different validation data set in order to perform a comparison with the GAN model of [21]. We evaluate both methods over the test dataset proposed in [21], consisting of 1000 mixtures of 1 second each. Each mixture combines a drum sample with a piano track randomly, thus independence in the test data is assumed, resulting in a more artificial setting with respect to the one present in Slakh2100. For [21] we use the pre-trained model given by the authors while for our method we use the ‘‘drums’’ and ‘‘piano’’ priors trained on Slakh2100 thus showing the cross-dataset generalization capability of our model.

All our experiments are performed on a Nvidia RTX 3080 GPU with 16 GB of VRAM. With this GPU our method can sample a batch of 200 candidate solutions (100 for each instrument) simultaneously. The code to reproduce our experiments is available at <https://github.com/michelemancusi/LQVAE-separation>. Interestingly, even if solutions selected by the rejection sampling algorithm have slightly lower metrics than supervised approaches, by individually selecting the best solution for each instrument we achieve performance in line with the state of the art (especially on ‘bass’ and ‘drum’ stems). This testifies the quality of our separation. Remarkably, our method employs 3 minutes on average for sampling a track of 3 seconds, compared to the more than 100 minutes of [4].

6. Conclusions

In this work, we introduced a simple algorithm to perform exact Bayesian inference in the discrete latent domain. Our method allows to achieve good separation results while being much faster than other likelihood-based unsupervised approaches.

The main bottleneck of our method lies in the rejection sampling strategy. Future work will attempt to improve this aspect by investigating the design of more accurate learning-based rejection samplers. Other benefits could come from the adoption of multi-level VQ-VAEs [7] or by leveraging deeper autoregressive priors.

7. References

- [1] 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 on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019.
- [2] H. Larochelle and I. Murray, "The neural autoregressive distribution estimator," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, 2011.
- [3] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016.
- [4] V. Jayaram and J. Thickstun, "Parallel and flexible sampling from autoregressive models via langevin dynamics," 2021.
- [5] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, 2017*.
- [6] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, 2019*.
- [7] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," 2020.
- [8] P. Comon, "Independent Component Analysis, a new concept?" *Signal Processing*, 1994.
- [9] P. Smaragdis, C. Févotte, G. J. Mysore, N. Mohammadiha, and M. Hoffman, "Static and dynamic source separation using non-negative factorizations: A unified view," *IEEE Signal Processing Magazine*, no. 3, 2014.
- [10] S. T. Roweis, "One microphone source separation," in *Advances in Neural Information Processing Systems 13, Papers from Neural Information Processing Systems (NIPS) 2000, Denver, CO, USA, 2000*.
- [11] S. Uhlich, F. Giron, and Y. Mitsufuji, "Deep neural network based instrument extraction from music," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015, 2015*.
- [12] S. Uhlich, M. Porcu, F. Giron, M. Enenkl, T. Kemp, N. Takahashi, and Y. Mitsufuji, "Improving music source separation based on deep neural networks through data augmentation and network blending," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017, 2017*.
- [13] J.-Y. Liu and Y.-H. Yang, "Denoising auto-encoder with recurrent skip connections and residual regression for music source separation," 2018.
- [14] N. T. an, "Mmdenselstm: An efficient combination of convolutional and recurrent," *ArXiv preprint*, 2018.
- [15] W. Choi, M. Kim, J. Chung, and S. Jung, "Lasoft: Latent source attentive frequency transformation for conditioned source separation," 2020.
- [16] N. Takahashi and Y. Mitsufuji, "D3net: Densely connected multi-dilated densenet for music source separation," 2020.
- [17] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "The MUSDB18 corpus for music separation," 2017.
- [18] G. Meseguer-Brocal and G. Peeters, "Conditioned-U-Net: Introducing a Control Mechanism in the U-Net for Multiple Source Separations," *arXiv:1907.01277 [cs, eess]*, 2019, arXiv: 1907.01277.
- [19] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music Source Separation in the Waveform Domain," *arXiv:1911.13254 [cs, eess, stat]*, 2019, arXiv: 1911.13254.
- [20] S. Wisdom, E. Tzinis, H. Erdogan, R. Weiss, K. Wilson, and J. Hershey, "Unsupervised sound separation using mixture invariant training," vol. 33, pp. 3846–3857, 2020. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/file/28538c394c36e4d5ea8ff5ad60562a93-Paper.pdf>
- [21] V. Narayanaswamy, J. J. Thiagarajan, R. Anirudh, and A. Spanias, "Unsupervised audio source separation using generative priors," 2020.
- [22] V. Jayaram and J. Thickstun, "Source separation with deep generative priors," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, ser. Proceedings of Machine Learning Research, 2020.
- [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 *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 57–60.
- [24] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural networks*, vol. 13, no. 4-5, pp. 411–430, 2000.
- [25] D. Fitzgerald, "Harmonic/percussive separation using median filtering," in *Proc. of DAFX*, vol. 10, no. 4, 2010.
- [26] Z. Rafii and B. Pardo, "Repeating pattern extraction technique (repet): A simple method for music/voice separation," *IEEE transactions on audio, speech, and language processing*, vol. 21, no. 1, pp. 73–84, 2012.
- [27] P. Seetharaman, F. Pishdadian, and B. Pardo, "Music/voice separation using the 2d fourier transform," in *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 36–40.
- [28] F.-R. Stöter, A. Liutkus, and N. Ito, "The 2018 signal separation evaluation campaign," in *Latent Variable Analysis and Signal Separation: 14th International Conference, LVA/ICA 2018, Surrey, UK, 2018*, pp. 293–305.