Self-Supervised Music Source Separation Using Vector-Quantized Source Category Estimates

Marco Pasini[*]
Queen Mary University
London, UK

Stefan Lattner[*]
Sony Computer Science Laboratories
Paris, France

György Fazekas
Queen Mary University
London, UK

Abstract

Music source separation is focused on extracting distinct sonic elements from composite tracks. Historically, many methods have been grounded in supervised learning, necessitating labeled data, which is occasionally constrained in its diversity. More recent methods have delved into N-shot techniques that utilize one or more audio samples to aid in the separation. However, a challenge with some of these methods is the necessity for an audio query during inference, making them less suited for genres with varied timbres and effects. This paper offers a proof-of-concept for a self-supervised music source separation system that eliminates the need for audio queries at inference time. In the training phase, while it adopts a query-based approach, we introduce a modification by substituting the continuous embedding of query audios with Vector Quantized (VQ) representations. Trained end-to-end with up to N classes as determined by the VQ's codebook size, the model seeks to effectively categorise instrument classes. During inference, the input is partitioned into N sources, with some potentially left unutilized based on the mix's instrument makeup. This methodology suggests an alternative avenue for considering source separation across diverse music genres. We provide examples and additional results online [1].

1 Introduction

Source separation is a crucial domain in signal processing, aiming to extract individual components from mixed signals. This challenge has been widely addressed in the audio domain across various fields, including universal source separation, (multi-)speaker separation, and music source separation (MSS). MSS stands as a significant sub-domain, with the goal of isolating individual instruments, vocals, or other sonic elements from a composite track. Traditionally, MSS has been driven by supervised methods that necessitate large amounts of labeled stems. However, dedicated, openly available datasets are rare, and limited in the number of instrument classes. As a result, the default benchmark in MSS only involves four classes: vocals, drums, bass, and other.

In order to lift this limitation, methods for unsupervised, semi-supervised, or self-supervised source separation have been investigated in recent years. A particular trend in semi- and self-supervised methods involves zero- one- or few-shot learning approaches. These models typically rely on one or multiple audio samples from a specific instrument class to guide the separation process. In practice, it is easier to find datasets of unlabelled or weakly labeled musical sounds, contained in public collections like AudioSet [1], Freesound [2], or proprietary datasets. Consequently, using audio queries instead of instrument labels makes it possible to train self-supervised source separation models on such datasets (e.g., by using different segments of a stem as query and separation target).

---

[*]Equal contribution
[1]`https://anonymous2732.github.io/vqss/`

However, the necessity of having the respective audio query available at inference time is a limiting factor in N-shot source separation. Particularly in contemporary popular or electronic music, the constituents can have peculiar timbres, like different synthesizers, guitar amplifiers, unusual samples, or sound effects. Therefore, ideally, a source separation system should be able to separate different instrument classes at inference time without requiring additional information.

This work introduces a proof-of-concept self-supervised music source separator that does not require audio queries at inference time. The method is a self-supervised query-based source separation system at train time, but we replace the continuous embedding space of the query audios with VQ embeddings. The model is trained end-to-end using a maximum of N classes, which is the codebook size of the VQ layer, to learn a helpful clustering of the respective instrument classes. During inference, the model divides the input mix into N sources. Depending on the presence of the different instrument classes in the mix, some outputs may remain empty.

2 Method

We propose a self-supervised MSS system that does not require audio queries during inference. The model is trained in a query-based fashion using random crops of stems as queries and targets. However, instead of using a continuous embedding space for the queries, we introduce a vector quantization layer to categorize the instrument classes into a discrete codebook. The overall architecture consists of three main components: A *Style Encoder E* that embeds the audio query into a latent space, a *Vector Quantizer Q* that quantizes the continuous embedding into a discrete codebook, and a *Generator G* that generates the separated source conditioned on the quantized code.

Let x be an unlabelled audio stem. We first randomly crop x two times to obtain a target excerpt x_{targ} and a reference excerpt x_{ref}. The target excerpt is linearly mixed with $k \sim \mathcal{U}(0, K)$ additional random stems to obtain the mixture x_{mix}. The reference excerpt x_{ref} is fed into the Style Encoder E to produce a d-dimensional embedding $z_{ref} = E(x_{ref})$. This continuous embedding is then quantized by a Vector Quantization layer Q to obtain a quantized embedding $\hat{z}_{ref} = Q(z_{ref})$ corresponding to the closest entry in a codebook of size N, representing the maximum number of different sources allowed by the system. The Generator G takes as input the mixture x_{mix} and the quantized embedding $\hat{z}_{ref}$, and is tasked with separating the target source x_{targ} from the mixture:

$$\hat{x}_{targ} = G(x_{mix}, \hat{z}_{ref}).$$

We parameterize the Generator with a U-Net [3] model which takes magnitude spectrograms as input and outputs magnitude and phase spectrograms which are then combined into a waveform sample via the iSTFT operator. The model is inspired by the Autoencoder proposed in [4] and is thus trained in a similar fashion. Differently from [4], the model is trained fully end-to-end using the following objectives: The *Reconstruction loss* uses the same reconstruction objective $\mathcal{L}_{rec}$ proposed in [4], which consists of a L1 loss between log-spectrograms and a multi-scale spectral loss [5], [6] between the output $\hat{x}_{targ}$ and target x_{targ}. Details about the reconstruction objective can be found in the original paper. The *Adversarial loss* utilizes the hinge adversarial loss $\mathcal{L}_{adv}$ [7] to implicitly model the phase spectrogram as:

$$\mathcal{L}_{adv} = \quad \mathbb{E}_{x_{targ}}[\min(0, -1 + D(\log(|\text{STFT}(x_{targ})|)))]$$
$$+ \mathbb{E}_{\hat{x}_{targ}}[\min(0, -1 - D(\log(|\text{STFT}(\hat{x}_{targ})|)))],$$

where D is a Discriminator model. Lastly, the *Commitment loss* is the VQ-specific objective $\mathcal{L}_{vq}$ that ensures the encoder outputs are close to the quantized codes from the codebook: $\mathcal{L}_{vq} = ||z_{ref} - \text{sg}(\hat{z}_{ref})||_2^2$, with sg denoting the stop-gradient operator. The codebook loss is realized through exponential moving average (EMA) updates, as suggested in [8], [9].

The overall training loss is a weighted sum of these objectives:

$$\mathcal{L} = \mathcal{L}_{adv} + \lambda_{rec}\mathcal{L}_{rec} + \lambda_{vq}\mathcal{L}_{vq}.$$

At inference time, the model takes as input the mixture x_{mix} and each of the final quantized embeddings and iteratively outputs N estimated sources, without requiring any audio query. Depending on the actual instruments present in the mixture, some of the outputs may be silent.

This approach eliminates the need for audio queries during inference. The VQ-based categorization acts as an instrument classifier that is learned in a self-supervised manner through the source separation task. Furthermore, by training fully end-to-end, the model learns a discrete embedding space specifically tailored for the task of source separation.
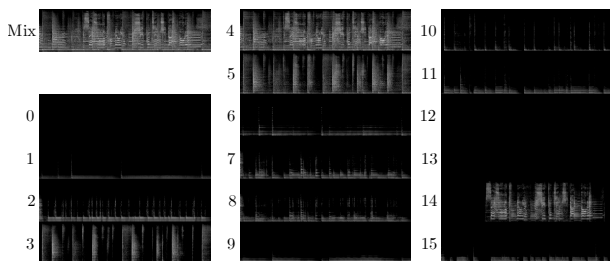
Figure 1: Mel-spectrograms of an example input mix and separated outputs. For each separation from 0 to 15 the Generator is conditioned on the corresponding quantized embedding. By cross-referencing this visualization with the histogram in Figure 2, it is possible to recognize specific instruments, such as *Vocals* for cluster number 14.
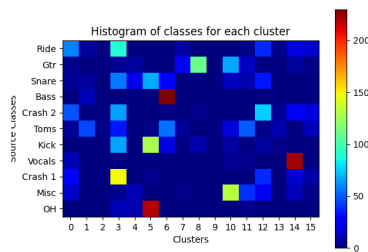

Figure 2: Distribution of the different classes of the test set for each quantized embedding found by the system. We notice a clear clustering of certain sources (*OH, Bass, Vocals*), while for other classes (*Toms, Snare, Ride*) the distribution is shared across different clusters.

3 Implementation Details

The Generator, Style Encoder and Critic are fully convolutional models using standard residual blocks [10] with group normalization layers [11] with 8 groups. The Generator is based on a U-Net architecture where the conditioning information is introduced via FiLM layers [12] at every block. The Style Encoder takes mel-spectrograms with 128 frequency bins as input and produces a single 512-dimensional vector via a Global Average Pooling (GAP) layer. The VQ layer is initialized with $N = 16$ embeddings via 10 iterations of k-means clustering of the first input training batch. As proposed by [13], we use cosine similarity instead of Euclidean distance and we factorize the codes in a lower dimensional space with dimensionality of 8 to improve codebook usage. The codebook loss is implemented via EMA with a factor of 0.99. Regarding the STFT parameters used during training, we adopt the ones proposed by [4]. We choose $\lambda_{rec} = 2.5$ and $\lambda_{vq} = 100$. During training we use random crops of 1.5 seconds with a sampling rate of 44.1 kHz to train the system. While we ensure that all x_{ref} crops are not completely silent, we allow x_{targ} crops to be silent, since it allows the model to produce a silent sample when the quantized embedding of a source that is not present in the input mix is used. We train the model for 1 million iterations with a batch size of 32, using Adam [14] with $\beta_1 = 0.5$ and $\beta_2 = 0.9$ as the optimizer.

4 Experiments and Results

	OH	Misc	Crash 1	Vocals	Kick	Toms	Crash 2	Bass	Snare	Gtr	Ride
L1	14.51	16.55	19.30	15.06	24.84	23.60	19.39	16.68	15.70	14.98	17.75
L1 rand	23.64	22.47	22.92	22.25	23.85	31.14	23.97	22.61	23.18	23.40	21.65

Table 1: L1 distance between log-spectrograms of separated and true sources for each class in the test set. For comparison, we also calculate the L1 distance when the model is conditioned on a random quantized embedding instead of the target one. The difficulty of the model to correctly separate 'Kick' from other drums results in high L1 for that class.

We train the system on an internal collection of 20,000 unlabelled stems. To evaluate the separation performance, we use an internal collection of 1,000 labelled stems with the following classes: *OH, Misc, Crash 1, Crash 2, Vocals, Kick, Toms, Bass, Snare, Gtr, Ride*. We first visualize in Figure 2 the distribution of the different classes for each final quantized embedding, or cluster, found by the system. It is worth noting how the data distribution of training samples differs from the limited labelled collection used for testing. For example, training stems may not contain clearly separated percussive instruments, despite them being separated in different classes for the testing samples. This can explain the sub-par clustering performance on some specific classes used for testing. Considering the specific capability of the system to perform separation of arbitrary learned sources, evaluation using standard metrics in the field of source separation [15] can be challenging, since different clusters

can represent a single instrument class, and some uncommon instrument classes can be excluded from the clustering altogether. In Table 1 we evaluate the separation performance of the model in terms of L1 distance between generated and target log-spectrograms for each class in the test set. More specifically, different crops serve as the target sample, which is mixed with 4 other sources to create the input mixture, and as the input to the Style Encoder to produce the quantized embedding used as conditioning. For comparison, we also calculate L1 distances when a random quantized embedding is used as conditioning. We also visualize in Figure 1 mel-spectrograms of an example test input mix and of the corresponding separations performed by the system, one for each of the codebook embeddings. By cross-referencing Figure 2, we can notice specific instruments being separated. However, we can also notice hallucination behavior that is characteristic of the proposed system: considering that we explicitly train a generative model to perform the task, the Generator may produce realistic separated outputs that are not actually present in the input mix.

5   Related Work

Supervised neural network-based MSS has been tackled with multi-layer perceptrons [16]–[18], CNNs and RNNs (and combinations thereof) [19]–[23], UNets [24], [25] and with Transformers [26]. In such supervised approaches, a specific model is typically trained for each source type. Self-supervised paradigms involve permutation-invariant training for universal source separation [27], and an explicit instrument-clustering approach for MSS [28]. Unsupervised approaches to source separation have been investigated using mixture-invariant training [29] and unsupervised blind source separation with Variational Autoencoders [30], [31]. Audio query-based (i.e., N-shot) approaches exist in supervised fashions for one-shot [32] and few-shot [33] learning. Such query-based approaches are suitable for self-supervised training, where two segments of a stem are typically used as a separation target and a query, leading to zero-shot [34] and one-shot [35] architectures. Such methods typically use a single separation network with conditioning that is derived from embeddings of an audio query. Such a single separation model is also used in a meta-learning approach [36] and for universal source separation with weakly labelled data [37], the latter uses conditioning from a separate audio tagging model. The model proposed in this work is a one-hot conditioned UNet like in [25] that is trained in a self-supervised one-shot fashion like in [35]. In addition, we perform an unsupervised clustering to inform the separation module like in [28]. However, in contrast to [28], we perform this clustering in an end-to-end fashion using a VQ embedding space.

6   Conclusion

This work introduced a proof-of-concept for a self-supervised MSS system that does not require audio queries during inference. The method adopts a fully end-to-end query-based approach during training, using random crops of stems as queries and targets. However, instead of a continuous embedding space, we introduced a vector quantization layer to categorize unlabelled sources into a discrete codebook. We demonstrate both the potential and current limitations of the proposed approach. On one hand, the model shows a clear ability to cluster certain common instrument classes fairly consistently. This indicates the feasibility of learning meaningful representations in a completely self-supervised manner using the source separation task as a proxy and without the explicit use of contrastive learning techniques. On the other hand, we also highlight some flaws that need to be addressed in future work. Firstly, the model can exhibit a hallucination behavior, where it tends to generate realistic outputs even for sources that are not present in the input mixture. While this phenomenon is not uncommon in adversarial networks, it is undesirable for source separation where the objective is to filter out existing sources. Hallucinations could be alleviated by training on a dataset with a more balanced class distribution or by increasing the weight of the reconstruction loss. Secondly, the perceived audio quality of the separated outputs is currently lacking, with noticeable artifacts. This suggests that the model has not learned a robust generalized representation of the diverse instrument classes.

Overall, this work should be considered a first step towards fully self-supervised source separation models that do not rely on audio queries at test time. The clustering-based separation indicates a promising direction, but more research is required to improve the fidelity of the outputs and reduce hallucination behavior. With further development, end-to-end self-supervised source separation could provide flexibility to handle the wide range of sounds that occur in real-world music.

References

[1] J. F. Gemmeke *et al.*, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP*, IEEE, 2017, pp. 776–780.

[2] F. Font *et al.*, "Freesound technical demo," in *ACM Multimedia*, ACM, 2013, pp. 411–412.

[3] O. Ronneberger *et al.*, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[4] M. Pasini and J. Schlüter, "Musika! fast infinite waveform music generation," in *ISMIR*, 2022, pp. 543–550.

[5] J. H. Engel *et al.*, "DDSP: differentiable digital signal processing," in *ICLR*, OpenReview.net, 2020.

[6] A. Caillon and P. Esling, "Rave: A variational autoencoder for fast and high-quality neural audio synthesis," *arXiv preprint arXiv:2111.05011*, 2021.

[7] J. H. Lim and J. C. Ye, "Geometric gan," *arXiv preprint arXiv:1705.02894*, 2017.

[8] A. van den Oord *et al.*, "Neural discrete representation learning," in *NeurIPS*, 2017.

[9] A. Razavi *et al.*, "Generating diverse high-fidelity images with VQ-VAE-2," in *NeurIPS*, 2019.

[10] K. He *et al.*, "Deep residual learning for image recognition," in *CVPR*, 2016.

[11] Y. Wu and K. He, "Group normalization," in *ECCV (13)*, ser. Lecture Notes in Computer Science, vol. 11217, Springer, 2018, pp. 3–19.

[12] E. Perez *et al.*, "Film: Visual reasoning with a general conditioning layer," in *AAAI*, AAAI Press, 2018, pp. 3942–3951.

[13] J. Yu *et al.*, "Vector-quantized image modeling with improved VQGAN," in *ICLR*, OpenReview.net, 2022.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[15] J. L. Roux *et al.*, "SDR - half-baked or well done?" In *ICASSP*, IEEE, 2019, pp. 626–630.

[16] A. A. Nugraha *et al.*, "Multichannel music separation with deep neural networks," in *EUSIPCO*, IEEE, 2016, pp. 1748–1752.

[17] S. Uhlich *et al.*, "Deep neural network based instrument extraction from music," in *ICASSP*, IEEE, 2015, pp. 2135–2139.

[18] N. Takahashi and Y. Mitsufuji, "Multi-scale multi-band densenets for audio source separation," in *WASPAA*, IEEE, 2017, pp. 21–25.

[19] N. Takahashi and Y. Mitsufuji, "D3net: Densely connected multidilated densenet for music source separation," *CoRR*, vol. abs/2010.01733, 2020.

[20] S. Uhlich *et al.*, "Improving music source separation based on deep neural networks through data augmentation and network blending," in *ICASSP*, IEEE, 2017, pp. 261–265.

[21] N. Takahashi *et al.*, "MMDenseLSTM: An efficient combination of convolutional and recurrent neural networks for audio source separation," in *IWAENC*, IEEE, 2018, pp. 106–110.

[22] F. Stöter *et al.*, "Open-unmix - A reference implementation for music source separation," *J. Open Source Softw.*, vol. 4, no. 41, p. 1667, 2019.

[23] Y. Luo and J. Yu, "Music source separation with band-split RNN," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 1893–1901, 2023.

[24] D. Stoller *et al.*, "Wave-u-net: A multi-scale neural network for end-to-end audio source separation," in *ISMIR*, 2018, pp. 334–340.

[25] W. Choi *et al.*, "Lasaft: Latent source attentive frequency transformation for conditioned source separation," in *ICASSP*, IEEE, 2021, pp. 171–175.

[26] W. T. Lu *et al.*, "Music source separation with band-split rope transformer," *CoRR*, vol. abs/2309.02612, 2023.

[27] D. Yu *et al.*, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *ICASSP*, IEEE, 2017, pp. 241–245.

[28] K. Chen *et al.*, "Pa5607-hubert: Self-supervised music source separation via primitive auditory clustering and hidden-unit bert," in *ICASSP Workshops*, IEEE, 2023, pp. 1–5.

[29] S. Wisdom *et al.*, "Unsupervised sound separation using mixture invariant training," in *NeurIPS*, 2020.

[30] J. Neri *et al.*, "Unsupervised blind source separation with variational auto-encoders," in *EUSIPCO*, IEEE, 2021, pp. 311–315.

[31] L. Li *et al.*, "Fastmvae2: On improving and accelerating the fast variational autoencoder-based source separation algorithm for determined mixtures," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 31, pp. 96–110, 2023.

[32] J. H. Lee *et al.*, "Audio query-based music source separation," in *ISMIR*, 2019, pp. 878–885.

[33] Y. Wang *et al.*, "Few-shot musical source separation," in *ICASSP*, IEEE, 2022, pp. 121–125.

[34] L. Lin *et al.*, "A unified model for zero-shot music source separation, transcription and synthesis," in *ISMIR*, 2021, pp. 381–388.

[35] B. Gfeller *et al.*, "One-shot conditional audio filtering of arbitrary sounds," in *ICASSP*, IEEE, 2021, pp. 501–505.

[36] D. Samuel *et al.*, "Meta-learning extractors for music source separation," in *ICASSP*, IEEE, 2020, pp. 816–820.

[37] Q. Kong *et al.*, "Universal source separation with weakly labelled data," *CoRR*, vol. abs/2305.07447, 2023.