A UNIFIED MODEL FOR ZERO-SHOT MUSIC SOURCE SEPARATION, TRANSCRIPTION AND SYNTHESIS

Liwei Lin[1]   **Qiuqiang Kong**[2]   **Junyan Jiang**[1]   **Gus Xia**[1]

[1] Music X Lab, New York University Shanghai
[2] ByteDance, Shanghai, China

`linliwei3916@gmail.com, kongqiuqiang@bytedance.com,jj2732@nyu.edu,gxia@nyu.edu`

ABSTRACT

We propose a unified model for three inter-related tasks: 1) to *separate* individual sound sources from a mixed music audio, 2) to *transcribe* each sound source to MIDI notes, and 3) to *synthesize* new pieces based on the timbre of separated sources. The model is inspired by the fact that when humans listen to music, our minds can not only separate the sounds of different instruments, but also at the same time perceive high-level representations such as score and timbre. To mirror such capability computationally, we designed a pitch-timbre disentanglement module based on a popular encoder-decoder neural architecture for source separation. The key inductive biases are vector-quantization for pitch representation and pitch-transformation invariant for timbre representation. In addition, we adopted a query-by-example method to achieve *zero-shot* learning, i.e., the model is capable of doing source separation, transcription, and synthesis for *unseen* instruments. The current design focuses on audio mixtures of two monophonic instruments. Experimental results show that our model outperforms existing multi-task baselines, and the transcribed score serves as a powerful auxiliary for separation tasks.

1. INTRODUCTION

Music source separation (MSS) is a core problem in music information retrieval (MIR), which aims to separate individual sound sources, either instrumental or vocal, from a mixed music audio. A good separation benefits various of downstream tasks of music understanding and generation [1,2] since many music-processing algorithms call for "clean" sound sources.

With the development of deep neural networks, we see significant performance improvements in MSS. The current mainstream methodology is to train on pre-defined music sources and then infer a mask on the spectrogram (or other data representations) of the mixed audio. More recently, we see several new efforts in MSS research, includ-

ing query-based method [3–6] for unseen (not pre-defined) sources, semantic-based separation that incorporates auxiliary information such as score or video [7–13], and multi-task settings [14].

This study conceptually combines the aforementioned new ideas but follows a very different methodology — instead of directly applying masks, *we regard MSS an audio pitch-timbre disentanglement and reconstruction problem*. Such strategy is inspired by the fact that when humans listen to music, our minds not only separate the sounds into different sources but also perceive high-level pitch and timbre representations that generalize well during both music understanding and creation. For example, humans can easily identify the same timbre in other pieces or identify the same piece played by other instruments. People can even mimic the learned timbre using human voice and sing (i.e., to synthesize via voice) the learned pitch sequence.

To mirror such capability computationally, we propose a zero-shot multi-task model jointly performing MSS, automatic music transcription (AMT), and synthesis. The model comprises four components: 1) a query-by-example (QBE) network, 2) a pitch-timbre disentanglement module, 3) a transcriptor, and 4) an audio encoder-decoder network. First, the QBE network summarizes the clean query example audio (which contains only one instrument) into a low-dimensional query vector, conditioned on which the audio encoder extracts the latent representation of an individual sound source. Second, the model disentangles the latent representation into pitch and timbre vectors while transcribing the score using the transcriptor. Finally, the audio decoder takes in both the disentangled pitch and timbre representations, generating a separated sound source. When the model further equips the timbre representation with a pitch-transformation invariance loss, the decoder becomes a synthesizer, capable of generating new sounds based on an existing timbre vector and new scores.

The current model focuses on audio mixtures of two monophonic instruments and performs in a frame-by-frame fashion. Also, it only transcribes pitch and duration information. We leave polyphonic and vocal scenarios as well as a more complete transcription for future work. In sum, our contributions are:

- **Zero-shot multi-task modeling**: To the best of our knowledge, it is the first model that jointly performs separation, transcription, and synthesis. It works

for both previously seen and unseen sources using a query-based method.

- **Well-suited inductive bias**: The neural structure is analogous to the "hardware" of the model, which alone is inadequate to achieve good disentanglement. We designed two extra inductive biases: vector-quantization for pitch representation and pitch-transformation invariant for timbre representation, which serves as a critical part of the "software" of the model.

- **None-mask-based MSS**: Our methodology regards MSS an audio pitch-timbre disentanglement and recreation problem, unifying music understanding and generation in a representation learning framework.

2. RELATED WORK

Most effective music source separation (MSS) methods are based on well-designed neural networks, such as U-Net [15] and MMDenseLSTM [16]. Here, we review three new trends of MSS related of our work: 1) multi-task learning, 2) zero-shot for unseen sources, and 3) taking advantage of auxiliary semantic information.

2.1 Multi-task Separation and Transcription

Several recent studies [14, 17, 18] conduct multi-task separation and transcription by learning a joint representation for both tasks. These works demonstrated that a multi-task setting benefits one or both of the two tasks due to the better generalized capability of the learned joint representation. Our model is a multi-task and can further disentangle pitch and timbre representation for sound synthesis.

2.2 Query-based Separation

Few-shot and zero-shot learning are becoming popular in MIR. For the MSS task, It is meaningful to separate unseen rather than pre-defined sources since it is unrealistic to collect training data that covers all the sources with considerable amounts. Query-by-example (QBE) network is one of solutions for zero-shot learning and recent researches [3, 4, 6, 19–22] show its nice performance. In this study, we adopt a QBE as in [3].

2.3 Semantic-based Separation

Many researches demonstrate that semantic information is a useful auxiliary for MSS. For example, Gover et al. [10] designs a score-informed wave-U-Net to separate choral music; Jeon et al. [23] performs the lyrics-informed separation; Meseguer-Brocal et al. [11] develops a phoneme-informed C-U-Net [24]; Zhao et al. [12] takes advantage of visual information to separate homogeneous instruments. But these methods cannot separate sources without additional semantic groundtruths during inference. Our study can also be regarded as score-informed MSS, but our model does not call for ground truth score during the inference time.

3. METHODOLOGY

In this section, we describe our proposed 1) multi-task and QBE model for source separation; 2) pitch-timbre disentanglement module; 3) pitch-translation invariance loss.

3.1 Multi-task Separation and Transcription

Different from previous works that tackle the music separation and music transcription problems separately, we learn a joint representation for both of them. Previous works [14, 17] have shown that the representation learnt by a joint separation and transcription task can generalize better than the representation learnt by single-task models.

We denote the waveform of two single-source audio segments from different sources as $\boldsymbol{s}_c \in \mathbb{R}^L$ and $\boldsymbol{s}_i \in \mathbb{R}^L$, respectively. We denote their mixture as:

$$\boldsymbol{x} = \boldsymbol{s}_c + \boldsymbol{s}_i. \tag{1}$$

Our aim is to separate $\boldsymbol{s}_c$ from $\boldsymbol{x}$. We denote the spectrogram of $\boldsymbol{x}$ and $\boldsymbol{s}_c$ as $\boldsymbol{X} \in \mathbb{R}^{T \times F}$ and $\boldsymbol{S}_c \in \mathbb{R}^{T \times F}$, respectively.

We first formalize the general MSS model using an encoder-decoder neural architecture. For instance, UNet [15] is an encoder-decoder architecture which is widely used in MSS. By ignoring the skip connections of UNet, the output of the encoder (the bottleneck of U-Net) can be used as a joint representation for separation and transcription. Different from previous MSS methods that estimate a single-target mask on the mixture spectrogram, we design the separation model to directly output spectrograms. In this way, the model can not only separate a source from a mixture, but can also synthesize new audio recordings from joint representations.

For the source separation system, we denote the encoder and decoder as follows:

$$\boldsymbol{h} = \text{Encoder}(\boldsymbol{X}), \tag{2}$$

$$\widehat{\boldsymbol{S}}_c = \text{Decoder}_c(\boldsymbol{h}), \tag{3}$$

where $\boldsymbol{h}$ is the learned joint representation and Decoder_c is the decoder for the target source c. The joint representation $\boldsymbol{h}$ is used as input to the a transcription model:

$$\widehat{\boldsymbol{y}}_c = \text{Transcriptor}_c(\boldsymbol{h}), \tag{4}$$

where $\widehat{\boldsymbol{y}}_c \in [0, 1]^{T \times N}$ are probabilities of the predicted MIDI roll. Typically, we set $N = 89$ including 88 notes on a piano and a silence state.

When designing neural networks, to remain the transcription resolution, we do not apply temporal pooling operation in the encoder, decoder and transcriptor. So that the temporal resolution of $\boldsymbol{h}$ is consistent with that of $\boldsymbol{X}$. We describe the details of the encoder, decoder, and transcriptor in Section 4.2.

3.2 Query-by-example Separation and Transcription

As described in Equation (3) and (4), we need to build J decoders to separate J target sources. With the number of target sources increases, the parameters number will

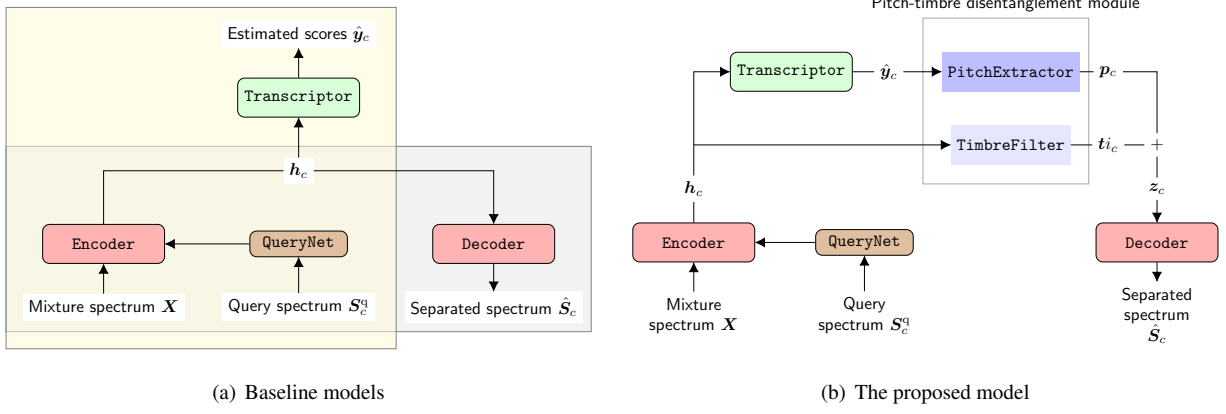(a) Baseline models (b) The proposed model

Figure 1. The baseline models and the proposed model. In the left figure, the large orange and gray box indicate a QBE transcription-only and QBE separation-only model respectively. The whole figure indicates a QBE multi-task model.

also increase. More importantly, the model trained for pre-defined sources can not adapt to unseen sources. To tackle these problems, we design a QBE module in our model. The advantage of using QBE is that we can separate unseen target sources. That is, we achieve a zero-shot separation.

Similar to the QueryNet [3], we design a QueryNet module as shown in Figure 1(a). The QueryNet module extracts the embedding vector $\boldsymbol{q}_c \in \mathbb{R}^M$ of an input spectrogram $\boldsymbol{S}_c^q \in \mathbb{R}^{T \times F}$, where M is the dimension of the embedding vector:

$$\boldsymbol{q}_c = \text{QueryNet}(\boldsymbol{S}_c^q). \tag{5}$$

Audio recordings from the same source will be learnt to have similar embedding vectors. We propose a contrastive loss to encourage embedding vectors from the same sources to be close, and embedding vectors from different sources to be far:

$$L_{\text{query}} = \frac{1}{C}\{\left\|\boldsymbol{q}_c - \boldsymbol{q}_c'\right\|_2 + \sum_{j \neq c}^{C} \max(m - \left\|\boldsymbol{q}_c - \boldsymbol{q}_j\right\|_2, 0)\}, \tag{6}$$

where $m > 0$ is a margin and $\boldsymbol{q}_c'$ and $\boldsymbol{q}_c$ are from the same source c, and $\boldsymbol{q}_c$ and $\boldsymbol{q}_j$ are from different sources. We set M to 6 and m to 0.125 in our experiments.

We input the embedding vector $\boldsymbol{q}_c$ as a condition to each layer of the encoder using Feature-wise Linear Modulation (FiLM) [25] layers. Then the encoder outputs a representation $\boldsymbol{h}_c \in \mathbb{R}^{T \times D}$. The embedding vector $\boldsymbol{q}_c$ controls what source to separate or transcribe. There are only one encoder, one decoder, and one transcriptor to separate any sources:

$$\boldsymbol{h}_c = \text{Encoder}(\boldsymbol{X}, \boldsymbol{q}_c), \tag{7}$$

$$\widehat{\boldsymbol{S}}_c = \text{Decoder}(\boldsymbol{h}_c), \tag{8}$$

$$\widehat{\boldsymbol{y}}_c = \text{Transcriptor}(\boldsymbol{h}_c). \tag{9}$$

3.3 Pitch-timbre Disentanglement Module

Previous MSS works do not disentangle pitch and timbre for separation. That is, those MSS methods implement separation systems without estimating pitches. In this section, we propose a pitch-timbre disentanglement module based on the query-based encoder-decoder architecture described in previous sections to learn interpretable representations for MSS. Such interpretable representations enable the model to achieve score-informed separation based on predicted scores.

As shown in Figure 1(b), the proposed pitch-timbre disentanglement module consists of a PitchExtractor and a TimbreFilter module. The output of PitchExtractor $\boldsymbol{p}_c$ only contains pitch information of $\boldsymbol{s}_c$, and the output of TimbreFitler is expected to only contain timbre information of $\boldsymbol{s}_c$. The PitchExtractor is modeled by an embedding layer $\boldsymbol{V} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_N\}$, where N is the number of vectors, which equals to the number of pitches 89 in our experiment. To explain, $\boldsymbol{e}_n \in \boldsymbol{V}(\boldsymbol{e}_n \in \mathbb{R}^K)$ denotes the quantized pitch vector for the n-th MIDI note. Then, we calculate the disentangled pitch representation for $\hat{\boldsymbol{y}}$ as $\boldsymbol{p}_c = [\boldsymbol{p}_c^{(1)}; \boldsymbol{p}_c^{(2)}; ...; \boldsymbol{p}_c^{(T)}]$, where $\boldsymbol{p}_c^{(t)} \in \mathbb{R}^K$:

$$\boldsymbol{p}_c^{(t)} = \sum_n^N \hat{y}_c^{(t,n)} \cdot \boldsymbol{e}_n, \tag{10}$$

where $\hat{y}_c^{(t,n)}$ is the output of the transcriptor containing predicted presence probability of the n-th MIDI note or the silence state at time t, and K is the dimension of the disentangled pitch representation. During synthesis, we can replace $\hat{\boldsymbol{y}}_c$ with one-hot encodings of new scores as input to Equation (10) to obtain pitch representation $\boldsymbol{p}_c$ for synthesizing audio recordings.

TimbreFilter is used to filter timbre information $\boldsymbol{ti}_c \in \mathbb{R}^{T \times D}$ from $\boldsymbol{h}_c$:

$$\boldsymbol{ti}_c = \text{TimbreFilter}(\boldsymbol{h}_c). \tag{11}$$

Here, TimbreFilter is modeled by a convolutional neural network. Then, we can synthesize $\hat{\boldsymbol{s}}_c$ using disentangled pitch $\boldsymbol{p}_c$ and timbre $\boldsymbol{ti}_c$. Inspired by the FiLM [25], we first split $\boldsymbol{p}_c$ into $\boldsymbol{p}_c^\gamma$ and $\boldsymbol{p}_c^\beta$, where $\boldsymbol{p}_c = [\boldsymbol{p}_c^\gamma \, \boldsymbol{p}_c^\beta]$. Then, we entangle $\boldsymbol{p}_c$ and $\boldsymbol{ti}_c$ together to produce $\widehat{\boldsymbol{S}}_c$:

$$\boldsymbol{z}_c = \boldsymbol{p}_c^\gamma \odot \boldsymbol{ti}_c + \boldsymbol{p}_c^\beta, \tag{12}$$

$$\widehat{\boldsymbol{S}}_c = \text{Decoder}(\boldsymbol{z}_c), \qquad (13)$$

and the separation loss is:

$$L_{\text{separation}} = \left\| \boldsymbol{S}_c - \widehat{\boldsymbol{S}}_c \right\|_1. \qquad (14)$$

Different from previous MSS works, we apply a separation loss and a transcription loss to train the proposed model. The transcription loss is:

$$L_{\text{transcription}} = \text{Cross\_entropy}(\boldsymbol{y}_c, \widehat{\boldsymbol{y}}_c). \qquad (15)$$

Here, $\boldsymbol{y}_c \in [0,1]^{T \times N}$ is the groundtruth of scores. The aggregated loss function is:

$$L = L_{\text{query}} + L_{\text{transcription}} + L_{\text{separation}}. \qquad (16)$$

The aggregated loss L drives the proposed model to be a *multi-task score-informed* model rather than a synthesizer due to the lack of inductive biases for further timbre disentanglement.

3.4 Pitch-translation Invariance Loss

We propose a pitch-translation invariance loss to further improve the timbre disentanglement performance. Based on the pitch-translation invariance, we assume that when the audio pitches with the corresponding MIDI is shifted within a certain interval, the timbre is unchanged.

We shift the pitch of $\boldsymbol{s}_c$ to generate an augmented audio $\boldsymbol{s}_c'$. The augmented audio $\boldsymbol{s}_c'$ has the same timbre as $\boldsymbol{s}_c$. According to Equation (1), we have a new mixture audio $\boldsymbol{x}'$:

$$\boldsymbol{x}' = \boldsymbol{s}_c' + \boldsymbol{s}_i. \qquad (17)$$

We denote the $\boldsymbol{X}'$ and $\boldsymbol{S}_c'$ as the spectrograms of $\boldsymbol{x}'$ and $\boldsymbol{s}_c'$ respectively. We extract the disentangled timbre vector of $\boldsymbol{s}_c$, and denote it as $\boldsymbol{ti}_c'$. Because $\boldsymbol{s}_c'$ is pitch shifted $\boldsymbol{s}_c$, so that the timbre $\boldsymbol{ti}_c'$ should be consistent with that of $\boldsymbol{ti}_c$. Therefore, the reconstructed spectrogram by the timbre $\boldsymbol{ti}_c'$ and the pitch $\boldsymbol{p}_c$ should be consistent with $\boldsymbol{S}_c$:

$$\widehat{\boldsymbol{z}}_c' = \boldsymbol{p}_c^{\gamma} \odot \boldsymbol{ti}_c' + \boldsymbol{p}_c^{\beta}, \qquad (18)$$

$$\widehat{\boldsymbol{S}}_c'' = \text{Decoder}(\boldsymbol{z}_c'), \qquad (19)$$

$$L_{\text{PTI}} = \left\| \boldsymbol{S}_c - \widehat{\boldsymbol{S}}_c'' \right\|_1. \qquad (20)$$

where $\boldsymbol{S}_c''$ is the reconstructed spectrogram. We denote L_{PTI} as a pitch-translation invariance loss. With L_{PTI}, our proposed model is capable of learning the disentanglement of pitch and timbre. A byproduct of the disentanglement system is that, the decoder of our system becomes a *synthesizer*, which can be used to synthesize audio recordings using timbre and pitches as input. When we change $\widehat{\boldsymbol{y}}_c$ to arbitrary scores, our model can synthesis a new piece of music with the timbre of $\boldsymbol{S}_c$.

In total, the objective function we exploit to train the proposed model with further disentanglement includes a QueryNet loss, a transcription loss, and a pitch-translation invariance loss:

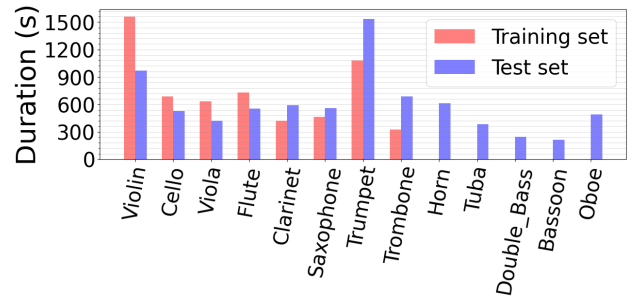$$L' = L_{\text{query}} + L_{\text{transcription}} + L_{\text{PTI}}. \qquad (21)$$

Figure 2. Duration of each instrument in the dataset.

4. EXPERIMENTS

4.1 Dataset and Pre-processing

We utilize the University of Rochester Multimodal Music Performance (URMP) dataset [26] as the experimental dataset. The URMP dataset is a multi-instrument audio-visual dataset covering 44 classical chamber music pieces remixed from 115 single-source tracks of 13 different *monophonic* instruments. The dataset provides note annotations for each single track. As shown in Figure 2, we divide these instruments into two groups (8 seen and 5 unseen instruments) and tracks into two sub-sets (55 tracks of 8 seen instruments for training and 32 songs by remixing 60 tracks of 13 instruments for test). Note that we calculate the duration of repeated tracks of different songs in the test set and do not exclude silence segments of all the tracks.

We resample all the tracks with a sample rate of 16KHz and extract them into Short-time Fourier transform (STFT) spectrograms with a window size of 1024 and 10ms overlap ($n_{\text{FFT}} = 2048$). During training, we randomly remix *2 arbitrary clips of different instruments* to generate a mixture. All the training data are augmented using pitch shifting (± 4 semitones) mentioned in Section 3.4.

4.2 Model Architecture

We design our models based on U-Net, the current prominent model in MSS. Figure 1(b) and 3 elaborates details of the proposed multi-task score-informed model (MSI) described in Section 3.3 and model with further disentanglement (MSI-DIS) illustrated in Section 3.4.

4.2.1 The Architecture of the MSI and MSI-DIS Model

The combination of the encoder and decoder is a general U-Net without temporal pooling. The QueryNet comprises 2 CNN blocks, each of which consists of 2 convolution layers and a 2×2 max pooling module. A fully-connected layer and a tanh activation layer are applied to the last feature maps. We then average output vectors over the temporal axis to get a 6-dimensional query embedding vector $\boldsymbol{q}_c$. The architecture of the transcriptor is similar to the QueryNet but without temporal pooling. Each blue block in TimbreFilter depicted in Figure 3 is a 2-dimension convolutional layer, the shape of the tensor output by which is as same as that of the input tensor. Each deep blue block in PitchExtractor is a 1-dimension 1×1 convolutional layer.
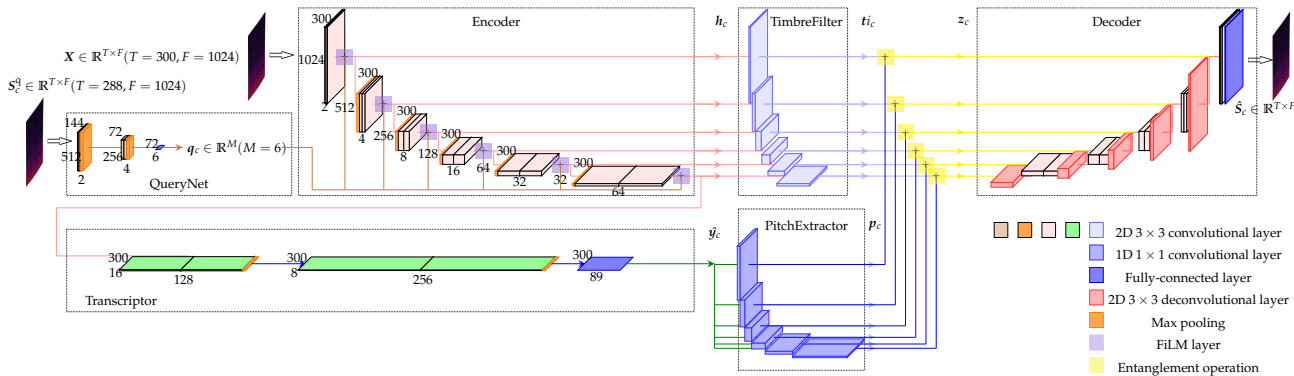
Figure 3. The model architecture with detailed hyper-parameter configuration.

Model	Separation (SDR)				Transcription (Precision)			
	MSS-only	Multi-task	MSI(ours)	MSI-DIS(ours)	AMT-only	Multi-task	MSI(ours)	MSI-DIS(ours)
Seen	4.69 ± 0.31	3.32 ± 0.18	$\mathbf{6.33 \pm 0.17}$	5.04 ± 0.16	$\mathbf{0.72 \pm 0.01}$	$\mathbf{0.72 \pm 0.04}$	0.71 ± 0.01	$\mathbf{0.72 \pm 0.01}$
Unseen	$\mathbf{6.20 \pm 0.26}$	4.63 ± 0.34	5.53 ± 0.15	3.99 ± 0.22	$\mathbf{0.61 \pm 0.01}$	0.58 ± 0.02	$\mathbf{0.61 \pm 0.01}$	0.59 ± 0.01
Overall	5.07 ± 0.22	3.65 ± 0.22	$\mathbf{6.13 \pm 0.15}$	4.77 ± 0.14	$\mathbf{0.69 \pm 0.01}$	0.69 ± 0.03	0.69 ± 0.01	0.68 ± 0.00

Table 1. The separation and transcription performance of all models..
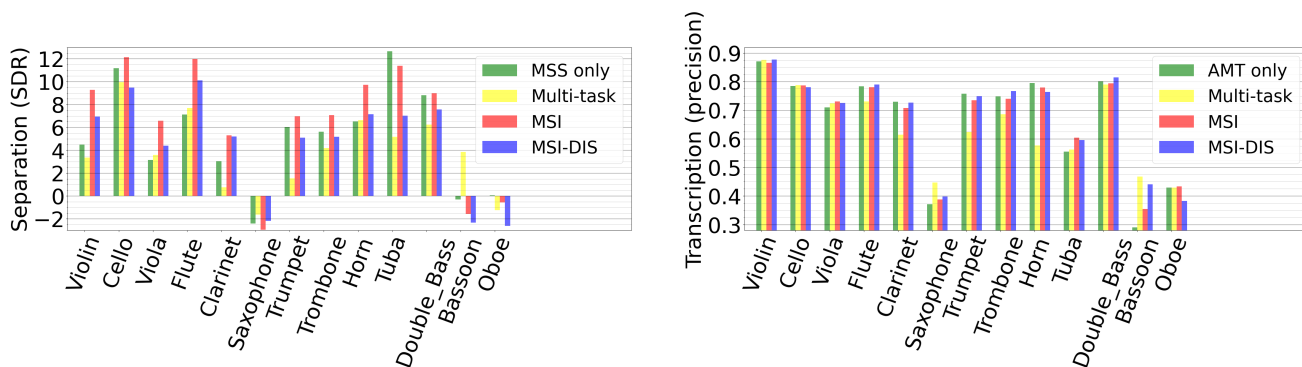

Figure 4. Instrument-wise performance of models. The last 5 instruments are unseen in the training set.

Typically, the bottleneck of U-Net is regarded as h_c. However, when constructing disentangled timbre representations, we regard the set of concatenate residual tensors as h_c to avoid non-disentangled representations leaking into the decoder.

Note that the kernel size of each 2-dimension convolutional layer is 3×3 and each 2-dimension convolutional layer (excepting TimbreFilter) is followed by a ReLU activation layer and a Batch Normalization layer.

4.2.2 Baseline Design

As shown in Figure 1(a), besides the proposed models illustrated above, we also report the performance of 3 extra baseline models in our experimental results. The QBE transcription-only baseline model (AMT-only) is composed of the queryNet, encoder, and transcriptor; the QBE separation-only baseline model (MSS-only) is a general U-Net; the QBE multi-task baseline model is composed of a U-Net and a transcriptor. All the hyper-parameters of components in these models are consistent with those of corresponding components in our models.

4.3 Training and Evaluation

All the models are trained with a mini-batch of 12 audio pairs for 200 epochs. All the models are evaluated with source-to-distortion (SDR) computed by mir_eval pakage [27] for separation and precision computed by sklearn package [28] for transcription. During training, each audio pair comprises 2 single-track audio clips of different instruments to generate a mixture, 2 correspondings augmented samples for pitch-transformation invariance loss, and 3 single-track audio clips that exclude silence segments for contrastive loss. During inference, each test pair comprises a 4-second audio mixture and query sample. During synthesis, we employ Griffin Lim Algorithm (GLA) [29] as the phase vocoder using torchaudio library. Since we do not divide a validation set to chose the best-performance model among all the training epochs, we report Micro-average results with a 95% confidence interval (CI) of models at the last 10 epochs. All the experimental results are reproducible via our realeased source code [1].

---

[1] https://github.com/kikyo-16/a-unified-model-for-zero-shot-musical-source-separation-transcription-and-synthesis
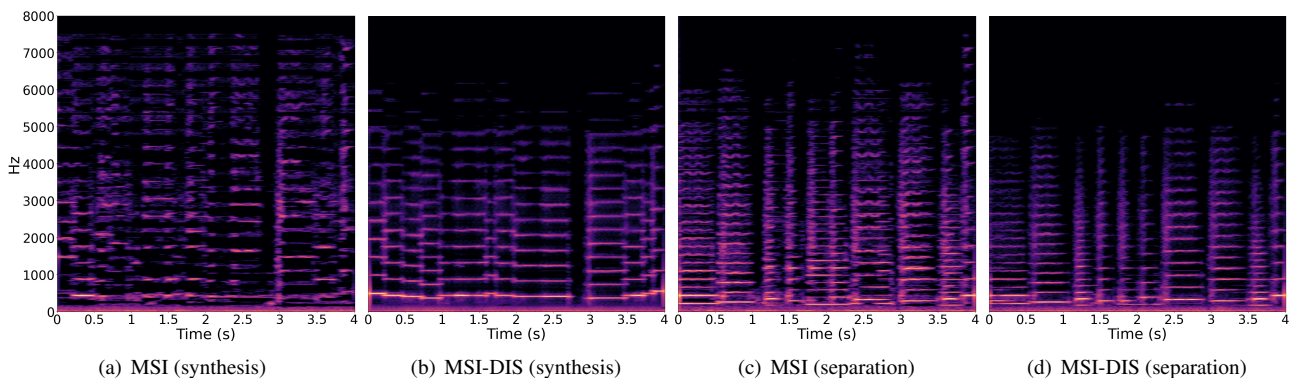
Figure 5. Spectrograms of audios synthesized and separated by the MSI and MSI-DIS model respectively. Models are expected to separate a viola source from the mixture of clarinet and viola. During synthesis, the two models are given the same new scores and are expected to synthesis new pieces with these scores and the separated viola timbre.

5. RESULTS

Experimental results shown in Table 1 demonstrate that the proposed MSI model outperforms baselines on separation without sacrificing performance on transcription. The instrument-wise performance on unseen instruments depicted in Figure 4 demonstrates that the proposed models are capable of performing zero-shot transcription and separation. We also release synthesized audio demos online [2]. These demos demonstrate the success of the proposed inductive biases for disentanglement.

5.1 Multi-task Baseline vs Single-task Baselines

As shown in Table 1, the multi-task baseline performs worse than the separation-only baseline, suggesting that the joint representation requires extra inductive biases to learn better generalization, i.e. deep clustering in Cerberus [14]. Our disentanglement strategy provides such inductive biases.

5.2 MSI model vs Baselines

With the auxiliary of the proposed pitch-timbre disentanglement module, compared with the multi-task baseline, the performance of MSI on separation becomes better. This indicates that the disentanglement module improve the generalization capability of the joint representation, leading to better separation results. Meanwhile, MSI outperforms the MSS-only baseline on separation by 1.06 points. This demonstrates that inaccurate scores transcribed by the model itself sever as a powerful auxiliary for separation.

5.3 MSI Model vs MSI-DIS Model

As depicted in Figure 5(a) and 5(b), it is interesting that despite the same ''hardware'' (neural network design) of the two models, the MSI model fails to synthesis but the MSI-DIS model achieves. It exactly demonstrates that the designed ''soft-ware'' (the pitch translation loss) takes effect on the success of the disentanglement. As for separation performance shown in Table 1, the MSI-DIS model

falls behind the MSI model. The observation that better synthesis quality does not implies better separation performance suggests a trade-off between disentanglement and reconstruction. It indicates that extra (well-suited) inductive biases are required to further improve pitch and timbre disentanglement at the same time reduce the loss of information necessary for reconstruction.

Comparing the performance on seen with unseen instruments shown in Table 1, we find that the separation quality of the MSI-DIS model is more sensitive to the accuracy of transcription results than that of the MSI model. This is because the MSI-DIS model synthesizes instead of separating sources, for which the separation performance of it relies more on the accuracy of transcription results and the capability of the decoder than the MSI model does. However, when comparing separated spectrograms shown in Figure 5(c) and 5(d), we find that the MSI model sometimes separates multiple pitches at the same time while the MSI-DIS model yields monophonic results that sound more "clean". We release more synthesized and separated audio demos online.

6. CONCLUSION AND FUTURE WORKS

We contributed a unified model for zero-shot music source separation, transcription, and synthesis via pitch and timbre disentanglement. The main novelty lies in the disentanglement-and-reconstruction methodology for source separation, which naturally empowers the model with transcription and synthesis capabilities. In addition, we designed well-suited inductive bias including pitch vector quantization and pitch-translation invariant timbre loss to achieve better disentanglement. Lastly, we successfully integrate the model with a query-based networks, so that all three tasks can be achieved in a zero-shot fashion for unseen sound sources. Experiments demonstrated the zero-shot capability of the model and the powerful auxiliary of disentangled pitch information to separation. Results of synthesized audio pieces further exhibit that the disentangled factors are well generalized. In the future, we plan to extent the proposed framework for multi-instrument and vocal scenarios as well as high-fidelity synthesis.

---

[2] https://kikyo-16.github.io/demo-page-of-a-unified-model-for-separation-transcriptiion-synthesis

7. REFERENCES

[1] T. Yoshioka, H. Erdogan, Z. Chen, and F. Alleva, "Multi-microphone neural speech separation for far-field multi-talker speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5739–5743.

[2] T. Yoshioka, H. Erdogan, Z. Chen, and F. Alleva, "Multi-microphone neural speech separation for far-field multi-talker speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5739–5743.

[3] J. H. Lee, H.-S. Choi, and K. Lee, "Audio query-based music source separation," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[4] P. Seetharaman, G. Wichern, S. Venkataramani, and J. Le Roux, "Class-conditional embeddings for music source separation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 301–305.

[5] D. Samuel, A. Ganeshan, and J. Naradowsky, "Meta-learning extractors for music source separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 816–820.

[6] E. Manilow, G. Wichern, and J. Le Roux, "Hierarchical musical instrument separation," in *21st International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[7] J. F. Woodruff, B. Pardo, and R. B. Dannenberg, "Remixing stereo music with score-informed source separation." in *ISMIR*. Citeseer, 2006, pp. 314–319.

[8] M. Miron, J. Janer Mestres, and E. Gómez Gutiérrez, "Monaural score-informed source separation for classical music using convolutional neural networks," in *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[9] S. Ewert and M. B. Sandler, "Structured dropout for weak label and multi-instance learning and its application to score-informed source separation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 2277–2281.

[10] M. Gover, "Score-informed source separation of choral music," in *21st International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[11] G. Meseguer-Brocal and G. Peeters, "Content based singing voice source separation via strong conditioning using aligned phonemes," in *21st International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[12] H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba, "The sound of pixels," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 570–586.

[13] C. Gan, D. Huang, H. Zhao, J. B. Tenenbaum, and A. Torralba, "Music gesture for visual sound separation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10 478–10 487.

[14] E. Manilow, P. Seetharaman, and B. Pardo, "Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 771–775.

[15] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep u-net convolutional networks," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 23–27.

[16] N. Takahashi, N. Goswami, and Y. Mitsufuji, "Mmdenselstm: An efficient combination of convolutional and recurrent neural networks for audio source separation," in *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*. IEEE, 2018, pp. 106–110.

[17] Y.-N. Hung and A. Lerch, "Multitask learning for instrument activation aware music source separation," in *21st International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[18] K. Tanaka, T. Nakatsuka, R. Nishikimi, K. Yoshii, and S. Morishima, "Multi-instrument music transcription based on deep spherical clustering of spectrograms and pitchgrams," in *21st International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[19] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 31–35.

[20] Z.-Q. Wang, J. Le Roux, and J. R. Hershey, "Alternative objective functions for deep clustering," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 686–690.

[21] Y. Luo, Z. Chen, and N. Mesgarani, "Speaker-independent speech separation with deep attractor network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 4, pp. 787–796, 2018.

[22] R. Kumar, Y. Luo, and N. Mesgarani, "Music source activity detection and separation using deep attractor network." in *INTERSPEECH*, 2018, pp. 347–351.

[23] C.-B. Jeon, H.-S. Choi, and K. Lee, "Exploring aligned lyrics-informed singing voice separation," in *21st International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[24] G. Meseguer-Brocal and G. Peeters, "Conditioned-u-net: Introducing a control mechanism in the u-net for multiple source separations," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[25] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[26] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, "Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 522–535, 2018.

[27] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, "Mir_eval: A transparent implementation of common mir metrics," in *In Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR*. Citeseer, 2014.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[29] A. Sharma, P. Kumar, V. Maddukuri, N. Madamshetti, K. Kishore, S. S. S. Kavuru, B. Raman, and P. P. Roy, "Fast griffin lim based waveform generation strategy for text-to-speech synthesis," *Multimedia Tools and Applications*, vol. 79, no. 41, pp. 30 205–30 233, 2020.