DON'T SEPARATE, LEARN TO REMIX: END-TO-END NEURAL REMIXING WITH JOINT OPTIMIZATION

Haici Yang[1], *Shivani Firodiya*[1], *Nicholas J. Bryan*[2], *Minje Kim*[1]

[1]Indiana University, Luddy School of Informatics, Computing, and Engineering, Bloomington, IN, USA
[2]Adobe Research, San Francisco, CA, USA

ABSTRACT

The task of manipulating the level and/or effects of individual instruments to recompose a mixture of recordings, or remixing, is common across a variety of applications such as music production, audio-visual post-production, podcasts, and more. This process, however, traditionally requires access to individual source recordings, restricting the creative process. To work around this, source separation algorithms can separate a mixture into its respective components. Then, a user can adjust their levels and mix them back together. This two-step approach, however, still suffers from audible artifacts and motivates further work. In this work, we re-purpose Conv-TasNet, a well-known source separation model, into two neural remixing architectures that learn to remix directly rather than just to separate sources. We use an explicit loss term that directly measures remix quality and jointly optimize it with a separation loss. We evaluate our methods using the Slakh and MUSDB18 datasets and report remixing performance as well as the impact on source separation as a byproduct. Our results suggest that learning-to-remix significantly outperforms a strong separation baseline and is particularly useful for small volume changes.

Index Terms— Music remix, source separation

1. INTRODUCTION

Remixing or the task of manipulating the level and/or effects of individual instruments to create a derivative recording is widely used for audio and music production applications such as music content creation (e.g., DJ performances), audio-visual post-production, remastering, podcasting, and more. Music remixing, in particular, is of critical interest and can be used to modify an original version of a song into a different version to suit a specific genre, e.g., from country to rock; or to alter the sound stage, e.g., re-position an instrument's stereophonic location from the center to the left.

In this paper, we focus on the application where a user wants to *boost* or *suppress* arbitrary instruments differently. This is a significant challenge for a computer algorithm as those multiple sound sources are recorded altogether as a single mixture. Approaches to solve this problem can be categorized into two kinds: feature-based methods [1, 2] and music source separation-based methods [3, 4]. Feature-based remixing systems can adjust a single instrument at a time by learning its specific feature. This kind of methods has been successful on processing drums, in both attenuation (-10 and -6 dB) and amplification (10 dB) tasks but has had limited effect on other instruments. In contrast, music source separation (MSS) algorithms allow users to manipulate estimates of multiple separated instruments to achieve remixing. However, in this MSS context, source separation and remixing are treated as two independent processes, where remixing serves as a post-processor that can be especially vulnerable to the separation artifacts and other failures. Therefore, lots of recent progress on remixing is related to the development of MSS, especially those with deep learning methods [5, 6, 7] in both the frequency-domain [8, 9, 10, 11, 12, 13] and time-domain [14, 15, 16, 17].

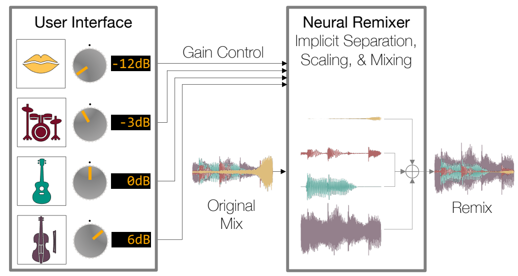


Fig. 1: We perform end-to-end remixing by taking the intended gain controls to train the model on the target remix directly.

In our proposal, we minimize separation artifacts found in remixing systems by learning to remix directly. We first justify our problem formulation with analysis on commonly used source separation evaluation metrics. Then, we extend one of the state-of-art source separation models, Conv-TasNet [17], and propose two adaptations toward end-to-end remixing, (a) one that applies remixing weights to the source estimates, and (b) another one applies weights to the latent variables, given that latent interacting have proven to be beneficial for source separation and related problems [18, 19]. Both models are optimized to perform separation and remixing simultaneously. Although combining the reconstruction terms of both multiple sources and mixture has been used to improve source separation performance [20, 21], we uniquely focus on the end goal of remixing. We evaluate the proposed methods on two music source separation datasets, Slakh [22] and MUSDB18 [23], and analyze the behavior of the models on various remixing scenarios. To summarize, we make the following key contributions:

- To the best of our knowledge, we propose the first end-to-end neural method that jointly learns MSS and remixing together.
- We show our proposed neural remixing method is capable of a wider range of volume change compared to existing methods, ranging from -12 to 12 dB, and can deal with up to five sources.

2. PROBLEM FORMULATION

We argue that the naïve concatenation of the separation and remixing processes is artifact-prone. In a two-source case, for example, we decompose the recovered source into three components [24]:

$$\hat{\boldsymbol{s}}_1 = \alpha_1 \boldsymbol{s}_1 + \beta_1 \boldsymbol{s}_2 + \boldsymbol{e}_1, \quad \hat{\boldsymbol{s}}_2 = \alpha_2 \boldsymbol{s}_2 + \beta_2 \boldsymbol{s}_1 + \boldsymbol{e}_2. \tag{1}$$

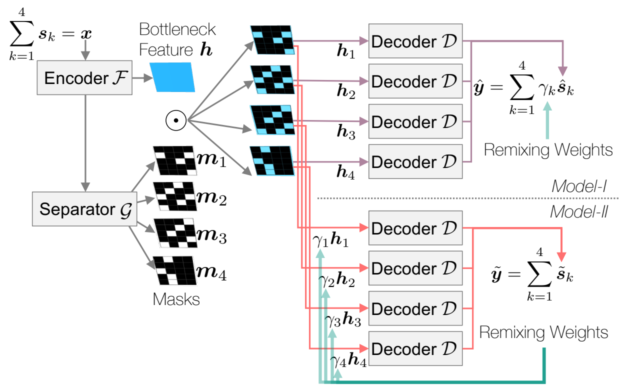


Fig. 2: Our neural remixer architectures. *Model-I* jointly optimizes a separation and remix loss. *Model-II* extends this further by performing remixing in the model latent space.

The reconstruction of the first source $\hat{\boldsymbol{s}}_1$, consists of the scaled ground-truth $\alpha_1\boldsymbol{s}_1$, scaled interference $\beta_1\boldsymbol{s}_2$, and the artifact generated during the separation process $\boldsymbol{e}_1$. The perfect scale-invariant separation is achieved when $\beta_1, \beta_2 = 0$ and $\boldsymbol{e}_1, \boldsymbol{e}_2 = \boldsymbol{0}$.

Here, we introduce a pair of non-negative remixing weights γ_1 and γ_2 as the intended scaling factors. The estimated remix $\hat{y}$ is derived from the reconstruction of sources:

$$\begin{aligned}\hat{y} &= \gamma_1\hat{\boldsymbol{s}}_1 + \gamma_2\hat{\boldsymbol{s}}_2 \\ &= (\gamma_1\alpha_1 + \gamma_2\beta_2)\boldsymbol{s}_1 + (\gamma_1\beta_1 + \gamma_2\alpha_2)\boldsymbol{s}_2 + \gamma_1\boldsymbol{e}_1 + \gamma_2\boldsymbol{e}_2.\end{aligned} \tag{2}$$

Clearly, imperfect separation can cause compromised weighting because the approximation $\gamma_1 \approx \gamma_1\alpha_1 + \gamma_2\beta_2$ is inaccurate if β_2 is too large or α_1 is too small. Furthermore, the artifact $\gamma_1\boldsymbol{e}_1 + \gamma_2\boldsymbol{e}_2$ is not guaranteed to cancel each other after scaling. Thus, we seek an alternative approach.

3. METHODOLOGY

We propose to jointly learn source separation and remixing objectives. Let L_k be the user's intended volume change amount for source k in terms of sound pressure level (SPL) relative to the original loudness, $L_k = 10\log_{10}(\gamma_k\boldsymbol{s}_k)^\top(\gamma_k\boldsymbol{s}_k)/\boldsymbol{s}_k^\top\boldsymbol{s}_k$, where γ_k denotes the corresponding remixing amplitude scaling. For example, $L_k = +10$ dB is converted into $\gamma_k = 3.16$. For a K-source mixture, the remix target is defined as $\boldsymbol{y} = \sum_{k=1}^{K}\gamma_k\boldsymbol{s}_k$, while the input mixture is $\boldsymbol{x} = \sum_{k=1}^{K}\boldsymbol{s}_k$.

3.1. The baseline: remixing estimated sources

Our baseline utilizes Conv-TasNet as a source separation module that takes time-domain signals as input and computes the loss in the time domain as well, i.e., $\mathcal{L}_{\text{BL}} = \sum_{k=1}^{K}\mathcal{E}(\boldsymbol{s}_k||\hat{\boldsymbol{s}}_k)$, where we use `bss_eval_images`'s SDR [24] for the error function $\mathcal{E}(\cdot||\cdot)$. Then, the user's intended source-specific scales γ_k are applied to the source estimates to approximate the remix with $\hat{\boldsymbol{y}}$: $\boldsymbol{y} \approx \hat{\boldsymbol{y}} = \sum_{k=1}^{K}\gamma_k\hat{\boldsymbol{s}}_k$. Note that this process is prone to inaccuracy in source control and artifacts as shown in eq. (2).

3.2. *Model-I*: the proposed remixing loss

We propose incorporating the remixing process in training—synthesizing the remix using the recovered sources, and comparing it to the target remix to compute the loss. Hence, our *Model-I* uses a loss that penalizes insufficient remix quality as well as the source-specific reconstruction:

$$\mathcal{L}_{\textit{Model-I}} = \psi\mathcal{E}(\boldsymbol{y}||\hat{\boldsymbol{y}}) + \lambda\sum_{k=1}^{K}\mathcal{E}(\boldsymbol{s}_k||\hat{\boldsymbol{s}}_k), \tag{3}$$

where ψ and λ control the contribution of the remix and source-specific reconstruction losses, respectively. The rationale behind our proposed loss is to calibrate the optimization results of source separation by imposing a cost on the remix quality. We also believe this change can help the system balance the SIR-SAR trade-off for a better remix quality.

3.3. *Model-II*: the proposed control of the latent space

Our second proposed approach is to control the loudness of sources in the latent space. Unlike *Model-I*, *Model-II* applies the remixing weights γ_k to one of the network's hidden layers, instead of source estimates as shown in Fig. 2. Conv-TasNet provides a convenient framework to do this. In Conv-TasNet, the separation is conducted first by computing the bottleneck feature map via the *encoder* module $\mathcal{F}(\cdot)$ and K masks using the *separator* module $\mathcal{G}(\cdot)$,

$$[\boldsymbol{m}_1, \boldsymbol{m}_2, \ldots, \boldsymbol{m}_K] \leftarrow \mathcal{G}(\boldsymbol{x}), \quad \boldsymbol{h} \leftarrow \mathcal{F}(\boldsymbol{x}), \tag{4}$$

where the masks are probabilistic, i.e., $\sum_{k=1}^{K}\boldsymbol{m}_k = \boldsymbol{1}$. Then, the bottleneck feature $\boldsymbol{h}$ is distributed to K different source-specific feature spaces via masking, which are then decoded back into the time-domain using the *decoder* module $\mathcal{D}(\cdot)$, respectively:

$$\boldsymbol{h}_k \leftarrow \boldsymbol{m}_k \odot \boldsymbol{h}, \quad \hat{\boldsymbol{s}}_k \leftarrow \mathcal{D}(\boldsymbol{h}_k), \quad \forall k, \tag{5}$$

where $\odot$ denotes Hadamard product.

By making use of the separated hidden space, *Model-II* modulates the hidden variables by multiplying their corresponding remixing weights γ_k: $\tilde{\boldsymbol{s}}_k \leftarrow \mathcal{D}(\gamma_k\boldsymbol{h}_k)$, where the output $\tilde{\boldsymbol{s}}_k$ attempts to reconstruct scaled sources $\gamma_k\boldsymbol{s}_k$ directly. *Model-II*'s loss is then

$$\mathcal{L}_{\textit{Model-II}} = \psi\mathcal{E}(\boldsymbol{y}||\tilde{\boldsymbol{y}}) + \lambda\sum_{k=1}^{K}\mathcal{E}(\gamma_k\boldsymbol{s}_k||\tilde{\boldsymbol{s}}_k), \tag{6}$$

where $\tilde{\boldsymbol{y}} = \sum_{k=1}^{K}\tilde{\boldsymbol{s}}_k$.

In theory, *Model-II* has the advantage over *Model-I*, because by involving the remixing weights into the feed-forward process, *Model-II* can potentially associate the separation behavior with the remix weights, while *Model-I* only produces a universal separation output to be used for all kinds remixing weights. Therefore, if a user wants to boost the j-th source while suppressing others, $\gamma_j \gg \gamma_k, \forall k \neq j$, *Model-II* can focus more on the precise reconstruction of the dominant source $\boldsymbol{s}_j$ than the other sources that will be suppressed anyway. Similarly, $\gamma_k\boldsymbol{e}_k$ is small to ignore when $k \neq j$, while suppressing $\boldsymbol{e}_j$ is important as γ_j is large. Another important corner case is when the remix target is very similar to the input, i.e., $\boldsymbol{y} \approx \boldsymbol{x}$ when $\gamma_k \approx 1, \forall k$. In this trivial case, *Model-II* can save unnecessary separation effort.

4. EXPERIMENTS

4.1. Datasets

We use two music datasets: Slakh [22] and MUSDB18 [23]. While Slakh consists of a large number of stem tracks synthesized using

minSDR / LD		Baseline	*Model-I*			*Model-II*		
Train + Test	K	$(\psi : \lambda) = (0 : 1)$	$(\psi : \lambda) = (1 : 1)$	$(\psi : \lambda) = (K : 1)$	$(\psi : \lambda) = (1 : 0)$	$(\psi : \lambda) = (1 : 1)$	$(\psi : \lambda) = (K : 1)$	$(\psi : \lambda) = (1 : 0)$
Slakh + Slakh	2	28.24 / 0.18	24.59 / 0.31	27.63 / 0.21	**28.84 / 0.19**	27.35 / 0.19	28.34 / 0.21	27.16 / 0.19
	3	18.72 / 0.67	19.88 / 0.8	19.7 / 0.87	**21.26 / 0.67**	20.09 / 0.69	19.81 / 0.77	19.26 / 0.81
	4	0.22 / 8.42	16.48 / 1.54	15.24 / 1.85	15.57 / 1.72	16.8 / 1.57	15.16 / 1.79	**17.23 / 1.51**
	5	-4.08 / 11.31	7.92 / 3.87	12.2 / 3.2	11.71 / 3.34	8.24 / 3.86	**12.44 / 3.15**	11.5 / 3.45
MUSDB18 + Slakh	2	23.83 / 0.35	23.19 / 0.47	23.01 / 0.45	24.96 / 0.39	23.99 / 0.44	23.97 / 0.41	**25.15 / 0.35**
	3	11.88 / 1.64	14.13 / 1.72	13.37 / 1.94	**15.3 / 1.6**	15.2 / 1.56	14.76 / 1.49	15.15 / 1.68
	4	-6.06 / 7.85	9.74 / 2.78	9.94 / 2.8	9.19 / 3.05	9.63 / 2.88	**10.2 / 2.78**	9.73 / 3.01
MUSDB18 + MUSDB18	2	17.33 / 0.92	17.55 / 0.98	17.28 / 0.88	18.08 / 0.95	17.7 / 0.96	17.87 / 0.84	**18.13 / 0.97**
	3	11.82 / 1.94	13.37 / 1.93	12.52 / 2.29	**14.49 / 1.72**	14.17 / 1.71	14.13 / 1.64	14.15 / 1.94
	4	-9.16 / 10.1	10.16 / 2.93	**11.01 / 2.85**	9.84 / 3.26	10.49 / 2.97	10.95 / 3.0	10.01 / 3.27
Slakh + MUSDB18	2	12.26 / 1.61	14.54 / 1.31	14.54 / 1.39	14.71 / 1.36	14.25 / 1.42	**15.1 / 1.29**	13.43 / 1.56
	3	8.27 / 2.59	9.37 / 2.85	10.16 / 2.73	10.21 / 2.75	9.69 / 2.72	10.18 / 2.62	**10.57 / 2.48**
	4	-6.33 / 9.88	7.46 / 3.77	**8.44 / 3.66**	8.34 / 3.65	7.75 / 3.68	8.29 / 3.68	8.06 / 3.76

Table 1: Cross-dataset evaluation on *minSDR* and loudness difference (LD) for re-mixture construction.

virtual instruments, the MUSDB18 samples are real professional-level recordings. However, the quantity of MUSDB18 is often not enough to train a large model that generalizes well to real-world music signals. We first train and test within each dataset independently. Cross-database testing follows, i.e., train on Slakh and test on MUSDB18, and vice versa. In MUSDB18, we use the sources in the order of `vocals`, `drums`, `bass`, and `other`. For two-source experiments, for example, we use `vocals` and `drums` to build the mixture. As for Slakh, we test up to five sources in the following order: `piano`, `drums`, `bass`, `guitars` and `string`.

4.2. Training setup

We build our models and experiments off of Asteroid's Conv-TasNet implementation [25]. We fix the order of the sources and do not use permutation invariant training (PIT) [26]. We use the entire training set for training (∼48h for Slakh and ∼6h for MUSDB18), and split the original test set (∼12h and 3h respectively) into validation and test sets evenly. The model is trained on one-second segments until the validation loss does not improve in 50 consecutive epochs. Only segments with all instruments active are included during training as we found this was better than having segments with silent instruments in initial experiments. For evaluation, no such restriction is applied. Adam is used as the optimizer with an initial learning rate of 1×10^{-3} [27]. During training, the K remix weights are sampled randomly from the range between -12 to 12 dB and used to form the target remix segments. All signals are sampled at $44,100$ kHz.

For our training loss, we substitute the original scale-invariant signal-to-distortion ratio (SI-SDR) loss function [28] in Conv-TasNet with `bss_eval_images`'s SDR [24] (equals to the classical source-to-noise ratio (SNR)) to suit the scale-sensitive problem. The scale-dependent SDR (SD-SDR) proposed together with SI-SDR is potentially another choice for general scaling control. However, the nature of SD-SDR is that it is insensitive to large up-scaling factors, while our model is designed to tackle both downside and upside scales. We also did an initial experiment comparing SDR and SD-SDR as the loss and results support our choice for SDR.

4.3. Evaluation setup

For evaluation, we explicitly evaluate the remix task, by synthesizing several representative ground-truth remix mixtures and then comparing our estimated remix mixtures to this ground truth. To create the remixes, we manipulate one source at a time ranging from -24 to $+24$ dB with a step size of 3 dB, totaling 17 discretized values, and sum the evaluation scores of those different remix cases. This range is twice wider than that used for training. We repeat the experiment for all K sources, individually. Note, while we focus on the common one-source manipulation cases for evaluation, our models can control multiple sources at the same time. After several initial experiments, we decided on a minimal set of handcrafted values for the weights of ψ and λ, i.e. $(\phi : \lambda) = \{(1 : 1), (K : 1), (1 : 0)\}$, to best highlight the interaction between our separation loss and remixture loss. One can also view the baseline as a special case of *Model-I*, where the $(\phi : \lambda) = (0 : 1)$.

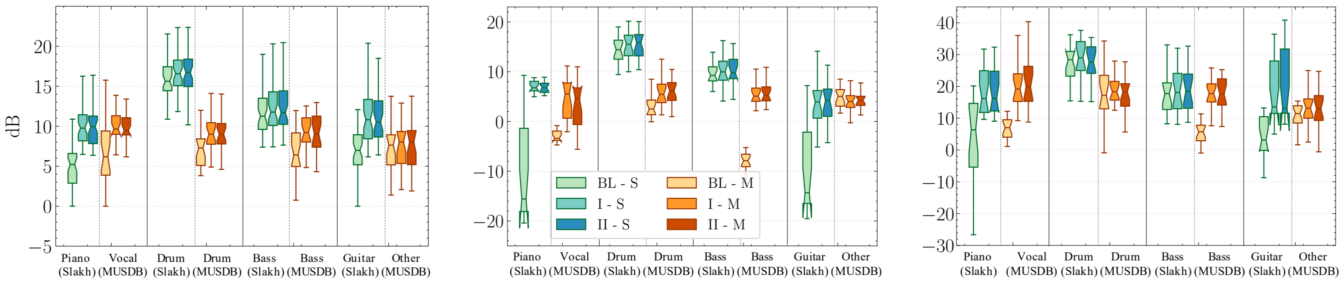
We objectively assess our models output based on the two criteria suggested in [29], sound quality and loudness balance, given the ground-truth remix mixtures. We considered using SDR or SD-SDR to measure sound quality. By nature, SD-SDR properly accounts for down-scaling errors and are less sensitive to up-scaling errors, while SDR performs in the opposite manner. Given that both the up-scaling and down-scaling in our problem is critical, we calculate both SDR and SD-SDR values and take the minimum of them as the final measure, per suggested by [28], and we denote it as *minSDR*. To evaluate the loudness balance, we consider the sources remixed in a linear time-invariant manner, and decompose the estimated re-mixture using a least squares algorithm. The coefficients obtained for each source in the process can be viewed as the actual scaling factors that have built the estimated re-mixture. Given the output and target scalars, we report the loudness difference in decibels. Finally, we additionally compute the SIR and SAR scores using `bss_eval_images` to investigate the impact of remixing on the separation behavior.

5. RESULTS AND DISCUSSION

5.1. Average remix performance

We report the *minSDR* and loudness difference of the estimated remix compared to the ground truth in Table 1. Overall, both our proposed models achieve a better remix quality than the baseline, and the improvement is more significant as the number of sources increases. The same trend can be found in the cross-database testing cases. We found no difference between *Model-I* and *II*.

When we look at the merit of our remix loss controlled by ψ, we find: 1) the remix-only loss ($(\lambda : \psi) = (0 : 1)$) is preferred in all but one $K = 2$ and $K = 3$ cases; 2) when the number of sources increases or the task gets harder (e.g., doing the cross-database test), it is beneficial to have higher λ for better separation control, i.e. $(\lambda : \psi) = (1 : 4)$, and 3) the proposed loss causes the significant performance gap between the baseline and proposed models, as the baseline model uses $\psi = 0$.



(a) ΔSDR (b) SAR (c) ΔSIR

Fig. 3: Source separation scores of the baseline (BL), *Model-I* (I), and *Model-II* (II) on Slakh (S) and MUSDB18 (M) for the four-source cases. The first and last box groups are `piano` and `guitars` in the Slakh experiments, and `vocals` and `others` in the MUSDB18 case.

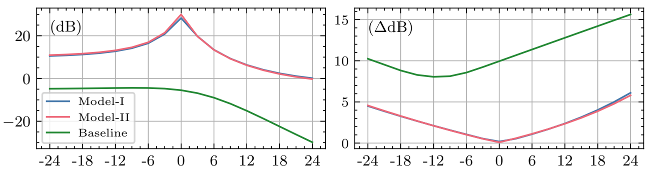


Fig. 4: Remix *minSDR* (left) and loudness difference (right) with respect to vocal volume adjustment.

5.2. Vocal remix vs. loudness performance

In Figure 4, we display the change of remix quality and loudness difference along with the extent of vocal volume control in the four-source cases using MUSDB18. The x-axis represents adjusting the level of vocal from -24 to $+24$ dB, while the other instruments are kept unchanged. This intends to mimic the scenario where the user can adjust the volume of one instrument source by turning a knob.

Figure 4 clearly shows that our proposed models outperform the baseline consistently in all the remixing weight choices, with higher reconstruction score and lower loudness difference. The trend is especially salient if the volume control amount is near 0 dB, that is when the intended remix is similar to the input mix. It is noticeable that the baseline's *minSDR* score decreases almost monotonically as the remix weights change, and the minimum loudness difference for the baseline model deviated from 0 dB. These patterns indicate that, when the target remix stays around the original mix, our proposed models show more advantage as they can effectively focus on the artifacts rather than trying to suppress interference. Note that *Model-I* addresses this issue only during training, while *Model-II* is capable of reflecting the remixing weights to the decoding process during testing. Meanwhile, for the baseline model, the artifact and interference contained in different source estimates stand out as distortion once they are remixed and summed up. Therefore, monotonic control of a source estimate can consequently monotonically influence the total remix quality. For the same reason, the loudness difference curve is asymmetric, where the low volume side has relatively smaller differences.

Because of the reason stated above, our proposed models' performance changes are more predictable—the performance graphs share a similar pattern in all experiments on various instruments, datasets, and both proposed models, providing a stable user experience. Their asymmetric shape is caused by our setup where we boost or suppress only one source: boosting tends to exhibit more artifacts. In contrast, the baseline's performance is less predictable. For some sources, volume amplification has a negative impact while others suffer from a reduced volume. This observation echoes the baseline's behavior reported in Figure 3, where performance varies a lot over the choice of source.

It is interesting to note that, although *Model-I* does not have any inference-time mechanism to adjust the separation behavior according to the different remixing weights, it still performs on par with *Model-II*. We believe that to achieve a good sound quality for remixing, it is most crucial to reduce the artifacts produced in the separation step, and these two proposed models almost reach the same level in pursuing this goal. However, we believe that *Model-II*'s source control in the latent space is a potentially more useful approach to more complex remixing tasks beyond volume adjustment such as source-specific nonlinear filtering.

5.3. Music source separation performance

To investigate the impact of our remix loss on the separation behavior, we compute the improvement of the SDR, SIR and SAR of the three models' separation results using the `BSS_Eval` toolbox [24] compared to the input SDR and SIR. For this experiment, we run the models on four-source mixtures, and set $(\lambda : \psi) = (1 : 4)$. Figure 3 summarizes the results.

Without any remix loss, the baseline fails in recovering certain instruments, i.e., `piano` and `guitars` in the Slakh experiments, and `vocals` and `bass` in the MUSDB18 case. We observe that the performance gap mostly comes from the SAR scores, while the SIR improvement is less drastic. This signifies our neural remixers' tendency to suppress artifacts as much as possible as the source-specific artifacts do not cancel out. Meanwhile, it also shows that the proposed remixing loss benefits general separation performance according to the SIR improvement.

6. CONCLUSION

We introduced a neural remixing model that works directly on the music mixture instead of assuming separated source tracks. Instead of a conventional separator-remixer workflow, we integrated the two processes into an end-to-end neural remixer via joint optimization. Results on Slakh and MUSDB18 showed that our proposed joint learning of remixing and separation greatly reduces artifacts caused by the process of source separation. Therefore, our models achieved significant improvement in the remix quality. From the perspective of user interaction, we demonstrated that the relationship between the estimated remix and the intended one is reasonably correlated as opposed to that induced from the baseline model. Sound examples and source code are available at `https://saige.sice.indiana.edu/research-projects/neural-remixer`

7. REFERENCES

[1] K. Yoshii, M. Goto, and H. Okuno, "INTER:D: a drum sound equalizer for controlling volume and timbre of drums," in *The 2nd European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology (EWIMT)*, 2005.

[2] O. Gillet and G. Richard, "Extraction and remixing of drum tracks from polyphonic music signals," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2005.

[3] J. F. Woodruff, B. Pardo, and R. B. Dannenberg, "Remixing stereo music with score-informed source separation." in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2006.

[4] K. Itoyama, M. Goto, K. Komatani, T. Ogata, and H. G. Okuno, "Instrument equalizer for query-by-example retrieval: Improving sound source separation based on integrated harmonic and inharmonic models." in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2008.

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

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

[7] F.-R. Stöter, S. Uhlich, A. Liutkus, and Y. Mitsufuji, "Open-unmix-a reference implementation for music source separation," *Journal of Open Source Software*, vol. 4, no. 41, p. 1667, 2019.

[8] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdis, "Joint optimization of masks and deep recurrent neural networks for monaural source separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2136–2147, Dec 2015.

[9] S. Uhlich *et al.*, "Improving music source separation based on deep neural networks through data augmentation and network blending," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2017.

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

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

[12] P. Seetharaman, G. Wichern, J. Le Roux, and B. Pardo, "Bootstrapping deep music separation from primitive auditory grouping principles," *Workshop on Self-supervision in Audio and Speech at International Conference on Machine Learning*, 2019.

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

[14] D. Stoller, S. Ewert, and S. Dixon, "Wave-U-Net: A multi-scale neural network for end-to-end audio source separation," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[15] F. Lluís, J. Pons, and X. Serra, "End-to-End Music Source Separation: Is it Possible in the Waveform Domain?" in *Proc. of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2019.

[16] Défossez *et al.*, "Music Source Separation in the Waveform Domain," *arXiv preprint arXiv:1911.13254*, 2019.

[17] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[18] H. Yang, K. Zhen, S. Beack, and M. Kim, "Source-aware neural speech coding for noisy speech compression," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2021.

[19] N. J. Bryan and G. J. Mysore, "An efficient posterior regularized latent variable model for interactive sound source separation," in *Proc. of the International Conference on Machine Learning (ICML)*, 2013.

[20] H.-S. Choi *et al.*, "Phase-aware speech enhancement with deep complex U-Net," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2018.

[21] R. Sawata, S. Uhlich, S. Takahashi, and Y. Mitsufuji, "All for one and one for all: Improving music separation by bridging networks," in *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

[22] 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. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2019.

[23] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "MUSDB18-HQ - an uncompressed version of MUSDB18," Aug. 2019. [Online]. Available: https://doi.org/10.5281/zenodo.3338373

[24] E. Vincent, C. Fevotte, and R. Gribonval, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[25] M. Pariente *et al.*, "Asteroid: The PyTorch-Based Audio Source Separation Toolkit for Researchers," in *Proc. of the Annual Conference of the International Speech Communication Association (Interspeech)*, 2020.

[26] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2017.

[27] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of the International Conference on Learning Representations (ICLR)*, 2015.

[28] J. Le Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "SDR – half-baked or well done?" in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2019.

[29] H. Wierstorf *et al.*, "Perceptual evaluation of source separation for remixing music," in *Audio Engineering Society Convention 143*, 2017.