

DIFF2MIX: CONTROLLABLE MUSIC MIXING VIA DIFFUSION MODELS AND DIFFERENTIABLE AUDIO EFFECTS

Yisu Zong Jinjie Shi Joshua Reiss

Centre for Digital Music, Queen Mary University of London, UK

y.zong@qmul.ac.uk

ABSTRACT

Automatic music mixing aims to combine multitrack recordings into a balanced and coherent musical piece. Because the content of different songs and the subjective preferences of mixing engineers jointly shape the final outcome, a practical system should deliver well-balanced mixes while allowing for controllable stylistic variation. However, most existing methods treat automatic mixing and mixing style control as separate tasks, making it difficult for a single system to produce high-quality mixes while remaining editable and style-aware. To address this limitation, this paper presents Diff2Mix, a generative automatic mixing system based on diffusion models and a differentiable mixing console. This system offers two levels of optional user control: a reference audio enables overall production style control, and the differentiable mixing console provides explicit audio effects parameters for interpretability and fine-grained optimization. We demonstrate our system’s competitive performance through both objective and subjective evaluations in terms of mixing quality and control ability. We provide code and audio samples at our project page ¹.

1. INTRODUCTION

Music mixing is a core stage of music production, where individual tracks are combined into a coherent and expressive final song through the application of audio effects such as equalization, compression, panning, and reverberation. In practice, mixing involves a tightly coupled set of technical and creative decisions. These include balancing audio sources, reducing masking, shaping the dynamics, controlling the spatial impression, and attaining an overall aesthetic goal [1]. Since these decisions depend both on the musical content and the subjective intent of the mix engineer, producing a high-quality mix remains a difficult and time-consuming task that typically requires substantial expertise. This challenge has motivated growing interest in intelligent music production systems that can automate or assist professional mixing workflows.

¹ <https://zys711.github.io/Diff2Mix>

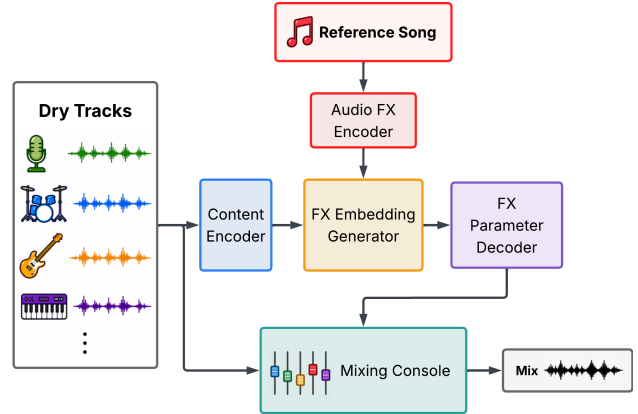


Figure 1. System overview of Diff2Mix.

Early approaches were dependent on hand-crafted rules derived from engineering practice, offering interpretability but limited flexibility and weak generalization to diverse musical material [2]. Later methods employed machine learning to learn mixing decisions directly from data. Among these methods, deep learning methods have achieved significant progress. These deep learning approaches can be generally classified into black-box systems [3, 4], which predict processed audio or full mixes in an end-to-end manner, and parameter estimation systems [5, 6], which infer control values for audio effects. Black-box models are capable of capturing complex nonlinear mappings; nevertheless, they might generate artifacts or overly averaged results. The applications of Differentiable Digital Signal Processing (DDSP) [7] in the audio research field have promoted the development of differentiable audio effects [8–10]. This presents a promising solution by incorporating audio effects into end-to-end parameter estimation systems [5, 11], enabling greater interpretability and control. However, they are commonly restricted to fixed effect chains and differentiable effects. Moreover, these methods typically formulate automatic mixing as a deterministic one-to-one mapping from input tracks to a single target mix. In practice, the same multitrack session can support multiple valid mixes, reflecting different production styles and artistic preferences. Recently, generative model approaches [12, 13] for automatic mixing have been put forward, which take into account the variety of mixing decisions and have achieved state-of-the-art mixing quality.

A central limitation of many previous approaches is that

they lack control over the production style, which is of great significance for collaboration with users. A substantial amount of research on production style transfer aims to capture the reference style and map it onto the target audio, including black-box methods [14, 15], parameter estimation methods [6, 16], and training-free optimization methods [17]. With the rapid advancement of large language models and multimodal models, text control could become a novel and promising production style control interface [18, 19]. However, the majority of these methods conduct style control on the mixes rather than the dry tracks. Such systems may struggle to capture the track-specific characteristics and inter-track interactions, and they have not been integrated with automatic mixing systems. Some work [6, 15] can perform style transfer of multitrack input given a reference, yet these systems only consider style similarity, not overall balance, which limits output mixing quality.

In this work, we propose a diffusion-based automatic multitrack mixing system that controls the overall production style using a reference song. First, we train a track-level audio effects embedding diffusion model conditioned on the musical content of dry tracks and the audio effects features of the reference song. Then, we compare two methods for translating the embedding into audio effects parameters via a differentiable mixing console: directly estimating the parameter using spectral loss, and estimating the parameter combination distribution. We conduct objective and subjective evaluations of both mixing quality and style transfer to demonstrate the effectiveness of our proposed method.

2. METHOD

The system architecture is shown in Figure 1. The first component is a diffusion model conditioned on a reference song’s production style, which generates the track-level audio effects embeddings. Subsequently, these embeddings are translated into the audio effects parameters via an audio effects parameter decoder, and these parameters and dry tracks are then passed to a differentiable mixing console to generate the mix.

2.1 Conditional Audio Effects Embedding Diffusion Model

Our diffusion model backbone is adapted from MEGAMI [12], a diffusion framework operating in the track-level audio effects embedding space. Let $Z = \{z_i\}_{i=1}^n$ denote the target audio effects embeddings and $C = \{c_i\}_{i=1}^n$ the content embeddings extracted from the input dry tracks, where n is the number of tracks. Given a reference song y_{ref} , our aim is to extract a global production-style embedding z_{ref} and model the conditional distribution $p_\theta(Z | C, z_{ref})$. In line with the EDM formulation [20], generation is cast as denoising under a continuous family of Gaussian perturbations rather than a discrete-time forward Markov chain. The forward diffusion process is defined as:

$$\tilde{z}_{i,\sigma} = z_i + \sigma \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, I_D), \quad i = 1, \dots, n \quad (1)$$

where z_i denotes the clean audio effects embedding of the i -th track, $\tilde{z}_{i,\sigma}$ is its noised latent at noise level σ , ϵ_i is Gaussian noise, and I_D is the $D \times D$ identity matrix with D the embedding dimension. The denoiser is trained to recover the clean embedding from the noised input conditioned on the content embeddings C and the reference style embedding z_{ref} . Equivalently, the network can be interpreted as learning the conditional score:

$$s_\theta(\tilde{Z}_\sigma, \sigma, C, z_{ref}) \approx \nabla_{\tilde{Z}_\sigma} \log p(\tilde{Z}_\sigma | C, z_{ref}) \quad (2)$$

The score network is designed to be permutation-equivariant with respect to the track set, so permuting the conditioning tracks leads to the same permutation in the predicted outputs. We implement this property with a Diffusion Transformer (DiT) [21] that applies self-attention over the noised embedding set $\tilde{Z}_\sigma$ and cross-attention to the conditioning set. In our setting, the reference style embedding z_{ref} is concatenated to each content embedding c_i before conditioning, so the network jointly receives track-wise musical content and global production-style information. To maintain correspondence between tracks across the noised embeddings and the conditioning inputs, we append a one-hot index encoding of track position to the representation of each track element. During training, track order is randomly permuted so that the model does not rely on any fixed sequence position as a proxy for instrument identity or musical role.

Since our objective is to achieve optional reference style control, and it is highly likely that the content of the dry tracks and the reference song will differ, it is necessary to disentangle the control signals of the audio effects style and musical content during training. To support both automatic and reference-guided mixing, we use classifier-free guidance (CFG) [22] for both style and content conditioning. Inspired by CFG settings for multiple conditioning in [23], during training, the model observes three conditioning states: full conditioning (C, z_{ref}) , content-only conditioning $(C, \emptyset)$, and unconditional input $(\emptyset, \emptyset)$. The latter two states are obtained by randomly masking the reference style embedding z_{ref} or masking both the content and reference embeddings C and z_{ref} . The score estimate is:

$$\begin{aligned} \tilde{s}_\theta = & s_\theta(\tilde{Z}_\sigma, \sigma; \emptyset, \emptyset) \\ & + w_C \left[s_\theta(\tilde{Z}_\sigma, \sigma; C, \emptyset) - s_\theta(\tilde{Z}_\sigma, \sigma; \emptyset, \emptyset) \right] \\ & + w_{ref} \left[s_\theta(\tilde{Z}_\sigma, \sigma; C, z_{ref}) - s_\theta(\tilde{Z}_\sigma, \sigma; C, \emptyset) \right] \end{aligned} \quad (3)$$

where w_C and w_{ref} are the guidance strengths for the two conditioning respectively.

We utilize a pretrained audio effects encoder, FxEncoder++ [24], to extract the content-invariant audio effects features for Z and z_{ref} . Following MEGAMI [12], the content encoder combines CLAP [25] with a wet-to-dry

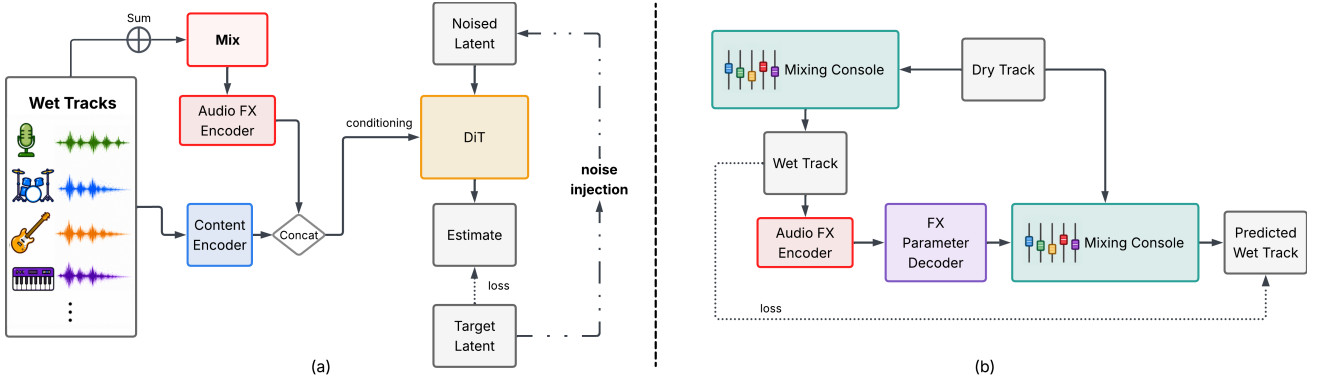


Figure 2. Training pipeline of proposed methods. (a) represents the training of the conditional diffusion model. (b) represents the training of the audio effects parameter decoder.

domain adaptor. The adaptor aligns the CLAP embeddings of dry tracks and their processed versions, enabling effects-invariant content features to be extracted solely from wet audio without requiring real wet-dry pairs.

2.2 Audio Effects Parameter Decoder

At this stage, our purpose is to translate the audio effects embeddings into audio effects parameters for each track and then pass them to the mixing console to carry out the mixing task. To accomplish this, we need to train a model to execute the audio effects parameter estimation task. Previous research [26, 27] regards this task as directly optimizing the parameter loss by applying audio effects with random parameters. However, this is a suboptimal objective since different parameter configurations may yield the same sounds, and it is challenging to generalize to out-of-domain data. DDSP enables a deep learning system to directly optimize the spectral loss [28, 29], and some recent studies [30, 31] model the probability distribution of the parameter configuration instead of directly predicting the parameters. We compare these two methods to build our decoder.

2.2.1 Explicit Parameter Estimation

The first method is to optimize the network directly using spectral loss. We augment the single dry track y_{dry} to the wet track y_{wet} by applying the same audio effects with random parameters in the mixing console as the input to the audio effects encoder. Subsequently, the embedding output from the encoder is passed through a 3-layer Multilayer perceptron (MLP) and then a sigmoid layer to output the audio effects parameters. Next, the differentiable mixing console transforms y_{dry} into the predicted wet track $\hat{y}_{wet}$, and the loss is calculated between y_{wet} and $\hat{y}_{wet}$. We use the stereo version of multi-resolution STFT (MSS) loss [5] and the audio effects embedding loss [12] for training:

$$\mathcal{L}_{\text{spec}} = \mathcal{L}_{\text{MSS}_{l+r}} + \mathcal{L}_{\text{MSS}_{l-r}} + \mathcal{L}_{\text{FXenc}} \quad (4)$$

where $\mathcal{L}_{\text{MSS}_{l+r}}$ represents the loss of the sum of the left and right channel, $\mathcal{L}_{\text{MSS}_{l-r}}$ represents loss of the difference between the two channels, and $\mathcal{L}_{\text{FXenc}}$ is the cosine

distance between the embeddings from the audio encoder. We denote the model using this decoder as Diff2Mix-P.

2.2.2 Modelling Distribution in Parameter Space

Instead of directly predicting the parameters, we extend the idea of a parameter mode distribution modelling method [30] to construct our decoder for the audio effects chain. Given the track-level audio effects embedding z_i , it is first encoded into the mean μ and variance σ of a multivariate Gaussian distribution $p_{\theta}(z_0|z_i)$ via a 2-layer MLP. Then, z_0 is sampled from this distribution and passed through a 2-layer deep sigmoidal flow (DSF) [32] and a sigmoid layer to output the audio effects parameters. Compared to the loss to train Diff2Mix-P, a regularization entropy term is added to balance the estimation accuracy and diversity:

$$\mathcal{L} = \mathcal{L}_{\text{spec}} - \beta \mathcal{H}(p_{\theta}(z_0|z_i)) \quad (5)$$

$$\mathcal{H}(p_{\theta}(z_0|z_i)) = \frac{1}{2} \ln|\det(\sigma I)| + \frac{N}{2} (1 + \ln(2\pi)) \quad (6)$$

where β is a weighting coefficient, which is set to 0.1 in our experiments, and N is the number of audio effects parameters. More derivation details can be found in [30]. During the training stage, z_0 is randomly sampled. At the inference stage for evaluation, we select the most probable point μ of the distribution to mitigate the influence of randomness in the decoder. It should be noted that the mean of the Gaussian distribution does not guarantee the best parameter combination perceptually. We denote the model using this decoder as Diff2Mix-D.

2.2.3 Mixing Console

Based on the design of the parameter decoder, our mixing console has an identical fixed audio effects chain for each track. All audio effect processors are differentiable to enable gradient backpropagation during training, and we adopt the implementation in `dasp-pytorch`². As shown in Figure 3, the audio effects chain is structured as Gain→Parametric Equalizer (EQ)→Dynamic Range Compressor (DRC)→Stereo Panning→Artificial Reverberation (Reverb). Reverb is applied as a send effect; that is, the Reverb

² <https://github.com/csteinmetz1/dasp-pytorch/>

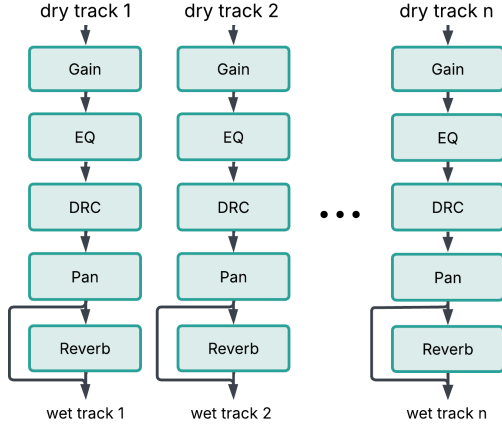


Figure 3. Differentiable Mixing Console

processes a copy of the input signal and mixes it back with the input signal. This allows the network to learn the appropriate amount of reverberation by controlling the added wet signal [4].

3. EXPERIMENTS

3.1 Data

We curated 196 multitracks from MedleyDB [33, 34] and 240 multitracks from MoisesDB [35], of which 90% were allocated for diffusion model training, and the remaining 10% were for objective evaluation. For the training of the parameter estimation model, similar to MEGAMI, we used multiple datasets including MedleyDB [33, 34], OpenSinger [36], IDMT-SMT Drums [37]/Bass [38]/Guitar [39] and GuitarSet [40]. This enables our model to learn how to handle a variety of music materials.

3.2 Baselines

We compared our models against multiple baselines in mixing quality and style controllability.

3.2.1 Mixing Quality Baselines

For automatic mixing quality, we compared against 3 baselines.

- **Equal Loudness:** A perceptual loudness balancing method.
- **FxNorm-automix** [4]: a supervised deep learning baseline trained with effect-normalized out-of-domain data. We evaluated the best-performing *Ours-S-Lb* version of this model.
- **MEGAMI** [12]: a generative automatic mixing framework operating in an effect-embedding space, which maintains the state-of-the-art performance in automatic mixing. We evaluated the public version that is trained on the same datasets as ours.

3.2.2 Style Transfer Baselines

For style controllability, we compared against two baselines.

	AFxRep	FxEncoder	FxEncoder++
Equal Loudness	40.40	45.33	31.15
FxNorm-automix	21.83	33.69	23.99
MEGAMI	18.82	33.18	12.33
Diff2Mix-P	29.21	41.69	15.45
Diff2Mix-D	29.27	30.60	15.83

Table 1. Kernel Audio Distance

	SCE↓	RMS↓	LUFS↓	MSD↓	SW↓	SI↓
Diff-MST	1490.00	5.28	3.18	1.37	3.34	0.21
ST-ITO	1393.15	5.75	5.24	1.28	13.04	0.57
Diff2Mix-P	1400.93	3.90	3.10	1.12	1.79	0.63
Diff2Mix-D	1288.95	4.18	2.22	1.10	1.86	0.61

Table 2. Style Transfer Audio Features Difference

- **Diff-MST** [6]: a reference-guided multitrack mixing style transfer system with a differentiable mixing console. We retrained the best-performing model *AF-16* on MedleyDB and multitrack data from Cambridge Music Technology (Cambridge-mt)³.
- **ST-ITO** [17]: an inference-time optimization method for audio production style transfer over effect parameters. We evaluated the model to control the VST version of the audio effects chain, which includes distortion, EQ, DRC, feedback delay, and Reverb. Note that ST-ITO directly conducts the style transfer on the mixes rather than the multitrack input. Therefore, to use this method, we employed equal-loudness mixes as its input. This might have a strong impact on the output quality since the model is designed for well-produced songs.

3.3 Evaluation Metrics

For evaluating the audio mixing quality, we selected Kernel Audio Distance (KAD) [41] as in MEGAMI. KAD is a distributional metric based on Maximum Mean Discrepancy between embeddings. We calculated KAD between model outputs and human mixes using several models that produce audio effects embeddings: AFxRep [17], FxEncoder [15], and FxEncoder++ [24].

For evaluating the style control ability, we report high-level audio feature loss associated with audio production styles, including dynamics, spatialization and spectral attributes. These features are spectral centroid error (SCE), RMS energy error (RMS), perceptual loudness error (LUFS), mel-spectral distance (MSD), stereo width error (SW) and stereo imbalance error (SI).

4. OBJECTIVE EVALUATION

4.1 Mixing Quality

To evaluate the automatic mixing quality, we inferred our Diff2Mix-P and Diff2Mix-D without the input of a reference song. In practice, we masked the global mix audio ef-

³ <https://cambridge-mt.com/>

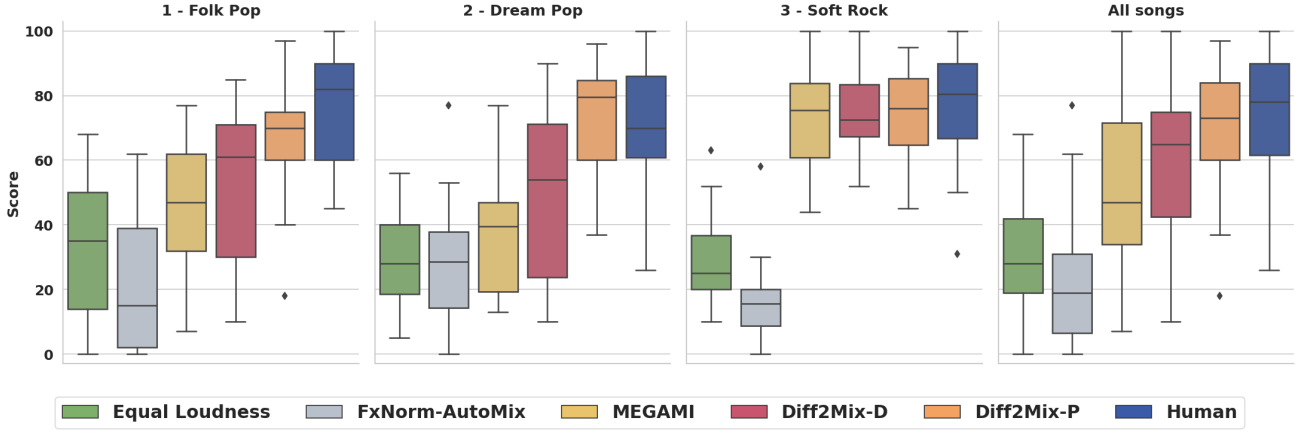


Figure 4. Listening test results of automatic mixing quality rating. The first three plots are ratings corresponding to each multitrack, and the plot on the right side shows the overall ratings.

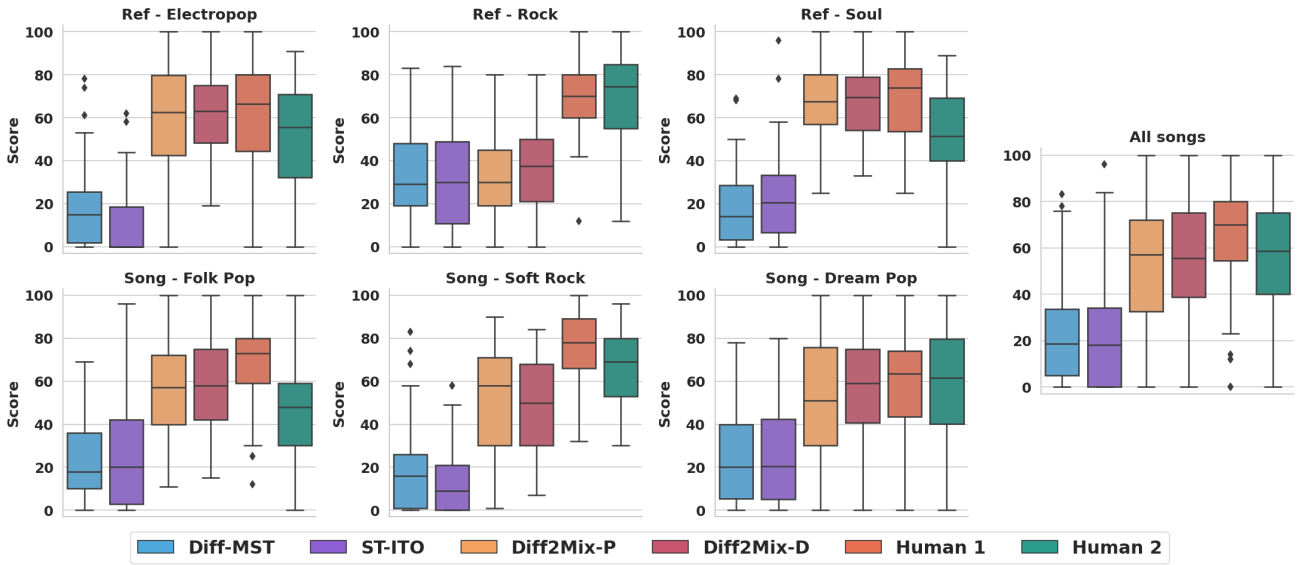


Figure 5. Listening test results of style similarity rating. The three plots in the first row are ratings corresponding to each reference song, and the three plots in the second row are ratings corresponding to each multitrack. The plot on the right side shows the overall ratings.

fects embedding and concatenated it with the content embedding. We compared the KAD scores of our proposed models and three baselines mentioned in Section 3.2.1. Table 1 summarizes the objective mixing quality results measured by KAD, where lower values indicate that the generated mix distribution is closer to the reference distribution in the corresponding embedding space. Note that FxEncoder++ embedding is used to train MEGAMI and our models, so it may not fairly reflect the model performance. Our models and MEGAMI generally exhibit more competitive performance compared to equal loudness and FxNorm-automix. MEGAMI achieves the best performance among AFxRep and FxEncoder++ representations, while Diff2Mix-D achieves the best in FxEncoder. This is consistent with expectations since MEGAMI translates the audio effects embeddings to wet audio through a black-box model, which is free from the restrictions of fixed audio effects and chain order. However, black-box

methods cannot provide the same level of interpretability and control ability as the DDSP method, which significantly limits their application in practical work. Compared to Diff2Mix-D, Diff2Mix-P achieves slightly better results in AFxRep and FxEncoder++, but it performs worse in FxEncoder. We selected the mean of the base distribution to infer Diff2Mix-D, so that potentially sampling better results can be achieved.

4.2 Style Control

To evaluate the production style control ability, we conduct a style transfer task and calculate the audio feature difference metrics listed in Section 3.3 between model outputs and reference songs. We select three songs from Cambridge-mt as our references. Table 2 reports the style transfer feature loss. Overall, our methods show the best performance under these metrics. Diff2Mix-D achieves

Model	Equal Loudness	FxNorm-AutoMix	MEGAMI	Diff2Mix-D	Diff2Mix-P	Human
Equal Loudness	—	< 0.05	< 0.01	< 0.001	< 0.001	< 0.001
FxNorm-AutoMix	< 0.05	—	< 0.01	< 0.001	< 0.001	< 0.001
MEGAMI	< 0.01	< 0.01	—	< 0.05	< 0.01	< 0.01
Diff2Mix-D	< 0.001	< 0.001	< 0.05	—	< 0.05	< 0.01
Diff2Mix-P	< 0.001	< 0.001	< 0.01	< 0.05	—	> 0.05
Human	< 0.001	< 0.001	< 0.01	< 0.01	> 0.05	—

Table 3. Holm-adjusted p-values for pairwise Wilcoxon signed-rank post hoc comparisons for all songs’ subjective automatic mixing quality ratings. Red text indicates the difference between these two methods is not statistically significant.

Model	Diff-MST	ST-ITO	Diff2Mix-P	Diff2Mix-D	Human 1	Human 2
Diff-MST	—	> 0.05	< 0.001	< 0.001	< 0.001	< 0.001
ST-ITO	> 0.05	—	< 0.001	< 0.001	< 0.001	< 0.001
Diff2Mix-P	< 0.001	< 0.001	—	> 0.05	< 0.001	> 0.05
Diff2Mix-D	< 0.001	< 0.001	> 0.05	—	< 0.01	> 0.05
Human 1	< 0.001	< 0.001	< 0.001	< 0.01	—	< 0.05
Human 2	< 0.001	< 0.001	> 0.05	> 0.05	< 0.05	—

Table 4. Holm-adjusted p-values for pairwise Wilcoxon signed-rank post hoc comparisons for all songs’ subjective style similarity ratings. Red text indicates the difference between these two methods is not statistically significant.

the best performance on SCE, LUFs, and MSD, while remaining close to the best model on SW. Diff2Mix-P performs optimally on RMS and SW, potentially indicating that direct parameter estimation can better preserve some energy and spatial-related attributes. In contrast, the baseline methods are only competitive on SI. Overall, both of the proposed variants demonstrate certain advantages, with Diff2Mix-D showing stronger consistency in overall production style, while Diff2Mix-P also has capabilities very close to those of the former.

5. SUBJECTIVE EVALUATION

We conducted a listening test to compare the performance of the models. We selected three multitrack and three reference songs from different musical genres in the Cambridge-mt dataset. Then, we formulated three questions regarding automatic mixing quality for each multitrack song and nine questions about production style similarity for each multitrack-reference pair, resulting in a total of 12 questions. For the mixing quality questions, we compared our models, the models in Section 3.2.1, and professional level human mixes of the multitracks. For the style transfer questions, instead of using our models and the baselines in Section 3.2.2, we recruited two mix engineers to mix the multitracks, with the mixing style being based on the provided reference. We asked the participants to rate the mixing quality and the style similarity to the reference on a scale from 0 to 100. Additionally, we instructed them to rate the best one above 80 and the worst below 20 to fully utilize the scale. The listening test was carried out online using webMUSHRA [42]. In total, we obtained ratings from 20 valid participants who had professional audio mixing experience.

Figure 4 shows per multitrack and overall subjective ratings for automatic mixing quality; the corresponding statistical test results are shown in Table 3. Overall, the proposed systems receive consistently stronger listener preference compared with equal loudness and FxNorm-automix.

Diff2Mix-P reaches the level of human mixing and is likely to be overall better than Diff2Mix-D ($0.01 < p < 0.05$), and Diff2Mix-D is also likely to be better than MEGAMI ($0.01 < p < 0.05$). In the case of per multitrack analysis, Diff2Mix-P, Diff2Mix-D and MEGAMI all perform at the human level of the song in the soft rock genre. However, only Diff2Mix-P maintains this level in the dream pop song, and it slightly outperforms Diff2Mix-D and MEGAMI in the folk pop song. Our objective evaluation suggests MEGAMI is overall better than the proposed methods, which is consistent with our experience in our pilot listening study, where MEGAMI is more stable in some cases. Due to the limitation of listening tests, in which only three multitracks were included, MEGAMI might be underrated because of the low ratings in the dream pop song case.

Figure 5 reports the style similarity ratings in relation to the reference songs, with statistical test results shown in Table 4. Overall, our proposed methods achieve human-level performance and outperform the baselines. Diff2Mix-P, Diff2Mix-D and Human 2 show similar performance, and they all perform slightly worse than Human 1. In terms of each multitrack, the models’ performance is consistent with the overall trend. In terms of each reference, the proposed models still perform consistently with the overall results for the electropop and soul references, but they all fail in the case of the rock song. A possible explanation is that the reference song is represented by a single global style embedding, while some rock-specific mixing characteristics depend on track-specific and inter-track interactions that are more difficult to capture with this shared conditioning signal.

6. CONCLUSION

In this paper, we presented Diff2Mix, a controllable automatic mixing framework that integrates a reference-conditioned diffusion model in the track-level audio effects embedding space with a differentiable mixing console. The framework supports user control at two levels: high-level guidance of the overall production style via a reference song and low-level adjustment through explicit audio effects parameters. To convert generated embeddings into effect controls, we investigated both direct parameter estimation and distribution-based parameter modelling. Objective evaluations demonstrate competitive mixing quality and strong performance on production style related metrics, while subjective listening tests indicate that Diff2Mix produces perceptually convincing mixes and effectively transfers production style from a reference song. These findings highlight the potential of combining diffusion-based, style-conditioned generation with differentiable audio processing for controllable intelligent music production. Future work will extend the framework to more diverse or dynamic effect chains, and introduce more user control, such as a text control interface. A lightweight, low-latency implementation could be another promising research direction for real-time mixing.

7. ACKNOWLEDGMENTS

This work was partly supported by the Yamaha / RAEng Research Chair in Audio Engineering, 2026-2031.

8. REFERENCES

- [1] A. Case, *Mix smart: Pro audio tips for your multitrack mix*. Taylor & Francis, 2012.
- [2] B. De Man, J. Reiss, and R. Stables, “Ten years of automatic mixing,” in *3rd Workshop on Intelligent Music Production (WIMP 2017)*, 2017.
- [3] M. A. Martínez-Ramírez, D. Stoller, and D. Moffat, “A deep learning approach to intelligent drum mixing with the wave-u-net,” *Journal of the Audio Engineering Society*, 2021.
- [4] M. A. M. Ramirez, W. Liao, C. Nagashima, G. Fabbro, S. Uhlich, and Y. Mitsufuji, “Automatic music mixing with deep learning and out- of-domain data,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2022.
- [5] C. J. Steinmetz, J. Pons, S. Pascual, and J. Serrà, “Automatic multitrack mixing with a differentiable mixing console of neural audio effects,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.
- [6] S. S. Vanka, C. J. Steinmetz, J.-B. Rolland, J. D. Reiss, and G. Fazekas, “Diff-mst: Differentiable mixing style transfer,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2024.
- [7] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, “Ddsp: Differentiable digital signal processing,” in *International Conference on Learning Representations*, 2019.
- [8] S. Nercessian, “Neural parametric equalizer matching using differentiable biquads,” in *International Conference on Digital Audio Effects*, 2020, pp. 265–272.
- [9] J. T. Colonel, C. J. Steinmetz, M. Michelen, and J. D. Reiss, “Direct design of biquad filter cascades with deep learning by sampling random polynomials,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- [10] C. J. Steinmetz, V. K. Ithapu, and P. Calamia, “Filtered noise shaping for time domain room impulse response estimation from reverberant speech,” in *WASPAA*, 2021.
- [11] J. T. Colonel and J. Reiss, “Reverse engineering of a recording mix with differentiable digital signal processing,” *The Journal of the Acoustical Society of America*, vol. 150, no. 1, pp. 608–619, 2021.
- [12] E. Moliner, M. A. Martínez-Ramírez, J. Koo, W.-H. Liao, K. W. Cheuk, J. Serrà, V. Välimäki, and Y. Mitsufuji, “Automatic music mixing using a generative model of effect embeddings,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2026.
- [13] X. Wu and A. Horner, “Diffusion models for automatic music mixing,” in *IEEE International Conference on Big Data (BigData)*, 2024, pp. 3242–3247.
- [14] J. Koo, S. Paik, and K. Lee, “End-to-end music remastering system using self-supervised and adversarial training,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.
- [15] J. Koo, M. A. Martínez-Ramírez, W.-H. Liao, S. Uhlich, K. Lee, and Y. Mitsufuji, “Music mixing style transfer: A contrastive learning approach to disentangle audio effects,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.
- [16] C. J. Steinmetz, N. J. Bryan, and J. D. Reiss, “Style transfer of audio effects with differentiable signal processing,” *Journal of the Audio Engineering Society*, vol. 70, no. 9, pp. 708–721, 2022.
- [17] C. J. Steinmetz, S. Singh, M. Comunità, I. Ibnyahya, S. Yuan, E. Benetos, and J. D. Reiss, “St-ito: Controlling audio effects for style transfer with inference-time optimization,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2024.
- [18] A. Chu, P. O’Reilly, J. Barnett, and B. Pardo, “Text2fx: Harnessing clap embeddings for text-guided audio effects,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025, pp. 1–5.
- [19] S. Doh, J. Koo, M. A. Martínez-Ramírez, W. Choi, W.-H. Liao, Q. Wu, J. Nam, and Y. Mitsufuji, “LLM2fx-tools: Tool calling for music post-production,” in *The Fourteenth International Conference on Learning Representations*, 2026.
- [20] T. Karras, M. Aittala, T. Aila, and S. Laine, “Elucidating the design space of diffusion-based generative models,” *Advances in neural information processing systems*, 2022.
- [21] W. Peebles and S. Xie, “Scalable diffusion models with transformers,” in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023.
- [22] J. Ho and T. Salimans, “Classifier-free diffusion guidance,” in *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*.
- [23] T. Brooks, A. Holynski, and A. A. Efros, “Instruct-pix2pix: Learning to follow image editing instructions,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023.

- [24] Y.-T. Yeh, J. Koo, M. Martínez-Ramírez, W.-H. Liao, Y.-H. Yang, and Y. Mitsufuji, “Fx-encoder++: Extracting instrument-wise audio effects representations from mixtures,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2025.
- [25] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, “Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.
- [26] O. Barkan, D. Tsiris, O. Katz, and N. Koenigstein, “Inversynth: Deep estimation of synthesizer parameter configurations from audio signals,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2019.
- [27] O. Barkan and D. Tsiris, “Deep synthesizer parameter estimation,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019.
- [28] N. Masuda and D. Saito, “Synthesizer sound matching with differentiable dsp,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2021.
- [29] Y. Yang, Z. Jin, C. Barnes, and A. Finkelstein, “White box search over audio synthesizer parameters,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2023.
- [30] C. Peladeau, D. Fourer, and G. Peeters, “Audio processor parameters: estimating distributions instead of deterministic values,” in *International Conference on Digital Audio Effects (DAFx)*, 2025.
- [31] B. Hayes, C. Saitis, and G. Fazekas, “Audio synthesizer inversion in symmetric parameter spaces with approximately equivariant flow matching,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2025.
- [32] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville, “Neural autoregressive flows,” in *International conference on machine learning*, 2018, pp. 2078–2087.
- [33] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, “Medleydb: A multitrack dataset for annotation-intensive mir research,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2014.
- [34] R. M. Bittner, J. Wilkins, H. Yip, and J. P. Bello, “Medleydb 2.0: New data and a system for sustainable data collection,” *ISMIR Late Breaking and Demo Papers*, vol. 36, 2016.
- [35] I. Pereira, F. Araújo, F. Korzeniowski, and R. Vogl, “Moisesdb: A dataset for source separation beyond 4-stems,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2023.
- [36] R. Huang, F. Chen, Y. Ren, J. Liu, C. Cui, and Z. Zhao, “Multi-singer: Fast multi-singer singing voice vocoder with a large-scale corpus,” in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3945–3954.
- [37] C. Dittmar and D. Gärtner, “Real-time transcription and separation of drum recordings based on nmf decomposition,” in *International Conference on Digital Audio Effects (DAFx)*, 2014, pp. 187–194.
- [38] J. Abeßer, H. Lukashevich, and G. Schuller, “Feature-based extraction of plucking and expression styles of the electric bass guitar,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010, pp. 2290–2293.
- [39] C. Kehling, J. Abeßer, C. Dittmar, and G. Schuller, “Automatic tablature transcription of electric guitar recordings by estimation of score-and instrument-related parameters,” in *International Conference on Digital Audio Effects (DAFx)*, 2014, pp. 219–226.
- [40] Q. Xi, R. M. Bittner, J. Pauwels, X. Ye, and J. P. Bello, “Guitarset: A dataset for guitar transcription,” in *International Society for Music Information Retrieval Conference (ISMIR)*, 2018.
- [41] Y. Chung, P. Eu, J. Lee, K. Choi, J. Nam, and B. S. Chon, “Kad: No more fad! an effective and efficient evaluation metric for audio generation,” in *ICML 2025 Workshop on Machine Learning for Audio*, 2025.
- [42] M. Schoeffler, S. Bartoschek, F.-R. Stöter, M. Roess, S. Westphal, B. Edler, and J. Herre, “webmushra—a comprehensive framework for web-based listening tests,” *Journal of open research software*, 2018.