

ITO-MASTER: INFERENCE-TIME OPTIMIZATION FOR AUDIO EFFECTS MODELING OF MUSIC MASTERING PROCESSORS

Junghyun Koo¹

Marco A. Martínez-Ramírez¹

Wei-Hsiang Liao¹

Giorgio Fabbro²

Michele Mancusi²

Yuki Mitsufuji^{1,3}

¹ Sony AI, Japan ² Sony Europe B.V., Germany ³ Sony Group Corporation, Japan

{firstname.lastname}@sony.com

ABSTRACT

Music mastering style transfer aims to model and apply the mastering characteristics of a reference track to a target track, simulating the professional mastering process. However, existing methods apply fixed processing based on a reference track, limiting users' ability to fine-tune the results to match their artistic intent. In this paper, we introduce the ITO-Master framework, a reference-based mastering style transfer system that integrates Inference-Time Optimization (ITO) to enable finer user control over the mastering process. By optimizing the reference embedding z_{ref} during inference, our approach allows users to refine the output dynamically, making micro-level adjustments to achieve more precise mastering results. We explore both black-box and white-box methods for modeling mastering processors and demonstrate that ITO improves mastering performance across different styles. Through objective evaluation, subjective listening tests, and qualitative analysis using text-based conditioning with CLAP embeddings, we validate that ITO enhances mastering style similarity while offering increased adaptability. Our framework provides an effective and user-controllable solution for mastering style transfer, allowing users to refine their results beyond the initial style transfer.

1. INTRODUCTION

Music mastering is the final step in the audio production process, ensuring professional sound quality and consistent playback across music distribution platforms. This process involves applying a series of audio effects such as equalization, compression, stereo imaging, and limiting, which collectively shape the sonic characteristics and enhance the overall quality of the audio [1, 2]. Traditionally, mastering has required skilled engineers who carefully adjust these effects based on the track's content and desired artistic outcome. However, with the increasing volume of music production and the demand for consistency across stream-

ing platforms, the need for automated mastering solutions has grown substantially.

In response to this demand, various automatic mastering systems have emerged [3–5]. However, these systems operate in an unconditioned manner, applying audio effects without direct user control. To introduce adaptability, reference-based approaches have been explored, where the processing characteristics of a reference track are applied to another [6, 7]. These methods aim to match audio features such as dynamics, tonal balance, and stereo width, offering an alternative to fully automatic mastering. However, significant challenges remain in achieving both high-quality results and controllability.

Existing reference-based approaches can be broadly categorized into *black-box* and *white-box* models. Black-box models, often based on end-to-end neural networks [6], can effectively capture high-level audio patterns but lack transparency and interpretability, making it difficult for users to modify specific aspects of the processing. In contrast, white-box models leverage either feature matching algorithms [7] or differentiable audio processors [5] to provide greater control over individual parameters. While white-box methods offer a structured and interpretable approach [8], they are often constrained by the simplicity of their differentiable processors, which may not fully replicate the complex tools used in professional mastering.

In this paper, we introduce the ITO-Master framework, a reference-based audio effects modeling of music mastering processors that incorporates Inference-Time Optimization (ITO) for finer user control. While previous style transfer methods apply fixed processing based on a reference, our approach allows users to dynamically refine the output when the initial result does not fully align with their preferences. By optimizing the reference embedding z_{ref} during inference, ITO-Master enables micro-level adjustments, allowing for more precise and targeted mastering refinements.

The key contributions of this work include: (1) **ITO-Master framework:** A novel approach for reference-based audio effects modeling of mastering processors using ITO. (2) **Comparison of black-box and white-box models:** A systematic study of two paradigms for evaluating their effectiveness and trade-offs. (3) **Realistic mastering processor chain:** Implementation of a structured differentiable mastering pipeline to enhance the realism of white-box processing. (4) **Comprehensive evaluation:** Performance validation



© J. Koo, M. Martínez-Ramírez, W-H. Liao, G. Fabbro, M. Mancusi, and Y. Mitsufuji. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** J. Koo, M. Martínez-Ramírez, W-H. Liao, G. Fabbro, M. Mancusi, and Y. Mitsufuji, "ITO-Master: Inference-Time Optimization for Audio Effects Modeling of Music Mastering Processors", in *Proc. of the 26th Int. Society for Music Information Retrieval Conf.*, Daejeon, South Korea, 2025.

via objective metrics, listening tests, and qualitative analysis using text-based conditioning with CLAP embeddings.

2. RELATED WORKS

2.1 Audio Effects Style Transfer

Audio effects style transfer has become a significant area of research in automating and enhancing music production. Recent advancements in deep learning have led to more sophisticated approaches, where neural networks are used to learn complex mappings between input and output audio signals. These methods have been applied to style transfer across single audio effects (Fx) or multiple sets of Fx (Fx chain) [5, 6, 9–14], effectively modeling temporal dependencies and applying style transfer based on a reference track at the waveform level. While these methods have shown success in controlled environments, challenges remain in extending their application to diverse real-world scenarios, particularly in adapting to different mastering styles and varying input signals.

2.2 Inference-Time Optimization

Recently, [15, 16] have explored inference-time optimization (ITO) in music generation tasks, where the initial latent embedding is optimized by backpropagating through diffusion-based models with the loss between a generated sample and a reference track. In the context of audio effects style transfer, ITO has been applied in methods like ST-ITO [17, 18], where interpretable parameters in a white-box differentiable Fx chain are optimized.

Our work focuses specifically on mastering style transfer, where handling heavily compressed audio—common in commercially released music—requires particular attention to limiters and dynamic range management. The ITO-Master framework introduces ITO on the reference embedding z_{ref} , allowing optimization at inference time in both black-box and white-box models. By fine-tuning z_{ref} , our approach adapts the mastering style of a reference track without retraining the entire model. ITO-Master ensures smooth adaptation to the reference track’s characteristics while preserving the ability to observe and adjust the underlying parameters in the white-box model. This makes our framework particularly suitable for mastering tasks, providing professionals and amateurs with precise control over the audio mastering process.

3. METHODOLOGY

In this section, we describe the components of the proposed mastering style transfer framework: the training pipeline, Mastering Style Converter, differentiable mastering chain, and ITO. The training pipeline simulates real-world mastering scenarios by applying random Fx manipulations. The Mastering Style Converter Ψ transfers the mastering style from a reference track to a target track and can be implemented using both black-box and white-box approaches. The differentiable mastering chain serves as a white-box processor that models various Fx in a structured sequence.

Lastly, the ITO process optimizes z_{ref} at inference time to enhance style transfer performance. The following subsections describe each component in detail.

3.1 Training Pipeline of Mastering Style Transfer

As shown in Figure 1(a), the training pipeline follows established methodologies in style transfer, utilizing a self-supervised training framework with random Fx manipulation [6, 11–13]. Based on the understanding that a single song maintains a consistent mastering style throughout [2], a song is first segmented into two parts, A and B . For the input to Ψ , random manipulation f_1 is applied to simulate a random style, similar to how the process would function in an application setting. Then, we apply Fx-Normalization [19] f_{norm} , which normalizes certain Fx characteristics to fixed target levels, to facilitate the performance of style transfer. f_{norm} is only applied to the equalizer (EQ), stereo imager, and loudness levels, allowing the model to capture a broader range of nonlinear FX transformations, such as compression levels and distortion. While normalizing compression is technically feasible, it is not well-suited for the on-the-fly training procedure. For distortion, normalization would require either removing all distortion from the given song or applying a consistent distortion level across all tracks, which is outside the scope of this paper. In summary, the input to Ψ is defined as $x_{\text{in}} = f_1(f_{\text{norm}}(A))$.

To achieve style transfer, the Fx information from the reference track x_{ref} is encoded to create the reference embedding that conditions Ψ . A second random manipulation f_2 is applied to segment B , which is then encoded by the reference encoder Φ , resulting in the reference embedding $z_{\text{ref}} = \Phi(f_2(B))$. The training process minimizes the loss between the model output $y' = \Psi(x_{\text{in}}, z_{\text{ref}})$ and the target signal $y = f_2(A)$. Since both A and B originate from the same mastered song, we assume that y and x_{ref} share the same mastering style.

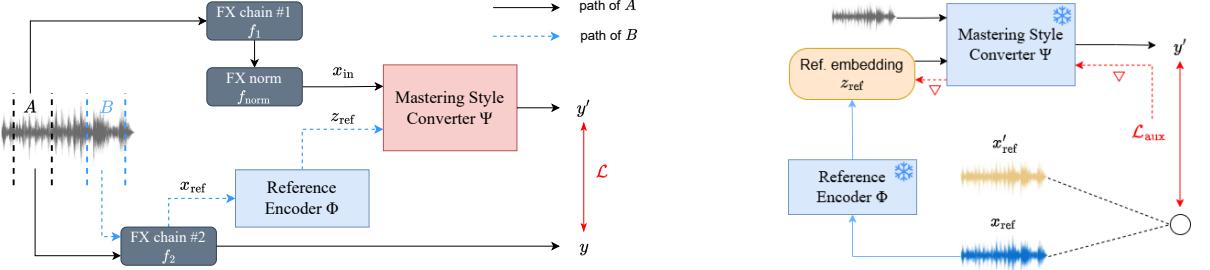
3.2 Mastering Style Converter

Ψ can be implemented using two distinct modeling approaches: black-box modeling and white-box modeling. In the black-box approach, Ψ directly models the waveform signal y' . Conversely, the white-box approach estimates the parameters Θ for the differentiable mastering Fx chain. The formulation of the differentiable mastering Fx chain in the white-box model is identical to that used in the Fx manipulator f for processing randomly mastered audio.

The training objective for Ψ is the multi-scale spectral loss \mathcal{L}_{MSS} , applied to both left-right and mid-side channels (where mid = left + right, side = left - right), as used in [12].

3.3 Differentiable Mastering Chain Modeling

The mastering chain is designed to be fully differentiable white-box processor, serving a dual purpose: functioning both as a mastering style converter and as a random mastering manipulation module for training the converter. By modeling a wide range of variability in mastering styles, the chain enables the system to robustly handle and replicate the



(a) Training pipeline of Mastering Style Converter Ψ . During this phase, Reference Encoder Φ is trained using diverse mastering styles generated by random FX manipulation f . The target signal y is synthesized by applying the same manipulation to segment A as to reference segment B , both from the same song.

(b) ITO is performed using an auxiliary (content-independent) objective function, allowing any reference music. Users can optimize z_{ref} based on their preferences for both the reference and objective function.

Figure 1: Overall pipeline of ITO-Master.

complexities of real-world music mastering. To simulate a realistic music mastering process [3], the chain includes six distinct Fx modules: 1. *6-band parametric equalizer*, 2. *distortion*, 3. *3-band compressor*, 4. *makeup gain*, 5. *stereo imager*, and 6. *limiter*. The order of these modules is fixed, with the probability of applying each Fx module for random manipulation during training set at 90%, 30%, 80%, 85%, 60%, and 100%, respectively. These probabilities are adopted to introduce greater variability while preventing the synthesis of overly unrealistic mastering styles to enhance Ψ 's modeling capability. The chain comprises a total of 46 controllable parameters.

To ensure differentiability, the Fx modules are implemented using open-source libraries^{1, 2} that support gradient-based optimization techniques. For modeling the 3-band multiband compression, a fourth-order Linkwitz-Riley crossover filter [20] is first applied to split the signal into three bands, followed by a differentiable all-pole filter. A key component in the chain is the use of these differentiable all-pole filters, as modeled by [21], which enable the computation of both compression and expansion effects in both the multiband compressor and limiter. This capability is crucial for practical applications, allowing the system to manage both limiters and delimiters, which is particularly important in real-world scenarios where most commercially released music is heavily compressed [2], requiring effective style transfer under such conditions.

3.4 Inference-Time Optimization on Reference Embedding

The primary contribution of this work is the introduction of ITO on z_{ref} . Instead of fine-tuning the entire model Ψ , the focus is on optimizing only z_{ref} while keeping the pre-trained Ψ fixed, as shown in Figure 1(b). Although optimizing z_{ref} during inference time in the Black-box model does not provide interpretability in terms of Fx processors, the White-box model preserves interpretability. In fact, the changes in the parameters Θ before and after the ITO process can be observed, providing insight into how Θ is adjusted. Additionally, users can optimize the system with

an alternative reference signal x'_{ref} , offering a different approach from conventional mastering style transfer, as this method merges mastering styles based on the combination of the new reference signal and the optimizing objective function. The advantages of ITO on z_{ref} include a significantly reduced number of optimization steps compared to optimizing the entire differentiable chain's Θ from scratch, as we compare in Section 5.

For ITO, the Audio Feature (AF) loss proposed by [13] is utilized as the auxiliary objective function \mathcal{L}_{aux} . The AF loss is a content-independent loss that combines various audio feature transformations, capturing the dynamics, spatialization, and spectral characteristics of the audio. Each transformation in the AF loss has its own predefined weighting factor, and these weighted transformations are summed together to compute the overall loss. In our experiments, the original weights from the reference paper are followed. We optimize z_{ref} iteratively using gradient descent: $z_{\text{ref}}^{(t+1)} = z_{\text{ref}}^{(t)} - \eta \nabla_z \mathcal{L}_{\text{aux}}(\Psi(x_{\text{in}}, z_{\text{ref}}^{(t)}), x_{\text{ref}})$, where η is the learning rate. For objective and subjective evaluation, we focus solely on using AF loss as the optimization objective for ITO. Since ITO can be optimized with any loss function, we also qualitatively explore optimization using a text prompt with CLAP embeddings [22] in Section 5.3.

4. EXPERIMENTS

4.1 Dataset

We utilized MoisesDB dataset [23] for training, and validated using the MUSDB18 validation subset [24]. Mixture samples from these datasets are employed, as they are not fully mastered, allowing random Fx manipulation with f to create synthetic mastered samples. For Fx-Normalization, mean statistics are precomputed on the MoisesDB dataset, and normalization is applied to match the EQ, stereo imager, and loudness levels. For evaluation, 200 songs are randomly selected from the MTG-Jamendo dataset [25]. Of these, 100 songs are used as x_{in} , and the remaining 100 serve as x_{ref} for input into Ψ . During training and validation, both segments A and B are 11.8 seconds long. For evaluation, 30-second samples are used since the fully convolutional architecture of Ψ can handle variable-length inputs.

¹ <https://github.com/csteinmetz1/dasp-pytorch>

² <https://github.com/DiffAPF/torchcomp>

4.2 Experimental Setup

The experimental setup includes two primary training configurations for Ψ : Black-box and White-box methods. Both configurations use the Temporal Convolutional Network (TCN) [26] architecture for processing x_{in} , with 10.5 million trainable parameters. Pre-trained weights of the *FXencoder* [12] is adopted as Φ and tested under two conditions: with and without Φ being trained alongside Ψ . All models are trained for 72,000 iterations with a batch size of 4.

In addition to training the Φ and Ψ models, ITO is performed on each test data to optimize the reference embedding z_{ref} , which has a dimensionality of 2048. The ITO process is run for a maximum of 100 steps or is stopped earlier if the loss value increases, indicating convergence. For comparison, an alternative ITO approach is also applied where optimization is performed solely on the parameters Θ of the differentiable mastering chain f . In this case, Θ is optimized for up to 2K steps to evaluate its effectiveness relative to the proposed ITO method focused on z_{ref} . All methods are optimized using the RAdam optimizer [27] with a learning rate of $2 \cdot 10^{-4}$. More details are available in our open-source repository³.

4.3 Evaluation Metrics

To objectively evaluate mastering style transfer, content-independent objectives are utilized, given that different content is being compared. The following metrics are employed:

- **Audio Feature (AF) Loss:** As discussed in Section 3.4, AF Loss measures how well the output y' matches the desired audio features.
- **Dynamic Range Variability (DRV):** DRV is a crucial metric for assessing the compression level of audio, particularly in the context of music mastering, where limiters play a significant role. The DRV metric is computed by first identifying peak values in the audio signal using a high-frequency content onset detection function. DRV is the standard deviation of these peak values after filtering out the lowest 25% of the values, which is defined as:

$$\text{DRV} = \frac{1}{C} \sum_{c=1}^C \text{std}(\{p_i^c : p_i^c > \text{percentile}(p^c, 75)\}) \quad (1)$$

where p^c denotes the set of peak values $\{p_1^c, p_2^c, \dots\}$ in channel c , and C is the total number of channels. The metric reflects the variability in dynamic range, with higher values indicating less consistent compression.

- **Fx Embedding Similarity (cos sim):** Cosine similarity measures the similarity between the reference embedding $\Phi(x_{\text{ref}})$ and the output embedding $\Phi(y')$. We adopt the pretrained FXencoder as Φ . This metric evaluates how closely the output matches the reference in terms of its learned Fx characteristics.
- **Fréchet Audio Distance (FAD):** FAD [28] assesses the perceptual quality of the generated audio by comparing the statistical distribution of the model outputs to

a reference distribution. FAD is calculated using three deep audio embeddings: CLAP [29], DAC [30], and EnCodec [31]. We chose these embeddings as they have shown strong correlation with human preference in acoustic quality, with codec-based models like DAC and EnCodec being particularly sensitive to acoustic effects, as demonstrated by [32]. The metric computes the distance between the distribution of features extracted from the model’s output and those extracted from a subset of the Jamendo dataset, measuring how natural the style-transferred audio sounds compared to real recordings.

4.4 Baseline Methods

The following baseline methods, representing existing mastering style transfer systems, are used for comparison:

- **Acoustic Feature Matching Approaches:** **Feature matching** approaches aim to adjust the Fx of an input track to match those of a reference track by directly aligning specific audio features.
- **Fx-Normalization** [19]: Instead of normalizing the given audio to the mean statistics of the target data distribution, this approach directly matches the Fx levels to those of the reference song. The official implementation⁴ is used to match the audio effects in the order of EQ, compression, stereo imaging, and loudness, respectively.
- **Matchering** [7]: An open-source library that matches the given song’s RMS, frequency response, peak amplitude, and stereo width to those of the reference track. The official implementation⁵ is used to infer the processed songs.
- **E2E Remastering** [6]: This end-to-end remastering approach is a black-box model that directly predicts the signal y' at the waveform level. The model is trained in a self-supervised manner using a large dataset of released pop songs. It leverages a pre-trained encoder and a projection discriminator to encourage the generation of realistic audio that accurately reflects the mastering style of the reference track.

5. RESULTS

5.1 Objective Evaluation

The performance of the proposed methods, along with the baseline approaches, is summarized in Table 1. The feature matching methods, specifically Fx-Normalization and Matchering, demonstrate strong performance on the AF and FAD metrics. This is expected, as these approaches directly apply Fx-related transformations to match the reference track’s characteristics. However, these methods perform poorly on the DRV metric, as they lack the control needed for proper dynamic range adjustments, which is crucial in real-world mastering tasks involving delimiting. E2E Remastering [6] shows good performance on the FAD with

⁴ <https://github.com/sony/FxNorm-automix>

⁵ <https://github.com/sergree/matchering>

³ <https://github.com/SonyResearch/ITO-Master>

Method		AF (\downarrow)	DRV (\downarrow)	cos sim (\uparrow)	FAD _{CLAP} (\downarrow)	FAD _{DAC} (\downarrow)	FAD _{EnCodec} (\downarrow)
Feature Matching	Fx-Normalization [19]	0.157	0.801	0.941	161.4	177.4	84.53
	Matcher [7]	0.160	0.823	0.942	110.8	126.1	59.34
Baseline	E2E Remastering [6]	0.288	0.858	0.942	104.3	176.7	37.19
Proposed	Black-box	0.346	0.685	0.944	160.8	378.4	51.12
	+ train Φ	0.125	0.577	0.945	159.8	177.4	46.94
	+ ITO on z_{ref}	0.099	0.567	0.946	182.2	180.5	42.82
	White-box	0.253	0.598	0.946	93.7	144.8	36.22
	+ train Φ	0.186	0.521	0.945	93.2	101.4	38.90
	+ ITO on z_{ref}	0.139	0.474	0.946	105.2	109.1	42.99
ITO on Θ		0.250	0.609	0.927	216.8	294.8	101.60

Table 1: Mastering Style Transfer on Jamendo dataset (real-world scenario).

CLAP and EnCodec embeddings, likely due to its use of an adversarial objective during training, which aids in generating realistic audio that closely matches the reference distribution. However, this system falls short on AF and DRV, indicating challenges in capturing precise audio feature transformations and managing dynamic range.

Among the proposed methods, Black-box approaches outperform White-box methods in terms of AF, indicating its effectiveness in capturing audio feature transformations from direct modeling of y' with \mathcal{L}_{MSS} . However, the White-box method shows better results across all FAD metrics, suggesting that it produces audio more aligned with real-world distributions. This may imply that while Black-box models capture more detailed transformations, White-box approaches produce outputs that are more perceptually consistent with real-world mastering styles.

When Ψ is trained while keeping the pre-trained FXencoder Φ fixed, the performance is generally inferior. This is likely because the FXencoder was trained on a different set of Fx chains and may not fully capture the manipulations applied to the reference songs in this context. However, when Φ is trained alongside Ψ , there is a significant improvement in performance, as this joint training allows the encoder to better adapt to the specific Fx manipulations used, leading to more accurate mastering style transfer.

Applying ITO on z_{ref} enhances AF performance, but introduces a trade-off in FAD scores, indicating while ITO can refine mastering style transfer, the number of optimization steps must be carefully calibrated to balance competing objectives. Conversely, applying ITO directly on Θ yields poor results across all metrics, even with a larger number of optimization steps. Interestingly, in reverse engineering tasks—where the input and output content are identical—optimizing Θ works well despite the complexity of the Fx chain [33–35]. However, in mastering style transfer, content-independent loss functions are used to replicate only the mastering style from the reference track. This distinction highlights why ITO on Θ is less effective in this context. Since Ψ is trained with a content-dependent objective, it leverages content information to enhance mastering style transfer. In contrast, \mathcal{L}_{aux} used in ITO fails to capture the intricacies of the task, making it unsuitable for optimizing the entire differentiable chain in this scenario.

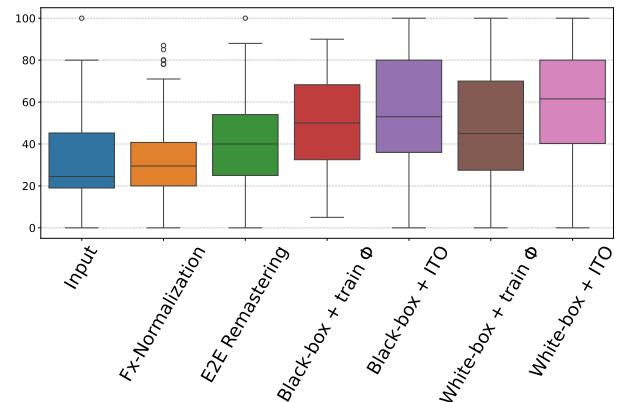


Figure 2: Subjective evaluation results.

Audio samples are available on our demo page⁶.

5.2 Subjective Evaluation

To further validate our proposed methods subjectively, we conducted a MUSHRA-type listening test with 10 participants, all familiar with music post-production and digital effects, having 2 to 5 years of experience in recording, mixing, or mastering. Participants rated various processed tracks based on their similarity in mastering audio effects to a reference track. The evaluation included 8 questions, with 30 seconds long music recordings for all stimuli. The reference audio contained different content from the stimuli, but we ensured the reference and stimuli were not too dissimilar in terms of genre or instrumentation. As a low anchor, the initial music track before being inputted into style transfer systems was presented. There was no high anchor, as the evaluation setup aimed to mimic real-world Fx style transfer using music tracks from the Jamendo dataset.

As illustrated in Figure 2, the subjective test results align with the trends observed in our objective evaluations. All our proposed methods surpass the baselines, with results further enhanced by ITO, showing audio effects characteristics more similar to those of the reference. The similarity scores for the proposed system ranged from 0 to 100, indicating that the listening test was highly challenging, even for experts with domain knowledge. Nevertheless, the proposed

⁶ <https://tinyurl.com/ITO-Master>

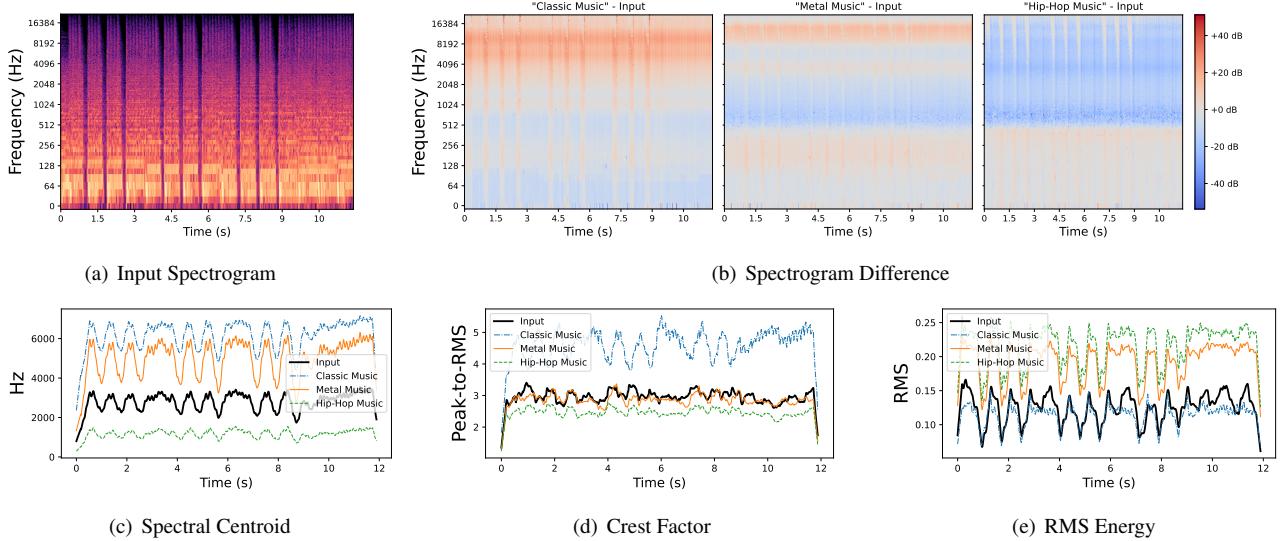


Figure 3: Comparison of different audio features between the input music and ITO-processed tracks using text prompts “Classic Music”, “Metal Music”, and “Hip-Hop Music”.

systems consistently outperformed the baseline, showing significant improvements (pairwise t-test, $p < 0.05$).

5.3 Qualitative Analysis of ITO with Text Prompts

To evaluate the effectiveness of ITO under different \mathcal{L}_{aux} , we perform a qualitative analysis using text prompts with CLAP embeddings [22], similar to the application demonstrated in [36]. Given an input music track, we optimize z_{ref} of the proposed white-box model using text-based conditioning, leveraging the CLAP embedding cosine similarity as the optimization objective. Specifically, we compute the audio embedding CLAP_{aud} of the steered output and the text embedding CLAP_{txt} of the given reference text prompt, then maximize their cosine similarity to guide the transformation. The input music piece used for this analysis is an 11.8-second-long instrumental rock track. Since the input content remains unchanged across different steered results, we can directly assess the influence of each text prompt on various musical features. We explore ITO with three different prompts: “Classic Music”, “Metal Music”, and “Hip-Hop Music” for analysis. This experimental setup can be explored through our interactive demo ⁷.

As shown in Figure 3, the optimized results exhibit distinct characteristics that align with general expectations for each genre. The spectrogram difference plots in 3(b) (steered output - input) highlight the frequency ranges most affected by ITO. Specifically, the “Classic Music” prompt exhibits notable changes in the mid and high frequencies, aligning with the characteristic brightness and clarity often associated with classical recordings. The “Metal Music” prompt shows differences in both low and high frequencies, reflecting the genre’s typical emphasis on powerful bass and sharp treble for aggressive instrumentation. In contrast, the “Hip-Hop Music” prompt predominantly affects the low-frequency range, reinforcing the genre’s signature

emphasis on deep bass and sub-bass elements, which are essential for driving rhythm-heavy beats.

These observations are further supported by the spectral centroid, crest factor, and RMS energy analyses. The spectral centroid results follow an expected trend, where *hip-hop* has the lowest centroid due to its bass-heavy nature, followed by *metal*, while *classic music* has the highest centroid, reflecting its emphasis on harmonic richness and treble clarity. The crest factor, representing peak-to-RMS ratio, is lowest for *hip-hop*, indicating a more compressed and bass-heavy dynamic structure, while *classic music* has the highest crest factor, aligning with its typically uncompressed, wider dynamic range. RMS energy shows an increasing trend from *classic* to *hip-hop*, with *metal* falling in between, which is consistent with the respective loudness and dynamic characteristics of these genres. These findings suggest that ITO, guided by CLAP_{txt} , successfully steers the mastering Fx chain to align with the expected sonic characteristics of the given text prompt, demonstrating its potential as a creative tool for music post-production.

6. CONCLUSION

In this paper, we introduced the ITO-Master framework, which leverages ITO on z_{ref} for mastering style transfer. Our experiments showed that training the reference encoder Φ alongside Ψ improves performance. Optimizing z_{ref} with ITO led to meaningful improvements with few steps, outperforming direct optimization of Θ in efficiency. Subjective evaluations confirmed that our method produces perceptually aligned mastering effects, and qualitative results highlighted the potential of text-conditioned ITO for creative applications. As future work, we plan to incorporate production quality and usability into the evaluation, alongside reference alignment. Since mastering is a curatorial task, poor reference choices can lead to suboptimal results despite high alignment, highlighting the need for perceptual preference metrics.

⁷ <https://huggingface.co/spaces/jhtonyKoo/ITO-Master>

7. REFERENCES

- [1] U. Zölzer, X. Amatriain, D. Arfib, J. Bonada, G. De Poli, P. Dutilleux, G. Evangelista, F. Keiler, A. Loscos, D. Rocchesso *et al.*, *DAFX-Digital audio effects*. John Wiley & Sons, 2002.
- [2] M. Shelvock, *Audio mastering as musical practice*. The University of Western Ontario (Canada), 2012.
- [3] M. Piotrowska, S. Piotrowski, and B. Kostek, “A study on audio signal processed by” instant mastering” services,” in *Audio Engineering Society Convention 142*. Audio Engineering Society, 2017.
- [4] J. Sterne and E. Razlogova, “Machine learning in context, or learning from landr: Artificial intelligence and the platformization of music mastering,” *Social Media+ Society*, vol. 5, no. 2, p. 2056305119847525, 2019.
- [5] M. A. M. Ramírez, O. Wang, P. Smaragdis, and N. J. Bryan, “Differentiable signal processing with black-box audio effects,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 66–70.
- [6] J. Koo, S. Paik, and K. Lee, “End-to-end music remastering system using self-supervised and adversarial training,” in *Proc. ICASSP*, 2022, pp. 4608–4612.
- [7] S. Grishakov, C.-Y. Yu, and Zicklag, “Matchering: Audio matching and mastering python library,” <https://github.com/sergree/matchering>.
- [8] J. Engel, C. Gu, A. Roberts *et al.*, “DDSP: Differentiable digital signal processing,” in *International Conference on Learning Representations*, 2020.
- [9] J. Koo, S. Paik, and K. Lee, “Reverb conversion of mixed vocal tracks using an end-to-end convolutional deep neural network,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 81–85.
- [10] S. Lee, J. Park, S. Paik, and K. Lee, “Blind estimation of audio processing graph,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [11] C. J. Steinmetz, N. J. Bryan, and J. D. Reiss, “Style transfer of audio effects with differentiable signal processing,” *J. Audio Eng. Soc*, vol. 70, no. 9, pp. 708–721, 2022.
- [12] 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 *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [13] S. S. Vanka, C. Steinmetz, J.-B. Rolland, J. Reiss, and G. Fazekas, “Diff-MST: Differentiable mixing style transfer,” in *Proc. ISMIR*, 2024.
- [14] Y.-H. Chen, Y.-T. Yeh, Y.-C. Cheng, J.-T. Wu, Y.-H. Ho, J.-S. R. Jang, and Y.-H. Yang, “Towards zero-shot amplifier modeling: One-to-many amplifier modeling via tone embedding control,” in *Proc. ISMIR*, 2024.
- [15] Z. Novack, J. McAuley, T. Berg-Kirkpatrick, and N. J. Bryan, “DITTO: Diffusion inference-time t-optimization for music generation,” in *Proc. ICML*, 2024.
- [16] Z. Novack, J. McAuley, T. Berg-Kirkpatrick, and N. Bryan, “DITTO-2: Distilled diffusion inference-time t-optimization for music generation,” in *Proc. ISMIR*, 2024.
- [17] C. Steinmetz, S. Singh, I. Ibnyahya, S. Yuan, E. Benetos, J. Reiss *et al.*, “ST-ITO: Controlling audio effects for style transfer with inference-time optimization,” in *Proc. ISMIR*, 2024.
- [18] C.-Y. Yu, M. A. Martínez-Ramírez, J. Koo, W.-H. Liao, Y. Mitsufuji, and G. Fazekas, “Improving inference-time optimisation for vocal effects style transfer with a gaussian prior,” *arXiv preprint arXiv:2505.11315*, 2025.
- [19] M. A. Martínez-Ramírez, W.-H. Liao, G. Fabbro, S. Uhlich, C. Nagashima, and Y. Mitsufuji, “Automatic music mixing with deep learning and out-of-domain data,” in *Proc. ISMIR*, 2022.
- [20] S. H. Linkwitz, “Active crossover networks for non-coincident drivers,” *Journal of the Audio Engineering Society*, vol. 24, no. 1, pp. 2–8, 1976.
- [21] C.-y. Yu, C. Mitcheltree, A. Carson, S. Bilbao, J. Reiss, and G. Fazekas, “Differentiable all-pole filters for time-varying audio systems,” in *27th International Conference on Digital Audio Effects (DAFx)*, 2024.
- [22] 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 *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [23] I. Pereira, F. Araújo, F. Korzeniowski, and R. Vogl, “MoisesDB: A dataset for source separation beyond 4-stems,” in *Proc. ISMIR*, 2023.
- [24] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimalakis, and R. Bittner, “MUSDB18-HQ - an uncompressed version of MUSDB18,” Aug. 2019. [Online]. Available: <https://doi.org/10.5281/zenodo.3338373>
- [25] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, “The mtg-jamendo dataset for automatic music tagging,” in *Proc. ICML*, 2019.

- [26] C. J. Steinmetz and J. D. Reiss, “Efficient neural networks for real-time modeling of analog dynamic range compression,” *arXiv preprint arXiv:2102.06200*, 2021.
- [27] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, “On the variance of the adaptive learning rate and beyond,” in *International Conference on Learning Representations*, 2020.
- [28] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, “Fréchet audio distance: A metric for evaluating music enhancement algorithms,” *arXiv preprint arXiv:1812.08466*, 2018.
- [29] B. Elizalde, S. Deshmukh, M. Al Ismail, and H. Wang, “CLAP: Learning audio concepts from natural language supervision,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [30] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, “High-fidelity audio compression with improved rvqgan,” *Advances in Neural Information Processing Systems*, vol. 36, 2024.
- [31] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, “High fidelity neural audio compression,” *Transactions on Machine Learning Research*, 2023.
- [32] A. Gui, H. Gamper, S. Braun, and D. Emmanouilidou, “Adapting frechet audio distance for generative music evaluation,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 1331–1335.
- [33] S. Lee, M. A. Martínez-Ramírez, W.-H. Liao, S. Uhlich, G. Fabbro, K. Lee, and Y. Mitsufuji, “Searching for music mixing graphs: A pruning approach,” in *27th International Conference on Digital Audio Effects (DAFx)*, 2024.
- [34] ———, “Reverse engineering of music mixing graphs with differentiable processors and iterative pruning,” *Journal of the Audio Engineering Society*, vol. 73, pp. 344–365, June 2025.
- [35] C.-Y. Yu, M. A. Martínez-Ramírez, J. Koo, B. Hayes, W.-H. Liao, G. Fazekas, and Y. Mitsufuji, “Diffvox: A differentiable model for capturing and analysing professional effects distributions,” in *28th International Conference on Digital Audio Effects (DAFx)*, 2025.
- [36] A. Chu, P. O’Reilly, J. Barnett, and B. Pardo, “Text2FX: Harnessing clap embeddings for text-guided audio effects,” in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.