INSTRUCTFX2FX: A MULTI-TURN TEXT-TO-EFFECT SYSTEM FOR SEQUENTIAL AUDIO EFFECT REFINEMENT

Song-Ze Yu, Milan Liessens Dujardin, Yuxuan Cai, Wantong Zhang, Brian Cruz, Jeremy Wagner and Carmine-Emanuele Cella**

Center for New Music and Audio Technologies (CNMAT)
University of California, Berkeley
<vaclis, milan.ld, yuxuancai1106, wtzhang, brian.cruz, jeremy.wagner, carmine.cella>@berkeley.edu
* Corresponding authors

ABSTRACT

We present *InstructFX2FX*, a system for sequential audio effect refinement through multi-turn natural-language instructions. Existing text-to-effect systems are largely single-shot, mapping one textual descriptor to one preset. Real audio engineering is instead sequential: engineers refine an existing effect chain through successive instructions. This poses a stateful problem that single-shot systems do not address: given the current effect parameters state and a new instruction, update the sound while preserving what earlier instructions already achieved. InstructFX2FX addresses this with a hybrid architecture that divides labor between a language model and CLAP-guided optimization. The LLM serves as a high-level planner that selects effects and proposes the initial parameter state, motivated by recent evidence that LLMs can outperform CLAP-based optimization for single-turn text-to-effect mapping [1]; CLAP-guided optimization then refines the existing parameter state, providing a more stable and robust refinement mechanism than LLM reprompting. In the demo, attendees drive a dry recording through successive natural-language instructions: after each turn, they choose how strongly the effect is applied, then issue the next instruction based on what still differs from the sound they intend. In a preliminary evaluation on SocialFX-derived descriptor pairs, CLAP-guided refinement achieves lower DSP-feature MMD than an LLM+LLM initialize-then-reprompt baseline on 9 of 10 pairs. Trajectory analysis further shows that, for differentiable effects, optimization tends to gradually move the audio toward the new target while retaining the effects of the previous instruction, highlighting the potential for gradual refinement. Audio demo and source code are available at `https://instructfx2fx.vaclis.net` and `https://github.com/vaclisinc/InstructFX2FX`.

1. INTRODUCTION

Audio effect design is rarely a one-shot process. In practice, audio engineers and producers establish a rough effect chain and then iteratively refine it through feedback such as "make it brighter," "add some weight," or "ease off the reverb." Each instruction is interpreted relative to the current sound and parameter state, not as an independent request from scratch.

We call this problem *sequential FX refinement*: each new instruction updates the existing effect chain and parameter state, so that the rendered audio moves toward the user's evolving intent while preserving prior instructions. This differs from conventional text-to-effect generation, where each prompt is processed independently. We formalize it in Section 2.

Recent works address only the single-turn case. Text2FX and FxSearcher both optimize effect parameters against a CLAP objective, differing mainly in the optimizer: Text2FX uses gradient descent through differentiable effects from a random initialization [2], while FxSearcher uses gradient-free Bayesian optimization for non-differentiable plugins [3]. LLM-based systems instead map text directly to parameters: LLM2Fx claims that an LLM, drawing on pretrained knowledge and in-context examples, can outperform CLAP-based optimization [1], and a subsequent tool-calling variant further selects and orders the FX chain through chain-of-thought planning [4]; both, however, remain single-shot. None of these directly support sequential refinement, and naïvely reprompting an LLM at each turn is unreliable: the model regenerates parameters non-deterministically, may perturb settings the user did not intend to change across iterations, and cannot hear the sound it is editing.

We therefore propose a hybrid design in which the system works like a single audio engineer, pairing a brain with an ear. The language model acts as the "brain": it selects which effects to apply and generates reasonable initial parameters. CLAP-guided optimization then acts as the "ear": it listens to the rendered result and iteratively adjusts the parameters to pull the audio embedding toward the user instruction. The LLM thus provides the starting point, and CLAP the refinement direction.

We realize these ideas in *InstructFX2FX*, a multi-turn text-to-effect system for sequential audio effect refinement. This paper makes three contributions. First, we formulate sequential FX refinement as a stateful, multi-turn parameter-update problem, distinct from single-shot text-to-effect generation. Second, we build a system that routes each instruction to reuse, initialize, or jointly refine an effect chain, pairing LLM-based planning with CLAP-guided gradient descent and Bayesian optimization. Third, we provide an interactive demonstration with a preliminary evaluation on the SocialFX dataset.

2. PROBLEM FORMULATION

We formulate sequential FX refinement as a stateful parameter-update problem. Let x denote the dry input audio, C_t the current effect chain at turn t, P_t the corresponding effect parameters, and $H_{t-1} = \{I_1, \ldots, I_{t-1}\}$ the prior instruction history. At each turn, given a user instruction I_t, the system produces an updated audio

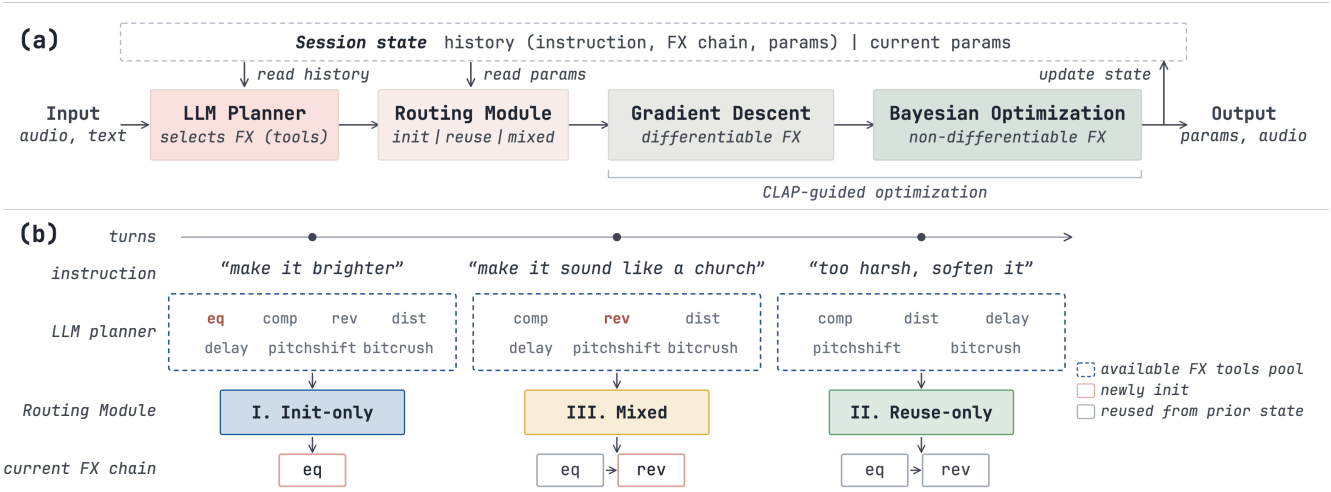


Figure 1: InstructFX2FX architecture. (a) The harness: an LLM planner and routing module drive a CLAP-guided optimization backend over a persistent session state, which makes the system iterative. (b) The three routing modes on an example three-turn session.

effects chain and parameter state through update function f:

$$(C_t, P_t) = f(x, C_{t-1}, P_{t-1}, I_t, H_{t-1}).$$

The instructions are applied sequentially and cumulatively across turns, denoted as $I_1 \rightarrow I_2 \rightarrow \ldots \rightarrow I_{t-1}$, where the right arrow reads as *then*.

By contrast, single-shot text-to-effect generation is defined as:

$$P = g(x, I),$$

where each instruction is processed separately, without tracking history. Sequential refinement, on the other hand, requires each instruction to move the current sound toward the new target while *preserving* the effects of prior instructions. This requirement motivates the three design choices detailed in Section 3.

3. SYSTEM OVERVIEW

InstructFX2FX maintains a persistent session state so that the update function f of Section 2 can build on earlier turns. As shown in Figure 1(a), this state has two parts: a history of past turns, each recording the instruction, its effect chain, and the resulting parameters; and the current parameter state. The three components of the system each consult this state before acting: the LLM planner, the routing module, and the CLAP-guided optimization backend, which we detail below.

3.1. LLM Planner

The LLM planner serves as the system's decision layer, translating each natural-language instruction into a decision about which effects the updated chain should contain. Leveraging its pretrained audio-production knowledge, the LLM orchestrates the composition of the effect chain across turns, conditioned on the session state and history. Selection occurs over an explicit pool of supported effects: each effect is exposed as a callable tool, and effects already present in the session are withheld from the available tool set; if needed, they are refined in-place.

3.2. Routing Module

From the LLM planner's selection, the routing module compares the required effects against the current chain and dispatches the instruction into one of three modes:

- **Initialize-only:** no existing effects are present, so the LLM initializes a new chain and parameters.
- **Reuse-only:** all relevant effects already exist, so the system refines the current parameters in place.
- **Mixed:** part of the existing chain is reused, while newly required effects are initialized and inserted.

Figure 1(b) illustrates these three modes in an example three-turn session.

3.3. Optimization Backends

Once the routing module has assembled the effect chain and parameter state, InstructFX2FX refines the parameter state through CLAP-guided optimization. We currently support two differentiable effects (EQ, reverb) and five non-differentiable Pedalboard effects (compressor, distortion, delay, bitcrush, pitch shift).

The two backends minimize CLAP-based objectives adapted from prior text-guided FX work: gradient descent for the differentiable effects and Bayesian optimization for the non-differentiable ones [2, 3]. Let $\phi_{\text{audio}}(\cdot)$ and $\phi_{\text{text}}(\cdot)$ denote the CLAP audio and text embedding functions [5], and let $\mathbf{x}$ denote the audio rendered by processing the dry input x through the current effect chain.

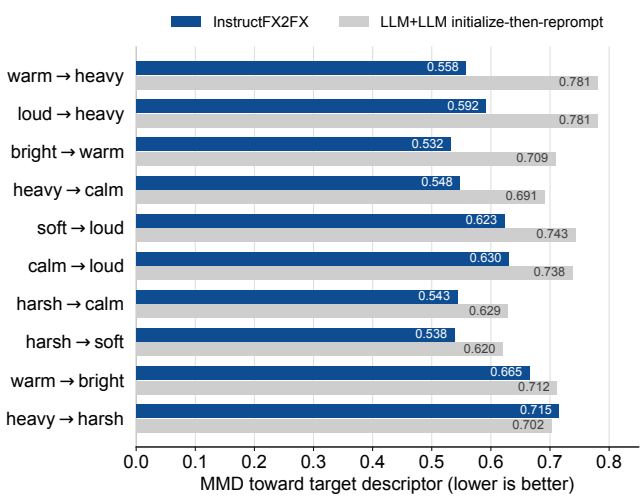
The framework provides three interchangeable objectives, any of which can be selected per turn. The semantic similarity objective directly aligns rendered audio with the target instruction:

$$\mathcal{L}_{\text{sem}} = 1 - \cos\big(\phi_{\text{audio}}(\mathbf{x}), \phi_{\text{text}}(I)\big). \quad (1)$$

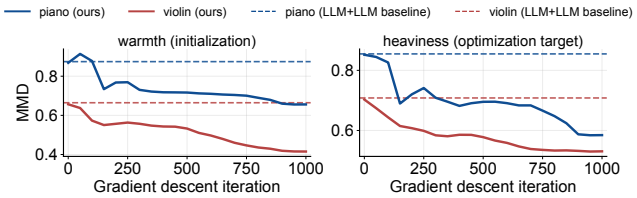
For sequential refinement, the directional objective aligns the change in the audio embedding with the semantic transition from a source descriptor S to a target descriptor T:

$$\mathcal{L}_{\text{dir}} = 1 - \cos\big(\phi_{\text{audio}}(\mathbf{x}(k)) - \phi_{\text{audio}}(\mathbf{x}_0), \phi_{\text{text}}(T) - \phi_{\text{text}}(S)\big), \quad (2)$$

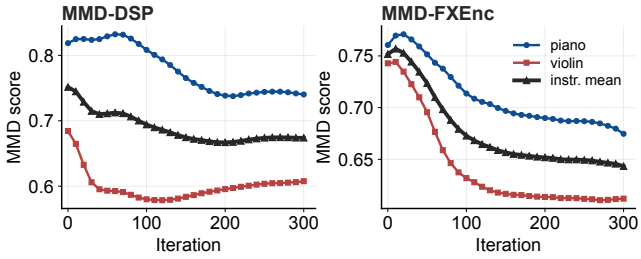
where $\mathbf{x}(k)$ is the rendered audio at optimization step k and $\mathbf{x}_0$ its value at initialization.



(a) Exp 1: Sequential MMD across prompt pairs.



(b) Exp 2: Case study of *warmth* (init) — *heaviness* (optimization target).



(c) Exp 3: LLM initialization ablation.

Figure 2: Evaluation results. (a) Exp 1: InstructFX2FX lowers MMD versus an LLM+LLM initialize-then-reprompt baseline on 9 of 10 pairs. (b) Exp 2: Refinement drives the target MMD (*heaviness*, right) below the LLM+LLM initialize-then-reprompt baseline while the initialized MMD (*warmth*, left) does not rise, showing that the audio moves toward the target without discarding earlier instruction. (c) Exp 3: departing from LLM initialization, CLAP-guided refinement achieves lower DSP-feature MMD (left) and Fx-Encoder++ MMD (right).

Finally, a guided objective pushes the rendered audio toward a positive target I^+ and away from a negative anchor I^-:

$$\mathcal{L}_{\text{guided}} = 1 - \cos\big(\phi_{\text{audio}}(\mathbf{x}), \phi_{\text{text}}(I^+)\big) + \cos\big(\phi_{\text{audio}}(\mathbf{x}), \phi_{\text{text}}(I^-)\big). \tag{3}$$

Because the desired strength of an effect is subjective, the system stores snapshots along optimization trajectories and exposes them in the demo interface as a slider, letting the user audition intermediate results and choose the effect strength they prefer.

4. EVALUATION

We run a preliminary evaluation on EQ descriptors from the SocialFX dataset [6], testing whether CLAP-guided refinement improves on both an LLM+LLM initialize-then-reprompt baseline and the LLM's own initialization.

4.1. Dataset

We evaluate on SocialFX [6], a crowd-sourced dataset that pairs natural-language descriptors with audio-effect settings. We restrict attention to EQ descriptors and keep those that are both frequent (at least 20 annotations) and reliable (inter-annotator consistency at least 0.45), which yields seven words: *warm*, *bright*, *soft*, *loud*, *harsh*, *calm*, and *heavy*. From these we form 10 pairs of sequential instructions (e.g., *warm* → *bright*, *harsh* → *soft*), which we apply to dry self-recorded violin and open-source piano recordings.

4.2. Metric: Maximum Mean Discrepancy

We measure quality with the Maximum Mean Discrepancy (MMD) between the system-output and ground-truth distributions, using a Gaussian kernel (median-bandwidth heuristic) over 35-dimensional DSP feature vectors (spectral, cepstral, and temporal descriptors); lower is better. Because MMD is computed on DSP features, it is independent of the optimization objective defined within the CLAP space, so a reduction in the MMD score compared to a baseline effectively shows that our system improves the manipulation of audio effects at the signal level. For Experiment 3, we additionally report MMD in the Fx-Encoder++ [7] embedding space, a pretrained audio-effect encoder, used in prior work on single-shot parameter setting [1].

4.3. Experiment 1: Sequential MMD

For each of the 10 pairs of sequential instructions $A \rightarrow B$ described in Section 4.1, we construct the ground-truth references by first applying the SocialFX EQ parameters of descriptor A to the dry audio, then stacking those of descriptor B on top. Experiment 1 compares the DSP-feature MMD of InstructFX2FX and an LLM+LLM initialize-then-reprompt baseline against this ground truth. Both methods begin with LLM initialization for descriptor A . InstructFX2FX then refines the parameters according to the applicable routing scheme, whereas the baseline applies an LLM reprompt for descriptor B. We find that InstructFX2FX achieves lower DSP-feature MMD than the baseline on 9 of 10 descriptor pairs (Figure2a). The largest improvement is observed for the strongest timbral transition, warm $\rightarrow$heavy (roughly 0.22).

4.4. Experiment 2: MMD Convergence Trajectory

For Experiment 2, we build two independent ground truth references, complementing the sequential ground truth used in Experiment 1. Specifically, we apply the SocialFX EQ parameters of descriptors A and B *separately* to the dry audio, and record the DSP-feature MMD of the refined audio to each single-word reference throughout CLAP optimization. A successful refinement should keep the MMD for the previous descriptor A from increasing while reducing the MMD to the target descriptor B. Inspired by prior work [2], we adopt the directional objective, where the source descriptor S is chosen as the semantic opposite of the target descriptor T. This objective emphasizes directional alignment

rather than direct similarity, encouraging refinement while retaining previous instructions. We observe this behavior for 9 of the 10 descriptor pairs across both piano and violin recordings. Figure 2b shows a representative example. Over CLAP gradient descent iterations, the DSP-feature MMD to the optimization target reference *heavy* decreases, while the DSP-feature MMD to the initialization reference *warm* does not rise, indicating that the optimization moves the audio toward the optimization target without discarding the previous instruction. Plots for all 10 descriptor pairs are available on the demo website.

4.5. Experiment 3: LLM Initialization Ablation

This ablation isolates the contribution of CLAP-guided refinement from that of the LLM initialization. Unlike previous experiments, we evaluate on single-descriptor words: seven descriptors from Section 4.1 plus *happy*. We obtain the ground truth references by applying the SocialFX EQ parameters of each descriptor to the dry audio. For each word and instrument, we sample the LLM at temperature 1.0 to obtain a set of stochastic initializations per pair. We then refine the parameter set by navigating the CLAP space toward the same target descriptor as the one used for initialization, making iteration 0 of every trajectory in Figure 2c the LLM baseline itself. Averaged across both instruments, refinement reduces DSP-feature MMD from 0.752 to 0.674 (about 10%) and Fx-Encoder++ MMD from 0.752 to 0.643 (about 14%), showing the value of CLAP optimization after LLM initialization for 7 of 8 words, the sole exception being *warm*.

Experiment 2 and 3 show that CLAP optimization offers a capability that LLMs alone cannot: gradual refinement. This strongly motivates the intermediate-checkpoint slider, allowing users to listen to gradual changes in the sound in order to ultimately settle on the parameter set that creates their desired effect.

5. LIMITATIONS AND FUTURE WORK

Several optimization and evaluation challenges remain. First, our quantitative evaluation centers primarily on EQ-related descriptors, because SocialFX provides cleaner descriptor-to-parameter associations for EQ than for more complex effects.

Second, CLAP embedding space does not capture all perceptual properties relevant to audio production [8]. Although CLAP-guided refinement often improves early behavior, trajectories may later drift or overshoot, and gains in CLAP space do not always correspond to gains in DSP-feature metrics or perceptual quality.

Third, non-differentiable effects remain difficult to refine reliably. Bayesian optimization extends support to Pedalboard effects such as distortion, delay, and bitcrushing, but its trajectories are often noisy and semantically inconsistent across iterations.

Finally, optimization currently requires seconds per interaction turn, limiting applicability within production DAW workflows.

Future directions include richer multi-effect chains, human listening studies, domain-adapted audio-effect embedding models, improving stability for non-differentiable refinement, and real-time VST/AU integration. Among these, the most promising in our view is to move beyond inference-time optimization toward generative parameter models, for example flow-matching models trained directly on iterative FX-editing trajectories that learn sequential refinement end-to-end.

6. CONCLUSION

We presented InstructFX2FX, a multi-turn text-to-effect system for sequential text-guided audio effect refinement. Unlike single-shot text-to-effect systems, InstructFX2FX treats audio effect editing as a stateful process in which each new instruction updates an existing chain and parameter state.

The core idea is leveraging complementary sources of machine intelligence: the LLM provides high-level effect planning and initialization, while CLAP-guided optimization provides perceptual refinement. Preliminary experiments on SocialFX-derived descriptors suggest that CLAP-guided refinement can improve over LLM reprompting in many sequential settings, while also exposing limitations such as CLAP/DSP-feature mismatch, optimization drift, and instability for non-differentiable effects. More broadly, this demo is a proof-of-concept for interactive audio-effect systems that better reflect practical creative workflows. We see it as a first step, and hope it motivates the community to take up *sequential FX refinement* as a problem in its own right.

7. REFERENCES

[1] S. Doh, J. Koo, M. A. Martínez-Ramírez, W.-H. Liao, J. Nam, and Y. Mitsufuji, "Can large language models predict audio effects parameters from natural language?" in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, 2025.

[2] A. Chu, P. O'Reilly, J. Barnett, and B. Pardo, "Text2FX: Harnessing CLAP embeddings for text-guided audio effects," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2025.

[3] H. Ki, J. Kim, M. Kwon, and J. Kim, "FxSearcher: Gradient-free text-driven audio transformation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2026.

[4] 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 *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2026.

[5] Y. Wu, K. Chen, T. Zhang, Y. Hui, M. Nezhurina, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2023.

[6] T. Zheng, P. Seetharaman, and B. Pardo, "SocialFX: Studying a crowdsourced folksonomy of audio effects terms," in *Proceedings of the 24th ACM International Conference on Multimedia*, Amsterdam, The Netherlands, 2016, pp. 182–186.

[7] Y.-T. Yeh, J. Koo, M. A. Martínez-Ramírez, W.-H. Liao, Y.-H. Yang, and Y. Mitsufuji, "Fx-Encoder++: Extracting Instrument-Wise Audio Effects Representations from Mixtures," in *Proc. Int. Soc. Music Inf. Retr. Conf. (ISMIR)*, 2025.

[8] Q. Deng, B. Pardo, and T. N. Pappas, "Do joint language-audio embeddings encode perceptual timbre semantics?" in *Proc. NeurIPS Workshop on AI for Music*, 2025.