EVALUATING LOSS FUNCTIONS IN DIFFERENTIABLE OUT-OF-DOMAIN SOUND-MATCHING WITH PARTIAL PARAMETER DISTANCE

Amir Salimi, Daniel Penner, Kalvin Eng, Abram Hindle, Osmar R. Zaïane

University of Alberta

ABSTRACT

In *out-of-domain* (OOD) sound-matching, a synthesizer is optimized to mimic a sound it did not generate. OOD evaluation of loss functions is underexplored in part because the standard "parameter loss" metric requires a shared parameter space between target and imitator, which OOD settings lack. We introduce Partial Parameter Distance (PPD), which applies parameter loss only to the critical parameters that mismatched synthesizers share (e.g., filter cutoffs), enabling automatically evaluated cross-synthesizer OOD experiments; we verify its results with blinded listening tests. Across seven scenarios involving band-pass filtering, amplitude modulation, and pitch-bending, we evaluate four differentiable loss functions (`SIMSE_Spec`, `L1_Spec`, `JTFS`, `DTW_Envelope`). Loss-function effectiveness remains tightly coupled to the method of synthesis: `SIMSE_Spec` excels at filter-cutoff recovery, `DTW_Envelope` at amplitude-modulation recovery, and `JTFS` at smooth pitch trajectories. Parameter-based evaluation agrees with listening tests on the top-ranked loss function in five of seven scenarios, supporting its use as a targeted diagnostic.

***Index Terms*—** Sound-matching, synthesizer programming, evolution strategies, loss functions

1. INTRODUCTION

In sound design, artists iteratively manipulate the parameters of their digital synthesizers to produce a desired sound [1]. This manual process can be a rewarding creative outlet, but often requires expertise and patience. *Sound-matching* refers to the automation of this workflow: given a target sound and a synthesizer, the objective is to identify parameters that produce a perceptually similar output [2, 3]. Sound-matching experiments are either *in-domain* or *out-of-domain* (OOD). For in-domain experiments, the target sound and the mimicking outputs are generated by the same synthesizer. In OOD settings, the target sound originates from a different synthesis process or recording. While OOD sound-matching more closely resembles practical sound design, prior work overwhelmingly focuses on in-domain evaluations, with the goal of replicating the entire sound [2, 4, 5, 6, 7]. Since manual audio evaluations are impractical at scale, the lack of reliable automatic evaluation methods is a major hurdle in OOD sound-matching.

We propose an approach to OOD experiments that can be automatically evaluated, and use this methodology to re-examine the results from recent in-domain experiments. In particular, recent work by Salimi et al. [7] tested audio loss functions (differentiable measures of similarity between two sounds) in different in-domain sound-matching tasks and observed that there is no best universal loss function; that is, the performance of loss functions depends on the method of synthesis. Here, we test whether the reported observations also occur in OOD scenarios, and verify the results with listening tests. Each scenario pairs a target synthesizer with an imitator whose parameters are optimized to mimic the target's output.

The synthesizers are designed to mismatch while retaining a partial, critical overlap in parameters. In addition, the synthesizers are minimal representations of common signal processing functions used by sound designers (e.g., filtering, amplitude/pitch modulation) [7]. This also simplifies the selection of a critical parameter for each synthesizer pair. For example, if two synthesizers apply amplitude modulation (AM) to different source signals, the modulation rate is the partially overlapping parameter and a reasonable starting point for measuring success. Matching a target sound's critical characteristics, rather than replicating it outright, mirrors the practice of sound design; success on critical parameters is thus a meaningful unit of progress for OOD sound-matching. We call the application of parameter loss to mismatched synthesizers *partial parameter distance* (PPD).

Our central hypothesis is that the performance rankings of loss functions vary with the synthesizers used in OOD scenarios. We address three research questions: **(RQ1)** Does the synthesis-dependent loss behavior observed in-domain also occur in OOD settings? **(RQ2)** Do the reported advantages and limitations of common loss functions persist under partial synthesizer mismatch? **(RQ3)** Can PPD serve as an automatic evaluation procedure for OOD sound-matching, and how well does it align with human perception? To answer these questions, we contribute a methodology for controlled differentiable OOD sound-matching experiments based on synthesizer pairs with partially overlapping functionality, with PPD as its automated evaluation procedure, compared against blinded listening tests; the framework can serve as a screening tool

for loss functions before they are deployed in more complex settings. Replication code, audio examples, and an interactive demo are available on the companion page.[1]

2. BACKGROUND

Sound-matching optimizes parameters θ of a synthesizer $g(\theta)$ so that $x_\theta = g(\theta)$ matches a target $x^t = g^t(\theta^t)$ under a similarity measure $L(x_\theta, x^t)$; in a differentiable pipeline, θ is updated via gradients of L [4, 8]. Most similarity measures fall into two families: audio-based and parameter-based. Audio-based losses compute distances between acoustic representations such as STFT spectrograms (e.g., L1, L2, or scale-invariant MSE) or amplitude envelopes; extensions such as Joint Time-Frequency Scattering (JTFS) apply wavelet transformations to a time-frequency representation and have been proposed for capturing *mesostructures* [9, 6, 10]. Parameter Loss (P-Loss) compares the target and imitator parameter vectors directly (e.g., $\|\theta - \theta^t\|_2$) and requires that both synthesizers share the same parameter space. While P-Loss can serve as a training objective when ground-truth parameters are available [4, 8], it is undefined in OOD settings, where there is no 1-to-1 parameter mapping between the target and imitator synthesizers.

Systematic comparison of loss functions under controlled OOD conditions is largely absent from the literature; most works evaluate exclusively in-domain [2, 3, 4, 5, 6, 11]. Recent OOD work has focused on training-time adaptation: fine-tuning parameter estimators on OOD targets [12, 13, 14] or using reinforcement learning for cross-domain matching [15], rather than comparing loss functions in controlled OOD conditions. We address this gap by comparing loss functions across OOD scenarios built from minimal synthesizer pairs, each representing a fundamental approach to sound design [7].

3. METHODOLOGY

3.1. Problem Setup

We evaluate four differentiable loss functions across seven scenarios (six OOD, one in-domain). A **scenario** is an imitator–target synthesizer pair (g, g^t); for each scenario we run one experiment per loss function, and each experiment consists of at least 300 independent trials in which the target parameters θ^t and the imitator parameters $\hat{\theta}$ are drawn uniformly over predefined ranges (targets are not shared across loss functions); 200 steps of gradient-based optimization then update $\hat{\theta}$ to minimize the loss between $g(\hat{\theta})$ and $g^t(\theta^t)$. All synthesizers produce 1-second signals at 44,100 Hz.

3.2. Loss Functions

The four loss functions are as follows. `L1_Spec` is the pointwise L1 distance between magnitude STFT spectrograms. `SIMSE_Spec` replaces the L1 term with Scale-Invariant Mean Squared Error (SIMSE) [16], computed on magnitudes clipped to $[0, 1]$; SIMSE rescales one spectrogram by a least-squares gain before taking the MSE, making it less sensitive to global level differences. `JTFS` applies the L2 distance to the 1-D Joint Time-Frequency Scattering transform [9] (J=10, Q=2), which captures wavelet-based time-frequency structure. `DTW_Envelope` computes the soft-DTW distance [17] (γ=1) between amplitude envelopes, where the envelope at each time step is obtained by summing all frequency bins of an STFT and smoothing with a Gaussian kernel (σ=3 frames). For all STFT-based computations (`L1_Spec`, `SIMSE_Spec`, and the `DTW_Envelope` envelope), we use a 512-point FFT (257 frequency bins), a Hann window, and a hop length of 100 samples.

3.3. Evaluation Methods

Partial Parameter Distance. In OOD settings, full parameter comparison is undefined because the target and imitator synthesizers have different architectures or parameter ranges. For example, if the target is a sine-wave AM synthesizer (carrier 200 Hz, modulation 10 Hz) and the imitator is a saw-wave AM synthesizer (carrier 30–5,000 Hz, modulation 1–15 Hz), the only reasonable point of comparison is the modulation rate. *Partial Parameter Distance* (PPD) applies P-Loss only to the subset of parameters with critical perceptual correspondence between the two synthesizers, computed after optimization:

$$\mathrm{PPD} = \|\theta^t_{\mathrm{crit}} - \hat{\theta}_{\mathrm{crit}}\|_2, \tag{1}$$

where θ_{crit} denotes normalized $[0, 1]$ parameters common to g and g^t. Selecting critical parameters requires domain knowledge and carries some subjectivity; our scenarios (Section 4) are designed with a single dominant synthesis task per pair to keep this selection unambiguous. PPD is thus a per-scenario evaluation procedure rather than a universal metric, consistent with the premise that no single similarity measure serves all synthesis tasks [7].

Listening tests. Listening tests serve as the primary ground truth. Following Salimi et al. [7], in each scenario we randomly sample 40 (target, imitator) audio pairs per loss function; four authors independently rate each pair on a 5-point Likert scale ("are the two sounds similar?"), yielding 160 scores per loss per scenario. Raters are blinded to the generating loss function, and three of the four raters were not involved in synthesizer implementation, limiting familiarity-driven bias.

3.4. Optimization Loop

The four loss functions are implemented as differentiable JAX functions, and the synthesizers are written in Faust and transpiled into JAX via DawDreamer [18]; this pipeline is faster than real-time and parallelizable. Parameters are updated using RMSProp with a fixed learning rate of 0.045 for 200 iterations,

[1] `https://imilas.github.io/ood-sound-matching-ppd/`

following [7]. Gradients are clipped to an L2 norm of 1 to mitigate exploding gradients [7].

3.5. Bootstrapping and Ranking Procedures

We follow the methodology of Salimi et al. [7] for ranking the loss functions. For each loss function and scenario, we bootstrap 1,000 resampled means from the PPD scores (or 160 Likert scores) to obtain nonparametric performance distributions [19]; since PPD is a distance, we use the reciprocal of each resampled mean (1/PPD, as plotted in Fig. 1) so that higher values indicate better performance for both measures. These distributions are then ranked using the *nonparametric Scott-Knott procedure* (NPSK), which recursively partitions distributions into statistically distinct groups without assuming normality [20]. Rank 1 indicates the best-performing loss; ties indicate distributions that NPSK cannot distinguish. We note that running NPSK on the raw inputs instead (no bootstrapping) leaves the top-ranked loss unchanged in every scenario; lower ranks partly merge into ties. The scenario rankings are summarized in Fig. 1.

4. SCENARIOS AND RESULTS

The scenarios tested here use simple synthesis methods common in prior work [21, 6] and sound design [22]; this simplicity isolates the interaction between loss function and synthesizer [7]. Searchable parameters are likewise restricted to those that dominate each sound's perceptual identity (e.g., filter cutoffs rather than resonance), keeping critical-parameter selection unambiguous; the pulsating chirp is deliberately uncommon: it composes a pitch trajectory with an LFO to probe `JTFS`'s reported strength on mesostructures [6]. Figure 1 shows the per-scenario bootstrapped distributions, NPSK ranks, and inter-rater reliability.

4.1. Band-Pass Matching

Two synthesizers apply band-pass filtering to different source signals: **BP-Saw** (target: band-passed saw oscillator) and **BP-Noise** (imitator: band-passed white noise). Both share cutoffs `lp_cut`$\in$[100,5,000] Hz and `hp_cut`$\in$[1,400] Hz; the differing source signals (broadband stochastic vs. deterministic harmonic series) make exact replication impossible. PPD is the L2 distance between the normalized cutoffs. Both PPD and listeners rank `SIMSE_Spec` first (Fig. 1), consistent with in-domain findings that spectrogram-based losses are best for filter-cutoff recovery [7].

4.2. AM-Synthesizer Matching

Three OOD scenarios are built from two AM synthesizers: **SineSaw-AM** (saw carrier modulated by a sine LFO) and **SineSine-AM** (sine carrier modulated by a sine LFO). Each has a modulation-rate parameter (`amp`) in [1,15] Hz and a carrier frequency (`carrier`). Because the modulator sits in the sub-audio range, the effect is a slow tremolo rather than the sidebands of true ring modulation. PPD is the L2 distance on `amp`; `carrier` is excluded because the carrier ranges or waveforms differ by design and, given logarithmic pitch perception, no single "correct" frequency match exists [23]. The three variants are: *(i) non-overlapping frequencies*: two **SineSine-AM** instances with disjoint carrier ranges (50–1,500 Hz target, 2,000–8,000 Hz imitator); *(ii) sine→saw* and *(iii) saw→sine*: mirrored scenarios between **SineSine-AM** and **SineSaw-AM** on overlapping carrier ranges (30–5,000 Hz), where numerical carrier values can coincide but harmonic content still differs. `DTW_Envelope` ranks first on PPD in all three, confirming accurate recovery of the modulation rate. Listeners agree in the non-overlapping case, but in the overlapping-carrier scenarios they prefer a spectrogram-based loss, whose outputs achieve better timbral similarity at the cost of envelope accuracy.

4.3. Pitch-Bending

Three "chirplet" [24, 10] scenarios use a base synthesizer that emits a sine tone at a starting pitch (30–100 Hz) that, after δ samples of steady tone, undergoes an exponential upward pitch bend at rate `increase_speed`$\in$[1,20]. (i) *Delayed* (OOD): δ is drawn uniformly from 5,000–30,000 samples and is not searched, so the imitator cannot replicate the onset time; PPD is the L2 distance on `starting_pitch` and `increase_speed`. (ii) *No-delay* (in-domain, $\delta = 0$): a close replication of the chirplet experiments in Vahidi et al. [6]. (iii) *Pulsating* (OOD): a saw-wave LFO with frequency `pulse_rate`$\in$[1,10] Hz oscillates the pitch around the increasing trajectory (`increase_speed` restricted to $[2, 7]$ for this variant); the pulse depth is an unsearched constant sampled per synthesizer (50–1,000 Hz), so target and imitator generally cannot coincide. PPD is the L2 distance on `increase_speed` and `pulse_rate`. `JTFS` ranks first in the delayed and no-delay scenarios under both parameter-based and listening-test rankings, consistent with its reported agnosticism to small time and frequency shifts [6]. In the pulsating scenario, `DTW_Envelope` displaces `JTFS` to a tie for second place with `SIMSE_Spec`.

5. DISCUSSION

We find that synthesis-dependent loss behavior generalizes to OOD settings **(RQ1)**: `SIMSE_Spec` is best for band-pass matching, `DTW_Envelope` for amplitude modulation and pulsating pitch, and `JTFS` for smooth pitch trajectories. `L1_Spec` ranks first in only one case (Likert on AM: saw→sine) and is assigned NPSK rank 4 in five of the fourteen rankings of Fig. 1, while `SIMSE_Spec` is never ranked last, suggesting that scale invariance makes spectrogram losses more robust across synthesis tasks **(RQ2)**. PPD agrees with listening tests

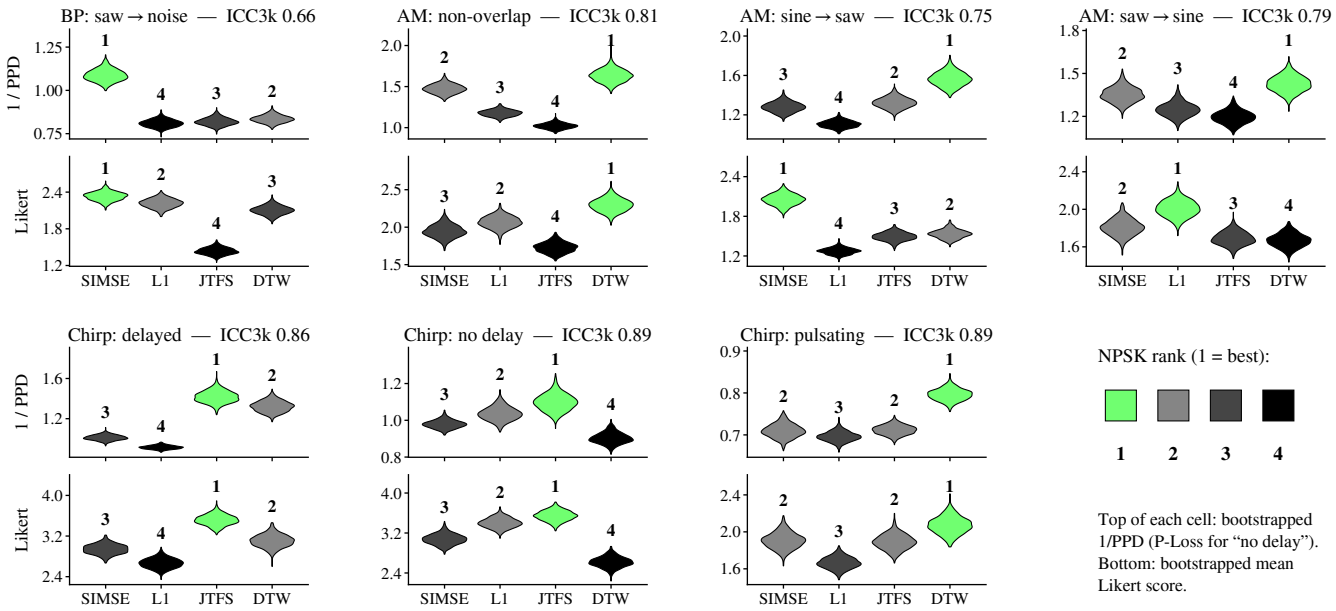


Fig. 1. Bootstrapped $1/\mathrm{PPD}$ (top panel of each cell; P-Loss for the in-domain "no-delay" scenario) and mean Likert rating (bottom panel) distributions for each scenario, with violins colored and numbered by NPSK rank (1 = best). Higher is better on both axes. Scenario labels read target → imitator; each title gives the ICC3k inter-rater reliability of the four-rater average.

on the top-ranked loss in five of seven scenarios **(RQ3)**; the two disagreements occur in AM with overlapping carriers (Section 4.2), where `DTW_Envelope` ranks first on PPD (accurate modulation rate) but listeners favor spectrogram losses (better timbral similarity). This emphasizes that PPD measures a targeted parameter set and should be paired with listening tests when multiple perceptual factors compete. Notably, `JTFS` excels on smooth pitch bends but is displaced by `DTW_Envelope` on the pulsating variant **(RQ2)**, indicating that its "mesostructure" domain is more narrowly scoped than previously suggested [7, 6]. Inter-rater reliability of the four-rater average used in our analysis, measured by the intraclass correlation coefficient for the mean of $k{=}4$ raters (ICC3k) [25], is good (pooled ICC3k = 0.86, $p < 0.001$); per-scenario ICC3k values appear in the panel titles of Fig. 1. The listening-derived rankings are also robust to panel composition: the top-ranked loss is unchanged under every leave-one-rater-out ablation (mean rank correlation $\rho = 0.98$). The noisiness of individual similarity judgments in OOD settings is itself an argument for complementing listening panels with automatic measures such as PPD.

Practical implications. Under a shared gradient-based configuration, loss selection should follow the synthesis task: spectrogram-based losses for filtering (with `SIMSE_Spec` preferred to `L1_Spec`), `DTW_Envelope` for amplitude modulation, and `JTFS` for smooth pitch trajectories. For OOD evaluation, PPD with overlapping synthesizers is a viable strategy provided the critical parameters are perceptually dominant; pairing with a small listening panel is recommended. PPD is not intended for arbitrary synthesizer pairs, where critical parameters may be ill-defined; rather, it serves as a screening tool: loss functions screened in controlled scenarios can then be deployed in complex settings where parameter correspondence no longer exists. PPD may also complement P-Loss for in-domain settings where the goal is imitation of specific sound characteristics rather than direct parameter replication.

Limitations and future work. Whether PPD scales to higher-dimensional parameter spaces or generalizes to real recorded targets is an open question; the selection of critical parameters also carries subjectivity that our simple synthesizer pairs sidestep. Our listening panel is small and composed of authors; blinding and high inter-rater reliability mitigate but do not remove this limitation, and validation against a larger external panel remains future work. All losses also share a single optimizer configuration, so rankings may partly reflect interactions between loss scaling and the learning rate rather than intrinsic loss behavior. Our results are also conditioned on gradient-based optimization: differentiable pipelines are known to struggle with non-convex loss surfaces induced by oscillatory parameters [26, 27, 7], so loss rankings may shift under other search strategies (e.g., evolutionary methods). Promising directions include OOD matching with pre-recorded targets, real-time sound-matching [28], and composite matching algorithms that deploy task-specific losses validated in isolation within a single system.

6. REFERENCES

[1] G. Krekovic, "Insights in habits and attitudes regarding programming sound synthesizers: A quantitative study," in *Proc. Sound and Music Computing Conf.*, 2019.

[2] A. Horner, J. Beauchamp, and L. Haken, "Machine tongues XVI: Genetic algorithms and their application to FM matching synthesis," *Computer Music Journal*, vol. 17, no. 4, pp. 17–29, 1993.

[3] T. J. Mitchell and D. P. Creasey, "Evolutionary sound matching: A test methodology and comparative study," in *Proc. ICMLA*, 2007, pp. 229–234.

[4] M. J. Yee-King, L. Fedden, and M. d'Inverno, "Automatic programming of VST sound synthesizers using deep networks and other techniques," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 2, pp. 150–159, 2018.

[5] H. Han, V. Lostanlen, and M. Lagrange, "Perceptual–neural–physical sound matching," in *Proc. IEEE ICASSP*. IEEE, 2023, pp. 1–5.

[6] C. Vahidi, H. Han, C. Wang, M. Lagrange, G. Fazekas, and V. Lostanlen, "Mesostructures: Beyond spectrogram loss in differentiable time–frequency analysis," *J. Audio Eng. Soc.*, vol. 71, no. 9, pp. 577–585, September 2023.

[7] A. Salimi, A. Hindle, and O. R. Zaïane, "Evaluating sound similarity metrics for differentiable, iterative sound-matching," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 33, pp. 4982–4994, 2025.

[8] P. Esling, N. Masuda, A. Bardet, R. Despres, and A. Chemla-Romeu-Santos, "Flow synthesizer: Universal audio synthesizer control with normalizing flows," *Applied Sciences*, vol. 10, no. 1, p. 302, 2019.

[9] J. Andén, V. Lostanlen, and S. Mallat, "Joint time-frequency scattering for audio classification," in *Proc. IEEE MLSP*, 2015, pp. 1–6.

[10] V. Lostanlen and F. Hecker, "The shape of remixxxes to come: audio texture synthesis with time-frequency scattering," in *Proc. DAFx*, 2019.

[11] O. Barkan, D. Tsiris, O. Katz, and N. Koenigstein, "Inversynth: Deep estimation of synthesizer parameter configurations from audio signals," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 12, pp. 2385–2396, 2019.

[12] N. Masuda and D. Saito, "Synthesizer sound matching with differentiable DSP," in *ISMIR*, 2021, pp. 428–434.

[13] ——, "Improving semi-supervised differentiable synthesizer sound matching for practical applications," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 863–875, 2023.

[14] N. Uzrad, O. Barkan, A. Elharar, S. Shvartzman, M. Laufer, L. Wolf, and N. Koenigstein, "DiffMoog: A differentiable modular synthesizer for sound matching," *arXiv preprint arXiv:2401.12570*, 2024.

[15] W. Shin and K. Lee, "SynthRL: Cross-domain synthesizer sound matching via reinforcement learning," in *Proc. IJCAI*, 2025, pp. 10 162–10 170.

[16] J. T. Barron and J. Malik, "Shape, illumination, and reflectance from shading," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1670–1687, 2015.

[17] M. Cuturi and M. Blondel, "Soft-DTW: A differentiable loss function for time-series," in *Proc. ICML*. PMLR, 2017, pp. 894–903.

[18] D. Braun, "DawDreamer: Bridging the gap between digital audio workstations and Python interfaces," in *Late-Breaking Demo, ISMIR*, 2021.

[19] M. R. Chernick, *Bootstrap methods: A guide for practitioners and researchers*. John Wiley & Sons, 2011.

[20] C. Tantithamthavorn, S. McIntosh, A. E. Hassan, and K. Matsumoto, "An empirical comparison of model validation techniques for defect prediction models," *IEEE Trans. Softw. Eng.*, vol. 43, no. 1, pp. 1–18, 2017.

[21] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *Proc. ICLR*, 2020.

[22] C. Roads, *The Computer Music Tutorial*. MIT Press, 1996.

[23] R. W. Young, "Terminology for logarithmic frequency units," *J. Acoust. Soc. Am.*, vol. 11, no. 1, pp. 134–139, 1939.

[24] S. Mann and S. Haykin, "The chirplet transform: Physical considerations," *IEEE Trans. Signal Process.*, vol. 43, no. 11, pp. 2745–2761, 1995.

[25] D. Liljequist, B. Elfving, and K. Skavberg Roaldsen, "Intraclass correlation–a discussion and demonstration of basic features," *PLOS ONE*, vol. 14, no. 7, p. e0219854, 2019.

[26] J. Turian and M. Henry, "I'm sorry for your loss: Spectrally-based audio distances are bad at pitch," *arXiv preprint arXiv:2012.04572*, 2020.

[27] B. Hayes, C. Saitis, and G. Fazekas, "Sinusoidal frequency estimation by gradient descent," in *Proc. IEEE ICASSP*, 2023, pp. 1–5.

[28] J. Shier, C. Saitis, A. Robertson, and A. McPherson, "Real-time timbre remapping with differentiable DSP," *arXiv preprint arXiv:2407.04547*, 2024.