COMMON OPTIMIZERS IN ITERATIVE SOUND-MATCHING: LOSS RANKINGS DO NOT TRANSFER ACROSS SEARCH METHODS

Amir Salimi, Abram Hindle, Osmar R. Zaïane

University of Alberta, Canada

ABSTRACT

Iterative sound-matching searches for synthesizer parameters that reproduce a target sound. Existing comparisons of loss functions for this task typically evaluate them under a single search method, implicitly assuming that a loss which performs well under one method performs well under others. We test this assumption by comparing gradient descent (GD), CMA-ES, LES (a learned evolution strategy), and random search across five loss functions and nine in-domain sound-matching tasks (five simple synthesizers, four complex FM synthesizers) at a fixed evaluation budget. We find that (i) parameter-loss rankings established under GD do not transfer to the gradient-free methods; (ii) CMA-ES is the strongest method on the simple synthesizers and GD the weakest; and (iii) on complex FM synthesizers, no guided method recovers parameters better than random search: the strongest, CMA-ES, only matches it. The performance of a loss function is therefore not separable from the search method that uses it.

***Index Terms*—** Sound-matching, synthesizer programming, differentiable digital signal processing, loss functions

1. INTRODUCTION

Sound-matching is the task of finding synthesizer parameters whose rendered output reproduces a target sound. It has been studied for decades as a tool for automatic synthesizer programming and sound design [1–3]. A more recent line of work makes the synthesizer itself differentiable, so that parameters can be optimized by gradient descent (GD) on an audio loss between the rendered and target sounds [4–6].

The weaknesses of GD in this setting are well documented: audio loss surfaces are highly non-convex in oscillator frequencies and related parameters, producing ripple-like landscapes on which descent stalls [7, 8], and spectrogram losses fail to capture mesostructural differences between sounds [9]. A natural response has been to seek better loss functions, and recent work compares candidate losses by how well GD performs under each [9–11]. Implicit in this setup is the assumption that a loss function's quality is a property of the loss alone. In practice, such a comparison measures the loss and the optimizer together, and the two can only be separated by testing the loss under several search methods. Evolutionary methods have a long history in sound-matching [2, 12–14], and random search has rarely been used as a baseline.

We compare four search methods: GD, the covariance matrix adaptation evolution strategy (CMA-ES) [15], a learned evolution strategy (LES) [16,17], and random search. We cross them with five loss functions spanning spectrogram, scattering, dynamic time-warping, and embedding domains, on nine in-domain sound-matching tasks (five simple synthesizers and four DX7-style FM synthesizers), with 300 trials per configuration and 200 evaluations per trial. Because targets are rendered by the synthesizer itself, the true parameters are known, and we evaluate both the audio objective and the normalized parameter distance (P-Loss). We answer three research questions:

1. *Does a loss ranking established under GD hold under gradient-free search?* No: rankings under CMA-ES and LES agree with each other far more strongly than either agrees with GD's (Sec. 4.1).
2. *Which search method performs best at a fixed evaluation budget?* CMA-ES on the simple synthesizers, but not better than random search in complex synthesis. GD is the weakest on average at various learning rates (Sec. 4.2).
3. *Does minimizing the audio loss also minimize the parameter distance?* Not on the DX7-style synthesizers: the guided methods still reduce the audio loss, with CMA-ES best on average, yet none recovers parameters better than random search (Sec. 4.3).

Together, these results show that a loss function cannot be evaluated apart from the search method that uses it, and that sound-matching pipelines should be tested under more than one search method, with random search as a baseline. Code, results, audio examples, and an interactive demo are available on the companion page: `https://imilas.github.io/soundmatching_heuristic_comparison/`

2. BACKGROUND

Sound-matching predates differentiable synthesis by decades. Horner et al. matched FM synthesizer parameters with a genetic algorithm, scoring candidates with a fixed spectral similarity measure [2], and Mitchell and Creasey proposed a test

methodology for evolutionary FM matching [13]. Yee-King et al. compared neural networks, a genetic algorithm, and a hill climber on the Dexed FM synthesizer, using mel-frequency cepstral coefficient distance as the fitness measure for the search methods [14]; Shier et al. released SpiegeLib, a library for building such pipelines, and found a genetic algorithm the best performer in a similar comparison [18]. In these studies, search methods vary while the similarity measure is held fixed.

Differentiable synthesizers allow parameters to be updated by GD on an audio loss [4–6]. The choice of that loss has become a research question of its own. Turian and Henry showed that common spectral distances form rippled surfaces that give poor gradients toward the correct pitch [7]. Vahidi et al. proposed differentiable joint time-frequency scattering and reported better matching of chirplet sounds than with spectrogram losses [9]. Salimi et al. compared four differentiable losses across four synthesizers, found that the best-performing loss depends on the synthesizer [10], and later extended the comparison to out-of-domain targets [11]. In these studies, the loss function varies under a fixed search method (GD).

Evolution strategies are optimizers that need only loss values, so the loss need not be differentiable. CMA-ES maintains a Gaussian search distribution whose mean and covariance are adapted from the ranking of sampled candidates [15]. LES replaces such hand-designed update rules with a parameterized rule meta-learned over a distribution of optimization problems [16]. Cherep et al. used LES to match CLAP embeddings of text descriptions with a modular synthesizer, finding it stronger than CMA-ES and random search in their setting [19].

3. METHODOLOGY

3.1. Task and synthesizers

Each trial is an in-domain sound-matching problem. Ground-truth parameters are drawn uniformly at random in the normalized parameter space $[0, 1]^d$, the target sound (1 s at 44.1 kHz) is rendered by the synthesizer itself, and the search method must find parameters that reproduce the target. Each trial draws an independent uniform initialization; GD descends from it, while CMA-ES and LES center their initial distribution at the midpoint of $[0, 1]^d$. Because the target lies in the synthesizer's own range, an exact match exists and the true parameters are known, so methods can be compared on parameter recovery directly; prior loss-function evaluations use the same protocol [10].

We use nine Faust synthesizers: five simple programs with $d = 2$ (bandpass-filtered noise, amplitude-modulated noise, an additive sine-saw pair, a sine-modulated saw, and a chirplet) and four DX7-style FM programs (three four-operator routings with $d = 8$, and a five-operator variant with an amplitude-modulation envelope, $d = 12$), after the Yamaha DX7 [20]. FM matching is a long-standing benchmark for synthesizer parameter search [2, 13, 14]. Four of the simple programs are taken from Salimi et al. [10]; the chirplet program is a sine oscillator with exponentially rising pitch, following Vahidi et al. [9]. Table 1 lists the searched parameters of each program.

Table 1. The nine synthesizer programs. d is the number of searched parameters. In the FM programs, f_0 is the fundamental, r_i are operator frequency ratios, and l_i are operator output levels.

Program	Synthesis	d	Parameters
BP-Noise	filtered noise	2	`lp_cut`, `hp_cut`
AM-Noise	AM noise	2	`carrier`, `amp`
SineSaw	additive	2	`sine_freq`, `saw_freq`
SineModSaw	AM	2	`carrier`, `amp`
Chirplet	rising chirp	2	`increase_speed`, `starting_pitch`
DX7-A1	FM, serial stack	8	f_0, r_2–r_4, l_1–l_4
DX7-A2	FM, two carrier pairs	8	as DX7-A1
DX7-A3	FM, three modulators	8	as DX7-A1
DX7-5op	FM, serial stack + AM	12	+ r_5, l_5, `am_rate`, `am_depth`

3.2. Loss functions

Five loss functions compare candidate and target audio. Four are those evaluated by Salimi et al. [10], with identical implementations and short-time Fourier transform (STFT) parameters: `L1` and `SIMSE`, the mean absolute difference and the scale-invariant mean squared error between magnitude spectrograms; `Scat` (called JTFS by Salimi et al.), the mean absolute distance between 1-D scattering coefficients [21]; and `DTW`, a soft-DTW distance [22] ($\gamma = 1$) between smoothed amplitude envelopes. The fifth, `CLAP`, is the Euclidean distance between CLAP audio embeddings [23], previously used as a sound-matching objective with gradient-free search [19]. `CLAP` is not differentiable in our pipeline, so GD runs only on the other four. All methods evaluate the same objective: GD minimizes JAX implementations of the losses, and those used by the gradient-free methods agree with them to within floating-point rounding error (relative differences below 10^{-6}).

3.3. Search methods and budget

Our GD implementation follows Salimi et al. [10]: the Faust program is compiled to a differentiable JAX function with DawDreamer, and RMSProp minimizes the loss with learning rate 0.045 and gradients clipped to 1. Since GD is sensitive to the learning rate, we repeat all GD runs at three additional rates (0.0225, 0.01125, 0.005625); these appear only in the GD (best lr) column of Table 2, and all other analyses use the base rate. CMA-ES [15] and LES [16] (evosax implementation [17]) search directly in the normalized parameter space, with their mean initialized at its midpoint, an initial standard deviation of 0.3 (σ_0 for CMA-ES), and population sizes of $4 + \lfloor 3 \ln d \rfloor$. Random search draws points uniformly. Every method receives a budget of 200 synthesizer evaluations per trial, matching the iteration cap used by Salimi et al. [10]: each rendered candidate and each GD step counts as one evaluation

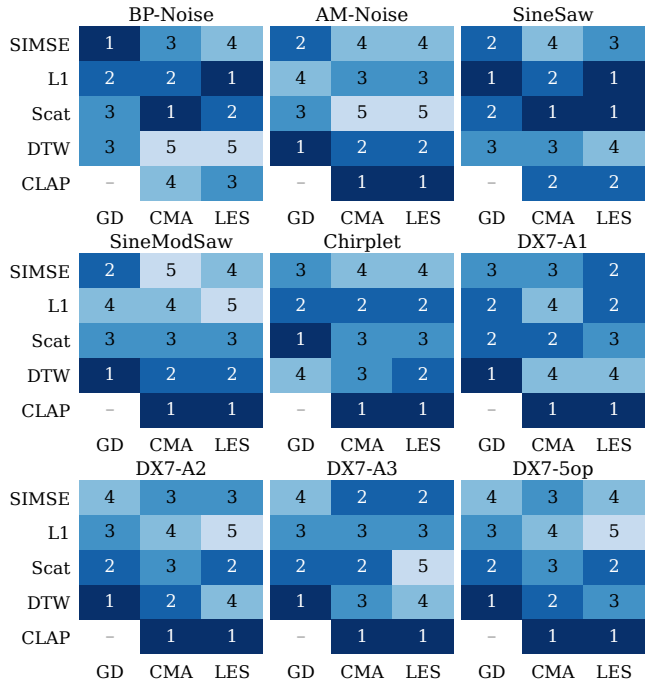


Fig. 1. NPSK rank of each loss function (1 = best, by median returned P-Loss) under each loss-guided method, per synthesizer. Losses in the same NPSK group share a rank. GD cannot use the non-differentiable `CLAP` loss (–).

(a GD step also includes a backward pass). The initialization counts as the first evaluation.

3.4. Protocol and metrics

We run 300 independent trials per (synthesizer, loss, method) configuration, each with a fresh random target and initialization. Each trial is scored at the *returned point*: the evaluated candidate with the lowest audio loss. We report two metrics at that point: the audio loss itself, and P-Loss, the Euclidean distance between the returned and true parameter vectors in normalized space. The reported P-Loss is therefore the parameter distance of the lowest-audio-loss candidate, not that of the closest point visited. We rank loss functions by P-Loss with the non-parametric Scott-Knott procedure (NPSK) on bootstrapped medians, following Salimi et al. [10]. Comparisons between methods use a Mann–Whitney test on the per-trial values, requiring $p < 10^{-3}$ and Cliff's $|\delta| \geq 0.147$, the conventional boundary between negligible and small effects [24].

4. RESULTS

All results below are computed at the returned point, the lowest-audio-loss candidate (Sec. 3.4), aggregated over 300 independent trials per (synthesizer, loss, method) configuration. P-Loss is the normalized parameter distance to the target; "audio loss" is the objective value. Random search samples the same points regardless of the loss function; only its returned candidate depends on the audio loss. Table 2 summarizes the median returned P-Loss per synthesizer and method.

Table 2. Median returned P-Loss (parameter distance of the lowest-audio-loss candidate) within the 200-evaluation budget. Each method is shown under its best loss for that synthesizer; GD (best lr) additionally takes the best of the four learning rates. Best per row in bold.

Synthesizer	GD	GD (best lr)	Random	CMA-ES	LES
BP-Noise	0.011	0.004	0.036	**0.000**	0.012
AM-Noise	0.503	0.503	0.046	**0.003**	0.024
SineSaw	0.524	0.498	**0.084**	0.126	0.113
SineModSaw	0.374	0.374	0.063	**0.026**	0.044
Chirplet	0.366	0.366	0.074	**0.043**	0.052
DX7-A1	1.201	1.086	**0.961**	0.976	1.033
DX7-A2	1.156	1.079	**0.996**	1.032	1.083
DX7-A3	1.202	1.090	**0.969**	1.012	1.065
DX7-5op	1.438	1.368	1.218	**1.188**	1.255

4.1. Loss rankings do not transfer across optimizers

For each synthesizer we rank the loss functions by the NPSK group of their median returned P-Loss, separately under each loss-guided method (GD, CMA-ES, LES); Fig. 1 shows the result. Agreement between GD and the evolution strategies is near zero: over the four losses available to all three methods, the mean Kendall τ_b between GD's ranking and CMA-ES's is $+0.13$, and between GD and LES $+0.07$, while CMA-ES and LES agree at $+0.49$ (rising to $+0.68$ when `CLAP`, which GD cannot use, is included). `CLAP` is in the top NPSK group in 14 of the 18 gradient-free cells. The two evolution strategies thus agree with each other far more than either agrees with GD, and a loss ranking established under GD carries little information for an evolution strategy. Therefore, claims that a loss function is well-suited to a synthesis task, when established under GD alone [9–11], do not necessarily generalize to other search methods.

4.2. CMA-ES leads on simple synthesizers; GD is last

Table 3 summarizes the mean rank of each method across (synthesizer, loss) cells, and Fig. 2 shows median returned P-Loss against budget on four tasks. On the simple synthesizers CMA-ES is the strongest method under both metrics (mean rank 1.2 in audio loss over the 20 `CLAP`-free cells, and the best audio loss in 22 of all 25 cells), while GD ranks last (3.65–3.70). GD's returned P-Loss typically stops improving within roughly 30 evaluations, except on the Chirplet synthesizer (Fig. 2).

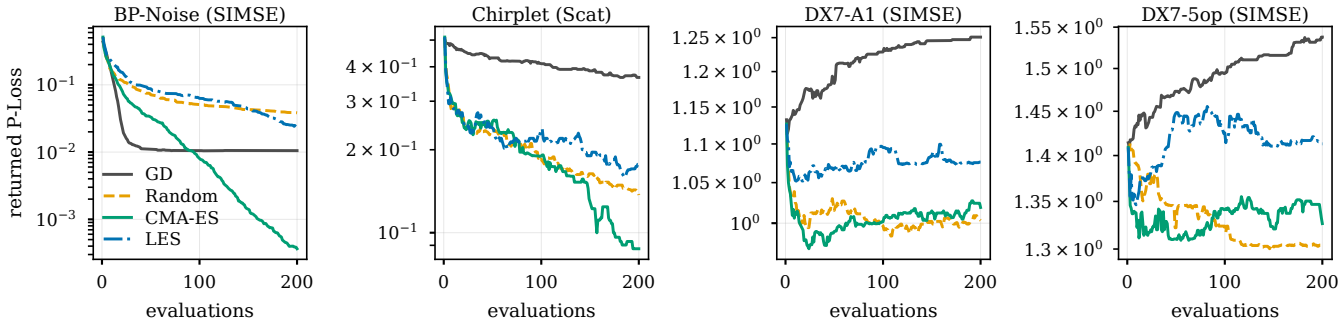


Fig. 2. Median P-Loss of the returned point as a function of budget spent, over 300 trials, for two simple synthesizers (left) and two DX7-style FM synthesizers (right), each under one loss. CMA-ES dominates the simple synthesizers. On the DX7 synthesizers GD's curve *rises*: as GD improves its audio loss, the point it would return moves farther from the true parameters.

Table 3. Mean rank of each method (1 = best) across (synthesizer, loss) cells, by median value at the returned point. `CLAP` cells are excluded so that every cell ranks all four methods.

	Simple (5 synths)		DX7 (4 synths)	
Method	audio	P-Loss	audio	P-Loss
GD	3.65	3.70	2.44	3.94
Random	3.05	2.35	2.88	**1.38**
CMA-ES	**1.20**	**1.65**	**1.94**	1.69
LES	2.10	2.30	2.75	3.00

4.3. On complex synthesizers, optimization decouples from identification

The correlation between audio loss and P-Loss weakens further on the complex synthesizers (Table 3). In audio loss, the guided methods still optimize successfully: CMA-ES attains the best mean rank (1.94). In P-Loss, they do not outperform random search (Tables 2 and 3). The best guided method, CMA-ES, is statistically indistinguishable from random search (Sec. 3.4), and GD's returned point usually lies farther from the target than its initialization. The budget is not what limits GD, which typically stalls within 30 evaluations; whether the evolution strategies would overtake random search at much larger budgets remains open. The per-trial Spearman correlation between audio loss and P-Loss falls from up to 0.94 on the simple synthesizers to $|\rho| \leq 0.30$ on the DX7 synthesizers. Sound-matching evaluations should therefore report parameter and audio metrics, and ideally include listening tests, particularly for complex synthesizers.

5. CONCLUSION

We compared four search methods across five loss functions and nine in-domain sound-matching tasks under a fixed budget of 200 synthesizer evaluations. The loss rankings produced under GD did not transfer to CMA-ES or LES: the effectiveness of a loss function is coupled to the search heuristic that optimizes it, and a ranking established under a single heuristic should not be treated as a property of the loss alone. In aggregate, both evolution strategies returned points closer to the true parameters than GD did on the simple and complex synthesizers alike, and random search proved a strong baseline: on the complex FM synthesizers no guided method returned closer points (i.e., P-Loss measured at point of best audio loss). Recent loss-function comparisons in differentiable sound-matching, including our own, have used GD alone [9–11], yet at this budget CMA-ES was the strongest guided method, with the added benefit of optimizing non-differentiable losses. Four limitations bound our claims: all tasks are in-domain, where the target is known to be reachable; we tested a single evaluation budget; P-Loss counts parameter symmetries and inaudible operators in the FM programs as errors, so DX7 parameters are only partly recoverable by any method; and we did not conduct listening tests, though the central result, that parameter-loss rankings do not transfer across heuristics, does not depend on perceptual judgments. Future evaluations of sound-matching losses should therefore include more than one search heuristic, a random-search baseline, both audio and parameter metrics, and, where feasible, listening tests.

6. REFERENCES

[1] J. Justice, "Analytic signal processing in music computation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 27, no. 6, pp. 670–684, 1979.

[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] M. Russ, *Sound synthesis and sampling*. Focal Press, 1996.

[4] N. Masuda and D. Saito, "Improving semi-supervised differentiable synthesizer sound matching for practical applications," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 863–875, 2023.

[5] 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.

[6] F. Caspe, A. McPherson, and M. Sandler, "FM tone transfer with envelope learning," in *Proceedings of the 18th International Audio Mostly Conference*, 2023, pp. 116–123.

[7] 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.

[8] B. Hayes, C. Saitis, and G. Fazekas, "Sinusoidal frequency estimation by gradient descent," in *ICASSP 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[9] C. Vahidi, H. Han, C. Wang, M. Lagrange, G. Fazekas, and V. Lostanlen, "Mesostructures: Beyond spectrogram loss in differentiable time–frequency analysis," *Journal of the Audio Engineering Society*, vol. 71, no. 9, pp. 577–585, 2023.

[10] A. Salimi, A. Hindle, and O. R. Zaïane, "Evaluating sound similarity metrics for differentiable, iterative sound-matching," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 33, pp. 4982–4994, 2025.

[11] A. Salimi, D. Penner, K. Eng, A. Hindle, and O. R. Zaïane, "Evaluating loss functions in differentiable out-of-domain sound-matching with partial parameter distance," *arXiv preprint arXiv:2608.27698*, 2026.

[12] J. Riionheimo and V. Välimäki, "Parameter estimation of a plucked string synthesis model using a genetic algorithm with perceptual fitness calculation," *EURASIP Journal on Applied Signal Processing*, vol. 2003, no. 8, pp. 791–805, 2003.

[13] T. J. Mitchell and D. P. Creasey, "Evolutionary sound matching: A test methodology and comparative study," in *Sixth International Conference on Machine Learning and Applications (ICMLA 2007)*. IEEE, 2007, pp. 229–234.

[14] M. J. Yee-King, L. Fedden, and M. d'Inverno, "Automatic programming of VST sound synthesizers using deep networks and other techniques," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 150–159, 2018.

[15] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.

[16] R. T. Lange, T. Schaul, Y. Chen, T. Zahavy, V. Dalibard, C. Lu, S. Singh, and S. Flennerhag, "Discovering evolution strategies via meta-black-box optimization," in *International Conference on Learning Representations (ICLR)*, 2023.

[17] R. T. Lange, "evosax: JAX-based evolution strategies," in *Proceedings of the Companion Conference on Genetic and Evolutionary Computation*, 2023, pp. 659–662.

[18] J. Shier, G. Tzanetakis, and K. McNally, "SpiegeLib: An automatic synthesizer programming library," in *148th Audio Engineering Society Convention*, May 2020.

[19] M. Cherep, N. Singh, and J. Shand, "Creative text-to-audio generation via synthesizer programming," in *International Conference on Machine Learning*. PMLR, 2024, pp. 8270–8285.

[20] F. Caspe, A. McPherson, and M. Sandler, "DDX7: Differentiable FM synthesis of musical instrument sounds," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 608–616.

[21] M. Andreux, T. Angles, G. Exarchakis, R. Leonarduzzi, G. Rochette, L. Thiry, J. Zarka, S. Mallat, J. Andén, E. Belilovsky, J. Bruna, V. Lostanlen, M. Chaudhary, M. J. Hirn, E. Oyallon, S. Zhang, C. Cella, and M. Eickenberg, "Kymatio: Scattering transforms in Python," *Journal of Machine Learning Research*, vol. 21, no. 60, pp. 1–6, 2020.

[22] M. Cuturi and M. Blondel, "Soft-DTW: A differentiable loss function for time-series," in *International Conference on Machine Learning*. PMLR, 2017, pp. 894–903.

[23] 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 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[24] J. Romano, J. D. Kromrey, J. Coraggio, and J. Skowronek, "Appropriate statistics for ordinal level data: Should we really be using t-test and Cohen's d for evaluating group differences on the NSSE and other surveys?" in *Annual Meeting of the Florida Association of Institutional Research*, 2006.