Multi-Scale Spectral Loss Revisited

Simon Schwär and Meinard Müller, *Fellow, IEEE*

***Abstract*—The Multi-Scale Spectral (MSS) loss is commonly used for comparing audio signals, as it provides a good trade-off between temporal and spectral resolution. However, some configuration choices, including window type and size, magnitude compression, as well as the distance between spectrograms, are often set implicitly, even though they can significantly impact the loss properties and the convergence of trained models. Particularly in the context of differentiable digital signal processing (DDSP), where learned parameters may explicitly control the frequency of synthesis components, the MSS loss often fails to provide informative gradients. The main goal of this letter is to gain a better understanding of how different configurations of the MSS loss affect this problem. As an illustrative example, we analyze the task of sinusoid frequency estimation via gradient descent to compare different configurations and their effect on the loss properties. Furthermore, we show that favorable configurations can also facilitate unsupervised training of a more complex DDSP additive synthesis autoencoder. Our results indicate that a careful configuration may benefit many applications where the MSS loss is utilized.**

***Index Terms*—Audio-to-audio distances, audio synthesis, differentiable DSP, loss functions.**

I. Introduction

A MULTITUDE of machine learning tasks require a loss function to compare audio signals, including many end-to-end approaches for sound, music and speech synthesis. *Spectral* loss functions are among the most commonly used distances between audio signals and rely on an element-wise comparison of spectrograms, which can be computed from time-domain signals using the short-time Fourier transform (STFT). This way, signals are compared in terms of the temporal and spectral distribution of signal energy, which better correlates with human perception than, for example, the numerical similarity of waveforms [1]. The spectrogram, however, is limited by the fundamental trade-off between time and frequency resolution of the STFT and thus—without phase information—cannot achieve high temporal and spectral accuracy at the same time. This trade-off can be mitigated by comparing multiple spectrograms with different time–frequency resolutions in a combined loss function [2], [3], so that signals must conform at all these resolutions simultaneously to minimize the loss.

Manuscript received 24 August 2023; revised 20 October 2023; accepted 13 November 2023. Date of publication 15 November 2023; date of current version 4 December 2023. This work was supported by German Research Foundation under Grant DFG MU 2686/13-2 (No. 401198673). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Federico Fontana. *(Corresponding author: Simon Schwär.)*

The authors are with the International Audio Laboratories (a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS), 91058 Erlangen, Germany (e-mail: simon.schwaer@audiolabs-erlangen.de; meinard.mueller@audiolabs-erlangen.de).

Digital Object Identifier 10.1109/LSP.2023.3333205

This concept of a *Multi-Scale Spectral* (MSS) loss is used extensively in the context of *Differentiable Digital Signal Processing* (DDSP) [4], [5]. DDSP was introduced as an umbrella term for the concept of back-propagating gradients through fixed DSP components, which allows for including domain knowledge (e.g., about the physics of a generative process) in model architectures to restrict them in a meaningful way (*inductive bias*). The DDSP paradigm has recently proven useful for various tasks in audio signal processing and music information retrieval, including fundamental frequency estimation [6], musical source separation [7], as well as estimating parameters for piano [8] and singing voice synthesis [9] or artificial reverberation [10]. All these applications use a variant of the MSS loss, but in most approaches, certain DSP parameters cannot be estimated simply by comparing the target and output audio signals. In particular, the MSS loss has been shown to be highly irregular and non-convex for parameters that directly or indirectly control the frequency of tonal synthesis components [11], [12], [13]. This way, optimization methods like stochastic gradient descent are unlikely to converge without additional means like self-supervision (e.g. [6], [14]) or external pitch estimation (e.g. [4], [7]) that increase the overall complexity of the system. While randomizing configurations of the STFT has been proposed to improve training robustness [15], to our knowledge, no systematic analysis of the influence of different configurations on the loss behavior has been presented.

In this letter, we show that, in certain situations, the MSS loss is able to provide gradients that allow for convergence to the true frequency parameter of a sinusoid (we call these *informative* gradients in the following), and that a significant part of both the favorable and the unfavorable loss characteristics can be ascribed to the effects of spectral leakage. Some loss configurations may amplify unfavorable aspects, so that e.g. the choice of window type and size, magnitude compression or spectrogram distance can influence convergence behavior. To illustrate the differences between configurations, we consider the simple scenario of sinusoidal parameter estimation via gradient descent, where we can explicitly analyze the *loss landscape* for the frequency parameter. From this analysis, we derive three example configurations and compare their performance in an unsupervised training setup using a DDSP additive synthesis autoencoder [6]. These experiments provide evidence for spectral leakage as a possible underlying cause of the MSS loss' failure to provide informative gradients for frequency parameters.

II. Definitions & Experimental Setup

An autoencoder consists of an encoder $\mathcal{E} : \mathcal{X} \to \mathcal{Z}$ and a decoder $\mathcal{D} : \mathcal{Z} \to \mathcal{X}$ which are chosen so that $\mathcal{D}(\mathcal{E}(x)) \approx x$ for all $x \in \mathcal{X}$. Often, $\mathcal{E}$ and $\mathcal{D}$ are NNs that jointly learn a suitable encoding and decoding by minimizing a loss function $\mathcal{L} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ between x and $\hat{x} = \mathcal{D}(\mathcal{E}(x))$ over a training

TABLE I
CONSIDERED MSS LOSS CONFIGURATION CHOICES

Configuration	Value	Description
Window Type	WR	Rectangular window
	WH	Hann window
	WF	Flat Top window
Window Size(s)	S1	$\mathcal{N} = \{64\}$
	S2	$\mathcal{N} = \{512\}$
	S3	$\mathcal{N} = \{2048\}$
	S4	$\mathcal{N} = \{64, 128, 256, 512, 1024, 2048\}$
	S5	$\mathcal{N} = \{67, 127, 257, 509, 1021, 2053\}$
Magnitude Compression	C0	$\mathcal{P} = \{x\}$
	C1	$\mathcal{P} = \{\log(x+\varepsilon)\}$, $\varepsilon = 10^{-7}$
	C2	$\mathcal{P} = \{\log(1+\gamma x)\}$, $\gamma = 1$
	C3	$\mathcal{P} = \{20\log_{10}(x+\varepsilon)\}$, $\varepsilon = 10^{-7}$
	C4	$\mathcal{P} = \{x,\ \log(x+\varepsilon)\}$, $\varepsilon = 10^{-7}$
Matrix Distance	D1	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \|\mathcal{Y} - \hat{\mathcal{Y}}\|_1$
	D2	$d(\mathcal{Y}, \hat{\mathcal{Y}}) = \|\mathcal{Y} - \hat{\mathcal{Y}}\|_2^2$

dataset. This minimization is typically achieved by a form of stochastic gradient descent on $\mathcal{L}(x, \hat{x})$.

A. Multi-Scale Spectral Loss

If $\mathcal{X} = \mathbb{R}^L$ is the space of real discrete audio signals of length L, the MSS loss is a popular choice for $\mathcal{L}(x, \hat{x})$. This loss function aggregates the distance between multiple spectrograms with specified window types, window sizes, and magnitude compressions to achieve a high temporal and spectral accuracy without enforcing phase coherence of the compared signals. Let

$$\mathcal{Y}_{w,N,p}(m,k) = p\left(\left|\sum_{n=0}^{N-1} x[n+mH]w[n]\exp\left(\frac{-i2\pi kn}{N}\right)\right|\right) \quad (1)$$

be the spectrogram of x, where $N \in \mathbb{N}$ is the window size in samples, $H \in \mathbb{N}$ is the hop size in samples (we set $H = \lfloor N/2 \rfloor$ throughout all experiments for simplicity, but generally, arbitrary hop sizes can be used), $w \in \mathbb{R}^N$ is a discrete window function, and $p : \mathbb{R}_+ \to \mathbb{R}$ is an (optional) magnitude compression function. Each time–frequency coefficient can be accessed with the time index $m \in [0 : M-1]$ with $M = L/H$, assuming for simplicity that L is a multiple of H, and frequency index $k \in [0 : K-1]$ with $K = \lceil (N+1)/2 \rceil$, discarding negative frequencies since spectra of real signals are symmetric. To further simplify notation, we consider $\mathcal{Y}_{N,w,p} \in \mathbb{R}_+^{(K \times M)}$ to be a matrix. Analogously, we denote the spectrogram matrix of $\hat{x}$ by $\hat{\mathcal{Y}}_{w,N,p}$.

With this, a generalized MSS loss can be defined as

$$\mathcal{L}_{\text{MSS}}(x, \hat{x}) := \sum_{N \in \mathcal{N}} \sum_{p \in \mathcal{P}} d(\mathcal{Y}_{w,N,p}, \hat{\mathcal{Y}}_{w,N,p}), \quad (2)$$

where w is a window function as defined above, $\mathcal{N}$ is a set of suitable window sizes, $\mathcal{P}$ is a set of suitable compression functions, and $d(\cdot, \cdot)$ is a distance between two matrices. This formulation allows for many different configurations of w, $\mathcal{N}$, $\mathcal{P}$, and d. In Table I, we introduce a coding scheme for the configurations used in our experiments. As an example, the "original" MSS loss proposed in [4] is attained by the configuration (WH, S4, C4, D1). The small value ε in C1, C3, and C4 is used to avoid taking the logarithm of zero. The parameter $\gamma \in \mathbb{R}_+$ in C2 can be used to control the strength of compression [16]. We set $\gamma = 1$ for comparability with C1.

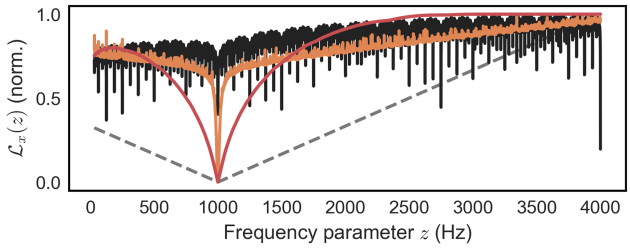

Fig. 1. Loss landscapes $\mathcal{L}_x(z)$ (normalized) with different MSS configurations. Black: Original MSS (WH, S4, C4, D1). Orange: Modified Hann MSS (WH, S5, C4, D2). Red: Smooth MSS (WF, S5, C2, D2). The grey dashed line shows an ideal convex loss, the absolute difference between z and f_{tgt}.

B. Sinusoidal Frequency Estimation

As a defining property of a DDSP autoencoder, $\mathcal{D}$ becomes a fixed mapping from a latent parameter space $\mathcal{Z}$ to $\mathcal{X}$, while only the encoder $\mathcal{E}$ is an NN with learnable parameters. The training objective for $\mathcal{E}$ given a fixed target signal x can thus be rephrased as a loss function $\mathcal{L}_x : \mathcal{Z} \to \mathbb{R}$ defined by

$$\mathcal{L}_x(z) = \mathcal{L}(x, \mathcal{D}(z)) \quad (3)$$

for an encoder output $z = \mathcal{E}(x)$. As opposed to a general autoencoder, the fixed properties of $\mathcal{D}$ for a given z considerably influence the convergence of $\mathcal{E}$ to a minimizer of $\mathcal{L}$.

Typically (see e.g. Section V), $\mathcal{D}$ is a non-trivial mapping with many control parameters which all have to be learned jointly. In the following, we consider a simpler but illustrative scenario where $\mathcal{D}$ is a single sinusoidal oscillator with fixed amplitude $A = 1$ that maps a single frequency parameter $z \in \mathbb{R}$ to an output signal $\hat{x}$ with

$$\hat{x}[n] = A\sin(2\pi z n / F_{\text{s}}), \quad (4)$$

for all $n \in [0 : L-1]$, where $F_{\text{s}} = 16000$ Hz is the sampling rate used in our experiments. We further assume that the target signal x is also generated according to (4), with frequency $f_{\text{tgt}} = 1000$ Hz. In this setting, we can visualize the *loss landscape* $\mathcal{L}_x(z)$ for different $\mathcal{L}_{\text{MSS}}$ as shown in Fig. 1. Particularly the original MSS loss (in black) appears to be very noisy and thus provide uninformative gradients $\mathrm{d}\mathcal{L}_x/\mathrm{d}z$ to find the optimal value $z = f_{\text{tgt}}$. Understanding the causes for this loss behavior and the differences between configurations in Fig. 1 is a main goal of this letter.

III. SPECTRAL LEAKAGE

The truncation of x and $\hat{x}$ in (1) leads to a "blurred" spectral representation of the sinusoids, since multiplication with a window function w in time domain is equivalent to convolution in frequency domain. Spectra of finite-length windows have multiple local maxima separated by zeros [17] that can be differentiated into a *mainlobe* (the central maximum around the sinusoid frequency up to the first zero on both sides) and *sidelobes* (all other local maxima). This effect of *spectral leakage* is illustrated in Fig. 2, showing the influence of different configuration choices on the spectrum of a windowed excerpt of x (in black) and $\hat{x}$ with an arbitrarily chosen z (in light

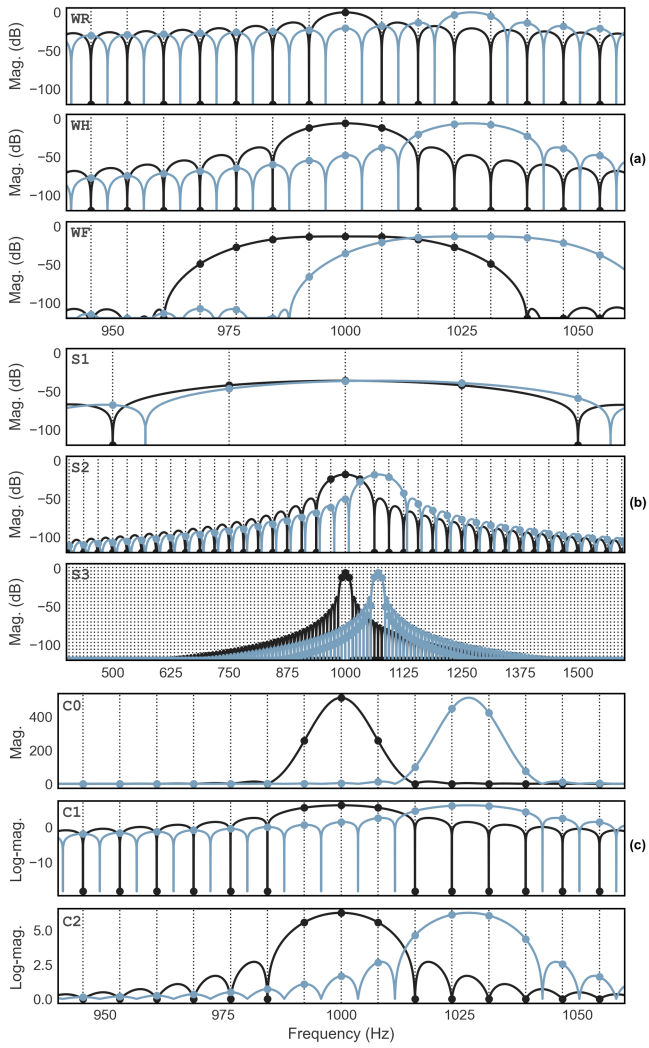

Fig. 2. Influence of **(a)** window type, **(b)** window size, and **(c)** magnitude compression on the spectrum of x (black) and $\hat{x}$ (light blue). Circles denote DFT coefficients and lines the approximate continuous spectra. In (a) and (b), some coefficient values are below the y-axis range, indicated by half circles. The default configuration is (`WH`, `S3`, `C3`) unless otherwise specified.

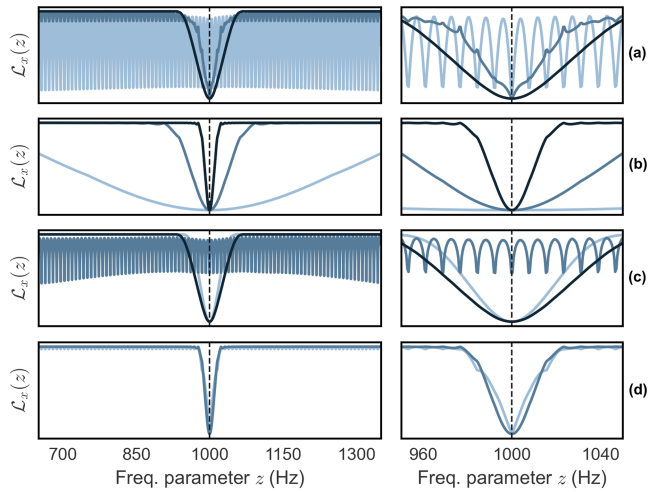

Fig. 3. Influence on $\mathcal{L}_x(z)$ of **(a)** window type (light: `WR`, medium: `WH`, dark: `WF`) compared using (`S3`, `C2`, `D2`), **(b)** window size (light: `S1`, medium: `S2`, dark: `S3`) compared using (`WH`, `C0`, `D2`), **(c)** magnitude compression (light: `C0`, medium: `C1`, dark: `C2`) compared using (`WF`, `S3`, `D2`), and **(d)** matrix distance (light: `D1`, medium: `D2`) compared using (`WH`, `S3`, `C0`).

blue). Each plot depicts the discrete Fourier transform (DFT) bin frequencies as vertical lines and the DFT coefficient values as circles. It further shows the (approximate) continuous spectra of the sinusoids, illustrating that they are shifted versions of the symmetric window spectra centered around f_{tgt} and z. The DFT bin frequencies form a "rigid sampling grid" on the frequency axis, so that all coefficient values change when the window spectrum is shifted relative to the grid. In our example, f_{tgt} is equal to a DFT bin frequency, so that most DFT coefficients coincide with a zero of the window spectrum, while z lies between two bins and the coefficients are non-zero. This can lead to large numerical differences between the two DFT spectra, especially when sidelobes are prominent.

IV. Sinusoid Frequency Estimation

Without any blurring of the sinusoid spectra, $\mathcal{L}_x(z)$ would not provide informative gradients at all. In our example, $\mathrm{d}\mathcal{L}_x/\mathrm{d}z$ would be zero when the spectral peaks do not overlap, since $\mathcal{L}_{\text{MSS}}$ depends on the element-wise difference between spectra (i.e., the vertical distance between circles in Fig. 2). However, an ideal "kernel" for blurring has one unique maximum, since in this case $\mathcal{L}_x(z)$ could only be reduced when z moves closer to f_{tgt} and not by other changes of z relative to the DFT bins. In other words, when choosing a suitable $\mathcal{L}_{\text{MSS}}$ configuration for frequency estimation via gradient descent, we aim for a wide mainlobe and numerically negligible sidelobes. Fig. 3 illustrates $\mathcal{L}_x(z)$ with different configurations for $\mathcal{L}_{\text{MSS}}$, which we will discuss in the following.

A. Window Type

The window spectra of three different window types—Rectangular (`WR`), Hann (`WH`), and Flat Top (`WF`)—are compared in Fig. 2(a) with an otherwise fixed configuration (`S3`, `C2`, `D2`). Many considerations influence window choices in practice [17, Ch. 5.3.3], since for example narrower windows have a higher *sensitivity* (i.e., better ability to detect sinusoids in noise), while wider windows have a better *dynamic range* (i.e., sidelobes of a loud sinusoid are less likely to mask a weaker sinusoid). For frequency estimation, the sidelobe level is a central property that influences the behavior of $\mathcal{L}_x(z)$. Using `WR` (and to a lesser extent `WH`) leads to strong periodic fluctuations of $\mathcal{L}_x(z)$ with a period of F_{s}/N (see Fig. 3(a)), due to the changes in spectral leakage depending on the relative value of z compared to the DFT bin frequencies. The low sidelobe levels of `WF` result in a smoother loss landscape with a unique local minimum at $z = f_{\text{tgt}}$.

B. Window Size

The width of mainlobe and sidelobes is also influenced by the window size, as illustrated in Fig. 2(b), using fixed (`WH`, `C0`, `D2`). The choice of a suitable set of window sizes was originally motivated by the resolution of amplitude comparisons, where a small window size leads to a better time resolution and a worse frequency resolution. For frequency estimation, the width of the locally convex sections depend on the maximal difference between f_{tgt} and z where the mainlobes still overlap, so that

TABLE II
GRADIENT-SIGN RANKING ACCURACY FOR SELECTED CONFIGURATIONS

Configuration	GRA			
	0.3 ct	3 ct	30 ct	300 ct
original (`WH`, `S4`, `C4`, `D1`)	0.523	0.529	0.573	0.755
mod. Hann (`WH`, `S5`, `C4`, `D2`)	0.613	0.635	0.708	**0.923**
smooth (`WF`, `S5`, `C2`, `D2`)	**0.999**	**0.993**	**0.952**	0.860

the convex section is wider for smaller window sizes. Larger windows, conversely, increase the loss' ability to discriminate sinusoids with similar frequencies.

C. Magnitude Compression

Compression enables a comparison of magnitudes over a wide dynamic range and is often perceptually motivated. However, it may also exacerbate the problem of periodically changing spectral leakage by decreasing the relative difference between mainlobe and sidelobe levels (see Fig. 2(c)). The simple logarithmic compression (`C1`) as used in [4] leads to large negative values at the zeroes of the window function bounded below by $\log(\varepsilon)$. This amplifies the periodic behavior of $\mathcal{L}_x(z)$ as shown in Fig. 3(c) with fixed (`WF`, `S3`, `D2`). To mitigate this issue, we propose to replace ε with 1 (`C2`), so that the compressed value is always larger or equal to 0. A factor $\gamma \geq 0$ can further be chosen to adjust the compression strength.

D. Spectrum Norm

We compare two distances between the spectrogram matrices, the ℓ^1 norm (`D1`) and the squared ℓ^2 norm (`D2`) in Fig. 3(d) with fixed (`WH`, `S3`, `C0`). Other considerations like outlier sensitivity are often relevant for choosing a distance, but `D1` slightly amplifies the periodic fluctuations in $\mathcal{L}_x(z)$.

E. Considered MSS Loss Configurations

Fig. 1 shows $\mathcal{L}_x(z)$ with three different configurations for $\mathcal{L}_{\text{MSS}}$, which we chose not to represent a "best" configuration, but to illustrate how different choices affect the loss landscape. In addition to this qualitative comparison, Table II shows the Gradient-Sign Ranking Accuracy (GRA) [11] for these loss landscapes, specifying how often on average $\mathcal{L}_x(z)$ decreases when moving towards a random f_{tgt} by c cents from a random initial frequency z_0. While it is not an analytic evaluation of the loss, a larger GRA indicates that $\mathcal{L}_x(z)$ tends to provide informative gradients, while a value of 0.5 suggests that changes in $\mathcal{L}_x(z)$ are random. We calculate the GRA for step sizes c of 0.3, 3, 30, and 300 cents with all other settings as in [11]. The *original* MSS results in a GRA near 0.5 for smaller step sizes, so that gradient descent algorithms are unlikely to converge. In fact, for noise-free signals, this loss behavior is entirely a result of spectral leakage at different window sizes, amplified by the logarithmic compression `C1`. The *modified Hann* MSS uses `D2` and `S5`, where all window sizes are prime instead of powers of two. This way, z does not coincide with the DFT bin frequencies of multiple window sizes at the same time, which reduces the amplitude of the fluctuations in $\mathcal{L}_x(z)$. The modified Hann MSS achieves a high GRA for $c = 300$ cents while some artifacts still impact the GRA for smaller step sizes. *Smooth* MSS results in the fewest spectral leakage artifacts by using `WF` and `C2`, while also having the widest mainlobe overlaps. The high GRA values for small step sizes down to 0.3 cents indicate that this configuration also approximates local convexity with a unique minimum in these synthetic conditions. However, performance decreases for the largest step size due to vanishing gradients when the mainlobes do not overlap.

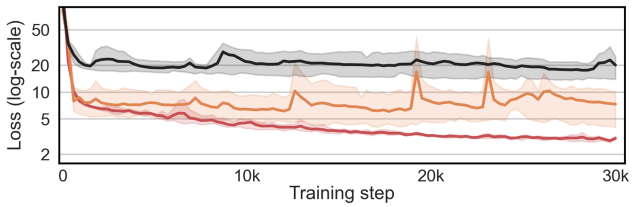

Fig. 4. Training loss for F^{θ}_{sin} with original MSS (black), mod. Hann MSS (orange), and smooth MSS (red), with mean and variance of three runs each.

V. UNSUPERVISED DDSP AUTOENCODER

The simple scenario above does not consider other influences on the loss like noise or mixed sinusoids with varying amplitudes. To investigate differences between the configurations from Table II in a more practically relevant scenario, we repeat a DDSP autoencoder experiment using the encoder (F^{θ}_{sin}) and sinusoidal synthesizer (S_{sin}) from [6] trained with the NSynth dataset [18], which consists of harmonic single notes with natural background noise and transients. The task of F^{θ}_{sin} is to estimate time-varying parameters for 100 sinusoids that are then synthesized by S_{sin}. Instead of relying on self-supervision as in [6], we use only a reconstruction loss ($\mathcal{L}_{\text{MSS}}$ with the respective configuration), resulting in fully unsupervised training. For comparable loss magnitudes, we multiply each $\mathcal{L}_{\text{MSS}}$ with an empirically estimated constant. The training loss with the different $\mathcal{L}_{\text{MSS}}$ configurations is shown in Fig. 4. With all other settings as in [6], only smooth MSS leads to consistent convergence of F^{θ}_{sin}.

To evaluate whether the model also learns to estimate meaningful parameters, we conduct two experiments. First, we create 1000 synthetic test signals of one second length with a random constant fundamental frequency (F0) between 30 and 800 Hz and 10 integer harmonics with random amplitude. F^{θ}_{sin} trained on NSynth with smooth MSS estimates the true frequencies with a mean error of 27±35 cents (original: 838±647 cents, mod. Hann: 1467±932 cents). Second, we evaluate the similarity between the output of S_{sin} and an input signal from MDB-melody-synth [19]. For this, we estimate the F0 of the output signal using CREPE [20] and compare it with the reference F0 from the dataset. Since CREPE relies on salient frequency components in its input, it would yield dissimilar F0 estimates if the output signal contained strong erroneous components. The raw pitch accuracy [21] for this comparison is 0.81±0.09 for F^{θ}_{sin} trained with smooth MSS (original: 0.01±0.03, mod. Hann: 0.27±0.15). While these preliminary experiments suggest that differences between configurations are also relevant for complex scenarios, further research is needed to fully understand the loss behavior in practice.

VI. CONCLUSION

Our results indicate that the properties of the MSS loss for frequency estimation considerably depend on the numerical relation between mainlobe and sidelobes in the compared spectra.

REFERENCES

[1] A. R. Müller, "Frequency analysis in the peripheral auditory system," in *Auditory Physiology*. Orlando, FL, USA: Academic Press, 1983, pp. 191–249.

[2] X. Wang, S. Takaki, and J. Yamagishi, "Neural source-filter waveform models for statistical parametric speech synthesis," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 28, pp. 402–415, 2020.

[3] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel WaveGAN: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 2020, pp. 6199–6203.

[4] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable digital signal processing," in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=B1x1ma4tDr

[5] B. Hayes, J. Shier, G. Fazekas, A. McPherson, and C. Saitis, "A review of differentiable digital signal processing for music & speech synthesis," *Comput. Res. Repository*, 2023, *arXiv:2308.15422*.

[6] J. Engel, R. Swavely, L. H. Hantrakul, A. Roberts, and C. Hawthorne, "Self-supervised pitch detection by inverse audio synthesis," in *Proc. Int. Conf. Mach. Learn. Workshop Self-supervision Audio Speech*, 2020.

[7] K. Schulze-Forster, C. S. J. Doire, G. Richard, and R. Badeau, "Unsupervised music source separation using differentiable parametric source models," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 1276–1289, 2023, doi: 10.1109/TASLP.2023.3252272.

[8] L. Renault, R. Mignot, and A. Roebel, "Differentiable piano model for MIDI-to-audio performance synthesis," in *Proc. 25th Int. Conf. Digit. Audio Effects*, 2022, pp. 232–239.

[9] D.-Y. Wu et al., "DDSP-based singing vocoders: A new subtractive-based synthesizer and a comprehensive evaluation," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2022, pp. 76–83.

[10] S. Lee, H.-S. Choi, and K. Lee, "Differentiable artificial reverberation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 2541–2556, 2022.

[11] J. Turian and M. Henry, "I'm sorry for your loss: Spectrally-based audio distances are bad at pitch," in *Proc. Adv. Neural Inf. Process. Syst.,*2020.

[12] B. Hayes, C. Saitis, and G. Fazekas, "Sinusoidal frequency estimation by gradient descent," in *Proc. IEEE Int. Conf. Acoustics, Speech Signal Process.*, Rhodes Island, Greece, 2023, pp. 1–5, doi: 10.1109/ICASSP49357.2023.10095188.

[13] F. Caspe, A. McPherson, and M. Sandler, "DDX7: Differentiable FM synthesis of musical instrument sounds," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2022, pp. 608–616.

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

[15] C. J. Steinmetz and J. D. Reiss, "auraloss: Audio focused loss functions in PyTorch," in *Proc. Digit. Music Res. Netw. One-Day Workshop*, 2020.

[16] M. Müller, *Fundamentals of Music Processing–Using Python and Jupyter Notebooks*, 2nd ed. Cham: Springer Verlag, 2021.

[17] K. M. M. Prabhu, *Window Functions and Their Applications in Signal Processing*. Boca Raton, FL, USA: CRC/Taylor & Francis, 2014.

[18] J. Engel et al., "Neural audio synthesis of musical notes with wavenet autoencoders," *Comput. Res. Repository*, 2017, pp. 1068–1077.

[19] J. Salamon, R. M. Bittner, J. Bonada, J. J. B. Vicente, E. Gémez, and J. P. Bello, "An analysis/synthesis framework for automatic f0 annotation of multitrack datasets," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2017, pp. 71–78.

[20] J. W. Kim, J. Salamon, P. Li, and J. P. Bello, "CREPE: A convolutional representation for pitch estimation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 161–165.

[21] C. Raffel et al., "MIR_EVAL: A transparent implementation of common MIR metrics," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2014, pp. 367–372.