OPEN ACCESS Freely available online

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, (2023 September).
DOI: https://doi.org/10.17743/jaes.2022.0103

Mesostructures: Beyond Spectrogram Loss in Differentiable Time–Frequency Analysis

CYRUS VAHIDI,[1] HAN HAN,[2] CHANGHONG WANG,[2] MATHIEU LAGRANGE,[2]
(c.vahidi@qmul.ac.uk) (han.han@ls2n.fr) (changhong.wang@ls2n.fr) (mathieu.lagrange@ls2n.fr)

GYÖRGY FAZEKAS,[1] AND VINCENT LOSTANLEN[2]
(george.fazekas@qmul.ac.uk) (vincent.lostanlen@ls2n.fr)

[1]*Centre for Digital Music, Queen Mary University of London, London, UK*
[2]*Nantes Université, École Centrale Nantes, Centre National de la Recherche Scientifique (CNRS), Laboratoire des Sciences du Numérique de Nantes (LS2N), UMR 6004, F-44000 Nantes, France*

Computer musicians refer to mesostructures as the intermediate levels of articulation between the microstructure of waveshapes and the macrostructure of musical forms. Examples of mesostructures include melody, arpeggios, syncopation, polyphonic grouping, and textural contrast. Despite their central role in musical expression, they have received limited attention in recent applications of deep learning to the analysis and synthesis of musical audio. Currently, autoencoders and neural audio synthesizers are only trained and evaluated at the scale of microstructure, i.e., local amplitude variations up to 100 ms or so. In this paper, the authors formulate and address the problem of mesostructural audio modeling via a composition of a differentiable arpeggiator and time-frequency scattering. The authors empirically demonstrate that time–frequency scattering serves as a differentiable model of similarity between synthesis parameters that govern mesostructure. By exposing the sensitivity of short-time spectral distances to time alignment, the authors motivate the need for a time-invariant and multiscale differentiable time–frequency model of similarity at the level of both local spectra and spectrotemporal modulations.

0 INTRODUCTION

0.1 Differentiable Time–Frequency Analysis

Time–frequency representations (TFRs) such as the short-time Fourier transform (STFT) or constant-Q transform (CQT) play a key role in music signal processing [1, 2] because they can demodulate the phase of slowly varying complex tones. As a consequence, any two sounds $\boldsymbol{x}$ and $\boldsymbol{y}$ with equal TFR magnitudes (i.e., spectrograms) are heard as the same by human listeners, even though the underlying waveforms may differ. For this reason, spectrograms can not only serve for visualization, but also for similarity retrieval. Denoting the spectrogram operator by $\boldsymbol{\Phi}$, the Euclidean distance $\|\boldsymbol{\Phi}(\boldsymbol{y}) - \boldsymbol{\Phi}(\boldsymbol{x})\|_2$ is much more informative than the waveform distance $\|\boldsymbol{y} - \boldsymbol{x}\|_2$, because the waveform distance diverges quickly even when phase differences are small.

In recent years, existing algorithms for STFT and CQT have been ported to deep learning frameworks such as PyTorch, TensorFlow, MXNet, and JAX [3–5]. By doing so, the developers have taken advantage of the paradigm of differentiable programming, defined as the ability to compute the gradient of mathematical functions by means of reverse-mode automatic differentiation. In the context of audio processing, differentiable programming may serve to train a neural network for audio encoding, decoding, or both. Hence, the umbrella term may be coined *differentiable time–frequency analysis* (DTFA) to describe an emerging subfield of deep learning in which stochastic gradient descent involves a composition of neural network layers as well as TFR. Previously, TFR were largely restricted to analysis frontends, but now play an integral part in learning architectures for audio generation.

The simplest example of DTFA is autoencoding. Given an input waveform $\boldsymbol{x}$, the autoencoder is a neural network architecture f with weights $\mathbf{W}$, which returns another waveform $\boldsymbol{y}$ [6, 7]. During training, the neural network f_W aims to minimize the following loss function:

$$\mathcal{L}_{\boldsymbol{x}}(\mathbf{W}) = \|(\boldsymbol{\Phi} \circ f_W)(\boldsymbol{x}) - \boldsymbol{\Phi}(\boldsymbol{x})\|_2, \tag{1}$$

on average over every sample $\boldsymbol{x}$ in an unlabeled dataset. The function above is known as *spectrogram loss* because $\boldsymbol{\Phi}$ maps $\boldsymbol{x}$ and $\boldsymbol{y}$ to the time–frequency domain.

Another example of DTFA is found in audio restoration. This time, the input of f_W is not $\boldsymbol{x}$ itself but some degraded version $h(\boldsymbol{x})$—noisy or bandlimited, for example [8, 9]. The goal of f_W is to invert the degradation operator $\boldsymbol{h}$ by producing a restored sound $(f_W \circ \boldsymbol{h})(\boldsymbol{x})$, which is close to $\boldsymbol{x}$ in terms of spectrogram loss:

$$\mathcal{L}_{\boldsymbol{x}}(\mathbf{W}) = \|(\boldsymbol{\Phi} \circ f_W \circ h)(\boldsymbol{x}) - \boldsymbol{\Phi}(\boldsymbol{x})\|_2. \tag{2}$$

Thirdly, DTFA may serve for sound matching, also known as synthesizer parameter inversion [6, 10, 11]. Given a parametric synthesizer $\boldsymbol{g}$ and an audio query $\boldsymbol{x}$, this task consists in retrieving the parameter setting $\boldsymbol{\theta}$ such that $\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{\theta})$ resembles $\boldsymbol{x}$. In practice, sound matching may be trained on synthetic data by sampling $\boldsymbol{\theta}$ at random, generating $\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{\theta})$, and measuring the spectrogram loss between $\boldsymbol{x}$ and $\boldsymbol{y}$:

$$\mathcal{L}_{\boldsymbol{\theta}}(\mathbf{W}) = \|(\boldsymbol{\Phi} \circ \boldsymbol{g} \circ f_W \circ \boldsymbol{g})(\boldsymbol{\theta}) - (\boldsymbol{\Phi} \circ \boldsymbol{g})(\boldsymbol{\theta})\|_2. \tag{3}$$

0.2 Shortcomings of Spectrogram Loss

Despite its proven merits for generative audio modeling, spectrogram loss suffers from counterintuitive properties when events are unaligned in time or pitch [12]. Although a low spectrogram distance implies a judgment of high perceptual similarity, the converse is not true: one can find examples in which $\boldsymbol{\Phi}(\boldsymbol{x})$ is far from $\boldsymbol{\Phi}(\boldsymbol{y})$ yet judged musically similar by a human listener. First, $\boldsymbol{\Phi}$ is only sensitive to time shifts up to the scale T of the spectrogram window, i.e., around 10–100 ms. The authors exemplify this in Fig. 3 with a visualization of a multi-scale spectrogram's (MSS) loss surface under time-shifts. In the case of autoencoding, if $f_W(\boldsymbol{x})(t) = \boldsymbol{x}(t - \tau)$ with $\tau \gg T$, $\mathcal{L}_{\boldsymbol{x}}(\mathbf{W})$ may be as large as $2\|\boldsymbol{\Phi}(\boldsymbol{x})\|_2$ even though the output of f_W would be easily realigned onto $\boldsymbol{x}$ by cross-correlation. In the case of audio restoration of pitched sounds, listeners are more sensitive to artifacts near the onset (e.g., pre-echo) [13], even though most of the spectrogram energy is contained in the sustain and release parts of the temporal profile.

Lastly, in the case of sound matching, certain synthesizers contain parameters that govern periodic structures at larger time scales while being independent of local spectral variations. In additive synthesis, periodic modulation techniques such as vibrato, tremolo, or trill have a "rate" parameter that is neither predictable from isolated spectrogram frames, nor reducible to a sequence of discrete sound events. A small perturbation to synthesis parameters of ε will induce a $\boldsymbol{g}(\boldsymbol{\theta} + \varepsilon)$ globally dilated or compressed but locally misaligned in time, rendering $\|(\boldsymbol{\Phi} \circ \boldsymbol{g})(\boldsymbol{\theta} + \varepsilon) - (\boldsymbol{\Phi} \circ \boldsymbol{g})(\boldsymbol{\theta})\|$ not indicative of the magnitude of ε. Comparison of timbre similarity is no longer possible at the time scale of isolated spectrogram frames.

Modular synthesizers shape sound via an interaction between control modules (sequencers, function generator) and sound processing and generating modules (oscillators, filters, waveshapers) [14]. In a "patch," sequencers determine the playback speed and actuate events, while amplitude envelopes, oscillator waveshapes and filters sculpt the timbre. Changing the clock speed of a patch would cause events to be unaligned in time, but not alter the spectral composition of isolated events.

0.3 Musical Timescales: Micro, Meso, Macro

The shortcomings of modeling music similarity solely at the microscale of short-time spectra is exemplified by the terminology of musical structure used in algorithmic composition. Curtis Roads outlines the challenge of coherently modeling multiscale structures in algorithmic composition [15]. Computer musicians refer to musical structures at a hierarchy of time scales. At one end is the *micro scale*, from sound particles of few samples up to the milliseconds of short-time spectral analysis [16]. Further up the hierarchy of time is the *meso scale,* structures that emerge from the grouping of sound objects and their complex spectrotemporal evolution [17], and the *macro scale* broadly includes the arrangement of a whole composition or performance. In granular synthesis, microstructure arises from individual grains, and their rate of playback forms texture clouds at the level of mesostructure. Beyond the micro scale and spectrogram analysis are sound structures that emerge from complex spectral and temporal envelopes, such as sound textures and instrumental playing techniques [18].

0.4 Contributions

In this paper, the authors pave the way toward DTFA of mesostructure. The key idea is to compute a 2D wavelet decomposition ("scattering") in the time–frequency domain for a sound $\boldsymbol{x}$. The result, named joint time–frequency scattering (JTFS) transform, is sensitive to relative time lags and frequency intervals between musical events. Meanwhile, JTFS remains stable to global time shifts: going back to the example of autoencoding, $f_W(\boldsymbol{x})(t) = \boldsymbol{x}(t - \tau)$ leads to $(\boldsymbol{\Phi}_{\text{JTFS}} \circ f_W)(\boldsymbol{x}) \approx \boldsymbol{\Phi}_{\text{JTFS}}(\boldsymbol{x})$, which is in line with human perception.

To illustrate the potential of JTFS in DTFA, an example of differentiable sound matching in which microscale distance is a poor indicator of parameter distance is presented. In this example, the target sound $\boldsymbol{x} = \boldsymbol{g}(\boldsymbol{\theta})$ is an arpeggio of short glissandi events ("chirplets"), which spans a scale of two octaves. The two unknowns of the problem are the number of chirplets per unit of time and the total duration of the arpeggio. The authors show that it is possible to retrieve these two unknowns without any feature engineering, simply by formulating a least squares inverse problem in JTFS space of the form:

$$\begin{aligned}\boldsymbol{\theta} &= \arg\min_{\widetilde{\boldsymbol{\theta}}} \mathcal{L}_{\boldsymbol{\theta}}(\widetilde{\boldsymbol{\theta}}) \\ &= \arg\min_{\widetilde{\boldsymbol{\theta}}} \|(\boldsymbol{\Phi} \circ \boldsymbol{g})(\widetilde{\boldsymbol{\theta}}) - (\boldsymbol{\Phi} \circ \boldsymbol{g})(\boldsymbol{\theta})\|_2^2.\end{aligned} \tag{4}$$

Intuitively, for the inverse problem above to be solvable by gradient descent, the gradient of $\mathcal{L}_{\boldsymbol{\theta}}$ should point towards $\boldsymbol{\theta}$ when evaluated at any initial guess $\widetilde{\boldsymbol{\theta}}$. The authors' main finding is that such is the case if $\boldsymbol{\Phi}$ is JTFS, but not if $\boldsymbol{\Phi}$ is the MSS. Moreover, the authors find that the gradient of $\mathcal{L}_{\boldsymbol{\theta}}$ remains informative even if the target sound is subject to random time lags of several hundred milliseconds. To explain this discrepancy, the concept of *differentiable*

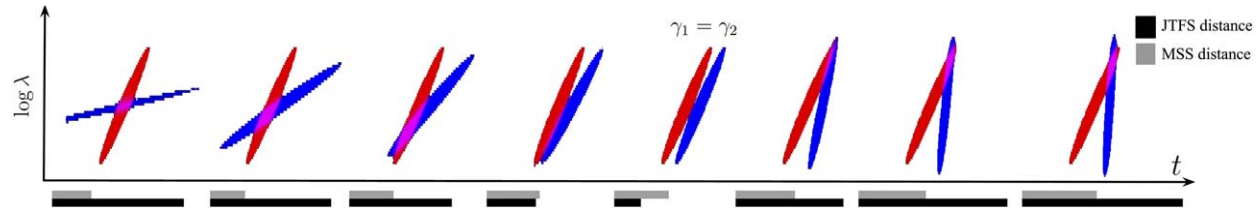


Fig. 1. Illustration of chirps overlapping in time and log–frequency. In each pair of chirps, one is displaced in time from the other. We progressively increase the chirp rate, γ, for one chirp in the pair (left to right). The bars indicate the distance between two chirps in the MSS (gray) and time–frequency scattering (black) domains, respectively. The authors observe that when the chirp rates γ governing *mesostructure* are equal, the JTFS distance is at a minimum, while spectrogram distance is around its maximum. JTFS distance correlates well with distance in γ. The authors give a more detailed discussion of the importance of a time-invariant differentiable mesostructural operator in SEC. 3.

mesostructural operator is defined as yielding the Jacobian matrix of $(\boldsymbol{\Phi} \circ \boldsymbol{x})$ at $\widetilde{\boldsymbol{\theta}}$, i.e., the composition between audio synthesis and JTFS analysis at the parameter setting of interest. This concept is not limited to sound matching but also finds equivalents when training neural networks for autoencoding and audio restoration.

The authors release a differentiable implementation of JTFS in Kymatio v0.4,[1] an open-source software for DTFA on GPU, which is interoperable with modern deep learning libraries [19]. To encourage reproducibility of numerical experiments, this paper is supplemented with open-source code.[2]

1 MOTIVATING EXAMPLE

1.1 Comparing Time-Delayed Chirps

Fig. 1 illustrates the challenge in DTFA of reliably computing similarity between chirps synthesized by $\boldsymbol{g}$. In the example, the first-order moments of two chirps in the time–frequency domain are equal, regardless of frequency modulation (FM) rate. Consider two chirps that are displaced from one another in time. Their spectrogram distance is at a maximum when the mesostructure is identical, i.e., the FM rates are equal and the two signals are disjoint. As the FM rate increases, the two chirps overlap in the time–frequency domain, resulting in a reduction of the spectrogram distance that does not correlate with correct prediction of $\boldsymbol{\theta}$. The spectrogram loss changes little as γ is varied. Moreover, local micro segments of a chirp are periodically shifted in both time and frequency under γ, implying that comparison of microstructure is an inadequate indicator of similarity. A possible solution would be to dynamically realign the chirps; however, this operation is numerically unstable and not differentiable. The following sections outline a differentiable operator that is capable of modeling distance in $\boldsymbol{\theta}$ and stable to time shifts. A representation that is well-equipped to disentangle these three factors of variability should provide neighborhood distance metrics in acoustic space that reflect distance in parameter space.

1.2 Chirplet Synthesizer

A chirplet is a short sound event that produces a diagonal line in the time–frequency plane. Generally speaking, chirplets follow an equation of the form $\boldsymbol{x}(t) = \boldsymbol{a}(t)\cos(2\pi\boldsymbol{\varphi}(t))$ where $\boldsymbol{a}$ and $\boldsymbol{\varphi}$ denote instantaneous amplitude and phase respectively. In this paper, the authors generate chirplets whose instantaneous frequency grows exponentially with time, so that their perceived pitch (roughly proportional to log-frequency) grows linearly. This FM is parametrized in terms of a chirp rate γ, measured in octaves per second. Denoting by f_c the instantaneous frequency of the chirplet at its onset, the following is obtained:

$$\boldsymbol{\varphi}(t) = \frac{f_c}{\gamma \log 2} 2^{\gamma t}. \tag{5}$$

Then, the instantaneous amplitude $\boldsymbol{a}$ of the chirplet is defined as the half-period of a sine function, over a time support of δ^t. This half-period is parameterized in terms of amplitude modulation (AM) frequency $f_m = \frac{1}{2}\delta^t$. Hence:

$$\boldsymbol{a}(t) = \sin(2\pi f_m t) \text{ if } 0 \leq f_m t < \tfrac{1}{2} \text{ and } 0 \text{ otherwise.} \tag{6}$$

At its offset, the instantaneous frequency of the chirplet is equal to $f_m = f_c 2^{\gamma\delta^t} = f_m 2^{\gamma/f_m}$. The notation $\boldsymbol{\theta}$ was used as a shorthand for the AM/FM tuple (f_m, γ).

1.3 Differentiable Arpeggiator

The authors now define an ascending "arpeggio" such that the offset of the previous event coincides with the onset of the next event in the time–frequency domain. To do so, the chirplet is shifted by $n\delta^t$ in time and multiply its phase by $2^{n\delta^f} = 2^{n\gamma\delta^t}$ for integer n. Lastly, a global temporal envelope is applied to the arpeggio, by means of a Gaussian window $(t \mapsto \boldsymbol{\phi}_w(\gamma t)/\gamma)$ of width γw where the bandwidth parameter w is expressed in octaves. Hence:

$$\begin{aligned}\boldsymbol{x}(t) &= \frac{1}{\gamma}\boldsymbol{\phi}_w(\gamma t) \sum_{n=-\infty}^{+\infty} \boldsymbol{a}\left(t - \frac{n}{f_m}\right) \cos\left(2^{\gamma \frac{n}{f_m}} \boldsymbol{\varphi}\left(t - \frac{n}{f_m}\right)\right) \\ &= \boldsymbol{g}_{\boldsymbol{\theta}}(t), \text{ where } \boldsymbol{\theta} = (f_m, \gamma).\end{aligned} \tag{7}$$

In the equation above, the number of events with non-negligible energy is proportional to:

$$\nu(\boldsymbol{\theta}) = \frac{f_m w}{\gamma}, \tag{8}$$

[1] Kymatio v0.4: https://github.com/kymatio/kymatio.

[2] Experiments repository: https://github.com/cyrusvahidi/meso-dtfa.

which is not necessarily an integer number because it varies continuously with respect to $\boldsymbol{\theta}$. Here it is seen that the parametric model $\boldsymbol{g}$, despite being very simple, controls an auditory sensation whose definition only makes sense at the mesoscale: namely, the number of notes ν in the arpeggio that form a sequential stream. Furthermore, this number results from the entanglement between AM (f_{m}) and FM (γ) and would remain unchanged after time shifts [replacing t by $(t-\tau)$] or frequency transposition (varying f_{c}). Thus, although the differentiable arpeggiator has limited flexibility, the authors believe that it offers an insightful test bed for the DTFA of mesostructure.

2 TIME–FREQUENCY SCATTERING

JTFS is a convolutional operator in the time–frequency domain [20]. Via two-dimensional wavelet filters applied in the time–frequency domain at various scales and rates, JTFS extracts multiscale spectrotemporal modulations from digital audio. When used as a frontend to a 2D convolutional neural network, JTFS enables state-of-the-art musical instrument classification with limited annotated training data [21]. Florian Hecker's compositions, e.g., *FAVN* in 2016, mark JTFS's capability of computer music resynthesis (see a full list of compositions from [22]).

2.1 Wavelet Scalogram

Let $\boldsymbol{\psi} \in \mathbf{L}^2(\mathbb{R}, \mathbb{C})$ be a complex-valued wavelet filter of unit center frequency and bandwidth $1/Q_1$. The authors define a constant-Q filterbank of dilations from $\boldsymbol{\psi}$ as $\boldsymbol{\psi}_\lambda : t \longmapsto \lambda\boldsymbol{\psi}(\lambda t)$, with constant quality factor Q_1. Each wavelet has a center frequency λ and a bandwidth of λ/Q_1. The frequency variable λ is discretized under a geometric progression of common ratio $2^{\frac{1}{Q_1}}$, starting from λ/Q_1. For a constant quality factor of $Q_1 = 1$, subsequent wavelet center frequencies are spaced by an octave, i.e., a dyadic wavelet filterbank.

Convolving the filterbank $\boldsymbol{\psi}$ with a waveform $\boldsymbol{x} \in \mathbf{L}^2(\mathbb{R})$ and applying a pointwise complex modulus gives the wavelet scalogram $\mathbf{U}_1$:

$$\mathbf{U}_1\boldsymbol{x}(t, \lambda) = |\boldsymbol{x} * \boldsymbol{\psi}_\lambda|(t). \tag{9}$$

$\mathbf{U}_1$ is indexed by time and log -frequency, corresponding to the commonly known CQT in time–frequency analysis.

2.2 Time–Frequency Wavelets

Similarly to Sec. 2.1, the authors define another two wavelets $\boldsymbol{\psi}^{\mathrm{t}}$ and $\boldsymbol{\psi}^{\mathrm{f}}$ along the time and log -frequency axes, with quality factors equivalent to Q_2 and Q_{fr}, respectively. Then, two filterbanks $\boldsymbol{\psi}^{\mathrm{t}}_\alpha$ and $\boldsymbol{\psi}^{\mathrm{f}}_\beta$ are derived, with center frequencies of α and β, in which

$$\boldsymbol{\psi}^{\mathrm{t}}_\alpha(t) = \alpha\boldsymbol{\psi}^{\mathrm{t}}(\alpha t), \tag{10}$$

$$\boldsymbol{\psi}^{\mathrm{f}}_\beta(\log_2 \lambda) = \beta\boldsymbol{\psi}^{\mathrm{f}}(\beta \log_2 \lambda). \tag{11}$$

As in the computation of $\mathbf{U}_1$, α and β are discretized by geometric progressions of common ratios $2^{\frac{1}{Q_2}}$ and $2^{\frac{1}{Q_{\mathrm{fr}}}}$. The frequency variable α and β are interpreted from a perspective of auditory spectrotemporal receptive fields [23]: α is the temporal modulation rate measured in Hz, and β is the frequential modulation scale measured in cycles per octave.

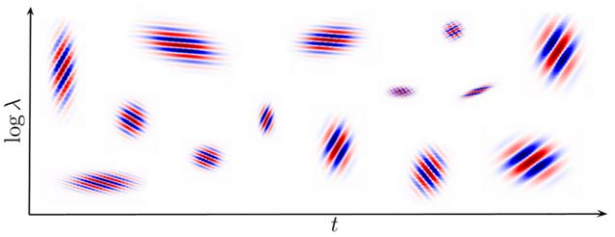


Fig. 2. Illustration of the shape of 2D time–frequency wavelets (second-order JTFS). Each pattern shows the response of the real part of 2D filters that arise from the outer product between 1D wavelets $\boldsymbol{\psi}_\alpha(t)$ and $\boldsymbol{\psi}_\beta(\log \lambda)$ of various rates α and scales β (respectively). Orientation is determined by the sign of β, otherwise known as the *spin* variable falling in $\{-1, 1\}$. See Sec. 2 for details on JTFS.

The outer product between $\boldsymbol{\psi}^{\mathrm{t}}_\alpha$ and $\boldsymbol{\psi}^{\mathrm{f}}_\beta$ forms a family of 2D wavelets of various rates α and scales β. $\boldsymbol{\psi}^{\mathrm{t}}_\alpha$ and $\boldsymbol{\psi}^{\mathrm{f}}_\beta$ are convolved with $\mathbf{U}_1\boldsymbol{x}$ in sequence and a pointwise complex modulus applied, resulting in a four-way tensor indexed (t, λ, α, β):

$$\mathbf{U}_2\boldsymbol{x}(t, \lambda, \alpha, \beta) = |\mathbf{U}_1\boldsymbol{x}(t, \lambda) * \boldsymbol{\psi}^{\mathrm{t}}_\alpha * \boldsymbol{\psi}^{\mathrm{f}}_\beta|. \tag{12}$$

In Fig. 2, the real part of the 2D wavelet filters are visualized in the time–frequency domain. The wavelets are of rate α, scale β and orientation (upward or downward) along $\log_2\lambda$, capturing multiscale oscillatory patterns in time and frequency.

2.3 Local Averaging

The authors compute first-order JTFS coefficients by convolving the scalogram $\mathbf{U}_1\boldsymbol{x}$ of Eq. (9) with a Gaussian low-pass filter $\boldsymbol{\phi}_T$ of width T, followed by convolution with $\boldsymbol{\psi}_\beta$ ($\beta \geqslant 0$) over the log-frequency axis, then pointwise complex modulus:

$$\mathbf{S}_1\boldsymbol{x}(t, \lambda, \alpha = 0, \beta) = |\mathbf{U}_1\boldsymbol{x}(t, \lambda) * \boldsymbol{\phi}_T * \boldsymbol{\psi}_\beta|. \tag{13}$$

Before convolution with $\boldsymbol{\psi}_\beta$, the output of $\mathbf{U}_1 x(t, \lambda) * \boldsymbol{\phi}_T$ is subsampled along time, resulting in a sampling rate proportional to $1/T$. Indeed, Eq. (13) is a special case of Eq. (12) in which modulation rate $\alpha = 0$ by the use of $\boldsymbol{\phi}_T$.

The authors define the second-order JTFS transform of $\boldsymbol{x}$ as

$$\mathbf{S}_2\boldsymbol{x}(t, \lambda, \alpha, \beta) = \mathbf{U}_2\boldsymbol{x}(t, \lambda) * \boldsymbol{\phi}_T * \boldsymbol{\phi}_F, \tag{14}$$

where $\boldsymbol{\phi}_F$ is a Gaussian low-pass filter over the log -frequency dimension of width F. For the special case of $\beta = 0$ in Eq. (12), $\boldsymbol{\psi}_\beta$ performs the role of $\boldsymbol{\phi}_F$, yielding

$$\mathbf{S}_2\boldsymbol{x}(t, \lambda, \alpha, \beta = 0) = |\mathbf{U}_1\boldsymbol{x}(t, \lambda) * \boldsymbol{\psi}^{\mathrm{t}}_\alpha * \boldsymbol{\phi}_F| * \boldsymbol{\phi}_T. \tag{15}$$

In both Eqs. (14) and (15), $\mathbf{S}_2\boldsymbol{x}$ is subsampled to sampling rates of T^{-1} and F^{-1} over the time and log-frequency axes, respectively. Low-pass filtering with $\boldsymbol{\phi}_T$ and $\boldsymbol{\phi}_F$ provides invariance to time shifts and frequency transpositions up to a scale of T and F respectively. The combination of

$\mathbf{S}_1\boldsymbol{x}$ and $\mathbf{S}_2\boldsymbol{x}$, i.e., $\mathbf{S}\boldsymbol{x} = \{\mathbf{S}_1\boldsymbol{x}, \mathbf{S}_2\boldsymbol{x}\}$, allows for covering all paths combining the variables (λ, α, β). SEC. 3 introduces the use of $\mathbf{S}\boldsymbol{x}$ as a DTFA operator for mesostructures.

Fig. 1 highlighted the need for a operator that models mesostructures. The stream of chirplets is displaced in frequency at a particular rate. At second-order, JTFS describes the larger scale spectrotemporal structure that is not captured by $\mathbf{S}_1$. Moreover, JTFS is time-invariant, making it a reliable measure of mesostructural similarity up to time scale T.

3 DIFFERENTIABLE MESOSTRUCTURAL OPERATOR

This section introduces a differentiable mesostructural operator for time–frequency analysis. Such an operator is needed in optimization scenarios that require a differentiable measure of similarity, such as autoencoding.

In SEC. 1, the authors defined a differentiable arpeggiator $\boldsymbol{g}$ whose parameters $\boldsymbol{\theta}$ govern the *mesostructure* in $\boldsymbol{x}$. The authors now seek a differentiable operator $\boldsymbol{\Phi} \circ \boldsymbol{g}$ that provides a model to control the low-dimensional parameter space $\boldsymbol{\theta}$. By way of distance and gradient visualization under $\boldsymbol{\Phi} \circ \boldsymbol{g}$, the authors set out to assess the suitability of $\boldsymbol{\Phi}$ for modeling $\boldsymbol{\theta}$ in a sound matching task.

Two DTFA operators in the role of $\boldsymbol{\Phi}$ are considered: (i) the MSS (approximately $\mathbf{U}_1\boldsymbol{x}$) and (ii) time–frequency scattering ($\mathbf{S}\boldsymbol{x} = \{\mathbf{S}_1\boldsymbol{x}, \mathbf{S}_2\boldsymbol{x}\}$) (JTFS). In case (i), a small distance between two sounds is deemed to be an indication of same *microstructure*. On the contrary, similarity in case (ii) suggests the same *mesostructure*. Although identical $\mathbf{U}_1$ implies equality in mesostructure, the reverse is not true, e.g., in the case of time shifts and non-stationary frequency.

Previously, JTFS has offered assessment of similarity between musical instrument playing techniques that underlie mesostructure. With the DTFA operator $\boldsymbol{\Phi}$, there is potential to model mesostructures by their similarity as expressed in terms of the raw audio waveform, synthesis parameters or neural network weights. In cases such as granular synthesis, it may be desirable to control mesostructure, while allowing microstructure to stochastically vary.

3.1 Gradient Computation and Visualization

A distance objective is evaluated under the operator $\boldsymbol{\Phi} \circ \boldsymbol{g}$ as a proxy for distance in $\boldsymbol{\theta}$:

$$\mathcal{L}_{\boldsymbol{\theta}}(\tilde{\boldsymbol{\theta}}) = \|(\boldsymbol{\Phi} \circ \boldsymbol{g})(\boldsymbol{\theta}) - (\boldsymbol{\Phi} \circ \boldsymbol{g})(\tilde{\boldsymbol{\theta}})\|_2^2. \tag{16}$$

For a given parameter estimate $\tilde{\boldsymbol{\theta}}$, the gradient $\nabla\mathcal{L}_\theta$ of the distance to the target $\boldsymbol{\theta}$ is

$$\nabla\mathcal{L}_\theta(\tilde{\theta}) = -2\Big((\boldsymbol{\Phi} \circ \boldsymbol{g})(\boldsymbol{\theta}) - (\boldsymbol{\Phi} \circ \boldsymbol{g})(\tilde{\boldsymbol{\theta}})\Big)^T \cdot \nabla(\boldsymbol{\Phi} \circ \boldsymbol{g})(\tilde{\theta}). \tag{17}$$

The first term in Eq. (17) is a row vector of length $P = \dim\Big((\boldsymbol{\Phi} \circ \boldsymbol{g})(\boldsymbol{\theta})\Big)$ and the second term is a matrix of dimension $P \times \dim(\tilde{\boldsymbol{\theta}})$. The dot product between the row vector in the first term and each column vector in the high-dimensional Jacobian matrix $\nabla(\boldsymbol{\Phi} \circ \boldsymbol{g})$ yields a low-dimensional vector of $\dim(\boldsymbol{\theta})$. Each column of the Jacobian matrix can be seen as the direction of steepest descent in the parameter space, such that distance in $\boldsymbol{\Phi}$ is minimized. Therefore the operator $\boldsymbol{\Phi} \circ \boldsymbol{g}$ should result in distances that reflect sensitivity and direction of changes in $\boldsymbol{\theta}$.

In $\mathcal{L}_\theta$ of Eq. (16), time–frequency scattering ($\boldsymbol{S}\boldsymbol{x}$) is adopted (see SEC. 2) in the role of $\boldsymbol{\Phi}$. Otherwise, $\mathcal{L}_{\boldsymbol{\theta}}^{MSS}$ is referred to when using the MSS. In the JTFS transform, the authors set $J = 12$, $J_{fr} = 5$, $Q_1 = 8$, $Q_2 = 2$, $Q_{fr} = 2$, and set $F = 0$ to disable frequency averaging.

Alternatively, $\mathcal{L}_{\boldsymbol{\theta}}^{MSS}$ is referred to when using the MSS. Let $\boldsymbol{\Phi}_{\text{STFT}}^{(n)}$ be the STFT coefficients computed with a window size of 2^n. The MSS loss is computed in Eq. (18), which is the average of L1 distances between spectrograms at multiple STFT resolutions:

$$\mathcal{L}_{\boldsymbol{\theta}}^{MSS}(\tilde{\boldsymbol{\theta}}) = \frac{1}{N}\sum_{i=5}^{10} |(\boldsymbol{\Phi}_{\text{STFT}}^{(n)} \circ \boldsymbol{g})(\boldsymbol{\theta}) - (\boldsymbol{\Phi}_{\text{STFT}}^{(n)} \circ \boldsymbol{g})(\tilde{\boldsymbol{\theta}})|. \tag{18}$$

The chosen resolutions account for the sampling rate of 8,192 Hz used by $\boldsymbol{g}$. The authors set $w = 2$ octaves in all subsequent experiments and normalize the amplitude of each $\boldsymbol{g}_{\boldsymbol{\theta}}$.

For this experiment, the authors uniformly sample a grid of 20 × 20 AM/FM rates ($\boldsymbol{f}_{\mathbf{m}}, \boldsymbol{\gamma}$) on a log-scale ranging from 4 to 16 Hz and 0.5 to 4 octaves per second, leading to 400 signals with a carrier frequency of $\boldsymbol{f}_{\mathbf{c}} = 512$ Hz. The center of the grid $\boldsymbol{f}_{\mathbf{m}} = 8.29$ Hz and $\boldsymbol{\gamma} = 1.49$ octaves / second is designated as the target sound. A constant time shift $\tau = 2^{10}$ samples is introduced to the target sound in order to test the stability of gradients under perturbations in microstructures. $\mathcal{L}_{\boldsymbol{\theta}}$ and $\nabla\mathcal{L}_{\boldsymbol{\theta}}$ associated to each sound are evaluated for the two DTFA operators $\boldsymbol{\Phi}_{\text{STFT}}$ and $\boldsymbol{\Phi}_{\text{JTFS}}$.

The loss surfaces and gradient fields are visualized with respect to $\widetilde{\boldsymbol{\theta}}$ in Fig. 3. The authors observe that the JTFS operator forms a loss surface with a single local minimum that is located at the target sound's $\boldsymbol{\theta}$. Meanwhile, gradients across the sampled parameters $\widetilde{\boldsymbol{\theta}}$ consistently point towards the target, despite certain exceptions at high γ, which acoustically correspond to very high FM rate. Contrarily, MSS loss gradient suffers from multiple local minima and does not reach the global minimum when $\widetilde{\boldsymbol{\theta}}$ is located at the target due to time shift equivariance. The authors highlight that the MSS distance is insensitive to variation along AM, making it unsuitable for modeling mesostructures.

In line with these findings, previous work [21] found that 3D visualizations of the manifold embedding of JTFS's nearest neighbor graph revealed a 3D mesh whose principal components correlated with parameters describing carrier frequency, AM and FM. Moreover, K-nearest neighbors regression using a nearest neighbors graph in JTFS space produced error ratios close to unity for each of the three parameters.

3.2 Sound Matching by Gradient Descent

Unlike classic sound matching literature, in which $\tilde{\boldsymbol{\theta}}$ is estimated from a forward pass through trainable $f_{\boldsymbol{W}}$ (i.e.,

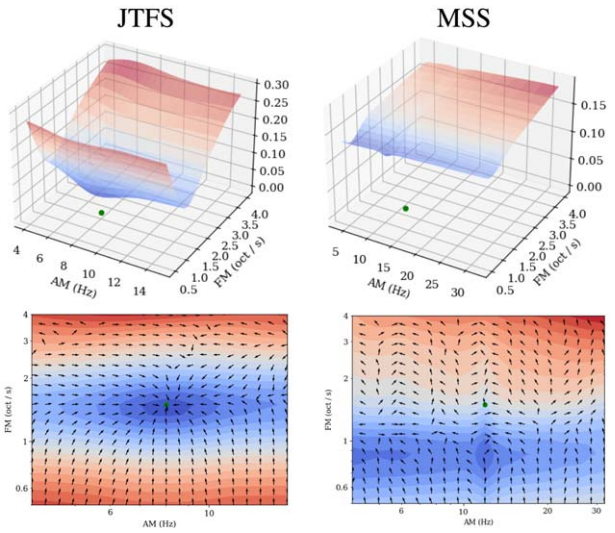


Fig. 3. Loss surface and gradient field visualization under $\mathbf{\Phi}$ as JTFS (a) and MSS (b) for sounds synthesized by $\boldsymbol{g}$ (see SEC. 1). Sounds are sampled from a logarithmically spaced grid on f_{m} and γ. The target sound is plotted as a dot and the loss is computed between the target and a sound generated at every point on the grid. The generated sound is time-shifted relative to the target by a constant of $\tau = 2^{10}$ samples. In the quiver plots, the gradient of the loss operator is evaluated with respect to synthesis parameters f_{m} and γ. The direction of the arrows is indicative of the informativeness of the distance computed on $\mathbf{\Phi} \circ \boldsymbol{g}$ with respect to $\boldsymbol{\theta}$. In the case of $\mathbf{\Phi}_{\mathrm{JTFS}}$, a 3D loss surface whose global minimum is centered around the target sound is observed, whereas gradients point toward the target. Contrarily, the global minimum of $\mathbf{\Phi}_{\mathrm{MSS}}$ does not center around the target or reach 0. In the presence of small time shifts, the MSS loss appears insensitive to differences in AM and uninformative with respect to $\boldsymbol{\theta}$.

neural network weights), sound matching is formulated as an inverse problem in $(\mathbf{\Phi} \circ \boldsymbol{g})$. For the sake of simplicity, the authors do not learn any weights to approximate $\boldsymbol{\theta}$.

Using the gradients derived in SEC. 3.1, sound matching of a target state in $\boldsymbol{\theta}$ is attempted using a simple gradient descent scheme with bold driver heuristics. Additive updates to $\tilde{\boldsymbol{\theta}}$ are performed along the direction dictated by gradient $\nabla_{\tilde{\boldsymbol{\theta}}} \mathcal{L}_{\boldsymbol{\theta}}$:

$$\tilde{\boldsymbol{\theta}} \leftarrow \tilde{\boldsymbol{\theta}} - \alpha \nabla_{\tilde{\boldsymbol{\theta}}} \mathcal{L}_{\boldsymbol{\theta}}. \tag{19}$$

The bold driver heuristic increases the learning rate α by a factor of 1.2 when $\mathcal{L}_{\boldsymbol{\theta}}$ decreases it by a factor of 2 otherwise. The evaluation metric in parameter space is defined as

$$\mathcal{L}_{\boldsymbol{\theta}}(\widetilde{\boldsymbol{\theta}}) = \|\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}\|_2^2. \tag{20}$$

Fig. 6 shows the mean L2 parameter error over gradient descent steps for each $\mathbf{\Phi}$. A fixed target and initial prediction are selected. Multiple optimizations are run that consider time shifts between 0 and 2^{10} samples on the target audio.

Across time-shifts within the support T of the low-pass filter in $\mathbf{\Phi}_{JTFS}$, convergence is stable and reaches close to 0. The authors observe that MSS does not converge and $\mathcal{L}_{\boldsymbol{\theta}}(\widetilde{\boldsymbol{\theta}})$ does not advance far from its initial value, including the case of no time shifts. In Fig. 7, the effects of time shifts for DTFA are further illustrated, validating that JTFS is a time-invariant mesostructural operator up to support T.

3.3 Time Invariance

In Fig. 4, the gradient convergence for different initializations of $\widetilde{\boldsymbol{\theta}}$ are explored but without time shifting the predicted sound. In each plot, gradient descent is performed for 5 different initializations of $\widetilde{\boldsymbol{\theta}}$: (i) far away from the target sound, (ii) in the local neighborhood of the target sound, and (iii) broadly across the parameter grid. The authors highlight that JTFS is able to converge to the solution in each of the three initialization schemes, as corroborated by its gradients in Fig. 5. The authors observe that even without time shifts, MSS fails to recover the target sound in the case that the parameter initialization is far from the target. MSS does indeed recover the target sound if $\widetilde{\boldsymbol{\theta}}$ is initialized

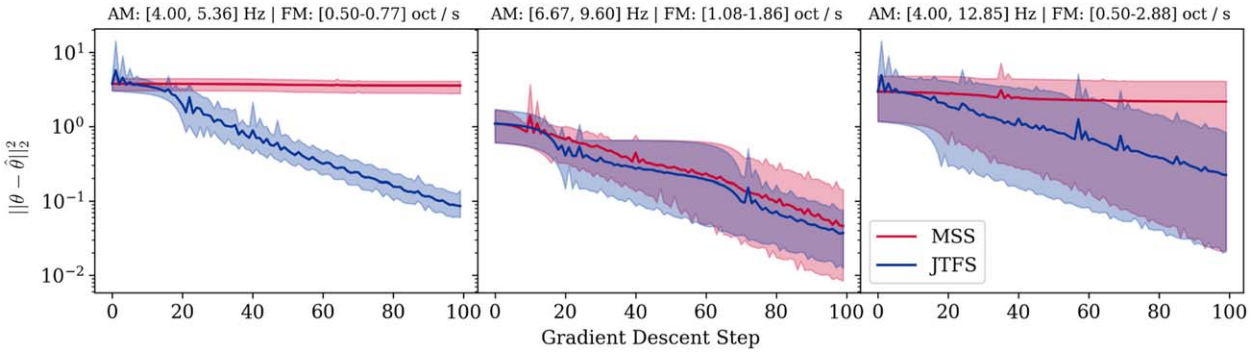


Fig. 4. Parameter distance $||\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}||$ (log-scale) over gradient descent iterations with $\mathbf{\Phi}$ as MSS and JTFS in three scenarios: (a) five initializations of $\widetilde{\boldsymbol{\theta}}$ far from the target, (b) five initializations of $\widetilde{\boldsymbol{\theta}}$ in the neighborhood of the target, and (c) five initializations of $\widetilde{\boldsymbol{\theta}}$ across the range of the grid. A time shift is not applied to the predicted sound (see Fig. 3 for gradient visualization). The target sound has parameters $\boldsymbol{\theta} = [8.49, 1.49]$. The lines indicate the mean distance at each iteration across five runs of different $\widetilde{\boldsymbol{\theta}}$ initialization. The shaded region indicates the range across the five initializations. The titles indicate the range of the initial $\widetilde{\boldsymbol{\theta}}$. The authors highlight that even with no time shifts, MSS only recovers $\boldsymbol{\theta}$ well when $\widetilde{\boldsymbol{\theta}}$ is initialized in its local neighborhood (b). When $\widetilde{\boldsymbol{\theta}}$ is initialized far from the target (a), MSS fails to converge. Starting anywhere (c) converges in the best case but on average fails to converge and is close to the worst case.

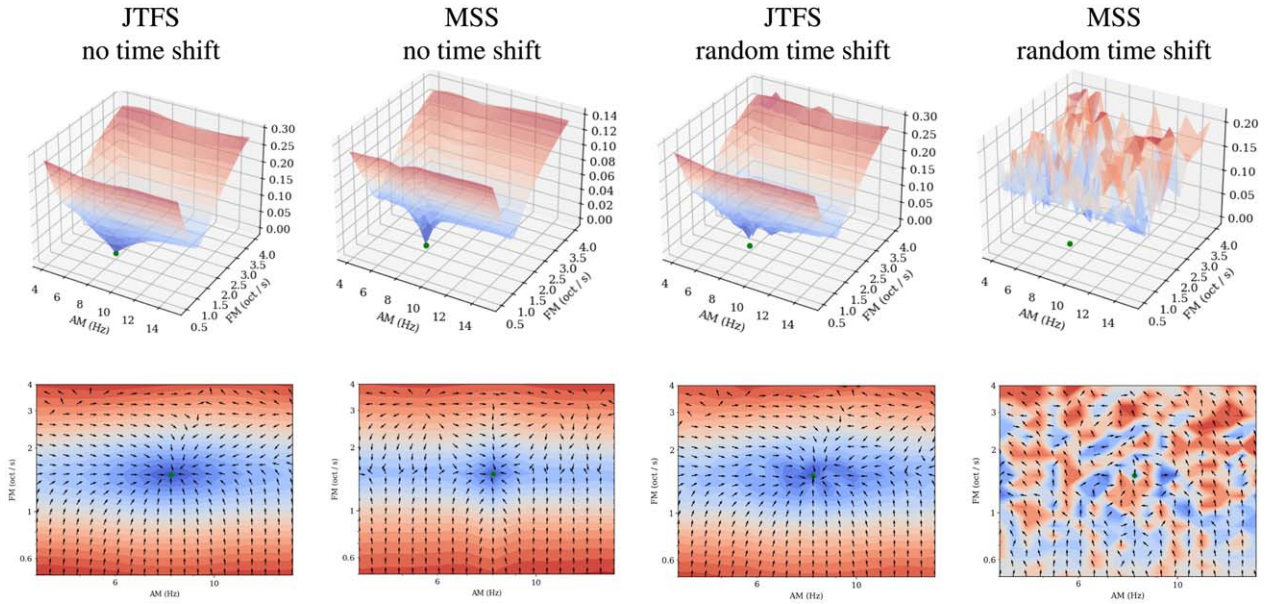


Fig. 5. Loss surfaces (a) and gradient fields (b) under $\mathbf{\Phi}_{\text{JTFS}}$ and the $\mathbf{\Phi}_{\text{MSS}}$ for sounds synthesized by $\boldsymbol{g}$ (see SEC. 1), sampled from a logarithmically spaced grid on f_{m} and γ. Each sound is randomly shifted in time relative to the target by 2^n samples, in which n is sampled uniformly between [8, 12]. The target sound is plotted as a dot and the loss is computed under $\mathbf{\Phi}_{\text{JTFS}}$ and $\mathbf{\Phi}_{\text{MSS}}$ between each sound and the target. In the quiver plots, the gradient of the loss operator is evaluated with respect to the synthesis parameters f_{m} and γ of the generated sound. In the case of both no time shifts, JTFS gradients point toward the target and the distance around 0 when is at the target. Without time shifts, MSS computes distance between objects that intersect in the time–frequency domain. Its gradients appear to lead to the target; however, it suffers from local minima along AM, as demonstrated by convergence in Fig. 4. In the presence of random time shifts, JTFS is appears robust while MSS is highly unstable and prone to local minima.

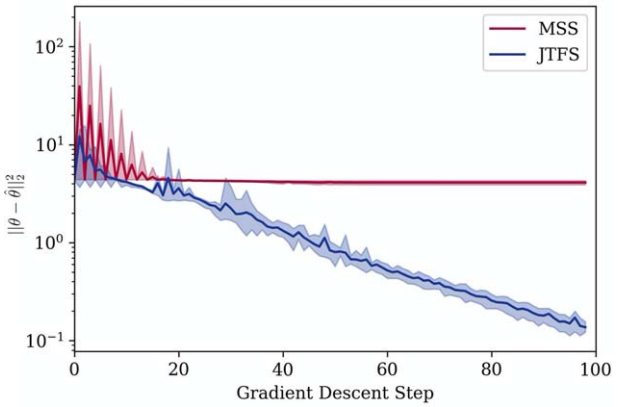


Fig. 6. Parameter distance $||\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}||_2$ over gradient descent iterations with $\mathbf{\Phi}$ as MSS and JTFS. The target sound has parameters $\boldsymbol{\theta} = [8.49, 1.49]$. The predicted sound is initialized at $\widetilde{\boldsymbol{\theta}}_0 = [4, 0.5]$. The line plots the mean distance at each iteration for multiple runs that shift the predicted sample in time by $\tau = \{2^2, 2^4, 2^7, 2^{10}\}$ samples. The shaded region indicates the range across different time shifts.

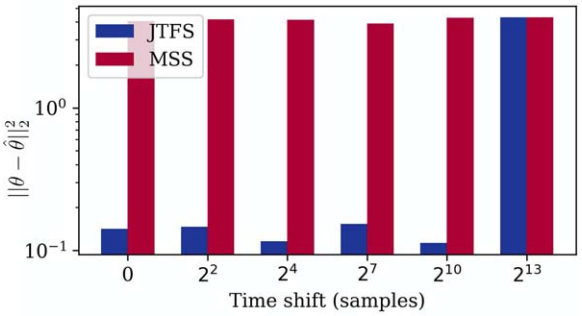


Fig. 7. Final parameter distance $||\boldsymbol{\theta} - \widetilde{\boldsymbol{\theta}}||_2$ after gradient descent for $\boldsymbol{g}(\boldsymbol{\theta})(t)$ and $\boldsymbol{g}(\widetilde{\boldsymbol{\theta}})(t - \tau)$, for $\boldsymbol{\theta} = [8.49, 1.49]$, $\widetilde{\boldsymbol{\theta}}_0 = [4, 0.5]$. Each run ($x$ axis) is optimized under a different time shift τ on the predicted audio. JTFS is invariant up to the support $T = 2^{13}$ of its low-pass filter. The authors observe that convergence in parameter recovery is stable to time shifts under the differentiable mesostructural operator $\mathbf{\Phi} \circ \boldsymbol{g}$, in the case that $\mathbf{\Phi}$ is JTFS. Optimization is unstable when $\mathbf{\Phi}$ is a spectrogram operator.

in the neighborhood of the target. Although when starting anywhere, MSS does indeed converge in the best case, but on average, it is close to the worst case, which does not converge.

Fig. 5 shows the loss surface and gradient fields for $\mathbf{\Phi}_{\text{JTFS}}$ and $\mathbf{\Phi}_{\text{MSS}}$ with no time shifts and random time shifts applied to the predicted sound. Despite MSS reaching the global minimum when the predicted sound is centered at the target, these experiments in gradient descent demonstrate that it is only stable when $\widetilde{\boldsymbol{\theta}}$ is initialized within the local region of the target $\boldsymbol{\theta}$. When a random time shift is applied to the predicted sound, the MSS loss is highly unstable and produces many local minima that are not located at the target sound. As expected, the JTFS gradient is highly stable with no time shifts. Even in the presence of random time shifts, JTFS is an invariant representation of spectrotemporal modulations up to time shifts T.

4 CONCLUSION

DTFA is an emerging direction for audio deep learning tasks. The current state-of-the-art for autoencoding, audio restoration, and sound matching predominantly perform DTFA in the spectrogram domain. However, spectrogram loss suffers from numerical instabilities when computing similarity in the context of (i) time shifts beyond the scale of the spectrogram window and (ii) nonstationarity that arises from synthesis parameters. These prohibit the reliability of spectrogram loss as a similarity metric for modeling multiscale musical structures.

This paper introduced the differentiable mesostructural operator, comprising of modeling synthesis parameters that generate mesostructure by way of DTFA with time–frequency scattering. Synthesis parameters are modeled for a sound matching task using the JTFS for DTFA of structures that are identifiable beyond the locality of microstructure, i.e., amplitude and frequency modulations of a chirplet synthesizer. Notably, JTFS offers a differentiable and scalable implementation of auditory spectrotemporal receptive fields, multiscale analysis in the time–frequency domain, and invariance to time shifts.

However, despite prior evidence that JTFS accurately models similarities in signals containing spectrotemporal modulations, JTFS is yet to be assessed in DTFA for inverse problems and control in sound synthesis. By analysis of the gradient of the DTFA operator with respect to synthesis parameters, the authors showed that in contrast to spectrogram losses, JTFS distance is suitable for modeling similarity in synthesis parameters that describe mesostructure. The stability of JTFS was demonstrated as a DTFA operator in sound matching by gradient descent, particularly in the case of time shifts.

This work lays the foundations for further experiments in DTFA for autoencoding, sound matching, resynthesis, and computer music composition. Indeed, the differentiable mesostructural operator could be used as a model of the raw audio waveform directly; however this approach is prone to resynthesis artifacts [24, 22]. The authors have shown that by means of DTFA, low-dimensional synthesis parameters that shape sequential audio events can be modeled. The mesostructural operator's invariance under frequency translations has yet to be investigated. Frequency invariant differentiable digital signal processing warrants an investigation of its own; the authors plan to address this in future work. Another direction for future work lies in differentiable parametric texture synthesis, in which texture similarity may be optimized in terms of parameters that derive larger scale structures, e.g., beyond the definition of individual grains in granular synthesis.

5 ACKNOWLEDGMENT

Cyrus Vahidi is a researcher at the UK Research and Innovation (UKRI) Centre for Doctoral Training (CDT) in AI and Music, supported jointly by the UKRI (grant number EP/S022694/1) and Music Tribe. This work was conducted during a research visit at LS2N, CNRS. Changhong Wang is supported by an Atlanstic2020 project on Trainable Acoustic Sensors (TrAcS).

6 REFERENCES

[1] C. Schörkhuber and A. Klapuri, "Constant-Q Transform Toolbox for Music Processing," in *Proceedings of the 7th Sound and Music Computing (SMC) Conference*, paper (Barcelona, Spain) (2010 Jul.).

[2] M. Müller, *Fundamentals of Music Processing: Audio, Analysis, Algorithms, Applications* (Springer, Cham, Switzerland, 2015).

[3] K. W. Cheuk, H. Anderson, K. Agres, and D. Herremans, "nnAudio: An On-the-Fly GPU Audio to Spectrogram Conversion Toolbox Using 1D Convolutional Neural Networks," *IEEE Access*, vol. 8, pp. 161981–162003 (2020 Aug.). https://doi.org/10.1109/ACCESS.2020.3019084.

[4] M. Andreux and S. Mallat, "Music Generation and Transformation With Moment Matching-Scattering Inverse Networks," in *Proceedings of the 19th International Society on Music Information Retrieval (ISMIR)*, pp. 327–333 (Paris, France) (2018 Sep.).

[5] Y.-Y. Yang, M. Hira, Z. Ni, et al., â€œTorchAudio: Building Blocks for Audio and Speech Processing," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6982–6986 (Singapore) (2022 May).

[6] J. Engel, C. Gu, A. Roberts, et al., "DDSP: Differentiable Digital Signal Processing," in *Proceedings of the International Conference on Learning Representations (ICLR)*, paper 435 (Addis Ababa, Ethiopia) (2020 Apr.).

[7] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An End-to-End Neural Audio Codec," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 30, pp. 495–507 (2021 Nov.). https://doi.org/10.1109/TASLP.2021.3129994.

[8] P. Manocha, A. Finkelstein, R. Zhang, et al., "A Differentiable Perceptual Audio Metric Learned From Just Noticeable Differences," in *Proceedings of the INTERSPEECH Conference*, pp. 2852–2856 (Shanghai, China) (2020 Oct.).

[9] J. Su, Y. Wang, A. Finkelstein, and Z. Jin, "Bandwidth Extension is All You Need," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 696–700 (Toronto, Canada) (2021 May).

[10] P. Esling, N. Masuda, A. Bardet, R. Despres, and A. Chemla-Romeu-Santos, "Flow Synthesizer: Universal Audio Synthesizer Control With Normalizing Flows," *Appl. Sci.*, vol. 10, no. 1, paper 302 (2019 Jan.). https://doi.org/10.3390/app10010302.

[11] N. Masuda and D. Saito, "Synthesizer Sound Matching With Differentiable DSP," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference (ISMIR)*, pp. 428–434 (Online) (2021 Nov.).

[12] J. Turian and M. Henry, "I'm Sorry for Your Loss: Spectrally-Based Audio Distances Are Bad at Pitch," presented at the *1st I Can't Believe It's Not Better Workshop (ICBINC@NeurIPS)* (Vancouver, Canada) (2020 Dec.).

[13] K. Brandenburg, "MP3 and AAC Explained," in *Proceedings of the Audio Engineering Society 17th International Conference: High-Quality Audio Coding* (1999 Sep.), paper 17-009.

[14] M. Subotnick, "The Use of the Buchla Synthesizer in Musical Composition," presented at the *38th Convention of the Audio Engineering Society* (1970 May), paper 709.

[15] C. Roads, "From Grains to Forms," in *Proceedings of the Iannis Xenakis International Symposium*, vol. 8, p. 4 (Paris, France) (2012 May).

[16] C. Roads, *Microsound* (MIT Press, Cambridge, MA, 2002). https://doi.org/10.7551/mitpress/4601.001.0001.

[17] C. Roads, "Rhythmic Processes in Electronic Music," in *Proceedings of the International Computer Music Conference (ICMC)*, pp. 27–31 (Athens, Greece) (2014 Sep.).

[18] V. Lostanlen, J. Andén, and M. Lagrange, "Fourier at the Heart of Computer Music: From Harmonic Sounds to Texture," *C. R. Phys.*, vol. 20, no. 5, pp. 461–473 (2019 Jul.). https://doi.org/10.1016/j.crhy.2019.07.005.

[19] M. Andreux, T. Angles, G. Exarchakis, et al., "Kymatio: Scattering Transforms in Python," *J. Mach. Learn. Res.*, vol. 21, no. 60, pp. 1–6 (2020 Jan.).

[20] J. Andén, V. Lostanlen, and S. Mallat, "Joint Time–Frequency Scattering," *IEEE Trans. Signal Process.*, vol. 67, no. 14, pp. 3704–3718 (2019 May). https://doi.org/10.1109/TSP.2019.2918992.

[21] J. Muradeli, C. Vahidi, C. Wang, Differentiable Time-Frequency Scattering on GPU," in *Proceedings of the 25th International Conference on Digital Audio Effects (DAFx)*, pp. 1–8 (Vienna, Austria) (2022 Sep.).

[22] V. Lostanlen and F. Hecker, "The Shape of RemiXXXes to Come: Audio Texture Synthesis With Time-Frequency Scattering," in *Proceedings of the 22nd International Conference on Digital Audio Effects (DAFx)*, paper 58 (Birmingham, UK) (2019 Sep.).

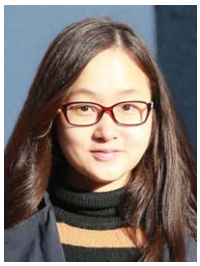
[23] T. Chi, P. Ru, and S. A. Shamma, "Multiresolution Spectrotemporal Analysis of Complex Sounds," *J. Acoust. Soc. Am.*, vol. 118, no. 2, pp. 887–906 (2005 Aug.). https://doi.org/10.1121/1.1945807.

[24] J. Engel, C. Resnick, A. Roberts, Neural Audio Synthesis of Musical Notes With Wavenet Autoencoders," in *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pp. 1068–1077 (Sydney, Australia) (2017 Jul.).

THE AUTHORS

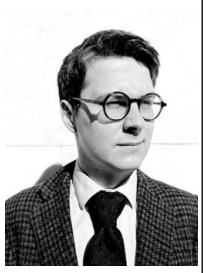

Cyrus Vahidi


Han Han


Changhong Wang


Mathieu Lagrange


György Fazekas


Vincent Lostanlen

Cyrus Vahidi is a Ph.D. researcher in AI and Music at the Centre for Digital Music, London. He received his B.Eng. in Computing at Imperial College London and M.Sc. in Sound and Music Computing at Queen Mary University of London. In 2022, he was a visiting scholar at LS2N, CNRS.

•

Han Han is a Ph.D. student at LS2N, CNRS. She obtained her B.S. in Computational Physics at Rice University and M.S. in Integrated Digital Media at New York University.

•

Changhong Wang is a postdoc researcher at LTCI, Télécom Paris, Institut Polytechnique de Paris. She obtained her Ph.D. from Queen Mary University of London in 2021. In 2022, she worked as a postdoc researcher at LS2N.

•

Mathieu Lagrange is a CNRS research scientist at LS2N. He obtained his Ph.D. from the University of Bordeaux in 2004. Before joining CNRS, he was a scientist in Canada (University of Victoria, McGill University) and in France (Télécom Paris, Ircam).

•

George Fazekas is a Senior Lecturer at the Center for Digital Music, Queen Mary University of London. He holds a B.Sc., M.Sc., and Ph.D. degree in Electronic Engineering. He is an investigator of UKRI's £6.5M Centre for Doctoral Training in Artificial Intelligence and Music (AIM CDT), and he was QMUL's Principal Investigator on the H2020-funded Audio Commons project. He was general chair of ACM's Audio Mostly 2017 and papers co-chair of the AES 53rd International Conference on Semantic Audio, and he received the Citation Award of the AES. He published over 150 papers in the fields of Music Information Retrieval, Semantic Web, Deep Learning, and Semantic Audio.

•

Vincent Lostanlen is a CNRS research scientist at LS2N. He obtained his Ph.D. from École normale supérieure in 2017. Before joining CNRS, he was a scientist at the Cornell Lab of Ornithology and New York University.