Learnable Adaptive Time-Frequency Representation via Differentiable Short-Time Fourier Transform

M. Leiber, Y. Marnissi, A. Barrau, S. Meignen, and L. Massoulié

***Abstract*—The short-time Fourier transform (STFT) is widely used for analyzing non-stationary signals. However, its performance is highly sensitive to its parameters, and manual or heuristic tuning often yields suboptimal results. To overcome this limitation, we propose a unified differentiable formulation of the STFT that enables gradient-based optimization of its parameters. This approach addresses the limitations of traditional STFT parameter tuning methods, which often rely on computationally intensive discrete searches. It enables fine-tuning of the time-frequency representation (TFR) based on any desired criterion. Moreover, our approach integrates seamlessly with neural networks, allowing joint optimization of the STFT parameters and network weights. The efficacy of the proposed differentiable STFT in enhancing TFRs and improving performance in downstream tasks is demonstrated through experiments on both simulated and real-world data.**

***Index Terms*—short-time Fourier transform, spectrogram, differentiable STFT, learnable STFT parameters, adaptive time-frequency representation**

I. Introduction

Fourier theory is a cornerstone of signal processing and finds widespread application across science and engineering. The *short-time Fourier transform* (STFT) is a fundamental technique for analyzing non-stationary signals in diverse fields, including audio processing [1], medicine diagnostics [2], and vibration analysis [3]. STFT-based representations such as spectrograms, mel spectrograms, and the constant-Q transform, are established tools for visualizing, understanding, and processing non-stationary signals. In recent years, the integration of *time-frequency* (TF) analysis with machine learning has garnered significant attention, with STFT-based representations being extensively used in tasks such as speech recognition [4], speech enhancement [5], music detection [6], data augmentation [7], and source separation [8], among others. The combination of neural networks and spectrograms has demonstrated remarkable success in these applications, leveraging the ability of neural networks to learn intricate patterns and the capacity of the spectrogram to capture essential signal characteristics in the TF domain.

However, the performance of these techniques is critically dependent on the *analysis window* (type and length) employed in the STFT definition, as well as the TF grid determined by the hop length. These parameters significantly influence the accuracy and interpretability of the resulting representation [9]–[12]. Common window functions include Hann, Hamming, and Gaussian windows, with the optimal choice often being application-specific [13]. The *window length* determines the trade-off between temporal and frequency resolution, in accordance with the Heisenberg uncertainty principle [14]. The *hop or overlap length*, which defines the shift between successive analysis windows, controls the balance between temporal resolution (i.e., capturing the smooth evolution of frequency content) and computational cost, as well as the temporal positioning of the analysis frames.

This paper proposes a novel gradient-based optimization approach for tuning STFT parameters, building upon our prior works [15]–[17]. Gradient-based optimization has attracted considerable interest in the scientific community due to increasing computational resources and the remarkable success of neural networks trained via backpropagation [18]–[20]. The unified framework we introduce here is particularly simple to implement and offers the advantage of seamless integration with neural networks, where the STFT can be viewed as a network layer with STFT parameters acting as weights. Furthermore, it provides enhanced computational efficiency compared to traditional techniques like adaptive STFT [21] or variable STFT [22], which typically involve optimizing over a discrete parameter set using computationally intensive greedy algorithms. Another key advantage of our optimization framework is that, by operating on real-valued parameters, it can achieve more accurate optimal values compared to methods restricted to a discrete set of parameter candidates [23], [24].

A key contribution of this work is the demonstration that the STFT is differentiable with respect to its window and hop lengths, provided these parameters are treated as real-valued. This differentiability enables the computation of optimal parameters using gradient-based optimization via forward and backward propagation. The remainder of this paper is organized as follows: Sec. II-A provides the necessary background on the STFT, and Sec. II-B reviews related work on parameter adaptation. In Sec. III, we introduce our differentiable STFT formulation with respect to the window and hop lengths. Sec. IV presents the partial derivatives and backpropagation formulas. Sec. V details the numerical implementation and computational complexity. We then propose two approaches for optimizing the STFT parameters through applications: a representation-driven optimization approach in Sec. VI and a task-driven optimization approach in Sec. VII. Finally, we provide in Sec. VIII some concluding remarks.

II. Background and related works

A. Short-Time Fourier Transform

The STFT operates in two stages: first, the input discrete-time signal is segmented into overlapping frames using a window function; second, the *discrete Fourier Transform*

(DFT) is computed for each frame. This process yields a two-dimensional complex-valued matrix, $\mathcal{S}_{\omega_L}[n,m]$, where n and m are integer indices standing for the time frame and frequency bin, respectively. More formally, the STFT involves sliding a window function, or *tapering function*, of even length L, across the input signal s. At each time frame, centered at a temporal position t_n, the DFT of the windowed signal segment is computed to analyze the local frequency content. Let ω_L denote the tapering function, the STFT of the signal s can be expressed as:

$$\mathcal{S}_{\omega_L}[n,m] = \sum_{k=t_n-L/2}^{t_n+L/2-1} \omega_L[k-t_n]s[k]e^{\frac{-j2\pi km}{L}}, \qquad (1)$$

where j is the imaginary unit. The integer time indices t_n, referred to as the *temporal positions*, are typically equally spaced. In such cases, the spacing between consecutive temporal positions, $H = t_n - t_{n-1}$, is constant and is known as the *hop length*. The hop length is often defined as a fraction of the window length L using an overlap ratio $0 < \alpha < 1$: $H = \lfloor \alpha L \rfloor$ where $\lfloor \cdot \rfloor$ denotes the floor operation. It is important to note that in Eq. (1), t_n represents the center of the window applied to the signal. Different temporal shifts of the window can be achieved by varying t_n over discrete values.

B. Related Works

This section provides an overview of existing approaches aimed at enhancing the TFR obtained from the STFT, with a particular emphasis on methods that strive to achieve higher energy concentration in the TF plane. These approaches can be broadly classified into two main categories: pre-processing methods, which focus on determining optimal STFT parameters, and post-processing techniques, which involve applying transformations to an initially computed STFT.

1) STFT parameters optimization: The concept of optimizing STFT parameters, particularly the window length, has been explored in numerous studies. Techniques such as *adaptive STFT* (ASTFT) [21] and variable STFTs (VSTFT) [22] were developed to address the inherent trade-offs associated with fixed window lengths.

Various implementations of ASTFT exist. In [24], a method for adapting the window length along the time axis was proposed; however, this approach is less effective for multi-component signals. Adaptations of the window length in the TF plane have also been introduced, based on local chirp-rate estimation [25], or by minimizing a correlation-based criterion at each TF point [23]. These methods typically employ a variable integer window length to achieve a finer control over the temporal and frequency resolution trade-off based on local signal characteristics or the resulting transform. However, determining the optimal parameters often involves computationally expensive discrete optimization techniques, such as grid search or trial-and-error over a predefined set of values. For instance, in TF window adaptation-based ASTFT, finding the optimal parameters using grid search can become rapidly prohibitive, since grid search which exhaustively evaluates all combinations of parameters, incurs an exponential computational cost of $O(P^n)$, where P is the parameter space size (e.g., number of possible window lengths) and n is the number of independent TF regions.

In VSTFT [22], the window length is varied based on the local instantaneous frequency, estimated within each window slice over time, with the length being inversely proportional to the instantaneous frequency. However, this local estimation can be unreliable in the presence of multi-component signals. An extension of the Gabor transform, allowing for frequency (resp. time) resolution that changes over time (resp. frequency), was proposed in [26]. However, this method does not consider simultaneous adaptation in both time and frequency.

In contrast to these traditional methods, which have primarily optimized the window size using discrete searches over a limited set of candidates, recent research has focused on optimizing STFT parameters via gradient descent. This involves treating STFT parameters as continuous, real-valued variables. For example, a gradient-based optimization of the window length was proposed in [15]. This idea was extended in [16] to include adaptive window lengths varying in both time and frequency. The optimization of the hop length using gradient descent was introduced in [17]. In [27], STFT parameters were optimized for sound classification, specifically using a Gaussian window. [28] and [29] have explored the optimization of window length for bearing fault diagnosis.

2) Post-processing techniques: Numerous post-processing techniques have been explored to enhance the readability of spectrograms after their initial computation. These include reassignment methods [30]–[34] and synchrosqueezing transforms [35]–[42].

Synchrosqueezing techniques aim to improve the sharpness of the TFR by reassigning the STFT coefficients of a signal. They offer an advantage over classical reassignment methods [34] by preserving the invertibility of the transform. However, the effectiveness of the reassignment process in synchrosqueezing, particularly when dealing with multi-component signals, heavily relies on the degree of separation between the modes in the TF plane. If two modes are closely spaced, and if the analysis window is not appropriately chosen, the reassignment may not be effective. To address this, new versions of the synchrosqueezing transform have been proposed that incorporate local time adaptation of the window [43]. Unfortunately, these transforms still do not allow for simultaneous adaptation of the window in both time and frequency. Another significant limitation of such reassignment processes is their sensitivity to noise. Furthermore, the optimization was based on a grid search. Synchrosqueezing was also adapted to the short-time fractional Fourier transform in [44], and an improvement including window adaptation was proposed in [45]. In that paper, the fractional order was also optimized locally only in time simultaneously with the window length, and grid search was considered for optimization. Finally, it is important to note that such an adaptation of the short-time fractional Fourier transform is essentially used , and very useful, to deal with multicomponent signals containing parallel modes.

3) Discussion: It is worth noting that post-processing techniques share a common goal with the STFT parameter optimization methods: to enhance the readability and interpretability of TFRs. However, their approaches differ fundamentally. Specifically, while ASTFT and VSTFT aim to adapt the TF resolution locally during the STFT computation, methods like reassignment and synchrosqueezing are applied to a STFT computed with fixed parameters, potentially optimized beforehand. As previously mentioned, the success of reassignment processes is contingent upon the initial STFT being well-suited for the signal characteristics. Therefore, pre-processing (parameter optimization) and post-processing techniques should be viewed as complementary strategies rather than mutually exclusive alternatives.

III. Differentiable Short-Time Fourier Transform

A. Differentiable STFT with respect to window and hop lengths

To define a differentiable formulation of the STFT with respect to its parameters -specifically, the window length and temporal positions- it is necessary to adapt the standard STFT definition given in Eq. (1) to accommodate real-valued parameters. Note that differentiating the STFT with respect to the frame temporal position t_n is equivalent to differentiating with respect to the hop length $H_n = t_n - t_{n-1}, H_0 = t_0$.

We begin by considering a base window function ω_L with compact support on $[-L/2, L/2]$. We then define a family of contracted windows as:

$$\omega(x,\theta) = \frac{L}{\theta}\omega_L(\frac{L}{\theta}x), \quad \forall\theta \in]0, L]. \tag{2}$$

By construction, $\omega(x,\theta)$ is compactly supported on $[-\theta/2, \theta/2]$, such that

$$\forall x \notin [-\theta/2, \theta/2], \quad \omega(x,\theta) = 0. \tag{3}$$

The normalization factor L/θ is crucial for preserving the L^1-norm of the window as the parameter L is scaled to θ. This ensures a fair comparison of STFTs with varying window lengths. Specifically, if the input signal is a pure tone $A\exp(j2\pi at)$, its STFT is $A\hat{\omega}(\eta - a, \theta)$, where $\hat{\omega}$ is the Fourier transform. To ensure that the magnitude of the STFT at zero frequency shift is independent of θ, i.e., $\hat{\omega}(0,\theta) = \int \omega(x,\theta)dx$ is constant, the normalization in Eq. (2) is necessary.

An example of such a window family derived from the normalized Hann function is:

$$\omega(x,\theta) = \frac{1}{2\theta}\left(1 + \cos\left(\frac{2\pi x}{\theta}\right)\right) 1_{|x| \le \theta/2}. \tag{4}$$

Another common example is derived from the Gaussian function with variance σ^2 where $\sigma = \theta/6$:

$$\omega(x,\theta) = \frac{1}{\sigma}\exp\left(-\pi\left(\frac{x}{\sigma}\right)^2\right) 1_{|x| \le \theta/2}. \tag{5}$$

The Gaussian window is widely used as its Fourier transform is also a Gaussian, $\exp(-\pi\sigma^2\eta^2)$, preserving the amplitude of pure harmonics in the frequency domain. In this case, if we consider an effective support $L > \theta = 6\sigma$, then $\omega(x,\theta) = \frac{L}{\theta}\omega_L(\frac{L}{\theta}x)$, with $\omega_L(x) = \exp(-\pi x^2)$ being very small at $x = \pm L/2$. While the Gaussian window is not strictly compactly supported, its effective support allows for practical implementation. The differentiability at the exact boundaries of the compact support is a theoretical point that we will not delve into further in this work.

We can now define a generalized form of the STFT, taking into account variable window length and temporal position:

$$\mathcal{S}_\omega(t,m,\theta) = \sum_{k\in\mathbb{Z}} \omega(k-t,\theta)s[k]e^{-\frac{j2\pi km}{L}}. \tag{6}$$

Then, considering N times indices and $M = L/2 + 1$ frequency indices, one introduces the operator $\mathcal{S}$ as:

$$\begin{aligned} \mathcal{S} : \mathbb{R}^N \times]0, L]^{M\times N} &\mapsto \mathbb{C}^{M\times N} \\ \Omega = (t_n, \theta_{m,n})_{m,n} &\to (\mathcal{S}_\omega(t_n, m, \theta_{m,n}))_{m,n}, \end{aligned} \tag{7}$$

where $\Omega = (t_n)_{n=0}^{N-1} \cup (\theta_{m,n})_{m=0,n=0}^{M-1,N-1}$ represents the set of trainable parameters. In this formulation, the window length $\theta_{m,n}$ can vary with both time and frequency indices, while the temporal positions t_n are real-valued and depend on the time index only. All these parameters are real-valued. The operator $\mathcal{S}$ yields an $M \times N$ matrix, with rows corresponding to frequency and columns to time. If the bidimensional window function $(x,\theta) \mapsto \omega(x,\theta)$ is differentiable for all (x,θ), then the modified STFT is differentiable with respect to both window and hop lengths (or temporal positions). In the following, we assume this property holds and refer to this modified STFT as DSTFT (for differentiable STFT).

The maximum frequency resolution in Eq. (6) is determined by the support L (the size of the DFT), as the effective window of length $\theta_{m,n} \le L$ is implicitly zero-padded to this length. The parameter $\theta_{m,n}$ governs the time resolution, analogous to L in the classical STFT, by defining the temporal extent of the signal segment analyzed for a local spectrum. The zero-padding operation does not alter the intrinsic frequency resolution, which is determined by the total length L; rather, it provides an interpolation of the frequency spectrum whose resolution is controlled by the window length $\theta_{m,n}$. Therefore, L should be primarily considered as an upper bound on the temporal resolution $\theta_{m,n}$ that allows maintaining a consistent size for the frequency dimension of the spectrogram, which is essential for differentiability, as will be discussed later.

In our DSTFT definition, the window length can a priori be different for each time and frequency index, and the hop length varies with the time index only. However, it is also possible to consider a window length that varies only with frequency, θ_m (as in the S-transform), or only with time, θ_n (as in some versions of ASTFT [24]), or a constant window length, θ, as in the classical STFT. The same applies to the temporal positions. Note that the classical STFT defined in Eq. (1) can be obtained by setting $\theta_{m,n} = L$ and $t_n = t_0 + nH$, where $H \in \mathbb{N}$ is the hop length. While we could define temporal positions that depend on both time and frequency indices, such a representation would be challenging to interpret as the signal samples involved in the DSTFT computation for a given time index would vary with the frequency index. Nevertheless, this could potentially be used as input for a learning algorithm, but this is beyond the scope of the present paper.

B. Fixed-Overlap DSTFT

In the preceding section, we assumed that the temporal positions of the tapering window, t_n, are independent of the window lengths, resulting in a constant number of time frames (columns) in the STFT representation. This configuration is particularly useful when the STFT serves as input to algorithms that require a fixed-size representation, such as neural networks. However, in certain applications, this assumption may not always be appropriate. For instance, in spectrogram visualization, a common practice is to define a fixed overlap ratio α (e.g., 50%) and make the temporal positions of the window dependent on the corresponding window lengths through the relationship $t_n = n\alpha\theta_n$. It is important to note that with this setting, the number of time frames increases as θ_n decreases:

$$\mathcal{S}_\omega(n\alpha\theta_n, m, \theta_n) = \sum_{k\in\mathbb{Z}} \omega(k - n\alpha\theta_n, \theta_n) s[k] e^{-\frac{j2\pi km}{L}}. \tag{8}$$

IV. Analytical Expressions for partial derivatives

In the preceding section, we introduced the DSTFT and established its differentiability with respect to its parameters, assuming a well-designed window function. This implies that the partial derivatives $\frac{\partial\mathcal{S}}{\partial\theta_{m,n}}$ and $\frac{\partial\mathcal{S}}{\partial t_n}$ are well-defined and finite. In this section, we derive the analytical expressions for these derivatives and discuss how these formulas can be integrated into more general applications.

A. Partial Derivatives with Respect to Window Length

For notational simplicity, we denote the partial derivative of a function f with respect to a variable x as $\partial_x f$. Considering the case where the window length $\theta_{m,n}$ varies with both frequency index m and time index n, the partial derivative of the DSTFT output $\mathcal{S}(\Omega)$ with respect to $\theta_{m,n}$ is given by:

$$\begin{aligned}\partial_{\theta_{m,n}}\mathcal{S}(\Omega)_{m',n'} &= \sum_{k\in\mathbb{Z}} \partial_{\theta_{m,n}}\omega(k - t_{n'}, \theta_{m',n'}) s\left[k\right] e^{-\frac{j2\pi km'}{L}} \\ &= \mathcal{S}_{\partial_\theta\omega}(t_{n'}, m', \theta_{m',n'})\delta_{m,m'}\delta_{n,n'},\end{aligned} \tag{9}$$

hence we denote:

$$\partial_{\theta_{m,n}}\mathcal{S}(\Omega) = (\mathcal{S}_{\partial_\theta\omega}(t_{n'}, m', \theta_{m',n'})\delta_{m,m'}\delta_{n,n'})_{m',n'}, \tag{10}$$

where $\delta_{m,m'}$ is the Kronecker delta, which equals 1 if $m = m'$ and 0 otherwise. Eq. (10) represents a matrix where only the coefficient at time index n and frequency index m is non-zero. This non-zero coefficient corresponds to the STFT of the signal $s[k]$ computed using the partial derivative of the tapering function with respect to window length, $\partial_\theta\omega$, instead of ω itself, evaluated at the specific time t_n frequency m with window length $\theta_{m,n}$.

In the case of frequency-only varying window length, $\mathcal{S}$ in defined on $\mathbb{R}^N\times]0, L]^N$ and the partial derivative of the DSTFT with respect to window length θ_n is the following matrix:

$$\partial_{\theta_n}\mathcal{S}(\Omega) = (\mathcal{S}_{\partial_\theta\omega}(t_{n'}, m, \theta_{n'})\delta_{n,n'})_{m,n'}. \tag{11}$$

In the case of a time-only varying window length, θ_n, where $\mathcal{S}$ is defined on $\mathbb{R}^N\times]0, L]^M$, the partial derivative of the DSTFT with respect to θ_n is the following matrix:

$$\partial_{\theta_m}\mathcal{S}(\Omega) = (\mathcal{S}_{\partial_\theta\omega}(t_n, m', \theta_m)\delta_{m,m'})_{m',n} \tag{12}$$

Finally, for a constant window length θ, where $\mathcal{S}$ is defined on $\mathbb{R}^N\times]0, L]$, we obtain:

$$\partial_\theta\mathcal{S}(\Omega) = (\mathcal{S}_{\partial_\theta\omega}(t_n, m, \theta))_{m,n}. \tag{13}$$

B. Partial Derivatives with Respect to Window Temporal Position and Hop-Length

Following a similar approach to the previous section, and noting that the temporal position of the n^{th} frame, t_n, depends solely on the time index n, we directly obtain the partial derivative of the DSTFT output with respect to t_n as:

$$\partial_{t_n}\mathcal{S}(\Omega) = -(\mathcal{S}_{\partial_x\omega}(t_{n'}, m, \theta_{m,n'})\delta_{n,n'})_{m,n'}. \tag{14}$$

This represents a matrix where only the coefficient at time index n is non-zero, corresponding to the STFT of the signal using the partial derivative of the tapering function with respect to time, $\partial_x\omega$, evaluated at time t_n, frequency m, and window length $\theta_{m,n}$.

In the case of time-varying hop length H_n, where $H_n = t_n - t_{n-1}$ and $H_0 = t_0$, we remark that $t_n = \sum_{i=0}^{n} H_i$. We can use the chain rule to rewrite the partial derivative with respect to t_n:

$$\begin{aligned}\partial_{H_n}\mathcal{S}(\Omega) &= (\partial_{H_n}\mathcal{S}_\omega(t_{n'}, m', \theta_{m',n'}))_{m',n'} \\ &= (-\mathcal{S}_{\partial_x\omega}(t_{n'}, m', \theta_{m',n'})1_{x>0}(n' - n))_{m',n'}\end{aligned} \tag{15}$$

where $1_{x>0}$ is the indicator function of $\mathbb{R}_*^+$.

For a constant hop length H, where $t_n = t_0 + nH$, the partial derivative with respect to H can be derived as:

$$\partial_H\mathcal{S}(\Omega) = (-n\mathcal{S}_{\partial_x\omega}(t_n, m, \theta_{m,n}))_{m,n}. \tag{16}$$

Remark IV.1. *An interesting characteristic of our DSTFT formulation is that all the derived partial derivative expressions retain the same structural form as the forward DSTFT. This implies that each derivative can be computed by performing a DSTFT operation using a modified tapering function (either $\partial_\theta\omega$ or $\partial_x\omega$).*

C. Backpropagation Formulas

The partial derivative expressions derived in the previous subsections enable the computation of the gradient of any almost everywhere smooth differentiable scalar loss function with respect to the DSTFT parameters. To minimize such a loss function, gradient descent techniques can be employed in conjunction with the backpropagation algorithm [46]. Backpropagation provides an efficient method for calculating the gradient of a scalar loss function with respect to the parameters of a given function by recursively applying the chain rule. While widely utilized in deep learning [20] for training neural networks, its applicability extends to computing partial derivatives of various functions.

Let $\mathcal{L}$ be an almost everywhere smooth differentiable scalar loss function that depends on the STFT output $\mathcal{S}$, which is a function of the parameter set Ω. We can express this as $\mathcal{L} \circ \mathcal{S}(\Omega)$, where $\circ$ denotes the composition of functions. To optimize $\mathcal{L}$ with respect to the DSTFT parameters, we can use gradient descent. The following derives the analytical expressions for backpropagation through the STFT.

First, we note that if we consider $\mathcal{L}$ as a function of $\mathcal{S}(\Omega)$, a matrix of $\mathbb{C}^{M\times N}$, then $\mathcal{L}$ is a function from $\mathbb{C}^{M\times N}$ in $\mathbb{R}$, and its derivative with respect to $\mathcal{S}(\Omega)$, denoted by $\partial_{\mathcal{S}}\mathcal{L}(\mathcal{S}(\Omega))$, is a linear operator from $\mathbb{C}^{M\times N}$ to $\mathbb{R}$, defined as:

$$\partial_{\mathcal{S}}\mathcal{L}(\mathcal{S}(\Omega))(C) = \sum_{m,n} \frac{\partial \mathcal{L}(\mathcal{S}(\Omega))}{\partial(\mathcal{S}(\Omega))_{m,n}} C_{m,n} \in \mathbb{R},\ \forall C \in \mathbb{C}^{M\times N}. \tag{17}$$

In the general case of a window length $\theta_{m,n}$ varying with both frequency and time, we obtain the gradient with respect to the window length as:

$$\begin{aligned}\partial_{\theta_{m,n}}(\mathcal{L}\circ\mathcal{S})(\Omega) &= \partial_{\mathcal{S}}\mathcal{L}(\mathcal{S}(\Omega))\partial_{\theta_{m,n}}\mathcal{S}(\Omega)\\ &= \sum_{k,p} \frac{\partial \mathcal{L}(\mathcal{S}(\Omega))}{\partial(\mathcal{S}(\Omega))_{k,p}}(\partial_{\theta_{m,n}}\mathcal{S}(\Omega))_{k,p}\\ &= \frac{\partial \mathcal{L}(\mathcal{S}(\Omega))}{\partial(\mathcal{S}(\Omega))_{m,n}}\mathcal{S}_{\partial_\theta\omega}(t_n, m, \theta_{m,n}).\end{aligned} \tag{18}$$

Similarly, the gradient with respect to the temporal position t_n is:

$$\begin{aligned}\partial_{t_n}(\mathcal{L}\circ\mathcal{S})(\Omega) &= \partial_{\mathcal{S}}\mathcal{L}(\mathcal{S}(\Omega))\partial_{t_n}\mathcal{S}(\Omega)\\ &= \sum_{k,p} \frac{\partial \mathcal{L}(\mathcal{S}(\Omega))}{\partial(\mathcal{S}(\Omega))_{k,p}}(\partial_{t_n}\mathcal{S}(\Omega))_{k,p}\\ &= -\sum_{k} \frac{\partial \mathcal{L}(\mathcal{S}(\Omega))}{\partial(\mathcal{S}(\Omega))_{k,n}}\mathcal{S}_{\partial_x\omega}(t_n, k, \theta_{k,n}),\end{aligned} \tag{19}$$

For the time-varying hop length H_n, we have the relationship derived earlier:

$$\partial_{H_n}(\mathcal{L}\circ\mathcal{S})(\Omega) = \partial_{\mathcal{S}}\mathcal{L}(\mathcal{S}(\Omega))\partial_{H_n}\mathcal{S}(\Omega). \tag{20}$$

When the window length is time-only varying, θ_n, the gradient is:

$$\begin{aligned}\partial_{\theta_n}(\mathcal{L}\circ\mathcal{S})(\Omega) &= \partial_{\mathcal{S}}\mathcal{L}(\mathcal{S}(\Omega))\partial_{\theta_n}\mathcal{S}(\Omega)\\ &= \sum_{k} \frac{\partial \mathcal{L}(\mathcal{S}(\Omega))}{\partial(\mathcal{S}(\Omega))_{k,n}}\mathcal{S}_{\partial_\theta\omega}(t_n, k, \theta_n).\end{aligned} \tag{21}$$

When the window length is only frequency varying, θ_m, the gradient is:

$$\begin{aligned}\partial_{\theta_m}(\mathcal{L}\circ\mathcal{S})(\Omega) &= \partial_{\mathcal{S}}\mathcal{L}(\mathcal{S}(\Omega))\partial_{\theta_m}\mathcal{S}(\Omega)\\ &= \sum_{p} \frac{\partial \mathcal{L}(\mathcal{S}(\Omega))}{\partial(\mathcal{S}(\Omega))_{m,p}}\mathcal{S}_{\partial_\theta\omega}(t_p, m, \theta_m).\end{aligned} \tag{22}$$

In the classical STFT case with constant window length θ and constant hop length H, the gradients are:

$$\begin{aligned}\partial_{\theta}(\mathcal{L}\circ\mathcal{S})(\Omega) &= \partial_{\mathcal{S}}\mathcal{L}(\mathcal{S}(\Omega))\partial_{\theta}\mathcal{S}(\Omega)\\ &= \sum_{k,p} \frac{\partial \mathcal{L}(\mathcal{S}(\Omega))}{\partial(\mathcal{S}(\Omega))_{k,p}}\mathcal{S}_{\partial_\theta\omega}(t_p, k, \theta),\end{aligned} \tag{23}$$

and

$$\begin{aligned}\partial_{H}(\mathcal{L}\circ\mathcal{S})(\Omega) &= \partial_{\mathcal{S}}\mathcal{L}(\mathcal{S}(\Omega))\partial_{H}\mathcal{S}(\Omega)\\ &= -\sum_{k,p} \frac{\partial \mathcal{L}(\mathcal{S}(\Omega))}{\partial(\mathcal{S}(\Omega))_{k,p}}p\mathcal{S}_{\partial_x\omega}(t_p, k, \theta_{k,p}).\end{aligned} \tag{24}$$

Remark IV.2. *These analytical expressions provide considerable flexibility, allowing for the differentiation of any scalar function $\mathcal{L}$ of the DSTFT outputs with respect to its tuning parameters. Notably, all backpropagation computations can be efficiently implemented using matrix multiplications, with the involved matrices having the same dimensions as those in the forward propagation. This leads to faster gradient calculations with exact values, offering an advantage over traditional automatic differentiation tools.*

Remark IV.3. *These backpropagation formulas enable the use of gradient descent optimization for minimizing any differentiable cost function. Unlike previous methods such as adaptive STFT, which typically evaluate the cost function over a predefined discrete set of window sizes to select the optimal one, our approach employs gradient-based optimization algorithms like stochastic gradient descent to directly minimize the loss. It is important to acknowledge that the inherent challenges of convergence and convexity associated with gradient-based methods in machine learning remain applicable.*

V. Computational Aspects

A. Numerical Implementation

To implement the DSTFT numerically, we address the infinite summation by leveraging the finite support of the tapering function $\omega(x,\theta)$, which is non-zero only for $x \in [-L/2, L/2]$. This allows us to restrict the summation over $k \in \mathbb{Z}$ to this finite interval. To handle non-integer temporal positions t_n, we decompose t_n into its integer part $\lfloor t_n \rfloor$ and its fractional part $\{t_n\} = t_n - \lfloor t_n \rfloor$. This leads to the following practical expression for the DSTFT:

$$\begin{aligned}&\mathcal{S}_\omega(t_n, m, \theta_{m,n}) = \sum_{k\in\mathbb{Z}} \omega(k - t_n, \theta_{m,n}) s[k] e^{-\frac{2j\pi k m}{L}}\\ &= \sum_{k=-L/2+1}^{L/2} \omega(k - \{t_n\}, \theta_{m,n}) s\left[\lfloor t_n \rfloor + k\right] e^{-\frac{2j\pi}{L}(k+\lfloor t_n\rfloor)m}\\ &= e^{-\frac{2j\pi(\lfloor t_n \rfloor)m}{L}}\\ &\sum_{k=-L/2+1}^{L/2} \omega(k - \{t_n\}, \theta_{m,n}) s\left[\lfloor t_n \rfloor + k\right] e^{-\frac{2j\pi k m}{L}}.\end{aligned} \tag{25}$$

This formulation effectively considers the signal samples centered around the integer time index $\lfloor t_n \rfloor$ and accounts for the fractional offset $\{t_n\}$ through a shift in the argument of ω and a phase factor $e^{-j2\pi(\lfloor t_n \rfloor)m/L}$ in the exponential term. The summation limits are set from $-L/2+1$ to $L/2$ because $0 \le \{t_n\} < 1$, and the tapering function ω is zero at the boundaries of its support. Consequently, the analysis window function remains a continuous and differentiable function of real-valued window lengths and temporal frame positions, and we evaluate this analysis window on discrete time indices.

Within the differentiability framework established in the preceding section, we assumed a fixed size for the spectrogram, resulting in a constant number N of time frames (columns). However, for a fixed-overlap implementation of the DSTFT where the number of time frames adapts to the window length, the number of columns with temporal overlap with the signal can vary. To accommodate this variability while adhering to the previously defined differentiability framework, we employ zero-padding of the input signal. This ensures that all windows potentially encompassing at least one original sample are considered in the analysis. This approach allows for an STFT with a number of time frames that adjusts based on the chosen time resolution, analogous to the classical STFT when maintaining a constant overlap ratio as the window length changes. Specifically, a differentiable fixed-overlap DSTFT with a time-varying window length can be defined by setting the initial time frame $t_0 = 0$ and subsequent time frame positions as $t_n = t_{n-1} + \alpha\theta_{n-1}$ for $n > 0$, where α represents the constant overlap factor.

The codes and experimental results are available on our GitHub repository: https://github.com/maxime-leiber/dstft.

B. Computational Complexity

For any set of parameters Ω, the DSTFT output $\mathcal{S}(\Omega)$ can be computed in $O(NL^2)$ operations. However, when the window length depends only on the time index ($\theta_{m,n} = \theta_n$) or is constant ($\theta_{m,n} = \theta$), the Fast Fourier Transform (FFT) can be utilized at each time index, reducing the computational cost to $O(NL \log L)$, similar to the classical STFT. This efficient computational complexity makes the time-varying window length approach particularly advantageous for real-time applications requiring fast processing.

Regarding the gradient computation, as detailed in Sec. IV-C, the backward pass has a computational complexity equivalent to the forward pass: $O(NL \log L)$ when using the FFT, and $O(NL^2)$ otherwise. This is because all expressions in Sec. IV-C involve the term $\partial_{\mathcal{S}}\mathcal{L}(\mathcal{S}(\Omega))$, which is computed once, and DSTFTs based on differentiated window functions, which have the same complexity as the original DSTFT.

To provide a direct comparison with classical discrete optimization techniques, let A denote the cardinality of the hop-length set and B that of the window lengths used in discrete optimization approach, and let P be the number of iterations in the gradient descent technique. Table I summarizes the approximate computational complexity of these two types of methods for various scenarios.

TABLE I
COMPARISON WITH DISCRETE OPTIMIZATION APPROACH

Case	Discrete approaches	DSTFT-based
Constant	$O(ABNL \log L)$	$O(2PNL \log L)$
Time-varying	$O((AB)^N NL \log L)$	$O(2PNL \log L)$
TF-varying	$O(A^N B^{NM} NL^2)$	$O(2PNL^2)$

As shown in Table I, the computational complexity of DSTFT-based approach scales with the number of gradient iterations P. In contrast, the discrete optimization method requires evaluating the cost function for every combination of candidate parameters within the defined search space. This leads to an exponential increase in complexity for time-varying and, particularly, TF varying parameters, often rendering such exhaustive searches computationally prohibitive or even infeasible.

This comparison highlights a key advantage of our DSTFT framework. For the case of constant window and hop lengths, the computational complexity is comparable to that of discrete search methods. More significantly, for the more complex and practical scenarios of time-varying and TF varying parameters, our DSTFT-based approach offers a computationally tractable alternative to discrete search algorithms.

VI. Applications: Representation-Driven Optimization

In the following section, we will explore applications of DSTFT-based approach, in three different optimization problems, aiming at enhancing the TF representation in different contexts. The first two examples are on simulated signals and involve respectively TF varying window length and time varying hop and window lengths. Though these examples were already studied in [16] and in [17], more details are provided here on relevant algorithmic choice and on implementation. Furthermore, the novel formalism detailed in previous sections provides a clearer view than the algorithmic approaches proposed in [16] and [17]. Finally, an novel illustration of the proposed framework on vibration signals conclude the section.

A. Time and Frequency Varying Window Length

We consider a simulated signal composed of three components: a non-stationary component, a stationary component that is close in frequency to the first component for a certain duration, and a transient component. Our focus in this example is on optimizing the window length while maintaining a constant hop length. All simulations in this study employ Hann windows.

First, we illustrate the limitations of using a single, fixed window length. When the STFT is computed with a short window, it provides good temporal localization but poor frequency resolution, particularly for the transient event. Additionally, the non-stationary and stationary components are not well distinguished (see the first row of Fig. 1). Conversely, a long window allows for better separation of the non-stationary and stationary components but obscures the transient event (see the second row of Fig. 1).

To address these limitations, we first propose optimizing the DSTFT with a single window length parameter, $\Omega = \theta$, by minimizing the Shannon entropy loss:

$$\mathcal{E} \circ \mathcal{S}(\theta) = -\sum_{m,n} p_{m,n} \log(p_{m,n}), \tag{26}$$

where $p_{m,n} = \frac{|\mathcal{S}_\omega(t_n,m,\theta)|}{\sum_{k,l} |\mathcal{S}_\omega(t_l,k,\theta)|}$. The Shannon entropy $\mathcal{E}$ is differentiable with respect to $\mathcal{S}(\theta)$ provided that $\mathcal{S}_\omega(t_n, m, \theta)$ is non-zero for all indices m and n. The Shannon entropy criterion is commonly used in the literature as it is related to minimizing interference in the TF plane [47]. Other entropy measures, such as Rényi entropy could also be considered [48]. With a fixed hop length H and a signal of length L_s, the number of time frames is set to $N = 1 + \lfloor \frac{L_s}{H} \rfloor$. The optimization process converges to a spectrogram (displayed in the third row of Fig. 1) that is optimal according to the entropy criterion. In this specific case, numerical analysis (see

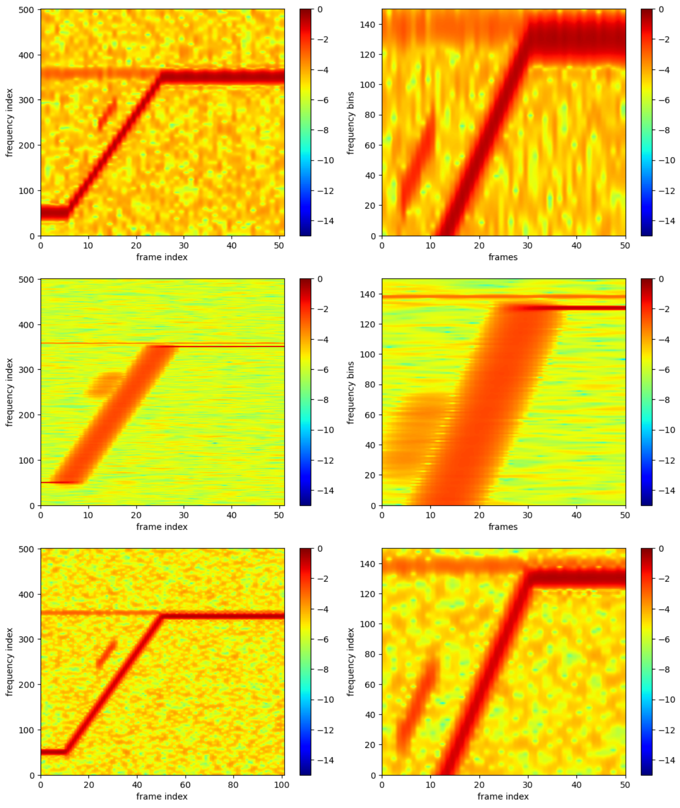

Fig. 1. Spectrograms (left) and zoomed-in spectrograms (right) with respectively from top to bottom small window of length 100, long window of length 1000, and constant-window DSTFT.

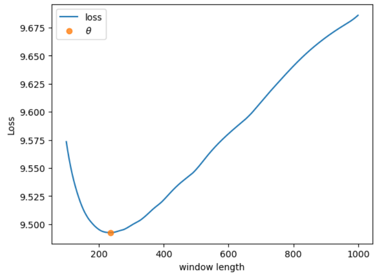

Fig. 2. Loss function corresponding to Eq. (26) with respect to window length

Fig. 2) suggests that the loss function is convex with respect to θ, ensuring the uniqueness of the optimum. Future work will focus on rigorously proving the convexity of this loss function.

Regarding computational cost, a grid-search approach with a window length ranging from 100 to 1000 samples would require evaluating 901 discrete window lengths ($B = 901$). In contrast, the DSTFT optimization typically converges within P iterations, where P depends on the choice of hyperparameters (e.g., learning rate, stopping condition) and the initial window length. Empirically, P usually ranges from 10 to 100. Considering that each iteration involves a forward and backward pass, the computational cost is roughly $2P$ times that of a single STFT computation. Thus, in the worst case, the DSTFT optimization requires approximately 200 forward/backward passes, which is significantly less than the 901 evaluations required by grid search. Furthermore, the DSTFT optimization can find optimal window lengths that are not restricted to a discrete grid, potentially leading to more accurate results.

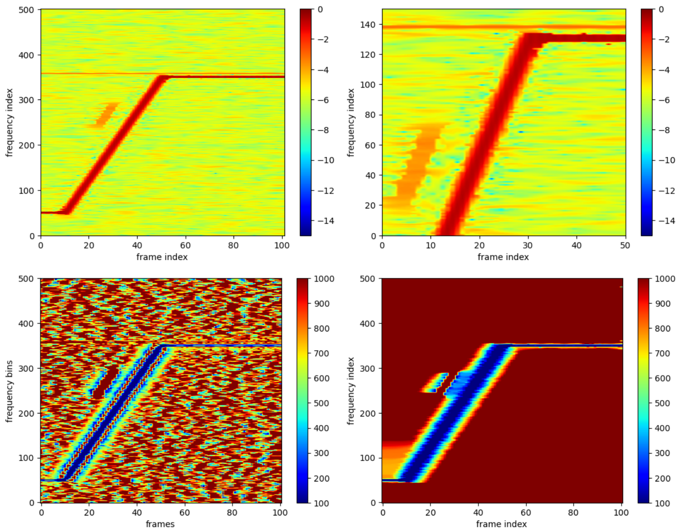

Fig. 3. Top row: Spectrogram (left) and its zoomed-in section (right) computed with a TF varying window length. Bottom row: Visualization of the associated window length distribution, illustrating the effect of the regularization term (left: without, right: with).

Next, we investigate the potential for further improvement in TFR by allowing the window length to vary with both time and frequency. For the considered simulated signal, finding a single constant window length that optimally represents all its diverse components in the spectrogram appears challenging. We therefore consider $\Omega = (\theta_{m,n})_{m,n}$ as the set of parameters to optimize the DSTFT. In this case, the Shannon entropy loss $\mathcal{E} \circ \mathcal{S}(\Omega)$ is no longer necessarily convex. We thus consider minimizing the following criterion:

$$\tilde{\mathcal{L}}(\Omega) = \mathcal{E} \circ \mathcal{S}(\Omega) + \lambda \mathcal{R}(\Omega), \tag{27}$$

where $\mathcal{E}$ is the Shannon entropy, $\mathcal{R}(\Omega)$ is a regularization term that encourages neighboring windows in the TF plane to have similar lengths, thereby enhancing robustness to noise, and λ is an hyperparameter controlling the trade-off between these two terms. The rationale for using such a regularization term is that as $\lambda \to +\infty$, the optimal solution should approach a constant window length, for which the loss function has been observed to be convex (as illustrated in Fig. 2). In our simulations, we considered the following regularization term:

$$\mathcal{R}(\Omega) = \sum_{n,m} \sqrt{\sum_{(n',m')\in N_{n,m}} (\theta_{n,m} - \theta_{n',m'})^2}, \tag{28}$$

where $N_{n,m}$ is a set of indices corresponding to the neighbors of the bin (n, m) in the TF plane. Note that $\mathcal{R}$ is differentiable with respect to Ω, and the gradient of $\tilde{\mathcal{L}}(\Omega)$ can be computed using the chain rule for the entropy term and direct differentiation for the regularization term. This regularization term is related to non-local total variation penalization, commonly used in image processing for its robustness to noise [49].

Results obtained with the DSTFT using a TF varying window length are shown in the first row of Fig. 3 for the spectrograms. The corresponding window length values, without and with the regularization term ($\lambda = 10^{-3}$), are displayed in the second row of Fig. 3. This clearly demonstrates the necessity of using a regularization term to obtain meaningful optimal window lengths. This approach allows for precise localization of all signal components in both time and frequency, significantly enhancing the overall quality of the TFR by adapting the window length to both transient and stationary characteristics of the signal.

Regarding the computational cost for a TF varying window length, a grid-search approach involving window lengths ranging from 100 to 1000 samples would require testing $B = 901^{101\times501}$ combinations, which is computationally intractable. In contrast, our DSTFT approach requires $2P$ iterations, where P depends on hyperparameters such as the learning rate, stopping condition, the regularization parameter, and the initial window lengths. To facilitate convergence towards a global optimum, a practical strategy is to initialize the window lengths with the optimal constant window length obtained in the previous convex optimization step. Empirically, P typically ranges from 50 to 300, resulting in a computational cost of at most 600 times that of a single forward pass, which is negligible compared to the cost of grid search. Furthermore, the proposed optimization technique is not restricted to a discrete set of window lengths, potentially leading to more accurate TFRs. To the best of our knowledge, this work presents a novel approach for computing TFRs associated with TF varying window lengths.

B. *Time-Varying Window and Hop-Lengths*

This section investigates the optimization of the DSTFT when both window and hop lengths are varied, using a simulated signal. In this scenario, the set of parameters is $\Omega = (t_n, \theta_n)_n$. To demonstrate the benefits of optimizing the DSTFT with respect to these parameters, we focus on signals containing transient events (shocks) with varying frequencies and durations, as depicted in the first row of Fig. 4. Gaussian white noise is added to the signal to achieve a signal-to-noise ratio (SNR) of 20 dB.

For such signals, employing frames uniformly localized along the time axis is suboptimal. Regardless of the chosen window length, energy leakage is observed in the spectrogram, as illustrated in the second, third, and fourth rows of Fig. 4. We therefore propose optimizing the window positions and lengths in the DSTFT using gradient descent. However, this optimization requires careful consideration to ensure that the set of translated windows used to compute the spectrogram adequately covers the original signal. With this in mind, to promote energy concentration within each time frame, we aim to maximize the kurtosis of the frame spectrum [27] while also incorporating a regularization term to ensure adequate signal coverage. This leads to the following loss function:

$$\tilde{\mathcal{L}}(\Omega) = \mathcal{K} \circ \mathcal{S}(\Omega) + \lambda \mathcal{C}(\Omega), \tag{29}$$

in which:

$$\mathcal{K} \circ \mathcal{S}(\Omega) = \frac{1}{N} \sum_n \frac{\mathbb{E}_m[|\mathcal{S}(t_n, m, \theta_n)|^4]}{\mathbb{E}_m[|\mathcal{S}(t_n, m, \theta_n)|^2]^2}. \tag{30}$$

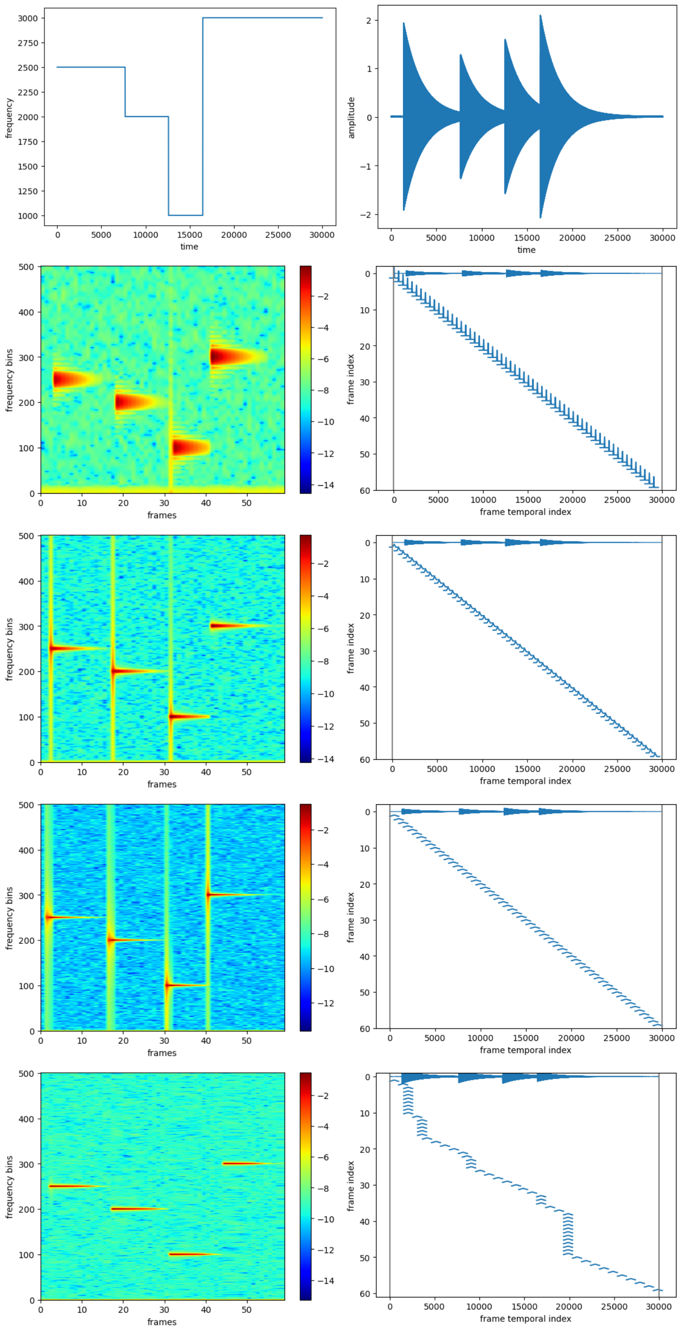

Fig. 4. In the first row, frequencies (left) and temporal signal (right). Spectrograms (left) and frame temporal position (right) with respectively from second row to bottom small window of length 100, medium window of length 400, long window of length 1000, DSTFT with time-varying window and hop length.

The number of time frames N is initially determined by choosing an arbitrary fixed hop length H and setting $N = 1 + \lfloor \frac{L_s}{H} \rfloor$, where L_s is the length of the signal. The penalization term $\mathcal{C}(\Omega)$ is introduced to ensure that the set of adaptive windows provides sufficient coverage of the entire signal. A condition for adequate coverage is that the overlap between successive windows should satisfy:

$$t_{n+1} - t_n \leq \frac{\theta_{n+1} + \theta_n}{2}. \tag{31}$$

To encourage maximum spectrogram coverage, we consider the following penalization term:

$$\begin{aligned}\mathcal{C}(\Omega) =& \frac{1}{L_s}\sum_{n=0}^{N-1} \min(t_{n+1} - t_n, \frac{\theta_{n+1} + \theta_n}{2}) \\ & 1_{x<L_s}(t_{n+1} - \frac{\theta_{n+1}}{2}) 1_{x>0}(t_n + \frac{\theta_n}{2}).\end{aligned} \tag{32}$$

When $\mathcal{C}(\Omega)$ is significantly smaller than 1, it indicates that the set of translated windows does not fully cover the signal. Note that $\mathcal{C}(\Omega)$ is differentiable with respect to Ω, anywhere in the parameter space. To better understand the motivation for this penalization term, Fig. 5 illustrates the overlapping constraints between two successive windows.

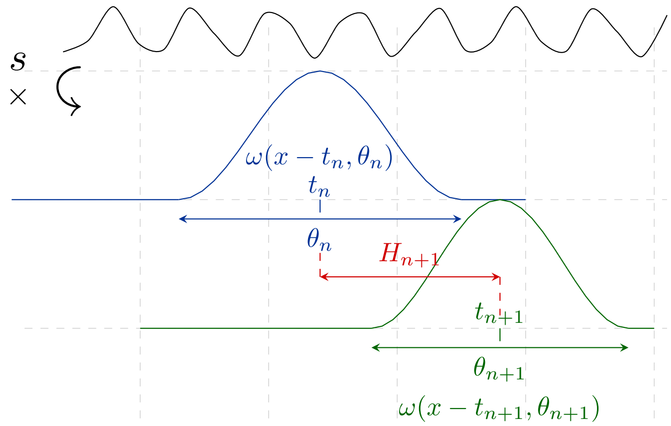
s
$\times$
$\omega(x - t_n, \theta_n)$
t_n
θ_n
H_{n+1}
t_{n+1}
θ_{n+1}
$\omega(x - t_{n+1}, \theta_{n+1})$

Fig. 5. The position of the tapering windows can smoothly shift along the time axis, while the window supports start at the integer part of the temporal position of the tapering windows.

As shown in the fifth row of Fig. 4, appropriate time positioning of the frames leads to better energy concentration due to reduced spectral leakage. Observing the time axis of the resulting TFR, one can also notice that the distribution of the frames along the time axis is no longer uniform.

C. *Window Length Optimization for Frequency Tracking from a Vibration Signal*

In this section, we consider optimizing the window length in DSTFT when applied to a real-world multi-harmonic vibration signal obtained from an aircraft engine. The primary objective is to estimate the rotational speed of the shaft, which corresponds to the main harmonic frequency, using techniques such as ridge tracking in the TF plane, leveraging the signal's multi-harmonic nature as described in [3]. A key challenge in this task is to locally determine an appropriate window length that can accurately track rapid frequency variations while maintaining sufficient frequency resolution. It is worth noting that the studied multi-harmonic signal is synthesized by considering multiples of the fundamental frequencies recorded during an actual aircraft flight, thus providing access to the ground truth of the main harmonic frequency. Additionally, white Gaussian noise is added to the signal to simulate realistic conditions.

In this example, our goal is to optimize the DSTFT, specifically focusing on time-varying window lengths, by minimizing the Shannon entropy loss. We then use the resulting TFR to estimate the main harmonic frequency. The first row of Fig. 6 displays the optimal spectrogram obtained using a single, constant window length (left) and the optimal spectrogram obtained using a time-varying window length (right), both achieved by minimizing the Shannon entropy. The second row of Fig. 6 shows zoomed-in views of these spectrograms in non-stationary regions, clearly illustrating the benefits of employing a time-varying window length during the optimization process for enhanced resolution.

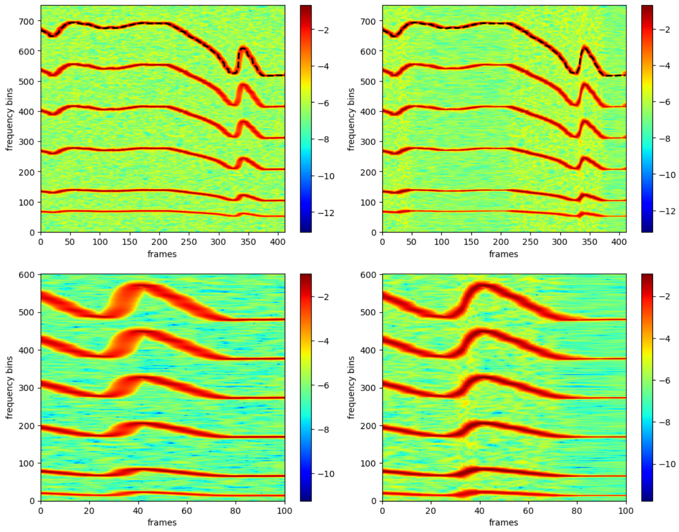

Fig. 6. Single window (left) and time-varying window (right) Spectrograms using DSTFT with respectively from top to bottom spectrogram with estimated instantaneous frequency, and zoomed-in spectrogram.

Enhancing the TFR is crucial in numerous vibration spectrogram-based applications, such as instantaneous frequency estimation. To highlight the advantages of TFR enhancement through adaptive window length in the time domain, we apply the instantaneous frequency tracking method proposed in [3] to both the spectrogram optimized with a single window and the spectrogram optimized with a time-varying window. The first row of Fig. 6 displays the tracked instantaneous frequencies for the tenth harmonic (dashed line). Dividing these tracked frequencies by 10 provides an estimate of the instantaneous frequency of the main harmonic. When a single window is used in the optimization, the mean square error of the estimated instantaneous frequency of the main harmonic is 7.05. In contrast, when a time-varying window length is employed, the mean square error is significantly reduced to 2.86, as indicated in the last row of Fig. 6. This demonstrates the effectiveness of optimizing the window length in time for improved frequency tracking accuracy in vibration analysis.

VII. Applications: Task-Driven Optimization

This section introduces the concept of task-driven optimization using the DSTFT, where the optimization aims to minimize a performance metric on a dataset relevant to a specific task. This contrasts with the earlier examples that focused on representation-driven optimization on individual signals based on TFR criteria. This section will illustrate this concept with two applications: joint optimization with frequency tracking and joint optimization with a neural network.

A. Window Length Optimization for Frequency Tracking from a Vibration Signal

In this experiment, the objective shifts from optimizing the window length for individual signals to determining a single, global window length that yields accurate frequency estimation on average across a dataset of signals. The training data for this task consists of J synthetic variable-period sine waves corrupted by additive white noise. The goal is to find the window length θ that minimizes the mean squared error between the estimated and true frequencies over the entire training dataset, as defined by the following loss function:

$$\mathcal{L} \circ (\hat{y}_j(\mathcal{S}(\theta)), \bar{y}_j) = \frac{1}{J}\sum_j \|\hat{y}_j(\mathcal{S}(\theta)) - \bar{y}_j\|^2 , \tag{33}$$

where J is the number of signals in the training dataset, $\bar{y}_j$ is the ground truth frequency of the j^{th} signal of the training data and $\hat{y}_j(S(\theta))$ is a differentiable estimation of the frequency from the spectrogram of the j^{th} signal computed with a window length θ. Note that the temporal positioning of the frames are fixed according to a predetermined hop-length H. For simplicity, we consider the frequency estimate for the j^{th} signal at each time frame n as a weighted average of the frequency bins:

$$\hat{y}_j(S(\theta)) = \left(\sum_m \frac{|\mathcal{S}^j_\omega(t_n, m, \theta)|}{\sum_l |\mathcal{S}^j_\omega(t_n, l, \theta)|} m\Delta f\right)_n , \tag{34}$$

where Δf is the frequency resolution. We employ the gradient descent algorithm to minimize the loss function in Eq. (33) with respect to θ. Upon convergence, the window length reaches a value that minimizes the loss. The first row of Fig. 7 shows the loss function as a function of the window length. The second row of Fig. 7 illustrates the resulting estimated angular speed on a sample from the simulated dataset, and the third row displays the spectrogram of that sample obtained using the optimized window length, which appears well-suited for the frequency tracking task. This simple simulation demonstrates the effectiveness of DSTFT-based backpropagation optimization for task-driven parameter learning. This approach can be generalized to any signal processing task that can be formulated as the minimization of a loss function defined with respect to the spectrogram and some target data. Indeed, one can simply replace the standard spectrogram computation with the DSTFT-based spectrogram and then optimize the window parameters by minimizing the task-specific loss function using gradient descent based on the derived backpropagation formulas.

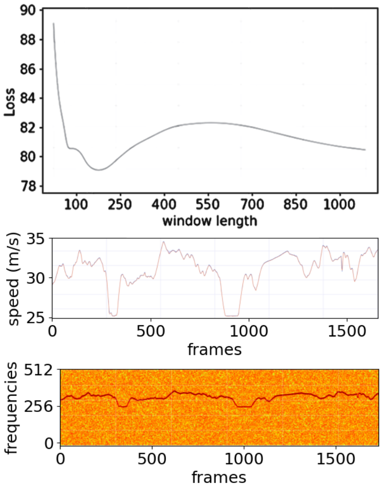

Fig. 7. Loss per window length (top), angular speed associated with optimal window length for a particular sample (middle), and the corresponding spectrogram (bottom).

B. Joint Optimization with a Neural Network

This experiment investigates the advantage of optimizing the window length in DSTFT within a neural network training process. This approach contrasts with conventional methods where the STFT parameters are typically fixed before the spectrogram is used as input to the network [4], [6]–[8], [50]. Traditional methods often involve generating spectrograms with fixed parameters chosen through heuristics or discrete optimization techniques. By treating the DSTFT as a differentiable layer with a trainable real-valued parameter (the window length in this case), we can optimize this parameter alongside the network weights to minimize a task-specific loss function in an end-to-end manner. This allows the TFR to adapt to the requirements of the classification task through global optimization.

In this experiment, the DSTFT is integrated as the initial layer of a Convolutional Neural Network (CNN) designed for spoken digit classification using the Free Spoken Digit Dataset (FSDD), also known as Audio MNIST. This dataset comprises 3000 recordings of spoken digits from 6 speakers. We randomly partitioned the dataset into 80% for training, 10% for validation, and 10% for testing. The primary goal is to jointly optimize the window length of the spectrogram generated by the DSTFT and the weights of the subsequent CNN layers. For a classification task with C=10 classes (digits 0-9), the optimization is performed by minimizing the cross-entropy loss between the CNN's predicted digit labels and the ground truth labels over the entire training dataset. For each sample j, the loss is defined as:

$$\mathcal{L}(y_j, \hat{y}_j) = -\sum_c y^c_j \log(\hat{y}^c_j), \tag{35}$$

where y^c_j is a one-hot encoded ground truth label (i.e., $y^c_j = 1$ if the true label for the j^{th} sample is c, and 0 otherwise), and $\hat{y}_j = (\hat{y}^c_j)_{c=1,\ldots,C}$ is the predicted probability of the j^{th} input signal belonging to the c^{th} digit class, as output by the CNN classifier. The prediction process can be viewed as a function where the input audio signal x_j is first transformed into a magnitude spectrogram using the DSTFT with a learnable window length θ, and this spectrogram is then fed into the CNN classifier F with learnable weights w:

$$\hat{y}_j = F_w\left(|\mathcal{S}_\theta(x_j)|\right). \tag{36}$$

The optimization of both θ and w is performed using the Adam optimizer, a gradient-based algorithm that requires the computation of the partial derivatives of the loss function with respect to both sets of parameters ($\partial_\theta \mathcal{L}$ and $\partial_w \mathcal{L}$), which can be derived using the chain rule and the differentiability of the DSTFT (as detailed in Sec. IV).

To evaluate the impact of the window size, a 2-layer CNN with 16 filters per layer was trained. Depthwise separable convolutions were used to balance computational efficiency and model accuracy. The network architecture also included ReLU activation functions for the hidden layers, dropout for regularization to prevent overfitting, and a final dense layer to produce the probability distribution over the ten digit classes. The entire system, from the input audio to the digit prediction, can be seen as a single neural network where the first layer is the DSTFT (with the window length as a trainable parameter), followed by the convolutional layers and the dense layer. Table II shows the test loss of this CNN when trained with spectrograms computed using different fixed window lengths, highlighting the sensitivity of the performance to this parameter. Table III presents the corresponding classification accuracies in percentage. As shown in Table III, the classification accuracy varies significantly with different fixed window lengths, further emphasizing the importance of this parameter. Notably, when the window length is jointly optimized with the network weights using the DSTFT, the model achieves a higher test accuracy (80.7%) compared to the best fixed window length case (79.7% at window length 40). While the presented results demonstrate the effectiveness of the proposed approach, higher classification accuracies could be achieved on this task by employing larger and more complex neural network architectures with a greater number of parameters. The relatively small CNN used in this experiment (2 layers with 16 filters per layer) was intentionally chosen to highlight the benefit of optimizing the window length parameter.

Current standard practices in similar tasks often involve training separate neural networks for a range of fixed STFT window sizes and subsequently selecting the network that yields the best performance on a validation set. Our proposed method offers a more efficient alternative by jointly optimizing the window length and the network parameters in a single end-to-end training process. To conclude, these two simple simulations demonstrate the effectiveness of our backpropagation procedure based on DSTFT. They illustrate a general window length tuning methodology applicable to any existing signal processing algorithm or neural network involving spectrograms: replace the standard spectrogram computation step with a DSTFT-based spectrogram, and then optimize the window length using gradient descent based on the derived backpropagation formulas. Differentiable STFT enables the use of gradient-based optimization for tuning its parameters in an end-to-end manner and optimizing the entire pipeline jointly.

TABLE II
TRAINING, VALIDATION AND TESTING LOSSES OF NN WITH FIXED AND LEARNABLE WINDOW LENGTH.

approaches	window length	train loss	val loss	test loss
STFT	10	0.37	1.15	1.04
STFT	20	0.03	0.44	0.32
STFT	30	0.01	0.24	0.27
STFT	40	0.01	0.26	0.23
STFT	50	0.01	0.49	0.29
DSTFT	$\theta = 34.9$	0.01	0.20	0.22

TABLE III
TRAINING, VALIDATION AND TESTING ACCURACY (IN PERCENTAGE) OF CNN WITH FIXED AND LEARNABLE WINDOW LENGTH.

approaches	window length	train loss	val loss	test loss
STFT	10	68.0	61.3	62.0
STFT	20	73.0	75.7	74.3
STFT	30	79.3	80.3	79.7
STFT	40	80.7	80.0	79.7
STFT	50	80.7	76.7	79.0
DSTFT	$\theta = 34.9$	80.3	82.3	80.7

VIII. CONCLUSION

We have introduced a differentiable formulation of the short-time Fourier transform that enables gradient-based optimization of window lengths and temporal positions. This approach offers advantages such as adaptive and automatic tuning of parameters for STFT-based TF representations like spectrograms. The significance of this contribution lies also in its potential applications within machine learning, where differentiable models are essential for efficient optimization algorithms.

REFERENCES

[1] K. M. Stafford, C. G. Fox, and D. S. Clark, "Long-range acoustic detection and localization of blue whale calls in the northeast pacific ocean," *The Journal of the Acoustical Society of America*, vol. 104, no. 6, pp. 3616–3625, 1998.

[2] J. Huang, B. Chen, B. Yao, and W. He, "ECG arrhythmia classification using STFT-based spectrogram and convolutional neural network," *IEEE access*, vol. 7, pp. 92 871–92 880, 2019.

[3] Q. Leclère, H. André, and J. Antoni, "A multi-order probabilistic approach for instantaneous angular speed tracking debriefing of the CMMNO14 diagnosis contest," *Mechanical Systems and Signal Process.*, vol. 81, pp. 375–386, 2016.

[4] Y. Gong, Y.-A. Chung, and J. Glass, "Ast: Audio spectrogram transformer," in *Interspeech 2021*, 2021, pp. 571–575.

[5] B. Nortier, M. Sadeghi, and R. Serizel, "Unsupervised speech enhancement with diffusion-based generative models," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 12 481–12 485.

[6] J. Schlüter and S. Böck, "Improved musical onset detection with convolutional neural networks," in *2014 IEEE Int. Conf. Acoust., speech and Signal Process. (ICASSP)*. IEEE, 2014, pp. 6979–6983.

[7] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[8] A. Défossez, "Hybrid spectrogram and waveform source separation," *arXiv preprint arXiv:2111.03600*, 2021.

[9] L. R. Rabiner, R. W. Schafer *et al.*, "Introduction to digital speech processing," *Foundations and Trends® in Signal Process.*, vol. 1, no. 1–2, pp. 1–194, 2007.

[10] C. García-Ruiz, A. M. Gomez, and J. M. Martín-Doñas, "The role of window length and shift in complex-domain DNN-based speech enhancement," in *IberSPEECH 2022*, 2022, pp. 146–150.

[11] B. Barai, N. Das, S. Basu, and M. Nasipuri, "An empirical study on analysis window functions for text-independent speaker recognition," *int. Journal of Speech Technology*, vol. 26, no. 1, pp. 211–220, 2023.

[12] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends® Signal Process.*, vol. 7, no. 3-4, pp. 197–387, 2014.

[13] K. M. Prabhu, *Window functions and their applications in Signal Process.* Taylor & Francis, 2014.

[14] V. Havin and B. Jöricke, *The uncertainty principle in harmonic analysis*. Springer Science & Business Media, 2012, vol. 28.

[15] M. Leiber, A. Barrau, Y. Marnissi, and D. Abboud, “A differentiable short-time Fourier transform with respect to the window length,” in *European Signal Process. conf. (EUSIPCO)*, 2022, pp. 1392–1396.

[16] M. Leiber, Y. Marnissi, A. Barrau, and M. El Badaoui, “Differentiable adaptive short-time Fourier transform with respect to the window length,” in *IEEE int. conf. on Acoustics, Speech and Signal Process. (ICASSP)*, 2023.

[17] ——, “Differentiable short-time Fourier transform with respect to the hop length,” in *IEEE Workshop on Statistical Signal Process. (SSP)*, 2023.

[18] R. Rojas and R. Rojas, “The backpropagation algorithm,” *Neural networks: a systematic introduction*, pp. 149–182, 1996.

[19] B. J. Wythoff, “Backpropagation neural networks: a tutorial,” *Chemometrics and Intelligent Laboratory Systems*, vol. 18, no. 2, pp. 115–155, 1993.

[20] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[21] R. N. Czerwinski and D. L. Jones, “Adaptive short-time Fourier analysis,” in *IEEE Signal Process. Letters*, vol. 4, 1997, pp. 42–45.

[22] J.-Y. Lee, “Variable short-time Fourier transform for vibration signals with transients,” *Journal of Vibration and Control*, vol. 21, no. 7, pp. 1383–1397, 2015.

[23] H. K. Kwok and D. L. Jones, “Improved instantaneous frequency estimation using an adaptive short-time Fourier transform,” *IEEE Trans. on Signal Process.*, vol. 48, pp. 2964–2972, 2000.

[24] J. Zhong and Y. Huang, “Time-frequency representation based on an adaptive short-time Fourier transform,” in *IEEE Trans. on Signal Process.*, vol. 58, 2010, pp. 5118–5128.

[25] S. Pei and S. Huang, “STFT with adaptive window width based on the chirp rate,” *IEEE Trans. on Signal Process.*, vol. 60, no. 8, pp. 4065–4080, 2012.

[26] P. Balazs, M. Dörfler, F. Jaillet, N. Holighaus, and G. Velasco, “Theory, implementation and applications of nonstationary Gabor frames,” *Journal of computational and applied mathematics*, vol. 236, no. 6, pp. 1481–1496, 2011.

[27] A. Zhao, K. Subramani, and P. Smaragdis, “Optimizing short-time Fourier transform parameters via gradient descent,” in *IEEE int. conf. on Acoustics, Speech and Signal Process. (ICASSP)*, 2021, pp. 736–740.

[28] D. G. Marx and K. Gryllias, “Differentiable short-time Fourier transform window length selection driven by cyclo-stationarity,” in *Proceedings of the Annual conf. of the Prognostics and Health Management Society, PHM*, vol. 15. PHM Society, 2023, pp. 1–10.

[29] C. He, H. Shi, and J. Li, “Idsn: A one-stage interpretable and differentiable stft domain adaptation network for traction motor of high-speed trains cross-machine diagnosis,” *Mechanical Systems and Signal Process.*, vol. 205, p. 110846, 2023.

[30] K. Kodera, C. De Villedary, and R. Gendrin, “A new method for the numerical analysis of non-stationary signals,” *Physics of the Earth and Planetary Interiors*, vol. 12, no. 2-3, pp. 142–150, 1976.

[31] K. Kodera, R. Gendrin, and C. Villedary, “Analysis of time-varying signals with small BT values,” *IEEE Trans. on Acoustics, Speech, and Signal Process.*, vol. 26, no. 1, pp. 64–76, 1978.

[32] F. Auger and P. Flandrin, “Improving the readability of time-frequency and time-scale representations by the reassignment method,” *IEEE Trans. on Signal Process.*, vol. 43, no. 5, pp. 1068–1089, 1995.

[33] P. Flandrin, F. Auger, and E. Chassande-Mottin, “Time frequency reassignment: from principles to algorithms,” *Applications in time-frequency Signal Process.*, pp. 179–204, 2018.

[34] F. Auger and P. Flandrin, “Improving the readability of time-frequency and time-scale representations by the reassignment method,” *IEEE Trans. on Signal Process.*, vol. 43, no. 5, pp. 1068–1089, 1995.

[35] F. Auger, P. Flandrin, Y.-T. Lin, S. McLaughlin, S. Meignen, T. Oberlin, and H.-T. Wu, “Time-frequency reassignment and synchrosqueezing: An overview,” *IEEE Signal Process. Magazine*, vol. 30, no. 6, pp. 32–41, 2013.

[36] G. Thakur and H.-T. Wu, “Synchrosqueezing-based recovery of instantaneous frequency from nonuniform samples,” *SIAM Journal on Mathematical Analysis*, vol. 43, no. 5, pp. 2078–2095, 2011.

[37] T. Oberlin, S. Meignen, and V. Perrier, “The Fourier-based synchrosqueezing transform,” in *2014 IEEE int. conf. on acoustics, speech and Signal Process. (ICASSP)*. IEEE, 2014, pp. 315–319.

[38] G. Thakur, E. Brevdo, N. S. Fučkar, and H.-T. Wu, “The synchrosqueezing algorithm for time-varying spectral analysis: Robustness properties and new paleoclimate applications,” *Signal Process.*, vol. 93, no. 5, pp. 1079–1094, 2013.

[39] T. Oberlin, S. Meignen, and V. Perrier, “Second-order synchrosqueezing transform or invertible reassignment? towards ideal time-frequency representations,” *IEEE Trans. on Signal Process.*, vol. 63, no. 5, pp. 1335–1344, 2015.

[40] R. Behera, S. Meignen, and T. Oberlin, “Theoretical analysis of the second-order synchrosqueezing transform,” *Applied and Computational Harmonic Analysis*, vol. 45, no. 2, pp. 379–404, 2018.

[41] L. Li, H. Cai, H. Han, Q. Jiang, and H. Ji, “Adaptive short-time Fourier transform and synchrosqueezing transform for non-stationary signal separation,” *Signal Process.*, vol. 166, p. 107231, 2020.

[42] D. Fourer, F. Auger, and P. Flandrin, “Recursive versions of the Levenberg-Marquardt reassigned spectrogram and of the synchrosqueezed STFT,” in *2016 IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*. IEEE, 2016, pp. 4880–4884.

[43] L. Li, H. Cai, and Q. Jiang, “Adaptive synchrosqueezing transform with a time-varying parameter for non-stationary signal separation,” *Applied and Computational Harmonic Analysis*, vol. 49, no. 3, pp. 1075–1106, 2020.

[44] J. Shi, J. Zheng, X. Liu, W. Xiang, and Q. Zhang, “Novel short-time fractional Fourier transform: Theory, implementation, and applications,” *IEEE Trans. on Signal Process.*, vol. 68, pp. 3280–3295, 2020.

[45] Z. Zhao and G. Li, “Synchrosqueezing-based short-time fractional Fourier transform,” *IEEE Trans. on Signal Process.*, vol. 71, pp. 279–294, 2023.

[46] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[47] R. G. Baraniuk, P. Flandrin, A. J. Janssen, and O. J. Michel, “Measuring time-frequency information content using the Rényi entropies,” *IEEE Trans. on Information theory*, vol. 47, no. 4, pp. 1391–1409, 2002.

[48] S. Meignen, M. Colominas, and D. Pham, “On the use of Rényi entropy for optimal window size computation in the short-time Fourier transform,” in *IEEE int. conf. on Acoustics, Speech and Signal Process. (ICASSP)*, 2020, pp. 5830–5834.

[49] G. Gilboa and S. Osher, “Nonlocal operators with applications to image processing,” *Multiscale Modeling & Simulation*, vol. 7, no. 3, pp. 1005–1028, 2009.

[50] A. M. Badshah, J. Ahmad, N. Rahim, and S. W. Baik, “Speech emotion recognition from spectrograms with deep convolutional neural network,” in *2017 Int. Conf. Platform Technology and Service (PlatCon)*. IEEE, 2017, pp. 1–5.