

# A LEARNING-BASED FRAMEWORK FOR LINE-SPECTRA SUPER-RESOLUTION

Gautier Izacard<sup>\*‡</sup>    Brett Bernstein<sup>†</sup>    Carlos Fernandez-Granda<sup>\*†</sup>

<sup>\*</sup> Center for Data Science, NYU, New York, USA

<sup>‡</sup> Ecole Polytechnique, Paris, France

<sup>†</sup> Courant Institute of Mathematical Sciences, NYU, New York, USA

## ABSTRACT

We propose a learning-based approach for estimating the spectrum of a multisinusoidal signal from a finite number of samples. A neural-network is trained to approximate the spectra of such signals on simulated data. The proposed methodology is very flexible: adapting to different signal and noise models only requires modifying the training data accordingly. Numerical experiments show that the approach performs competitively with classical methods designed for additive Gaussian noise at a range of noise levels, and is also effective in the presence of impulsive noise.

**Index Terms**— Line-spectra estimation, spectral super-resolution, deep learning, neural networks, sparse noise.

## 1. INTRODUCTION

### 1.1. Super-resolution of Spectral Lines

Estimating the spectra of multisinusoidal signals from a finite number of noisy samples is a fundamental problem in signal processing, with applications in sonar, radar, communications, geophysics, speech analysis, and other domains (see [1] for an extensive list of references). Consider a signal  $S : \mathbb{R} \rightarrow \mathbb{C}$  given by

$$S(t) := \sum_{j=1}^m a_j \exp(i2\pi f_j t), \quad (1)$$

where  $a \in \mathbb{C}^m$  contains the amplitudes of the different sinusoidal components. The Fourier transform of such signals is a superposition of Dirac deltas, or *spectral lines*, located at the frequencies  $f_1, f_2, \dots, f_m \in \mathbb{R}$ . Our goal is to estimate these frequencies from a *finite* number of noisy samples obtained at the Nyquist rate. Assuming (without loss of generality) that  $0 \leq f_j \leq 1$ ,  $1 \leq j \leq m$ , so that the Nyquist rate is equal to one, the data are given by

$$y_k := S(k) + z_k, \quad 1 \leq k \leq n, \quad (2)$$

where  $z_k \in \mathbb{C}$  is an additive perturbation. The line-spectra estimation problem is often referred to as *spectral super-resolution* because truncating the signal in time is equivalent to convolving the line spectra with a blurring sinc kernel of width  $2/n$ . As a result, the spectral resolution of the data is limited by the number of available samples even in the absence of noise.

### 1.2. Contributions

Inspired by recent advances in deep learning [2], we present a learning-based methodology to tackle the line-spectra estimation problem that involves calibrating a deep neural network using simulated data. Training the neural network is costly but can be carried out offline. However, once the model is trained, processing new data is significantly faster than applying parametric or variational techniques. An insight underlying the design of the network is that estimating frequency locations directly is less effective than generating a smoothed estimate of the spectrum, as demonstrated in Section 4.2. In Section 4.3 the approach is shown to perform competitively with classical methods in a range of signal-to-noise ratios (SNRs), matching the robustness of linear nonparametric estimators at low SNR, as well as the accuracy of parametric estimators at high SNR. Finally, Section 4.4 illustrates the flexibility of our framework with an application to line-spectra estimation in the presence of sparse noise.

## 2. STATE OF THE ART AND RELATED WORK

The simplest method to estimate line spectra is to window the data and then compute its Fourier transform. This technique, known as the periodogram, provides a linear nonparametric estimate composed of a superposition of sinc kernels of width  $1/2n$  centered at the position of the line spectra. The interference between these kernels complicates locating the line spectra precisely. As a result, the periodogram does not yield an exact estimate of the spectrum, even in the absence of noise. However, when the SNR is low, the windowed periodogram is an effective estimator (see Chapter 2 in [3]).

In contrast to the periodogram, Prony's method (see [4], as well as [5] for a modern exposition) is guaranteed to re-

---

B.B. is supported by a MacCracken, and a Barkey and Yhap fellowship. C.F. is supported by NSF award DMS-1616340.

cover line spectra exactly from noiseless measurements. Parametric methods based on Prony’s method are very popular and highly effective at high and medium SNRs. They include techniques based on matrix pencils [6], and subspace methods such as MUSIC [7, 8]. These approaches are significantly more computationally heavy than the periodogram because they require computing eigendecompositions of matrices built from the samples.

Recently, variational techniques based on sparse recovery have been proposed for the line-spectra estimation problem [9, 10]. This approach is computationally intensive, as it requires solving a semidefinite program or a convex program involving a large dictionary of discretized sinusoids. An important advantage is that it can easily be adapted to deal with missing data [11] and outliers [12].

Finally, from a methodological point of view, our work is related to recent neural-network based approaches to sparse recovery [13, 14], point-source deconvolution [15], and acoustic source localization [16, 17, 18].

### 3. METHODOLOGY

We propose to perform line-spectra super-resolution using a deep neural network, which we call pseudo-spectrum net (PSnet). The input to the network are data generated according to the model in equation 1. The output of the network is an approximation to the spectrum, called a *pseudo-spectrum*. Numerical experiments reported in Section 4.2 show that generating such an approximation is more effective than trying to train a network to output the frequency locations of the spectral lines directly. A similar phenomenon has been observed when calibrating small deep-learning models to approximate larger networks: approximating the softmax outputs of the larger model produces better results than trying to fit its discrete predictions [19].

We define the pseudo-spectrum  $P_f : \mathbb{R} \rightarrow \mathbb{R}$  as the convolution of the spectral lines of the signal with a kernel  $K : \mathbb{R} \rightarrow \mathbb{R}$ :

$$P_f(u) := \sum_{i=1}^m K(u - f_i), \quad (3)$$

where  $f$  is a vector containing  $m$  frequencies. In all our numerical experiments,  $K$  is a triangular kernel. The PSnet is a neural network  $PS_w : \mathbb{C}^n \rightarrow \mathbb{R}^g$  parametrized by linear weights  $w$  that outputs a fine discretization of the pseudospectrum on a grid of size  $g$ . The network treats complex numbers as pairs of real numbers, which makes it possible to leverage standard optimization packages to train it. The weights are calibrated by using the Adam optimizer [20] to minimize the approximation error between the output of the network and the discretized pseudospectrum on a training set with  $N$  ex-

amples:

$$w := \min_{\tilde{w}} \sum_{l=1}^N \left\| P_{f^{[l]}} - F_{\tilde{w}}(y^{[l]}) \right\|_2^2. \quad (4)$$

The  $l$ th training example is given by

$$y_k^{[l]} := \sum_{j=1}^m a_j^{[l]} \exp(i2\pi f_j^{[l]} k) + z_k^{[l]}, \quad 1 \leq k \leq n, \quad 1 \leq l \leq N,$$

where the frequencies  $f^{[l]}$ , the amplitudes  $a^{[l]}$  and the noise  $z^{[l]}$  are sampled from predefined distributions. Once the PSnet is calibrated, the position of the line spectra for a new vector of noisy data  $y$  can be estimated by locating the peaks of the corresponding estimated pseudospectrum  $PS_w(y)$ .

The architecture of the PSnet consists of a linear layer followed by several convolutional layers and a final linear layer. The layers are separated by rectified linear units (ReLU), a standard non-linearity in deep learning, and include batch normalization [21]. Intuitively, the first layer maps the data to a frequency representation (one can check that the rows of the corresponding matrices are sinusoidal). In the frequency domain, the contribution of each spectral line to the data is concentrated around it and displays translation invariance: shifting the line just shifts its corresponding component in the data. This motivates using convolutional layers, which consist of localized filters that are convolved with the input, to build the rest of the network. In computer vision, convolutional layers are a fundamental tool for exploiting translation invariance [22].

### 4. NUMERICAL EXPERIMENTS

In this section, we provide numerical evidence that the PSnet generalizes effectively on test data not present in the training set used to calibrate the model. In all experiments, we train the network on a variable number of frequencies, i.e., the number of spectral lines is not fixed in the training set.

#### 4.1. Experimental Design

In our experiments, the training and test sets are generated by sampling the frequency locations, amplitudes and noise in the measurement model of equation 2 independently at random. The coefficients are given by  $a_j := (0.1 + |w_j|)e^{i\theta_j}$ ,  $j = 1, \dots, m$ , where  $w_j$  is standard Gaussian distribution and  $\theta_j$  is uniform in  $[0, 2\pi]$ . In all sections except 4.4  $z_l$  is standard Gaussian noise scaled to ensure a given SNR.

In order to design an appropriate distribution for the frequencies of the spectral lines, it is necessary to take into account that the minimum separation  $\Delta_{\min} := \min_{j \neq j'} |f_j - f_{j'}|$  between them determines whether the problem is well posed. At minimum separations below  $\Delta_{\min} := 1/n$  the problem is severely ill posed, in the sense that estimating the

| SNR   | 1     |       | 10    |       | 100   |       |
|-------|-------|-------|-------|-------|-------|-------|
|       | FN    | MD    | FN    | MD    | FN    | MD    |
| Pair. | 52.7% | 0.478 | 19.4% | 0.286 | 23.4% | 0.331 |
| DL    | 38.3% | 0.433 | 13.1% | 0.229 | 21.1% | 0.294 |
| PS    | 15.8% | 0.137 | 11.1% | 0.099 | 15.1% | 0.122 |

**Table 1:** Comparison of the average false-negative rate (FN) and matched-distance error (MD) normalized by  $1/n$  for the experiment described in Section 4.2. The neural networks trained using the pairing (Pair.) and the DeepLoco (DL) losses are outperformed by the network trained to output a pseudo-spectrum (PS) over the three different SNRs.

amplitudes requires solving a linear system that is very ill conditioned even if the true frequencies are known [23]. To ensure that the training set contains well posed instances the inter-frequency separations of each signal are given by  $f_{j+1} - f_j = (\text{sign}(u_j) \Delta + u_j) \bmod 1$  for  $j = 1, \dots, m-1$  where the  $u_j$  are i.i.d. draws from a centered Gaussian distribution with standard deviation  $2.5/n$  and  $\Delta > 1/n$ . Finally,  $n$  is fixed to 50 and the number of frequencies  $m$  is chosen uniformly between 1 and 10.

In order to produce an estimate of the frequencies from the pseudospectrum generated by the network we locate the highest  $m$  peaks. To measure recovery accuracy we use two metrics: false-negative rate, and matched distance. The false negative rate is defined by

$$\text{FN}(f, \hat{f}) = \frac{1}{m} \sum_{j=1}^m \mathbf{1} \left( \{\hat{f}_1, \dots, \hat{f}_m\} \cap [f_j - \frac{1}{2n}, f_j + \frac{1}{2n}] = \emptyset \right).$$

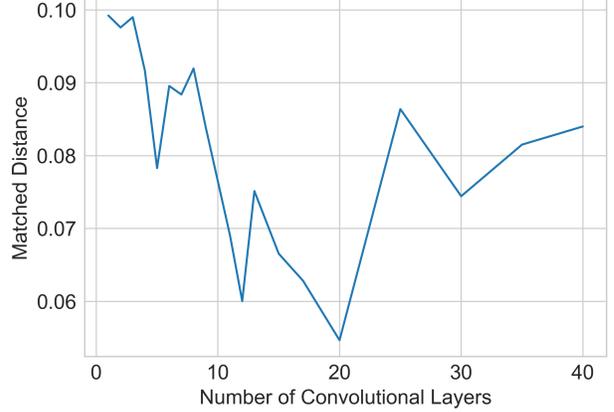
In words, a false negative occurs when there is no estimated frequency that is closer than  $\frac{1}{2n}$  to a true frequency. The matched distance between the true frequencies and the estimate, denoted by  $\hat{f}_1, \dots, \hat{f}_m$ , is given by

$$\text{MD}(f, \hat{f}) = n \cdot \text{avg} \{ \text{avg}_j \min_k |f_j - \hat{f}_k|, \text{avg}_k \min_j |f_j - \hat{f}_k| \}.$$

In words, we match each frequency with its closest counterpart, and record the average error, normalized by  $1/n$ . To remove the influence of large errors that are accounted for by the false negative rate, we only average over frequencies where the closest counterpart is within  $\frac{1}{2n}$ .

## 4.2. Comparison to Direct Estimation of Frequencies

Our proposed methodology is based on producing a pseudo-spectrum from which to estimate spectral-line locations. In this section we compare this choice to the alternative approach of training a neural network to directly output the frequency estimates  $\hat{f}_1, \dots, \hat{f}_m$ . This requires a careful choice of the training loss used to calibrate the network. A natural approach is to associate each frequency of the signal to an element of the output using the minimal pairing distance over



**Fig. 1:** Plot of validation set performance as a function of network depth at SNR 100. Each layer of the network consists of 8 circular convolution filters (with filter size 3), batch normalization, and a ReLU non-linearity as described in Section 3.

all possible permutations  $\mathfrak{S}$ ,

$$\min_{\sigma \in \mathfrak{S}} \sum_{j=1}^m |f_j - \hat{f}_{\sigma(j)}|^2. \quad (5)$$

Recent work on point-source deconvolution [15] introduces an alternative loss, where the distance between the estimated and the true frequencies is computed after smoothing with a kernel (e.g. a Laplacian or Gaussian kernel). This approach is closer to pseudo-spectrum estimation; it computes a pseudo-spectrum that is parametrized by the estimated frequencies. In contrast, our methodology produces a nonparametric estimate of the pseudo-spectrum.

To compare direct frequency estimation with our proposed methodology we use the same architecture to perform direct estimation and to estimate a pseudo-spectrum. We fix the architecture to be a fully connected network with 9 hidden layers, the first of which contains 5000 neurons and the rest of which contain 500 neurons. Empirically, this seems to yield the best results for the direct-estimation losses. By adding a last linear layer with an output of dimension  $m$ , the network can be trained to produce frequency estimates using the minimal pairing distance and the DeepLoco loss. By adding a last layer with an output of dimension  $g := 10^3$  it can be trained using our methodology to produce an estimate of the pseudo-spectrum.

A complication that arises when performing direct estimation is how to output a variable number of estimated frequencies. Our approach does not suffer from this problem; a varying number of spectral lines simply results in a different number of peaks in the estimated pseudo-spectrum. However, here we consider a simple case where  $m$  is fixed and equal to two ( $m := 2$ ). Table 1 compares the performance of these three different options at three different SNRs (1, 10,

| SNR         | 1      |       | 10     |       | 100    |       | Blind  |       |
|-------------|--------|-------|--------|-------|--------|-------|--------|-------|
|             | FN     | MD    | FN     | MD    | FN     | MD    | FN     | MD    |
| MUSIC       | 38.04% | 0.144 | 8.79%  | 0.054 | 3.02%  | 0.087 | 4.95%  | 0.032 |
| Periodogram | 30.14% | 0.120 | 13.73% | 0.091 | 13.36% | 0.087 | 15.47% | 0.089 |
| PSnet       | 27.04% | 0.143 | 2.62%  | 0.063 | 2.10%  | 0.054 | 2.63%  | 0.061 |

**Table 2:** Comparison of the average false-negative rate (FN) and matched-distance error (MD) normalized by  $1/n$  for the experiments described in Section 4.2.

and 100). At each SNR the training and test sets contain  $10^4$  signals generated as described in Section 4.1 with  $\Delta := 2/n$ . Our results suggest that generating a pseudo-spectrum significantly outperforms direct estimation of frequencies. Designing an architecture to produce accurate frequency estimates directly is an interesting direction for future research.

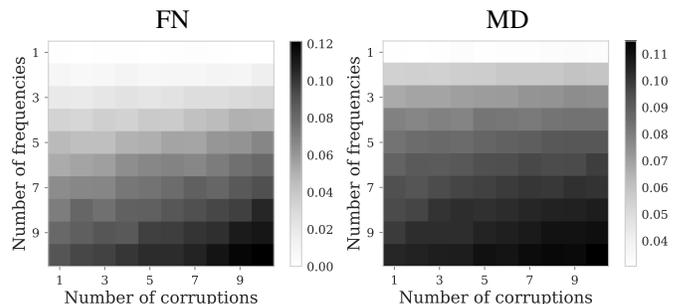
### 4.3. Comparison to Traditional Methods

In this section we compare the performance of the PSnet to two of the main traditional methods for line-spectra estimation: the periodogram [3], and MUSIC [7, 8] (see Section 2). We train PSnets with the architecture detailed in Section 3 on data generated as described in Section 4.1 with  $\Delta := 1/n$  for a range of SNRs. Figure 1 shows the performance of the network in terms of matched distance error for different depths. For the comparison with other methods, we set the number of convolutional layers at 20, each with 8 filters (each of size 3), and the dimension of the initial linear layer to 100.

The results of the comparison are shown in Table 2. We calibrate a different PSnet on training data with an SNR of 1, 10 and 100. In addition, we train a single PSnet for a *blind-noise* scenario where the noise level is not known beforehand by varying the SNR of the signals in the training and test sets (the SNR is uniformly sampled between 1 and 100). In the high noise regime (SNR 1) the PSnet and the periodogram have similar performance, while MUSIC has a considerably larger false negative rate. In the lower noise regimes (SNR 10 and 100) the PSnet and MUSIC outperform the periodogram, with the PSnet having the lowest false negative rate. In the blind-noise regime, the PSnet again outperforms both other methods in terms of false-negative rate and is competitive with MUSIC in matched-distance error.

### 4.4. Line-Spectra Estimation from Corrupted Data

A promising feature of learning-based methods is that they can easily incorporate prior assumptions on the measurements. In this section we consider the problem of performing line-spectra estimation when a subset of the data are completely corrupted, i.e., when the vector  $z$  in equation 2 is sparse. In particular, we consider a regime where the corruptions have a standard deviation on the same order as the amplitude of the sampled signal, so they produce significant perturbations while being challenging to detect.



**Fig. 2:** Comparison of the average false-negative rate (FN) and matched-distance error (MD) normalized by  $1/n$  for the experiment described in Section 4.4.

To evaluate the performance of our network we generate training and test sets with  $2 \cdot 10^4$  examples, where each example has between 1 and 10 spectral lines. The data are simulated as described in Section 4.1 with  $\Delta := 1/n$ , except for the noise. The noise is set to have a support with fixed cardinality ranging from 1 to 10 (i.e., up to 20% of the measurements). Its amplitude is iid Gaussian with a standard deviation equal to  $1/2$  (for reference the  $\ell_2$  norm of the signal is normalized). The network architecture follows the description in Section 3; the dimensionality of the linear layer is 500, the number of convolutional layers is 20, and the number of filters per layer is 8. Figure 2 shows the results: the proposed methodology produces a small rate of false negatives and is very accurate.

## 5. CONCLUSION AND FUTURE WORK

Our results suggest that learning-based methods are a promising avenue for tackling signal-processing problems such as line-spectra estimation. An important difference between these approaches and traditional methods is that their performance depends on the probabilistic assumptions encoded in the training set. This is an attractive feature, as it makes it straightforward to adapt the approach to different signal and noise models. However, it also presents a crucial challenge for future research: understanding under what conditions training on simulated measurements ensures robust generalization to real data.

## 6. REFERENCES

- [1] Petre Stoica, “List of references on spectral line analysis,” *Signal Processing*, vol. 31, no. 3, pp. 329–340, 1993.
- [2] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436, 2015.
- [3] Petre Stoica and R. L. Moses, *Spectral analysis of signals*, Prentice Hall, Upper Saddle River, New Jersey, 1 edition, 2005.
- [4] Baron Gaspard Riche de Prony, “Essai expérimental et analytique: sur les lois de la dilatabilité de fluides élastique et sur celles de la force expansive de la vapeur de l’alkool, à différentes températures,” *Journal de l’École Polytechnique*, vol. 1, no. 22, pp. 24–76, 1795.
- [5] Martin Vetterli, Pina Marziliano, and Thierry Blu, “Sampling signals with finite rate of innovation,” *IEEE Trans. on Signal Processing*, vol. 50, no. 6, pp. 1417–1428, 2002.
- [6] Y Hua and T.K Sarkar, “Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 5, pp. 814–824, May 1990.
- [7] G. Bienvenu, “Influence of the spatial coherence of the background noise on high resolution passive methods,” in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 1979, vol. 4, pp. 306 – 309.
- [8] R. Schmidt, “Multiple emitter location and signal parameter estimation,” *IEEE Trans. on Antennas and Propagation*, vol. 34, no. 3, pp. 276 – 280, 1986.
- [9] Badri Narayan Bhaskar, Gongguo Tang, and Benjamin Recht, “Atomic norm denoising with applications to line spectral estimation,” *IEEE Trans. on Signal Processing*, vol. 61, no. 23, pp. 5987–5999, 2013.
- [10] Emmanuel J Candès and Carlos Fernandez-Granda, “Towards a mathematical theory of super-resolution,” *Communications on Pure and Applied Mathematics*, vol. 67, no. 6, pp. 906–956, 2014.
- [11] G.T. Tang, B. N. Bhaskar, P. Shah, and B. Recht, “Compressed sensing off the grid,” *IEEE Trans. on Information Theory*, vol. 59, no. 11, pp. 7465–7490, 2013.
- [12] Carlos Fernandez-Granda, Gongguo Tang, Xiaodong Wang, and Le Zheng, “Demixing sines and spikes: Robust spectral super-resolution in the presence of outliers,” *Information and Inference*, vol. 7, no. 1, pp. 105–168, 2017.
- [13] Bo Xin, Yizhou Wang, Wen Gao, David Wipf, and Baoyuan Wang, “Maximal sparsity with deep networks?,” in *Advances in Neural Information Processing Systems*, 2016, pp. 4340–4348.
- [14] Hao He, Bo Xin, Satoshi Ikehata, and David Wipf, “From Bayesian sparsity to gated recurrent nets,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5554–5564.
- [15] Nicholas Boyd, Eric Jonas, Hazen P Babcock, and Benjamin Recht, “DeepLoco: Fast 3D localization microscopy using neural networks,” *BioRxiv*, p. 267096, 2018.
- [16] Sharath Adavanne, Archontis Politis, and Tuomas Virtanen, “Direction of arrival estimation for multiple sound sources using convolutional recurrent neural network,” *arXiv preprint arXiv:1710.10059*, 2017.
- [17] Xiong Xiao, Shengkui Zhao, Xionghu Zhong, Douglas L Jones, Eng Siong Chng, and Haizhou Li, “A learning-based approach to direction of arrival estimation in noisy and reverberant environments,” in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 2015, pp. 2814–2818.
- [18] Soumitro Chakrabarty and Emanuël AP Habets, “Broadband DOA estimation using convolutional neural networks trained with noise signals,” in *Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2017, pp. 136–140.
- [19] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean, “Distilling the knowledge in a neural network,” *arXiv preprint arXiv:1503.02531*, 2015.
- [20] Diederik P Kingma and Jimmy Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [21] Sergey Ioffe and Christian Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” *arXiv preprint arXiv:1502.03167*, 2015.
- [22] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [23] Ankur Moitra, “Super-resolution, extremal functions and the condition number of Vandermonde matrices,” in *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC)*, 2015.