Title

Super-Resolution With Binary Priors: Theory and Algorithms

Authors

Sarangi, Pulak

Hattori, Ryoma

Komiyama, Takaki

et al.

Publication Date

2023

DOI

10.1109/tsp.2023.3260564

Peer reviewed

Published in final edited form as:

IEEE Trans Signal Process. 2023 ; 71: 3213–3228. doi:10.1109/tsp.2023.3260564.

Super-resolution with Binary Priors: Theory and Algorithms

Pulak Sarangi,
Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92092
USA

Ryoma Hattori,
Neurobiology Section, Center for Neural Circuits and Behavior, Department of Neurosciences, and Halıcıoğlu Data Science Institute, University of California, San Diego, La Jolla, CA 92092
USA

Takaki Komiyama,
Neurobiology Section, Center for Neural Circuits and Behavior, Department of Neurosciences, and Halıcıoğlu Data Science Institute, University of California, San Diego, La Jolla, CA 92092
USA

Piya Pal
Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92092
USA

Abstract

The problem of super-resolution is concerned with the reconstruction of temporally/spatially localized events (or spikes) from samples of their convolution with a low-pass filter. Distinct from prior works which exploit sparsity in appropriate domains in order to solve the resulting ill-posed problem, this paper explores the role of binary priors in super-resolution, where the spike (or source) amplitudes are assumed to be binary-valued. Our study is inspired by the problem of neural spike deconvolution, but also applies to other applications such as symbol detection in hybrid millimeter wave communication systems. This paper makes several theoretical and algorithmic contributions to enable binary super-resolution with very few measurements. Our results show that binary constraints offer much stronger identifiability guarantees than sparsity, allowing us to operate in “extreme compression” regimes, where the number of measurements can be significantly smaller than the sparsity level of the spikes. To ensure exact recovery in this “extreme compression” regime, it becomes necessary to design algorithms that exactly enforce binary constraints without relaxation. In order to overcome the ensuing computational challenges, we consider a first order auto-regressive filter (which appears in neural spike deconvolution), and exploit its special structure. This results in a novel formulation of the super-resolution binary spike recovery in terms of binary search in one dimension. We perform numerical experiments that validate our theory and also show the benefits of binary constraints in neural spike deconvolution from real calcium imaging datasets.

Index Terms—

Binary compressed sensing; super-resolution; spike deconvolution; sparsity; binary search; beta-expansions

I. Introduction

The problem of recovering localized events (spikes) from their convolution with a blurring kernel, arises in a wide range of scientific and engineering applications such as fluorescence microscopy [1], neural spike deconvolution [2]-[4], hybrid millimeter wave (mmWave) communication [5], to name a few. Consider K temporal spikes, which can be represented as:

$$x_{\text{hi}}\left(t\right) = \sum_{k=1}^{K} c_k \delta(t - n_k T_{\text{hi}})$$

Here, the high-rate spikes are supported on a fine temporal grid with spacing T_{hi}, $n_k \in \mathbb{Z}$ is an integer corresponding to the time index of the k^{th} spike and c_k denotes its amplitude. The convolution of spikes with a filter $h(t)$ is typically uniformly (down)sampled at a (low) rate $T_{\text{lo}} = \text{D}T_{\text{hi}}(\text{D} > 1)$, yielding measurements:

$$y[n] = x_{\text{hi}}(t) \star h(t)|_{t = nT_{\text{lo}}} = \sum_{k=1}^{K} c_k h(nT_{\text{lo}} - n_k T_{\text{hi}}) \tag{1}$$

The goal of super-resolution is to recover the spike locations n_k and amplitudes $c_k, k = 1,2, \cdots, K$ from a limited number (M) of low-rate samples $\{y[n]\}_{n=0}^{M-1}$. The problem is typically illposed due to systematic attenuation of high-frequency contents of the spikes by the low-pass filter $h(t)$. In order to make the problem well-posed, it becomes necessary to exploit priors such as sparsity [6]-[9] and/or non-negativity [10], [11]. In recent times, there has been a substantial progress towards developing efficient algorithms for provably solving the super-resolution problem [7]-[19].

In this paper, we investigate the problem of *binary super-resolution*, where the amplitudes of the spikes are known apriori to be $c_k = A$, but their number (K) and locations (n_k) are unknown. Motivated by the problem of neural spike deconvolution in two-photon calcium imaging [2], [20], we will focus on a blurring kernel that can be represented as a stable first order auto-regressive (AR(1)) filter. Each neural spike results in a sharp rise in Ca^{2+} concentration followed by a slow exponential decay (modeled as the impulse response of an AR(1) filter), which results in an overlap of the responses from nearby spiking events, leading to poor temporal resolution [2], [21].

A. Related Works

Early works on super-resolution date back to algebraic/subspace-based techniques such as Prony's method, MUSIC [12], [22], ESPRIT [8], [23] and matrix pencil [9], [24]. Following the seminal work in [6], substantial progress has been made in understanding the role of sparsity as a prior for super-resolution [7], [25], [26]. In recent times, convex optimization-based techniques have been developed that employ Total Variational (TV) norm and atomic norm regularizers, in order to promote sparsity [7], [18], [19], [25], [26] and/or non-negativity [10], [11], [27]. These techniques primarily employ sampling in the Fourier/frequency domain by assuming the kernel $h(t)$ to be (approximately) bandlimited. However, selecting the appropriate cut-off frequency is crucial for super-resolution and needs careful consideration [25], [28]. Unlike subspace-based methods, theoretical guarantees for these convex algorithms rely on a minimum separation between the spikes, which is also shown to be necessary even in absence of noise [29]. The finite rate of innovation (FRI) framework [30]-[34] also considers the recovery of spikes from measurements acquired using an exponentially decaying kernel, which includes the AR(1) filter considered in this paper. In the absence of noise, FRI enables the exact recovery of K spikes with *arbitrary amplitudes* from $M = \Omega(K)$[1] measurements, without any separation condition [32]. It is to be noted that all of the above methods require $M > K$ measurements for resolving K spikes. In contrast, we will show that it is possible to recover K spikes from $M \ll K$ measurements by exploiting the *binary nature of the spiking signal*. The above algorithms are designed to handle *arbitrary real-valued amplitudes* and as such, they are oblivious to binary priors. Therefore, they cannot successfully recover spikes in the regime $M < K$, which is henceforth referred to as the *extreme compression regime*.

The problem of recovering binary signals from underdetermined linear measurements (with more unknowns than equations/measurements) has been recently studied under the parlance of Binary Compressed Sensing (BCS) [35]-[42]. In BCS, the undersampling operation employs random (and typically dense) sampling matrices, whereas we consider a deterministic and structured measurement matrix derived from a filter, followed by uniform downsampling. Moreover, existing theoretical guarantees for BCS crucially rely on sparsity assumptions that will be shown to be inadequate for our problem (discussed in Section II-C). Most importantly, in order to achieve computational tractability, BCS relaxes the binary constraints and solves continuous-valued optimization problems. Consequently, their theoretical guarantees do not apply in the extreme compression regime $M < K$.

As mentioned earlier, our study is motivated by the problem of neural spike deconvolution arising in calcium imaging [3], [4], [20], [32], [43]-[45]. A majority of the existing spike deconvolution techniques [4], [43], [44] infer the spiking activity at *the same (low) rate at which the fluorescence signal is sampled*, and a single estimate such as spike counts or rates are obtained over a temporal bin equal to the resolution of the imaging rate. Although sequential Monte-Carlo based techniques have been proposed that generate spikes at a rate higher than the calcium frame rate [3], no theoretical guarantees are available that prove that these methods can indeed uniquely identify the high-rate spiking activity. Algorithms that

[1] This notation essentially means that there exists a positive constant c such that $M \geq cK$.

rely on sparsity and non-negativity [43], [44] alone are ineffective for inferring the neural spiking activity that occurs at a much higher rate than the calcium sampling rate. On the other hand, at the high-rate, the spiking activity is often assumed to be binary since the probability of two or more spikes occurring within two time instants on the fine temporal grid is negligible [2], [46]. Therefore, we propose to exploit the inherent binary nature of the neural spikes and provide the first theoretical guarantees that it is indeed possible to resolve the high-rate binary neural spikes from calcium fluorescence signal acquired at a much lower rate.

B. Our Contributions

We make both theoretical and algorithmic contributions to the problem of binary super-resolution in the setting when the spikes lie on a fine grid. We theoretically establish that at very low sampling rates, sparsity and non-negativity are inadequate for the exact reconstruction of binary spikes (Lemma 2). However, by exploiting the binary nature of the spiking activity, much stronger identifiability results can be obtained compared to classical sparsity-based results (Theorem 1). In the absence of noise, we show that it is possible to uniquely recover K binary spikes from only $M = \Omega(1)$ low-rate measurements. The analysis also provides interesting insights into the interplay between binary priors and the “infinite memory” of the AR(1) filter.

Although it is possible to uniquely identify binary spikes in the extreme compression regime ($M \ll K$), the combinatorial nature of binary constraints introduce computational hurdles in exactly enforcing them. Our second contribution is to leverage the special structure of the AR(1) measurements to overcome this computational challenge in the extreme compression regime $M < K$ (Section III-A). Our formulation reveals an interesting and novel connection between binary superresolution, and finding the generalized radix representation of real numbers, known as β-expansion [47]-[49] (Section III). In order to circumvent the problem of exhaustive search, we preconstruct and store (in memory) a binary tree that is completely determined by the model parameters (filter and undersampling factor). When the low-rate measurements are acquired, we can *efficiently perform a binary search* to traverse the tree and find the desired binary solution. This ability to trade-off memory for computational efficiency is made possible by the unique structure of the measurement model governed by the AR(1) filter. The algorithm guarantees exact super-resolution even when the measurements are corrupted by a small bounded (adversarial) noise, the strength of which depends on the AR filter parameter and the undersampling factor. When the measurements are corrupted by additive Gaussian noise, we characterize the probability of erroneous decoding (Theorem 3) in the extreme compression regime $M < K$ and indicate the trade-off among the filter parameter, SNR and the extent of compression. Finally, we also demonstrate how binary priors can improve the performance of a popularly used spike deconvolution algorithm (OASIS [43]) on real calcium imaging datasets.

II. Fundamental Sample Complexity of Binary Super-resolution

Let $y_{\mathrm{hi}}[n]$ be the output of a stable first-order Autoregressive AR(1) filter with parameter $\alpha, 0 < \alpha < 1$, driven by an unknown binary-valued input signal $x_{\mathrm{hi}}[n] \in \{0, A\}, A > 0$:

$$y_{\mathrm{hi}}[n] = \alpha y_{\mathrm{hi}}[n-1] + x_{\mathrm{hi}}[n] \tag{2}$$

In this paper, we consider a super-resolution setting where we do not directly observe $y_{\mathrm{hi}}[n]$, and instead acquire M measurements $\{y_{10}[n]\}_{n=0}^{M-1}$ at a lower-rate by uniformly subsampling $y_{\mathrm{hi}}[n]$ by a factor of D:

$$y_{\mathrm{lo}}[n] = y_{\mathrm{hi}}[\mathrm{D}n],\ n = 0, 1, \cdots, M-1, \tag{3}$$

The signal $y_{10}[n]$ corresponds to a filtered and downsampled version of the signal $x_{\mathrm{hi}}[n]$ where the filter is an infinite impulse response (IIR) filter with a single pole at α. Let $\mathbf{y}_{\mathrm{lo}} \in \mathbb{R}^{M}$ be a vector obtained by stacking the low-rate measurements $\{y_{10}[n]\}_{n=0}^{M-1}$:

$$\mathbf{y}_{\mathrm{lo}} = [y_{\mathrm{lo}}[0], y_{\mathrm{lo}}[1], \cdots, y_{\mathrm{lo}}[M-1]]^{\top}$$

Since (2) represents a causal filtering operation, the low rate signal y_{lo} only depends on the present and past high-rate binary signal. Denote $L: = (M-1)\mathrm{D}+1$. The M low-rate measurements in $\mathbf{y}_{10}$ are a function of L samples of the high rate binary input signal $\{x_{\mathrm{hi}}[n]\}_{n=0}^{L-1}$. These L samples are given by the following vector $\mathbf{x}_{\mathrm{hi}} \in \{0, A\}^{L}$:

$$\mathbf{x}_{\mathrm{hi}} := [x_{\mathrm{hi}}[0], x_{\mathrm{hi}}[1], \cdots, x_{\mathrm{hi}}[L-1]]^{\top}.$$

Assuming the system to be initially at rest, i.e., $y_{\mathrm{hi}}[n] = 0, n < 0$, we can represent the M samples from (3) in a compact matrix-vector form as:

$$\mathbf{y}_{\mathrm{lo}} := \mathbf{S}_{\mathrm{D}}\mathbf{y}_{\mathrm{hi}} = \mathbf{S}_{\mathrm{D}}\mathbf{G}_{\alpha}\mathbf{x}_{\mathrm{hi}} \tag{4}$$

where $\mathbf{G}_{\alpha} \in \mathbb{R}^{L \times L}$ is a Toeplitz matrix given by:

$$\mathbf{G}_{\alpha} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \alpha & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^{L-1} & \alpha^{L-2} & \cdots & 1 \end{bmatrix} \tag{5}$$

and $\mathbf{S}_{\mathrm{D}} \in \mathbb{R}^{M \times L}$ is defined as:

$$[\mathbf{S}_{\mathrm{D}}]_{i,j} = \begin{cases} 1, & j = (i-1)\mathrm{D}+1 \\ 0, & \text{else} \end{cases}.$$

The matrix $\mathbf{S}_{\mathrm{D}}$ represents the D-fold downsampling operation. Our goal is to infer the unknown high-rate binary input signal $x_{\mathrm{hi}}[n]$ from the low-rate measurements $y_{10}[n]$. This is essentially a “super-resolution” problem because the AR(1) filter first attenuates the high-frequency components of $x_{\mathrm{hi}}[n]$, and the uniform downsampling operation systematically discards measurements. As a result, it may seem that the spiking activity $\{x_{\mathrm{hi}}[(n-1)\mathrm{D}+k]\}_{k=1}^{\mathrm{D}}$ occurring “in-between” two low-rate measurements $y_{10}[n-1]$ and $y_{10}[n]$ is apparently lost. One can potentially interpolate arbitrarily, making the problem hopeless. In the next section, we will show that surprisingly, x_{hi} still remains identifiable from $\mathbf{y}_{\mathrm{lo}}$ in the absence of noise, due to the binary nature of x_{hi} and “infinite memory” of the AR(1) filter.

A. Identifiability Conditions for Binary super-resolution

Consider the following partition of $\mathbf{x}_{\mathrm{hi}}$ into M disjoint blocks, where the first block is a scalar and the remaining $M-1$ blocks are of length D, $\mathbf{x}_{\mathrm{hi}} = \left[x_{\mathrm{hi}}^{(0)}, \mathrm{x}_{\mathrm{hi}}^{(1)\top}, \ldots, \mathbf{x}_{\mathrm{hi}}^{(M-1)\top}\right]^{\top}$. Here, $x_{\mathrm{hi}}{}^{(0)} = x_{\mathrm{hi}}\left[0\right]$ and $\mathrm{x}_{\mathrm{hi}}{}^{(n)} \in \left\{0, A\right\}^{\mathrm{D}}$ is given by:

$$\left[\mathbf{x}_{\mathrm{hi}}^{(n)}\right]_{k} = x_{\mathrm{hi}}\left[\left(n-1\right)\mathrm{D}+k\right], 1 \le n \le M-1 \tag{6}$$

The sub-vectors $\mathbf{x}_{\mathrm{hi}}{}^{(n)}$, and $\mathbf{x}_{\mathrm{hi}}{}^{(n-1)}\left(n \ge 1\right)$ represent consecutive and disjoint blocks (of length D) of the high-rate binary spike signal. In order to study the identifiability of $\mathbf{x}_{\mathrm{hi}}$ from $\mathbf{y}_{\mathrm{lo}}$, we first introduce an alternative (but equivalent) representation for (4), by constructing a sequence $c[n]$ as follows $c[0] = y_{10}[0]$,

$$c\left[n\right] = y_{\mathrm{lo}}\left[n\right] - \alpha^{\mathrm{D}} y_{\mathrm{lo}}\left[n-1\right], 1 \le n \le M-1 \tag{7}$$

Given the high rate AR(1) model defined in (2), it is possible to recursively represent $y_{\mathrm{hi}}[\mathrm{D}n]$ in terms of $y_{\mathrm{hi}}[\mathrm{D}n-1]$, which in turn, can be represented in terms of $y_{\mathrm{hi}}[\mathrm{D}n-2]$, and so on. By this recursive relation, we can represent $y_{\mathrm{hi}}[\mathrm{D}n-1]$ in terms of $y_{\mathrm{hi}}[\mathrm{D}n-\mathrm{D}]$ and $\{x_{\mathrm{hi}}[\mathrm{D}n-i]\}_{i=0}^{\mathrm{D}-1}$ and re-write $y_{\mathrm{lo}}[n]$ as

$$\begin{aligned}
& y_{\mathrm{lo}}\left[n\right] = y_{\mathrm{hi}}\left[\mathrm{D}n\right] = \alpha y_{\mathrm{hi}}\left[\mathrm{D}n-1\right] + x_{\mathrm{hi}}\left[\mathrm{D}n\right] \\
& = \alpha^{\mathrm{D}} y_{\mathrm{hi}}\left[\mathrm{D}n-\mathrm{D}\right] + \alpha^{\mathrm{D}-1} x_{\mathrm{hi}}\left[\mathrm{D}\left(n-1\right)+1\right] + \cdots \\
& \ + \alpha x_{\mathrm{hi}}\left[\mathrm{D}n-1\right] + x_{\mathrm{hi}}\left[\mathrm{D}n\right], \\
& y_{\mathrm{lo}}\left[n\right] - \alpha^{\mathrm{D}} y_{\mathrm{lo}}\left[n-1\right] = \alpha^{\mathrm{D}-1} x_{\mathrm{hi}}\left[\mathrm{D}\left(n-1\right)+1\right] + \cdots \\
& \ + \alpha x_{\mathrm{hi}}\left[\mathrm{D}n-1\right] + x_{\mathrm{hi}}\left[\mathrm{D}n\right]
\end{aligned} \tag{8}$$

The last equality holds due to the fact that $y_{\mathrm{lo}}[n-1] = y_{\mathrm{hi}}[\mathrm{D}n-\mathrm{D}]$. Combining (7) and (8), the sequence $c[n]$ can be re-written as $c\left[0\right] = y_{\mathrm{lo}}\left[0\right] = x_{\mathrm{hi}}{}^{(0)}$, and for $1 \le n \le M-1$

$$c\left[n\right] = \sum_{i=1}^{D} \alpha^{D-i} x_{\text{hi}}\left[\left(n-1\right)D + i\right] = \mathbf{h}_\alpha^T \mathbf{x}_{\text{hi}}^{(n)} \tag{9}$$

where $\mathbf{h}_\alpha = \left[\alpha^{D-1}, \alpha^{D-2}, \ldots, \alpha, 1\right]^T \in \mathbb{R}^D$. This implies that $c[n]$ depends only on the block $\mathbf{x}_{\text{hi}}{}^{(n)}$. Denote $\mathbf{c} := \left[c\left[0\right], c\left[1\right], \ldots, c\left[M-1\right]\right]^\top \in \mathbb{R}^M$. For any D, (9) can be compactly represented as:

$$\mathbf{c} = \mathbf{H}_D(\alpha)\mathbf{x}_{\text{hi}} \tag{10}$$

where $\mathbf{H}_D\left(\alpha\right) \in \mathbb{R}^{M \times L}$ is given by:

$$\mathbf{H}_D\left(\alpha\right) = \begin{bmatrix} 1 & 0^\top & \mathbf{0}^\top & \cdots & \mathbf{0}^\top \\ 0 & \mathbf{h}_\alpha^\top & \mathbf{0}^\top & \cdots & \mathbf{0}^\top \\ 0 & \mathbf{0}^\top & \mathbf{h}_\alpha^\top & \cdots & \mathbf{0}^\top \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \mathbf{0}^\top & \mathbf{0}^\top & \cdots & \mathbf{h}_\alpha^\top \end{bmatrix}$$

The following Lemma establishes the equivalence between (4) and (10).

Lemma 1. *Given* $\mathbf{y}_{10}$, *construct* $\mathbf{c}$ *following* (7). *Then, there is a unique binary* $x_{\text{hi}} \in \{0, A\}^L$ *satisfying* (4) *if and only if* x_{hi} *is a unique binary vector satisfying* (10).

Proof. First suppose that there is a unique binary $x_{\text{hi}} \in \{0, A\}^L$ satisfying (4) but (10) has a non-unique binary solution, i.e., there exists $\mathbf{x}_{\text{hi}}' \in \{0, A\}^L$, $\mathbf{x}_{\text{hi}}' \neq \mathbf{x}_{\text{hi}}$, such that

$$\mathbf{c} = \mathbf{H}_D(\alpha)\mathbf{x}_{\text{hi}} = \mathbf{H}_D(\alpha)\mathbf{x}_{\text{hi}}' \tag{11}$$

Define $\mathbf{y}_{\text{hi}}' := \mathbf{G}_\alpha \mathbf{x}_{\text{hi}}'$ whose entries are given by:

$$y_{\text{hi}}'\left[n\right] = \sum_{k=0}^{n} \alpha^{n-k} x_{\text{hi}}'\left[k\right], 0 \le n \le L-1 \tag{12}$$

Notice that (7) can be re-written as

$$\begin{aligned}
&y_{\mathrm{lo}}[0] = c[0] = x_{\mathrm{hi}}[0],\, y_{\mathrm{lo}}[1] = c[1] + \alpha^{\mathrm{D}} y_{\mathrm{lo}}[0] = c[1] + \alpha^{\mathrm{D}} c[0] \\
&y_{\mathrm{lo}}[2] = c[2] + \alpha^{\mathrm{D}} y_{\mathrm{lo}}[1] = c[2] + \alpha^{\mathrm{D}} c[1] + \alpha^{2\mathrm{D}} c[0] \\
&\vdots
\end{aligned}$$

Following this recursive relation, and using (9) and (11), we can further re-write $y_{\mathrm{lo}}[n]$ as:

$$\begin{aligned}
y_{\mathrm{lo}}[n] &= \sum_{i=0}^{n} \alpha^{(n-i)\mathrm{D}} c[i] = \alpha^{n\mathrm{D}} x'_{\mathrm{hi}}{}^{(0)} + \sum_{i=1}^{n} \alpha^{(n-i)\mathrm{D}} \mathbf{h}_{\alpha}^{\top} \mathbf{x}_{\mathrm{hi}}{}'^{(i)} \\
&= \alpha^{n\mathrm{D}} x'_{\mathrm{hi}}{}^{(0)} + \sum_{i=1}^{n} \sum_{j=1}^{\mathrm{D}} \alpha^{n\mathrm{D}-(i-1)\mathrm{D}-j} x'_{\mathrm{hi}}[(i-1)\mathrm{D}+j] \\
&\overset{(a)}{=} \sum_{k=0}^{n\mathrm{D}} \alpha^{n\mathrm{D}-k} x'_{\mathrm{hi}}[k] \overset{(b)}{=} y'_{\mathrm{hi}}[n\mathrm{D}]
\end{aligned} \tag{13}$$

The equality (a) follows by a re-indexing of the summation into a single sum, and (b) follows from (12). By arranging (13) in a matrix form we obtain the following relation:

$$\mathbf{y}_{\mathrm{lo}} = \mathbf{S}_{\mathrm{D}} \mathbf{G}_{\alpha} \mathbf{x}_{\mathrm{hi}}'$$

However from (4), we have $\mathbf{y}_{\mathrm{lo}} = \mathbf{S}_{\mathrm{D}} \mathbf{G}_{\alpha} \mathbf{x}_{\mathrm{hi}}$. This contradicts the supposition that (4) has a unique binary solution.

Next, suppose that (10) has a unique binary solution but the binary solution to (4) is non-unique, i.e., there exists $\mathbf{x}_{\mathrm{hi}}' \in \{0, A\}^{L}$, $\mathrm{x}_{\mathrm{hi}}' \neq \mathrm{x}_{\mathrm{hi}}$ such that

$$\mathbf{y}_{\mathrm{lo}} = \mathbf{S}_{\mathrm{D}} \mathbf{G}_{\alpha} \mathbf{x}_{\mathrm{hi}}' = \mathbf{S}_{\mathrm{D}} \mathbf{G}_{\alpha} \mathbf{x}_{\mathrm{hi}}$$

By following (7) and (10), we also have $\mathbf{c} = \mathbf{H}_{\mathrm{D}}(\alpha)\mathbf{x}_{\mathrm{hi}}' = \mathbf{H}_{\mathrm{D}}(\alpha)\mathbf{x}_{\mathrm{hi}}$ which contradicts the assumption that solution of (10) is unique. ☐

Lemma 1 assures that a binary $\mathbf{x}_{\mathrm{hi}}$ is uniquely identifiable from measurements $\mathbf{y}_{\mathrm{lo}}$ if and only if there is a unique binary solution $\mathrm{x}_{\mathrm{hi}} \in \{0, A\}^{L}$ to (10). From (9), it can be seen that $c[n]$ and $c[n-1]$ have contributions from only disjoint blocks of high rate spikes $\mathrm{x}_{\mathrm{hi}}{}^{(n)}$, and $\mathrm{x}_{\mathrm{hi}}{}^{(n-1)}$. Hence effectively, we only have a *single scalar measurement* $c[n]$ to decode an entire block $\mathrm{x}_{\mathrm{hi}}{}^{(n)}$ of length D, regardless of how sparse it is. The task of decoding $\mathrm{x}_{\mathrm{hi}}{}^{(n)}$ from a single measurement seems like a hopelessly "ill-posed" problem, caused by the uniform downsampling operation. But this is precisely where the binary nature of x_{hi} can be used as a powerful prior to make the problem well-posed. Theorem 1 specifies conditions under which it is possible to do so.

Theorem 1. *(Identifiability) For any* $\alpha \in (0,1)$, with the possible exception of α *belonging to a set of Lebesgue measure zero, there is a unique* $\mathrm{x}_{\mathrm{hi}} \in \{0, A\}^{L}$ *that satisfies* (10) *for every* $\mathrm{D} \geq 1$.

Proof. In Appendix A. □

Using Lemma 1 and Theorem 1, we can conclude that $\mathbf{x}_{hi}$ is uniquely identifiable from $\mathbf{y}_{10}$ for almost all $\alpha \in (0,1)$. It can be verified that for $\alpha = 1$ the mapping is non-injective. Theorem 1 establishes that it is fundamentally possible to decode each block $\mathrm{x}_{hi}^{(n)}$ of length D, from effectively a single measurement $c[n]$. Since $\mathrm{x}_{hi}^{(n)}$ can take 2^{D} possible values, in principle, one can always perform an exhaustive search over these 2^{D} possible binary sequences and by Theorem 1, only one of them will satisfy $c\left[n\right] = \mathbf{h}_{\alpha}^{\top}\mathbf{x}_{hi}^{(n)}$. Since exhaustive search is computationally prohibitive, this leads to the natural question regarding alternative solutions. In Section III, we will develop an alternative algorithm that leverages the trade-off between memory and computation to achieve a significantly lower run-time decoding complexity.

B. Comparison with Finite Rate of Innovation Approach

In a related line of work [30]-[32], [34], the FRI framework has been developed to reconstruct spikes from the measurement model considered here. However, in the general FRI framework, there is no assumption on the amplitude of the spikes, and there are a total of 2D real valued unknowns corresponding to the locations and amplitudes of D spikes. In [32], it was shown that by leveraging the property of exponentially reproducing kernels, it is possible to recover arbitrary amplitudes and spike locations using Prony-type algorithms, provided at least 2D+1(> D) low-rate measurements are available. However, since we exploit the binary nature of spiking activity, we can operate at a much smaller sample complexity than FRI. In fact, Theorem 1 shows that when we exploit the fact that the spikes occur on a high-resolution grid with binary amplitudes, $M = \Omega(1)$ measurements suffice to identify D spikes regardless of how large D is. A direct application of the FRI approach cannot succeed in this regime, since the number of spikes is larger than the number of measurements. That being said, with enough measurements, FRI techniques are powerful, and they can also identify off-grid spikes. In future, it would be interesting to combine the two approaches by incorporating binary priors to FRI based techniques and remove the grid assumptions.

C. Curse of Uniform Downsampling: Inadequacy of sparsity and non-negativity

By virtue of being a binary signal, x_{hi} is naturally sparse and non-negative. Therefore, one may ask if sparsity and/or non-negativity are sufficient to uniquely identify $\mathbf{x}_{hi}$ from $\mathbf{c}$, without the need for imposing any binary constraints. In particular, we would like to understand if the solution to the following problem that seeks the sparsest non-negative vector in $\mathbb{R}^{L}$ satisfying (10) indeed coincides with the true $\mathrm{x}_{hi} \in \{0, A\}^{L}$

$$\min_{\mathbf{x} \in \mathbb{R}^{L}} \|\mathbf{x}\|_0 \text{ subject to } \mathbf{c}=\mathbf{H}_{\mathrm{D}}(\alpha)\mathbf{x},\ \mathbf{x} \geq \mathbf{0} \tag{P0}$$

Lemma 2. *For every* $\mathrm{x}_{hi} \in \{0, A\}^{L}$ *(except* $\mathrm{x}_{hi} = A\mathbf{e}_1$), and $\mathbf{c} \in \mathbb{R}^{M}$ *satisfying* (10), *the following are true*

- **i.** There exists a solution $\mathrm{x}^{\star} \neq \mathrm{x}_{hi}$ to (P0) satisfying

$$\| \mathbf{x}^{\star} \|_0 \le \| \mathbf{x}_{\mathrm{hi}} \|_0 \tag{14}$$

ii. The inequality in (14) is strict as long as there exists an integer $n_0 \ge 1$ such that the block $\mathbf{x}_{\mathrm{hi}}^{(n_0)}$ of $\mathbf{x}_{\mathrm{hi}}$ (defined in (6) satisfies $\| \mathrm{x}_{\mathrm{hi}}^{(n_0)} \|_0 \ge 2$.

Proof. The proof is in Appendix B. □

Lemma 2 shows there exist other non-binary solution(s) to (10) (different from x_{hi}) that have the same or smaller sparsity as the binary signal $\mathrm{x}_{\mathrm{hi}} \in \{0, A\}^L$. Furthermore, there exist problem instances where the sparsest solution to (P0) is strictly sparser than x_{hi}. Hence, sparsity and/or non-negativity are inadequate to identify the ground truth x_{hi} uniquely.

Implicit Bias of Relaxation: The optimization problem (P0) is non-convex and the binary constraints are not enforced. In binary compressed sensing [35], [36], it is common to relax the binary constraints using box-constraint and $\boldsymbol{l}_0$ norm is relaxed to $\boldsymbol{l}_1$ norm in the following manner:

$$\min_{\mathbf{x} \in \mathbb{R}^L} \| \mathbf{x} \|_1 \text{ subject to } \mathbf{c} = \mathbf{H}_{\mathrm{D}}\left(\alpha\right)\mathbf{x},\, 0 \le \mathbf{x} \le A1 \tag{P1-B}$$

In the following Lemma, we show that there is an implicit bias introduced to the solution of (P1-B).

Lemma 3. *For every* $\mathrm{x}_{\mathrm{hi}} \in \{0, A\}^L$, *and* $\mathbf{c} \in \mathbb{R}^M$ *satisfying* (10). *There exists a solution* $\mathrm{x}^{\star}$ *to* (P1-B) *satisfying*

$$\| \mathrm{x}^{\star} \|_1 \le \| \mathrm{x}_{\mathrm{hi}} \|_1 . \tag{15}$$

Moreover, for all $n \ge 1$, *the blocks* $\mathbf{x}^{(n)\star} \in \mathbb{R}^{\mathrm{D}}$ of $\mathbf{x}^{\star}$ *satisfy:*

$$supp\left(\mathbf{x}^{(n)\star}\right) = \{\mathrm{D}, \mathrm{D}-1, \cdots, \mathrm{D}-j_n\}, \text{if} \, c\left[n\right] \ne 0 \tag{16}$$

for some $0 \le j_n \le \mathrm{D} - 1$ *and* $\mathbf{x}^{(n)\star} = 0$ *if* $c[n] = 0$, *irrespective of the support of* x_{hi}.

Proof. The proof is in Appendix B. □

Lemma 3 shows that even in the noiseless setting, introducing the box-constraint as a means of relaxing the binary constraint introduces a bias in the support of the recovered spikes. The optimal solution always results in spikes with support clustered towards the end of each block of length D, irrespective of the ground truth spiking pattern x_{hi} that generated the measurements. This bias is a consequence of the nature of relaxation, as well as the specific structure of the measurement matrix $\mathbf{H}_{\mathrm{D}}(\alpha)$ arising in the problem.

D. Role of Memory in Super-resolution: IIR vs. FIR filters

The ability to identify the high-rate binary signal $\mathrm{x}_{\mathrm{hi}} \in \{0, A\}^L$ from D-fold undersampled measurements $\mathbf{y}_{\mathrm{lo}}$ (for arbitrarily large D) in the absence of noise, is in parts also due to the "infinite memory" or infinite impulse response of the AR(1) filter. Indeed, for an Finite Impulse Response (FIR) filter, there is a limit to downsampling without losing identifiability. This was recently studied in our earlier work [40] where we showed that the undersampling limit is determined by the length of the FIR filter. To see this, consider the convolution of a binary valued signal $\mathbf{x}_{\mathrm{hi}}$ with a FIR filter $\mathbf{u} = \left[u\left[0\right], u\left[1\right], \cdots, u\left[r-1\right]\right]^T \in \mathbb{R}^r$ of length r: $z_f\left[n\right] = \sum_{i=0}^{r-1} u\left[r-1-i\right] x_{\mathrm{hi}}\left[n+i\right]$. These samples are represented in the vector form as $\mathbf{z}_f := \mathbf{u} \star \mathbf{x}_{\mathrm{hi}} \in \mathbb{R}^L$ (by suitable zero padding). Suppose, as before, we only observe a D-fold downsampling of the output $z_{\mathrm{D}}[n] = z_f[\mathrm{D}n]$. Two consecutive samples $z_{\mathrm{D}}[p], z_{\mathrm{D}}[p+1]$ of the low-rate observation are given by:

$$z_{\mathrm{D}}[p] = \sum_{i=0}^{r-1} u[r-1-i] x_{\mathrm{hi}}[\mathrm{D}p+i],$$

$$z_{\mathrm{D}}\left[p+1\right] = \sum_{i=0}^{r-1} u\left[r-1-i\right] x_{\mathrm{hi}}\left[\mathrm{D}\left(p+1\right)+i\right]$$

If $\mathrm{D} > r$, notice that none of the measurements is a function of the samples $x_{\mathrm{hi}}[\mathrm{D}p+r], x_{\mathrm{hi}}[\mathrm{D}p+r+1], \cdots, x_{\mathrm{hi}}[\mathrm{D}(p+1)-1]$. Hence, it is possible to assign them arbitrary binary values and yet be consistent with the low-rate measurements $z_{\mathrm{D}}[n]$. This makes it impossible to exactly recover $\mathbf{x}_{\mathrm{hi}}$ (even if it is known to be binary valued) if the decimation is larger than the filter length $(\mathrm{D} > r)$. The following lemma summarizes this result.

Lemma 4. *For every FIR filter* $\mathbf{u} \in \mathbb{R}^r$, *if the undersampling factor exceeds the filter length, i.e.* $\mathrm{D} > r$, *there exist* $\mathbf{x}_0, \mathbf{x}_1 \in \{0, A\}^L, \mathbf{x}_0 \neq \mathbf{x}_1$ *such that* $\mathbf{S}_{\mathrm{D}}(\mathbf{u} \star \mathbf{x}_0) = \mathbf{S}_{\mathrm{D}}(\mathbf{u} \star \mathbf{x}_1)$.

This shows that the identifiability result presented in Theorem 1 is not merely a consequence of binary priors but the infinite memory of the autoregressive process is also critical in allowing arbitrary undersampling D > 1 in absence of noise. For such IIR filters, the memory of all past (binary) spiking activity is encoded (with suitable weighting) into every measurement captured after the spike, which would not be the case for a finite impulse response filter.

III. Efficient Binary Super-Resolution Using Binary Search with Structured Measurements

By Theorem 1, we already know that it is possible to uniquely identify x_{hi} from $\mathbf{c}$ (or equivalently, each block $\mathrm{x}_{\mathrm{hi}}^{(n)}$ from a single measurement $c[n]$) by exhaustive search.

We now demonstrate how this exhaustive search can be avoided by formulating the decoding problem in terms of “binary search” over an appropriate set, and thereby attaining computational efficiency. We begin by introducing some notations and definitions. Given a non-negative integer $k, 0 \le k \le 2^{\mathrm{D}} - 1$, let $(b_1(k), b_2(k), \cdots, b_{\mathrm{D}}(k))$ be the unique D-bit binary representation of $k : k = \sum_{d=1}^{\mathrm{D}} 2^{\mathrm{D}-d} b_d\left(k\right), b_d\left(k\right) \in \left\{0,1\right\} \forall 1 \le d \le \mathrm{D}$. Here $b_1(k)$ is the most significant bit and $b_{\mathrm{D}}(k)$ is the least significant bit. Using this notation, we define the following set:

$$\mathcal{S}_{\text{all}} := \{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{2^{\mathrm{D}}-1}\}, \tag{17}$$

where each $\mathbf{v}_k \in \{0, A\}^{\mathrm{D}}$ is a binary vector given by

$$[\mathbf{v}_k]_d = A b_d(k) \,.\, 1 \le d \le \mathrm{D} \tag{18}$$

In other words, the binary vector $\frac{1}{A}\mathbf{v}_k$ is the D-bit binary representation of its index k. Using this convention, $\mathbf{v}_0 = 0$ (i.e., a binary sequence of all 0 ’s) and $\mathbf{v}_{2^{\mathrm{D}}-1} = A1$ (i.e., a binary sequence of all A's). Recall the partition of $\mathbf{x}_{\text{hi}}$ defined in (6), where each block $\mathrm{x}_{\text{hi}}^{(n)}\left(n \ge 1\right)$ is a binary vector of length D and $x_{\text{hi}}^{(0)} \in \left\{0, A\right\}$ is a scalar. It is easy to see that (17) comprises of all possible values that each block $\mathrm{x}_{\text{hi}}^{(n)}$ can assume. According to (9) each scalar measurement $c[0] = x^{(0)}$, $c[n] = \mathrm{h}_\alpha^{\top} \mathbf{x}_{\text{hi}}^{(n)}$, $1 \le n \le M - 1$. For every α, we define the following set:

$$\Theta_\alpha := \{\theta_0, \theta_1, \cdots, \theta_{2^{\mathrm{D}}-1}\}, \text{where } \theta_k := \mathbf{h}_\alpha^{\top} \mathbf{v}_k \tag{19}$$

Observe that every measurement $c\left[n\right] = \sum_{i=1}^{\mathrm{D}} \alpha^{\mathrm{D}-i} x_{\text{hi}}\left[\left(n-1\right)D + i\right]$ takes values from this set Θ_α, depending on the value taken by the underlying block of spiking pattern from $\mathcal{S}_{\text{all}}$. Our goal is to recover the spikes $\{x_{\text{hi}}[(n-1)\mathrm{D} + i]\}_{i=1}^{\mathrm{D}}$ from $c[n]$.

In the following, we show that this problem is equivalent to finding the representation of a real number over an arbitrary radix, which is known as “β-expansion” [49]. Given a real (potentially non-integer) number $\beta > 1$, the representation of another real number $p \ge 0$ of the form:

$$p = \sum_{n=1}^{\infty} a_n \beta^{-n}, \text{where } 0 \le a_n < \left\lfloor \beta \right\rfloor \tag{20}$$

is referred to as a β-expansion of p. The coefficients $0 \le a_n < \lfloor \beta \rfloor$ are integers. This is a generalization of the representation of numbers beyond integer-radix to a system where the radix can be chosen as an arbitrary real number. This notion of representation over arbitrary radix was first introduced by Renyi in [49], and since then has been extensively studied [47], [48], [50]. There is a direct connection between β-expansion and the binary super-resolution problem considered here. In the problem at hand, any element $\theta_k \in \Theta_\alpha$ can be written as:

$$\theta_k = \mathbf{h}_\alpha^\top \mathbf{v}_k = \sum_{i=1}^{\mathrm{D}} \alpha^{\mathrm{D}-i} [\mathbf{v}_k]_i$$

When $1/2 < \alpha < 1$, by letting $\beta = 1/\alpha$, we see that the coefficients in (20) must satisfy $0 \le a_n < \lfloor 1/\alpha \rfloor < 2$, i.e., they are restricted to be binary valued $a_n \in \{0,1\}$. *Therefore, decoding the spikes* $\mathbf{v}_k$ *from the observation* θ_k *is equivalent to finding a* D*-bit representation for the number* θ_k / A *over the non-integer radix* $\beta = 1/\alpha$. Questions regarding the existence of β-expansion, and finding the coefficients of a finite β-expansion (whenever it exists) has been an active topic of research [47], [48], [50], [51]. When $\beta \ge 2$ (equivalently, $0 < \alpha \le 1/2$), it is possible to find the coefficients using a greedy algorithm which proceeds in a fashion similar to finding the D-bit binary representation of an integer [47], [51]. However, the regime $\beta \in (1,2)$ (equivalently $1/2 < \alpha < 1$), is significantly more complicated and is of continued research interest [47], [48], [50]. To the best of our knowledge, there are no known computationally efficient ways to find the finite β-expansion when $1/2 < \alpha < 1$ (if it exists) [N. Sidorov, personal communication, May 24, 2022]. In practice, we encounter filter values $\alpha(= 1/\beta)$ that are much closer to 1, and hence, we need an alternative approach to find this finite β-radix representation for θ_k. In the next section, we show that by performing a suitable preprocessing, finite β-radix representation can be formulated as a binary search problem which is guaranteed to succeed for all values of β that permit unique finite β-expansions.

A. Formulation as a Binary Search Problem

Before describing the algorithm, we first introduce the notion of a *collision-free* set.

Definition 1 (Collision Free set). *Given an undersampling factor* D, *define a class of "collision free" AR(1) filters as:*

$$\mathcal{G}_{\mathrm{D}} = \left\{ \alpha \in (0,1) \text{s.t.} \mathbf{h}_\alpha^\top \mathbf{v}_i \ne \mathbf{h}_\alpha^\top \mathbf{v}_j \forall i \ne j, \mathbf{v}_i, \mathbf{v}_j \in \mathcal{S}_{\mathrm{all}} \right\}$$

The set $\mathcal{G}_{\mathrm{D}}$ denotes permissible values of the AR(1) filter parameter α such that each of the 2^{D} binary sequences in $\mathcal{S}_{\mathrm{all}}$ maps to a unique element in the set Θ_α. In other words, every $\theta_k \in \Theta_\alpha$ has a unique D-bit expansion for all $\alpha \in \mathcal{G}_{\mathrm{D}}$. This naturally raises the question "How large is the set $\mathcal{G}_{\mathrm{D}}$?". Theorem 1 already provided the answer to this question, where the identifiability result implies that for every D, almost all $\alpha \in (0,1)$ belong to this set $\mathcal{G}_{\mathrm{D}}$ (with the possible exception of a measure zero set). Hence, Theorem 1 ensures that there are infinite choices for collision-free filter parameters.

Lemma 5. *For every* $\alpha \in \mathscr{G}_{\mathrm{D}}$, *the mapping* $\mathbf{\Phi}_{\alpha}(\,.\,):\,.\,\mathscr{S}_{\text{all}} \to \Theta_{\alpha}$, $\mathbf{\Phi}_{\alpha}(\mathbf{v}) = \mathbf{h}_{\alpha}^{\top}\mathbf{v}$ *forms a bijection between* $\mathscr{S}_{\text{all}}$ and Θ_{α}.

Proof. Since $\alpha \in \mathscr{G}_{\mathrm{D}}$, from the definition of the set $\mathscr{G}_{\mathrm{D}}$, it is clear that for any $\mathbf{v}_i, \mathbf{v}_j \in \mathscr{S}_{\text{all}}, \mathbf{v}_i \neq \mathbf{v}_j$ we have $\mathbf{h}_{\alpha}{}^{\top}\mathbf{v}_i \neq \mathbf{h}_{\alpha}{}^{\top}\mathbf{v}_j$. Therefore, the mapping is injective. Furthermore, from (19) we also have $|\Theta_{\alpha}| \leq |\mathscr{S}_{\text{all}}| = 2^{\mathrm{D}}$. Since $\mathbf{\Phi}_{\alpha}(\,\cdot\,)$ is injective, we must also have $|\Theta_{\alpha}| = 2^{\mathrm{D}}$ and hence the mapping $\mathbf{\Phi}_{\alpha}(\,.\,)$ forms a bijection between $\mathscr{S}_{\text{all}}$ and Θ_{α}. □

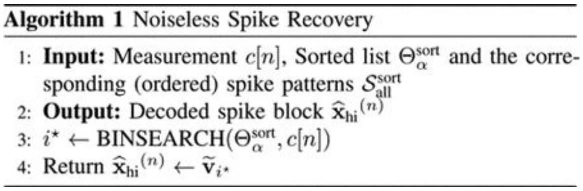
When $\alpha \in \mathscr{G}_{\mathrm{D}}$, Lemma 5 states that the finite beta expansion for every $\theta_k \in \Theta_{\alpha}$ is unique. Lemma 5 provides a way to avoid exhaustive search over $\mathscr{S}_{\text{all}}$, and yet identify $\mathbf{x}_{\text{hi}}{}^{(n)}$ from $c[n]$ in a computationally efficient way. From Lemma 5, we know that each of the 2^{D} spiking patterns in $\mathscr{S}_{\text{all}}$ maps to a unique element in Θ_{α}, and each element in Θ_{α} has a corresponding spiking pattern. Hence instead of searching $\mathscr{S}_{\text{all}}$, we can equivalently search the set Θ_{α} in order to determine the unknown spiking pattern. Since Θ_{α} permits "ordering", searching Θ_{α} has a distinct computational advantage over searching $\mathscr{S}_{\text{all}}$. This ordering enables us to employ binary search over (an ordered) Θ_{α} and find the desired element in a computationally efficient manner. To do this, we first sort the set Θ_{α} (in ascending order) and *arrange the corresponding elements of* $\mathscr{S}_{\text{all}}$ *in the same order*. Given Θ_{α} as an input, the function SORT(·) returns a sorted list $\Theta_{\alpha}^{\text{sort}}$, and an index set $\mathscr{I} = \{i_0, i_1, \cdots, i_{2^{\mathrm{D}}-1}\}$ containing the indices of the sorted elements in the list Θ_{α}.

$$\Theta_{\alpha}^{\text{sort}}, \mathscr{I} \leftarrow \text{SORT}(\Theta_{\alpha})$$

Let us denote the elements of the sorted lists as $\Theta_{\alpha}^{\text{sort}} = \left\{\tilde{\theta}_0, \cdots, \tilde{\theta}_{2^{\mathrm{D}}-1}\right\}$, and $\mathscr{S}_{\text{all}}^{\text{sort}} = \{\tilde{\mathbf{v}}_0, \cdots, \tilde{\mathbf{v}}_{2^{\mathrm{D}}-1}\}$ where:

$$\tilde{\theta}_0 < \tilde{\theta}_1 < \cdots < \tilde{\theta}_{2^{\mathrm{D}}-1} \text{and} \tilde{\theta}_j = \theta_{i_j}, \tilde{\mathbf{v}}_j = \mathbf{v}_{i_j} \,\forall j\,.$$

It is important to note that this sorting step does not depend on the measurements $\mathbf{c}$, and can therefore be part of a preprocessing pipeline that can be performed offline. However, it does require memory to store the sorted lists. In the noiseless setting, we know that every scalar measurement $c\big[n\big] = \mathbf{h}_{\alpha}^{\top}\mathbf{x}_{\text{hi}}{}^{(n)}$ belongs to the set $\Theta_{\alpha}^{\text{sort}}$. Therefore, if we identify its index, say $i^{\star}$, then we can successfully recover $\mathrm{x}_{\text{hi}}{}^{(n)}$ by returning the corresponding binary vector $\tilde{\mathbf{v}}_{i^{\star}}$ from $\mathscr{S}_{\text{all}}^{\text{sort}}$. Therefore, we can formulate the decoding problem as searching for the input $c[n]$ in the sorted list $\Theta_{\alpha}^{\text{sort}}$. This can be efficiently done by using "Binary Search". The noiseless spike decoding procedure is summarized as Algorithm 1. Since the complexity of performing a binary search over an ordered list of N elements is $O(\log N)$, the complexity of Algorithm 1 is logarithmic in the cardinality of $\Theta_{\alpha}^{\text{sort}}$, which results in a complexity of $O\Big(\log\big(2^{\mathrm{D}}\big)\Big) = O\big(\mathrm{D}\big)$. We summarize this result in the following Lemma.



Algorithm 1 Noiseless Spike Recovery

```
Input: Measurement $c[n]$, Sorted list $\Theta_\alpha^{\text{sort}}$ and the corresponding (ordered) spike patterns $\mathcal{S}_{\text{all}}^{\text{sort}}$
Output: Decoded spike block $\widehat{\mathbf{x}}_{\text{hi}}{}^{(n)}$
$i^\star \leftarrow \text{BINSEARCH}(\Theta_\alpha^{\text{sort}}, c[n])$
Return $\widehat{\mathbf{x}}_{\text{hi}}{}^{(n)} \leftarrow \widetilde{\mathbf{v}}_{i^\star}$
```

Lemma 6. *Assume* $\alpha \in \mathcal{G}_{\mathrm{D}}$. *Given the ordered set* $\Theta_\alpha^{\text{sort}}$, *and an input* $c\left[n\right] = \mathbf{h}_\alpha^\top \mathbf{x}_{\text{hi}}{}^{(n)}$, Algorithm 1 terminates in $O(\mathrm{D})$ *steps and its output* $\widehat{\mathbf{x}}_{\text{hi}}{}^{(n)}$ satisfies $\widehat{\mathbf{x}}_{\text{hi}}{}^{(n)} = \mathrm{x}_{\text{hi}}{}^{(n)}$.

B. *Noisy Measurements and* 1 D *Nearest Neighbor Search*

We demonstrate how binary search can still be useful in presence of noise by formulating noisy spike detection as a one dimensional *nearest neighbor search* problem. Suppose $\{z_{\text{lo}}[n]\}_{n=0}^{M-1}$ denote noisy D-fold decimated filter output

$$z_{\text{lo}}[n] = y_{\text{lo}}[n] + w[n],\ 0 \leq n \leq M - 1 \tag{21}$$

Here $w[n]$ represents the additive noise term that corrupts the (noiseless) low-rate measurements $y_{\text{lo}}[n]$. Similar to (7), we compute $c_e[n]$ from $z_{\text{lo}}[n]$ as follows:

$$c_e[n] = z_{\text{lo}}[n] - \alpha^{\mathrm{D}} z_{\text{lo}}[n-1] \tag{22}$$

$$= \sum_{i=1}^{\mathrm{D}} \alpha^{\mathrm{D}-i} x_{\text{hi}}[(n-1)\mathrm{D} + i] + e[n] = c[n] + e[n] \tag{23}$$

where $c\left[n\right] = \mathbf{h}_\alpha^\top \mathbf{x}_{\text{hi}}{}^{(n)} \in \Theta_\alpha^{\text{sort}}$, and $e\left[n\right] = w\left[n\right] - \alpha^{\mathrm{D}} w\left[n-1\right]$. We can interpret $c_e[n]$ as a noisy/ perturbed version of an element $c[n] \in \Theta_\alpha^{\text{sort}}$, with $e[n]$ representing the noise. This perturbed signal may no longer belong to $\Theta_\alpha^{\text{sort}}$ (i.e. $c_e[n] \notin \Theta_\alpha^{\text{sort}}$) and hence, we cannot find an exact match in the set $\Theta_\alpha^{\text{sort}}$. Instead, we aim to find the closest element in $\Theta_\alpha^{\text{sort}}$ (the nearest neighbor of $c_e[n]$) by solving the following problem:

$$\widehat{\mathbf{x}}_{\text{hi}}^{(n)} = \arg \min_{\mathbf{v} \in \mathcal{S}_{\text{all}}^{\text{sort}}} \left| c_e\left[n\right] - \mathbf{h}_\alpha^\top \mathbf{v} \right| \tag{24}$$

Solving (24) is equivalent to finding the spike sequence $\widetilde{\mathbf{v}} \in \mathcal{S}_{\text{all}}^{\text{sort}}$ that maps to the nearest neighbor of $c_e[n]$ in the set $\Theta_\alpha^{\text{sort}}$. By leveraging the sorted list $\Theta_\alpha^{\text{sort}}$, it is no longer necessary to parse the list sequentially (which would incur $O\left(2^{\mathrm{D}}\right)$ complexity), instead we can perform a modified binary search as summarized in Algorithm 2, that keeps track of additional indices compared to the vanilla binary search. Finally, we return the unique spiking pattern from $\mathcal{S}_\alpha^{\text{sort}}$ that gets mapped to the nearest neighbor of the noisy measurement $c_e[n]$. It is well-known that the nearest neighbor for any query could be found in $O\left(\log\left(2^{\mathrm{D}}\right)\right) = O\left(\mathrm{D}\right)$ steps, instead of the linear complexity of $O\left(2^{\mathrm{D}}\right)$. This guarantees a computationally efficient decoding of spikes by solving (24).

Next, we characterize the error events that lead to erroneous detection of a block of spikes. Recall that the set $\Theta_\alpha^{\text{sort}}$ is sorted, and its elements satisfy the ordering:

$$0 = \tilde{\theta}_0 < \tilde{\theta}_1 < \cdots < \tilde{\theta}_{l_{\mathrm{D}}} = 1 + \alpha + \cdots + \alpha^{\mathrm{D}-1}$$

where $l_{\mathrm{D}} := 2^{\mathrm{D}} - 1$. We also have $\tilde{\theta}_k = \mathbf{h}_\alpha^\top \tilde{\mathbf{v}}_k$, where $\tilde{\mathbf{v}}_k \in \mathcal{S}_{\text{all}}^{\text{sort}}$ is a binary spiking sequence of length D.

For each $\tilde{\mathbf{v}}_k$ and each n, we will determine the error event $\hat{\mathbf{x}}_{\text{hi}}^{(n)} \neq \mathbf{x}_{\text{hi}}^{(n)}$, when $\mathbf{x}_{\text{hi}}^{(n)} = \tilde{\mathbf{v}}_k$. First, consider the scenario when $\mathbf{x}_{\text{hi}}^{(n)} = \tilde{\mathbf{v}}_k$ for some $0 < k < l_{\mathrm{D}}$ (excluding $\tilde{\mathbf{v}}_0, \tilde{\mathbf{v}}_{l_D}$). The corresponding noiseless measurement is $c\left[n\right] = \tilde{\theta}_k = \mathbf{h}_\alpha^\top \tilde{\mathbf{v}}_k$ which satisfies $\tilde{\theta}_{k-1} < c\left[n\right] = \tilde{\theta}_k < \tilde{\theta}_{k+1}$. Since $\Theta_\alpha^{\text{sort}}$ is sorted, it can be easily verified that the nearest neighbor of $c_e[n]$ will be $\tilde{\theta}_k$, if and only if $c_e[n]$ satisfies the following condition:

$$\left(\tilde{\theta}_{k-1} + \tilde{\theta}_k\right)/2 \le c_e\left[n\right] \le \left(\tilde{\theta}_{k+1} + \tilde{\theta}_k\right)/2 \tag{25}$$

Since $\tilde{\theta}_k = \mathbf{h}_\alpha^\top \tilde{\mathbf{v}}_k$, the solution to (24) is attained at $\tilde{\mathbf{v}}_k \in \mathcal{S}_{\text{all}}^{\text{sort}}$, and the decoding is successful. Therefore Algorithm 2 produces an erroneous estimate of $\tilde{\mathbf{v}}_k$ if and only if $c_e[n]$ violates (25). The event $c_e\left[n\right] \notin \left[\frac{\tilde{\theta}_{k-1} + \tilde{\theta}_k}{2}, \frac{\tilde{\theta}_{k+1} + \tilde{\theta}_k}{2}\right]$ is equivalent to $e[n] \in \mathcal{E}_k$ ($e[n]$ is defined earlier in (23)), where

$$\mathcal{E}_k = \left\{e\left[n\right] < -\frac{\tilde{\theta}_k - \tilde{\theta}_{k-1}}{2}, \text{or}\, e\left[n\right] > \frac{\tilde{\theta}_{k+1} - \tilde{\theta}_k}{2}\right\} \tag{26}$$

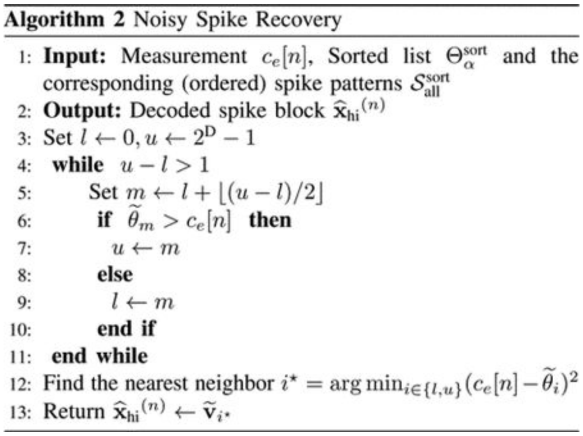
Finally, we characterize the error events for $k = 0, l_{\mathrm{D}}$. The error events for $c[n] = \theta_0 = 0$ or $c[n] = \theta_{l_{\mathrm{D}}}$ are given by:

$$\mathcal{E}_0 = \left\{e\left[n\right] \ge \tilde{\theta}_1/2\right\}, \mathcal{E}_{l_{\mathrm{D}}} = \left\{e\left[n\right] \le -\left(\tilde{\theta}_{l_{\mathrm{D}}} - \tilde{\theta}_{l_{\mathrm{D}}-1}\right)/2\right\} \tag{27}$$

Define the “minimum distance” between points in $\Theta_\alpha^{\text{sort:}}$

$$\Delta\theta_{\min}\left(\alpha, \mathrm{D}\right) = \min_{1 \leq k \leq l_{\mathrm{D}}} \left|\widetilde{\theta}_k - \widetilde{\theta}_{k-1}\right|.$$

This minimum distance depends on A, α and D. From (26), (27) it can be verified that if $2|w[n]| < \Delta\theta_{\min}(\alpha, \mathrm{D})/2$ (which would imply $|e[n]| < \Delta\theta_{\min}(\alpha, \mathrm{D})/2$) for all n, then $\widehat{\mathbf{x}}_{\mathrm{hi}}{}^{(n)} = \mathbf{x}_{\mathrm{hi}}{}^{(n)}$. As summarized in Theorem 2, Algorithm 2 can exactly recover the ground truth spikes from measurements corrupted by bounded adversarial noise, the extent of the robustness is determined by the parameters A, α, D.



Algorithm 2 Noisy Spike Recovery

```
Input: Measurement $c_e[n]$, Sorted list $\Theta_\alpha^{\mathrm{sort}}$ and the corresponding (ordered) spike patterns $\mathcal{S}_{\mathrm{all}}^{\mathrm{sort}}$
Output: Decoded spike block $\widehat{\mathbf{x}}_{\mathrm{hi}}{}^{(n)}$
Set $l \leftarrow 0, u \leftarrow 2^{\mathrm{D}} - 1$
while $u - l > 1$
    Set $m \leftarrow l + \lfloor (u - l)/2 \rfloor$
    if $\widetilde{\theta}_m > c_e[n]$ then
        $u \leftarrow m$
    else
        $l \leftarrow m$
    end if
end while
Find the nearest neighbor $i^* = \arg\min_{i \in \{l,u\}} (c_e[n] - \widetilde{\theta}_i)^2$
Return $\widehat{\mathbf{x}}_{\mathrm{hi}}{}^{(n)} \leftarrow \widetilde{\mathbf{v}}_{i^*}$
```

Theorem 2. *Assume* $\alpha \in \mathcal{G}_{\mathrm{D}}$. *Given the ordered set* $\Theta_\alpha^{\mathrm{sort}}$, *the output of* Algorithm 2 *with input* $c_e[n]$ *exactly coincides with the solution of the optimization problem* (24) in at most $O(\mathrm{D})$ *steps. Furthermore, if for all* n, $|w[n]| < \Delta\theta_{\min}(\alpha, \mathrm{D})/4$, *then the output of* Algorithm 2 *satisfies* $\widehat{\mathbf{x}}_{\mathrm{hi}}{}^{(n)} = \mathbf{x}_{\mathrm{hi}}{}^{(n)}$.

From Theorem 2, it is evident that $\Delta\theta_{\min}(\alpha, \mathrm{D})$ plays an important role in characterizing the upper bound on noise. We attempt to gain insight into how $\Delta\theta_{\min}(\alpha, \mathrm{D})$ varies as a function of α when D is held fixed.

Lemma 7. *Given* D, $\Delta\theta_{\min}\left(\alpha, \mathrm{D}\right) = A\alpha^{\mathrm{D}-1}$ for $\alpha \in (0,0.5]$.

Proof. The proof for $A = 1$ is in Appendix C and it can be scaled to obtain the desired bound. □

When $\alpha \in (0,0.5]$, $\Delta\theta_{\min}(\alpha, \mathrm{D})$ is monotonically increasing with α. However, for $\alpha > 0.5$ the trend fluctuates with α differently for different D, and becomes quite challenging to predict. This is also confirmed by the empirical plot in Fig. 1 A refined analysis of $\Delta\theta_{\min}(\alpha, \mathrm{D})$ to gain insight into desirable filter parameters α is an interesting direction for future work.

C. Trade-off between memory and computational complexity

A crucial aspect of Algorithms 1 and 2 is that they achieve efficient run-time complexity by leveraging the offline construction of the sorted list $\Theta_\alpha^{\mathrm{sort}}$ and $\mathcal{S}_{\mathrm{all}}^{\mathrm{sort}}$. These lists, each with 2^{D}

elements, need to be stored in memory and made available during run-time. Since there is no free lunch, the resulting computational efficiency of $O(\mathrm{D})$ at run-time is attained at the expense of the additional memory that is required to store the sorted lists $\Theta_{\alpha}^{\mathrm{sort}}, \mathcal{S}_{\mathrm{all}}^{\mathrm{sort}}$.

D. Parallelizable Implementation

Algorithm 2 (also Algo. 1) only takes $c_e[n](c[n])$ as input and returns $\hat{\mathbf{x}}_{\mathrm{hi}}^{(n)}$, and is completely de-coupled from any other $\hat{\mathbf{x}}_{\mathrm{hi}}^{(n')}, n' \neq n$. Recall that in reality, we are provided with measurements $z_{\mathrm{lo}}[n](y_{\mathrm{lo}}[n])$, and $c_e[n]($ respectively $c[n])$ needs to be computed. Due to this de-coupling, we can compute $c_e[n]'s$ in parallel using two consecutive low-rate samples $z_{\mathrm{lo}}[n], z_{\mathrm{lo}}[n-1]$ and perform a nearest neighbor search without waiting for any previously decoded spikes. Therefore, the total decoding complexity can be further improved depending on the available parallel computing resources.

IV. Error Analysis for Gaussian Noise

Algorithm 2 solves (24) without requiring any knowledge of the noise statistics. However, in order to analyze its performance, we will make the following (standard) assumptions on the statistics of the high-rate spiking signal x_{hi} and the measurement noise $w[n]$ as follows:

- **(A1)** The entries of the binary vector $\mathbf{x}_{\mathrm{hi}} \in \{0, A\}^{L}$ are i.i.d random variables distributed as $x_{\mathrm{hi}}[n] \sim A\mathrm{Bern}(p)$.
- **(A2)** The additive noise $w[n], 0 \leq n \leq M-1$ is independent of $x_{\mathrm{hi}}[n]$, and distributed as $w\left[n\right] \sim \mathcal{N}\left(0, \sigma^{2}\right)$

A. Probability of Erroneous Decoding

Under assumption **(A2)**, the ML estimate of $\mathbf{x}_{\mathrm{hi}}$ is given by the solution to the following problem:

$$\hat{\mathbf{x}}_{\mathrm{ML}} = \arg \min_{\mathbf{v} \in \{0, A\}^{L}} \| \mathbf{z}_{\mathrm{lo}} - \mathbf{S}_{\mathrm{D}} \mathbf{G}_{\alpha} \mathbf{v} \|_{2} \tag{PNN}$$

The proposed Algorithm 2 does not attempt to solve (P_{NN}), which is computationally intractable. Instead, it solves a set of $M-1$ one dimensional nearest neighbor search problems, by finding the nearest neighbor of $c_e[n]$ for each $n = 1,2, \cdots, M-1$. This scalar nearest neighbor search is implemented in a computationally efficient manner by using parallel binary search on a pre-sorted list. Notice that by the operation (22), the variance of the equivalent noise term $e[n]$ gets amplified by a factor of at most $\left(1 + \alpha^{2\mathrm{D}}\right) < 2$. This can be thought of as a price paid to achieve computational efficiency and parallelizability. The following theorem characterizes the dependence of certain key quantities of interest, such as the signal-to-noise ratio (SNR), undersampling factor D, and filter's frequency response (controlled by α) on the performance of Algorithm 2.

Theorem 3. *Suppose* $\alpha \in \mathscr{G}_{\mathrm{D}}$ *and assumptions* **(A1-A2)** *hold. Given* $\delta > 0$, *if the following condition is satisfied:*

$$\Delta\theta_{\min}^{2}\left(\alpha, \mathrm{D}\right)/\sigma^{2} \geq 4\ln\left(2M/\delta\right) \tag{28}$$

then Algorithm 2 *can exactly recover the binary signal* x_{hi} *with probability at least* $1 - \delta$.

Proof. The proof follows standard arguments for computing the probability of error for symbol detection in Gaussian noise, followed by certain simplifications and is included in Appendix D for completeness. □

In Fig. 1, we plot $\Delta\theta_{\min}(\alpha, \mathrm{D})$ as a function of D for different values of α. As expected, $\Delta\theta_{\min}(\alpha, \mathrm{D})$ decays as the D increases. Understandably, for a fixed α, as D increases, it becomes harder to recover the spikes exactly, and higher SNR is needed to compensate for the lower sampling rate. This can be interpreted as the price paid for super-resolution in presence of noise. This phenomenon is also reminiscent of the noise amplification effect in super-resolution, where the ability to super-resolve point sources becomes more severely hindered by noise as the target resolution grid becomes finer [6]. In Fig. 1, we plot $\Delta\theta_{\min}(\alpha, \mathrm{D})$ as a function of α and as predicted by Lemma 7, it monotonically increases upto 0.5, but for $\alpha > 0.5$, the behavior becomes much more erratic and a precise characterization becomes challenging. It is to be noted that in Theorem 3, we aim to *exactly recover* x_{hi}. The SNR requirement can be relaxed if our goal is to recover only spike counts instead of the true spikes as discussed in the next subsection. One can define other notions of approximate recovery, the analysis of which will be a topic of future research.

B. Relaxed Spike reconstruction: Count Estimation

As shown in Theorem 2, exact recovery of spikes is possible under somewhat restrictive condition on the noise in terms of $\Delta\theta_{\min}(\alpha, \mathrm{D})$, which becomes quite small as D increases. This naturally calls for other relaxed notions of recovery which can handle larger noise levels. In neuroscience, it is believed that information is encoded as either the spike timing (temporal code) or the firing rates (rate coding) of individual neurons in the brain. Therefore, the spike counts over an interval can be informative to understand neural functions, even when it is impossible to temporally localize the neural spikes. For example, neurons in the visual cortex encode stimulus orientations as their firing rates [52]. We will therefore focus on spike count as an approximate recovery metric, which concerns estimating the number of spikes occurring between two consecutive low-rate measurements instead of resolving the individual spiking activity at a higher resolution.

Let $\gamma[n]$ denote the total number of spikes occurring between two consecutive low-rate samples $z_{\mathrm{lo}}[n]$ and $z_{\mathrm{lo}}[n-1]$. Since x_{hi} and its estimate $\hat{\mathrm{x}}_{\mathrm{hi}}$ are both binary valued (amplitude A), the true spike count ($\gamma[n]$) and estimated count ($\hat{\gamma}[n]$) are given by: $\gamma\left[n\right] = \| \mathrm{x}_{\mathrm{hi}}{}^{(n)} \|_0$, $\hat{\gamma}\left[n\right] = \| \hat{\mathbf{x}}_{\mathrm{hi}}^{(n)} \|_0$, $n = 1, \cdots, M-1$, $\gamma[0] = x_{\mathrm{hi}}[0]/A$ and $\hat{\gamma}[0] = \hat{x}_{\mathrm{hi}}[0]/A$ since the first block is of size 1 as described in (6). Define a set $\mathscr{C}_k^{\mathrm{D}}$ as:

$$\mathscr{C}_k^{\mathrm{D}} := \left\{\mathbf{v} \in \{0, A\}^{\mathrm{D}},\ \|\,\mathbf{v}\,\|_0 \ = k\right\}, 0 \le k \le \mathrm{D}$$

It is a collection of all binary vectors (of length D) with spike count k. The ground truth spike block belongs to $\mathscr{C}_{\gamma[n]}^{\mathrm{D}}$. Any element from $\mathscr{C}_{\gamma[n]}^{\mathrm{D}}$ will give the true spike count. Hence, exact recovery of count can be possible even when spikes cannot be recovered.

For a fixed D, we define a set of α denoted by $\mathscr{F}_{\mathrm{D}}$:

$$\mathscr{F}_{\mathrm{D}} := \left\{\alpha \in \left(0,1\right) \mid \alpha^{\mathrm{D}} - \alpha^{\mathrm{D}-k_0} - 1 - \alpha^{k_0} + 1 < 0\right\} \tag{29}$$

where $k_0 = \lfloor \mathrm{D}/2 \rfloor$. We will obtain a sufficient condition for robust spike count estimation when $\alpha \in \mathscr{F}_{\mathrm{D}}$. It can be shown that for any $\mathrm{D}, \mathscr{F}_{\mathrm{D}}$ will always be non-empty. Define

$$\theta_{\min}^{k} := \min_{\mathbf{u} \in \mathscr{C}_k^{\mathrm{D}}} \mathbf{h}_\alpha^\top \mathbf{u}\, \theta_{\max}^{k} := \max_{\mathbf{u} \in \mathscr{C}_k^{\mathrm{D}}} \mathbf{h}_\alpha^\top \mathbf{u} \tag{30}$$

Observe that if

$$\theta_{\min}^{k+1} > \theta_{\max}^{k}, k = 0, 1, \cdots, \mathrm{D} - 1 \tag{31}$$

then all spike patterns $\mathbf{u}_i \in \mathscr{C}_k^{\mathrm{D}}$ (with the same spike count k) are clustered together when mapped on to the real line by the transformation $\mathbf{h}_\alpha^\top \mathbf{u}$ as shown in Figure 2 When (31) holds, we can define a “cluster-restricted minimum distance” as:

$$\Delta_{\min}^{c}\left(\alpha, \mathrm{D}\right) := \min_{0 \le k \le \mathrm{D} - 1} \theta_{\min}^{k+1} - \theta_{\max}^{k} \tag{32}$$

Given a noisy observation $c_e\left[n\right] = \mathbf{h}_\alpha^\top \mathbf{x}_{\mathrm{hi}}^{(n)} + e\left[n\right]$, the solution to the nearest neighbor problem (24) may return an incorrect neighbor $\theta_j \ne \mathbf{h}_\alpha^\top \mathbf{x}_{\mathrm{hi}}^{(n)}$. However, when (31) holds and if the noisy observation satisfies the following conditions:

$$\left(\theta_{\min}^{\gamma[n]} + \theta_{\max}^{\gamma[n]-1}\right)/2 < c_e\left[n\right] < \left(\theta_{\min}^{\gamma[n]+1} + \theta_{\max}^{\gamma[n]}\right)/2 \tag{33}$$

then the nearest-neighbor decision rule in Algorithm 2 will still ensure that $\theta_j \in \mathscr{C}_{\gamma[n]}^{\mathrm{D}}$. This has also been visualized in Fig. 2 where each colored band represents the “safe-zone” for each count and the black dotted-line denotes the boundary. This will result in correct identification of the spike count but will incur error in terms of spiking pattern. We formally summarize this in the following Theorem that provides robustness guarantee for exact count recovery from measurements corrupted by adversarial noise (similar to Theorem 2 for spike recovery).

Theorem 4. *Assume* $\alpha \in \mathscr{F}_{\mathrm{D}}$. *Given the ordered set* $\Theta_{\alpha}^{\mathrm{sort}}$, *let* $\hat{\gamma}[n]$ *be the estimated spike count obtained from* Algorithm 2 *with input* $c_e[n]$. *If for all* n, $|w[n]| < \Delta_{\min}^{c}(\alpha, \mathrm{D})/4$, *then the count can be exactly recovered, i.e.*, $\hat{\gamma}[n] = \gamma[n]$.

Proof. Proof is in Appendix E. □

It is clear that when (31) holds, $\Delta_{\min}^{c}(\alpha, \mathrm{D})$ is no smaller than $\Delta\theta_{\min}(\alpha, \mathrm{D})$, since the former is computed over neighboring elements of the cluster whereas $\Delta\theta_{\min}(\mathrm{D}, \alpha)$ computes the minimum distance over all consecutive elements (both inter-cluster as well as intra-cluster) in $\Theta_{\alpha}^{\mathrm{sort}}$. This essentially suggests that estimation of counts (for this range of α and D) can be more robust compared to inferring the individual spiking patterns. We also illustrate this numerically in Figure 1 (top), where we plot both $\Delta_{\min}^{c}$ and $\Delta\theta_{\min}$ as a function of α and the start of the interval $\mathscr{F}_{\mathrm{D}}$ (computed numerically) is denoted using dotted lines. For both values of D, we can see that $\Delta_{\min}^{c} > \Delta\theta_{\min}$ and the gap grows as α increases.

V. Numerical Experiments

We conduct numerical experiments to evaluate the performance of the proposed super-resolution spike decoding algorithm on both synthetic and real calcium imaging datasets.

A. Synthetic Data Generation and Evaluation Metrics

We create a synthetic dataset by generating high-rate binary spike sequence $\mathrm{x}_{\mathrm{hi}} \in \{0,1\}^{L}(A = 1$ and $L = 1000)$ that satisfies assumption (**A1**). The spiking probability p controls the average sparsity level given by $s := \mathbb{E}[\| \mathrm{x}_{\mathrm{hi}} \|_0] = Lp$. We aim to reconstruct x_{hi} from $M \approx L/\mathrm{D}$ low-rate measurements $z_{\mathrm{lo}}[n]$ defined in (21). Notice that we operate in a regime where the expected sparsity is greater than the total number of low-rate measurements, i.e., $s > M$. We employ the widely-used *F-score* metric to evaluate the accuracy of spike detection [4], [10]. The F-score is computed by first matching the estimated and ground truth spikes. An estimated spike is considered a “match” to a ground truth spike if it is within a distance of t_0 of the ground truth (many-to-one matching is not allowed) [4], [10]. Let K and K' be the total number of ground truth and estimated spikes, respectively. The number of spikes declared as true positives is denoted by T_p. After the matching procedure, we compute the recall $\left(R = \frac{T_p}{K}\right)$ which is defined as the ratio of true positives (T_p) and the total number of ground truth spikes (K). Precision $\left(P = \frac{T_p}{K'}\right)$ measures the fraction of the total detected spikes which were correct. Finally, the F-score is given by the harmonic mean of recall and precision $\mathrm{F{-}score} = 2PR/(P + R)$.

B. Noiseless Recovery: Role of Binary priors and memory

We first consider the noiseless setting ($w[n] = 0$ in (21)). We compare the performance of Algorithm 2 against box-constrained l_1 minimization method [35], [36], where we solve:

$$\min_{\mathbf{x} \in \mathbb{R}^L} \| \mathbf{x} \|_1 \text{ s.t.} \| \mathbf{y}_{\text{lo}} - \mathbf{S}_{\text{D}} \mathbf{G}_\alpha \mathbf{x} \|_2 \le \epsilon, \mathbf{0} \le \mathbf{x} \le A\mathbf{1} \tag{P1}$$

For synthetic data, ϵ is chosen using the norm of the noise term $\| \mathbf{w} \|_2$. This *oracle* choice ensures most favorable parameter tuning for the (P1), although a more realistic choice would be to set $\epsilon = \sqrt{M}\sigma$ according to the noise power (σ). In the noiseless setting, we choose $\epsilon = 0$. The problem (P1) is a standard convex relaxation of (P0) which promotes sparsity as well as tries to impose the binary constraint via the box-relaxation (introduced in Section II-C). In Fig. 3 (Top), we plot the F-score ($t_0 = 0$) as a function of D. As can be observed, Algorithm 2 consistently achieves an F-score of 1, whereas the F-score of l_1 minimization shows a decay as D increases. This confirms Lemma 3 that for D > 1, using box-constraints with l_1 norm minimization is not enough to enable exact recovery from low rate measurements. In absence of noise, the performance of Algorithm 2 is not affected by the filter parameter α as shown in Fig. 3 (Top).

Next, we compare the reconstruction from the decimated output of (i) an AR(1) filter and (ii) an FIR filter of length r driven by the same input $\text{x}_{\text{hi}} \in \{0,1\}^{1000}$. We choose the FIR filter $\mathbf{h} = \left[1, \alpha, \cdots, \alpha^{r-1}\right]^\top$ (truncation of the IIR filter) with $\alpha = 0.5$. Algorithm 2 is applied to the low-rate AR(1) measurements, whereas the algorithm proposed in [40] is used for the FIR case. The algorithm applied for the FIR case can provably operate with the optimal number of measurements when $\alpha = 0.5$ and hence, we chose this specific value for the filter parameter. In Figure 3 (Bottom), we again compare the average F-score as a function of D, averaged over 10000 Monte Carlo runs, for $p = 0.5$. As predicted by Lemma 4 despite utilizing binary priors, the error for the FIR filter shows a phase transition when $\text{D} > r$. This demonstrates the critical role played by the infinite memory of the AR(1) filter in achieving exact recovery with arbitrary D.

C. Performance of noisy spike decoding

We generate noisy measurements of the form (21), where $w[n]$ and $x_{\text{hi}}[n]$ satisfy assumptions (**A1-A2**). We illustrate some representative examples of recovered spikes on synthetic data. In Fig. (4), we display the recovered super-resolution estimates on synthetically generated measurements for two undersampling factors D = 5 (left), 10 (right). For each D, the top plots show the spikes recovered using Algorithm 2 and l_1 minimization with box-constraint where the noise realization obeys the bound in Theorem 2, while the bottom plots show the same for noise realization violating the bound. The output of l_1 minimization with box-constraint is inaccurate, and the spikes are clustered towards the end of each block of length D. This bias is consistent with the prediction made by our theoretical results in Lemma 3. When the noise is small enough (top), Algorithm 2 exactly decodes the spikes, including the ones occurring between two consecutive low-rate samples as predicted by Theorem 2. In presence of larger noise (violating the bound), the spikes estimated using l_1 minimization continue to be biased to be clustered towards the end of the block. Although the spikes recovered using Algorithm 2 are not exact, most of the detected spikes are within a tolerance window of ground truth spikes. In fact, the spike count estimation is perfect as predicted by Theorem 4. We next quantitatively evaluate the performance in presence of noise, where the metrics are computed with $t_0 = 2$. In Fig. 5 (Top), we plot the F-score as a function of D for different values of α. For a fixed α, the F-score of both methods decays with increasing D, but Algorithm 2 consistently attains a higher F-score compared to l_1 minimization. We observe that $\alpha = 0.5$ leads to a higher F-score potentially due to having a larger $\Delta\theta_{\min}(\alpha, \mathrm{D})$ compared to $\alpha = 0.9$. Next, in Fig. 7, we study the behavior of spike detection as a function of the spiking probability p, while keeping D fixed at D = 5. When σ is fixed, the performance trend is not significantly affected by the spiking probability. At first, this may seem surprising as the expected sparsity is growing while the number of measurements is unchanged. However, since our algorithm exploits the binary nature of the spikes (and not just sparsity), it can handle larger sparsity levels. The spikes reconstructed using l_1 minimization achieve a much lower F-score than Algorithm 2 since the former fails to succeed when the sparsity is large. As expected, smaller σ leads to higher F-scores.

In Fig. 8, we study the probability of erroneous spike detection as a function of D and validate the upper bound derived in Theorem 3. Recall that the decoding is considered successful if “every” spike is detected correctly. Therefore, it becomes more challenging to “exactly super-resolve” all the spikes in presence of noise as the desired resolution becomes finer. We calculate the empirical probability of error and overlay the corresponding theoretical bound. As shown in Fig. 8, the empirical probability of error is indeed upper bounded by the bound computed by our analysis. The empirical probability of error increases as a function of undersampling factor D.

Finally, we evaluate the noise tolerance of the proposed methodology by comparing the average F-score as a function of the noise level σ, while keeping the spiking rate and undersampling factor fixed at $p = 0.35$ and D = 5, respectively. As seen in Fig. 6 (Top), the performance of both algorithms degrades with increasing noise level and this is also consistent with the intuition that it becomes harder to super-resolve spikes with more noise. However, for both filter parameters considered in this experiment Algorithm 2 has a higher F-score compared to box-constrained l_1 minimization. For large noise levels (comparable to spike amplitude $A = 1$), the performance gap decreases for $\alpha = 0.9$ but Algorithm 2 achieves a much higher F-score for $\alpha = 0.5$ at all noise levels.

As discussed in Section IV-B, we next study a relaxed notion of spike recovery which focuses on the spike counts occurring between two consecutive low-rate samples. Let $\mathbf{\Gamma} = [\gamma[0], \cdots, \gamma[M-1]]^{\top}$ be the vector of counts and $\widehat{\mathbf{\Gamma}}$ be its estimate. In Fig. 6 (Bottom) we plot the average l_1 distance $\| \mathbf{\Gamma} - \widehat{\mathbf{\Gamma}} \|_1$ as a function of the noise level. We observe that for $\alpha = 0.9$ (it can be verified from Fig. 1 (Top) that $0.9 \in \mathscr{F}_5$), it is possible to exactly recover the spike counts at higher noise even though the F-score (for timing recovery) has dropped below 1. However, this is not the case for $\alpha = 0.5$, since $0.5 \notin \mathscr{F}_5$. This is consistent with the conclusion of Theorem 4 which states that when $\alpha \in \mathscr{F}_{\mathrm{D}}$, the noise tolerance for exact count recovery can be much larger than exact spike recovery since $\Delta_{\min}^{c}(\alpha, \mathrm{D}) > \Delta\theta_{\min}(\alpha, \mathrm{D})$.

D. Spike Deconvolution from Real Calcium Imaging Datasets

We now discuss how the mathematical framework developed in this paper can be used for super-resolution spike deconvolution in calcium imaging. Two-photon calcium imaging is a widely used imaging technique for large scale recording of neural activity with high spatial but poor temporal resolution. In calcium imaging, the signal $\mathbf{x}_{\mathrm{hi}}$ corresponds to the underlying neural spikes which is modeled to be binary valued on a finer temporal scale [2], [46]. Each neural spike results in a sharp rise in Ca^{2+} concentration followed by a slow exponential decay, leading to superposition of the responses from nearby spiking events [2]–[4]. This calcium transient can be modeled by the first order autoregressive model introduced in Section II. The decay time constant depends on the calcium indicator and essentially determines the filter parameter α. The signal $y_{\mathrm{hi}}[n]$ is an unobserved signal corresponding to sampling the calcium fluorescence at a high sampling rate (at the same rate as the underlying spikes). The observed calcium signal $y_{\mathrm{lo}}[n]$ corresponds to downsampling $y_{\mathrm{hi}}[n]$ at an interval determined by the frame rate of the microscope. The frame rate of a typical scanning microscopy system (that captures the changes in the calcium fluorescence) is determined by the amount of time required to spatially scan the desired field of view, which makes it significantly slower compared to the temporal scale of the neural spiking activity. We model this discrepancy by the downsampling operation (by a factor D). Therefore, the mathematical framework developed in this paper can be directly applied to reconstruct the underlying spiking activity at a temporal scale finer than the sampling rate of the calcium signal. Using real calcium imaging data, we demonstrate a way to fuse our algorithm with a popular spike deconvolution algorithm called OASIS [43]. OASIS solves an l_1 minimization problem similar to (P1) with only the non-negativity constraint, in order to exploit the sparse nature of the spiking activity. Unlike our approach where we wish to obtain spikes representation on a finer temporal scale, OASIS returns the spike estimates on the low-resolution grid. This is typically used to infer the spiking rate over a temporal bin equal to the sampling interval. We demonstrate that our proposed framework can be integrated with OASIS and improve its performance. As we saw in the synthetic experiments, the noise level is an important consideration. By augmenting Algorithm 2 with OASIS, referred as “B-OASIS”, the denoising power of l_1 minimization can be leveraged.Let $\widehat{\mathbf{x}}_{11} \in \mathbb{R}^{M}$ be the estimate obtained on a low-resolution grid by solving the l_1 minimization problem such as the one implemented in OASIS. We can obtain an estimate of the denoised calcium signal as $\widehat{y}_{\mathrm{lo}}\left[n\right] = \alpha^{\mathrm{D}}\widehat{y}_{\mathrm{lo}}\left[n\right] + \widehat{x}_{11}\left[n\right], n \geq 1$ and $\widehat{y}_{\mathrm{lo}}[0] = \widehat{x}_{11}[0]$. We can now utilize the denoised calcium signal $\widehat{y}_{\mathrm{lo}}[n]$ generated by OASIS to obtain the estimate $c_e[n]$ indirectly. Due to the non-linear processing done by OASIS, it is difficult to obtain the resulting noise statistics. An important advantage of Algorithm 2 is that it does not rely on the knowledge of the noise statistics. Hence, we can directly apply Algorithm 2 on $\widehat{c}_e\left[n\right] = \widehat{y}_{\mathrm{lo}}\left[n\right] - \alpha^{\mathrm{D}}\widehat{y}_{\mathrm{lo}}\left[n - 1\right]$ (instead of $c_e[n]$) to obtain a binary “fused super-resolution spike estimate”.

E. Results

We evaluate the algorithms on the publicly available GENIE dataset [53], [54] which consists of simultaneous calcium imaging and *in vivo* cell-attached recording from the mouse visual cortex using genetically encoded GCaMP6f calcium indicator GCaMP6f [53], [54]. The calcium images were acquired at a frame rate of 60 Hz and the ground truth electrophysiology signal was digitized at 10 KHz and synchronized with the calcium frames. In addition to using the original data, we also synthetically downsample it to emulate the effect of a lower frame rate of 30 Hz, and evaluate how the performance changes by this downsampling operation.

In Fig. 10, we extract an interval of ~ 2 sec (from the neuron 1 of the GCaMP6f indicator dataset) and qualitatively compare the detected spikes with the ground truth. We downsample the data by a factor of 2 to emulate frame rate of 30 Hz, the low-rate grid becomes coarser. As a result of which, we observe an offset between ground truth spikes and estimate produced by OASIS. However, with the help of binary priors (B-OASIS), we can output spikes that are not restricted to be on the coarser scale, and this mitigates the offset observed in the raw estimates obtained by OASIS.

We quantify the improvement in the performance by comparing the F-scores of OASIS and B-OASIS at both sampling rates (60 and 30 Hz). Since the output of OASIS is non-binary, the estimated spikes are binarized by thresholding. To ensure a fair comparison, we select the threshold by a 80 – 20 cross-validation scheme that maximizes the average F-score on a held-out validation set (averaged over 3-random selections of the validation set). The tolerance for the F-score was set at 100 ms. The dataset consisted of 34 traces of length ~ 234 s. The OASIS algorithm has an automated routine to estimate the parameter α, which we utilize for our experiments. The amplitude A is estimated using the procedure described in Appendix F. We use D = 12 to obtain the spike representation for B-OASIS. In order to quantify the performance boost achieved by augmentation, we isolate the traces where the F–score of OASIS drops below 0.5 and compare the average F-score and recall for these data points. As shown in Fig. 9, at both sampling rates, we see a significant improvement in the average F-score of B-OASIS over OASIS, attributed to an increase in recall while keeping the precision unchanged. Additionally, despite downsampling, the spike detection performance is not significantly degraded with binary priors, although the detection criteria were unchanged.

VI. Conclusion

We theoretically established the benefits of binary priors in super-resolution, and showed that it is possible to achieve significant reduction in sample complexity over sparsity-based techniques. Using an AR(1) model, we developed and analyzed an efficient algorithm that can operate in the extreme compression regime $(M \ll K)$ by exploiting the special structure of measurements and trading memory for computational efficiency at run-time. We also demonstrated that binary priors can be used to boost the performance of existing neural spike deconvolution algorithms. In the future, we will develop algorithmic frameworks for incorporating binary priors into different neural spike deconvolution pipelines and evaluate the performance gain on diverse datasets. The extension of this binary framework for higher-order AR filters is another exciting future direction.

Acknowledgement

The authors would like to thank Prof. Nikita Sidorov, Department of Mathematics at the University of Manchester, for helpful discussions regarding computational challenges in finding finite β-expansion in the range $\beta \in (1,2)$. We would also like to thank the anonymous reviewers for their helpful comments that resulted in addition of new theoretical results. This work was supported by Grants ONR N00014-19-1-2256, N00014-19-1-2227, DE-SC0022165, NSF 2124929, and NSF CAREER ECCS 1700506.

Appendix

APPENDIX A: Proof of Theorem 1

Proof. We show that for any α in $0 < \alpha < 1$, except possibly for a set consisting of only a finite number of points, (10) always has a unique binary solution. Consider all possible D–dimensional ternary vectors with their entries chosen from $\{-1, 0, 1\}$, and denote them as $\mathbf{v}^{(i)} = \left[v_1^{(i)}, v_2^{(i)}, \cdots, v_{\mathrm{D}}^{(i)}\right]^T \in \left\{-1, 0, 1\right\}^{\mathrm{D}}, 0 \leq i \leq 3^{\mathrm{D}} - 1$. We use the convention that $\mathbf{v}^{(0)} = 0$. For every $i > 0$, we define a set $\mathcal{Z}_{\mathbf{v}^{(i)}}$ determined by $\mathbf{v}^{(i)}$ as $\mathcal{Z}_{\mathbf{v}^{(i)}} := \left\{x \in (0,1) \mid \sum_{k=1}^{\mathrm{D}} v_k^{(i)} x^{\mathrm{D}-k} = 0\right\}$. Notice that $p_i(x) := \sum_{k=1}^{\mathrm{D}} v_k^{(i)} x^{\mathrm{D}-k}$ denotes a polynomial (in x) of degree at most D–1, whose coefficients are given by the ternary vector $\mathbf{v}^{(i)}$. The set $\mathcal{Z}_{\mathbf{v}^{(i)}}$ denotes the set of zeros of $p_i(x)$ that are contained in (0, 1). Since the degree of $p_i(x)$ is at most D − 1, $\mathcal{Z}\mathbf{v}^{(i)}$ is a finite set with cardinality at most D–1.

Now suppose that the binary solution of (10) is non-unique, i.e., there exist $\mathbf{u}, \mathbf{w} \in \{0, A\}^L, \mathbf{u} \neq \mathbf{w}$, such that

$$\mathbf{H}_{\mathrm{D}}(\alpha)\mathbf{u} = \mathbf{H}_{\mathrm{D}}(\alpha)\mathbf{w} \Rightarrow \mathbf{H}_{\mathrm{D}}(\alpha)\mathbf{u} - \mathbf{H}_{\mathrm{D}}(\alpha)\mathbf{w} = 0 \tag{34}$$

By partitioning $\mathbf{u}, \mathbf{w}$ into blocks $\mathbf{u}^{(n)}, \mathbf{w}^{(n)}$ in the same way as in (6), we can re-write (34) as $u^{(0)} = w^{(0)}$ and

$$\sum_{i=1}^{\mathrm{D}} \frac{1}{A}\left(\left[\mathbf{u}^{(j)}\right]_i - \left[\mathbf{w}^{(j)}\right]_i\right)\alpha^{\mathrm{D}-i} = 0,\ 1 \leq j \leq M - 1 \tag{35}$$

Since $\mathbf{u} \neq \mathbf{w}$, they differ at least at one block, i.e., there exists some $j_0, 1 \leq j_0 \leq M - 1$ such that $\mathbf{u}^{(j_0)} \neq \mathbf{w}^{(j_0)}$. Define $\mathbf{b} := \frac{1}{A}\left(\mathbf{u}^{(j_0)} - \mathbf{w}^{(j_0)}\right)$. Then, $\mathbf{b}$ is a non-zero ternary vector, i.e., $\mathbf{b} \in \{-1, 0, 1\}^{\mathrm{D}}$. Now from (35), we have

$$\sum_{i=1}^{\mathrm{D}} \left[\mathbf{b}\right]_i \alpha^{\mathrm{D}-i} = 0,$$

(36)

which implies that $\alpha \in \mathcal{Z}_b$. Since **b** can be any one of the 3^{D}–1 ternary vectors $\left\{\mathbf{v}^{(i)}\right\}_{i=1}^{3^{\mathrm{D}}-1}$, (36) holds if and only if $\alpha \in \mathbb{S} := \cup_{i=1}^{3^{\mathrm{D}}-1} \mathcal{Z}_{\mathbf{v}^{(i)}}$, i.e., α is a root of at least one of the polynomials $p_i(x)$ defined by the vectors $\mathbf{v}^{(i)}$ as their coefficients. For each $\mathbf{v}^{(i)}$, since the cardinality of $\mathcal{Z}_{\mathbf{v}^{(i)}}$ is at most D–1, $\mathbb{S}$ is a finite set (of cardinality at most (D–1) (3^{D}–1)), and therefore its Lebesgue measure is 0. This implies that 10 has a non-unique binary solution only if α belongs to the measure zero set $\mathbb{S}$, thereby proving the theorem. ☐

Appendix B: Proof of Lemma 2 and Lemma 3

Proof. (i) Let s_n denote the sparsity (number of non-zero elements) of the n^{th} block $\mathbf{x}_{\text{hi}}^{(n)}$ of $\mathbf{x}_{\text{hi}}$. Then, the total sparsity is $\|\mathbf{x}_{\text{hi}}\|_0 = \sum_{n=0}^{M-1} s_n$. We will construct a vector $\mathbf{v} \in \mathbb{R}^L, \mathbf{v} \neq \mathbf{x}_{\text{hi}}$ that satisfies $\mathbf{c} = \mathbf{H}_{\mathrm{D}}(\alpha)\mathbf{v}$ and $\|\mathbf{x}_{\text{hi}}\|_0 \geq \|\mathbf{v}\|_0$. Following (6), consider the partition of $\mathbf{v} = \left[v^{(0)}, \mathbf{v}^{(1)\top}, \cdots, \mathbf{v}^{(M-1)^\top}\right]^\top$. Firstly, we assign $v^{(0)} = c\left[0\right] = x_{\text{hi}}^{(0)}$. We construct $\mathbf{v}^{(n)}$ as follows. For each $n \geq 1$, there are three cases:

Case I: $s_n = 0$. In this case, $\mathbf{x}_{\text{hi}}^{(n)} = 0$ and hence $c[n] = 0$. Therefore, we assign $\mathbf{v}^{(n)} = \mathbf{x}_{\text{hi}}^{(n)} = 0$.

Case II: $s_n = 1$. First suppose that $\left[\mathbf{x}_{\text{hi}}^{(n)}\right]_{\mathrm{D}} = 0$. We construct $\mathbf{v}^{(n)}$ as follows:

$$\left[\mathbf{v}^{(n)}\right]_k = \begin{cases} c[n], & \text{if} k = \mathrm{D} \\ 0, & \text{else} \end{cases}.$$

(37)

Next suppose that $\left[\mathbf{x}_{\text{hi}}^{(n)}\right]_{\mathrm{D}} \neq 0$. Since $s_n = 1$, this implies that $\left[\mathrm{x}_{\text{hi}}^{(n)}\right]_k = 0, k = 1, \cdots,$ D–1. In this case, we construct $\mathbf{v}^{(n)}$ as follows:

$$\left[\mathbf{v}^{(n)}\right]_k = \begin{cases} c[n]/\alpha, & \text{if} k = \mathrm{D} - 1 \\ 0, & \text{else} \end{cases}.$$

(38)

Notice that both (37) and (38) ensure that $\mathbf{v}^{(n)} \neq \mathbf{x}_{\text{hi}}^{(n)}$ and $c\left[n\right] = \mathbf{h}_\alpha^T \mathbf{v}^{(n)}$. Moreover, $\|\mathbf{v}^{(n)}\|_0 = s_n$.

Case III: $s_n \geq 2$. In this case, we follow the same construction as (37). As before $\mathbf{v}^{(n)}$ satisfies $c\left[n\right] = \mathbf{h}_\alpha^\top \mathbf{v}^{(n)}$. Since $\|\mathbf{x}_{\text{hi}}^{(n)}\|_0 \geq 2$ and $\|\mathbf{v}^{(n)}\|_0 = 1$, we automatically have $\mathbf{v}^{(n)} \neq \mathbf{x}_{\text{hi}}^{(n)}$, and $\|\mathbf{v}^{(n)}\|_0 < s_n$. Therefore, combining the three cases, we can construct the desired vector $\mathbf{v}$ that satisfies $\mathbf{v} \neq \mathbf{x}_{\text{hi}}, \mathbf{c} = \mathbf{H}_{\mathrm{D}}(\alpha)\mathbf{v}$, and $\|\mathbf{v}\|_0 \leq \sum_{n=0}^{M-1} s_n = \|\mathbf{x}_{\text{hi}}^{(n)}\|_0$. Therefore, the solution $\mathrm{x}^\star$ to (P0) satisfies $\|\mathbf{x}^\star\|_0 \leq \|\mathbf{v}\|_0 \leq \|\mathrm{x}_{\text{hi}}^{(n)}\|_0$.

(ii) In this case, we construct $\mathbf{v}^{(n_0)}$ according to Case III. Since $\| \mathbf{v}^{(n_0)} \|_0 < s_{n_0}$, and $\| \mathbf{v}^{(n)} \|_0 \le s_n, n \ne n_0$, we have $\| \mathbf{v} \|_0 < \| \mathbf{x}_{\mathrm{hi}} \|_0$, implying $\| \mathbf{x}^\star \|_0 \le \| \mathbf{v} \|_0 < \| \mathbf{x}_{\mathrm{hi}} \|_0$. □

A. Proof of Lemma 3

Proof. We will construct a vector $\mathbf{v} \in \mathbb{R}^L$ whose support is of the form (16), that is feasible for (P1-B), and we will prove that it has the smallest l_1 norm. Using the block structure given by (6), we choose $\mathbf{v}^{(0)} = c[0]$. For each $n \ge 1$, we construct $\mathbf{v}^{(n)}$ based on the following two cases:

Case I: $c[n] \ge A$. Let k_n be the largest integer such that the following holds: $\mu[n] := A\left(1 + \alpha + \cdots + \alpha^{k_n - 1}\right) \le c[n]$, where $1 \le k_n \le \mathrm{D}$. Note that $k_n = 1$ always produces a valid lower bound. However, we are interested in the largest lower bound on $c[n]$ of the above form. We choose

$$\left[\mathbf{v}^{(n)}\right]_k = \begin{cases} A, \text{ if } \mathrm{D} - k_n + 1 \le k \le \mathrm{D} \\ \left(c[n] - \mu[n]\right)/\alpha^{k_n}, \text{if } k = \mathrm{D} - k_n \\ 0, \text{else} \end{cases}$$

It is easy to verify that $\mathbf{h}_\alpha^\top \mathbf{v}^{(n)} = c[n]$. From the definition of k_n, it follows that $\mu[n] \le c[n] < \mu[n] + A\alpha^{k_n}$ and hence, $0 \le \left(c[n] - \mu[n]\right)/\alpha^{k_n} < A$, which ensures that $\mathbf{v}$ obeys the box-constraints in (P1-B). Now, let $\mathbf{v}_f \in \mathbb{R}^L$ be any feasible point of (P1-B) which must be of the form $\mathbf{v}_f^{(0)} = c[0], \mathbf{v}_f^{(n)} = \mathbf{v}^{(n)} + \mathbf{r}^{(n)}$, where $\mathbf{r}^{(n)} \in \mathcal{N}\left(\mathbf{h}_\alpha^\top\right)$ is a vector in the null-space of $\mathbf{h}_\alpha^\top$. It can be verified that the following vectors $\{\mathbf{w}_t\}_{t=1}^{D-1}$ form a basis for $\left(\mathbf{h}_\alpha^\top\right)$:

$$[\mathbf{w}_t]_k = \begin{cases} 1, & k = t \\ -\alpha, & k = t + 1, \\ 0, & \text{else} \end{cases}$$

Therefore, $\exists \left\{\beta_t^{(n)}\right\}_{t=1}^{\mathrm{D}-1}$ such that $\mathbf{r}^{(n)} = \sum_{t=1}^{\mathrm{D}-1} \beta_t^{(n)} \mathbf{w}_t$. We further consider two scenarios: (i) $1 \le k_n \le \mathrm{D} - 2$. In this case $\left[\mathbf{v}^{(n)}\right]_1 = 0$, and for $k = 1,2, \cdots D$, $\left[\mathbf{v}_f^{(n)}\right]_k$ satisfies[2]

$$\left[\mathbf{v}_f^{(n)}\right]_k = \begin{cases} \beta_k^{(n)}, \text{if } k = 1 \\ \beta_k^{(n)} - \alpha\beta_{k-1}^{(n)}, \text{if } 2 \le k \le \mathrm{D} - k_n - 1 \\ \left[\mathbf{v}^{(n)}\right]_k + \beta_k^{(n)} - \alpha\beta_{k-1}^{(n)}, \text{if } k = \mathrm{D} - k_n \\ A + \beta_k^{(n)} - \alpha\beta_{k-1}^{(n)}, \text{if } \mathrm{D} - k_n + 1 \le k \le \mathrm{D} - 1 \\ A - \alpha\beta_{k-1}^{(n)}, \text{if } k = \mathrm{D} \end{cases}$$

To ensure $\mathbf{v}_f^{(n)}$ is a feasible point for (P1-B), the following must hold: $0 \le \beta_{\mathrm{D}-1}^{(n)} \le A/\alpha$ and $0 \le \beta_1^{(n)} \le A$. For $2 \le k \le \mathrm{D} - k_n - 1$, the constraint $\left[\mathbf{v}_f^{(n)}\right]_k \ge 0$ implies $\beta_k^{(n)} \ge \alpha\beta_{k-1}^{(n)}$. Since $\beta_1^{(n)} \ge 0$,

[2] In the definition of $\mathbf{v}_f^{(n)}$, an assignment will be ignored if the specified interval for k is empty.

it follows that $\beta_k^{(n)} \geq 0$ for all $2 \leq k \leq \mathrm{D} - k_n - 1$. For $\mathrm{D} - k_n + 1 \leq k \leq \mathrm{D} - 1$, the constraint $\left[\mathbf{v}_f^{(n)}\right]_k \leq A$ implies $\beta_{k-1}^{(n)} \geq \beta_k^{(n)}/\alpha$. Since $\beta_{\mathrm{D}-1}^{(n)} \geq 0$, it follows that $\beta_k^{(n)} \geq 0$ for all $\mathrm{D} - k_n \leq k \leq \mathrm{D} - 1$. (ii) $k_n \in \{\mathrm{D} - 1, \mathrm{D}\}$. In this case, for $k = 1,2, \cdots, D$, $\left[\mathbf{v}_f^{(n)}\right]_k$ satisfies

$$\left[\mathbf{v}_f^{(n)}\right]_k = \begin{cases} \left[\mathbf{v}^{(n)}\right]_1 + \beta_1^{(n)}, \text{if } k = 1 \\ A + \beta_k^{(n)} - \alpha\beta_{k-1}^{(n)}, \text{if } 2 \leq k \leq \mathrm{D} - 1 \\ A - \alpha\beta_{k-1}^{(n)}, \text{if } k = \mathrm{D} \end{cases}$$

For $2 \leq k \leq \mathrm{D} - 1$, the box-constraint $\left[\mathbf{v}_f^{(n)}\right]_k \leq A$ implies $\beta_{k-1}^{(n)} \geq \beta_k^{(n)}/\alpha$. Since $\beta_{\mathrm{D}-1}^{(n)} \geq 0$, it follows that $\beta_k^{(n)} \geq 0$ for all $1 \leq k \leq \mathrm{D} - 1$. Summarizing, we have established that $\beta_i^{(n)} \geq 0, \forall i$.

Case II: $c[n] < A$. In this case, $\mathbf{v}^{(n)}$ is constructed following (37), and hence $\mathbf{v}_f^{(n)}$ has the following structure:

$$\left[\mathbf{v}_f^{(n)}\right]_k = \begin{cases} \beta_k^{(n)}, \text{if } k = 1 \\ -\alpha\beta_{k-1}^{(n)} + \beta_k^{(n)}, \text{if } 2 \leq k \leq \mathrm{D} - 1 \\ c\left[n\right] - \alpha\beta_{k-1}^{(n)}, \text{if } k = \mathrm{D} \end{cases}$$

To ensure $\mathbf{v}_f^{(n)}$ is a feasible point, it must hold that $\beta_1^{(n)} \geq 0, \beta_k^{(n)} \geq \alpha\beta_{k-1}^{(n)} \geq 0$ for $2 \leq k \leq \mathrm{D} - 1$. Hence, in both Cases I and II, we established that $\beta_k^{(n)} \geq 0$. For each case, since $\mathbf{v}_f^{(n)}$ is a non-negative vector $\forall n$, it can be verified that

$$\begin{aligned} \| \mathbf{v}_f \|_1 &= \sum_{n=0}^{M-1} \| \mathbf{v}_f^{(n)} \|_1 = v_f^{(0)} + \sum_{n=1}^{M-1} \sum_{k=1}^{\mathrm{D}} \left[\mathbf{v}_f^{(n)}\right]_k \\ &= \underbrace{c[0] + \sum_{n=1}^{M-1} \sum_{k=1}^{\mathrm{D}} \left[\mathbf{v}^{(n)}\right]_k}_{\| \mathbf{v} \|_1} + \sum_{n=1}^{M-1} \sum_{k=1}^{\mathrm{D}-1} (1-\alpha)\beta_k^{(n)} \end{aligned}$$

We used the fact that $\sum_{k=1}^{D} \sum_{t=1}^{D-1} \beta_t^{(n)}[\mathbf{w}_t]_k = \sum_{t=1}^{\mathrm{D}-1} \left(1-\alpha\right)\beta_t^{(n)}$. If $\mathbf{v}_f \neq \mathbf{v}$, we must have $\beta_k^{(n)} \neq 0$ for some k and $n > 0$. This implies that $\| \mathbf{v}_f \|_1 > \| \mathbf{v} \|_1$. It is easy to see that the support of the constructed vector is of the form (16). Moreover, based on the above argument, $\mathbf{v}$ is the only vector that has the minimum l_1 norm among all possible feasible points of (P1-B). □

APPENDIX C: Proof of Lemma 7

Proof. For any $0 < \alpha \leq 0.5$, we begin by showing that for an integer $p \geq 1$ the following inequality holds:

$$\sum_{k=1}^{p} \alpha^{\mathrm{D}-k} = \alpha^{\mathrm{D}-p-1}\left(\frac{1-\alpha^p}{1/\alpha - 1}\right) < \alpha^{\mathrm{D}-p-1} \tag{39}$$

since $1/\alpha - 1 \geq 1$ and $1 - \alpha^{p} < 1$ in the regime $0 < \alpha \leq 0.5$. Let $\mathcal{S}_1 = \left\{0, \alpha^{\mathrm{D}-1}, \alpha^{\mathrm{D}-2}, \alpha^{\mathrm{D}-1} + \alpha^{\mathrm{D}-2}\right\}$. Notice that the elements of $\mathcal{S}_1$ are sorted in ascending order for any α and D. Now, we recursively define the sets $\mathcal{S}_i$ as follows:

$$\mathcal{S}_i := \left\{\mathcal{S}_{i-1}, \mathcal{S}_{i-1} + \alpha^{\mathrm{D}-1-i}\right\}, 2 \leq i \leq \mathrm{D} - 1 \tag{40}$$

Our hypothesis is that for every $2 \leq i \leq \mathrm{D} - 1 \alpha \in (0{,}0.5]$ and D, the set $\mathcal{S}_i$ as defined in (40), is automatically sorted in ascending order. We prove this via induction. For $i = 2$, the sets $\mathcal{S}_1$ and $\mathcal{S}_1 + \alpha^{\mathrm{D}-3}$ are individually sorted. Moreover from (39), we can show that: $\max_{a \in \mathcal{S}_1} a = \alpha^{\mathrm{D}-1} + \alpha^{\mathrm{D}-2} < \alpha^{\mathrm{D}-3} = \min_{b \in \mathcal{S}_1 + \alpha^{\mathrm{D}-3}} b$. This shows that $\mathcal{S}_2$ is ordered, establishing the the base case of our induction. Now, assume $\mathcal{S}_i$ is ordered for some $2 \leq i \leq \mathrm{D} - 2$. We need to show that $\mathcal{S}_{i+1}$ is also ordered. As a result of the induction hypothesis, both $\mathcal{S}_i$ and $\mathcal{S}_i + \alpha^{\mathrm{D}-2-i}$ are ordered. Using the ordering of $\mathcal{S}_i$, we have: $\max_{a \in \mathcal{S}_i} a = \sum_{j=1}^{i+1} \alpha^{\mathrm{D}-j}, \min_{b \in \mathcal{S}_i + \alpha^{\mathrm{D}-2-i}} b = \alpha^{\mathrm{D}-(i+1)-1}$. From (39), we can conclude that $\max_{a \in \mathcal{S}_i} a < \min_{b \in \mathcal{S}_i + \alpha^{\mathrm{D}-2-i}} b$ and hence, $\mathcal{S}_{i+1}$ is also ordered. This completes the induction proof. Also, note that for $\alpha \in (0{,}0.5]$, we have $\Theta_{\alpha}^{\mathrm{sort}} = \mathcal{S}_{\mathrm{D}-1}$.

Let $\Delta_{\min}(\mathcal{S}_i)$ be the min. distance between the elements of the set $\mathcal{S}_i$. It is easy to see that $\Delta_{\min}(\mathcal{S}_i) = \Delta_{\min}\left(\mathcal{S}_i + \alpha^{\mathrm{D}-2-i}\right)$. Since $\mathcal{S}_i$ is sorted for $\alpha \in (0{,}0.5]$, $\Delta_{\min}(\mathcal{S}_i)$ is given by:

$$\begin{aligned}\Delta_{\min}(\mathcal{S}_i) &= \min\left(\Delta_{\min}(\mathcal{S}_{i-1}), \min_{x \in \mathcal{S}_{i-1} + \alpha^{\mathrm{D}-1-i}} x - \max_{y \in \mathcal{S}_{i-1}} y\right) \\ &= \min\left\{\Delta_{\min}(\mathcal{S}_{i-1}), \alpha^{\mathrm{D}-i-1} - \sum_{j=1}^{i} \alpha^{\mathrm{D}-j}\right\}.\end{aligned} \tag{41}$$

Now, we use induction to establish the following conjecture:

$$\Delta_{\min}(\mathcal{S}_i) = \alpha^{\mathrm{D}-1}, 1 \leq i \leq \mathrm{D} - 1 \tag{42}$$

For the base case $i = 1, \Delta_{\min}(\mathcal{S}_1) = \min\left(\alpha^{\mathrm{D}-1}, \alpha^{\mathrm{D}-2} - \alpha^{\mathrm{D}-1}\right) = \alpha^{\mathrm{D}-1}$, where the last equality holds since $\alpha \in \left(0{,}0.5\right] \Rightarrow \alpha^{\mathrm{D}-1}\left(1/\alpha - 1\right) \geq \alpha^{\mathrm{D}-1}$. Suppose (42) holds for some $1 \leq i \leq \mathrm{D} - 2$. From the definition of $\Delta_{\min}(\mathcal{S}_{i+1})$ and the induction hypothesis that $\Delta_{\min}(\mathcal{S}_i) = \alpha^{D-1}$, it follows that $\Delta_{\min}(\mathcal{S}_{i+1}) = \min\left\{\alpha^{\mathrm{D}-1}, \alpha^{\mathrm{D}-(i+1)-1} - \sum_{j=1}^{i+1} \alpha^{\mathrm{D}-j}\right\}$. Again, from the definition of $\Delta_{\min}(\mathcal{S}_i)$ in (41), and the induction hypothesis we also have $\alpha^{\mathrm{D}-i-1} - \sum_{j=1}^{i} \alpha^{\mathrm{D}-j} \geq \Delta_{\min}(\mathcal{S}_i) = \alpha^{\mathrm{D}-1}$. Using this and the fact that $\alpha \leq 0.5$, we can show:

$$\alpha^{\mathrm{D}-i-2}-\alpha^{\mathrm{D}-i-1}-\sum_{j=1}^{i}\alpha^{\mathrm{D}-j}\geq\alpha^{\mathrm{D}-i-2}-2\alpha^{\mathrm{D}-i-1}+\alpha^{\mathrm{D}-1}$$
$$\geq\alpha^{\mathrm{D}-1}+\alpha^{\mathrm{D}-i-1}\left(1/\alpha-2\right)\geq\alpha^{\mathrm{D}-1}$$

Therefore $\Delta_{\min}(\mathcal{S}_{i+1})=\min\left\{\alpha^{\mathrm{D}-1},\alpha^{\mathrm{D}-i-2}-\sum_{j=1}^{i+1}\alpha^{\mathrm{D}-j}\right\}=\alpha^{D-1}$. Thus, we can conclude that $\Delta_{\min}\left(\alpha,\mathrm{D}\right)=\Delta_{\min}(\mathcal{S}_{\mathrm{D}-1})=\alpha^{\mathrm{D}-1}$. □

APPENDIX D: Proof of Theorem 3

Proof. The probability of incorrectly identifying $\mathrm{x}_{\mathrm{hi}}{}^{(n)}$ from a single measurement $c_e[n]$ is given by

$$p_e:=\mathbb{P}\left(\hat{\mathbf{x}}_{\mathrm{hi}}{}^{(n)}\neq\mathbf{x}_{\mathrm{hi}}{}^{(n)}\right)=\sum_{k=0}^{l_{\mathrm{D}}}\mathbb{P}\left(\hat{\mathbf{x}}_{\mathrm{hi}}{}^{(n)}\neq\mathbf{x}_{\mathrm{hi}}{}^{(n)}\mid\mathbf{x}_{\mathrm{hi}}{}^{(n)}=\tilde{\mathbf{v}}_k\right)\mathbb{P}\left(\mathbf{x}_{\mathrm{hi}}{}^{(n)}=\tilde{\mathbf{v}}_k\right)$$

Given a binary vector $\mathbf{z}\in\{0,1\}^{\mathrm{D}}$, define the function $\psi\left(\mathbf{z}\right):=\sum_{k=1}^{\mathrm{D}}z_k$, which denotes the count of ones in **z**. Since the noisy observations are given by $c_e[n]=c[n]+e[n]$, where $e\left[n\right]=w\left[n\right]-\alpha^{D}w\left[n-1\right]$, it follows from assumption (A2) that $e\left[n\right]\sim\mathcal{N}\left(0,\sigma_1^2\right)$ where $\sigma_1^2=\left(1+\alpha^{2\mathrm{D}}\right)\sigma^2$. From [27], we obtain $\mathbb{P}\left(\hat{\mathbf{x}}_{\mathrm{hi}}{}^{(n)}\neq\mathbf{x}_{\mathrm{hi}}{}^{(n)}\mid\mathbf{x}_{\mathrm{hi}}{}^{(n)}=\tilde{\mathbf{v}}_0\right)=\mathbb{P}(e[n]\in\mathcal{E}_0)=Q\left(\alpha^{\mathrm{D}-1}/(2\sigma_1)\right)$. Similarly, $\mathbb{P}\left(\hat{\mathbf{x}}_{\mathrm{hi}}{}^{(n)}\neq\mathbf{x}_{\mathrm{hi}}{}^{(n)}\mid\mathbf{x}_{\mathrm{hi}}{}^{(n)}=\tilde{\mathbf{v}}_{l_{\mathrm{D}}}\right)=\mathbb{P}(e\left[n\right]\in\mathcal{E}_{l_D})=Q\left(\left(\tilde{\theta}_{l_{\mathrm{D}}}-\tilde{\theta}_{l_{\mathrm{D}}-1}\right)/(2\sigma_1)\right)\equiv Q\left(\alpha^{\mathrm{D}-1}/(2\sigma_1)\right)$. The last equality follows from the fact that $\tilde{\theta}_{l_{\mathrm{D}}}-\tilde{\theta}_{l_{\mathrm{D}}-1}=\alpha^{\mathrm{D}-1}$. Finally, when conditioned on $\mathrm{x}_{\mathrm{hi}}{}^{(n)}=\tilde{\mathbf{v}}_k$ for $0<k<l_{\mathrm{D}}$, from (26), we obtain $\mathbb{P}\left(\hat{\mathbf{x}}^{(n)}\neq\mathrm{x}_{\mathrm{hi}}{}^{(n)}\mid\mathbf{x}_{\mathrm{hi}}{}^{(n)}=\tilde{\mathbf{v}}_k\right)=\mathbb{P}\left(e\left[n\right]\in\mathcal{E}_k\right)=Q\left(\frac{\tilde{\theta}_k-\tilde{\theta}_{k-1}}{2\sigma_1}\right)+Q\left(\frac{\tilde{\theta}_{k+1}-\tilde{\theta}_k}{2\sigma_1}\right)$. Due to Assumption (**A1**) on $\mathbf{x}_{\mathrm{hi}}$, we have $\mathbb{P}\left(\mathbf{x}_{hi}{}^{(n)}=\tilde{\mathbf{v}}_k\right)=p^{\psi(\tilde{\mathbf{v}}_k)}(1-p)^{D-\psi(\tilde{\mathbf{v}}_k)}$. Therefore, p_e is given by

$$p_e=Q\left(\alpha^{\mathrm{D}-1}/(2\sigma_1)\right)(1-p)^{\mathrm{D}}+Q\left(\alpha^{\mathrm{D}-1}/(2\sigma_1)\right)p^{\mathrm{D}}+\sum_{k=1}^{l_{\mathrm{D}}-1}\left(Q\left(\frac{\tilde{\theta}_k-\tilde{\theta}_{k-1}}{2\sigma_1}\right)+Q\left(\frac{\tilde{\theta}_{k+1}-\tilde{\theta}_k}{2\sigma_1}\right)\right)p^{\psi(\mathbf{v}_k)}(1-p)^{\mathrm{D}-\psi(\mathbf{v}_k)} \tag{43}$$

The spike train x_{hi} is incorrectly decoded if at least one of the blocks are decoded incorrectly, hence, the total probability of error is given by:

$$\mathbb{P}\left(\underset{n=0}{\overset{M}{\cup}}\widehat{\mathbf{x}}^{(n)} \neq \mathbf{x}_{\text{hi}}(n)\right) \leq \sum_{n=0}^{M-1}\mathbb{P}\left(\widehat{\mathbf{x}}^{(n)} \neq \mathbf{x}_{\text{hi}}(n)\right) = M p_e \overset{(a)}{\leq} 2MQ$$
$$(\Delta\theta_{\min}(\alpha,\mathrm{D})/(2\sigma_1)) \sum_{j=0}^{\mathrm{D}} p^j(1-p)^{\mathrm{D}-j}\binom{\mathrm{D}}{j}$$
$$\overset{(b)}{\leq} 2M\exp\left(-\Delta\theta_{\min}^2(\alpha,\mathrm{D})/\left(4\sigma_1^2\right)\right) \tag{44}$$

where the first inequality follows from union bound and second equality is a consequence of (43). The inequality *(a)* follows from the monotonically decreasing property of $Q(.)$ function and the sum can be re-written by grouping all terms with the same count, i.e., $\psi(\mathbf{v}_k) = j$. The inequality *(b)* follows from the inequality $Q(x) \leq \exp(-x^2/2)$ for $x > 0$. If the SNR condition (28) holds then from (44) the total probability of error is bounded by δ. □

Appendix E: Proof of Theorem 4

Proof. We first begin by showing that $\alpha \in \mathcal{F}_{\mathrm{D}}$ implies that (31). holds and hence the mapping of spikes with the same counts are clustered. Notice that for $k = 0, \theta_{\max}^k = \theta_{\min}^k = 0$. For $k \geq 1$, it is easy to verify that $\theta_{\max}^k$ and $\theta_{\min}^k$ are attained by the spiking patterns 00…1111 (with k consecutive spikes at the indices $\mathrm{D}-k+1$ to D) and 111…000 (with consecutive spikes at the indices 1 to k), which allows us to simplify (31) as $\alpha^{\mathrm{D}-1} > 0$ for $k = 0$ and $\sum_{i=1}^{k+1}\alpha^{\mathrm{D}-i} > \sum_{j=0}^{k-1}\alpha^j, k = 1, \cdots, \mathrm{D}-1$. The values of α that satisfy each of these relations can be described by the following sets:

$$\mathcal{G}_0 = \left\{\alpha \in (0,1) \mid \alpha^{\mathrm{D}-1} > 0\right\}, \mathcal{G}_k = \{\alpha \in (0,1) \mid r_k(\alpha) < 0\},$$

where $r_k(\alpha) = \alpha^{\mathrm{D}} - \alpha^{\mathrm{D}-k-1} - \alpha^k + 1$ for $1 \leq k \leq \mathrm{D}-1$. It is easy to see that $\mathcal{F}_{\mathrm{D}} = \mathcal{G}_{k0}$. Observe that the relations are symmetric, i.e., $\mathcal{G}_k = \mathcal{G}_{\mathrm{D}-k-1}$. Furthermore, for $1 \leq k \leq \mathrm{D}/2$, we show that $\mathcal{G}_k \subseteq \mathcal{G}_{k-1}$ as follows. Trivially, $\mathcal{G}_1 \subset \mathcal{G}_0$. For $2 \leq k \leq D/2$, observe that $r_k(\alpha) - r_{k-1}(\alpha) = \alpha^{\mathrm{D}-k}(1-1/\alpha) - \alpha^k(1-1/\alpha) = (1/\alpha - 1)(\alpha^k - \alpha^{\mathrm{D}-k}) \geq 0$. Therefore, $\alpha \in \mathcal{G}_k \Rightarrow \alpha \in \mathcal{G}_{k-1}, k = 1,2\cdots, k_0$. Moreover, since $\mathcal{G}_k = \mathcal{G}_{\mathrm{D}-k-1}$, it follows that $\mathcal{F}_{\mathrm{D}} = \mathcal{G}_{k_0} = \cap_{k=0}^{\mathrm{D}-1}\mathcal{G}_k$. Hence, $\alpha \in \mathcal{F}_{\mathrm{D}} \Rightarrow \alpha \in \mathcal{G}_i$ for all $0 \leq i \leq \mathrm{D}-1$, which implies that (31) holds. If the noise perturbation satisfies $|w[n]| < \Delta_{\min}^{\mathrm{c}}(\alpha,\mathrm{D})/4$, it implies $|e[n]| < \Delta_{\min}^{\mathrm{c}}(\alpha,\mathrm{D})/2$. For any block $\mathbf{x}_{\text{hi}}{}^{(n)} \in \mathcal{C}_k^{\mathrm{D}}, \theta_{\min}^k \leq \mathbf{h}_\alpha^\top\mathbf{x}_{\text{hi}}{}^{(n)} \leq \theta_{\max}^k$. If $|e[n]| < \Delta_{\min}^{\mathrm{c}}(\alpha,\mathrm{D})/2$, we have

$$\mathbf{h}_\alpha^\top\mathbf{x}_{\text{hi}}{}^{(n)} + e\left[n\right] < \theta_{\max}^k + \frac{\Delta_{\min}^c(\alpha,\mathbf{D})}{2} < \theta_{\max}^k + \frac{\theta_{\min}^{k+1} - \theta_{\max}^k}{2}$$

$$\mathbf{h}_\alpha^\top\mathbf{x}_{\text{hi}}^{(n)} + e[n] > \theta_{\min}^k - \frac{\Delta_{\min}^c(\alpha,\mathbf{D})}{2} > \theta_{\min}^k - \frac{\theta_{\min}^k - \theta_{\max}^{k-1}}{2}$$

This shows that whenever $\alpha \in \mathscr{F}_{\mathrm{D}}$, the condition $|e[n]| < \Delta^{\mathrm{c}}_{\min}(\alpha, \mathrm{D})/2$ is sufficient for (33) to hold $\forall\gamma[n]$ and hence the spike count can be exactly recovered. □

APPENDIX F: Amplitude Estimation

We suggest a procedure to estimate the binary amplitude A, if it is unknown. We first evaluate the signal $c[n]$ from different time instants $n = 1, 2, \cdots, M - 1$. For some $1 \leq n_0 \leq M - 1$, we estimate a set $\mathscr{A} = \{A_k\}$ of candidate amplitudes: $A_k := c[n_0]/\mathbf{h}_\alpha^T\mathbf{v}_k$ where $\mathbf{v}_k \in \mathscr{S}_{\mathrm{all}}$. Only a certain amplitudes can generate $c[n_0]$ from a valid binary spiking pattern $\mathbf{v}_k \in \mathscr{S}_{\mathrm{all}}$. Our goal is to prune $\mathscr{A}$ by sequentially eliminating certain candidate amplitudes from the set based on a consistency test across the remaining measurements $c[n]$. At the t^{th} stage $(t = 2, 3, \cdots)$, for every remaining candidate amplitude $A_k \in \mathscr{A}$, we perform the following consistency test with $c[n]$, to identify if a candidate amplitude can potentially generate the corresponding measurement $c[n]$. Suppose there exists a spiking pattern $\mathbf{v}_l \in \mathscr{S}_{\mathrm{all}}$ such that

$$c[n] = A_k\mathbf{h}_\alpha^T\mathbf{v}_l \tag{45}$$

then A_k remains a valid candidate. If we cannot find a corresponding $\mathbf{v}_l \in \mathscr{S}_{\mathrm{all}}$ for an amplitude A_k, we remove it, $\mathscr{A} = \mathscr{A} \setminus A_k$. In presence of noise, (45) can be modified to allow a tolerance γ as we may not find an exact match. The tolerance γ is chosen to be 0.5 in the experiments on the GENIE dataset. This procedure prunes out possible values for the amplitude by leveraging the shared amplitude across multiple measurements $c[n]$.

References

[1]. Small A and Stahlheber S, “Fluorophore localization algorithms for super-resolution microscopy,” Nature methods, vol. 11, no. 3, pp. 267–279, 2014. [PubMed: 24577277]

[2]. Brette R and Destexhe A, Handbook of neural activity measurement. Cambridge University Press, 2012.

[3]. Vogelstein JT, Watson BO, Packer AM, Yuste R, Jedynak B, and Paninski L, “Spike inference from calcium imaging using sequential monte carlo methods,” Biophysical journal, vol. 97, no. 2, pp. 636–655, 2009. [PubMed: 19619479]

[4]. Deneux T, Kaszas A, Szalay G, Katona G, Lakner T, Grinvald A, Rózsa B, and Vanzetta I, “Accurate spike estimation from noisy calcium signals for ultrafast three-dimensional imaging of large neuronal populations in vivo,” Nature communications, vol. 7, p. 12190, 2016.

[5]. Yang S and Hanzo L, “Fifty years of mimo detection: The road to large-scale mimos,” IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 1941–1988, 2015.

[6]. Donoho DL, “Superresolution via sparsity constraints,” SIAM journal on mathematical analysis, vol. 23, no. 5, pp. 1309–1331, 1992.

[7]. Candès EJ and Fernandez-Granda C, “Towards a mathematical theory of super-resolution,” Communications on pure and applied Mathematics, vol. 67, no. 6, pp. 906–956, 2014.

[8]. Li W, Liao W, and Fannjiang A, “Super-resolution limit of the esprit algorithm,” IEEE Transactions on Information Theory, vol. 66, no. 7, pp. 4593 – 4608, 2020

[9]. Batenkov D, Goldman G, and Yomdin Y, “Super-resolution of near-colliding point sources,” Information and Inference: A Journal of the IMA, vol. 10, no. 2, pp. 515–572, 2021.

[10]. Schiebinger G, Robeva E, and Recht B, “Superresolution without separation,” Information and Inference: A Journal of the IMA, vol. 7, no. 1, pp. 1–30, 2017.

[11]. Bendory T, “Robust recovery of positive stream of pulses,” IEEE Transactions on Signal Processing, vol. 65, no. 8, pp. 2114–2122, 2017.

[12]. Liao W and Fannjiang A, “Music for single-snapshot spectral estimation: Stability and super-resolution,” Applied and Computational Harmonic Analysis, vol. 40, no. 1, pp. 33–67, 2016.

[13]. Qiao H and Pal P, “Guaranteed localization of more sources than sensors with finite snapshots in multiple measurement vector models using difference co-arrays,” IEEE Transactions on Signal Processing, vol. 67, no. 22, pp. 5715–5729, 2019.

[14]. —, “A non-convex approach to non-negative super-resolution: Theory and algorithm,” in ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 4220–4224.

[15]. Qiao H, Shahsavari S, and Pal P, “Super-resolution with noisy measurements: Reconciling upper and lower bounds,” in ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 9304–9308.

[16]. Shahsavari S, Millhiser J, and Pal P, “Fundamental trade-offs in noisy super-resolution with synthetic apertures,” in ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 4620–4624

[17]. Qiao H and Pal P, “On the modulus of continuity for noisy positive super-resolution,” in 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018, pp. 3454–3458.

[18]. Chi Y and Da Costa MF, “Harnessing sparsity over the continuum: Atomic norm minimization for superresolution,” IEEE Signal Processing Magazine, vol. 37, no. 2, pp. 39–57, 2020.

[19]. Bhaskar BN, Tang G, and Recht B, “Atomic norm denoising with applications to line spectral estimation,” IEEE Transactions on Signal Processing, vol. 61, no. 23, pp. 5987–5999, 2013.

[20]. Grewe BF, Langer D, Kasper H, Kampa BM, and Helmchen F, “High-speed in vivo calcium imaging reveals neuronal network activity with near-millisecond precision,” Nature methods, vol. 7, no. 5, p. 399, 2010. [PubMed: 20400966]

[21]. Pnevmatikakis EA, Soudry D, Gao Y, Machado TA, Merel J, Pfau D, Reardon T, Mu Y, Lacefield C, Yang W et al. , “Simultaneous denoising, deconvolution, and demixing of calcium imaging data,” Neuron, vol. 89, no. 2, pp. 285–299, 2016. [PubMed: 26774160]

[22]. Schmidt R, “Multiple emitter location and signal parameter estimation,” IEEE transactions on antennas and propagation, vol. 34, no. 3, pp 276–280, 1986.

[23]. Roy R and Kailath T, “Esprit-estimation of signal parameters via rotational invariance techniques,” IEEE Transactions on acoustics, speech, and signal processing, vol. 37, no. 7, pp. 984–995, 1989.

[24]. Hua Y and Sarkar TK, “Matrix pencil method for estimating parameters of exponentially damped/undamped sinusoids in noise,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 38, no. 5, pp. 814–824, 1990

[25]. Bernstein B and Fernandez-Granda C, “Deconvolution of point sources: a sampling theorem and robustness guarantees,” Communications on Pure and Applied Mathematics, vol. 72, no. 6, pp. 1152–1230, 2019.

[26]. Koulouri A, Heins P, and Burger M, “Adaptive superresolution in deconvolution of sparse peaks,” IEEE Transactions on Signal Processing, vol. 69, pp. 165–178, 2020.

[27]. Morgenshtern VI and Candes EJ, “Super-resolution of positive sources: The discrete setup,” SIAM Journal on Imaging Sciences, vol. 9, no. 1, pp. 412–444, 2016.

[28]. Batenkov D, Bhandari A, and Blu T, “Rethinking super-resolution: the bandwidth selection problem,” in ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019, pp. 5087–5091.

[29]. Da Costa MF and Dai W, “A tight converse to the spectral resolution limit via convex programming,” in 2018 IEEE International Symposium on Information Theory (ISIT). IEEE, 2018, pp. 901–905.

[30]. Blu T, Dragotti P-L, Vetterli M, Marziliano P, and Coulot L, “Sparse sampling of signal innovations,” IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 31–40, 2008.

[31]. Urigüen JA, Blu T, and Dragotti PL, “Fri sampling with arbitrary kernels,” IEEE Transactions on Signal Processing, vol. 61, no. 21, pp. 5310–5323, 2013.

[32]. Onativia J, Schultz SR, and Dragotti PL, “A finite rate of innovation algorithm for fast and accurate spike detection from two-photon calcium imaging,” Journal of neural engineering, vol. 10, no. 4, p. 046017, 2013. [PubMed: 23860257]

[33]. Tur R, Eldar YC, and Friedman Z, “Innovation rate sampling of pulse streams with application to ultrasound imaging,” IEEE Transactions on Signal Processing, vol. 59, no. 4, pp. 1827–1842, 2011.

[34]. Rudresh S and Seelamantula CS, “Finite-rate-of-innovation-sampling-based super-resolution radar imaging,” IEEE Transactions on Signal Processing, vol. 65, no. 19, pp. 5021–5033, 2017.

[35]. Stojnic M, “Recovery thresholds for l_1 optimization in binary compressed sensing,” in 2010 IEEE International Symposium on Information Theory. IEEE, 2010, pp. 1593–1597.

[36]. Keiper S, Kutyniok G, Lee DG, and Pfander GE, “Compressed sensing for finite-valued signals,” Linear Algebra and its Applications, vol. 532, pp. 570–613, 2017.

[37]. Flinth A and Keiper S, “Recovery of binary sparse signals with biased measurement matrices,” IEEE Transactions on Information Theory, vol. 65, no. 12, pp. 8084–8094, 2019.

[38]. Fosson SM and Abuabiah M, “Recovery of binary sparse signals from compressed linear measurements via polynomial optimization,” IEEE Signal Processing Letters, vol. 26, no. 7, pp. 1070–1074, 2019.

[39]. Tian Z, Leus G, and Lottici V, “Detection of sparse signals under finite-alphabet constraints,” in 2009 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2009, pp. 2349–2352.

[40]. Sarangi P and Pal P, “No relaxation: Guaranteed recovery of finite-valued signals from undersampled measurements,” in ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 5440–5444.

[41]. —, “Measurement matrix design for sample-efficient binary compressed sensing,” IEEE Signal Processing Letters, 2022.

[42]. Razavikia S, Amini A, and Daei S, “Reconstruction of binary shapes from blurred images via hankel-structured low-rank matrix recovery,” IEEE Transactions on Image Processing, vol. 29, pp. 2452–2462, 2019.

[43]. Friedrich J, Zhou P, and Paninski L, “Fast online deconvolution of calcium imaging data,” PLoS computational biology, vol. 13, no. 3, p. e1005423, 2017. [PubMed: 28291787]

[44]. Jewell SW, Hocking TD, Fearnhead P, and Witten DM, “Fast nonconvex deconvolution of calcium imaging data,” Biostatistics, vol. 21, no. 4, pp. 709–726, 2020. [PubMed: 30753436]

[45]. Sarangi P, Hücümenoğlu MC, and Pal P, “Effect of undersampling on non-negative blind deconvolution with autoregressive filters,” in ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020, pp. 5725–5729.

[46]. Rupasinghe A and Babadi B, “Robust inference of neuronal correlations from blurred and noisy spiking observations,” in 2020 54th Annual Conference on Information Sciences and Systems (CISS). IEEE, 2020, pp. 1–5.

[47]. Sidorov N, “Almost every number has a continuum of β-expansions,” The American Mathematical Monthly, vol. 110, no. 9, pp. 838–842, 2003.

[48]. Glendinning P and Sidorov N, “Unique representations of real numbers in non-integer bases,” Mathematical Research Letters, vol. 8, no. 4, pp. 535–543, 2001

[49]. Rényi A, “Representations for real numbers and their ergodic properties,” Acta Mathematica Academiae Scientiarum Hungarica, vol. 8, no. 3–4, pp. 477–493, 1957.

[50]. Frougny C and Solomyak B, “Finite beta-expansions,” Ergodic Theory Dynam. Systems, vol. 12, no. 4, pp. 713–723, 1992

[51]. Komornik V and Loreti P, “Expansions in noninteger bases.” Integers, vol. 11, no. A9, p. 30, 2011.

[52]. Hubel DH and Wiesel TN, “Receptive fields of single neurones in the cat’s striate cortex,” The Journal of physiology, vol. 148, no. 3, p. 574,1959 [PubMed: 14403679]

[53]. Chen T-W, Wardill TJ, Sun Y, Pulver SR, Renninger SL, Baohan A, Schreiter ER, Kerr RA, Orger MB, Jayaraman V et al. , “Ultrasensitive fluorescent proteins for imaging neuronal activity,” Nature, vol. 499, no. 7458, pp. 295–300, 2013. [PubMed: 23868258]

[54]. H. K. S. c. GENIE Project, Janelia Farm Campus, “Simultaneous imaging and loose-seal cell-attached electrical recordings from neurons expressing a variety of genetically encoded calcium indicators,” CRCNS. org, 2015.

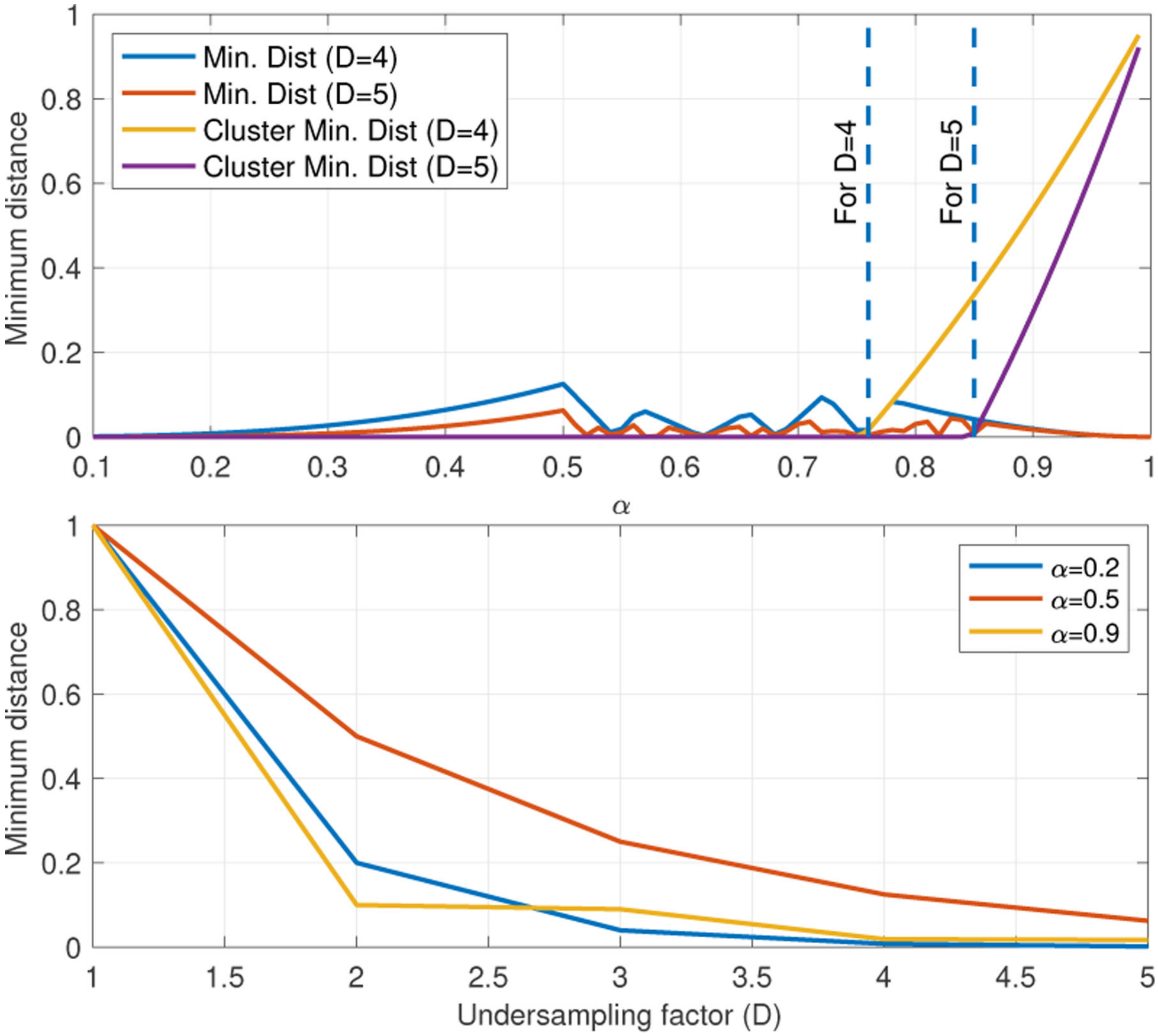


Fig. 1:

Variation of $\Delta\theta_{\min}(\alpha, \mathrm{D})$ as a function of undersampling factor D and α. The cluster-distance $\Delta^{c}_{\min}(\alpha, \mathrm{D})$ vs. α is also overlaid. Each dotted line denotes the start of the interval $\mathscr{F}_{\mathrm{D}}$.

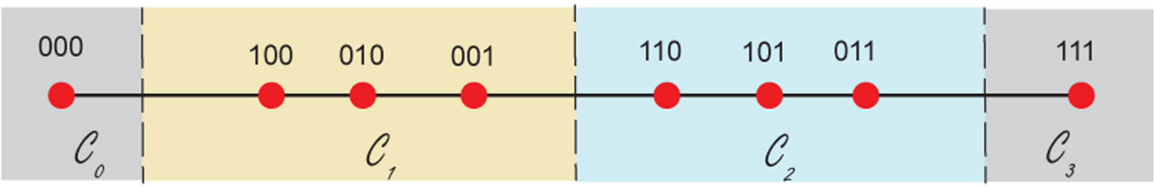


Fig. 2:

Visualization of the sets $\mathscr{C}_k^{\mathrm{D}}$ for D = 3. In this scenario, the spiking patterns corresponding to the same count are clustered together and hence, are favorable for robust count estimation.

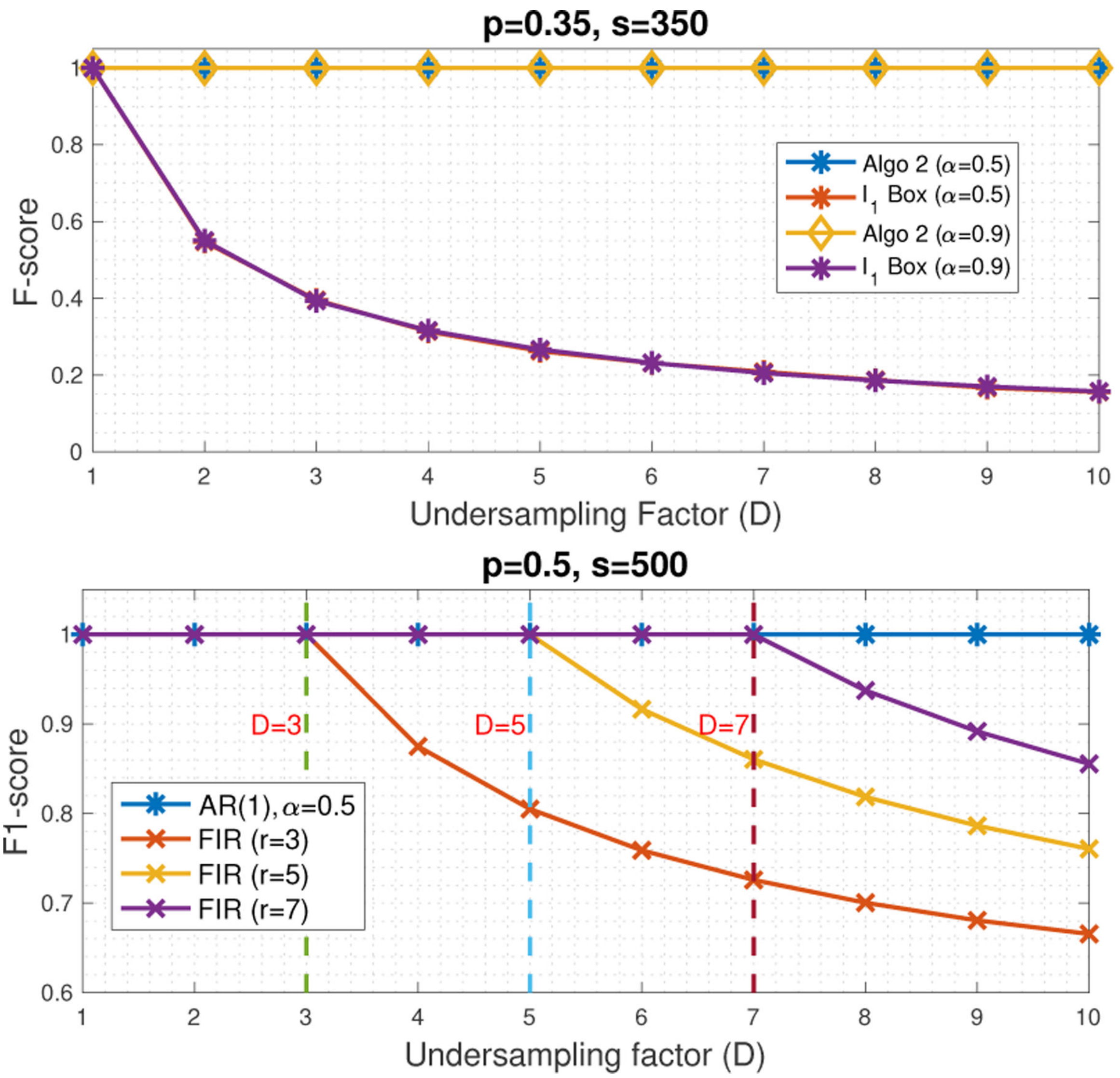


Fig. 3:

(Top) Quantitative comparison of Algorithm 2 against box-constrained l_1 minimization method with noiseless measurements (with tolerance $t_0 = 0$). (Bottom) (Role of Filter Memory): Average F-score vs. D for FIR and IIR (AR(1)) filters. Each dotted line indicates the corresponding theoretical transition point (D = r).

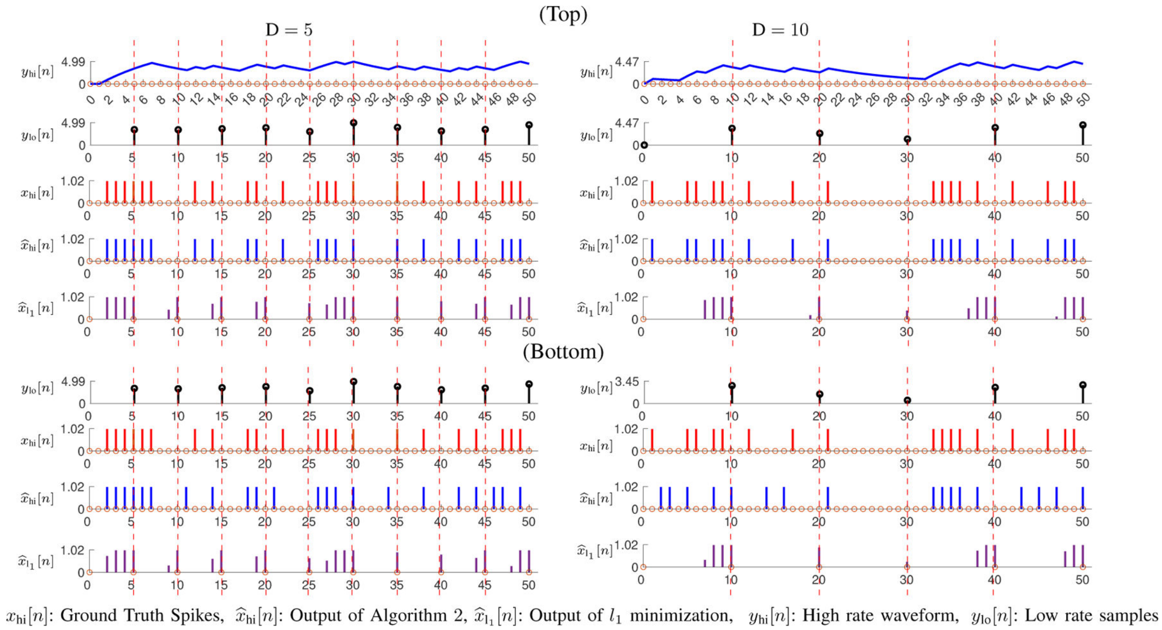


Fig. 4:

Qualitative comparison of Algorithm 2 and box-constrained l_1 minimization on simulated data. For each simulation noisy measurements are generated with $\alpha = 0.9$ such that the noise realization (Top) obeys the bound $|w[n]| \leq \Delta\theta_{\min}$ (from Theorem 2) and (Bottom) violates the bound. For larger noise (Bottom), the spike recovery is imperfect but the spike count can still be exactly recovered using Algorithm 2.

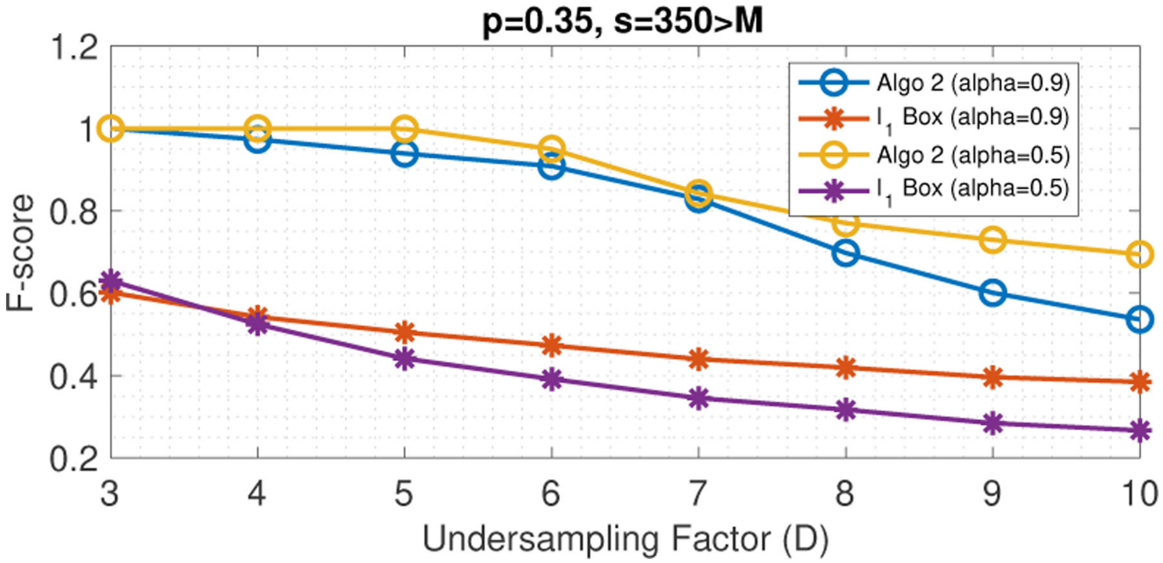


Fig. 5:

Spike detection performance with noisy measurements. (Top) F-score vs. D for different filter parameters $\alpha(\sigma = 0.01)$. Here, $L = 1000$ and expected sparsity $s = 350$ where we operate in the regime $s > M$. The F-score is computed with a tolerance of $t_0 = 2$.

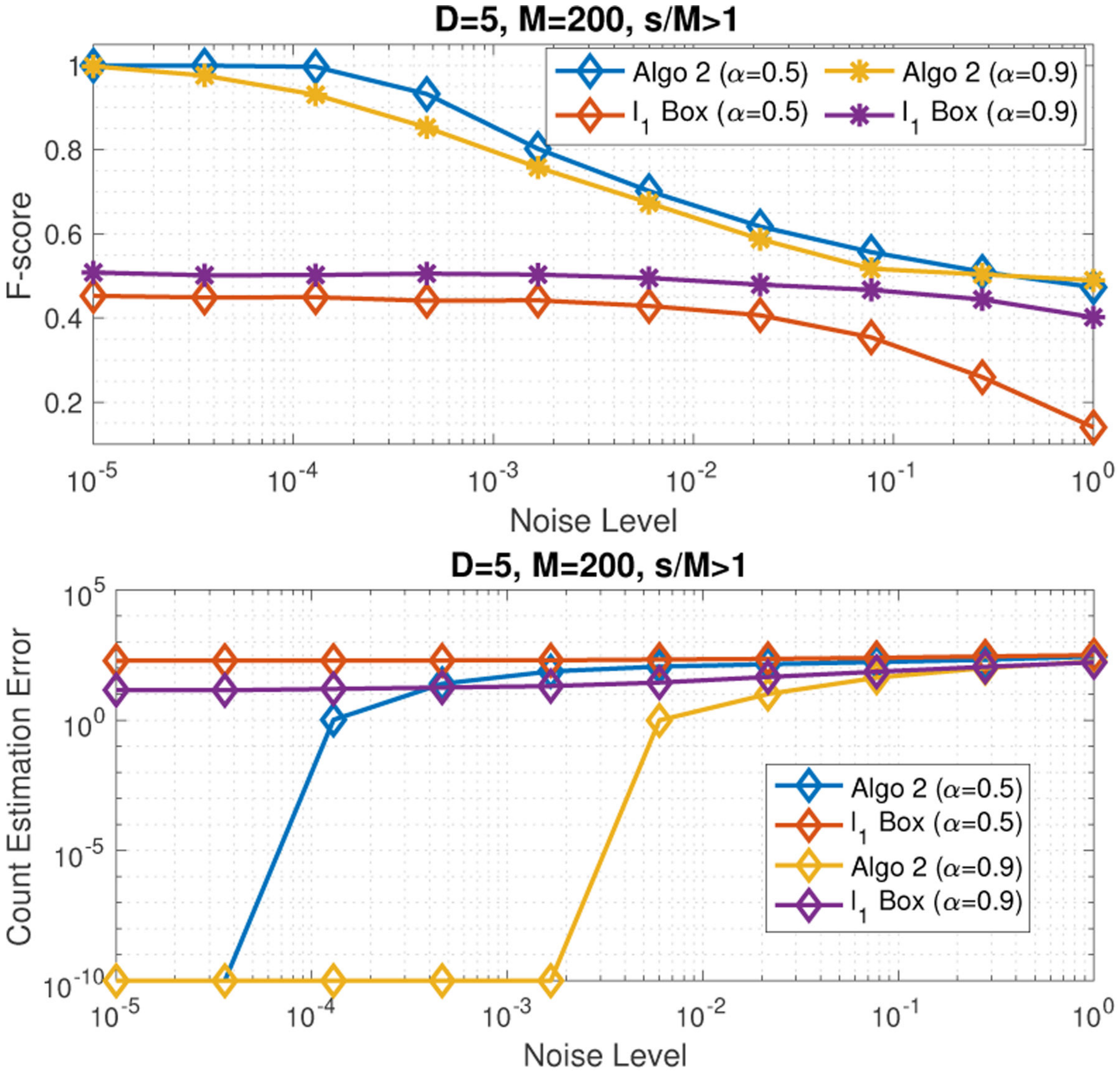


Fig. 6:

Spike detection performance with noisy measurements for different filter parameters α. (Top) F-score vs. noise level (σ) (Bottom) Count estimation error vs. noise level. Here, $L = 1000$ and expected sparsity is fixed at $s = 350$ where we operate in the regime $s > M$. The F-score is computed with a tolerance of $t_0 = 2$.

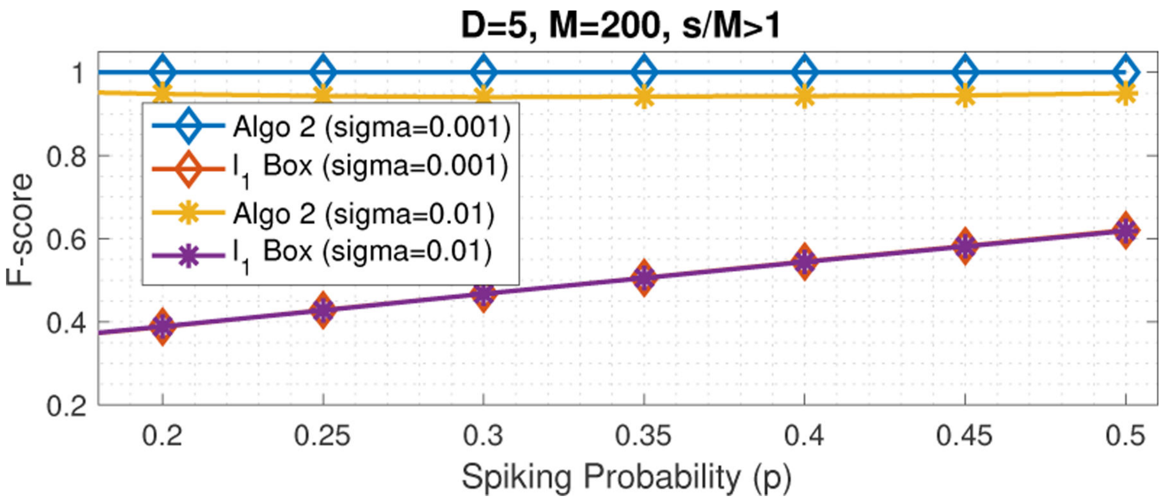


Fig. 7:

Spike detection performance with noisy measurements. F-score vs. spiking probability (p) for different noise levels σ (fix $\alpha = 0.9$, D = 5, $L = 1000$) in the extreme compression regime $s > M$.

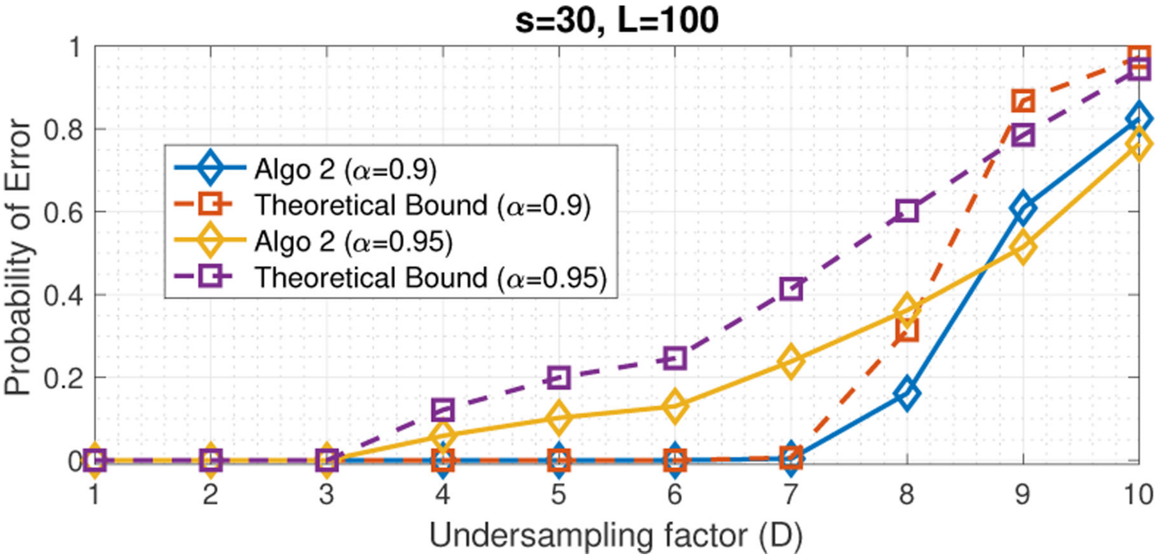


Fig. 8:

Probability of erroneous detection of high-rate spikes $\mathbf{x}_{\text{hi}} \in \{0,1\}^L$ as a function of the undersampling factor D. Theoretical upper bounds are overlaid using dotted lines. Here, $L = 100$.

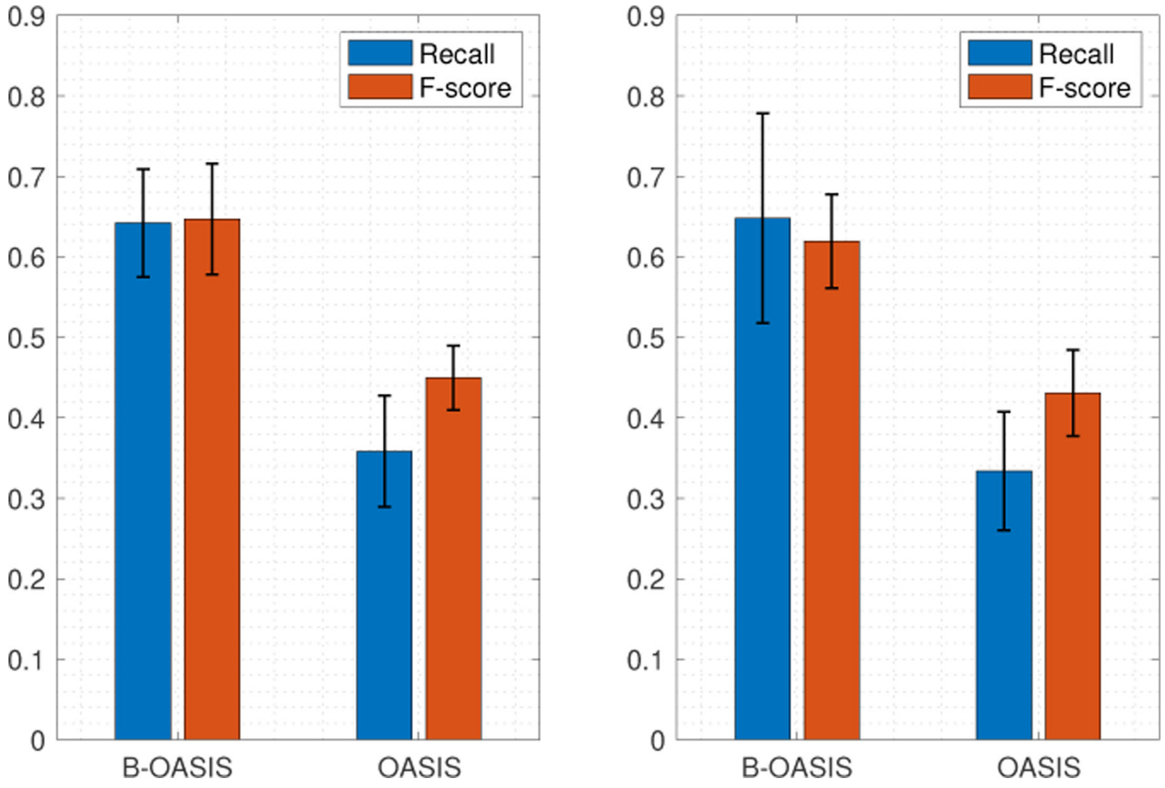


Fig. 9:

Spike detection performance of OASIS and B-OASIS on GCaMP6f dataset sampled at (Left) 60 Hz and (Right) 30 Hz. We compare the average F-score of data points where the F-score of OASIS is < 0.5. Standard deviation is depicted using the error bars.

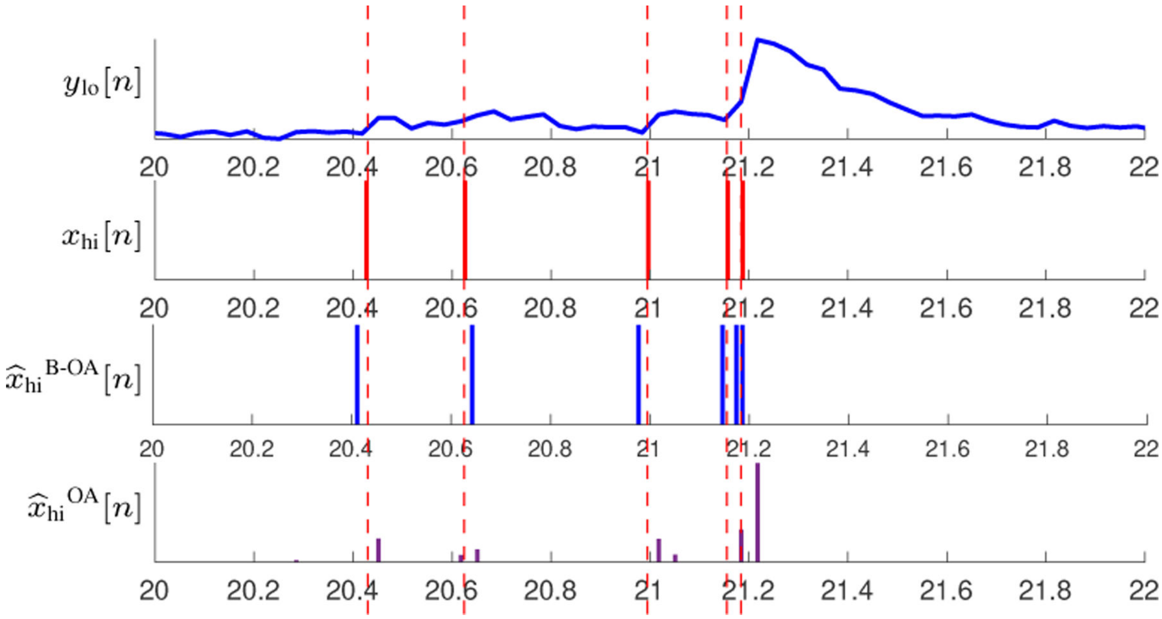


Fig. 10:

Example of spike reconstruction on GENIE dataset (GCaMP6f indicator) using OASIS and B-OASIS (binary augmented) with calcium signal sampled at 30 Hz.