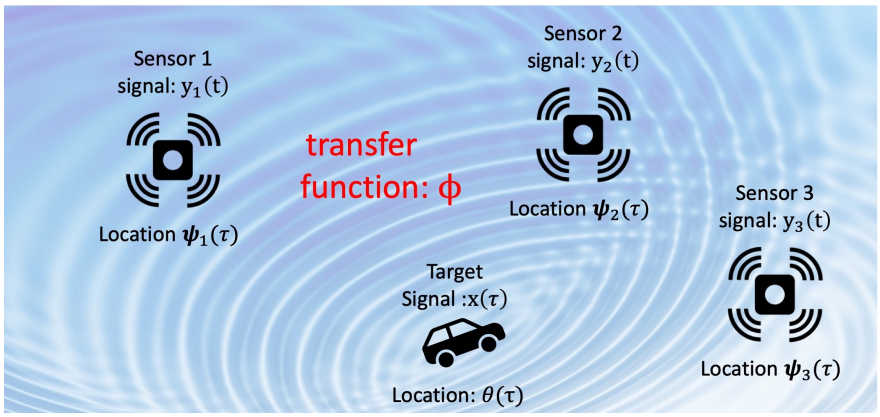

Figure 1: An example of a sensor network. The goal of the network is to track a car. The location of the car as a function of time is $\theta(\tau)$, and the goal of the network to produce $\hat{\theta}(\tau)$. The car emits a sound wave, which we denote by $x(\tau)$. The sound wave travels through the physical environment and arrives at ψ_i - the location of sensor i, where it is digitized and made into a time sequence $y_i(t)$. The transformation of the signal $x(\tau)$ into $y_i(t)$ is represented by a transfer function ϕ. In this car tracking example $y_i(t) = \phi(x(\tau), \theta(\tau), \psi_i(\tau))$ We are seeking an inverse transformation that would map the measurements vector $y_1(t), y_2(t), y_3(t)$ to an estimate of the car location $\hat{\theta}(t)$

1 Introduction

The fact that humans have *two* eyes and *two* ears has clear benefits. Our two eyes provide us with depth perception. Our two ears allow us to detect the direction from which a sound is coming.

The same holds for artificial sensors. Multiple sensors with overlapping receptive fields can produce better representations of the environment. However, to produce such representations, the data streams generated by the sensors need to be compared and combined. One approach is to send all of the signal to a center and use a single computer to combined them. This approach requires high bandwidth communication and powerful computers and does not scale well to networks with hundreds or thousands of sensors.

An alternative is *pushing computation to the edge*. Instead of performing all of the computation on a central machine, each sensor has some computing power which it uses to perform some of the computation and reduce the amount of information that needs to be sent. This reduction is not compressionas we are not asking for a reconstruction of the original signal. Instead what we want to produce is a high-level representation of the environment. For example, the location of an object, or a count of the people in a particular area. Computing this high level representation will usually require much less information from each sensors than a compressed version of the raw signal.

We propose a novel approach we call *Learning Sensor Networks*. This approach assumes that sensors are placed at fixed locations and remain there for a significant length of time. Each sensor uses *statistical learning* to model it's environment. It communicates with neighboring sensors in order to know their models and to calibrate parameters such as relative location. After an initial training period, which might last minutes, days or months depending on the context, the network can distinguish normal from abnormal behaviour. Based on that ability, it will transmit to the other sensors only a *sketch* [] which captures the novel, or salient aspect of the signals.

Learning sensor networks promise much lower energy and bandwidth consumption than current sensor networks. To realize this potential we plan to develop new new mathematical models, signal processing methods and distributed algorithms for computing and combining sketches.

2 Framework

This proposal combines several related lines of work. To facilitate the exposition, we start by introducing some terminology and notation that will be used throughout. **Figure 1** describes a simple sensor network that tracks a car using the sound waves the car is emitting.

We now expand this simple example into a more general framework. We assume that the network consists of a n sensors and m targets. Sensor i's state at time τ is denoted $\psi_i(\tau)$. Similarly, the state of target j at time τ is denoted $\theta_j(\tau)$. Here and in the rest of this section, we don't specify the spaces in which ψ_i or θ_j are members. This allows for a general introduction, and more which will be made more specific in later sections. When appropriate, we will denote the combined state of all targets by $\Theta(\tau)$ and the combined state of all sensor by $\Psi(\tau)$. The third state component is the state of physical environment in which the targets and sensors reside. We denote the state of the environment by $\mathbf{E}(\tau)$

The targets generate signals, which we call the *raw* signals. We denote the raw signal generated by target i as $\mathbf{x}_i(\tau)$. We denote the collection of all m signals by $\mathbf{X}(\tau)$. On the receiving end, each sensor i captures a digital signal $\mathbf{y}_i$. These digital signals are the inputs to the computations we will discuss. As the signals arrive at physically separated sensors, the computation is inherently distributed. The main goal of this proposal is to develop algorithms that achieve desired tasks with minimal communication between the sensors.

Rayan : How are we deciding what variables are in bold and what variables are not? For example, why is $y_i(t)$ not bold in the caption, but $\mathbf{x}_i(\tau)$ in bold?

The transfer function $\mathbf{\Phi}$ defines the way by which the raw signals $\mathbf{X}$ are transformed into the digitized signals $\mathbf{Y}$. This function represents both the point transfer function of the physical environment, the analog-to-digital transformation of the physical signal into a discrete time physical signal, and the noise that is added through this process. The transformation is defined by:[1]

$$\mathbf{Y} = \mathbf{\Phi}(\mathbf{X}, \Theta, \Psi, \mathbf{E}) \tag{1}$$

Many of the problems we plan to tackle in this proposal are inverse computation problems. We assume that some aspects of the physical space are known, i.e. we know a subset of $\mathbf{X}, \Theta, \Psi, \mathbf{E}$. Given the digitized signals $\mathbf{Y}$, our task is to estimate the unknown parts of the physical space. Reliable methods for computing such estimates exist. However, they typically require high communication bandwidth. The goal of this proposal is to find distributed estimation algorithms that achieve good performance while using significantly less communication.

2.1 Some specific tasks

We give a few specific examples of tasks. We will elaborate on some of these tasks below

1. **Target Localization:** Figure 1 depicts an archetypal target localization task. In this case the locations of the sensors Ψ and the state of the environment $\mathbf{E}$ are assumed to be fixed. A typical additional assumption, which is represented in the transfer function $\mathbf{\Phi}$, is that strongest signal corresponds to the straight line of transmission between the target and the sensors. A common approach to target localization is to estimate the delay between the arrival of the signal at different sensors by using some type of cross correlation []. This calculation is performed It is well known that placing sensors far from each other provides the most accurate localization. However, achieving this accuracy with bounded communication between the sensors remains a challenge.
2. **Sensor Calibration:** A common situation is that many sensors are installed in an existing environment such as a home, and the state of these sensors Ψ which would typically include location and orientation, is not known. Manual calibration is often labor intensive or impossible. The challenge is to design algorithms through which the network can self-calibrate. We consider two types of calibration, in the easier *active calibration* the system can control the generated signals, while in *blind calibration* the calibration has to be done using signals that are generated by the environment.
3. **Signal reconstruction** Voice based systems such as speakerphones, and voice activated computers, need to reconstruct the speech signal. Microphone arrays are sensor networks where the sensor is a microphone.

[1] Note that the transfer function $\mathbf{\Phi}$ operates on the whole sequences, not just on the sequence at a single time τ. That is because signal propagation takes time, so $\mathbf{Y}(t)$ depends on $\mathbf{X}$ at multiple time points.

Accurately reproducing the speech signal when there is more than one speaker is an open challenge. In this problem the location of the sensors is known, the goal is to reproduce the raw signal $\mathbf{X}$. This task is relatively easy when $\Theta, \Psi, \mathbf{E}$ are known and the transfer function $\mathbf{\Phi}$ is known and simple. It becomes significantly harder when some of $\Theta, \Psi, \mathbf{E}$ are unknown or when $\mathbf{\Phi}$ is complex, such as multi-path radio signals or reverberating audio signals.

4. **Optimizing sensor placement:** The accuracy of target localization depends on the location of the target and of the sensors. While the location of the target is not under our control, the location of the sensors is. Methods for optimizing the locations of the sensors will be described in Section sec:sensor-placement.
5. **Mapping the environment** Sometime the goal of the system is to estimate the environment $\mathbf{E}$. One example is to use Radar, Sonar or Lidar to create a 2D or 3D representation of the environment for a smart car. Another example, coming from seismology is to use controlled vibration sources and many acceleration sensors to map the subterranean earth. In these settings the locations of the sensors Ψ and the targets Θ (called transmitters in this context) is fixed and known, as is the raw (transmitted) signal $\mathbf{X}$. The goal is to deduce $\mathbf{E}$ from the collected signal $\mathbf{Y}$. (Peter, does this make sense to you, can you use this formulation in your sections about tomography and dictionaries?)
6. **Monitoring** In many situations the goal of the sensor network is to track the environment, identify trends and detect anomalies. Motivating examples include: Security systems, systems for monitoring patients or the elderly, highway monitoring and factory floor monitoring. Many of these environments are too complex to estimate a fully detailed representation. Instead, we suggest building a statistical model which implicitly captures the major degrees of freedom of the environment and the way they relate to major variables such as time of day and day of week. The challenge here is to learn such a model in an unsupervised or weakly supervised way, without heavy use of computational or communication resources. Here, we will propose, and rigorously analyze techniques based on Kernel methods combined with sketching and low dimensional binary embedding techniques. The combination of these tools will simultaneously facilitate performing various desired statistical tasks, while minimizing storage, communication, and computational costs.

3 Target Localization Using Minimal Number of Sensors

Yoav : Piya, can you use the notation I defined in the framework section? Also I would like to merge this subsection and the following one, which describes sensing geometry and the goal of minimizing the number of sensors. **Piya :** yes, done. A sensor network consisting of M sensing units aims to capture information of interest (often described in terms of parameters) regarding the physical environment by acquiring measurements in space (dictated by sensor locations) and in time (dictated by the sampling technique employed at each sensor). In many applications (especially those concerning high-resolution/super-resolution imaging), the goal is to detect certain (possibly time varying) parameters $\Theta(\tau)$ from certain targets (or sources) of interest in the environment by acquiring signals emitted by them. The methodologies proposed in this proposal will be primarily developed for passive sensing, although they can also be integrated into an active sensing scenario if the sensors are also allowed to actively emit signals for localizing targets.

Yoav : This describes a more general framework, using $\mathbb{C}^P$ (does that mean each coordinate is complex?). What is gained from this generality? maybe drop the general notation? Also how does high resolution/super-resolution fit here? If you have worked on such problem, I suggest you devote a paragraph and cite, rather than just mentioning in passing. **Piya :** I removed the complex notation. I will talk about super-resolution under review of difference sets - coming soon!

As an example, consider a network consisting of active radar units (for example, those mounted on autonomous vehicles) attempting to create a map of the environment. In this case, K can denote the total number of pedestrians, bicyclist's and other cars and $\theta_i \in \mathbb{R}^3$ for the ith target will consist of its location $\mathbf{x}_i = [x_i, y_i]^T$ and velocity (v_i) parameters, i.e.

$$\Theta = [K, \{x_i, y_i, vx_i, vy_i\}_{i=1}^K]^T \tag{2}$$

Yoav : Shouldn't K be estimated? **Piya :** Yes, K is now a parameter Mathematically the space-time

measurements collected at the mth sensing element can be described as

$$y_m(t) = \sum_{i=1}^{K} \phi(\mathbf{d}_m, \theta_i, t) + w_m(t), \quad 1 \le m \le M \tag{3}$$

where $w_m(t)$ is the additive noise. Here $\mathbf{d}_m \in \mathbb{R}^3$ denotes the location of the mth sensor and the function $\phi(.)$ characterizes the measurement model (often referred to as the point-spread function in the context of imaging) that depends on the physical laws governing wave propagation, and properties of the medium. Depending on the application and model assumptions, the function $\phi(.)$ can be linear, non-linear, and potentially, even non-convex. However, it can be *partially designed* by choice of senor locations $\mathbf{d}_m$. This will be a key enabler towards obtaining compressed sketches of measurements (or reducing the number of sensing units) while preserving the ability to reliably infer the parameter $\Theta(\tau)$.

The basic model assumes targets as point sources, but in many situations, they are distributed. **Piya :** Perhaps Peter can help characterize this model, since SONAR deals with such targets.
The main objective is to obtain estimates $\hat{\Theta}(\tau)$ of the parameter of interest $\Theta(\tau)$ using *minimal number of measurements/minimizing the number of sensing elements*. These estimates essentially are some appropriate functions of the spatio-temporal measurements $\mathbf{Y} = \{y_m(t), 1 \le m \le M, 1 \le t \le T\}$, i.e.,

$$\hat{\Theta}(T) = \mathbf{g}(\mathbf{Y}) \tag{4}$$

In passive localization, $\hat{\Theta}(T)$ is typically a function of the *cross correlation* between sensor measurements. In other words, the function $\mathbf{g}$ can be composed as $\mathbf{g} = \mathbf{g}_1 \circ \mathbf{g}_2$ where $\mathbf{g}_2(\mathbf{Y}) = \frac{1}{T}\mathbf{Y}\mathbf{Y}^H$. Hence, it is important to understand how the sensing geometry influences our estimate of the parameter via such quadratic correlation maps.

3.1 Correlation-Based Passive Localization and Geometry of Sensing:

In many scenarios, the parameters of interest can be reliably inferred from the *correlation of the measurements*. In other words, the correlation of the measurements act as a sufficient statistic for the parameters to be inferred. Depending on the application, the correlation matrix can be spatial (when the source signals are stationary), or spatio-temporal (when the temporal dynamics need to be tracked, such as for change-point detection). In fact, majority of passive sensing and localization techniques heavily rely upon computing cross correlation between sensor measurements to estimate location parameters. In these cases, we can effectively summarize the large amount of raw sensor measurements by only retaining and communicating their correlation. **Yoav :** The way I was thinking about it, each sensor has only one signal. In a one scenario, the quantity of interest is the "time delay of arrival" or the time shift of one signal relative to another that would maximize the correlation. Is there anything known about computing this time delay without communicating the whole time series? **Piya :** Since this is passive localization based on narrowband signals (or wideband signals decomposed into narrow frequency bins), the time delay of arrival translates to phase offset across the array. One sensor can localize two targets, but it takes more than two sensors to localize multiple targets
Spatial Correlation and Localization: Suppose we compute the empirical spatial correlation between $y_m(t)$ and $y_n(t)$ by averaging over T time samples (the signals are assumed to be stationary over this interval)[2]

$$\hat{\mathbf{R}}_{m,n} = \frac{1}{T}\sum_{t=1}^{T} y_m(t) y_n^*(t) \tag{5}$$

We can summarize the self and cross correlation between M time-series measurements (collected at M sensors) using these M^2 correlation values (collected in the form of a correlation matrix $\hat{\mathbf{R}}$). Owing to the geometry of the measurements, these correlation values directly depend on the sensor locations $\mathbf{d}_m$ (via the mapping $\phi(.)$). Two questions are of particular interest to our project:

[2]Reasonable to do so when the source signals are stationary and emit independent signals. This is the common practice in source localization using antenna arrays. We can also use more sophisticated regularized estimation of correlation.

1. Can we exploit the geometry of the measurement model to further compress the correlation matrix $\hat{\mathbf{R}}$? What is the role of sensor geometry in this case? We should still be able reliably infer Θ from such a compressed sketch.
2. How large should M be (in comparison to K) for multi-target localization ?

4 Localization of weak sources (Peter)

Yoav : Can the description of SCM be folded into Piya's introduction?

The focus here is detecting weak sources within a sensor network without a fusion center. To observe weak sources, as much information as possible should be used. Thus, at first there is no attempt to reduce the information in the data by sketching or special sensor arrangements. The network could consist of sensors with know location, partially unknown or unknown positions.

The propagation path from a given source location would here represent multiple propagation paths in a non-uniform media. The frequency domain transfer function from a source location to N receivers $\mathbf{a}$. Assuming K uncorrelated sources of complex amplitude $\mathbf{s}$ at spatial location $\mathbf{x}_k$, the received signal $\mathbf{y} \in \mathcal{R}^N$ on N receivers is

$$\mathbf{y} = \mathbf{A}\mathbf{s} + \mathbf{n}, \tag{6}$$

where $\mathbf{A} = [\mathbf{a}_1, \dots \mathbf{a}_K]$ and $\mathbf{n}$ is uncorrelated noise. The sources might be located in the near field and composed of many propagation paths. Examples of many propagation paths from a single source could be waves from 1) a source in a house propagating though the air and though the wall. 2) a cell phone signal with a direct path, a reflected path or refracted path. 3) a car radiating noise though the air and though the ground. Further, the sensors are not placed in a regular order, but where practical and maybe with unknown location. Thus the elements in $\mathbf{a}_k$ are unknown.

Yoav : What is the relationship between a_k and x_k?

To make observations of weak sources we observe L snapshots assuming stationarity $\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_L]$. We can here form the sample covariance matrix (SCM)

$$\mathbf{S} = \mathbf{Y}\mathbf{Y}^H/L \tag{7}$$

and form the the normalized SCM $\hat{\mathbf{S}}$ or coherence with elements

$$S_{ji} = \frac{S_{ji}}{\sqrt{S_{ii}S_{jj}}} \tag{8}$$

Yoav : I am confused about the definition of coherence, should it not be the maximum correlation when one signal to be shifted relative to the other?

Forming the ensemble mean over multiple snapshots give the cross spectral density matrix$\mathbf{C} \in \mathcal{R}^N \times N$

$$\mathbf{C} = \mathcal{E}[\mathbf{y}\mathbf{y}^H] = \mathbf{A}\mathbf{s}\mathbf{s}^H\mathbf{A}^H + \mathbf{N}, \tag{9}$$

The array signal processing literature is ample with processing of this type, especially with the structure of the $\mathbf{A}$ matrix partially known. In this work we will focus on pushing the computations to the sensor nodes and thus only observing part of SCM.

5 Passive Narrow-band Multi-Target Localization With Distributed Sensors (Piya)

Yoav : How is this problem different from the problem of finding the best placement for the sensors in order to maximize the accuracy of localization, ignoring communication bendwidth? **Piya :** Majority of existing sensor selection problems (based on submodular optimization, or solving subset selection problem subject to maximum tolerable error bounds) utilize a linear model, where the measurements are *linear functions* of the quantity of interest. Target localization is inherently non-linear since the information is contained in phase. Moreover, these algorithms do not give theoretical guarantee on the minimum number of required sensors.

With the aim of obtaining a compressive sketch of the correlation matrix (also termed as compressive covariance sensing), we will optimize the design of sensor array (i.e. choice of $\mathbf{d}_m, 1 \leq m \leq M$) by understanding

how the array geometry controls the algebraic structure of R_T. One of the main objectives will be to understand how much communication is needed (and between which subset of sensors) to achieve a certain level of accuracy. To illustrate this, we briefly discuss Co-PI Pal's recent work in structured sampler design (e.g., nested, coprime and generalized nested samplers) which utilize the idea of difference sets.

5.1 Background and Prior Work: Difference Set-Inspired Designs:

I will review some results in the context of array processing and DOA estimation...(to be filled in).

5.2 Proposed Research: Correlation-Aware Sensor Selection for Distributed Sensing

Motivated by these results, our goal will be to develop a rigorous framework for further developing the key idea of correlation-aware sensing to a distributed scenario and make it applicable for imaging problems beyond point target localization. The idea of difference set inspired sampler design can be actually generalized beyond that of antenna arrays, to acquire *compressive sketches* of the correlation between signals acquired between pairs of sensors. In general, given N sensors, it is natural to think that one needs to compute the correlation between all $\binom{N}{2}$ time series (from all possible sensor-pairs) to construct the overall $N \times N$ correlation matrix $\mathbf{R}$ [3]. However, using the idea of difference-set sampling, one can only compute cross-correlation values between a much smaller subset of size $\sim \sqrt{N}$ of *suitably selected sensor-pairs* and recreate the entire $N \times N$ correlation matrix $\mathbf{R}$. In the context of distributed sensing, this automatically means that only these sensors need to communicate and exchange information.

Key Idea: Exploiting Distance-based Redundancies Using Generalized Nested Samplers: The main idea behind achieving such reduction is to exploit the redundancies present in the correlation values that naturally result from the physical spatial signal model. A widely used example of such a redundancy is that the correlation $\mathbf{R}_{m,n} = E\Big(y_m(t)y_n^*(t)\Big)$ between mth and nth sensors is of the following form

$$\mathbf{R}_{m,n} \approx f(\mathbf{d}_m - \mathbf{d_n}) \tag{10}$$

In other words, the correlation is spatially only a function of the *inter-sensor displacement* $\mathbf{d}_m - \mathbf{d_n}$, and this is a direct consequence of the functional form of $\phi(.)$ **Piya :** Can give specific examples if needed. This is also referred to as spatial stationarity and it is (exactly or approximately) true for many applications as narrowband and wideband radar [4], super-resolution optical imaging [], mmWave wireless channels [] and so forth. Hence, depending on the inter-sensor distances, many of these $\binom{N}{2}$ correlation values are actually repeated/redundant. Based on this observation, we propose to use a new sketching technique developed by co-PI Pal, called **Generalized Nested Sampling (GNS) to reduce the amount of inter-sensor communication**. Suppose the sensors are located on a uniform grid. In one dimension, (10) implies that the ideal correlation matrix $\mathbf{R}$ has Toeplitz structure and GNS provides an optimal way to select sensors to sketch such a matrix. Given any integer $L \geq 6$, GNS is defined in terms of two integer valued parameters $\Theta(N)$ and $\Gamma(N)$ given by

$$\Theta(N) = \lfloor \sqrt{N + \frac{1}{4}} - \frac{1}{2} \rfloor \quad \Gamma(N) = 1 + L - \Theta^2(N) \tag{11}$$

Given $\Theta(N)$ and $\Gamma(N)$, a GNS can be defined as the following measurement/sensor-selection matrix

Definition 1 *For any integer $N \geq 6$, a Generalized Nested Sampling matrix $\mathbf{A}_{GNS} \in \mathbb{R}^{M \times N}$, with $M = \Gamma(N) + \Theta(N) - 1$ is given by*

$$[\mathbf{A}_{GNS}]_{i,j} = \begin{cases} 1 & \text{if } j = i,\ 1 \leq i \leq \Gamma(N) \\ 1 & \text{if } j = (i - \Gamma(N))\Theta(N) + i, \quad \Gamma(N) < i \leq M \\ 0 & \text{else} \end{cases} \tag{12}$$

[3] $\mathbf{R}$ represents the ideal correlation matrix, whereas $\hat{\mathbf{R}}$ represents an estimate of $\mathbf{R}$ computed with finite data

[4] In the latter case, this holds at individual frequency bands after splitting the wideband signal into narrow frequency bins using a filter bank

It can be noted that $\mathbf{A}_{\text{GNS}}$ is essentially a *row-selection* matrix that selects M out of N elements of a vector. The indices of these M rows crucially tell us *which sensors to select* so that we can obtain a lossless sketch of $\mathbf{R}$ by computing the cross-correlation between these M sensors. Such selection is governed by the need to exploit distance-based redundancies as captured in (10). As discussed earlier, when the sensors are located on a one uniform dimensional grid, (10) essentially implies that the full correlation matrix $\mathbf{R}$ is a Toeplitz matrix. Hence, GNS dictates how to select a subset $\mathcal{S}_{\text{GNS}}$ of $M == \Theta(\sqrt{N})$ sensors out of N available sensors so that the Toeplitz $\mathbf{R}$ can be *exactly reconstructed* from the pair-wise correlation between sensors in $\mathcal{S}_{\text{GNS}}$. Let $\mathbf{R}_{\mathcal{S}_{\text{GNS}}} \in \mathbb{C}^{M\times M}$ be the correlation matrix computed by aggregating the signals from these sensors. Then, we have

$$\mathbf{R}_{\mathcal{S}_{\text{GNS}}} = \mathbf{A}_{\text{GNS}}\mathbf{R}\mathbf{A}_{\text{GNS}}^T \tag{13}$$

The structure of $\mathbf{A}_{\text{GNS}}$ ensures that $\mathbf{R}_{\mathcal{S}_{\text{GNS}}}$ is a *lossless* sketch of the high-dimensional correlation matrix $\mathbf{R}$. In turn, this tells us that ideally, it is sufficient for only these M sensors from $\mathcal{S}_{\text{GNS}}$ to communicate and exchange their measurements in order to preserve the correlation information from all $\binom{N}{2}$ sensor pairs. Since $M = \Theta(\sqrt{N})$, we can significantly save the cost of communication in a distributed sensor network by such correlation-aware sensor selection. Given the basic idea of GNS, we will now focus on the following specific tasks to further extend the capabilities of GNS

1. **Two Dimensional GNS and Hierarchical Sensor Selection:** Most sensor networks of interest will be distributed over a two-dimensional area. Hence, it is important to extend the basic version of GNS (which was developed for sensors on a line) to two dimensions. In such cases, the geometry of sensor configurations have very interesting implications on the redundancy relation (10) and the structure of $\mathbf{R}$. As an example, if we assume the sensors to be located on a uniform rectangular grid in two dimensions of size $N_x \times N_y$, then (10) implies that $\mathbf{R} \in \mathbb{C}^{N_xN_y\times N_xN_y}$ is a *two-level* (or block) Toeplitz matrix of the form

$$\mathbf{R} = \begin{bmatrix} \mathbf{T}_0 & \mathbf{T}_1 & \cdots & \mathbf{T}_{(N_y-1)} \\ \mathbf{T}_{-1} & \mathbf{T}_0 & \cdots & \mathbf{T}_{(N_y-2)} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{T}_{-(N_y-1)} & \mathbf{T}_{-(N_y-2)} & \cdots & \mathbf{T}_0 \end{bmatrix} \tag{14}$$

   where $\mathbf{T}_i \in \mathbb{C}^{N_x\times N_x}$ is a Toeplitz matrix for each $i \in [0, N_y - 1]$.
   We will develop two-dimensional versions of Generalized Nested Sampling (2D-GNS) to guide our sensor selection strategy so that we can produce a lossless sketch of $\mathbf{R}$ with far fewer sensors by exploiting its unique algebraic structure captured in (14). Notice that the extension is quite non-trivial given that $\mathbf{R}$ has Toeplitz structure in individual blocks, as well as the blocks are repeated along each (sub)-diagonal. The optimal sketching matrix should exploit the redundancies in both levels to produce a sketch of size $\backsim \sqrt{N_xN_y}$. We will draw inspirations from our previous works in designing two-dimensional nested arrays to develop the corresponding sketching 2D GNS operator $\mathbf{A}_{\text{2D-GNS}} \in \mathbb{R}^M \times N_xN_y$, where $M = O(\sqrt{N_xN_y})$. In particular, we propose to select $M = N_d + N_s$ sensors from a total of N_xN_y sensors as follows. We choose a nonsingular *integer* matrix $\mathbf{P} \in \mathbb{Z}^{2\times 2}$ such that $\det(\mathbf{P}) = N_d$. The N_s sensors belong to the following subset of the uniform rectangular grid

$$\mathcal{S}_s = \{\mathbf{P}[n_1, n_2]^T, 0 \le n_1 \le N_1 - 1, 0 \le n_2 \le N_2 - 1\}, \quad N_1N_2 = N_s$$

   and the N_d sensors are located at

$$\mathcal{S}_d = \{\mathbf{v} \in \text{FPD}(\mathbf{P}), \mathbf{v} \text{ is integer valued}\}$$

   Here FPD($\mathbf{P}$) (aka Fundamental Parallelopiped of $\mathbf{P}$) is the set of all vectors of the form $\{\mathbf{Px}, \mathbf{x} \in [0, 1)^2\}$. It can be verified that indeed the cardinality of $\mathcal{S}_d = \det(\mathbf{P}) = N_d$. We further require $N_sN_d = N_xN_y$. Hence GNS selects a subset of sensors $\mathcal{S}_{\text{2D GNS}} = \mathcal{S}_s \cup \mathcal{S}_d$ which consists of a union of two subsets of the original lattice: (i) $\mathcal{S}_d$ consisting of N_d densely spaced sensors on a rectangular grid, and (ii) $\mathcal{S}_s$ consisting of $N_s = N_1N_2$ sensors on a larger (sparser) lattice generated by the integer matrix $\mathbf{P}$ (which is also a

sub-lattice of the desired uniform rectangular grid). The vector difference set $\{\mathbf{d}_m - \mathbf{d}_n, \mathbf{d}_m, \mathbf{d}_n \in \mathcal{S}_{\text{2D GNS}}\}$ consists of all $N_x N_y$ consecutive integer points on the 2D rectangular lattice. Using this property of 2D Nested arrays, we will develop the corresponding sketching matrix $\mathbf{A}_{2DGNS}$ and develop an effective way to sketch the two-level Toeplitz structured $\mathbf{R}$ using only $M = O(\sqrt{N_x N_y})$ communicating sensors.

An interesting aspect of our design is that it will use a combination of closely spaced sensors (given by $\mathcal{S}_d$) and far-apart sensors (given by $\mathcal{S}_s$). This will naturally have two advantages. Firstly, it is well known that for passive sensing, sensor pairs that are further away automatically provide larger time delays of arrival between signals from two different sources, thereby resulting in higher accuracy of detection (and better spatial resolution). However, if the source is farther away from each sensing unit, the longer path loss can result in degraded signal-to-noise ratio (SNR). The proposed 2D GNS sampler provides a balance between these two aspects by using a combination of both close and far sensors. Such an architecture will lead to development of efficient algorithms that will utilize the higher spatial resolution offered by the array of distant sensors and counterbalance the effect of low SNR and ambiguity by utilizing the structure of the smaller and denser array.

A second interesting aspect of 2D GNS is that it lends itself to a hierarchical selection of communicating sensors, depending on how many targets to detect, and how large the field of view is. For example, we can divide a wide field of view into smaller segments. Depending on which segment captures the target of interest [5], we can zoom into that part and determine the subset of communicating sensors using the 2D GNS-based subset selection rule. We can implement such subset selection by dynamically turning sensors on and off. Alternatively, in a multi-target environment where the targets are sufficiently far apart, we can dedicate different segments to localize different targets and dynamically select the communicating sensors from each segment. Finally, it is to be noted that the orientation of the communicating sensors on the lattice is determined by the integer matrix $\mathbf{P}$ and such orientation can affect target localization performance (especially the source angles). When we have partial knowledge of target location, it will also be of interest to design $\mathbf{P}$ to orient the subset so as to maximize the target detection probability. For multi-target scenario, we can design different $\mathbf{P}$ matrices (if needed) for each of the aforementioned segments.

2. **Finite Sample Performance:**
3. **Wideband Signal and Spatio-temporal GNS:**

Piya : These tasks can be further integrated with the binary embedding based sketching ideas proposed by Rayan and Alex. **Rayan :** Agree!

5.3 Localizing weak sources with graph signal processing

Yoav : I think this section can be combined with Piya's sections. Choosing which pairs should communicate is clearly related to their geometric layout. **Peter: That is fine.**

A graph signal processing approach was used in Ref [56] for a 5000 element seismic array by processing the whole normalized SCM $\hat{\mathbf{S}}$ at once, i.e., using a fusion center. When the SCM $\hat{\mathbf{S}}$ is above a certain threshold at element ij it is likely that a signal is observed and has propagated between nodes i and j, essentially forming an edge between nodes i and j in a graph. When a sufficient set of connected edges are detected in a region of the network a source is likely in that region. Part of the extracted SCM $\hat{\mathbf{S}}$ can then be used to localize the source more precisely.

To extract very weak signals with a well estimated and robust SCM is needed. Thus we pass the full time series between local nodes i and j, not the whole array and develop robust signal processing methods[68]. This will represent a lot of communication demand and thus we will only pass signal between neighboring stations. Once a graph edge is formed it could either be communicated further to a wider set of nodes.

Robust signal processing methods[68] would entail making the processing insensitive to outliers. Qualitative robustness can be investigated via the influence function. A qualitatively robust estimator is characterized by an IF that is continuous and bounded. Continuity implies that small changes in the observed sample cause only small changes in the estimate. The boundedness implies that a small amount of contamination

[5] There can be several criterion for determining this. A simple option would be to divide the sectors based on received signal strength

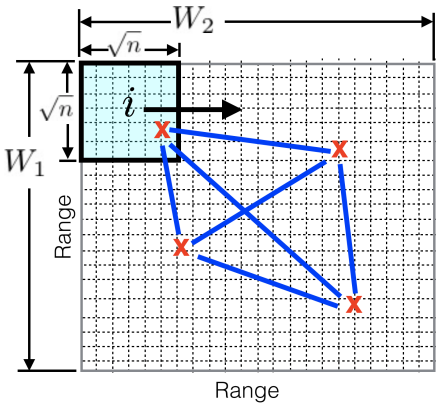

Figure 2: 2D velocity image corresponds to map divided into pixels (dashed boxes). The square image patch i contains n pixels. W_1 and W_2 are the vertical and horizontal dimensions of the image (in pixels), which give I unique patches. Sensors are shown as red x's and the ray paths between the sensors are shown as blue lines.

cannot lead to an unbounded error the estimate.
We propose to work on
(1) A scheme for processing locally at each node.
(2) Estimate of the SCM should be robust to outliers[68]. Thus we will investigate the processing for empirical influence functions. Kernelized methods could likely increase the robustness, see Section 6.2. **Alex :** Is the sentence true?
(3) Extract relevant physical signals that can be used in the extraction of environmental parameters **E**.

5.4 Tomography

Yoav : I believe you are talking here about tomography, or reconstructing the environment. Can you write a paragraph of introduction, what is the problem? What is the desired solution? **Rayan :** I might be able to throw in some text about learning fast dictionaries, i.e., dictionaries learned from the data, but that also admit fast transforms like FFTs. Peter, Yoav, what do you think? **Peter: Rayan, if it binds the proposal better together it would be excellent. We can always work on it anyway.**
After having observed and localized the sources we are here concerned with extracting the the environment **E**. The environment is needed for any physical modeling and understanding of how the machine learning approaches work in practice. The models developed here could be aided by some know information about the environment **E**, say the rooms in a house, or probability distributions of sound speed in the ocean or atmosphere.
We will have information about the travel-time and attenuation between a source and sensors. Or travel-time and attenuation between sensors could be extracted by just observing noise[62, 60, 25]. In the initial setup, source and receivers locations are assumed known.
We have worked on this problem for seismic modeling to develop separately two velocity models[4], deemed the *global* and *local* models, and briefly discuss dictionaries for sparse modeling. The global model considers the larger scale or global features and relates travel times to slowness. The local model considers smaller scale or more localized features with sparse modeling and is considered to consist of sparse linear combinations of atoms from a dictionary We propose to use dictionary learning during the inversion to adapt dictionaries to specific slowness maps.
The global model has ben the basis of tomography and is repeated here for reference, slowness pixels (see Fig. 2(a)) are represented by the vector $\mathbf{s}' = \mathbf{s}_g + \mathbf{s}_0 \in \mathbb{R}^N$, where $\mathbf{s}_0$ is reference slownesses and $\mathbf{s}_g$ is perturbations from the reference, here referred to as the *global slowness*, with $N = W_1 W_2$. Similarly, the travel times of the M rays are given as $\mathbf{t}' = \mathbf{t} + \mathbf{t}_0$, where $\mathbf{t}$ is the travel time perturbation and $\mathbf{t}_0$ is the reference travel time. The tomography matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ gives the discrete path lengths of M straight-rays

through N pixels (see Fig. 2(a)). Thus $\mathbf{t}$ and $\mathbf{s}_{\mathrm{g}}$ are related by the linear measurement model

$$\mathbf{t} = \mathbf{A}\mathbf{s}_{\mathrm{g}} + \epsilon, \tag{15}$$

where $\epsilon \in \mathbb{R}^M$ is Gaussian noise $\mathcal{N}(\mathbf{0}, \sigma_\epsilon^2 \mathbf{I})$, with mean $\mathbf{0}$ and covariance $\sigma_\epsilon^2 \mathbf{I}$. We estimate the perturbations, with $\mathbf{s}_0$ and $\mathbf{t}_0 = \mathbf{A}\mathbf{s}_0$ known. We call (15) the *global model*, as it captures the large-scale features that span the discrete map and generates $\mathbf{t}$.

Extracting of the local features is obtained via sparse modeling and dictionary learning. Sparse modeling assumes that signals can be reconstructed using a few (sparse) vectors, called atoms, from a potentially large set of atoms, called a dictionary. Recent ocean acoustics works utilizing sparse modeling is beamforming[67], matched field processing [23], and geoacoustic inversion [24]. One challenge in sparse modeling is finding the best dictionary for sparsely representing specific signals. Such dictionaries can be composed of wavelets, or the discrete cosine transform (DCT). These predefined dictionaries perform well for many signals. However, using a form of unsupervised machine learning, called dictionary learning, optimal dictionaries can be learned directly from specific data[45]. It has been shown that learned dictionaries outperform generic dictionaries when sufficient signal examples are available. Machine learning, and specifically dictionary learning, have recently obtained compelling results in ocean acoustics [3] and seismology[39].

Whereas many machine learning techniques in geoscience[39], are reliant on large amounts of training data, LST[**?**] requires none. In LST we adopt the adaptive dictionary learning paradigm from image denoising [19] and medical imaging[55], in which dictionaries are learned directly from patches of the corrupted image, obtained in (15). In LST, slowness dictionaries are learned from patches of a least squares regularized inversion, and are then used to reconstruct a sparsity-constrained slowness image. Assuming sufficiently dense ray sampling, the dictionary is initially unknown and is learned in parallel with the inversion. LST[**?**] obtains high resolution by assuming that small patches of discrete slowness maps are repetitions of few elemental patterns from a dictionary of patterns, that is dictionary learning.

Assuming that the travel paths between sensors has been estimated[60, 25] We propose the future development of machine learning-based tomography methods in acoustics. Such methods will help to more fully-exploit both existing environmental data $\mathbf{E}$, as well as very dense sampling from future arrays with many sensors. Such large scale, mobile, and deformable arrays, will use ambient noise processing [60], to obtain very dense and rich data sets. We propose to:

(1) further develop a dictionary learning-based travel time tomography [**?**], accounting for uncertainty in the measurements and physics;

(2) formulate the dictionary learning-based approach as CNN via CSC;

(3) apply this CSC tomography framework to data assimilation, to obtain higher-resolution estimates of the environment $\mathbf{E}$.

We further propose to develop (4) For EM signals, where the wave speed does not vary much the above formulation could be modified to do attenuation tomography. (5) Acoustic event detection methods that leverage recent advances in machine learning. (6) For unknown sensor location, only a graph of the sensor response is obtained. The edges of the graph will contain information related to travel time between nodes. It could here be useful to extract the information onto a manifold. From there further information could be extracted in the spatial dimension. A well-known example of this is that the location of cities in Switzerland could be constructed from the train schedule[17]. With more sensors we would be able to extract both location and environmental information.

6 Quantized Sketches of Complex Signals

Consider a monitoring application as described in Sec. 2.1. Here a network of sensors is used to monitor an area or a system. As an example, consider a network of cameras that are used to monitor a highway. The goal of the camera network is to provide a real-time summary of the status of the highway and detect critical events such as accidents, obstacles on the highway, or speeding cars. One approach is to transmit all video streams to a central location and process them there. This approach is expensive both in terms of the required communication bandwidth and the computational resources needed in the central location.

A common practical approach is to compress the videos before sending them and decompress them in a central location. This can provide a 10-100 fold improvement in terms of communication bandwidth but increases the computational load on the center and sensors. Lossy compression solves a reconstruction problem: how to encode a video stream to allow reconstruction so the reconstructed stream is visually indistinguishable from the original. In contrast, the information we must retain in the monitoring task is more specific and opens the possibility that the encoding can be much smaller than the one for lossy compression.
Specifically, we consider the task of detecting the difference between two distributions. The difference can be a result of spatial differences – two sensors that are at different locations, or it can be temporal – two samples drawn from the same sensor at different times. Our approach is based on quantized sketches. It is a main goal of this section to develop both algorithmic tools and rigorous theoretical analysis for this framework, as well as examine the types of statistical questions that can be addressed. To that end, we will first provide some relevant background, both on kernel and compressive statistics [26, 27, 6] as well as quantization and binary embeddings [35, 59, 58, 30].

6.1 Background and Prior Work: Quantization and Binary Embeddings

We will rely on extremely coarse, e.g., binary, quantization of the data-sketches. These methods minimize storage and computation costs (see e.g., [20, 5]) and have the added benefit of being appealing in hardware implementation particularly if they are computationally inexpensive and thus promote speed in hardware devices [35, 40]. A growing body of work, which co-PI Saab has contributed significantly to (e.g., [59, 38, 58, 57, 31, 16]), studies signal reconstruction from coarsely quantized measurements. *One important theme that emerges from this line of work is that if one collects more coarsely quantized (even 1-bit) measurements than a critical minimal number, and uses sophisticated quantization schemes, then the extra measurements can be efficiently used in quantization-aware algorithms to rapidly drive the reconstruction error down as a function of the number of measurements.* This theme has held true in a wide range of signal and measurement contexts, including band-limited functions [16], finite frame expansions [33], and compressed sensing of approximately sparse vectors [59, 58], low-rank matrices [57], and manifold valued signals [31].
Saab has recently extended this observation beyond signal reconstruction, to the context of (Euclidean) distance-preserving binary embeddings [30]. Here the goal is to map points in $\mathbb{R}^n$ to the binary cube $\{\pm 1\}^m$, where $m \ll n$, in such a way that distances in $\mathbb{R}^n$ can be well-approximated by appropriate functions on $\{\pm 1\}^m \times \{\pm 1\}^m$. We now briefly describe this contribution as it is pertinent to our ensuing discussion. In [30], $A : \mathcal{T} \subset \mathbb{R}^n \to \mathbb{R}^m$ is a *random* Johnson-Lindenstrauss [36] (i.e., distance preserving linear) map and $\mathcal{T} \subset \mathbb{R}^n$ is a set of finite or infinite cardinality. $Q : \mathbb{R}^m \to \{\pm 1\}^m$ is a stable noise-shaping quantizer (e.g., a $\Sigma\Delta$ [15], or β [7] quantizer) which act sequentially on the measurements $\mathbf{y}_i$, and yields

$$\mathbf{q} = \mathbf{y} - H\mathbf{u} \quad \text{with} \quad (\|\mathbf{y}\|_\infty \le c_1 \implies \|\mathbf{u}\|_\infty \le c_2.)$$

Here H is a lower-triangular matrix associated with the scheme and $\mathbf{u}$ is a vector of state-variables $(u_1, ..., u_m)$, initialized via $u_i = 0, i \le 0$ and updated sequentially so that u_j is a function of q_i, y_i, u_i, for a subset of indices $i < j$. Quantizers of interest include stable rth-order (with $r \ge 1$) $\Sigma\Delta$ schemes [15] where $\mathbf{q} = \mathbf{y} - D^r\mathbf{u}$, and distributed-$\beta$ encoding schemes [7] where H is block diagonal with identical lower-triangular blocks G, given by $G_{i,i} = 1, G_{i+1,i} = -\beta$, and $G_{i,j} = 0$ otherwise. With these quantization schemes playing a prominent role, in [30] co-PI Saab constructed approximately isometric (i.e., distance preserving) embeddings between the metric space $(\mathcal{T}, \|\cdot\|_2)$ and the binary cube $\{-1, +1\}^m$ endowed with the pseudo-metric $d_V(\tilde{\mathbf{q}}, \mathbf{q}) := \|V(\tilde{\mathbf{q}} - \mathbf{q})\|_2$ where V is a carefully constructed matrix. For a matrix $A \in \mathbb{R}^{m\times n}$, and a noise-shaping quantizer Q, as above, the algorithm for computing these embeddings is simply given by

$$g : \mathcal{T} \to \{\pm 1\}^m \quad \text{where} \quad x \mapsto \mathbf{q} = Q(A\mathbf{x}).$$

In particular, when A is a fast Johnson-Lindenstrauss matrix (e.g., [1]), *the constructed embeddings support fast computation* and despite their highly quantized non-linear nature, they perform as well as linear Johnson-

Lindenstrauss methods! Indeed, when $\mathcal{T}$ is finite [30] shows that with high probability

$$m \gtrsim \frac{\log(|\mathcal{T}|)\log^4 n}{\alpha^2} \quad \Longrightarrow \quad \left|d_{\widetilde{V}}(g(\mathbf{x}), g(\tilde{\mathbf{x}})) - \|\mathbf{x}-\tilde{\mathbf{x}}\|_2\right| \leq \alpha\|\mathbf{x}-\tilde{\mathbf{x}}\|_2 + c\eta(m), \quad \text{where} \quad \eta(m) \xrightarrow[m\to\infty]{} 0.$$

Above, $\eta(m)$ decays polynomially fast in m (when $\mathcal{Q}$ is a $\Sigma\Delta$ quantizer), or exponentially fast (when $\mathcal{Q}$ is a distributed noise shaping quantizer). Additionally, when $\mathcal{T}$ is arbitrary (with possibly infinite cardinality, e.g., a compact manifold) [30] show that with high probability and for prescribed distortion α

$$m \gtrsim \frac{\log^4 n}{\alpha^4} \cdot \frac{\omega(\mathcal{T})^2}{\mathcal{R}(\mathcal{T})^2} \quad \Longrightarrow \quad \left|d_{\widetilde{V}}(g(\mathbf{x}), g(\tilde{\mathbf{x}})) - \|\mathbf{x}-\tilde{\mathbf{x}}\|_2\right| \leq \alpha\mathcal{R}(\mathcal{T}) + c\eta(m)$$

where $\eta(m)$ is as before and $\mathcal{R}(\mathcal{T})$ and $\omega(\mathcal{T})$ denote the Euclidean radius of $\mathcal{T}$ and its Gaussian width (which roughly scales with the average radius of $\mathcal{T}$, so intrinsically low-dimensional sets have a small width). *In short, with* very few measurements *compared to the ambient dimension of the signals, one can very efficiently (roughly at the cost of a fast Fourier transform) obtain low-dimensional binary sketches of the data. These sketches approximately preserve all pairwise distances in the original set and the distances in the embedded space can be computed efficiently.* Signal reconstruction is not a focus of this section, but the promising results obtained in that context, and in the context of binary embeddings, lead us to believe that the above techniques can be generalized to other tasks.

6.2 Background and Prior Work: Kernel Statistics

Statistical distances, or accurately measuring distances between distributions, arise in a large number of applications. For sensors, these are important quantities for monitoring and tracking. An example of this would be for acoustic sensors, where each sensor collects the local power spectral density of a signal. Each sensor i now has a set of high dimensional data $\mathbf{Y_i} = \{\mathbf{y}_i(t)\}_{t=t_0}^{t_n} \subset \mathbb{R}^d$, and the question is whether $\mathbf{Y_1}$ and $\mathbf{Y_2}$ are distributionally the same, up to a time shift. After constructing a kernel $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}_+$ to define similarity between any two points, and ignoring communication constraints, a simple statistic to construct for a point $z \in \mathbf{Y_i}$ is a difference of kernel means over the two data sets $\frac{1}{n}\sum_{x\in\mathbf{Y_1}} K(z,x) - \frac{1}{n}\sum_{x\in\mathbf{Y_2}} K(z,x)$. If $\mathbf{Y_1}$ and $\mathbf{Y_2}$ came from the same distribution, then this statistic would be unbiased at $z \in \mathbf{Y_i}$, otherwise there would be a bias for particular z. Thus, computing the mean square error over all $z \in \mathbf{Y_1} \cup \mathbf{Y_2}$ yields a statistic that would be close to 0 if $\mathbf{Y_1}, \mathbf{Y_2} \sim p$, and would be biased if $\mathbf{Y_1} \sim p$ and $\mathbf{Y_2} \sim q$ for $p \neq q$. This is a well studied statistic known as kernel Maximum Mean Discrepancy [26]. In what follows, we will describe the mathematical framework and guarantees established, and propose methods for dealing with communication and computation constraints in this framework through randomized sketching and binary embeddings, as well as how to establish guarantees of convergence under non-i.i.d. sampling situations such as time series data. We will also detail a larger set of sensor problems that can be addressed in this framework.

Generally, the approach described above measures the distance between the distributions' *mean embeddings* [49]. A mean embedding of a distribution $\mu_p : \mathbb{R}^d \to \mathbb{R}_+$ of a probability distribution p is computed as

$$\mu_p(z) := \mathbb{E}_{x\sim p}\big[K(z,x)\big].$$

Effectively, the mean embedding $\mu_p \in \mathcal{H}$ indexed by p is a unique point in the Reproducing Kernel Hilbert Space $\mathcal{H}$ that is induced by the kernel K.

When K is a *universal* kernel (e.g. Gaussian) [47], then the mean embedding satisfies a key property that $\|\mu_p - \mu_q\|_{\mathcal{H}}$ is bi-Lipschitz with respect to $\|p-q\|_{L^\infty}$ for absolutely continuous distributions p and q. This effectively means that the mean embedding transform maintains the same information as working in $\mathbb{R}^d$, with the benefit that mean embeddings also satisfy nice statistical convergence properties. One key property is that, if we are only given n finite samples $\mathbf{Y_1} \sim p$ and $\mathbf{Y_2} \sim q$ to compute the empirical mean embeddings $\widehat{\mu}_X$ and $\widehat{\mu}_Y$, and we compute the mean embedding at all $z \in \mathbf{Y_1} \cup \mathbf{Y_2}$, then $\|\widehat{\mu}_{\mathbf{Y_1}} - \widehat{\mu}_{\mathbf{Y_2}}\|_2 \to \|\mu_p - \mu_q\|_{\mathcal{H}}$ at a rate $O\left(\frac{1}{\sqrt{n}}\right)$. A statistical interpretation of the mean embedding distance is that it computes a difference of means test on the eigenfunctions of K rather than in the original space $\mathbb{R}^d$. This means that two distributions having matching means in the eigenfunction space is equivalent to the distributions having

all moments matching in $\mathbb{R}^d$. A large benefit of the mean embeddings is that this calculation can be done without explicitly computing the eigendecomposition of K.
However, statistics of this type suffer from a number of issues under computation and communication bottlenecks, as they require storing and communicating all points in $\mathbf{Y_1} \cup \mathbf{Y_2}$. In particular, computing $\hat{\mu}(z)$ at any one point z requires evaluating the kernel at all points in $\mathbf{Y_1} \cup \mathbf{Y_2}$, which can be prohibitively expensive. This has motivated the computational speed up presented by co-PI Cloninger in [6] via undersampling $z \in S \subset \mathbf{Y_1} \cup \mathbf{Y_2}$ under the condition that a kernel matrix K can be decomposed as $K \approx RR^T$ for R that can be efficiently accessed. There is a similar vein of computation speed up accomplished through kernel compression via randomized sketching [27]. However, even these approaches still require communication of a number of points between sensors, or passing double precision complex valued summary statistics.

6.3 Framework

Our goal is to present a complete, theoretically rigorous framework for performing various statistical, signal processing, and learning tasks from highly quantized data representations. We focus on the case of 1-bit representations as a theoretical extreme case, but emphasize that our methods should apply to more finely quantized data. We will propose, and analyze (a) algorithms for producing quantized sketches of data as well as (b) associated algorithms for performing the aforementioned tasks. Our strategy will be to develop these algorithms in tandem; that is, we will propose task-based quantization algorithms and quantization-aware algorithms for performing the tasks. We will strive for methods that support fast computation, and that lend themselves to distributed computing on, e.g., a sensor network.
Suppose signals of interest are represented as N vectors $\mathbf{x}[j] \in \mathbf{X} \subset \mathbb{R}^d$ for $j \in \{1, ..., N\}$, where $\mathbf{X}$ could be a finite set, i.e., a point cloud, or an infinite set (e.g., a compact manifold, or the set of all sparse vectors). If the data is collected as a time series, then $\mathbf{x}[j] = \mathbf{x}(t_j)$. Further, assume that the measurement operator, accounting for all the digitization and measurements at all the sensors, is given by $\mathbf{\Phi} : \mathbb{R}^d \to \mathbb{R}^m$. In the case of distributed sensing systems, we can assume each sensor collects/computes a portion, $\mathbf{\Phi}_i(\mathbf{X}) \subset \mathbb{R}^{m_i}$, of $\mathbf{\Phi}(\mathbf{X}) \subset \mathbb{R}^m$ (so that $\sum_i m_i = m$). Henceforth, the map at sensor i, $\widetilde{\mathbf{\Phi}}_i$, will be a slight modification of the transfer function Φ_i presented in (1); we include a randomized embedding of the digitized signal at each sensor. Here $\widetilde{\mathbf{\Phi}}_i = \mathcal{A}\Phi_i$, where $\mathcal{A}$ is a randomized linear (or nonlinear) Johnson-Lindenstrauss map. The corresponding signal at sensor i, $\mathbf{Y}_i = \widetilde{\mathbf{\Phi}}_i(\mathbf{X})$ can then be quantized via $Q(\widetilde{\mathbf{\Phi}}_i(\mathbf{X}))$ with a quantization map $Q : \mathbb{R}^{m_i} \to \{\pm 1\}^{m_i}$ that maps the measurements to bits. A function f, to be designed, is then applied to the resulting bits, to further compress them while simultaneously retaining enough information to perform a (say, statistical) task of interest. The resulting map of all sensors can be represented as

$$x \mapsto Q(\widetilde{\mathbf{\Phi}}(\mathbf{x})) = \begin{bmatrix} Q(\widetilde{\mathbf{\Phi}}_1(\mathbf{x})) \\ \vdots \\ Q(\widetilde{\mathbf{\Phi}}_k(\mathbf{x})) \end{bmatrix} \mapsto f(Q(\widetilde{\mathbf{\Phi}}(\mathbf{x}))) =: g(\mathbf{x}). \tag{16}$$

6.4 Proposed Work: Binary Embeddings of Nonlinear Similarity Measures

The type of information that can be derived from $f(Q(\widetilde{\mathbf{\Phi}}(\mathbf{x})))$ depends on the choice of measurement operator $\widetilde{\mathbf{\Phi}}_i$ at each sensor, as well as the quantization scheme. In [30], Co-PI Saab constructed f, $\widetilde{\mathbf{\Phi}}$, and Q that all admit fast computation, so that $\|g(\mathbf{x}) - g(\mathbf{x}')\|_2 \approx \|\mathbf{x} - \mathbf{x}'\|_2$ (see Sec. 6.1 for the details). For several of the statistical tasks below, we require construction of the quantization scheme to approximate more complex relationships between $\mathbf{x}, \mathbf{x}'$. We propose developing a new set of quantization schemes that incorporate nonlinear functions of the data $\mathbf{x}$. Using ideas from noise-shaping quantization (Sec. 6.1) we will design f, $\widetilde{\mathbf{\Phi}}$, and Q so that $\langle g(\mathbf{x}), g(\mathbf{x}')\rangle = \langle f(Q(\widetilde{\mathbf{\Phi}}(\mathbf{x}))), f(Q(\widetilde{\mathbf{\Phi}}(\mathbf{x}')))\rangle$ approximates an arbitrary kernel, for example the Gaussian kernel $\langle g(\mathbf{x}), g(\mathbf{x}')\rangle \approx e^{-\|\mathbf{x}-\mathbf{x}'\|_2^2/\sigma^2}$. First, we will show that this can be achieved by appropriately quantizing $cos(2\pi\langle w, \cdot\rangle + b)$ for appropriate random choices of w, b [54]. To implement this in practice, each physical sensor could collect one or more linear measurements $\langle \omega, \cdot\rangle$, and then the non-linearity (e.g., cosine) could be implemented as part of the quantizer. Alternatively, the sensor could measure the non-linear function directly prior to quantization.
To give some context, in the non-quantized setting, random Fourier Features (RFFs) were introduced in

[54] to approximate shift-invariant kernels (e.g., $k(\mathbf{x},\mathbf{x}') = \exp(-\|\mathbf{x}-\mathbf{x}'\|_2^2/2)$. For $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^d$ [54] set $\widetilde{\mathbf{\Phi}}(\mathbf{x}) = \sqrt{\frac{2}{m}}\cos(A\mathbf{x}+b)$, where A is a Gaussian matrix and b is a uniform random vector over $[0, 2\pi]$. To show $\langle \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{x}')\rangle \approx k(\mathbf{x},\mathbf{x}')$, they showed (using an ϵ-net argument) that

$$\sup_{\mathbf{x},\mathbf{x}'\in\mathcal{X}} |\langle \mathbf{\Phi}(\mathbf{x}), \mathbf{\Phi}(\mathbf{x}')\rangle - k(\mathbf{x},\mathbf{x}')| = \sup_{\mathbf{x},\mathbf{x}'\in\mathcal{X}} |\frac{2}{m}\sum \cos(\langle a_i,\mathbf{x}\rangle + b_i)\cos(\langle a_i,\mathbf{x}'\rangle + b_i) - k(\mathbf{x},\mathbf{x}')|$$

concentrated with high probability around zero [44]. Their result was improved in [63] using Dudley's inequality [44]. The above results' dependence of m on $\mathcal{X}$'s geometric properties (e.g., its intrinsic dimension) was sub-optimal and the application of Gaussian matrices to large data sets could be prohibitively expensive as they do not admit fast transforms. To address the second issue, inspired by [14], [41] considered a structured random matrix A with a fast matrix-vector multiply, to speed up RFFs. For this construction [42] claimed a concentration result analogous to that of [54], again using an ϵ-net argument. As for quantized RFF's, starting with the first work [53], the results so far only show that one can approximate a (complicated) function of the kernel, rather than the kernel itself.

Instead, with an eye towards excellent approximations of the kernel itself, we will use noise-shaping quantization schemes (to promote compressed representations), couple them with fast randomized transforms A (to promote fast computation), and we will design f from (16) so that $\langle f(Q(\widetilde{\mathbf{\Phi}}(\mathbf{x}))), f(Q(\widetilde{\mathbf{\Phi}}(\mathbf{x}')))\rangle = \mathbb{E}(k(\mathbf{x},\mathbf{x}'))$. We will then show that $\langle f(Q(\widetilde{\mathbf{\Phi}}(\mathbf{x}))), f(Q(\widetilde{\mathbf{\Phi}}(\mathbf{x}')))\rangle$ concentrates, with high probability, around its mean, the kernel. To that end, we will utilize tools, such as generic chaining [64], from high-dimensional probability [66] to allow us to handle the difficulties induced not only by the non-linearities but also by the structured (i.e., not fully independent) random matrices. As an added benefit, we believe our methods will allow us to obtain improved dependence of m on the geometric properties of $\mathcal{X}$, rather than on the ambient dimension.

6.5 Proposed Work: Quantized Statistics

Alex : under construction: notation being worked on

Our overarching goal is to reduce the communication constraints by utilizing both sketching and binarization through use of $g(x) = f(Q(\widetilde{\mathbf{\Phi}}(x)))$. The idea is that, rather than communicate and compute with $x \in \mathbb{R}^d$ where d may be large, one only needs to compute $Q(\widetilde{\mathbf{\Phi}}(x)) \in \{\pm 1\}^m$ with $m \ll d$, and communicate $f(Q(\widetilde{\mathbf{\Phi}}(x)))$. In what follows, $\mathbb{E}_{x\in X} g(x)$ serves as a proxy for the empirical mean embedding of the data set X. For notational convenience in this section, it will at times be easier to separate the transfer function ϕ from the randomized measurement operator $\mathcal{A}$ and define

$$<<<<<<< HEAD \widetilde{g}(y) = f(Q(\mathcal{A}(y))), \text{ where } y = \mathbf{\Phi}(x). ======= \widetilde{g}(y) = f(Q(\mathcal{A}(y))), \quad \text{where} \quad y = \mathbf{\Phi}(x). >>>>>>>$$

We aim to prove that this low complexity vector still converges to a type of mean embedding, and to provide a rigorous analysis of the statistical power, convergence rates, and minimal detectable separation criteria between the distributions. Below we highlight the benefit of this approach in a number of different sensor problems.

- Two Sample Testing: In the context of sensors, the two sample problem can be summarized as follows: each sensor collects a data set $\mathbf{\Phi}_i(\mathbf{X}) = \mathbf{Y}_i \sim p_i$, and the goal is to determine whether the $\mathbf{Y}_i$ were distributed similarly. To address this, we will analyze the binarized statistic

$$\left\| \frac{1}{n}\sum_{x\in\mathbf{Y}_1} \widetilde{g}(x) - \frac{1}{m}\sum_{y\in\mathbf{Y}_2} \widetilde{g}(y) \right\|,$$

  under appropriate norm, and seek to characterize the minimal conditions under which a deviation between p_1 and p_2 can be detected. The approach requires characterizing the types of deviations $p_1 - p_2$ that can be detected, namely those for which

$$\left\| \int e^{-\|x-y\|^2/\sigma^2}(p_1(y) - p_2(y))dy \right\| > \epsilon,$$

as well as the rates at which these deviations can be detected. The communication benefit of such a statistic is that the sensors need only transmit the mean of $g(x)$, rather than all the individual points.

Alex : Could expand on this, or reference other comments in proposal and a few papers about this for JL embeddings

- Change Point Detection: A variant of the two sample testing problem is change point detection, in which the data is streaming $X(t)$ according to some underlying stochastic process. At some time t^*, the distribution of the stochastic process changes from one distribution to another, and the issue is how quickly after t^* this change can be detected. Unlike the two sample context in which we were testing whether the sensors detected the same distribution, here we can use the sensors collaboratively by constructing the concatenated sketching matrix $g(\mathbf{X})$. There exists a mean embedding approach to change point analysis [28] which uses the kernel Fisher discriminant ratio and mean embedding to measure the homogeneity between time segments of the process. However, this once again requires storage of all points over the length of the detection window and communication of those points across sensors for both computing the window mean and variance. It also suffers from an inability to begin computing the change point statistic until all points in the window have been collected. We aim to introduce the binarized sketching framework to produce an efficient computation to the kernel mean as in two sample testing, as well as a fast construction of a low rank approximation to the kernel covariance matrix. We will derive it's new limiting distribution under the null model of no change, as well as the consistency under the alternative distribution when a change does occur.
- Multiple Sensor Common Factor Identification: A common issue is aggregating multiple sensors to identify and magnify the signal detected by both sensors. Under a linear model, algorithms such as Canonical Correlation Analysis [29] act on multiple streams of simultaneously collected data $\mathbf{y}_i(t)$ to filter noise and recover highly correlated linear projections from two the data sets. Recently, a kernel CCA technique called alternating diffusion [43] has be used to identify common nonlinear effects by building a kernel K_i from each sensing modality and analyzing the product kernel $(K_1K_2)^t$. Concretely, assume that there exists a hidden manifold $\mathcal{M}$ and two nuisance manifolds $\mathcal{N}_1$ and $\mathcal{N}_2$, and samples $\mathbf{\Phi}((x_i, z_i^{(1)}, z_i^{(2)}))$ for $(x_i, z_i^{(1)}, z_i^{(2)}) \in \mathcal{M} \times \mathcal{N}_1 \times \mathcal{N}_2$. Due to sensor location or modality, sensor 1 collects data points $Y_1 = transfer_1((x_i, z_i^{(1)}, \xi))$ and sensor 2 collecting data points $Y_2 = \mathbf{\Phi}_2((x_i, \zeta, z_i^{(2)}))$. Then roughly speaking, for some assumptions on $\mathbf{\Phi}$ and for $K_1 : Y_1 \times Y_1 \to \mathbb{R}$ and $K_2 : Y_2 \times Y_2 \to \mathbb{R}$, Talmon and Wu [65] proved $(K_1K_2)^t$ is the diffusion kernel on $\mathcal{M}$ only. This is a very beneficial feature, as it means that one can compute kernel statistics for distributions defined on $\mathcal{M}$ only, independent of the nuisance features that may differ between sensors due only to modality or location.

  However, this requires communication of all n points across sensors to compute the kernel product. We propose to analyze such approaches under minimal communication constraints by utilizing the low rank binarized decomposition $K_i \approx \widetilde{g}(Y_i)\widetilde{g}(Y_i)^*$. As the key feature of alternating diffusion is computing the inner product matrix $\widetilde{g}(Y_1)^*\widetilde{g}(Y_2)$, which has dimension $d < n$ and is also low rank. This implies it is possible to sketch $\widetilde{g}(Y_1)$ (resp. $\widetilde{g}(Y_2)$) with a small set of vectors v_j (resp. w_k) and only communicate vectors $\widetilde{g}(Y_1)^*v_j$ (resp. $w_k^*\widetilde{g}(Y_2)$) such that $\mathbb{E}_{j,k}[\widetilde{g}(Y_1)^*v_j w_k^*\widetilde{g}(Y_2)] \approx \widetilde{g}(Y_1)^*\widetilde{g}(Y_2)$. The amplification of the common factors observed by both sensors can serve to boost the power of the two sample and change point statistics on the shared observable manifold $\mathcal{M}$, as well as other compressed statistics that can be computed [27].

7 Tell me something new

Consider a video system for monitoring a highway. Lets say that the highway is 100 miles long and that we have a 5 cameras per mile, i.e. 500 cameras. Each camera generates a large raw data stream, on the order of 100MB per second. Using video compression this stream can be brought down to XXMB/sec. Totalling XXXMB/sec. A common approach is to send all of this data to a centralized site where a team of analysts monitor it to identify accidents, road hazards etc.

Several issues should be noted:

- A high bandwidth channel is needed to transmit the information to the center. If this channel is wireless, this transmission will be costly in terms of energy and available bandwidth.

- Manually monitoring 500 video channels a significant task for the analysts. AI methods can be used to aid in that task, our goal is to perform this AI on the camera, rather than in the centralized site.
- Under normal conditions, highway traffic is highly predictable. Only unpredictable situations such as an accident, an object on the road or a speeding car warrant the transmission of high-resolution images.

We propose a general framework for designing sensor networks for anomaly detection called "tell me something new" [**?**]. In this approach, sensors analyze the data they collect, and send message to the center only when it estimates that this information is significantly different than what the center will predict without the information.

Let $\hat{\theta}_1(t), \hat{\theta}_2(t)$ be the estimates of the target location for each sensor. *In addition, each sensor maintains an estimate of the estimate of the other sensor.* $\hat{\theta}_{1,2}(t), \hat{\theta}_{2,1}$. Each sensor updates it's estimate of the velocity of the location of the target according to the signal it measures, but it does not update it's estimate of the other's estimate. If the two estimate are close to each other $\hat{\theta}_i(t) \approx \hat{\theta}_{i,j}(t)$ then sensor i sends no information to sensor j. On the other hand, if $\hat{\theta}_i(t)$ is far from $\hat{\theta}_{i,j}(t)$, then $\hat{\theta}_i$ is transmitted from sensor i to sensor j. Thus if the target is moving in constant speed, uninterrupted, there is not communication between the sensors.

The basic idea here is that a sensor sends out information only if that information cannot be predicted by the receiver. Similar ideas have been used in arithmetic coding and **Yoav :** I think in $\Sigma\Delta$ encoding.

Recently, PI Freund [**?**] proposed an asynchronous computation model called "Tell Me Something New" in which each agent broadcasts a message only when the estimate it computes differs from the existing estimates in a statistically significant way.

One important application of sensor networks is to monitor activity and identify anomalies. Examples include: building security systems, factory floors, highway monitoring, health monitoring for the sick or elderly and many others.

On its face, this might seem like an under-constrained impossible problem. However, note that for all of the environments listed above there is a highly repetitive pattern from day to day and from week to week. Add to that the sensors are stationary, and one would expect that most sensors observe highly regular and highly predictable patterns.

The approach we propose in this case is that each sensor creates a model of the characteristics of the signals that it observes during normal operations. It alerts neighboring sensors if it observes something that is abnormal, i.e. a signal that has very low probability according to the model. When several sensors send an alert with a short time window, and when the alerts are consistent with each other, a global alert is sent to the human operators.

8 Broad Impact

Sensor networks are an important emerging technology with applications in retail, manufactoring, security and medicine. These networks collect vast amounts of raw data, most of which is not relevant to the task of the system as a whole.

On the other hand, the deployment of such systems places strict constraints on the power and bandwidth available to each sensor. Sensors are expected to operate for years on a small battery or by foraging energy from the environment. A related problem is the bandwidth, range and energy consumption of wireless communication protocols, which greatly limit the data-rate sent in an out of each sensor.

Our proposed approach of *learning sensor networks* if successful, will enable a new generation of sensor networks which will imact all aspect of modern society.

9 Intellectual Merit

Sensor networks present some challanges which are hard to...

10 Results from previous grants

R. Saab: *Sampling and quantization theorems for modern data acquisition (DMS-1517204, 08/01/2015–07/31/19, $160,404)* This grant resulted in 9 published or accepted journal articles [38, 46, 51, 59, 52, 57, 34, 30, 22], and 4 conference papers [50, 21, 34, 31]. **Intellectual Merit:** [38, 59, 22, 57, 31] study quantization of compressed sensing (CS) measurements, under various signal, measurement, and quantization models.

[30] develops techniques and theory for embedding data into the binary cube while [52] provides a framework for using 1-bit measurements for classification. [46, 51] study the use of prior support information in CS. [34, 32] study phase retrieval. **Broader Impacts:** Saab disseminated the results through multiple invited talks, and has developed and taught graduate courses on (1) compressed sensing and its applications, (2) applied and computational harmonic analysis, and (3) mathematical methods in data science. He mentored a UCSD undergraduate (currently a PhD student at UCLA), and advised four UCSD graduate students and two postdocs, some of whom are co-authors on the works above, while others are taking part in ongoing work. Two of the mentees are female, and two are members of under-represented groups.
P. Gerstoft: 2014-2018; PLR-1246151 (SIO; 844k); Collaborative Research: Dynamic Response of the Ross Ice Shelf to Wave-induced Vibrations. Intellectual Merit: This project (DRIS) investigates the response of ice shelves to ocean wave forcing to (1) infer bulk elastic properties from signal propagation characteristics, and (2) to determine how IG wave and other gravity wave forcing propagates across the Ross Ice Shelf (RIS), and (3) to monitor seasonal variability of the RIS response and icequake activity. In 2014, 16 DRIS broadband stations were installed to acquire data for 2 years. Subsequently, 13 geodetic GPS stations were installed during the 2015 field season recovery of the first year's seismic data. Joint processing of the 2016 seismic and GPS data are underway. **Broader Impacts**: Supported 2 postdocs, 2 graduate students. NSF Artists and Writers awardee Glenn McClure produced symphonic and choral works from the DRIS data. The Birch Aquarium at Scripps is developing an Antarctic exhibit featuring DRIS results. Publications: [Bromirski et al., 2015; Diez et al., 2015; Bromirski et al., 2017; Chen et al., 2018]. A project web site [Bromirski, 2014], suitable for informing the public, is maintained.
A. Cloninger: NSF MSPRF, award number DMS-1402254, "A generalized framework for heterogeneous data fusion without point registration", from July 2014 to June 2017. This grant resulted in 12 accepted or submitted publications []. **Intellectual merit:** [48, 11, 61] study the algorithmic and theoretical aspects of deep learning in manifold contexts and for finding a encoding features that are shared across multiple networks or data sets. [13, 9, 10] study a related question of crafting new embeddings and latent spaces that move beyond the Laplace Beltrami eigenfunctions to embeddings involving predicting external functions, directed networks, and solutions to the wave equation, all of which can have benefit in the context of aligning embeddings across data sets. [8, 18, 2, 37] focus on various applications of these frameworks in various medical problems. And [6, 12] developed methods for testing whether non-registered heterogeneous data sources were distributionally similar. **Broader impacts:** The broader impact of the project was in generalizing the notion of heterogeneous data and avoiding heuristic approaches. This created opportunities for interdepartmental collaborative efforts, continuing collaboration with the investigator's colleagues at the National Institutes of Health, Brigham and Women's Hospital, Cincinnati Children's Hospital Medical Center, and the Center for Outcome Research and Evaluation. The grant also led to an undergraduate research project that has culminated in a well-cited paper, where the undergraduate was a lead author [**?**].

References

[1] N. Ailon and B. Chazelle. The fast johnson–lindenstrauss transform and approximate nearest neighbors. *SIAM Journal on computing*, 39(1):302–322, 2009.

[2] J. Bates and A. Cloninger. Outcome based matching. *arXiv preprint arXiv:1712.05063*, 2017.

[3] M. Bianco and P. Gerstoft. Dictionary learning of sound speed profiles. *J. Acoust. Soc. Am.*, 141(3):1749–1758, 2017.

[4] M. J. Bianco and P. Gerstoft. Travel time tomography with adaptive dictionaries. *IEEE Transactions on Computational Imaging*, 4(4):499–511, 2018.

[5] P. T. Boufounos and R. G. Baraniuk. 1-bit compressive sensing. In *2008 42nd Annual Conference on Information Sciences and Systems*, pages 16–21. IEEE, 2008.

[6] X. Cheng, A. Cloninger, and R. R. Coifman. Two-sample statistics based on anisotropic kernels. *arXiv preprint arXiv:1709.05006*, 2017.

[7] E. Chou and C. S. Güntürk. Distributed noise-shaping quantization: I. beta duals of finite frames and near-optimal quantization of random measurements. *Constructive Approximation*, 44(1):1–22, 2016.

[8] A. Cloninger. Function driven diffusion for personalized counterfactual inference. *arXiv preprint arXiv:1610.10025*, 2016.

[9] A. Cloninger. A note on markov normalized magnetic eigenmaps. *Applied and Computational Harmonic Analysis*, 43(2):370–380, 2017.

[10] A. Cloninger. Prediction models for graph-linked data with localized regression. In *Wavelets and Sparsity XVII*, volume 10394, page 103940S. International Society for Optics and Photonics, 2017.

[11] A. Cloninger, R. R. Coifman, N. Downing, and H. M. Krumholz. Bigeometric organization of deep nets. *Applied and Computational Harmonic Analysis*, 44(3):774–785, 2018.

[12] A. Cloninger, B. Roy, C. Riley, and H. M. Krumholz. People mover's distance: Class level geometry using fast pairwise data adaptive transportation costs. *Applied and Computational Harmonic Analysis*, 2018.

[13] A. Cloninger and S. Steinerberger. Spectral echolocation via the wave embedding. *Applied and Computational Harmonic Analysis*, 43(3):577–590, 2017.

[14] A. Dasgupta, R. Kumar, and T. Sarlós. Fast locality-sensitive hashing. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1073–1081. ACM, 2011.

[15] I. Daubechies and R. DeVore. Approximating a bandlimited function using very coarsely quantized data: A family of stable sigma-delta modulators of arbitrary order. *Annals of mathematics*, 158(2):679–710, 2003.

[16] I. Daubechies and R. Saab. A deterministic analysis of decimation for sigma-delta quantization of bandlimited functions. *IEEE Signal Processing Letters*, 22(11):2093–2096, 2015.

[17] I. Dokmanic, R. Parhizkar, J. Ranieri, and M. Vetterli. Euclidean distance matrices: essential theory, algorithms, and applications. *IEEE Signal Processing Magazine*, 32(6):12–30, 2015.

[18] N. S. Downing, A. Cloninger, A. K. Venkatesh, A. Hsieh, E. E. Drye, R. R. Coifman, and H. M. Krumholz. Describing the performance of us hospitals by applying big data analytics. *PloS one*, 12(6):e0179603, 2017.

[19] M. Elad. *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.

[20] J. Fang, Y. Shen, H. Li, and Z. Ren. Sparse signal recovery from one-bit quantized data: An iterative reweighted algorithm. *Signal Processing*, 102:201–206, 2014.

[21] J.-M. Feng, F. Krahmer, and R. Saab. Quantized compressed sensing for partial random circulant matrices. In *2017 International Conference on Sampling Theory and Applications (SampTA)*, pages 236–240. IEEE, 2017.

[22] J.-M. Feng, F. Krahmer, and R. Saab. Quantized compressed sensing for random circulant matrices. *Applied and Computational Harmonic Analysis*, 2019.

[23] K. L. Gemba, S. Nannuru, P. Gerstoft, and W. Hodgkiss. Multi-frequency sparse Bayesian learning for robust matched field processing. *J. Acoust. Soc. Am.*, 141(5):3411–3420, 2017.

[24] P. Gerstoft, C. F. Mecklenbräuker, W. Seong, and M. Bianco. Introduction to compressive sensing in acoustics, 2018.

[25] P. Gerstoft, K. G. Sabra, P. Roux, W. Kuperman, and M. C. Fehler. Green's functions extraction and surface-wave tomography from microseisms in southern california. *Geophysics*, 71(4):SI23–SI31, 2006.

[26] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13(Mar):723–773, 2012.

[27] R. Gribonval, G. Blanchard, N. Keriven, and Y. Traonmilin. Compressive statistical learning with random feature moments. *arXiv preprint arXiv:1706.07180*, 2017.

[28] Z. Harchaoui, E. Moulines, and F. R. Bach. Kernel change-point analysis. In *Advances in neural information processing systems*, pages 609–616, 2009.

[29] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor. Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664, 2004.

[30] T. Huynh and R. Saab. Fast binary embeddings, and quantized compressed sensing with structured matrices. *arXiv preprint arXiv:1801.08639*, 2018.

[31] M. Iwen, E. Lybrand, A. Nelson, and R. Saab. New algorithms and improved guarantees for one-bit compressed sensing on manifolds. *arXiv preprint arXiv:1902.03726*, 2019.

[32] M. Iwen, B. Preskitt, R. Saab, and A. Viswanathan. Phase retrieval from local measurements in two dimensions. In *Wavelets and Sparsity XVII*, volume 10394, page 103940X. International Society for Optics and Photonics, 2017.

[33] M. Iwen and R. Saab. Near-optimal encoding for sigma-delta quantization of finite frame expansions. *Journal of Fourier Analysis and Applications*, 19(6):1255–1273, 2013.

[34] M. A. Iwen, B. Preskitt, R. Saab, and A. Viswanathan. Phase retrieval from local measurements: Improved robustness via eigenvector-based angular synchronization. *Applied and Computational Harmonic Analysis*, 2018.

[35] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk. Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. *IEEE Transactions on Information Theory*, 59(4):2082–2102, 2013.

[36] W. B. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984.

[37] J. L. Katzman, U. Shaham, A. Cloninger, J. Bates, T. Jiang, and Y. Kluger. Deepsurv: Personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC medical research methodology*, 18(1):24, 2018.

[38] K. Knudson, R. Saab, and R. Ward. One-bit compressive sensing with norm estimation. *IEEE Transactions on Information Theory*, 62(5):2748–2758, 2016.

[39] Q. Kong, D. T. Trugman, Z. E. Ross, M. J. Bianco, B. J. Meade, and P. Gerstoft. Machine learning in seismology: Turning data into insights. *Seismological Research Letters*, 90(1):3–14, 2018.

[40] B. Le, T. W. Rondeau, J. H. Reed, and C. W. Bostian. Analog-to-digital converters. *IEEE Signal Processing Magazine*, 22(6):69–77, 2005.

[41] Q. Le, T. Sarlós, and A. Smola. Fastfood-approximating kernel expansions in loglinear time. In *Proceedings of the international conference on machine learning*, volume 85, 2013.

[42] Q. V. Le, T. Sarlos, and A. J. Smola. Fastfood: Approximate kernel expansions in loglinear time. *arXiv preprint arXiv:1408.3060*, 2014.

[43] R. R. Lederman and R. Talmon. Learning the geometry of common latent variables using alternating-diffusion. *Applied and Computational Harmonic Analysis*, 44(3):509–536, 2018.

[44] M. Ledoux. *The concentration of measure phenomenon*. Number 89. American Mathematical Soc., 2001.

[45] S. Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[46] H. Mansour and R. Saab. Recovery analysis for weighted ℓ_1-minimization using the null space property. *Applied and Computational Harmonic Analysis*, 43(1):23–38, 2017.

[47] C. A. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(Dec):2651–2667, 2006.

[48] G. Mishne, U. Shaham, A. Cloninger, and I. Cohen. Diffusion nets. *Applied and Computational Harmonic Analysis*, 2017.

[49] K. Muandet, K. Fukumizu, B. Sriperumbudur, B. Schölkopf, et al. Kernel mean embedding of distributions: A review and beyond. *Foundations and Trends® in Machine Learning*, 10(1-2):1–141, 2017.

[50] D. Needell, R. Saab, and T. Woolf. Simple object classification using binary data. In *2017 AAAI Fall Symposium Series*, 2017.

[51] D. Needell, R. Saab, and T. Woolf. Weighted-minimization for sparse recovery under arbitrary prior information. *Information and Inference: A Journal of the IMA*, 6(3):284–309, 2017.

[52] D. Needell, R. Saab, and T. Woolf. Simple classification using binary data. *The Journal of Machine Learning Research*, 19(1):2487–2516, 2018.

[53] M. Raginsky and S. Lazebnik. Locality-sensitive binary codes from shift-invariant kernels. In *Advances in neural information processing systems*, pages 1509–1517, 2009.

[54] A. Rahimi and B. Recht. Random features for large-scale kernel machines. In *Advances in neural information processing systems*, pages 1177–1184, 2008.

[55] S. Ravishankar and Y. Bresler. Mr image reconstruction from highly undersampled k-space data by dictionary learning. *IEEE transactions on medical imaging*, 30(5):1028–1041, 2011.

[56] N. Riahi and P. Gerstoft. Using graph clustering to locate sources within a dense sensor array. *Signal Processing*, 132:110–120, 2017.

[57] R. Saab and E. Lybrand. Quantization for low-rank matrix recovery. *Information and Inference: A Journal of the IMA*, 8(1):161–180, 05 2018.

[58] R. Saab, R. Wang, and Ö. Yılmaz. Quantization of compressive samples with stable and robust recovery. *Applied and Computational Harmonic Analysis*, 44(1):123–143, 2018.

[59] R. Saab, R. Wang, and O. Yılmaz. From compressed sensing to compressed bit-streams: Practical encoders, tractable decoders. *IEEE Transactions on Information Theory*, 64(9):6098–6114, Sep. 2018.

[60] K. G. Sabra, P. Gerstoft, P. Roux, W. Kuperman, and M. C. Fehler. Surface wave tomography from microseisms in southern california. *Geophysical Research Letters*, 32(14), 2005.

[61] U. Shaham, A. Cloninger, and R. R. Coifman. Provable approximation properties for deep neural networks. *Applied and Computational Harmonic Analysis*, 44(3):537–557, 2018.

[62] N. M. Shapiro, M. Campillo, L. Stehly, and M. H. Ritzwoller. High-resolution surface-wave tomography from ambient seismic noise. *Science*, 307(5715):1615–1618, 2005.

[63] B. Sriperumbudur and Z. Szabó. Optimal rates for random fourier features. In *Advances in Neural Information Processing Systems*, pages 1144–1152, 2015.

[64] M. Talagrand. *The generic chaining: upper and lower bounds of stochastic processes*. Springer Science & Business Media, 2006.

[65] R. Talmon and H.-t. Wu. Latent common manifold learning with alternating diffusion: analysis and applications. *Applied and Computational Harmonic Analysis*, 2018.

[66] R. Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press, 2018.

[67] A. Xenaki, P. Gerstoft, and K. Mosegaard. Compressive beamforming. *J. Acoust. Soc. Am.*, **136**(1):260–271, 2014.

[68] A. M. Zoubir, V. Koivunen, E. Ollila, and M. Muma. *Robust Statistics for Signal Processing*. Cambridge University Press, 2018.