Deterministic Sparse Fourier Transform for Continuous Signals with Frequency Gap

Abstract

The Fourier transform serves as a fundamental tool in theoretical computer science and signal processing. In particular, when the signal is *sparse* in the frequency domain—having only k distinct frequencies—sparse Fourier transform (SFT) algorithms can recover the signal in a sublinear time (proportional to the sparsity k). Most prior research focused on SFT for discrete signals, designing both randomized and deterministic algorithms for one-dimensional and high-dimensional discrete signals. However, SFT for continuous signals (i.e., $x^*(t) = \sum_{j=1}^{k} v_j e^{2\pi \mathbf{i} f_j t}$ for $t \in [0, T]$) is a more challenging task. Note that the discrete SFT algorithms are not directly applicable to continuous signals due to the sparsity blow-up from the discretization. In a seminal work by Price and Song (FOCS 2015), they proposed the first *randomized* algorithm that achieves an ℓ_2 recovery guarantee in $O(k \operatorname{poly}\log(FT/\eta))$ time, where F is the band-limit of the frequencies and η is the frequency gap. Nevertheless, it remains open whether we can solve this problem without using any randomness. In this work, we address this gap and introduce (to the best of our knowledge) the first sublinear-time *deterministic* sparse Fourier transform algorithm in the continuous setting. Specifically, our algorithm uses in $O(k^2 \operatorname{poly}\log(FT/\eta))$ samples and reconstructs the signal in the ℓ_1 regime. It is the optimal recovery guarantee that can be achieved by *any* deterministic approach.

1 Introduction

The Fourier transform (FT) was introduced by Joseph Fourier in 1822 [Fou22]. Today, it is widely used in the fields of theoretical computer science (TCS) and applied mathematics. In TCS, it has many applications, including integer multiplication [Für09], Subset Sum and 3SUM [CLRS09, Bri17, KX17], linear programming [LSZ19, JSWZ21], distributional learning [DKS16a, DKS16b, DKS16c], learning mixture of regressions [CLS20], fast Johnson-Lindenstrauss transform [LDFU13], and TensorSRHT [AKK+20, SWYZ21] with its applications to optimization [SZZ21]. In applied math, the Fourier transform serves as a key mathematical tool in solving partial differential equations and doing function approximation. Furthermore, Fourier transform has a vast range of real-world applications, including signal processing, electrical engineering, pattern recognition, feature extraction, image/audio/video compression, etc. Due to its significance in both theory and practice, finding an efficient algorithm to calculate the Fourier transform is of utmost importance.

Since the seminal fast Fourier transform (FFT) algorithm by [CT65], the TCS community has been seeking a faster $o(N \log N)$-time algorithm – without success yet. In contrast, as quoted from Indyk and Kapralov [IK14]:

> *"many of these applications rely on the fact that most of the Fourier coefficients of the signals are small or equal to zero, i.e., the signals are (approximately) sparse."*

I.e., when the Fourier spectrum is approximately k-*sparse* for a *sublinear* parameter $k = o(N)$, we even hope for better sublinear-sample/-time algorithms. These practical and theoretical desires motivate the research interest in the *sparse Fourier transform* problem (Sparse FT).

Over the last two decades, the Sparse FT problem has been explored and extended in various directions. Now, we can say it constitutes a "subarea" within sublinear algorithms. The prior works follow two lines: (i) those in the discrete settings (or the compressed sensing literature) [KM93, GGI+02, GMS05, CT06, Don06, RV08, BD08, HIKP12a, HIKP12b, CGV13, IKP14, IK14, Bou14, Kap16, HR16, Kap17, LN20, KVZ19, NSW19, NS19]; and (ii) those in the continuous setting [BCG+12, Moi15, PS15, CKPS16, CP19b, CP19a, JLS23, SSWZ23].

Although more researchers have concentrated their attention on DFT algorithms, the continuous setting is also important since lots of signals have a continuous domain. Currently, all of the previous work about continuous Fourier transform algorithm are randomized. It is natural to define and study the continuous setting where a deterministic algorithm is possible. This work presents a positive answer to that.

We state our main result as follows,

Theorem 1.1 (Informal version of Theorem 2.9)**.** *Let $\mathcal{F} = [-F, -F+\eta, \cdots, -\eta, 0, \eta, \cdots, F-\eta, F]$ denote the candidates of heavy frequency. Let $g(t)$ denote the noise whose Fourier spectrum spans over the entire frequency domain and it satisfies mild assumptions in the time domain. Let $x(t) = x^*(t) + g(t)$ denote the observed signal. For $T \geq \widetilde{O}(1/\eta)$, there exists a deterministic algorithm that observes the signal at time points in $S \subset [0, T]$ of size $|S| = O(k^2 \operatorname{poly}(FT))$, and recovers all the f_i and v_i accurately.*

We remark that the naive/straightforward algorithm will take $F/\eta = O(FT)$ samples and run a discrete Fourier transform. Our algorithm only needs $\widetilde{O}(k^2)$ samples which is sublinear in $O(FT)$.

Notations

We use $a \gtrsim b$ to denote that $a \geq C \cdot b$ for some constant $C > 0$. Let n be a positive integer, $[n] := \{1, 2, \cdots, n\}$. We use $\mathbf{i} := \sqrt{-1}$. And we use ω to represent $e^{-2\pi \mathbf{i}}$. For a complex number

$z \in \mathbb{C}$ with $z = a + \mathbf{i}b$ where $a, b \in \mathbb{R}$. We use $|z|$ to denote $\sqrt{a^2 + b^2}$. z can also be expressed as $z = r \cdot e^{\mathbf{i}\theta}$, where $r \in \mathbb{R}_{>0}$ and $\theta \in [0, 2\pi]$. We define $\arg(z) = \theta$. Let $\{z_i\}$ be a sequence of complex numbers. It's median is defined as $\operatorname{median} z_i = \operatorname{median} \operatorname{Re}(z_i) + \mathbf{i} \operatorname{median} \operatorname{Im}(z_i)$. We use $\Pr[]$ to denote probability. We use $\mathbb{E}[]$ to denote expectation. For $x \in \mathbb{R}$, we use $\operatorname{round}(x)$ to denote the integer with the closest distance to x. Let $x = i + q$ where i is an integer and $q \in [0, 1)$, we define $x \pmod 1 := q$. In this article, we consider interval $[0, 1]$ and $[-1/2, 1/2]$ as the same. For example for a function $f : [0, 1] \to \mathbb{R}$, we consider $f(-x) = f(1 - x)$. Let $x(t) : \mathbb{R} \to \mathbb{C}$ be a function. For a finite set $S \subset \mathbb{R}$, we define $\|x_S\|_1 := \sum_{t \in S} |x(t)|$. Now let S be a finite sequence, we define x_S be the vector in which t-th entry is $x(S_t)$ where S_t denotes the t-th element in S. We use $\mathbf{0}_n$ to denote a vector formed by n zeros.

2 Technical Overview

In this section, we provide an overview of our contribution. We start by summarizing the framework in [LN20], and then we present the motivation and details for our algorithm.

2.1 Summary of previous works

Related works Sparse FFT algorithm searches for the active frequency by binning them into a small number of bins. In the discrete setting, [HIKP12a, HIKP12b] introduced new methods for locating the isolated signal and updating the signal by directly filtering the bins, which improved the time and sample complexity. [IK14] presented a recursive single-entry reduction algorithm which gives a ℓ_∞ norm guarantee. Based on their result and a modified HashToBins with initial guess from [Kap17], [LN20] introduced a deterministic algorithm by de-randomization w.r.t. the hashing functions. In the continuous setting, [PS15, CKPS16] defined the k-sparse continuous signal and used a randomized time-sampling technique to control time-varying noise. They extended the guarantee of the fast DFT algorithm to the continuous setting by the identity between CFT, DTFT, and DFT, which is helpful in the analysis of our work. [JLS23] is a higher-dimensional generalization of [PS15]. With previous work, our research is able to provide the first deterministic continuous sparse FT algorithm. We result in a ℓ_1, ℓ_2-mixed norm guarantee for error.

Contribution of [LN20] and their limitation in Continuous setting During the hashing and detection of active signal, it is possible that two distinct active frequencies are hashed into the same bin and hence cannot be recovered. [LN20] gave a formal definition of this event and used a de-randomization strategy to find a fixed sequence of hashing functions that prevent this event. Their method is to use a pessimistic estimator to upper-bound the possibility of bad events and to reduce it by choosing proper hashing parameters. Then, they embedded this deterministic HashToBins algorithm into the ℓ_∞ norm reduction algorithm in [IK14] to reach the final result.

Notice that the strategy of [LN20] is not directly feasible for continuous signals. For there is only a finite light-hitter in the discrete setting, it is able to traverse through all frequency points and generate a deterministic hashing sequence. However, we need to deal with the continuous noise function in our algorithm. Moreover, the hashing scheme in the discrete setting cannot apply to the continuous setting. Not like DFT, our CFT algorithm takes samples from the time interval with unequal length with the active frequency set, which leads to a different hashing and filtering strategy. We need to change our de-randomization steps to fit in the new hashing functions.

2.2 Our techniques

In this work, we generalize [LN20] to the continuous setting and overcome the limitations. In this section, we first discuss the problem setting. Then, we show how to generate the de-randomized hashing sequence under the hashing scheme of sparse CFT. Next, we propose a reasonable noise model that enables an efficient and robust deterministic sparse Fourier transform algorithm. We further show how to combine the continuous HashToBins with the de-randomized hash sequence, which leads to our main theorem (Theorem 2.9).

Problem formulation As we mentioned before, we study the Fourier transform of a continuous k-sparse signal, which is defined as follows.

Definition 2.1 (Continuous-time, k-Fourier-sparse signal)**.** *Let $k \in \mathbb{Z}_{>0}$. Let $\delta_{f_i}(f)$ denote the Dirac function centered at f_i. We define the k-Fourier-sparse signal $\widehat{x}^*(f)$ to be as follows:*

$$\widehat{x}^*(f) = \sum_{j=1}^{k} v_j \cdot \delta_{f_j}(f) \qquad \xrightarrow{\text{CFT}} \qquad x^*(t) = \sum_{j=1}^{k} v_j \cdot e^{2\pi \mathbf{i} f_j t}$$

where $v_j \in \mathbb{C}$ is the coefficient and $f_j \in \mathcal{F}$ is the frequency for each $j \in [k]$. We use $\mathcal{K}$ to denote the set of f_j's.

Moreover, we assume that the frequency range $\mathcal{F}$ is a set of equidistant points in $[-F, F]$. See the formal definition below.

Definition 2.2 (Possible range of active frequency)**.** *For a given frequency gap η and bounded range $[-F, F]$, we use $\mathcal{F}$ to denote the set*

$$\{i \cdot \eta \mid \forall i \in \mathbb{Z}, \text{ and } i \cdot \eta \in [-F, F]\}.$$

In our observation of signal $x^*(t)$, we receive a time-varying noise, denoted by the continuous function $g(t)$. Therefore, the observed $x(t)$ is the summation of real signal $x^*(t)$ and $g(t)$.

De-randomization To discover the k active frequency from $\mathcal{F}$, we use a hashing and filtering method. First, we hash the points in $\mathcal{F}$ into $B = O(k)$ buckets. Then we use a pair of filter functions $(G, \widehat{G})$ to select the active frequency. $\widehat{G}$ is constructed to be close to 1 in the center of its domain and close to 0 elsewhere. Let σ, b be parameters in the hash function, we use function $o_{f,\sigma,b}(f')$ to measure the distance from the hashing of f' to the center of the bucket where f is hashed into. Morally, if $o_{f,\sigma,b}(f)$ is big and $o_{f,\sigma,b}(f')$ is small, i.e., f is hashed close to the center of a bucket, while f' is not close or in a different bucket, then we are able to discover f by the filter function. However, two active frequencies, f and f', may hash to the center of the same bucket, which obstructs the discovery. This gives us the motivation to define bad events as below.

Definition 2.3 (Bad events $A_{f,f'}$)**.** *We use d to denote the time we call* HashToBins *in one round of sparse detection. We use β as a factor depending on $|\mathcal{F}|$, which will be determined later. Let $\{(\sigma_r, b_r)\}_{r=1}^{d}$ be a sequence of hashing parameters. For any $f, f' \in \mathcal{F}$, $f \neq f'$, we define $A_{f,f'}$ to be the event that*

$$\sum_{r=1}^{d} \widehat{G}_{o_{f,\sigma_r,b_r}(f')} \geq \beta.$$

This definition says that, although we cannot guarantee that the collision of f and f' does not happen in a single turn of hashing, we can control the total time of the collision in a sequence of hashing. Then, we can run the algorithm multiple times and take the median of output to reach a good approximation.

Previous CFT research (e.g. [PS15, JLS23]) used a randomized hashing function to control the impact of bad events. They bounded the expectation of error in each stage and repeated multiple stages to reach a small failure probability. Different from them, this work finds a sequence of deterministic hashing function $\{(\sigma_r, b_r)\}_{r=1}^{d}$, which prevents the happening of bad events. The recursive procedure can be summarised as follows.

- Initial state: Let $h_r(f, f', \sigma_1, b_1, \cdots, \sigma_r, b_r)$ be a sequence of function satisfying

$$\sum_{f,f' \in \mathcal{F}: f \neq f'} h_0(f, f') < 1$$

  Notice that $h_0(f, f')$ is determined at initial state while $h_r(f, f')$ depends on σ_r, b_r which are chosen later.

- De-randomization step: Let $\Pr[A_{f,f'} \mid \sigma_1, b_1, \ldots, \sigma_r, b_r]$ be defined as the probability of bad events conditioned on $\{(\sigma_k, b_k)\}_{k=1}^{r}$. Given $h_r(f, f', \sigma_1, b_1, \cdots, \sigma_r, b_r)$, we choose σ_{r+1}, b_{r+1} to satisfy below equations

$$\begin{aligned} h_{r+1}(f, f'; \sigma_1, b_1, \cdots, \sigma_{r+1}, b_{r+1}) &\geq \Pr[A_{f,f'} \mid \sigma_1, b_1, \ldots, \sigma_{r+1}, b_{r+1}] \\ h_{r+1}(f, f'; \sigma_1, b_1, \cdots, \sigma_{r+1}, b_{r+1}) &\leq h_r(f, f', \sigma_1, b_1, \cdots, \sigma_r, b_r) \end{aligned}$$

- The procedure ends at $r = d$.

This process outputs $\{(\sigma_r, b_r)\}_{r=1}^{d}$ such that $\sum_{f,f' \in \mathcal{F}: f \neq f'} \Pr[A_{f,f'} \mid \sigma_1, b_1, \ldots, \sigma_d, b_d] < 1$. Since $A_{f,f'} \mid \sigma_1, b_1, \ldots, \sigma_d, b_d$ is a determined event, this means the probability of bad events is zero. Using this specific hashing tuple sequence, we can safely hash the possible active frequency points.

(C, ξ)-noise Similar to controlling the effect of bad events, [PS15, JLS23] use the randomness of their sample technique to de-noise. However, it is hard to tackle the noise when we apply a deterministic algorithm, for we can only take samples at finite points. For example, if $g(t)$ is extremely large at our fixed sample points compared to its integral, then it will disturb our observation and prevent the active frequency from being detected. Therefore, we need to introduce extra assumptions on the upper bound of the noise function. Our assumption consists of the energy bound and a $g(t)$-dependent factor ξ, which describes the suitability of $g(t)$ for a deterministic algorithm. See the definition below.

Definition 2.4 ((C, ξ)-noise)**.** *Let $g(t) : [0, T] \to \mathbb{R}$ be the noise function. Let C be some fixed constant. Let ξ be a parameter depending on $g(t)$. Then we say $g(t)$ is a (C, ξ)-noise if it satisfies*

$$\max_{t \in [0,T]} |g(t)|^2 \leq C \cdot \frac{1}{T} \int_0^T |g(t)|^2 \mathrm{d}t + \xi$$

Deterministic HashToBins under continuous setting HashToBins is a commonly-used algorithm in FFT. It takes discrete samples from the time interval and uses DFT to measure the signal in each hashing bucket. This measurement reflects the tone of active frequency in the corresponding bucket. [Kap17] applies a modified HashToBins algorithm to recursively determine

the active tones in the discrete setting. Our work extends this algorithm to the continuous setting. For notation simplicity, we define a vector $v_f \in \mathbb{C}^{|\mathcal{F}|}$ as below to represent the k-sparse tones of signal $\widehat{x}^*(f)$.

Notation 2.5. *Consider $\{f_i\}_{i=1}^{|\mathcal{F}|}, f_i \in \mathcal{F}$ as a finite sequence of points ordered by their magnitude in the frequency interval. Recall $\widehat{x}^*(f) = \sum_{j=1}^{k} v_j \cdot \delta_{f_j}(f)$. Let $v_f := v_i$ if f_i is an active frequency, otherwise, $v_f := 0$.*

Given an initial guess $\widehat{z} \in \mathbb{C}^{|\mathcal{F}|}$ and a discrete sample of signal $x(t)$, this algorithm is able to return the bucket-wise measurement of the difference between v and $\widehat{z}$. The guarantee of this procedure is stated as follows.

Lemma 2.6 (HashToBins (Informal version of Lemma 5.5))**.** *We use B to denote the number of buckets. We use $h_{\sigma,b}(f)$ to denote the index of the bucket where f is hashed into. Given a vector $\widehat{z} \in \mathbb{C}^{|\mathcal{F}|}$, there exists a deterministic procedure* HashToBins *which computes $u \in \mathbb{C}^B$ with the following guarantee: for any $f \in \mathcal{F}$,*

$$|u_{h_{\sigma,b}(f)} - \sum_{f' \in \mathcal{F}} \widehat{G}_{o_{f,\sigma,b}(f')}(v_{f'} - \widehat{z}_{f'})\omega^{a\sigma f'}| \leq O(\frac{\log k}{k} \cdot (\frac{C}{T}\int_0^T |g(t)|^2 \mathrm{d}t + \xi)^{\frac{1}{2}})),$$

The algorithm takes $O(B\log(B))$ samples. The time complexity of the algorithm is $O(B\log^2(B) + B \cdot \log(F/\eta)$.

The analysis of this lemma is by combining the bound for noiseless input (HashToBins$(x^*(t), \widehat{z}_f)$) and noise-only input (HashToBins$(g(t), \mathbf{0}_{|\mathcal{F}|})$), where the former has the similar performance as discrete setting, and the latter is controlled by our assumption in (C, ξ)-noise. Hence, our upper bound on error is the summation of $\|\widehat{z}\|_2 \cdot k^{-c}$ and the energy-ξ bound.

As we mentioned before, the de-randomization step finds a deterministic sequence of hashing parameters that avoids the bad events. Taking the median of output of HashToBins$(x, \widehat{z}, (\sigma_r, b_r))$ for $r \in [d]$ gives a close approximation of $v_f - \widehat{z}_f$.

Lemma 2.7 (HashToBins with de-randomized hash sequence (Informal version of Lemma 5.7))**.** *Let $\widehat{w}_f := v_f - \widehat{z}_f$. Let $\{H_r\}_{r\in[d]} = (\sigma_r, b_r)$ be the sequence of hashing found by the de-randomization process. Let u_r be the output of* HashToBins$(x, \widehat{z}, H_r)$ *. Then, we have, for all $f \in \mathcal{F}$*

$$|\widehat{w}_f - \underset{r\in[d]}{\text{median}}\, \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)}(u_r)_{h_r(f)}| \leq \frac{1}{\alpha k}(\sum_{f\in\mathcal{F}} |\widehat{w}_f| + O(\frac{\log k}{k} \cdot (\frac{C}{T}\int_0^T |g(t)|^2 \mathrm{d}t + \xi)^{\frac{1}{2}})) := \mathcal{N}(\widehat{w})$$

A Recursive sparse recovery algorithm Last, we construct our main algorithm by embedding HashToBins into a recursive sparse recovery algorithm, which echoes with the idea in [IK14] and [LN20]. In each iteration, we use HashToBins to measure the difference between approximated tone $\widehat{z}_f$ and real tone v_f. Then, we use a threshold ν to determine whether to change the approximated tone. We set the recovery threshold $\nu = O(\mathcal{N})$ at the initial stage, and scale by a constant γ in each iteration. Since the detection is applied on each entry of v, this algorithm gives us an error bound in ℓ_∞ norm.

Similar as [HIKP12b] and [LN20], we introduce the definition of signal-to-noise ratio R^*. It measures the ratio between the magnitude of each tone and the average of noise. We assume that $R^* = O(\text{poly}(F/\eta)$, which allows us to run only $O(\log(F/\eta))$ interations of one-stage sparse recovery.

Definition 2.8 (Signal-to-Noise Ratio)**.** *We define the average of noise ν as*

$$\mu := O(C \cdot \frac{1}{kT} \int_0^T |g(t)|^2 \mathrm{d}t + \xi))$$

Then, the signal-to-noise ratio R^* is defined as

$$R^* := \|v\|_\infty / \mu$$

Based on all the above discussions, we obtain the following main theorem:

Theorem 2.9 (Main theorem)**.** *Suppose that $\widehat{x}^*(f)$ is a k-Fourier-sparse signal with its SNR satisfying $R^* \leq (F/\eta)^m$ for some constant parameter m. Let $g(t)$ be a (C, ξ)-noise. Then, there exists an deterministic algorithm which finds an $O(k)$-sparse vector $\widehat{x}' \in \mathbb{C}^{|\mathcal{F}|}$ satisfying $|v_f - \widehat{x}'_f| \leq O(\mathcal{N})$ for all $f \in \mathcal{F}$, where $\mathcal{N} := \frac{1}{\alpha k}(\sum_{f \in \mathcal{F}} |v_f| + O(\frac{\log k}{k} \cdot (\frac{C}{T} \int_0^T |g(t)|^2 \mathrm{d}t + \xi)^{\frac{1}{2}}))$. The algorithm takes $O(k^2 \log k \cdot \log(F/\eta))$ samples and runs in $O((F/\eta) k \log^2(F/\eta))$ time.*

3 Preliminaries

In this section, we introduce some basic definitions and tools in sparse Fourier transforms.

3.1 Fourier transform

We define the discrete Fourier transform (DFT) and continuous Fourier transform (CFT) below.

Definition 3.1 (Discrete Fourier transform)**.** *Given complex vector $x \in \mathbb{C}^n$, we have following definitions*

- *F is said to be the Discrete Fourier transform matrix if $F_{i,j} := \frac{1}{\sqrt{n}} e^{-2\pi \mathbf{i} \cdot ij/n}$*
- *We define the Discrete Fourier transform of x to be $\widehat{x} = Fx$*

Definition 3.2 (Continuous Fourier Transform)**.** *Given function $x(t) : [0, T] \in \mathbb{C}$ and $\widehat{x}(f) : [-F, F] \to \mathbb{C}$. We have the definition of Continuous Fourier Transform and Continuous Inverse Fourier Transform as follows*

$$\widehat{x}(f) = \int_{-\infty}^{\infty} x(\tau) e^{-2\pi \mathbf{i} f \tau} \mathrm{d}\tau \qquad \text{and} \qquad x(t) = \int_{-\infty}^{\infty} \widehat{x}(\sigma) e^{2\pi \mathbf{i} \sigma t} \mathrm{d}\sigma$$

3.2 Convolution

We define the discrete and continuous convolution as below.

Definition 3.3. *For two functions f, g with same domain $\mathcal{D}$, we have*

$$(f * g)(t) = \int_{\tau \in \mathcal{D}} f(t - \tau) \cdot g(\tau) \tau$$

For two vectors f, g with same length n, we have

$$(f * g)[i] = \sum_{f \in [n]} f_{i-j} \cdot g_j$$

3.3 Fourier-sparse signal and noise model

In this article, we discretize the continuous setting by assuming that active frequency of signal $x(t)$ can only be taken in a finite set $\mathcal{F}$ defined as below.

Definition 3.4 (Possible range of active frequency)**.** *In the continuous setting, for a given frequency gap η and bounded range $[-F, F]$, we use $\mathcal{F}$ to denote the set*

$$\{i \cdot \eta \mid \forall i \in \mathbb{Z}, \text{ and } i \cdot \eta \in [-F, F]\}.$$

Definition 3.5 (Continuous-time, k-Fourier-sparse signal)**.** *Let $k \in \mathbb{Z}_{>0}$. Let $\delta_{f_i}(f)$ denote the Dirac function centered at f_i. We define the k-Fourier-sparse signal $\widehat{x}^*(f)$ to be as follows:*

$$\widehat{x}^*(f) = \sum_{j=1}^{k} v_j \cdot \delta_{f_i}(f)$$

where $v_j \in \mathbb{C}$ represents each tone and $f_j \in \mathcal{F}$ for each $j \in [k]$. We use $\mathcal{K}$ to denote all such f_js. For notation simplicity, we define a $\widehat{x}^*$ inspired discrete vector $v \in \mathbb{C}^{|\mathcal{F}|}$. Consider $\{f_i\}_{i=1}^{|\mathcal{F}|}, f_i \in \mathcal{F}$ as a finite sequence of points ordered by their magnitude in the frequency interval. Let $v_f := v_i$ if f_i is an active frequency, otherwise, $v_f := 0$.

Remark 3.6. *By our definition of CFT in Definition 3.2, the inverse CFT of $\widehat{x}^*(f)$ is*

$$x^*(t) = \sum_{j=1}^{k} v_j \cdot e^{2\pi \mathbf{i} f_j t}$$

Then, we define the model of noisy observations, which is commonly used in the literature (e.g. [PS15, CKPS16, SSWZ23]).

Definition 3.7 (Noisy observation)**.** *We define the observed signal $x(t)$ as follows:*

$$x(t) = x^*(t) + g(t) = \sum_{j=1}^{k} v_j e^{2\pi \mathbf{i} f_j t} + g(t),$$

where $g(t)$ is an arbitrary function.

3.4 Hash functions

In this section, we introduce some hash functions used in the sparse Fourier transform algorithms.

Definition 3.8 (Hashing functions, Definition 4.1 in page 9 in [LN20], Section 3 in page 4 in [HIKP12a] Definition A.5, A.6, A.7 in page 17 of [PS15])**.** *Let $\sigma \in \mathbb{R}$ and $b \in [-F, F]$.*

- *We define function $\pi_{\sigma,b} : \mathcal{F} \to [0, 1]$ to be $\pi_{\sigma,b}(f) = \sigma(f - b) \pmod 1$*
- *We define function $h_{\sigma,b} : \mathcal{F} \to [B]$ to be $h_{\sigma,b}(f) = \mathrm{round}(B \cdot \pi_{\sigma,b}(f))$*
- *Fix $f \in \mathcal{F}$, we define function $o_{f,\sigma,b} : \mathcal{F} \to [0, 1]$ to be $o_{f,\sigma,b}(f') = \pi_{\sigma,b}(f') - (1/B)h_{\sigma,b}(f)$*

Definition 3.9 (Pseudorandom permutation, Definition 4.2 in page 10 in [LN20], Definition A.1 in page 15 in [PS15])**.** *For $\sigma \in \mathbb{R}, a \in [0, T], b \in [-F, F]$, the permutation $P_{\sigma,a,b}$ is defined as*

$$(P_{\sigma,a,b}x)(t) = x(\sigma(t - a)) \cdot \omega^{t\sigma b}$$

Definition 3.10 (Sequence of Hashings, Definition 4.4 in page 10 in [LN20]). *We use* $\{(\sigma_r, a_r, b_r)\}_{r\in[d]}$ *to denote the parameters of a sequence of* d *hashings. Each* (σ_r, a_r, b_r) *specifies three hashing functions:* $\pi_{\sigma_r,b_r}, h_{\sigma_r,b_r}, o_{f,\sigma_r,b_r}$*, and one pseudo permutation:* P_{σ_r,a_r,b_r}.

Definition 3.11 (Tuple of Hashing). *We use* $H = (\sigma, a, b)$ *to denote a tuple of hashing. In a sequence of hashings* $\{(\sigma_r, a_r, b_r)\}_{r\in[d]}$*, we use* H_r *to represent* (σ_r, a_r, b_r).

3.5 Filter function

Filtering is one of the most important techniques for sparse Fourier transform, which allows us to isolate each individual frequency and reduces the k-sparse signal to a set of "almost" one-sparse signals.

Definition 3.12 (Filter function in the continuous setting, Definition A.3 in page 15 in [PS15], see discrete variations in Definition 4.6 in page 10 in [LN20], Definition 2.3 in page 6 in [Kap17], Definition 2.3 in page 5 in [HIKP12a]). *Let* $B \in \mathbb{Z}_{>0}$ *be a power of two. Let* N *be some fixed integer. Let offset* o *be defined as Definition 3.8. We say* $\widehat{G} : [0,1] \to \mathbb{R}$*, with* G *being its Fourier transform, is a* flat filter *with* B *buckets, sharpness* ϵ *if the followings hold:*

- *Property 1:* $\widehat{G}_{o_{f,\sigma_r,b_r}} \in [0,1]$ *for all* $o_{f,\sigma_r,b_r} \in [0,1]$
- *Property 2:* $\widehat{G}_{o_{f,\sigma,b}} \geq 1-\epsilon$ *for all* $o_{f,\sigma,b} \in [-\frac{1}{2B}, \frac{1}{2B}]$
- *Property 3:* $\widehat{G}_{o_{f,\sigma,b}} \leq \epsilon$ *for all* $o_{f,\sigma,b} \in [0,1] \backslash [-\frac{1}{B}, \frac{1}{B}]$
- *Property 4:* $\sum_{i\in\mathbb{Z}} G(i)^2 = O(\frac{1}{B})$
- *Property 5:* $\mathrm{supp}(G) \subset [-D, D]$ *where* $D = O(\log(B))$ *and* D *is an integer*

Remark 3.13. *The construction of* $(G, \widehat{G})$ *can be found in [PS15] and [JLS23]. Notice that* $\widehat{G} : [0, 2\pi] \to \mathbb{R}$ *in their construction, we can simply extend to our setting by scaling.*

3.6 Measurement

Definition 3.14 (Measurement without noise, implicitly in Lemma 3.4 in page 23 in [PS15], see discrete variation in Definition 4.8 in page 10 in [LN20]). *Suppose the following conditions hold:*

- *Let* $\widehat{x}$*,* v *be defined as Definition 3.5*
- *Let* $H = (\sigma, a, b)$ *be a tuple of hashing*
- *Let* $\widehat{G}$ *be a flat filter with* B *buckets and sharpness* ϵ *(refer to Definition 3.12).*

A measurement $m_H(\widehat{x}(f)) \in \mathbb{C}^B$ *is defined as*

$$(m_H(\widehat{x}(f)))_{h_{\sigma,b}(f)} = \sum_{f\in\mathcal{F}} \widehat{G}_{o_{f,\sigma,b}(f')} \cdot \omega^{a\sigma f} \cdot v_f \quad \forall h_{\sigma,b}(f) \in [B],$$

where π *is a hash function induced from* H.

Claim 3.15. *Under the conditions of Definition 3.14, we have*

$$\begin{aligned} &(\widehat{G}_{o_{f,\sigma,b}(f)})^{-1} \cdot (m_H(\widehat{x}(f)))_{h_{\sigma,b}(f)} \cdot \omega^{-a\sigma f} \\ =\ & v_f + (\widehat{G}_{o_{f,\sigma,b}(f)})^{-1} \cdot \sum_{f'\in\mathcal{F}\backslash\{f\}} \widehat{G}_{o_{f,\sigma,b}(f')} \cdot v_{f'} \cdot \omega^{a\sigma(f'-f)}. \end{aligned}$$

Proof. By the definition of m_H (Definition 3.14), we have

$$\begin{aligned}
&(\widehat{G}_{o_{f,\sigma,b}(f)})^{-1} \cdot (m_H)_{h_{\sigma,b}(f)} \cdot \omega^{-a\sigma f} \\
&= (\widehat{G}_{o_{f,\sigma,b}(f)})^{-1} \cdot \Big(\sum_{f' \in \mathcal{F}} \widehat{G}_{o_{f,\sigma,b}(f')} \cdot \omega^{a\sigma f'} \cdot v_{f'} \Big) \cdot \omega^{-a\sigma f} \\
&= v_f + (\widehat{G}_{o_{f,\sigma,b}(f)})^{-1} \cdot \sum_{f' \in \mathcal{F} \setminus \{f\}} \widehat{G}_{o_{f,\sigma,b}(f')} v_{f'} \omega^{a\sigma(f'-f)}
\end{aligned}$$

□

3.7 Signal-to-noise ratio

Definition 3.16 (Signal-to Noise Ratio, Definition in page 12 in [LN20], defined as ρ in Lemma 3.6 in page 28 in [PS15])**.** *We define the average of noise μ as*

$$\mu := O(C \cdot \frac{1}{kT} \int_0^T |g(t)|^2 \mathrm{d}t + \xi))$$

Then, the signal-to-noise ratio R^* is defined as

$$R^* := \|v\|_\infty / \mu$$

4 De-randomization

In this section, our goal is to find a sequence of hashing $\{(\sigma_r, a_r, b_r)\}_{r \in [d]}$ satisfying the following condition (see Lemma 4.21):

$$\sum_{r \in [d]} (\widehat{G}_{o_{f,\sigma_r,b_r}(f)})^{-1} \widehat{G}_{o_{f,\sigma_r,b_r}(f')} \leq \frac{\beta}{1-\epsilon}.$$

This condition will be used to derive a guarantee for algorithm SubRecovery (i.e., the fifth condition in Lemma 5.2).

4.1 Parameter constraints

This section lists our choice for several parameters used in this section.

Definition 4.1. *We define the following parameters:*

- $\lambda := C_1$ *where* C_1 *is some fixed constant belongs to* $(\frac{1}{2}, 1)$
- $\epsilon := \frac{20}{B}$
- $\beta := \frac{6}{C_1} \cdot \log |\mathcal{F}|$
- $d := \frac{3C_1}{40} \cdot B \log |\mathcal{F}|$

The parameters in Definition 4.1 are chosen to satisfy below inequalities, which we will use in the proofs of this section.

Observation 4.2. *We assume the parameters satisfy the following conditions:*

1. $\lambda \in (0,1)$
2. $\epsilon \in (0,1)$
3. $\beta \geq 4\epsilon d$
4. $\lambda\beta \geq 6 \log |\mathcal{F}|$
5. $\frac{10}{B} \cdot (\lambda(1-\epsilon)+1) \leq \lambda\epsilon$

Proof. **Proof of Part 5**

$$\begin{aligned}\frac{10}{B} \cdot (\lambda(1-\epsilon)+1) &\leq \frac{10}{B} \cdot (\lambda+1) \\ &\leq \frac{\lambda\epsilon}{2} + \frac{10}{B} \\ &\leq \lambda\epsilon\end{aligned}$$

the first step is by $B > 20$, the 2nd step is by the choice of ϵ, the 3rd step is by $\lambda > 1/2$ □

4.2 Sample Range Set

This section defines the set where we sample the hashing parameters σ and b.

Definition 4.3 (Sample range of b)**.** *Let $\mathcal{B}$ be the range of the hashing parameter b. We take $\mathcal{B} := \mathcal{F}$.*

Definition 4.4 (Sample range of σ)**.** *Let Σ be the range of the hashing parameter σ. We define S to be the set of positive odd integers in $[0, F/\eta]$, and $\Sigma := \frac{1}{F} \cdot S = \{x/F \mid x \in S\}$.*

4.3 Bad Event

Definition 4.5 (Bad events, Definition 5.4 on page 14 in [LN20])**.** *Suppose that*

- *d is defined as Definition 4.1, and*
- *β is defined as Observation 4.2.*

For any $f, f' \in \mathcal{F}$, $f \neq f'$, we define $A_{f,f'}$ to be the event that

$$\sum_{r=1}^{d} \widehat{G}_{o_{f,\sigma_r,b_r}(f')} \geq \beta.$$

4.4 Pessimistic Estimator

Definition 4.6 (Pessimistic estimator, Definition 5.5 in page 15 in [LN20])**.** *Let $\lambda > 0$ be a fixed parameter. We define the* pessimistic estimator *as follows:*

$$h_r(f, f'; \sigma_1, b_1, \cdots, \sigma_r, b_r) := \exp(-\lambda\beta) \cdot \exp\left(\lambda \sum_{l=1}^{r} \widehat{G}_{o_{f,\sigma_l,b_l}(f')}\right) \cdot (M(\lambda))^{d-r},$$

where

$$M(\lambda) := \exp(\lambda\epsilon) \cdot \left(\frac{5}{B} \cdot e^{\lambda(1-\epsilon)} + 1\right).$$

For $r \geq 1$, the value of σ_r, b_r is determined by the following minimization procedure:

$$\sigma_r, b_r = \arg \min_{\sigma \in \Sigma, b \in \mathcal{B}} \sum_{f,f' \in \mathcal{F}: f \neq f'} h_r(f, f'; \sigma_1, b_1, \cdots, \sigma_{r-1}, b_{r-1}, \sigma, b).$$

This function can be evaluated in $\mathrm{poly}(F/\eta)$ *time.*

Remark 4.7. *[LN20] chose d to be $O(k \log n)$. In this work, we choose d to be $O(k \log(F/\eta))$.*

4.5 A Round and Mod Tool

Fact 4.8 (Change order in taking modulus)**.** *For any positive integer y and real number x, it holds that*

$$(y^{-1} \cdot \mathrm{round}(yx)) \mod 1 = y^{-1} \cdot \mathrm{round}(y \cdot (x \mod 1)) - c,$$

where $c = 0$ or 1.

Proof. We assume $x = x' + q$ where x is an integer and $q \in [0, 1)$, then we have

$$\begin{aligned}(y^{-1} \cdot \mathrm{round}(yx)) \mod 1 &= (y^{-1} \cdot \mathrm{round}(yx' + yq)) \mod 1 \\ &= (y^{-1} \cdot (yx' + \mathrm{round}(yq))) \mod 1 \\ &= (y^{-1} \cdot \mathrm{round}(yq)) \mod 1\end{aligned}$$

where the first step is because $x = x' + q$, the 2nd step is because yx' is an integer, the 3rd step is because $x' \mod 1 = 0$. Then, we consider two cases of q:

Case 1: $q \in [0, 1 - 1/(2y))$. Then, $yq < y - 1/2$, and $\mathrm{round}(yq) \leq y - 1$. Thus, $y^{-1} \cdot \mathrm{round}(yq) \in [0, 1)$. In this case,

$$\begin{aligned}\mathrm{LSH} &= (y^{-1} \cdot \mathrm{round}(yq)) \mod 1 \\ &= (y^{-1} \cdot \mathrm{round}(yq)) \\ &= y^{-1} \cdot \mathrm{round}(y \cdot (x \mod 1)),\end{aligned}$$

where the last step follows from $q = x \mod 1$. And the fact is proved with $c = 0$ in this case.

Case 2: $q \in [1 - 1/(2y), 1)$. Then, $yq \in [y - 1/2, y)$, and $\mathrm{round}(yq) = y$. Then, $y^{-1} \cdot \mathrm{round}(yq) = 1$. In this case,

$$\mathrm{LSH} = (y^{-1} \cdot \mathrm{round}(yq)) \mod 1 = 0.$$

On the other hand,

$$\mathrm{RHS} = y^{-1} \cdot \mathrm{round}(y \cdot (x \mod 1)) = (y^{-1} \cdot \mathrm{round}(yq)) = 1.$$

Hence, the fact follows with $c = 1$ in this case.

As Case 1 and Case 2 consider all possible values of q, the fact is then proved. □

4.6 Reformulation of $o_{f,\sigma,b}(f')$

Lemma 4.9 (Simplified $o_{f,\sigma,b}$)**.** *Let $\pi_{\sigma,b}(f)$, $o_{f,\sigma,b}(f')$ be defined as Definition 3.8. We have*

$$o_{f,\sigma,b}(f') = \big(\sigma(f'-f) + \sigma(f-b) - \frac{1}{B}\mathrm{round}(B\sigma(f-b)) \mod 1\big) - c,$$

for some $c = 0$ or 1.

Proof.

$$\begin{aligned}
o_{f,\sigma,b}(f') &= \pi_{\sigma,b}(f') - (1/B)h_{\sigma,b}(f) \mod 1 \\
&= \sigma(f'-b) \mod 1 - \frac{1}{B}\mathrm{round}(B(\sigma(f-b) \mod 1)) \\
&= \sigma(f'-b) \mod 1 - \big(\Big(\frac{1}{B}\mathrm{round}(B\sigma(f-b))\Big) \mod 1\big) - c \\
&= \big(\Big(\sigma(f'-b) - \frac{1}{B}\mathrm{round}(B\sigma(f-b))\Big) \mod 1\big) - c \\
&= \big(\Big(\sigma(f'-f) + \sigma(f-b) - \frac{1}{B}\mathrm{round}(B\sigma(f-b))\Big) \mod 1\big) - c,
\end{aligned}$$

where the first step and second step follows by Definition 3.8, the 3rd step uses Fact 4.8 by taking $y = B$ and $x = \sigma(f-b)$, and $c = 0$ or 1, the 4th step holds by property of taking modulus, the last step is a rearrangement.

□

Notice that, the $-c$ will not affect the result showing below since $\sigma(f-f') \mod 1 \in [0,1)$, so we ignore it.

4.7 Upper bound on $Z_\sigma(b)$

Definition 4.10. *Suppose b is uniformly sampled from the set $\mathcal{B}$ (Definition 4.3), and σ is uniformly sampled from the set Σ (Definition 4.4). Define*

$$Z_\sigma(b) := \sigma(f-b) - \frac{1}{B}\mathrm{round}(B\sigma(f-b)).$$

Lemma 4.11 (The range of $Z_\sigma(b)$)**.** *Let $Z_\sigma(b)$ be defined as in Definition 4.10. Then, it always holds that*

$$Z_\sigma(b) \in [-\frac{1}{2B}, \frac{1}{2B}].$$

Proof.

$$\begin{aligned}
|Z_\sigma(b)| &= |\sigma(f-b) - \frac{1}{B}\mathrm{round}(B\sigma(f-b))| \\
&= \frac{1}{B} \cdot |B\sigma(f-b) - \mathrm{round}(B\sigma(f-b))| \\
&\le \frac{1}{2B}
\end{aligned}$$

where the first step is by definition of $Z_\sigma(b)$ (see Definition 4.10), the 2nd step is from simple calculation, the 3rd step holds because $|x - \mathrm{round}(x)| \le 1/2$ for all $x \in \mathbb{R}$.

□

4.8 Distribution of $\sigma(f' - f) \pmod 1$

Lemma 4.12 (Reformulation of $\sigma(f' - f)$)**.** *Under the following conditions:*

- *Let σ be uniformly random from set Σ (see Definition 4.4)*
- *Let $m := \sigma F$*
- *Let $i := (f' - f)/\eta$*

then m is uniformly distributed over S (see Definition 4.4), and

$$(\sigma(f' - f)) \mod 1 = \frac{\eta}{F} \cdot (mi \mod \frac{F}{\eta}). \tag{1}$$

Proof. By Definition 4.4), we know that $\Sigma = \frac{1}{F} \cdot S$, where S contains all the odd numbers on $[F/\eta]$. Since σ is uniformly sampled from Σ and $m = \sigma F$, we get that m is uniformly distributed in S.

By the definitions of m and i, we have

$$\sigma(f' - f) = \frac{m}{F} \cdot i\eta = mi \cdot \frac{\eta}{F}.$$

Now suppose $\sigma(f' - f) = C + D$, where $C \in \mathbb{Z}$, $D \in [0, 1)$. Then, we have

$$mi \cdot \frac{\eta}{F} = C + D, \quad \text{i.e.,} \quad mi = C\frac{F}{\eta} + D\frac{F}{\eta}.$$

Since $D\frac{F}{\eta} \in [0, \frac{F}{\eta})$, it implies that

$$mi \mod \frac{F}{\eta} = D\frac{F}{\eta}.$$

Now, we can conclude that:

$$(\sigma(f' - f)) \mod 1 = D = \frac{\eta}{F} \cdot (mi \mod \frac{F}{\eta}).$$

□

Lemma 4.13 (Distribution of $(\sigma(f' - f) \mod 1)$)**.** *Under the following conditions,*

- *Suppose $F/\eta := 2^p$, where p is a positive integer*
- *Let σ be uniformly random from set Σ (see Definition 4.4)*
- *Let $f' \neq f$ be frequencies from $\mathcal{F}$*
- *Let m, i be defined as in Lemma 4.12*
- *Let $i := 2^s K$ where s is a non-negative integer and K is an odd number*

then we have

1. *$(\sigma(f' - f) \mod 1)$ is uniformly distributed over its support*
2. *The support of $(\sigma(f' - f) \mod 1)$ is symmetric*

3. *The support of* $(\sigma(f'-f) \mod 1)$ *is a sequence of equidistant points, with wraparound distance* $D := \frac{\eta}{F} 2^{s+1}$

In particular, $(\sigma(f'-f) \mod 1)$ *is uniformly distributed over the following set:*

$$\left\{ \pm(\frac{1}{2} + j) \cdot D : 0 \le j \le 2^{p-s-2} - 1, j \in \mathbb{N} \right\}.$$

Proof. By Lemma 4.12, $(\sigma(f'-f) \mod 1) = 2^{-p} \cdot (mi \mod 2^p)$. Hence, we only need to consider the distribution of

$$mi \pmod{2^p},$$

where m is uniformly sampled from odd integers in $[2^p]$ and $i = 2^s K \in [2^p]$.

If $i = 2^p$, then $\mathrm{supp}(\sigma(f'-f) \mod 1) = \{0\}$. And the lemma trivially holds. In the following proof, we assume that $i < 2^p$, i.e., $0 \le s < p$.

Proof of Part 1

Let m_1, m_2 be two possible value of m, when $m_1 - m_2 = 2^{p-s}$, we have

$$m_1 i - m_2 i = 2^p K \equiv 0 \pmod{2^p}$$

Hence, the value of $(mi \mod 2^p)$ has a period of length at least 2^{p-s}. And we only need to consider $m \in [2^{p-s}]$.

Let $m_1, m_2 \in [2^{p-s}]$, $m_1 > m_2$. Then

$$(m_1 - m_2) 2^s < 2^p$$

Since K is odd, then $(m_1 - m_2)i = (m_1 - m_2)2^s K$ cannot be divided by 2^p, which imples that

$$m_1 i \pmod{2^p} \neq m_2 i \pmod{2^p}$$

Therefore, each odd integer $m \in [2^{p-s}]$ generates a unique value for $mi \pmod{2^p}$. And $mi = m'i \pmod{2^p}$ for every $m' \in \{m + j2^{p-s} \mid m + j2^{p-s} \le 2^p, j \in \mathbb{N}\}$. Suppose that $i < 2^p$, then each m' in this set is an odd number. Moreover, for any odd $m \in [2^{p-s}]$,

$$|\{m + j2^{p-s} \mid m + j2^{p-s} \le 2^p, j \in \mathbb{N}\}| = 2^s.$$

Therefore, S, the support of m, is divided into 2^s-sized equivalence classes, and each class gives a distinct value for $mi \pmod{2^p}$. Since m is uniformly sampled, $mi \pmod{2^p}$ is uniform on its support, so does $\sigma(f'-f) \pmod 1$.

Proof of Part 2

For any $m \in [2^{p-s}]$, we have

$$mi + (2^{p-s} - m)i = 2^p K \equiv 0 \pmod{2^p}.$$

Therefore,

$$(mi \mod 2^p) + ((2^{p-s} - m)i \mod 2^p) = 2^p,$$

that is, the support of $(mi \mod 2^p)$ is symmetric in $[2^p]$. Thus, the support of $(\sigma(f'-f) \mod 1)$ is symmetric in $[0, 1]$ with respect to 0 under wraparound distance.

Proof of Part 3

For any $m_1 > m_2 \in [2^{p-s}]$, $m_1 - m_2$ can be written as $2g$ for some integer $g \neq 0$ since they are all odd numbers. We have

$$(m_1 - m_2)i = 2^{s+1}gK = C \cdot 2^p + D$$

where C is an integer and $D \in (0, 2^p)$. Then, we know that D can be divided by 2^{s+1} since $2^{s+1}gK$ and $C \cdot 2^p$ both can be divided by 2^{s+1}.

We show that D is the wraparound distance between any two points in the support of ($mi \mod 2^p$):

$$\begin{aligned}
&((m_1 i \pmod{2^p}) - (m_2 i \pmod{2^p}) \mod 2^p) \\
&= (m_1 - m_2)i \mod 2^p \\
&= 2^{s+1}gK \mod 2^p \\
&= D,
\end{aligned}$$

where the first step is by the property of taking modulus, the 2nd step is because $m_1 - m_2 = 2g$, and the 3rd step follows from the definition of D. Then, we know that the wraparound distance between two points is at least 2^{s+1}.

On the other hand, we show that the distance is at most 2^{s+1}. By **Part 1** of this lemma, we know that

$$|\{mi \pmod{2^p} : m \in [2^p],\ m \text{ is odd}\}| = |\{m \in [2^{p-s}],\ m \text{ is odd}\}| = 2^{p-s-1}.$$

Thus,

$$D \cdot 2^{p-s-1} \leq 2^p,$$

which implies that $D \leq 2^{s+1}$.

Therefore, we get that the support of ($mi \mod 2^p$) is a sequence of equidistant points, with wraparound distance 2^{s+1}. By scaling a factor of $\frac{\eta}{F}$, we get the wanted result. □

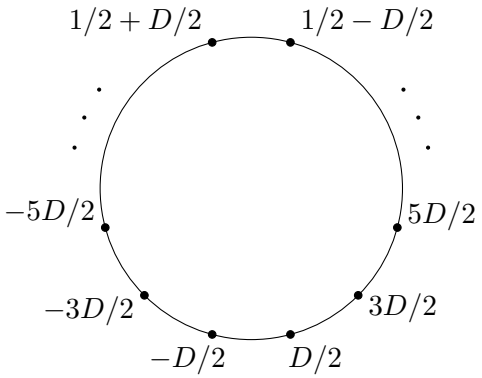

Figure 1: The support of ($\sigma(f' - f) \mod 1$), where $D = \frac{\eta}{F}2^{s+1}$.

4.9 Distribution of $o_{f,\sigma,b}(f')$

Lemma 4.14 (Analogous to Lemma 5.6 in page 15 in [LN20])**.** *If the following conditions hold:*

- *Suppose $F/\eta := 2^p$, where p is a positive integer*
- *Let σ be uniformly random from set Σ (see Definition 4.4)*
- *Let $f' \neq f$ be frequencies from $\mathcal{F}$*
- *Let m, i be defined as in Lemma 4.12*
- *Let $i := 2^s K$ where s is a non-negative integer and K is an odd number*
- *Suppose $B = 2^q$*

then for any $f \neq f'$ we have,

$$\Pr_{\sigma,b}\left[o_{f,\sigma,b}(f') \in \left[-\frac{1}{B}, \frac{1}{B}\right]\right] \leq \frac{5}{B}.$$

Proof. By Lemma 4.9, we know that

$$o_{f,\sigma,b}(f') = \sigma(f' - f) + Z_\sigma(b) \mod 1,$$

where $Z_\sigma(b)$ is a random variable defined by

$$Z_\sigma(b) = \sigma(f - b) - \frac{1}{B}\text{round}(B\sigma(f - b)).$$

Then, the distribution of $o_{f,\sigma,b}(f')$ is a convolution of $Z_\sigma(b) \in [-\frac{1}{2B}, \frac{1}{2B}]$ and $\sigma(f' - f) \mod 1$ (a sequence of equidistant points).

We first consider the random variable σ, and then conditioned on σ, we consider the random variable b.

By Lemma 4.13, the distance between two consecutive points in the support of $\sigma(f'-f) \mod 1$ is $D = \frac{\eta}{F}2^{s+1} = 2^{-p+s+1}$. In the following proof, we discuss two cases based on the value of $p-q-s$.

Case 1: $p - q \geq s$**.** We have

$$\frac{1}{B} = 2^{-q} = D \cdot 2^{p-q-s-1} \geq \frac{D}{2}. \tag{2}$$

Then, we have

$$\begin{aligned}
&\Pr\left[o_{f,\sigma,b}(f') \in \left[-\frac{1}{B}, \frac{1}{B}\right]\right] \\
=\ &\Pr\left[(\sigma(f' - f) + Z_\sigma(b) \mod 1) \in \left[-\frac{1}{B}, \frac{1}{B}\right]\right] \\
=\ &\sum_{x \in \text{supp}(\sigma(f'-f)) \mod 1} \Pr_b\left[(x + Z_\sigma(b) \mod 1) \in \left[-\frac{1}{B}, \frac{1}{B}\right] \,\middle|\, (\sigma(f' - f) \mod 1) = x\right] \cdot \\
&\Pr_\sigma[(\sigma(f' - f) \mod 1) = x] \\
\leq\ &\Pr\left[(\sigma(f' - f) \mod 1) \in \left[-\frac{3}{2B}, \frac{3}{2B}\right]\right]
\end{aligned}$$

$$
\begin{aligned}
&\leq \frac{(1/D)\cdot(3/B)+1}{1/D} \\
&= \frac{3}{B} + D \\
&\leq \frac{5}{B}
\end{aligned}
$$

where the 1st step is by simple algebra, the 2nd step follows from the conditional probability, the 3rd step is because $Z_\sigma(b) \in [-\frac{1}{2B}, \frac{1}{2B}]$ (see Lemma 4.11), the 4th step is because the uniform distribution of $\sigma(f'-f) \mod 1$ on its support described in Lemma 4.13, where the numerator is the maximum number of points inside interval $[-\frac{3}{2B}, \frac{3}{2B}]$, the denominator is the total number of points in the whole range, the 5th step is a rearrangement, and the last step uses Eq. (2).

Case 2: $p-q \leq s-1$. By Lemma 4.13, the closest point to the origin in the support of $\sigma(f'-f) \mod 1$ is $\pm D/2$. Then, in this case, it holds that

$$
\frac{D}{2} = 2^{-p+s} = \frac{2}{B}\cdot 2^{q-1-p+s} \geq \frac{2}{B} > \frac{3}{2B}.
$$

Hence, by the same analysis as in Case 1, $o_{f,\sigma,b}(f')$ will not take value in $[-\frac{1}{B}, \frac{1}{B}]$.

□

Lemma 4.15 (Analogous to Lemma 5.7 in page 16 in [LN20])**.** *Under following conditions*

- *Suppose $|\mathcal{F}|$ is power of 2*
- *Let σ be uniformly random from set Σ (see Definition 4.4)*
- *Let b be uniformly random in set $\mathcal{B}$ (see Definition 4.3)*
- *Suppose B is power of 2*
- *Let $\widehat{G}$ be a flat filter defined in Definition 3.12*
- *Let $M(\lambda)$ be defined as Definition 4.6*

For all $f, f' \in \mathcal{F}$,

$$
\mathop{\mathbb{E}}_{\sigma,b}[\exp(\lambda \widehat{G}_{o_{f,\sigma,b}(f')})] \leq M(\lambda)
$$

Proof. By Lemma 4.14 we have,

$$
\Pr[\widehat{G} \geq \epsilon] = \Pr[o_{f,\sigma,b}(f') \in [-\frac{1}{B}, \frac{1}{B}]] \leq \frac{5}{B} \tag{3}
$$

where the first step is by the definition of G (see Definition 3.12), the second step uses **Part 1** of Lemma 4.14.

Therefore, we have

$$
\begin{aligned}
\mathop{\mathbb{E}}_{\sigma,b}[\exp(\lambda \widehat{G}_{o_{f,\sigma,b}(f')})] \leq\ & \Pr[\widehat{G} \geq \epsilon]\cdot \exp(\lambda \cdot \sup \widehat{G}) + \Pr[\widehat{G} \leq \epsilon]\cdot \exp(\lambda\epsilon) \\
\leq\ & \Pr[\widehat{G} \geq \epsilon]\cdot \exp(\lambda) + \Pr[\widehat{G} \leq \epsilon]\cdot \exp(\lambda\epsilon) \\
\leq\ & \frac{5}{B}\cdot e^{\lambda} + e^{\lambda\epsilon}
\end{aligned}
$$

$$= M(\lambda)$$

where the first step follows from the definition of expectation, the second step follows by $\widehat{G} \in [0,1]$ (see Definition 3.12), the 3rd step is by Eq. (3), and the last step follows from the definition of $M(\lambda)$ (see Definition 4.6).

□

4.10 Range of $o_{f,\sigma,b}(f)$

Lemma 4.16. *If the following conditions hold:*

- *Let σ be randomly chose from set Σ (see Definition 4.4)*
- *Let b be uniformly random from set β (see Definition 5.6)*

then for any $f \in \mathcal{F}$ we have,

$$o_{f,\sigma,b}(f) \in [-\frac{1}{2B}, \frac{1}{2B}]$$

Proof.

$$\begin{aligned} o_{f,\sigma,b}(f) &= \sigma(f - f) + Z_\sigma(b) \\ &= Z_\sigma(b) \\ &\in [-\frac{1}{2B}, \frac{1}{2B}] \end{aligned}$$

where the first step is from Lemma 4.9, the 3rd step is by Lemma 4.11 □

4.11 Upper bound for probability of bad event

Lemma 4.17 (Pessimistic Estimator, Analogous to Lemma 5.8 in page 16 in [LN20])**.** *Under following conditions*

- *Suppose $|\mathcal{F}|$ is power of 2*
- *Let $r \in [d]$*
- *Let σ be uniformly random from set Σ (see Definition 4.4)*
- *Let b be uniformly random in set $\mathcal{B}$ (see Definition 4.3)*
- *Let h_r be defined as Definition 4.6*
- *Let $H_r = \{(\sigma_r, a_r, b_r)\}_{r\in[d]}$ be a sequence of hashing chose by procedure in Definition 4.6*
- *Suppose $f, f' \in \mathcal{F}$ satisfy $f \neq f'$*
- *Let $A_{f,f'}$ denote the bad event defined as Definition 4.5, where $(\sigma_{r+1}, b_{r+1}), \ldots, (\sigma_d, b_d)$ are uniformly and independently sampled from $\Sigma \times \mathcal{B}$*

we have,

$$h_r(f, f'; \sigma_1, b_1, \cdots, \sigma_r, b_r) \geq \Pr[A_{f,f'} \mid \sigma_1, b_1, \ldots, \sigma_r, b_r].$$

Proof. We define z as follows,

$$z := \sum_{l=1}^{r} \widehat{G}_{o_{f,\sigma_l,b_l}(f')}$$

Conditioned on $\sigma_1, b_1, \ldots, \sigma_r, b_r$, z is a fixed constant.

Then we have

$$\begin{aligned}
\Pr[A_{f,f'} \mid \sigma_1, b_1, \cdots, \sigma_r, b_r] &= \Pr[z + \sum_{l=r+1}^{d} \widehat{G}_{o_{f,\sigma_l,b_l}(f')} > \beta] \\
&= \Pr[\exp(\lambda(z + \sum_{l=r+1}^{d} \widehat{G}_{o_{f,\sigma_l,b_l}(f')})) > e^{\lambda\beta}] \\
&\leq e^{-\lambda\beta} e^{\lambda z} \mathbb{E}[\exp(\lambda \sum_{l=r+1}^{d} \widehat{G}_{o_{f,\sigma_l,b_l}(f')})] \\
&= e^{-\lambda\beta} e^{\lambda z} \mathbb{E}[\exp(\lambda \widehat{G}_{o_{f,\sigma,b}(f')})]^{d-r} \\
&= e^{-\lambda\beta} e^{\lambda z} (M(\lambda))^{d-r}
\end{aligned}$$

where the 3rd step is by Markov inequality, the 4th step follows from the independence, the 5th step is given by Lemma 4.15, and the expression in the last line is exactly h_r (see Definition 4.6). □

4.12 Upper bound for $M(\lambda)$

Lemma 4.18 (Upper bound for $M(\lambda)$)**.** *Under following conditions*

- *Let $M(\lambda)$ be defined as in Definition 4.6*
- *Suppose that $\epsilon \in (0, 1)$*

Then we have,

$$M(\lambda) \leq \exp(2\lambda\epsilon)$$

Proof.

$$\begin{aligned}
M(\lambda) &= \exp(\lambda\epsilon) \cdot (\frac{5}{B} \cdot e^{\lambda(1-\epsilon)} + 1) \\
&\leq \exp\left(\lambda\epsilon + \log(1 + \frac{5}{B} \cdot e^{\lambda(1-\epsilon)})\right) \\
&\leq \exp\left(\lambda\epsilon + \frac{5}{B} \cdot e^{\lambda(1-\epsilon)}\right) \\
&\leq \exp\left(\lambda\epsilon + \frac{10}{B} \cdot (\lambda(1-\epsilon) + 1)\right) \\
&\leq \exp\left(2\lambda\epsilon\right)
\end{aligned}$$

where the 1st step is due to definition of $M(\lambda)$ (see Definition 4.6), the 2nd is by simple algebra, the 3rd step is because $\log(x+1) \leq x$ for $x \neq 0$, the 4th step is because $e^x \leq 2x + 1$ for $x \in (0, 1)$, and $\lambda, \epsilon \in (0, 1)$, the 5th step is by **Part 5** of Observation 4.2

□

4.13 Initial Constraint

Lemma 4.19 (Initial constraint, Analogous to Lemma 5.9 in page 17 in [LN20])**.** *Under following conditions*

- *Suppose* $|\mathcal{F}|$ *is power of 2*
- *Let* σ *be uniformly random from set* Σ *(see Definition 4.4)*
- *Let* b *be uniformly random in set* $\mathcal{B}$ *(see Definition 4.3)*
- *Let* h_r *be defined as Definition 4.6*
- *Suppose* $f, f \in \mathcal{F}$ *satisfy* $f \neq f'$

we have,

$$\sum_{f,f' \in \mathcal{F}: f \neq f'} h_0(f, f') < 1$$

Proof.

$$\begin{aligned}
\sum_{f,f' \in \mathcal{F}: f \neq f'} h_0(f, f') &= e^{-\lambda\beta} \sum_{f,f' \in \mathcal{F}: f \neq f'} (M(\lambda))^d \\
&\leq |\mathcal{F}|^2 \exp(-\lambda\beta + 2\lambda\epsilon d) \\
&\leq |\mathcal{F}|^2 \exp(-0.5\lambda\beta) \\
&\leq |\mathcal{F}|^2 \exp(-3 \log |\mathcal{F}|) \\
&< 1
\end{aligned}$$

where the first step is from definition of h_0 (see Definition 4.6), the second step follows from Lemma 4.18, the third step is by $\beta \geq 4\epsilon d$ (**Part 3** of Observation 4.2), the fourth step follows by $\lambda\beta \geq 6 \log |\mathcal{F}|$) (**Part 4** of Observation 4.2), the fifth step is by simple algebra. □

4.14 Induction step

Lemma 4.20 (Derandomization, Analogous to Lemma 5.10 in page 17 in [LN20])**.** *Under following conditions*

- *Suppose* $|\mathcal{F}|$ *is power of 2*
- *Let* σ *be uniformly random from set* Σ *(see Definition 4.4)*
- *Let* b *be uniformly random in set* $\mathcal{B}$ *(see Definition 4.3)*
- *Let* $r \in [d-1]$
- *Let* $H_r = \{(\sigma_j, a_j, b_j)\}_{j \in [j]}$ *be a sequence of hashing chose by procedure in Definition 4.6*
- *Let* h_r *be defined as Definition 4.6*
- *Suppose* $f, f \ \in [n]$ *satisfy* $f \neq f'$

we have,

$$h_r(f, f'; \sigma_1, b_1, \cdots, \sigma_r, b_r) \geq \mathop{\mathbb{E}}_{\sigma_{r+1}, b_{r+1}} [h_{r+1}(f, f'; \sigma_1, b_1, \cdots, \sigma_{r+1}, b_{r+1})]$$

Proof. We define z as follows,

$$z := \sum_{l=1}^{r} \widehat{G}_{o_{f,\sigma_l,b_l}(f')}$$

Then we have

$$\begin{aligned}
\mathop{\mathbb{E}}_{\sigma_{r+1},b_{r+1}}[h_{r+1}(f,f';\sigma_1,b_1,\cdots,\sigma_{r+1},b_{r+1})] &= \mathop{\mathbb{E}}_{\sigma,b}[e^{-\lambda\beta}e^{\lambda(z+\widehat{G}_{o_{f,\sigma,b}(f')})}(M(\lambda))^{d-r-1}] \\
&= e^{-\lambda\beta}e^{\lambda z}(M(\lambda))^{d-r-1}\mathop{\mathbb{E}}_{\sigma,b}[\exp(\lambda\widehat{G}_{o_{f,\sigma,b}(f')})] \\
&\leq e^{-\lambda\beta}e^{\lambda z}(M(\lambda))^{d-r} \\
&= h_r(f,f';\sigma_1,b_1,\cdots,\sigma_r,b_r)
\end{aligned}$$

where the 1st step is due to definition of h_r (see Definition 4.6), the 2nd step holds since $e^{\lambda z}$ and $M(\lambda)$ are independent of σ, b, the 3rd step uses Lemma 4.15, the 4th step is due to definition of h_r (see Definition 4.6). □

4.15 Putting it all together

Lemma 4.21. *If the following happens*

- *Let β be defined as Definition 4.1*
- *Let $\epsilon \in (0,1)$ be defined as Definition 4.1*
- *Let $H_d = \{(\sigma_r, a_r, b_r)\}_{r\in[d]}$ be a sequence of hashing chose by procedure in Definition 4.6*
- *Let $\widehat{G}$ be a flat filter in accordance of hashing functions in H_r (see Definition 3.12)*

it holds that, for all $f \neq f'$

$$\sum_{r\in[d]} \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)}\widehat{G}_{o_{f,\sigma_r,b_r}(f')} \leq \frac{\beta}{1-\epsilon}$$

Proof. Note that in each step, we choose σ_{r+1}, b_{r+1} to minimize

$$\sum_{f,f'\in\mathcal{F}:f\neq f'} h_{r+1}(f,f';\sigma_1,b_1,\cdots,\sigma_{r+1},b_{r+1}).$$

Then, we know that

$$\sum_{f,f'\in\mathcal{F}:f\neq f'} h_{r+1}(f,f';\sigma_1,b_1,\cdots,\sigma_{r+1},b_{r+1}) \leq \sum_{f,f'\in\mathcal{F}:f\neq f'} \mathop{\mathbb{E}}_{\sigma'_{r+1},b'_{r+1}}[h_{r+1}(f,f';\sigma_1,b_1,\cdots,\sigma'_{r+1},b'_{r+1})], \tag{4}$$

which follows from the linearity of expectation. By Lemma 4.20, it holds that

$$\sum_{f,f'\in\mathcal{F}:f\neq f'} \mathop{\mathbb{E}}_{\sigma'_{r+1},b'_{r+1}}[h_{r+1}(f,f';\sigma_1,b_1,\cdots,\sigma'_{r+1},b'_{r+1})] \leq \sum_{f,f'\in\mathcal{F}:f\neq f'} h_r(f,f';\sigma_1,b_1,\cdots,\sigma_r,b_r).$$

Hence by induction, we have

$$\sum_{f,f'\in\mathcal{F}:f\neq f'} h_d(f,f';\sigma_1,b_1,\cdots,\sigma_d,b_d) \leq \sum_{f,f'\in\mathcal{F}:f\neq f'} h_0(f,f') < 1$$

where the 2nd step is given by Lemma 4.19.

Therefore, by Lemma 4.17,

$$\sum_{f,f'\in\mathcal{F}:f\neq f'} \Pr[A_{f,f'} \mid \sigma_1,b_1,\cdots,\sigma_d,b_d] \leq \sum_{f,f'\in\mathcal{F}:f\neq f'} h_d(f,f';\sigma_1,b_1,\cdots,\sigma_d,b_d) < 1.$$

Note that conditioned on $\sigma_1,b_1,\cdots,\sigma_d,b_d$, $A_{f,f'}$ is a deterministic event. That is, the conditional probability for each pair of $f\neq f'$ is either zero or one. By the inequality, we get that

$$\Pr[A_{f,f'} \mid \sigma_1,b_1,\cdots,\sigma_d,b_d] = 0 \quad \forall f\neq f'\in\mathcal{F}.$$

Then by the definition of $A_{f,f'}$ (Definition 4.5), it implies that

$$\sum_{r\in[d]} \widehat{G}_{o_{f,\sigma_r,b_r}(f')} \leq \beta. \tag{5}$$

Note that $o_{f,\sigma_r,b_r}(f) \in [-\frac{1}{2B},\frac{1}{2B}]$ (see Lemma 4.16). Thus, by definition of $\widehat{G}$ (see Definition 3.12), we have

$$\widehat{G}_{o_{f,\sigma_r,b_r}(f)} \in [1-\epsilon,1].$$

Then Eq. (5) gives that, for all $f\neq f'\in\mathcal{F}$,

$$\sum_{r\in[d]} \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)} \widehat{G}_{o_{f,\sigma_r,b_r}(f')} \leq \frac{\beta}{1-\epsilon}.$$

The lemma is then proved. □

5 Super-linear Algorithm

5.1 The Guarantees of Measurement

According to our choice of parameters in Definition 4.1, we have the below relationship between d/β and B. This allows us to normalize the ℓ_1 bound of our algorithm by a factor of $1/k$.

Observation 5.1. *Let β,d,ϵ be chosen as Definition 4.1, we have*

$$\frac{\beta}{(1-\epsilon)d} = \Theta(\frac{1}{B})$$

Proof.

$$\frac{\beta}{(1-\epsilon)d} = \frac{\Theta(\log|\mathcal{F}|)}{(1-\Theta(1/B))\cdot\Theta(B\log(|\mathcal{F}|))} = \Theta(\frac{1}{B})$$

where the above equation is due to Definition 4.1. □

Lemma 5.2 (A variation of Lemma 5.1 in page 11 [LN20])**.** *Under following conditions*

- *Let* $H_r = \{\sigma_r, a_r, b_r\}_{r\in d}$ *be a sequence of hashing defined in Definition 3.10*
- *We have* $\widehat{G}$ *being a flat filter with* ϵ *buckets and sharpness* ϵ *(see Definition 3.12)*
- *Let* $\widehat{x}, v$ *be defined as Definition 3.5*
- *For all* $f \neq f' \in \mathcal{F}$ *it holds that,*

$$\sum_{r\in[d]} \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)} \widehat{G}_{o_{f,\sigma_r,b_r}(f')} \leq \frac{\beta}{1-\epsilon}$$

Then for every vector $x : [0, T] \to \mathbb{C}^n$ *and every* $f \in \mathcal{F}$*, for at least* $0.8d$ *indices* $r \in [d]$ *we have*

$$|v_f - \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)} (m_{H_r})_{h_{\sigma_r,b_r}(f)} \omega^{-a_r\sigma f}| \leq \Theta(\frac{1}{B}) \cdot \sum_{f'\in\mathcal{F}\setminus\{f\}} |v_{f'}| \tag{6}$$

Proof. The proof is close to [LN20], we keep it here for the completeness.

$$\begin{aligned}
&\sum_{r\in[d]} |v_f - \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)} (m_{H_r})_{h_r(f)} \omega^{-a\sigma f}| \\
=& \sum_{r\in[d]} |\widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)} \sum_{f'\in\mathcal{F}\setminus\{f\}} \widehat{G}_{o_{f,\sigma_r,b_r}(f')} v_{f'} \omega^{a_r\sigma_r(f'-f)}| \\
\leq& \sum_{r\in[d]} \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)} \sum_{f'\in\mathcal{F}\setminus\{f\}} \widehat{G}_{o_{f,\sigma_r,b_r}(f')} |v_{f'}| \\
=& \sum_{f'\in\mathcal{F}\setminus\{f\}} |v_{f'}| \sum_{r\in[d]} \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)} \widehat{G}_{o_{f,\sigma_r,b_r}(f')} \\
\leq& \sum_{f'\in\mathcal{F}\setminus\{f\}} |v_{f'}| \frac{\beta}{1-\epsilon}
\end{aligned}$$

where the first step uses Claim 3.15, the 2nd step is given by triangle inequality, the 3rd step is a change of summation order, the 4th step is by the condition of this lemma. the last step is by the Observation 5.1.

Therefore, at most $\frac{1}{5}$ fraction of $r \in [d]$ satisfy

$$\begin{aligned}
|v_f - \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)} (m_{H_r})_{h_{\sigma_r,b_r}(f)} \omega^{-a\sigma f}| &> \frac{5\beta}{(1-\epsilon)d} \cdot \sum_{f'\in\mathcal{F}\setminus\{f\}} |v_{f'}| \\
&= \Theta(\frac{1}{B}) \cdot \sum_{f'\in\mathcal{F}\setminus\{f\}} |v_{f'}|,
\end{aligned}$$

where the last step is by Observation 5.1.

Thus, we complete the proof. □

5.2 Assumption of noise function

In this section, we introduce the definition of (C, ξ)-noise function. We will show its restriction on the output of HashToBins algorithm in next section.

Definition 5.3 ((C, ξ)-noise)**.** *Under following conditions*

- *Let $g(t) : [0, T] \to \mathbb{R}$ be the noise function*
- *Let C be a fixed constant*
- *Let ξ be a parameter depend on $g(t)$*

then we say $g(t)$ is a (C, ξ)-noise if it satisfies

$$\max_{t\in[0,T]} |g(t)|^2 \leq C \cdot \frac{1}{T} \int_0^T |g(t)|^2 \mathrm{d}t + \xi$$

5.3 Hash to bins

This section presents the deterministic HashToBins in the continuous setting.

Algorithm 1 HashToBins

procedure HashToBins($x, \widehat{z}, H = (\sigma, a, b)$) ▷ Lemma 5.5
  for $j \in [BD]$ **do**
    $y_j \leftarrow G(j) \cdot P_{\sigma,a,b}(x)(j)$
  end for
  for $j \in [B]$ **do**
    $u_j \leftarrow \sum_{j\in[D]} y_{Bi+j} - \widehat{G} * \widehat{P_{\sigma,a,b}(z)}(j/B)$
  end for
  return The DFT $\widehat{u} \in \mathbb{C}^B$ of u
end procedure

Lemma 5.4 (Identities of DFT and CFT, Lemma 4.3 and Fact 4.1 in [JLS23])**.** *Under following conditions*

- *Let $P_{\sigma,a,b}x$ be defined as Definition 3.9*
- *Let $x^*(t)$ be the noiseless signal*
- *Let $z(t) := \sum_{f\in\mathcal{F}} \widehat{z}_f \cdot e^{2\pi \mathbf{i} f t}$*
- *Let $\widehat{u}$ be the output of* HashToBins

we have

- *Property 1: Identity of pseudo-permutation:*

$$\widehat{P_{\sigma,a,b}x^*}(t) = \frac{1}{\sigma}\widehat{x}^*(\frac{t}{\sigma} + b)e^{2\pi \mathbf{i} a(t+b\sigma)}$$

- *Property 2: Identity of output of* HashToBins*: For any $j \in [B]$*

$$\widehat{u}_j = \widehat{G} * \widehat{P_{\sigma,a,b}(x^* - z)}(j/B)$$

Proof. Notice that **Property 2** in [JLS23] contains only x. However, we can extend to our result by linear operation. □

Lemma 5.5 (HashToBins)**.** *If the following conditions hold:*

- *let* $H = (\sigma, a, b)$ *be a tuple of hashing defined in Definition 3.8*
- *Let* $\widehat{G}$ *be a flat filter (see Definition 3.12)*
- *let* $\widehat{x}, v$ *be defined as Definition 3.5*
- *let* $\widehat{z} \in \mathbb{C}^{|\mathcal{F}|}$ *be a vector,*
- *let* $g(t) = 0$ *for any* $t \in [0, T]$*,*

then there exists a deterministic procedure HashToBins$(x, \widehat{z}, H)$ *which computes* $u \in \mathbb{C}^B$ *with the following guarantees:*

- *Noiseless version: Let* $g(t) \equiv 0$*, for any* $f \in \mathcal{F}$*, the output* $\widehat{u}$ *of* HashToBins$(x, \widehat{z}, H)$ *satisfies*

$$\widehat{u}_{h_{\sigma,b}(f)} = \sum_{f' \in \mathcal{F}} \widehat{G}_{o_{f,\sigma,b}(f')}(v_{f'} - \widehat{z}'_f)\omega^{a\sigma f'},$$

- *Noise-only version: Let* $x^*(t) \equiv 0$*, the output* $\widehat{u}$ *of* HashToBins$(g, \mathbf{0}_B, H)$ *satisfies*

$$\|\widehat{u}\|_\infty \leq O(\frac{D}{B} \cdot (C \cdot (\frac{1}{T}\int_0^T |g(t)|^2 \mathrm{d}t + \xi))^{\frac{1}{2}})$$

- *the algorithm takes* $O(BD)$ *samples,*
- *the time complexity of the algorithm is* $O(BD + B\log(B))$*.*

Proof. **Part 1: Noiseless version** Let $\widehat{z} := \sum_{f \in \mathcal{F}} \widehat{z}_f \cdot \delta_f(\xi)$, we have

$$\begin{aligned}
\widehat{u}_{h_{\sigma,b}(f)} &= \widehat{G} * \widehat{P_{\sigma,a,b}(x^* - z)}(h(f)/B) \\
&= \int_{\xi \in \mathbb{R}} \widehat{G}(h_{\sigma,b}(f)/B - \xi)\widehat{P_{\sigma,a,b}(x^* - z)}(\xi)\mathrm{d}\xi \\
&= \int_{\xi \in \mathbb{R}} \widehat{G}(h_{\sigma,b}(f)/B - \xi) \cdot \frac{1}{\sigma}(\widehat{x}(\frac{\xi}{\sigma} + b) - \widehat{z}(\frac{\xi}{\sigma} + b)) \cdot e^{2\pi \mathbf{i} a(\xi + b\sigma)}\mathrm{d}\xi \\
&= \int_{\xi \in \mathbb{R}} \widehat{G}(h_{\sigma,b}(f)/B - \sigma(\xi - b)) \cdot (\widehat{x}(\xi) - \widehat{z}(\xi)) \cdot e^{2\pi \mathbf{i} a\sigma\xi}\mathrm{d}\xi \\
&= \int_{\xi \in \mathbb{R}} \widehat{G}(h_{\sigma,b}(f)/B - \sigma(\xi - b)) \cdot (\sum_{f \in \mathcal{F}}(v_f - \widehat{z}_f) \cdot \delta_f(\xi)) \cdot e^{2\pi \mathbf{i} a\sigma\xi}\mathrm{d}\xi \\
&= \sum_{f \in \mathcal{F}} \widehat{G}(h_{\sigma,b}(f)/B - \sigma(f - b)) \cdot (v_f - \widehat{z}_f) \cdot e^{2\pi \mathbf{i} a\sigma f} \\
&= \sum_{f \in \mathcal{F}} \widehat{G}(-o_{f,\sigma,b}(f)) \cdot (v_f - \widehat{z}_f) \cdot e^{2\pi \mathbf{i} a\sigma f}
\end{aligned}$$

where the first step is by **Property 2** of Lemma 5.4, the 3rd step is by **Property 1** of Lemma 5.4, the 4th step is by integral substitution, the 5th step is by the definition of sparse signal (see Definition 3.5), the 6th step is by definition of delta function, the last step is by the definition of hashing functions (see Definition 3.8).

The first part is then proved by the symmetricity of $\widehat{G}$.

Part 2: Noise-only version

For any $j \in [BD]$

$$\begin{aligned} y_j^2 &= G(j)^2 \cdot x^2(\sigma(t-a)) \\ &\leq O(\frac{1}{B^2}) \cdot x^2(\sigma(t-a)) \\ &\leq O(\frac{1}{B^2}) \cdot (C \cdot \frac{1}{T} \int_0^T |g(t)|^2 \mathrm{d}t + \xi) \end{aligned}$$

where the first step is by definition in line 4 of Algorithm 1, the 2nd step is by **Property 5** of Definition 3.12, the 3rd step is by Assumption 5.3.

Therefore, we have

$$|y_j| \leq O(\frac{1}{B} \cdot (C \cdot \frac{1}{T} \int_0^T |g(t)|^2 \mathrm{d}t + \xi)^{\frac{1}{2}})$$

We get the result by timing a D (see line 7 of Algorithm 1).

□

5.4 Guarantee of median of HashToBins

Combining the previous results, we derive the Guarantee of the median of Deterministic HashToBins.

Definition 5.6 (Choice of B)**.** *Let α be some constant to be determined later. Let k be the sparsity of the signal. We define B to be such that*

- $B = \Theta(k)$
- *B is a power of 2*
- *we choose the constant in $B = \Theta(k)$ such that the upper bound of Lemma 5.7 satisfies:*

$$\Theta(\frac{1}{B}) \leq \frac{1}{\alpha k}$$

Lemma 5.7 (Median of HashToBins outputs)**.** *Under following conditions,*

- *Let $\widehat{x}, v$ be defined as Definition 3.5*
- *Let $\widehat{z} \in \mathbb{C}^{|\mathcal{F}|}$*
- *Let $\widehat{w}_f := v_f - \widehat{z}_f$*
- *We have $\widehat{G}$ being a flat filter (see Definition 3.12)*
- *Let $\{H_r\}_{r\in[d]} = (\sigma_r, a_r, b_r)$ be a sequence of hashing defined in Definition 4.6*
- *Let u_r be the output of* HashToBins$(x, \widehat{z}, H_r)$

we have

- *For all $f \in \mathcal{F}$*

$$\begin{aligned} &|\widehat{w}_f - \operatorname*{median}_{r\in[d]} \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)}(u_r)_{h_{\sigma_r,b_r}(f)} \cdot \omega^{-a_r\sigma f}| \\ \leq\ & \frac{1}{\alpha k}(\sum_{f\in\mathcal{F}} |\widehat{w}(f)| + O(\frac{\log k}{k} \cdot (\frac{C}{T} \int_0^T |g(t)|^2 \mathrm{d}t + \xi)^{\frac{1}{2}})) := \mathcal{N}(\widehat{w}) \end{aligned}$$

Proof. Let $u_r^{(noiseless)}$ denote the output of HashToBins$(x, \widehat{z}, H_r)$ when $g(t) = 0$. Let $u_r^{(noise)}$ denote the output of HashToBins$(g, \mathbf{0}_{|\mathcal{F}|}, H_r)$.

Part 1: Guarantee of noiseless HashToBins

Using Lemma 4.21, we know the fifth condition in Lemma 5.2 should hold. Then we have for at least $0.8d$ indices $r \in [d]$

$$\begin{aligned}
&|\widehat{w}_f - \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)}(u_r)^{(noiseless)}_{h_{\sigma_r,b_r}(f)}\omega^{-a\sigma f}| \\
&= |\widehat{w}_f - \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)}(m_{H_r}(\widehat{w}(f)))_{h_{\sigma_r,b_r}(f)}\omega^{-a\sigma f}| \\
&\leq \Theta(\frac{1}{B}) \cdot \|\widehat{w}_{\mathcal{F}\setminus\{f\}}\|_1 \\
&\leq \frac{1}{\alpha k} \cdot \|\widehat{w}_{\mathcal{F}\setminus\{f\}}\|_1 \qquad (7)
\end{aligned}$$

where the first step is by Lemma 5.5, the 2nd step uses Lemma 5.2, the 3rd is given by choice of B in Definition 5.6.

Part 2: Bounds on output of HashToBins with noise function as the input

Since $\widehat{G}^{-1} \in (0, 1)$ by Definition 3.12, combining **Part 2** of Lemma 5.5 we have

$$|\operatorname{median} \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)}(u_r)^{(noise)}_{h_{\sigma_r,b_r}(f)}\omega^{-a\sigma f}| \leq O(\frac{D}{B} \cdot (\frac{C}{T}\int_0^T |g(t)|^2 \mathrm{d}t + \xi)^{\frac{1}{2}}) \qquad (8)$$

Part 3: Putting them together

Notice that every operations in HashToBins are linear. Therefore, we have $u_r = u_r^{(noiseless)} + u_r^{(noise)}$.

Then by taking median of outputs for hash functions in H_r we have,

$$\begin{aligned}
&|\widehat{w}(f) - \operatorname{median} \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)}(u_r)_{h_{\sigma_r,b_r}(f)}\omega^{-a\sigma f}| \\
&\leq |\widehat{w}(f) - \operatorname{median} \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)}(u_r)^{(noiseless)}_{h_{\sigma_r,b_r}(f)}\omega^{-a\sigma f}| + |\operatorname{median} \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)}(u_r)^{(noise)}_{h_{\sigma_r,b_r}(f)}\omega^{-a\sigma f}| \\
&\leq \frac{1}{\alpha k}\|\widehat{w}_{\mathcal{F}\setminus\{f\}}\|_1 + |\operatorname{median} \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)}(u_r)^{(noise)}_{h_{\sigma_r,b_r}(f)}\omega^{-a\sigma f}| \\
&\leq \frac{1}{\alpha k}\|\widehat{w}_{\mathcal{F}\setminus\{f\}}\|_1 + O(\frac{D}{B} \cdot (\frac{C}{T}\int_0^T |g(t)|^2 \mathrm{d}t + \xi)^{\frac{1}{2}})
\end{aligned}$$

where the first step is by triangle inequality, the 2nd step is by Eq. (7), the 3rd step is by Eq. (8).

Hence, we prove the wanted result. □

5.5 The Guarantee of Subrecovery

Lemma 5.8 (Lemma 5.2 in page 12 in [LN20])**.** *If the following conditions hold:*

- *let $\widehat{x}$, v be defined as Definition 3.5*
- *let $\widehat{z} \in \mathbb{C}^{|\mathcal{F}|}$,*
- *let B be defined as Definition 5.6,*
- *let $\widehat{w}_f := v_f - \widehat{z}_f$,*
- *let $\nu \geq 16\mathcal{N}(\widehat{x})$ be a constant to denote a threshold for heavy index,*

then the output of the Procedure SubRecovery *(Algorithm 2)* $\widehat{w}'$ *satisfies:*

- $|\widehat{w}_f| \geq (7/16)\nu$ *for all* $f \in \mathrm{supp}(\widehat{w}')$
- $|\widehat{w}_f - \widehat{w}'_f| \leq |\widehat{w}_f|/7$ *for all* $f \in \mathrm{supp}(\widehat{w}')$
- $\{f \in \mathcal{F} : |\widehat{w}_f| \geq \nu\} \subseteq \mathrm{supp}(\widehat{w}')$

Proof. **Proof of Part 1**

For any $f \in \mathcal{F}$, we have

$$\begin{aligned} |\widehat{w}_f - \widehat{w}'_f| &= |\widehat{w}_f - \operatorname*{median}_{r\in[d]} \widehat{G}^{-1}_{o_{f,r}(f)}(u_r)_{h_r(f)} \cdot \omega^{-a\sigma f}| \\ &\leq \mathcal{N} \\ &\leq \frac{\nu}{16} \end{aligned} \tag{9}$$

where the first step is the output of Algorithm 2, the 2nd step is by Lemma 5.7, the last step is by the 5th assumption of this lemma.

Then we have,

$$\begin{aligned} |\widehat{w}_f| &\geq |\widehat{w}'_f| - |\widehat{w}_f - \widehat{w}'_f| \\ &\geq \nu/2 - |\widehat{w}_f - \widehat{w}'_f| \\ &\geq \nu/2 - \nu/16 = (7/16)\nu \end{aligned} \tag{10}$$

where the first step uses triangle inequality, the 2nd step is by the threshold condition in 8th line of Algorithm 2, the 3rd step is due to Eq. (9).

Proof of Part 2

$$\begin{aligned} |\widehat{w}_f - \widehat{w}'_f| &\leq \frac{\nu}{16} \\ &\leq \frac{1}{16} \cdot (16/7)|\widehat{w}_f| = |\widehat{w}_f|/7 \end{aligned}$$

where the first step is given by Eq. (9), the 2nd step is by Eq. (10).

Proof of Part 3

$$\begin{aligned} |\widehat{w}'_f| &\geq |\widehat{w}_f| - |\widehat{w}_f - \widehat{w}'_f| \\ &\geq \nu - |\widehat{w}_f - \widehat{w}'_f| \\ &\geq \nu - \nu/16 > \nu/2 \end{aligned}$$

where the 1st step is by triangle inequality, the 2nd step is since $f \in \{f \in \mathcal{F} : |\widehat{w}_f| \geq \nu\}$, the 3rd step is by Eq. (9).

Since $|\widehat{w}'_f|$ is bigger than the threshold, it will be recover. □

5.6 Linear time main algorithm

Definition 5.9 (Constraints of constant parameters)**.** *We list some constraints for constant parameters in the main algorithm.*

- *Part 1 :* $C > 1$

Algorithm 2 Linear-time Sparse Recovery for $\widehat{x} - \widehat{z}$

procedure SUBRECOVERY($x \in \mathbb{C}^n$) ▷ Lemma 5.8
  $S \leftarrow \emptyset$
  for $r = 1 \to d$ **do**
    $u_r \leftarrow \text{HASHTOBINS}(x, \widehat{z}, (\sigma_r, 0, b_r))$ ▷ Lemma 5.5
  end for
  for $f \in \mathcal{F}$ **do**
    $\widehat{w}'_f \leftarrow \text{median}_{r\in[d]} \widehat{G}^{-1}_{o_{f,\sigma_r,b_r}(f)}(u_r)_{h_{\sigma_r,b_r}(f)} \cdot \omega^{-a\sigma f}$ ▷ Lemma 5.7
    if $|\widehat{w}'_f| > \nu/2$ **then**
      $S \leftarrow S \cup \{f\}$
    end if
  end for
  return $\widehat{w}'_S$
end procedure

- *Part 2 :* $C(1 - \frac{16}{\alpha}) \geq \frac{16}{\alpha}(1 + \frac{1}{\rho})$
- *Part 3 :* $\frac{7}{16}C \geq \frac{1}{\rho}$
- *Part 4 :* $\gamma \leq 7$

Below is a group of parameters that satisfies above constraints.

Definition 5.10 (Choice of constant parameters)**.** *We let* $C = 2, \rho = 32, \alpha = 32, \gamma = 2$.

Lemma 5.11 (ℓ_∞ norm reduction, analogous to Lemma 5.3 in page 13 in [LN20], Lemma 3.8 and Lemma 3.9 in page 10 in [PS15])**.** *If following holds*

- *Let* μ *be defined as Definition 3.16*
- *Let the SNR* R^* *(see Definition 3.16) satisfy* $R^* \leq (F/\eta)^m$ *for some constant parameter* m
- *let* $\widehat{x}$, v *be defined as Definition 3.5*
- *Let* C, β, ρ, γ *be some constant to be determined as Definition 5.10*
- *Let* $r_f^{(t)} := v_f - \widehat{z}_f^{(t)}$

For all $0 \leq t \leq T^*$*, there is an algorithm (Algorithm 3) outputs a vector* $\widehat{w}^{(T^*)} \in \mathbb{C}^n$ *which satisfies*

- *Let* $I := \{f : |v_f| \geq \frac{\mu}{\rho}\}$, $\widehat{x}(f) = r_f^{(t)}$ *for all* $f \notin I$
- $|r_f^{(t)}| \leq |v_f|$ *for all* f
- $\|r_I^{(T)}\|_\infty \leq \nu^{(t)}$
- *The algorithm takes* $O(k^2 \log k \cdot \log(F/\eta))$ *samples*
- *The algorithm runs in* $O((F/\eta)k \log^2(F/\eta))$ *time.*

Proof. The proof of the first three claims is by mathematical induction.

Initial Condition

The first and second claim clearly holds since $\widehat{z}_f^{(0)} = 0$ for all $f \in \mathcal{F}$ and hence $r_f^{(0)} = \widehat{x}(f)$.

For the 3rd claim, we have

$$\nu^{(0)} = C\mu\gamma^T = C\mu R^* = C\|v\|_\infty > \|v\|_\infty = \|r^{(0)}\|_\infty$$

where the 2nd step is by $T = \log_\gamma R^*$, the 3rd step is by the definition of R^* (see Definition 3.16), the 4th step is by $C > 1$ (**Part 1** of Definition 5.9), the last step is by definition of $\|r^{(0)}\|_\infty$.

Induction step

Now, we assum the first three claims hold for t, we want to prove its correctness for $t+1$.

For simplicity, we define a notation for the integral of noise on time interval,

$$J := O(\frac{\log k}{k} \cdot (\frac{C}{T}\int_0^T |g(t)|^2 \mathrm{d}t + \xi)^{\frac{1}{2}})$$

We have

$$\begin{aligned}
\frac{16}{\alpha k} \cdot (\|r^{(t)}\|_1 + J) &= \frac{16}{\alpha k} \cdot (\|r^{(t)}_{\mathcal{K}\cap I}\|_1 + \|r^{(t)}_{\mathcal{K}\setminus I}\|_1 + \|r^{(t)}_{\mathcal{F}\setminus\mathcal{K}}\|_1 + J) \\
&\leq \frac{16}{\alpha k} \cdot (k \cdot \|r_I^{(k)}\|_\infty + \|r^{(t)}_{\mathcal{K}\setminus I}\|_1 + \|r^{(t)}_{\mathcal{F}\setminus\mathcal{K}}\|_1 + J) \\
&\leq \frac{16}{\alpha k} \cdot (k \cdot C\mu\gamma^{T^*-t} + \|r^{(t)}_{\mathcal{K}\setminus I}\|_1 + \|r^{(t)}_{\mathcal{F}\setminus\mathcal{K}}\|_1 + J) \\
&\leq \frac{16}{\alpha k} \cdot (k \cdot C\mu\gamma^{T^*-t} + \frac{k\mu}{\rho} + \|r^{(t)}_{\mathcal{F}\setminus\mathcal{K}}\|_1 + J) \\
&= \frac{16}{\alpha k} \cdot (k \cdot C\mu\gamma^{T^*-t} + \frac{k\mu}{\rho} + k\mu) \\
&\leq \frac{16}{\alpha k} \cdot (k \cdot C\mu\gamma^{T^*-t} + \frac{\alpha}{16}(1 - \frac{16}{\alpha})Ck\mu) \\
&\leq \frac{16}{\alpha k} \cdot (k \cdot C\mu\gamma^{T^*-t} + \frac{\alpha}{16}(1 - \frac{16}{\alpha})Ck\mu\gamma^{T^*-t}) \\
&= C\mu\gamma^{T^*-t} := \nu^{(t)}
\end{aligned}$$

where the first step and 2nd step are trivial calculation, the 3rd step is by the third claim in this lemma and induction hypothesis, the 4th step is by the definition of I, the 5th step is by the definition of μ (see Definition 3.16), the 6th step is derived from **Part 2** of Definition 5.9, the 7th step is by $\gamma^{T^*-t} > 1$ and rearrangement.

Therefore, we've proved the fifth condition for Lemma 5.8.

Claim 1:

For index $f \in \mathcal{F}\setminus I$, we have

$$\begin{aligned}
||\widehat{x}(f)| &\leq \frac{\mu}{\rho} \\
&\leq \frac{7}{16}C\mu \\
&\leq \frac{7}{16}C\mu\gamma^{T^*-t} := \nu^{(t)}
\end{aligned}$$

where the first step is by definition of I, the 2nd step is by **Part 3** of Definition 5.9, the 3rd step is by $\gamma^{T^*-t} > 1$.

By **Part 1** of Lemma 5.8, we know that $|\widehat{x}(f)|$ will never be recovered in this procedure.

Claim 2:

Notice that $r_f^{(t)}$ is defined to be $\widehat{x}(f) - \widehat{z}_f^{(t)}$, then we have,

$$\begin{aligned} |r_f^{(t+1)}| &:= |\widehat{w}_f - \widehat{w}'_f| \\ &\leq |\widehat{w}_f|/7 \\ &:= |r_f^{(t)}|/7 \end{aligned} \tag{11}$$

where the first and the third step is by definition of r, the 2nd step uses **Part 2** of Lemma 5.8

Since we know $|r_f^{(0)}| = |\widehat{x}(f)|$, Claim 2 clearly holds.

Claim 3:

For $f \in I$ such that $|r_f^{(t)}| \leq \nu^{(t+1)}$. Claim 3 is proved by Eq. (11).

Otherwise, we have $|r_f^{(t)}| > \nu^{(t+1)}$, then we have,

$$|r_f^{(t+1)}| \leq |r_f^{(t)}|/7 \leq \nu_f^{(t)}/7 \leq \nu^{(t)}/\gamma := \nu^{(t+1)}$$

where the first step is by Eq. (11), the 2nd step is by induction hypothesis, the 3rd step is by **Part 4** of Definition 5.9.

Therefore, Claim 3 holds and we verify the induction step.

Proof of Sample Complexity.

The sample complexity of the algorithm is counted as below,

$$\begin{aligned} \text{Sample Complexity} &= d \cdot \textsc{HashToBins} \\ &= d \cdot O(B \log(B)) \\ &= O(B \log(F/\eta) \cdot O(\log B) \cdot O(B) \\ &= O(k^2 \log k \cdot \log(F/\eta)) \end{aligned}$$

where the first step is since we have d hashing tuples, the 2nd step is by Lemma 5.5, the third step is by choice of d (see Definition 4.1), the 5=4th step is by $B = \Theta(k)$ (see Definition 5.6).

Notice that we can reuse the sample, so we only need to count d times of HashToBins.

Proof of Running Time.

The time complexity of SubRecovery is

$$\begin{aligned} d \cdot \textsc{HashToBins} + (F/\eta) \cdot \text{Taking Median} &= d \cdot O(k \log(F/\eta)) + F/\eta \cdot \text{Taking Median} \\ &= d \cdot O(k \log(F/\eta)) + (F/\eta) \cdot d \\ &= O(k^2 \log^2(F/\eta) + (F/\eta) k \log(F/\eta)) \\ &= O((F/\eta) k \log(F/\eta)) \end{aligned}$$

where the 2nd step is because we scan the sequence of hashing when taking the median, the 3rd step is by the choice of d, the 4th step holds for $F/\eta \gg k$.

Hence, the time complexity for the main algorithm is $T^* \cdot O((F/\eta) k \log(F/\eta)) = O((F/\eta) k \log^2(F/\eta))$ since $R^* = O((F/\eta)^m)$ by the first assumption of this lemma.

□

Algorithm 3 Linear-time sparse recovery for $\widehat{x}$

procedure MAIN($x \in \mathbb{C}^n$) ▷ Lemma 5.11
  $T^* \leftarrow \log_\gamma R^*$
  $\widehat{z}^{(0)} \leftarrow \mathbf{0}_{|\mathcal{F}|/\eta}$
  $\nu^{(0)} \leftarrow C\mu\gamma^T$
  for $t = 0 \to T-1$ **do**
    $\widehat{z}^{(t+1)} \leftarrow \widehat{z}^{(t)} + \text{SUBRECOVERY}(x, \widehat{z}^{(t)}, \nu^{(t)})$ ▷ Algorithm 2
    $\nu^{(t+1)} \leftarrow \nu^{(t)}/\gamma$
  end for
  return $\widehat{z}$
end procedure

5.7 Main Result

Theorem 5.12 (Deterministic CFT (restatement of Theorem 2.9))**.** *If the following conditions hold*

- *Let $x : [0, T] \to \mathbb{C}$ has Continuous Fourier Transform $\widehat{x} : [-F, F] \to \mathbb{C}$*
- *Let the SNR R^* (see Definition 3.16) satisfy $R^* \leq (F/\eta)^m$ for some constant parameter m*
- *Let $|\mathcal{F}|, B$ be the powers of 2*
- *Let the noise function $g(t)$ satisfy Definition 5.3*

Then for any vector $x \in \mathbb{C}^n$ satisfying above assumptions, there is an algorithm that

- *it finds an $O(k)$-sparse vector $\widehat{x}' \in \mathbb{C}^{|\mathcal{F}|}$*
- *We have $|v_f - \widehat{x}'_f| \leq O(\mathcal{N})$ for all $f \in \mathcal{F}$, where*

$$\mathcal{N} := \frac{1}{\alpha k}\left(\sum_{f\in\mathcal{F}} |v_f| + O\left(\frac{\log k}{k} \cdot \left(\frac{C}{T}\int_0^T |g(t)|^2 \mathrm{d}t + \xi\right)^{\frac{1}{2}}\right)\right)$$

- *The algorithm takes $O(k^2 \log k \cdot \log(F/\eta))$ samples*
- *The algorithm runs in $O((F/\eta)k\log^2(F/\eta))$ time.*

Proof. This theorem is a direct corollary of Lemma 5.11.

Proof of sparsity output

This is followed by $|I| = O(k)$ and $f \notin I$ is not recovered (**Part 1** of Lemma 5.11).

Proof of Guarantee

$$\begin{aligned}
||r^{(T^*)}||_\infty &= \max\{||r_I^{(T^*)}||_\infty, ||r_{I^c}^{(T^*)}||_\infty\} \\
&\leq \max\{||\nu^{(T^*)}||_\infty, ||r_{\mathcal{F}\setminus I}^{(T^*)}||_\infty\} \\
&\leq \max\{\nu^{(T^*)}, ||\widehat{x}_{\mathcal{F}\setminus I}^{(T^*)}||_\infty\} \\
&\leq \max\{C\mu, ||\widehat{x}_{\mathcal{F}\setminus I}^{(T^*)}||_\infty\} \\
&\leq \max\{C\mu, (1/\rho)\mu\} = O(\mathcal{N})
\end{aligned}$$

where the 2nd step is by **Part 3** of Lemma 5.11, the 3rd step is by bf Part 2 of Lemma 5.11, the 4th step is by definition of μ, the 5th step is by definition of I.

Proof of Sample Complexity and Time Complexity

They are calculated in Lemma 5.11. □

References

[AKK+20] Thomas D. Ahle, Michael Kapralov, Jakob Bæk Tejs Knudsen, Rasmus Pagh, Ameya Velingker, David P. Woodruff, and Amir Zandieh. Oblivious sketching of high-degree polynomial kernels. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 141–160, 2020.

[BCG+12] Petros Boufounos, Volkan Cevher, Anna C Gilbert, Yi Li, and Martin J Strauss. What's the frequency, Kenneth?: Sublinear Fourier sampling off the grid. In *Algorithmica(A preliminary version of this paper appeared in the Proceedings of RANDOM/APPROX 2012, LNCS 7408, pp.61–72)*, pages 1–28. Springer, 2012.

[BD08] Thomas Blumensath and Mike E Davies. Iterative thresholding for sparse approximations. *Journal of Fourier analysis and Applications*, 14(5-6):629–654, 2008.

[Bou14] Jean Bourgain. An improved estimate in the restricted isometry problem. In *Geometric Aspects of Functional Analysis*, pages 65–70. Springer, 2014.

[Bri17] Karl Bringmann. A near-linear pseudopolynomial time algorithm for subset sum. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1073–1084. SIAM, https://arxiv.org/pdf/1610.04712.pdf, 2017.

[CGV13] Mahdi Cheraghchi, Venkatesan Guruswami, and Ameya Velingker. Restricted isometry of Fourier matrices and list decodability of random linear codes. *SIAM Journal on Computing*, 42(5):1888–1914, 2013.

[CKPS16] Xue Chen, Daniel M Kane, Eric Price, and Zhao Song. Fourier-sparse interpolation without a frequency gap. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 741–750. IEEE, 2016.

[CLRS09] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2009.

[CLS20] Sitan Chen, Jerry Li, and Zhao Song. Learning mixtures of linear regressions in subexponential time via Fourier moments. In *STOC*. https://arxiv.org/pdf/1912.07629.pdf, 2020.

[CP19a] Xue Chen and Eric Price. Active regression via linear-sample sparsification. In *Conference on Learning Theory (COLT)*, pages 663–695. PMLR, 2019.

[CP19b] Xue Chen and Eric Price. Estimating the frequency of a clustered signal. In *ICALP*, 2019.

[CT65] James W Cooley and John W Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[CT06] Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE transactions on information theory*, 52(12):5406–5425, 2006.

[DKS16a] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. The Fourier transform of poisson multinomial distributions and its algorithmic applications. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing (STOC)*, pages 1060–1073, 2016.

[DKS16b] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Optimal learning via the Fourier transform for sums of independent integer random variables. In *Conference on Learning Theory (COLT)*, pages 831–849, 2016.

[DKS16c] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Properly learning poisson binomial distributions in almost polynomial time. In *Conference on Learning Theory (COLT)*, pages 850–878, 2016.

[Don06] David L. Donoho. Compressed sensing. *IEEE Trans. Information Theory*, 52(4):1289–1306, 2006.

[Fou22] Jean Baptiste Joseph baron Fourier. *Théorie analytique de la chaleur*. F. Didot, 1822.

[Für09] Martin Fürer. Faster integer multiplication. *SIAM Journal on Computing*, 39(3):979–1005, 2009.

[GGI+02] Anna C Gilbert, Sudipto Guha, Piotr Indyk, S Muthukrishnan, and Martin Strauss. Near-optimal sparse Fourier representations via sampling. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 152–161. ACM, 2002.

[GMS05] Anna C Gilbert, S Muthukrishnan, and Martin Strauss. Improved time bounds for near-optimal sparse Fourier representations. In *Optics & Photonics 2005*, pages 59141A–59141A. International Society for Optics and Photonics, 2005.

[HIKP12a] Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. Nearly optimal sparse fourier transform. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 563–578, 2012.

[HIKP12b] Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. Simple and practical algorithm for sparse Fourier transform. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms (SODA)*, pages 1183–1194. SIAM, https://groups.csail.mit.edu/netmit/sFFT/soda_paper.pdf, 2012.

[HR16] Ishay Haviv and Oded Regev. The restricted isometry property of subsampled Fourier matrices. In *SODA*, pages 288–297. https://arxiv.org/pdf/1507.01768.pdf, 2016.

[IK14] Piotr Indyk and Michael Kapralov. Sample-optimal Fourier sampling in any constant dimension. In *IEEE 55th Annual Symposium onFoundations of Computer Science (FOCS)*, pages 514–523. IEEE, https://arxiv.org/pdf/1403.5804.pdf, 2014.

[IKP14] Piotr Indyk, Michael Kapralov, and Eric Price. (Nearly) Sample-optimal sparse Fourier transform. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 480–499. SIAM, 2014.

[JLS23] Yaonan Jin, Daogao Liu, and Zhao Song. Super-resolution and robust sparse continuous fourier transform in any constant dimension: Nearly linear time and sample complexity. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 4667–4767. SIAM, 2023.

[JSWZ21] Shunhua Jiang, Zhao Song, Omri Weinstein, and Hengjie Zhang. Faster dynamic matrix inverse for faster lps. In *STOC*. https://arxiv.org/pdf/2004.07470.pdf, 2021.

[Kap16] Michael Kapralov. Sparse Fourier transform in any constant dimension with nearly-optimal sample complexity in sublinear time. In *Symposium on Theory of Computing Conference (STOC)*. https://arxiv.org/pdf/1604.00845.pdf, 2016.

[Kap17] Michael Kapralov. Sample efficient estimation and recovery in sparse FFT via isolation on average. In *58th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. https://arxiv.org/pdf/1708.04544, 2017.

[KM93] Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993.

[KVZ19] Michael Kapralov, Ameya Velingker, and Amir Zandieh. Dimension-independent sparse Fourier transform. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2709–2728. SIAM, https://arxiv.org/pdf/1902.10633.pdf, 2019.

[KX17] Konstantinos Koiliaris and Chao Xu. A faster pseudopolynomial time algorithm for subset sum. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1062–1072. SIAM, https://arxiv.org/pdf/1507.02318.pdf, 2017.

[LDFU13] Yichao Lu, Paramveer Dhillon, Dean P Foster, and Lyle Ungar. Faster ridge regression via the subsampled randomized hadamard transform. In *Advances in neural information processing systems*, pages 369–377, 2013.

[LN20] Yi Li and Vasileios Nakos. Deterministic sparse fourier transform with an ℓ_∞ guarantee. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[LSZ19] Yin Tat Lee, Zhao Song, and Qiuyi Zhang. Solving empirical risk minimization in the current matrix multiplication time. In *COLT*. https://arxiv.org/pdf/1905.04447.pdf, 2019.

[Moi15] Ankur Moitra. The threshold for super-resolution via extremal functions. In *STOC*. https://arxiv.org/pdf/1408.1681.pdf, 2015.

[NS19] Vasileios Nakos and Zhao Song. Stronger L2/L2 compressed sensing; without iterating. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing (STOC)*. https://arxiv.org/pdf/1903.02742, 2019.

[NSW19] Vasileios Nakos, Zhao Song, and Zhengyu Wang. (Nearly) Sample-optimal sparse Fourier transform in any dimension; RIPless and Filterless. In *FOCS*. https://arxiv.org/pdf/1909.11123.pdf, 2019.

[PS15] Eric Price and Zhao Song. A robust sparse fourier transform in the continuous setting. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 583–600. IEEE, 2015.

[RV08] Mark Rudelson and Roman Vershynin. On sparse reconstruction from Fourier and Gaussian measurements. *Communications on Pure and Applied Mathematics*, 61(8):1025–1045, 2008.

[SSWZ23] Zhao Song, Baocheng Sun, Omri Weinstein, and Ruizhe Zhang. Quartic samples suffice for fourier interpolation. In *FOCS*. arXiv preprint arXiv:2210.12495, 2023.

[SWYZ21] Zhao Song, David Woodruff, Zheng Yu, and Lichen Zhang. Fast sketching of polynomial kernels of polynomial degree. In *International Conference on Machine Learning*, pages 9812–9823. PMLR, 2021.

[SZZ21] Zhao Song, Lichen Zhang, and Ruizhe Zhang. Training multi-layer over-parametrized neural network in subquadratic time. *arXiv preprint arXiv:2112.07628*, 2021.