Distributed Adaptive Sampling for Kernel Matrix Approximation

Daniele Calandriello **Alessandro Lazaric** **Michal Valko**
SequeL team, INRIA Lille - Nord Europe

Abstract

Most kernel-based methods, such as kernel regression, kernel PCA, ICA, or k-means clustering, do not scale to large datasets, because constructing and storing the kernel matrix $\mathbf{K}_n$ requires at least $\mathcal{O}(n^2)$ time and space for n samples. To reduce this complexity, it is common to retain only a small dictionary of important samples, while forgetting and discarding those that are redundant. Recent works [1, 10] show that sampling points with replacement according to their ridge leverage scores (RLS) generates small dictionaries of relevant points with strong spectral approximation guarantees for $\mathbf{K}_n$. The drawback of RLS-based methods is that computing exact RLS requires constructing and storing the whole kernel matrix. In this paper, we introduce SQUEAK, a new algorithm for kernel approximation based on RLS sampling that *sequentially* processes the dataset, storing a dictionary which creates accurate kernel matrix approximations with a number of points that only depends on the effective dimension $d_{\text{eff}}(\gamma)$ of the dataset. Moreover since all the RLS estimations are efficiently performed using only the small dictionary, SQUEAK never constructs the whole matrix $\mathbf{K}_n$, runs in linear time $\widetilde{\mathcal{O}}(nd_{\text{eff}}(\gamma)^3)$ w.r.t. n, requires a single pass over the dataset, and can continuously update its solution as new data arrives. We also propose a parallel and distributed version of SQUEAK achieving similar accuracy in as little as $\widetilde{\mathcal{O}}(\log(n)d_{\text{eff}}(\gamma)^3)$ time.

1 Introduction

Non-parametric models are particularly suited to continued learning problems, since they automatically adapt to the processed data, incorporating useful examples to increase their model complexity. This makes them particularly suited to *streaming* and *lifelong* learning tasks. Unfortunately, one of the major limits of kernel ridge regression (KRR), kernel PCA [14], and other kernel methods is that for n samples storing and manipulating the kernel matrix $\mathbf{K}_n$ requires $\mathcal{O}(n^2)$ space, which becomes rapidly infeasible for even a relatively small n. For larger sizes (or streams) we cannot even afford to store or process the data on a single machine.

Many solutions focus on how to scale kernel methods by reducing its space (and time) complexity without compromising the prediction accuracy. A popular approach is to construct low-rank approximations of the kernel matrix by randomly selecting a subset (dictionary) of m columns from $\mathbf{K}_n$, thus reducing the space complexity to $\mathcal{O}(nm)$. These methods, often referred to as *Nyström approximations*, mostly differ in the distribution used to sample the columns of $\mathbf{K}_n$ and the construction of low-rank approximations. Both of these choices significantly affect the accuracy of the resulting approximation [13]. Bach [2] showed that uniform sampling preserves the prediction accuracy of KRR (up to ε) only when the number of columns m is proportional to the maximum degree of freedom of the kernel matrix. This may require sampling $\mathcal{O}(n)$ columns in datasets with high coherence [7], i.e., a kernel matrix with weakly correlated columns. On the other hand, Alaoui and Mahoney [1] showed that sampling columns according to their ridge leverage scores (RLS) (i.e., a measure of the influence of a point on the regression) produces an accurate Nyström approximation with only a number of columns m proportional to the average degrees of freedom of the matrix, called *effective dimension*. Unfortunately, the complexity of computing RLS requires storing the whole kernel matrix, thus making this approach infeasible. However, Alaoui and Mahoney [1] proposed a fast method to compute a constant-factor approximation of the RLS and showed that accuracy

Accepted at Lifelong Learning: A Reinforcement Learning Approach Workshop @ICML, Sydney, Australia, 2017.

A conference version of this paper appears under the same title in the proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS) 2017.

and space complexity are close to the case of sampling with exact RLS at the cost of an extra dependency on the inverse of the minimal eigenvalue of the kernel matrix. Unfortunately, the minimal eigenvalue can be arbitrarily small in many problems. Calandriello et al. [3] addressed this issue by processing the dataset *incrementally* and updating estimates of the ridge leverage scores, effective dimension, and Nyström approximations on-the-fly. The resulting algorithm (INK-Estimate) constructs w.h.p a dictionary that *retains* only the samples necessary for the reconstruction of $\mathbf{K}_n$, automatically increasing the size of the model to match the increasing difficulty of the problem as new samples are processed. At the same time, in order to maintain a small space complexity, it *forgets* redundant samples when it can do so without compromising the reconstruction accuracy. Although the space complexity of INK-Estimate does not depend on the minimal eigenvalue anymore, it introduces a dependency on the largest eigenvalue of $\mathbf{K}_n$, which in the worst case can be as big as n, thus losing the advantage of the method.

In this paper we introduce an algorithm for SeQUEntial Approximation of Kernel matrices (SQUEAK), a new algorithm that builds on INK-Estimate, but uses *unnormalized* RLS. This improvement, together with a new analysis, opens the way to major improvements over current leverage sampling methods (see Sect. 6 for a comparison with existing methods) closely matching the dictionary size achieved by exact RLS sampling. First, unlike INK-Estimate, SQUEAK is simpler, does not need to compute an estimate of the effective dimension for normalization, and exploits a simpler, more accurate RLS estimator. This new estimator only requires access to the points stored in the dictionary. Since the size of the dictionary is much smaller than n, SQUEAK needs to actually observe only a fraction of the kernel matrix $\mathbf{K}_n$, resulting in a runtime linear in n. Second, since our dictionary updates require only access to local data, our algorithm allows for distributed processing where machines operating on different dictionaries do not need to communicate with each other. In particular, intermediate dictionaries can be extracted in parallel from small portions of the dataset and they can be later merged in a hierarchical way. Third, the sequential nature of SQUEAK requires a more sophisticated analysis that take into consideration the complex interactions and dependencies between successive resampling steps. The analysis of SQUEAK builds on a new martingale argument that could be of independent interest for similar online resampling schemes. We note there exist other ways to avoid the intricate dependencies with simpler analysis, for example by resampling [10], but with negative algorithmic side effects: these methods need to pass through the dataset multiple times. SQUEAK passes *through the dataset only once*[1] and is therefore the first provably accurate kernel approximation algorithm that can handle both *streaming and distributed* settings. Finally, our SQUEAK can naturally incorporate new data without the need of recomputing the whole resparsification from scratch and therefore it can be applied in streaming and lifelong learning settings. In particular, it makes no assumption on the input data and guarantees that the dictionary size closely matches the empirical effective dimension of the data processed so far. If the input data comes from multiple tasks (e.g. different distribution), the dictionary size will automatically grow to adapt to the different problems, while at the same time exploiting similarities across subsequent problems to eliminate redundant samples.

2 Background

In this section, we introduce the notation and basics of kernel approximation used through the paper.

Notation. We use curly capital letters $\mathcal{A}$ for collections. We use upper-case bold letters $\mathbf{A}$ for matrices and operators, lower-case bold letters $\mathbf{a}$ for vectors, and lower-case letters a for scalars, with the exception of f, g, and h which denote functions. We denote by $[\mathbf{A}]_{ij}$ and $[\mathbf{a}]_i$, the (i, j) element of a matrix and ith element of a vector respectively. We denote by $\mathbf{I}_n \in \mathbb{R}^{n\times n}$, the identity matrix of dimension n and by $\operatorname{Diag}(\mathbf{a}) \in \mathbb{R}^{n\times n}$ the diagonal matrix with the vector $\mathbf{a} \in \mathbb{R}^n$ on the diagonal. We use $\mathbf{e}_{n,i} \in \mathbb{R}^n$ to denote the indicator vector for element i of dimension n. When the dimension of $\mathbf{I}$ and $\mathbf{e}_i$ is clear from the context, we omit the n. We use $\mathbf{A} \succeq \mathbf{B}$ to indicate that $\mathbf{A} - \mathbf{B}$ is a Positive Semi-Definite (PSD) matrix or operator. Finally, the set of integers between 1 and n is denoted as $[n] := \{1, \ldots, n\}$, and between i and j as $[i : j] := \{i, \ldots, j\}$.

Kernel. We consider a positive definite kernel function $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ and we denote with $\mathcal{H}$ its induced Reproducing Kernel Hilbert Space (RKHS), and with $\varphi : \mathcal{X} \to \mathcal{H}$ its corresponding feature map. Using φ, and without loss of generality, for the rest of the paper we will replace $\mathcal{H}$ with a high dimensional space $\mathbb{R}^D$ where D is large and potentially infinite. With this notation, the kernel evaluated between to points can be expressed as $\mathcal{K}(\mathbf{x}, \mathbf{x}') = \langle \mathcal{K}(\mathbf{x}, \cdot), \mathcal{K}(\mathbf{x}', \cdot)\rangle_{\mathcal{H}} = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}')\rangle_{\mathcal{H}} = \varphi(\mathbf{x})^{\mathsf{T}}\varphi(\mathbf{x}')$. Given a dataset of points $\mathcal{D} = \{\mathbf{x}_t\}_{t=1}^n$, we define the (empirical) ker-

[1] Note that there is an important difference in whether the method passes through *kernel matrix* only once or through the *dataset* only once, in the former, the algorithm may still need access one data point up to n times, thus making it unsuitable for the streaming setting and less practical for distributed computation.

nel matrix $\mathbf{K}_t \in \mathbb{R}^{t\times t}$ as the application of the kernel function on all pairs of input values (i.e., $[\mathbf{K}_t]_{ij} = k_{i,j} = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$ for any $i, j \in [t]$), with $\mathbf{k}_{t,i} = \mathbf{K}_t\mathbf{e}_{t,i}$ as its i-th column. We also define the feature vectors $\boldsymbol{\phi}_i = \varphi(\mathbf{x}_i) \in \mathbb{R}^D$ and after introducing the matrix $\boldsymbol{\Phi}_t = [\boldsymbol{\phi}_1, \boldsymbol{\phi}_2, \ldots, \boldsymbol{\phi}_t] \in \mathbb{R}^{D\times t}$ we can rewrite the kernel matrix as $\mathbf{K}_t = \boldsymbol{\Phi}_t^\mathsf{T}\boldsymbol{\Phi}_t$.

Kernel approximation by column sampling. One of the most popular strategies to have low space complexity approximations of the kernel $\mathbf{K}_t$ is to randomly select a subset of its columns (possibly reweighted) and use them to perform the specific kernel task at hand (e.g., kernel regression). More precisely, we define a column dictionary as a collection $\mathcal{I}_t = \{(i, w_i)\}_{i=1}^t$, where the first term denotes the index of the column and w_i its weight, which is set to zero for all columns that are not retained. For the theoretical analysis, we conveniently keep the dimension of any dictionary $\mathcal{I}_t$ to t, while in practice, we only store the non-zero elements. In particular, we denote by $|\mathcal{I}_t|$ be the size of the dictionary corresponding to the elements with non-zero weights w_i. Associated with a column dictionary, there is a selection matrix $\mathbf{S}_t = \mathrm{Diag}(\sqrt{w_1}\ldots\sqrt{w_t}) \in \mathbb{R}^{t\times t}$ such that for any matrix $\mathbf{A}_t \in \mathbb{R}^{t\times t}$, $\mathbf{A}_t\mathbf{S}_t$ returns a $t \times t$ matrix where the columns selected by $\mathcal{I}_t$ are properly reweighted and all other columns are set to 0. Despite the wide range of kernel applications, it is possible to show that in most of them, the quality of a dictionary can be measured in terms of how well it approximates the projection associated to the kernel. In kernel regression, for instance, we use $\mathbf{K}_t$ to construct the projection (hat) matrix that projects the observed labels $\mathbf{y}_t$ to $\widehat{\mathbf{y}}_t$. In particular, let $\mathbf{P}_t = \mathbf{K}_t\mathbf{K}_t^+$ be the projection matrix (where $\mathbf{K}_t^+$ indicates the pseudoinverse), then $\widehat{\mathbf{y}}_t = \mathbf{P}_t\mathbf{y}_t$. If $\mathbf{K}_t$ is full-rank, then $\mathbf{P}_t = \mathbf{I}_t$ is the identity matrix and, we can reconstruct any target vector $\mathbf{y}_t$ exactly. On the other hand, the only sampling scheme which guarantees to properly approximate a full rank $\mathbf{P}_t$ requires all columns to be represented in $\mathcal{I}_t$. In fact, all columns have the same "importance" and no low-space approximation is possible. Nonetheless, kernel matrices are often either rank deficient or have extremely small eigenvalues (exponentially decaying spectrum), as a direct (and desired) consequence of embedding low dimensional points $\mathbf{x}_i$ into a high dimensional RKHS. In this case, after soft thresholding the smaller eigenvalues to a given value γ, $\mathbf{K}_t$ can be effectively approximated using a small subset of columns. This is equivalent to approximating the γ-ridge projection matrix

$$\mathbf{P}_t \overset{\text{def}}{=} (\mathbf{K}_t + \gamma\mathbf{I})^{-1/2}\mathbf{K}_t(\mathbf{K}_t + \gamma\mathbf{I})^{-1/2}.$$

We say that a column dictionary is accurate if the following condition is satisfied.

Definition 1. *A dictionary $\mathcal{I}_t = \{(i, w_i)\}_{i=1}^t$ and its associated selection matrix $\mathbf{S}_t \in \mathbb{R}^{t\times t}$ are ε-accurate w.r.t. a kernel matrix $\mathbf{K}_t = \mathbf{K}_t^{1/2}\mathbf{K}_t^{1/2}$ if* [2]

$$\|\mathbf{P}_t - \widetilde{\mathbf{P}}_t\| \leq \varepsilon, \tag{1}$$

where for a given $\gamma > 0$, the approximated projection matrix is defined as

$$\widetilde{\mathbf{P}}_t \overset{\text{def}}{=} (\mathbf{K}_t + \gamma\mathbf{I}_t)^{-\frac{1}{2}}\mathbf{K}_t^{1/2}\mathbf{S}_t\mathbf{S}_t^\mathsf{T}\mathbf{K}_t^{1/2}(\mathbf{K}_t + \gamma\mathbf{I}_t)^{-\frac{1}{2}}.$$

Notice that this definition of accuracy is purely theoretical, since $\widetilde{\mathbf{P}}_t$ is never computed. Nonetheless, as illustrated in Sect. 5, ε-accurate dictionaries can be used to construct suitable kernel approximation in a wide range of problems.

Ridge leverage scores sampling. Alaoui and Mahoney [1] showed that an ε-accurate dictionary can be obtained by sampling columns proportionally to their γ-ridge leverage scores (RLS) defined as follows.

Definition 2. *Given a kernel matrix $\mathbf{K}_t \in \mathbb{R}^{t\times t}$, the γ-ridge leverage score (RLS) of column $i \in [t]$ is*

$$\tau_{t,i} = \mathbf{e}_{t,i}^\mathsf{T}\mathbf{K}_t(\mathbf{K}_t + \gamma\mathbf{I}_t)^{-1}\mathbf{e}_{t,i}, \tag{2}$$

Furthermore, the effective dimension $d_{\mathit{eff}}(\gamma)_t$ of the kernel matrix $\mathbf{K}_t$ is defined as

$$d_{\mathit{eff}}(\gamma)_t = \sum_{i=1}^t \tau_{t,i}(\gamma) = \mathrm{Tr}\left(\mathbf{K}_t(\mathbf{K}_t + \gamma\mathbf{I}_t)^{-1}\right). \tag{3}$$

The RLS can be interpreted and derived in many ways, and they are well studied [5, 4, 18] in the linear setting (e.g. $\boldsymbol{\phi}_t = \mathbf{x}_t$). Patel et al. [12] used them as a measure of incoherence to select important points, but their deterministic algorithm provides guarantees only when $\mathbf{K}_t$ is exactly low-rank. Here we notice that

$$\tau_{t,i} = \mathbf{e}_{t,i}^\mathsf{T}\mathbf{K}_t(\mathbf{K}_t + \gamma\mathbf{I}_t)^{-1}\mathbf{e}_{t,i} = \mathbf{e}_{t,i}^\mathsf{T}\mathbf{P}_t\mathbf{e}_{t,i},$$

which means that they correspond to the diagonal elements of the $\mathbf{P}_t$ itself. Intuitively, this correspond to selecting each column i with probability $p_{t,i} = \tau_{t,i}$ will capture the most important columns to define $\mathbf{P}_t$, thus minimizing the approximation error $\|\mathbf{P}_t - \widetilde{\mathbf{P}}_t\|$. More formally, Alaoui and Mahoney [1] state the following.

Proposition 1. *Let $\varepsilon \in [0, 1]$ and $\mathcal{I}_n$ be the dictionary built with m columns randomly selected proportionally to RLSs $\{\tau_{n,i}\}$ with weight $w_i = 1/(m\tau_{n,i})$. If $m = \mathcal{O}(\frac{1}{\varepsilon^2}d_{\mathit{eff}}(\gamma)_n\log(\frac{n}{\delta}))$, then w.p. at least $1-\delta$, the corresponding dictionary is ε-accurate.*

Unfortunately, computing exact RLS requires storing $\mathbf{K}_n$ and this is seldom possible in practice. In the next

[2] the matrix norm we use is the operator (induced) norm

Algorithm 1 The SQUEAK algorithm

```
Input: Dataset $\mathcal{D}$, parameters $\gamma, \varepsilon, \delta$
Output: $\mathcal{I}_n$
Initialize $\mathcal{I}_0$ as empty, $\overline{q}$ (see Thm. 1)
for $t = 1, \dots, n$ do
    Read point $\mathbf{x}_t$ from $\mathcal{D}$
    $\overline{\mathcal{I}} = \mathcal{I}_{t-1} \cup \{(t, \widetilde{p}_{t-1,t}=1, q_{t-1,t}=\overline{q})\}$   ▷EXPAND
    $\mathcal{I}_t =$ DICT-UPDATE($\overline{\mathcal{I}}$) using Eq. 4
end for
```

Subroutine 1 The DICT-UPDATE algorithm

```
Input: $\overline{\mathcal{I}}$
Output: $\mathcal{I}_t$
Initialize $\mathcal{I}_t = \emptyset$
for all $i \in \{1, \dots, t\}$ do   ▷SHRINK
    if $q_{t-1,i} \neq 0$ then
        Compute $\widetilde{\tau}_{t,i}$ using $\overline{\mathcal{I}}$
        Set $\widetilde{p}_{t,i} = \min\{\widetilde{\tau}_{t,i}, \widetilde{p}_{t-1,i}\}$
        Set $q_{t,i} \sim \mathcal{B}(\widetilde{p}_{t,i}/\widetilde{p}_{t-1,i}, q_{t-1,i})$
    else
        $\widetilde{p}_{t,i} = \widetilde{p}_{t-1,i}$ and $q_{t,i} = q_{t-1,i}$
    end if
end for
```

section, we introduce SQUEAK, an RLS-based incremental algorithm able to preserve the same accuracy of Prop. 1 *without* requiring to know the RLS in advance. We prove that it generates a dictionary only a constant factor larger than exact RLS sampling.

3 Sequential RLS Sampling

In the previous section, we showed that sampling proportionally to the RLS $\{\tau_{t,i}\}$ leads to a dictionary such that $\|\mathbf{P}_t - \widetilde{\mathbf{P}}_t\| \leq \varepsilon$. Furthermore, since the RLS correspond to the diagonal entries of $\mathbf{P}_t$, an accurate approximation $\widetilde{\mathbf{P}}_t$ may be used in turn to compute accurate estimates of $\tau_{t,i}$. The SQUEAK algorithm (Alg. 1) builds on this intuition to sequentially process the kernel matrix $\mathbf{K}_n$ so that exact RLS computed on a small matrix ($\mathbf{K}_t$ with $t \ll n$) are used to create an ε-accurate dictionary, which is then used to estimate the RLS for bigger kernels, which are in turn used to update the dictionary and so on. While SQUEAK shares a similar structure with INK-ESTIMATE [3], the sampling probabilities are computed from different estimates of the RLS $\tau_{t,i}$ and no renormalization by an estimate of $d_{\text{eff}}(\gamma)_t$ is needed. Before giving the details of the algorithm, we redefine a dictionary as a collection $\mathcal{I} = \{(i, \widetilde{p}_i, q_i)\}_i$, where i is the index of the point $\mathbf{x}_i$ stored in the dictionary, $\widetilde{p}_i$ tracks the probability used to sample it, and q_i is the number of copies (multiplicity) of i. The weights are then computed as $w_i = q_i/(\overline{q}\widetilde{p}_i)$, where $\overline{q}$ is an algorithmic parameter discussed later. We use $\widetilde{p}_i$ to stress the fact that these probabilities will be computed as approximations of the actual probabilities that should be used to sample each point, i.e., their RLS τ_i.

SQUEAK receives as input a dataset $\mathcal{D} = \{\mathbf{x}_t\}_{t=1}^n$ and processes it *sequentially.* Starting with an empty dictionary $\mathcal{I}_0$, at each time step t, SQUEAK receives a new point $\mathbf{x}_t$. Adding a new point $\mathbf{x}_t$ to the kernel matrix can either decrease the importance of points observed before (i.e., if they are correlated with the new point) or leave it unchanged (i.e., if their corresponding kernel columns are orthogonal) and thus for any $i \leq t$, the RLS evolves as follows.

Lemma 1. *For any kernel matrix $\mathbf{K}_{t-1}$ at time $t-1$ and its extension $\mathbf{K}_t$ at time t, we have that the RLS are monotonically decreasing and the effective dimension is monotonically increasing,*

$$\frac{1}{\tau_{t-1,i}+1}\tau_{t-1,i} \leq \tau_{t,i} \leq \tau_{t-1,i}, \quad d_{\mathit{eff}}(\gamma)_t \geq d_{\mathit{eff}}(\gamma)_{t-1}.$$

The previous lemma also shows that the RLS cannot decrease too quickly and since $\tau_{t-1,i} \leq 1$, they can at most halve when $\tau_{t-1,i} = 1$. After receiving the new point $\mathbf{x}_t$, we need to update our dictionary $\mathcal{I}_{t-1}$ to reflect the changes of the $\tau_{t,i}$. We proceed in two phases. During the EXPAND phase, we directly add the new element $\mathbf{x}_t$ to $\mathcal{I}_{t-1}$ and obtain a temporary dictionary $\overline{\mathcal{I}}$, where the new element t is added with a sampling probability $\widetilde{p}_{t-1,t} = 1$ and a number of copies $q_{t-1,t} = \overline{q}$, i.e., $\overline{\mathcal{I}} = \mathcal{I}_{t-1} \cup \{(t, \widetilde{p}_{t-1,t} = 1, q_{t-1,t} = \overline{q})\}$. This increases our memory usage, forcing us to update the dictionary using DICT-UPDATE, in order to decrease its size. Given as input $\overline{\mathcal{I}}$, we use the following estimator to compute the approximate RLS $\widetilde{\tau}_{t,i}$,

$$\begin{aligned}\widetilde{\tau}_{t,i} &= (1-\varepsilon)\boldsymbol{\phi}_i^\mathsf{T}(\boldsymbol{\Phi}_t\overline{\mathbf{S}}\overline{\mathbf{S}}^\mathsf{T}\boldsymbol{\Phi}_t^\mathsf{T} + \gamma\mathbf{I})^{-1}\boldsymbol{\phi}_i \\ &= \tfrac{1-\varepsilon}{\gamma}(k_{i,i} - \mathbf{k}_{t,i}^\mathsf{T}\overline{\mathbf{S}}(\overline{\mathbf{S}}^\mathsf{T}\mathbf{K}_t\overline{\mathbf{S}} + \gamma\mathbf{I}_t)^{-1}\overline{\mathbf{S}}^\mathsf{T}\mathbf{k}_{t,i}), \quad (4)\end{aligned}$$

where ε is the accuracy parameter, γ is the regularization and $\overline{\mathbf{S}}$ is the selection matrix associated to $\overline{\mathcal{I}}$. This estimator follows naturally from a reformulation of the RLS. In particular, if we consider $\boldsymbol{\phi}_i$, the RKHS representation of $\mathbf{x}_i$, the RLS $\tau_{t,i}$ can be formulated as $\tau_{t,i} = \boldsymbol{\phi}_i^\mathsf{T}(\boldsymbol{\Phi}_t\mathbf{I}_t\boldsymbol{\Phi}_t^\mathsf{T} + \gamma\mathbf{I})^{-1}\boldsymbol{\phi}_i$, where we see that the importance of point $\mathbf{x}_i$ is quantified by how orthogonal (in the RKHS) it is w.r.t. the other points. Because we do not have access to all the columns ($\overline{\mathbf{S}}\overline{\mathbf{S}}^\mathsf{T} \neq \mathbf{I}_t$), similarly to what [4] did for the special case $\boldsymbol{\phi}_i = \mathbf{x}_i$, we choose to use $\widetilde{\tau}_{t,i} \approx \boldsymbol{\phi}^\mathsf{T}(\boldsymbol{\Phi}_t\overline{\mathbf{S}}\overline{\mathbf{S}}^\mathsf{T}\boldsymbol{\Phi}_t^\mathsf{T} + \gamma\mathbf{I})^{-1}\boldsymbol{\phi}_i$, and then we use the kernel trick to derive a form that we can actually compute, resulting in Eq. 4. The approximate RLSs are then used to define the new sampling probabilities as $\widetilde{p}_{t,i} = \min\{\widetilde{\tau}_{t,i}, \widetilde{p}_{t-1,i}\}$. For each element in $\overline{\mathcal{I}}$, the SHRINK step draws a sample from the binomial $\mathcal{B}(\widetilde{p}_{t,i}/\widetilde{p}_{t-1,i}, q_{t-1,i})$, where the minimum taken in the definition of $\widetilde{p}_{t,i}$ ensures that the binomial probability is well defined (i.e., $\widetilde{p}_{t,i} \leq \widetilde{p}_{t-1,i}$). This resampling step basically *tracks* the changes in the RLS and constructs a new dictionary $\mathcal{I}_t$, which is *as if* it was created from scratch using all the RLS up to time t (with high probability). We see that the new element $\mathbf{x}_t$ is only added to the dictionary with a large number of copies (from 0 to $\overline{q}$) if its estimated relevance $\widetilde{p}_{t,t}$ is high, and that over time elements originally in $\mathcal{I}_{t-1}$ are stochastically reduced to reflect the reductions of the RLSs. The lower $\widetilde{p}_{t,i}$ w.r.t. $\widetilde{p}_{t-1,i}$, the lower the number of copies $q_{t,i}$ w.r.t. $q_{t-1,i}$. If the probability $\widetilde{p}_{t,i}$ continues to decrease over time, then $q_{t,i}$ may become zero, and the column i is completely dropped from the dictionary (by setting its weight to zero). The approximate RLSs enjoy the following guarantees.

Lemma 2. *Given an ε-approximate dictionary $\mathcal{I}_{t-1}$ of matrix $\mathbf{K}_{t-1}$, construct $\overline{\mathcal{I}}$ by adding element $(t, 1, \overline{q})$ to it, and compute the selection matrix $\overline{\mathbf{S}}$. Then for all i in $\overline{\mathcal{I}}$ such that $q_{t-1,i} \neq 0$, the estimator in Eq. 4 is α-accurate, i.e., it satisfies $\tau_{t,i}/\alpha \leq \widetilde{\tau}_{t,i} \leq \tau_{t,i}$, with $\alpha = (1+\varepsilon)/(1-\varepsilon)$. Moreover, given RLS $\tau_{t-1,i}$ and $\tau_{t,i}$, and two α-accurate RLSs, $\widetilde{\tau}_{t-1,i}$ and $\widetilde{\tau}_{t,i}$, the quantity* $\min\{\widetilde{\tau}_{t,i}, \widetilde{\tau}_{t-1,i}\}$ *is also an α-accurate RLS.*

This result is based on the property that whenever $\mathcal{I}_{t-1}$ is ε-accurate for $\mathbf{K}_{t-1}$, the projection matrix $\mathbf{P}_t$ can be approximated by $\overline{\mathbf{P}}_t$ constructed using the temporary dictionary $\overline{\mathcal{I}}$ and thus, the RLSs can be accurately estimated and used to update $\mathcal{I}_{t-1}$ and obtain a new ε-accurate dictionary for $\mathbf{K}_t$. Since $\widetilde{\tau}_{t,i}$ is used to sample the new dictionary $\mathcal{I}_t$, we need each point to be sampled *almost* as frequently as with the true RLS $\tau_{t,i}$, which is guaranteed by the lower bound of Lem. 2. Since RLSs are always smaller or equal than 1, this could be trivially achieved by setting $\widetilde{\tau}_{t,i}$ to 1. Nonetheless, this would retain all columns in the dictionary. Consequently, we need to force the RLS estimate to decrease as much as possible, so that low probabilities allow us to forget redundant samples and reduce the space as much as possible. This is obtained by the upper bound in Lem. 2, which guarantees that the estimated RLS are always smaller than the exact RLS. As a result, Shrink sequentially preserves the overall accuracy of the dictionary and *at the same time* keeps its size as small as possible, as shown in the following theorem.

Theorem 1. *Let $\varepsilon > 0$ be the accuracy parameter, $\gamma > 1$ the regularization, and $0 < \delta < 1$ the probability of failure. Given an arbitrary dataset $\mathcal{D}$ in input together with parameters ε, γ, and δ, we run* SQUEAK *with*

$$\overline{q} = \frac{39\alpha \log(2n/\delta)}{\varepsilon^2},$$

where $\alpha = (1+\varepsilon)/(1-\varepsilon)$. Then, w.p. at least $1-\delta$, SQUEAK *generates a sequence of random dictionaries $\{\mathcal{I}_t\}_{t=1}^n$ that are ε-accurate (Eq. 1) w.r.t. any of the intermediate kernels $\mathbf{K}_t$, and the size of the dictionaries is bounded as $\max_{t=1,\ldots,n} |\mathcal{I}_t| \leq 3\overline{q}d_{\mathit{eff}}(\gamma)_n$.*

As a consequence, on a successful run the overall complexity of SQUEAK *is bounded as*

$$\textit{space complexity} = \Big(\max_{t=1,\ldots,n} |\mathcal{I}_t|\Big)^2 \leq (3\overline{q}d_{\mathit{eff}}(\gamma)_n)^2,$$

$$\textit{time complexity} = \mathcal{O}\left(nd_{\mathit{eff}}(\gamma)_n^3\overline{q}^3\right).$$

We show later that Thm. 1 is special case of Thm. 2 and give a sketch of the proof with the statement of Thm. 2. We postpone an in-depth comparison with previous results to Sect. 6 and focus now on SQUEAK inclusion rule, and its space and time complexity.

Sequential RLS sampling using $\widetilde{\tau}_{t,i}$ greatly improves over approximate linear dependency (ALD) [6], or fixed-probability (uniform) inclusion [2]. In particular, uniform sampling must either reduce the inclusion probability over time, resulting in an algorithm that cannot adapt to changes in the input data distribution, or suffer a continuous increase in dictionary size. While it is possible to show that inclusion rules based on ALD can drop enough redundant samples and maintain a finite dictionary size [15], they cannot guarantee that these exclusions do not lead to catastrophic forgetting and loss of accuracy. SQUEAK not only detects when to increase the size of the dictionary to reflect an increase in the problem's complexity, but also guarantees an accurate reconstruction, closely matching that of batch sampling with exact RLS. Notice that for the time/space complexity and accuracy guarantees to hold $\overline{q}$ should be set proportionally to $\log(n)$, but in some cases we might not know n in advance, or wish to process additional data (e.g. up to $n' > n$ samples) after we computed $\mathcal{I}_n$. From the Theorem, we see that in this case the guarantees deteriorate gracefully, and still apply for a slightly larger $\varepsilon' > \varepsilon$ that grows only logarithmically in n'. In particular, processing $n' = 2n$ samples without changing $\overline{q}$ results only in a $\varepsilon' = (1 + \log(2)/\log(n))\varepsilon$ degraded accuracy.

While the dictionaries $\mathcal{I}_t$ always contain t elements for notational convenience, Shrink actually *never* updates the probabilities of the elements with $q_{t-1,i} = 0$. This feature is particularly important, since at any step t, it only requires to compute approximate RLSs for the elements which are actually included in $\mathcal{I}_{t-1}$ and the new point $\mathbf{x}_t$ (i.e., the elements in $\overline{\mathcal{I}}$) and thus it does not require recomputing the RLSs of points $\mathbf{x}_s$ $(s < t)$ that have been dropped before! This is why SQUEAK computes an ε-accurate dictionary with a *single pass over the dataset*. Furthermore, the esti-

Algorithm 2 The distributed SQUEAK algorithm

```
Input: Dataset $\mathcal{D}$, parameters $\gamma, \varepsilon, \delta$
Output: $\mathcal{I}_\mathcal{D}$
Partition $\mathcal{D}$ into disjoint sub-datasets $\mathcal{D}_i$
Initialize $\mathcal{I}_{\mathcal{D}_i} = \{(j, \widetilde{p}_{0,i} = 1, q_{0,i} = \overline{q}) : j \in \mathcal{D}_i\}$
Build set $\mathcal{S}_1 = \{\mathcal{I}_{\mathcal{D}_i}\}_{i=1}^k$
for $h = 1, \dots, k-1$ do
    if $|\mathcal{S}_h| > 1$ then                    ▷DICT-MERGE
        Pick two dictionaries $\mathcal{I}_\mathcal{D}, \mathcal{I}_{\mathcal{D}'}$ from $\mathcal{S}_h$
        $\overline{\mathcal{I}} = \mathcal{I}_\mathcal{D} \cup \mathcal{I}_{\mathcal{D}'}$
        $\mathcal{I}_{\mathcal{D},\mathcal{D}'} =$ DICT-UPDATE($\overline{\mathcal{I}}$) using Eq. 5
        Place $\mathcal{I}_{\mathcal{D},\mathcal{D}'}$ back into $\mathcal{S}_{h+1}$
    else
        $\mathcal{S}_{h+1} = \mathcal{S}_h$
    end if
end for
Return $\mathcal{I}_\mathcal{D}$, the last dictionary in $\mathcal{S}_k$
```

mator in Eq. 4 does not require computing the whole kernel column $\mathbf{k}_{t,i}$ of dimension t. In fact, the components of $\mathbf{k}_{t,i}$, corresponding to points which are no longer in $\overline{\mathcal{I}}$, are directly set to zero when computing $\mathbf{k}_{t,i}^{\mathsf{T}}\overline{\mathbf{S}}$. As a result, for any new point $\mathbf{x}_t$ we need to evaluate $\mathcal{K}(\mathbf{x}_s, \mathbf{x}_t)$ only for the indices s in $\overline{\mathcal{I}}$. Therefore, SQUEAK never performs more than $n(3\overline{q}d_{\text{eff}}(\gamma)_n)^2$ kernel evaluations, which means that it does not even need to observe large portions of the kernel matrix. Finally, the runtime is dominated by the n matrix inversions used in Eq. 4. Therefore, the total runtime is $\mathcal{O}(n\big(\max_{t=1,\dots,n}|\mathcal{I}_t|\big)^3) = \mathcal{O}(nd_{\text{eff}}(\gamma)_n^3\overline{q}^3)$. In the next section, we introduce DISQUEAK, which improves the runtime by independently constructing separate dictionaries in parallel and then merging them recursively to construct a final ε-accurate dictionary.

4 Distributed RLS Sampling

In this section, we show that a minor change in the structure of SQUEAK allows us to parallelize and distribute the computation of the dictionary $\mathcal{I}_n$ over multiple machines, thus reducing even further its time complexity. Beside the computational advantage, a distributed architecture is needed as soon as the input dimension d and the number of points n is so large that having the dataset on a single machine is impossible. Furthermore, distributed processing can reduce contention on bottleneck data sources such as databases or network connections. DIstributed-SQUEAK (DISQUEAK, Alg. 2) partitions $\mathcal{D}$ over multiple machines and the (small) dictionaries that are generated from different portions of the dataset are integrated in a hierarchical way. The initial dataset is partitioned over k disjoint sub-datasets $\mathcal{D}_i$ with $i = 1, \dots, k$ and k dictionaries $\mathcal{I}_{\mathcal{D}_i} = \{(j, \widetilde{p}_{0,i} = 1, q_{0,i} = \overline{q}) : j \in \mathcal{D}_i\}$ are initialized simply by placing all samples in $\mathcal{D}_i$ into $\mathcal{I}$ with weight 1 and multiplicity $\overline{q}$. Alternatively, if the datasets $\mathcal{D}_i$ are too large to fit in memory, we can run SQUEAK to generate the initial dictionaries. The dictionaries $\mathcal{I}_{\mathcal{D}_i}$ are added to a dictionary collection $\mathcal{S}_1$, and at each step $h \in [k]$ Alg. 2 arbitrarily chooses two dictionaries $\mathcal{I}_\mathcal{D}$ and $\mathcal{I}_{\mathcal{D}'}$ from $\mathcal{S}_h$, and merges them. DICT-MERGE first combines them into a single dictionary $\overline{\mathcal{I}}$ (the equivalent of the EXPAND phase in SQUEAK) and then DICT-UPDATE is run on the merged dictionaries to create an updated dictionary $\mathcal{I}_{\mathcal{D}\cup\mathcal{D}'}$, which is placed back in the dictionary collection $\mathcal{S}$. This sequence of merges can be represented using a binary merge tree, as in Fig. 1. Since DICT-MERGE only takes the two dictionaries as input and does not require any information on the dictionaries in the rest of the tree, separate branches can be run simultaneously on different machines, and only the resulting (small) dictionary needs to be propagated to the parent node for the future DICT-MERGE. Moreover, the tree does not need to be fixed in advance, and it can be adapted at run-time to compensate the fact that some dictionary merges will be slower than other. Unlike in SQUEAK, DICT-UPDATE is run on the union of two distinct dictionaries rather than one dictionary and a new single point. As a result, we need to derive the "distributed" counterparts of Lemmas 1 and 2 to analyze the behavior of the RLSs and the quality of the estimator.

Lemma 3. *Given two disjoint datasets $\mathcal{D}, \mathcal{D}'$, for every $i \in \mathcal{D} \cup \mathcal{D}'$, $\tau_{i,\mathcal{D}} \geq \tau_{i,\mathcal{D}\cup\mathcal{D}'}$ and*

$$2d_{\textit{eff}}(\gamma)_{\mathcal{D}\cup\mathcal{D}'} \geq d_{\textit{eff}}(\gamma)_\mathcal{D} + d_{\textit{eff}}(\gamma)_{\mathcal{D}'} \geq d_{\textit{eff}}(\gamma)_{\mathcal{D}\cup\mathcal{D}'}.$$

While in SQUEAK we were merging an ε-accurate dictionary $\mathcal{I}_t$ and a new point, which is equivalent to a perfect, 0-accurate dictionary, in DISQUEAK both dictionaries used in a merge are only ε-accurate. To balance this change, we introduce a new estimator,

$$\widetilde{\tau}_{\mathcal{D}\cup\mathcal{D}',i} = \tfrac{1-\varepsilon}{\gamma}(k_{i,i} - \mathbf{k}_i^{\mathsf{T}}\overline{\mathbf{S}}(\overline{\mathbf{S}}^{\mathsf{T}}\mathbf{K}\overline{\mathbf{S}} + (1+\varepsilon)\gamma\mathbf{I})^{-1}\overline{\mathbf{S}}^{\mathsf{T}}\mathbf{k}_i), \tag{5}$$

where $\overline{\mathbf{S}}$ is the selection matrix associated with the temporary dictionary $\overline{\mathcal{I}} = \mathcal{I}_\mathcal{D} \cup \mathcal{I}_{\mathcal{D}'}$. Eq. 5 has similar guarantees as Lem. 2, with only a slightly larger α.

Lemma 4. *Given two disjoint datasets $\mathcal{D}, \mathcal{D}'$, and two ε-approximate dictionaries $\mathcal{I}_\mathcal{D}$, $\mathcal{I}_{\mathcal{D}'}$, let $\overline{\mathcal{I}} = \mathcal{I}_\mathcal{D} \cup \mathcal{I}_{\mathcal{D}'}$ and $\overline{\mathbf{S}}$ be the associated selection matrix. Let $\mathbf{K}$ be the kernel matrix computed on $\mathcal{D} \cup \mathcal{D}'$, $\mathbf{k}_i$ its i-th column, and $\tau_{\mathcal{D}\cup\mathcal{D}',i}$ the RLS of $\mathbf{k}_i$. Then for all i in $\overline{\mathcal{I}}$ such that $q_i \neq 0$, the estimator in Eq. 5 is α-accurate, i.e. it satisfies $\tau_{\mathcal{D}\cup\mathcal{D}',i}/\alpha \leq \widetilde{\tau}_{\mathcal{D}\cup\mathcal{D}',i} \leq \tau_{\mathcal{D}\cup\mathcal{D}',i}$, with $\alpha = (1-\varepsilon)/(1+3\varepsilon)$.*

Given these guarantees, the analysis of DISQUEAK follows similar steps as SQUEAK. Given ε-accurate dictionaries, we obtain α-accurate RLS estimates

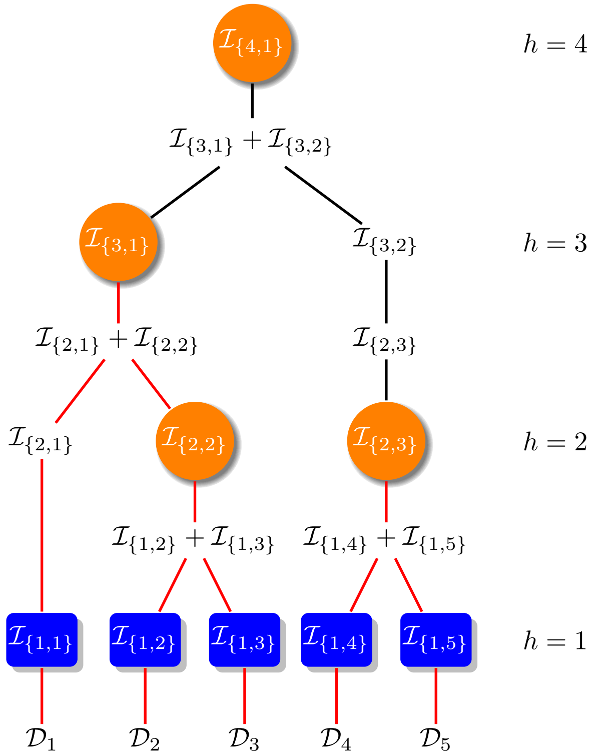


Figure 1: Merge tree for Alg. 2 with an arbitrary partitioning and merging scheme.

$\widetilde{\tau}_{\mathcal{D}\cup\mathcal{D}',i}$ that can be used to resample all points in $\overline{\mathcal{I}}$ and generate a new dictionary $\mathcal{I}_{\mathcal{D},\mathcal{D}'}$ that is ε-accurate. To formalize a result equivalent to Thm. 1, we introduce additional notation: We index each node in the merge tree by its height h and position l. We denote the dictionary associated to node $\{h,l\}$ by $\mathcal{I}_{\{h,l\}}$ and the collection of all dictionaries available at height h of the merge tree by $\mathcal{S}_h = \{\mathcal{I}_{\{h,l\}}\}$. We also use $\mathbf{K}_{\{h,l\}}$ to refer to the kernel matrix constructed from the datasets $\mathcal{D}_{\{h,l\}}$, which contains all points present in the leaves reachable from node $\{h,l\}$. For instance in Fig. 1, node $\{3,1\}$ is associated with $\mathcal{I}_{\{3,1\}}$, which is an ε-approximate dictionary of the kernel matrix $\mathbf{K}_{\{3,1\}}$ constructed from the dataset $\mathcal{D}_{\{3,1\}}$. $\mathcal{D}_{\{3,1\}}$ contains $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ (descendent nodes are highlighted in red) and it has dimension ($|\mathcal{D}_1|+|\mathcal{D}_2|+|\mathcal{D}_3|$). Theorem 2 summarizes the guarantees for DISQUEAK.

Theorem 2. *Let $\varepsilon > 0$ be the accuracy parameter, $\gamma > 1$ the regularization factor, and $0 < \delta < 1$ the prob. of failure. Given an arbitrary dataset $\mathcal{D}$ and a merge tree structure of height k as input together with parameters ε, γ, and δ, we run* DISQUEAK *with*

$$\overline{q} = \frac{39\alpha \log(2n/\delta)}{\varepsilon^2}.$$

where $\alpha = (1+3\varepsilon)/(1-\varepsilon)$. Then, w.p. at least $1-\delta$, DISQUEAK *generates a sequence of collections of dictionaries $\{\mathcal{S}_h\}_{h=1}^k$ such that each dictionary $\mathcal{I}_{\{h,l\}}$ in $\mathcal{S}_h$ is ε-accurate (Eq. 1) w.r.t. to $\mathbf{K}_{\{h,l\}}$, and that at any node l of height h the size of the dictionary is bounded as $|\mathcal{I}_{\{h,l\}}| \leq 3\overline{q}d_{\text{eff}}(\gamma)_{\{h,l\}}$. The cumulative (across nodes) space and time requirementsof the algorithm depend on the exact shape of the merge tree.*

Theorem 2 gives approximation and space guarantees for *every* node of the tree. In other words, it guarantees that each intermediate dictionary processed by DISQUEAK is both an ε-accurate approximation of the datasets used to generate it, and requires a small space proportional to the effective dimension of the same dataset. From an accuracy perspective, DISQUEAK provides exactly the same guarantees of SQUEAK. Analysing the complexity of DISQUEAK is however more complex, since the order and arguments of the DICT-UPDATE operations is determined by the merge tree. We distinguish between the time and work complexity of a tree by defining the time complexity as the amount of time necessary to compute the final solution, and the work complexity as the total amount of operations carried out by all machines in the tree in order to compute the final solution. We consider two special cases, a fully balanced tree (all inner nodes have either two inner nodes as children or two leaves), and a fully unbalanced tree (all inner nodes have exactly one inner node and one leaf as children). For both cases, we consider trees where each leaf dataset contains a single point $\mathcal{D}_i = \{\mathbf{x}_i\}$. In the fully unbalanced tree, we always merge the current dictionary with a new dataset (a single new point) and no DICT-MERGE operation can be carried out in parallel. Unsurprisingly, the sequential algorithm induced by this merge tree is strictly equivalent to SQUEAK. Computing a solution in the fully unbalanced tree takes $\mathcal{O}(nd_{\text{eff}}(\gamma)_{\mathcal{D}}^3\overline{q}^3)$ time with a total work that is also $\mathcal{O}(nd_{\text{eff}}(\gamma)_{\mathcal{D}}^3\overline{q}^3)$, as reported in Thm. 1. On the opposite end, the fully balanced tree needs to invert a $d_{\text{eff}}(\gamma)_{\{h,l\}}$ dimensional matrix at each layer of the tree for a total of $\log(n)$ layers. Bounding all $d_{\text{eff}}(\gamma)_{\{h,l\}}$ with $d_{\text{eff}}(\gamma)_{\mathcal{D}}$, gives a complexity for computing the final solution of $\mathcal{O}(\log(n)\overline{q}^3 d_{\text{eff}}(\gamma)_{\mathcal{D}}^3)$ time, with a huge improvement on SQUEAK. Surprisingly, the total work is only twice $\mathcal{O}(n\overline{q}^3 d_{\text{eff}}(\gamma)_{\mathcal{D}}^3)$, since at each layer h we perform $n/2^h$ inversions (on $n/2^h$ machines), and the sum across all layers is $\sum_{h=1}^{\log(n)} n/2^h \leq 2n$. Therefore, we can compute a solution in a much shorter time than SQUEAK, with a comparable amount of work, but at the expense of requiring much more memory across multiple machines, since at layer h, the sum $\sum_{l=1}^{|\mathcal{S}_h|} d_{\text{eff}}(\gamma)_{\{h,l\}}$ can be much larger than $d_{\text{eff}}(\gamma)_{\mathcal{D}}$. Nonetheless, this is partly alleviated by the fact that each node $\{h,l\}$ locally requires only $d_{\text{eff}}(\gamma)_{\{h,l\}}^2 \leq d_{\text{eff}}(\gamma)_{\mathcal{D}}^2$ memory.

Proof sketch: Although DISQUEAK is conceptually simple, providing guarantees on its space/time

	Time	$\lvert\mathcal{I}_n\rvert$ (Total space $= \mathcal{O}(n\lvert\mathcal{I}_n\rvert)$)	Increm.
Exact	n^3	n	-
Bach [2]	$\frac{n{d_{\max}}_n^2}{\varepsilon} + \frac{{d_{\max}}_n^3}{\varepsilon}$	$\frac{d_{\max,n}}{\varepsilon}$	No
Alaoui and Mahoney [1]	$n(\lvert\mathcal{I}_n\rvert)^3$	$\left(\frac{\lambda_{\min} + n\mu\varepsilon}{\lambda_{\min} - n\mu\varepsilon}\right) d_{\text{eff}}(\gamma)_n + \frac{\text{Tr}(\mathbf{K}_n)}{\mu\varepsilon}$	No
Calandriello et al. [3]	$\frac{\lambda_{\max}^3}{\gamma^3} \frac{n d_{\text{eff}}(\gamma)_n^3}{\varepsilon^2}$	$\frac{\lambda_{\max}}{\gamma} \frac{d_{\text{eff}}(\gamma)_n}{\varepsilon^2}$	Yes
SQUEAK	$\frac{n d_{\text{eff}}(\gamma)_n^3}{\varepsilon^2}$	$\frac{d_{\text{eff}}(\gamma)_n}{\varepsilon^2}$	Yes
RLS-sampling	$\frac{n d_{\text{eff}}(\gamma)_n^2}{\varepsilon^2}$	$\frac{d_{\text{eff}}(\gamma)_n}{\varepsilon^2}$	-

Table 1: Comparison of Nyström methods. $\lambda_{\max}$ and $\lambda_{\min}$ refer to largest and smallest eigenvalues of $\mathbf{K}_n$.

complexity and accuracy is far from trivial. The first step in the proof is to carefully decompose the failure event across the whole merge tree into separate failure events for each merge node $\{h, l\}$, and for each node construct a random process $\mathbf{Y}$ that models how Alg. 2 generates the dictionary $\mathcal{I}_{\{h,l\}}$. Notice that these processes are sequential in nature and the various steps (layers in the tree) are not i.i.d. Furthermore, the variance of $\mathbf{Y}$ is potentially large, and cannot be bounded uniformly. Instead, we take a more refined approach, inspired by Pachocki [11], that 1) uses Freedman's inequality to treat $\mathbf{W}$, the variance of process $\mathbf{Y}$, as a random object itself, 2) applies a stochastic dominance argument to $\mathbf{W}$ to reduce it to a sum of i.i.d. r.v. and only then we can 3) apply i.i.d. concentrations to obtain the desired result.

5 Applications

In this section, we show how our approximation guarantees translate into guarantees for typical kernel methods. As an example, we use kernel ridge regression. We begin by showing how to get an accurate approximation $\widetilde{\mathbf{K}}_n$ from an ε-accurate dictionary.

Lemma 5. *Given an ε-accurate dictionary $\mathcal{I}_t$ of matrix $\mathbf{K}_t$, and the selection matrix $\mathbf{S}_t$, the* regularized *Nyström approximation of $\mathbf{K}_t$ is defined as*

$$\widetilde{\mathbf{K}}_n = \mathbf{K}_n \mathbf{S}_n (\mathbf{S}_n^\mathsf{T} \mathbf{K}_n \mathbf{S}_n + \gamma \mathbf{I}_m)^{-1} \mathbf{S}_n^\mathsf{T} \mathbf{K}_n, \tag{6}$$

and satisfies

$$\mathbf{0} \preceq \mathbf{K}_t - \widetilde{\mathbf{K}}_t \preceq \frac{\gamma}{1-\varepsilon} \mathbf{K}_t (\mathbf{K}_t + \gamma \mathbf{I})^{-1} \preceq \frac{\gamma}{1-\varepsilon} \mathbf{I}. \tag{7}$$

This is not the only choice of an approximation from dictionary $\mathcal{I}_n$. For instance, Musco and Musco [10] show similar result for an unregularized Nyström approximation and Rudi et al. [13] for a smaller $\mathbf{K}_n$, construct the estimator only for the points in $\mathcal{I}_n$. Let $m = \lvert\mathcal{I}_n\rvert$, $\mathbf{W} = (\mathbf{S}_n^\mathsf{T} \mathbf{K}_n \mathbf{S}_n + \gamma \mathbf{I}_m) \in \mathbb{R}^{m \times m}$ is nonsingular and $\mathbf{C} = \mathbf{K}_n \mathbf{S}_n \in \mathbb{R}^{n \times m}$, applying the Woodbury formula we compute the regression weights as

$$\begin{aligned} \widetilde{\mathbf{w}}_n =& (\widetilde{\mathbf{K}}_n + \mu \mathbf{I}_n)^{-1} \mathbf{y}_n = (\mathbf{C} \mathbf{W}^{-1} \mathbf{C}^\mathsf{T} + \mu \mathbf{I}_n)^{-1} \mathbf{y}_n \\ =& \frac{1}{\mu} \left(\mathbf{y}_n - \mathbf{C} \left(\mathbf{C}^\mathsf{T} \mathbf{C} + \mu \mathbf{W} \right)^{-1} \mathbf{C}^\mathsf{T} \mathbf{y}_n \right). \end{aligned} \tag{8}$$

Computing $(\mathbf{C}^\mathsf{T}\mathbf{C} + \mu\mathbf{W})^{-1}$ takes $\mathcal{O}(nm^2)$ time to construct the matrix and $\mathcal{O}(m^3)$ to invert it, while the other matrix-matrix multiplication take at most $\mathcal{O}(nm^2)$ time. Overall, these operations require to store at most an $n \times m$ matrix. Therefore the final complexity of computing a KRR using the dictionary is reduced from $\mathcal{O}(n^3)$ to $\mathcal{O}(nm^2 + m^3)$ time, and from $\mathcal{O}(n^2)$ to $\mathcal{O}(nm)$ space. The following corollary provides guarantees for the empirical risk of the solution $\widetilde{\mathbf{w}}_n$ in a fixed design setting.

Corollary 1 ([1, Thm. 3])**.** *For an arbitrary dataset $\mathcal{D}$, let $\mathbf{K}$ be the kernel matrix constructed on $\mathcal{D}$. Run* SQUEAK *or* DISQUEAK *with regularization parameter γ. Then, the solution $\widetilde{\mathbf{w}}$ computed using the regularized Nyström approximation $\widetilde{\mathbf{K}}$ satisfies*

$$\mathcal{R}_\mathcal{D}(\widetilde{\mathbf{w}}) \leq \left(1 + \frac{\gamma}{\mu} \frac{1}{1-\varepsilon}\right)^2 \mathcal{R}_\mathcal{D}(\widehat{\mathbf{w}}),$$

where μ is the regularization of kernel ridge regression and $\mathcal{R}_\mathcal{D}(\widetilde{\mathbf{w}})$ is the empirical risk on $\mathcal{D}$.

For the random design setting, [13] provides a similar bound for batch RLS sampling, under some mild assumption on the kernel function $\mathcal{K}(\cdot,\cdot)$ and the dataset $\mathcal{D}$. Deriving identical guarantees for SQUEAK and DISQUEAK is straightforward.

Other applications. The projection $\mathbf{P}_n$ naturally appears in some form across nearly all kernel-based methods. Therefore, in addition to KRR in the fixed [1] and random [13] design setting, any kernel matrix approximation that provides ε-accuracy guarantees on $\mathbf{P}_n$ can be used to provide guarantees for a variety of other kernel problems. As an example, Musco and Musco [10] show this is the case for kernel

PCA [14], kernel CCA with regularization, and kernel K-means clustering.

6 Discussion

Tab. 1 compares several kernel approximation methods w.r.t. their space and time complexity. For all methods, we omit $\mathcal{O}(\log(n))$ factors. We first report RLS-SAMPLING, a fictitious algorithm that receives the exact RLSs as input, as an ideal baseline for all RLS sampling algorithms. The space complexity of uniform sampling [2] scales with the maximal degree of freedom $d_{\max}$. Since $d_{\max} = n \max_i \tau_{n,i} \geq \sum_i \tau_{n,i} = d_{\text{eff}}(\gamma)_n$, uniform sampling is often outperformed by RLS sampling. While Alaoui and Mahoney [1] also sample according to RLS, their two-pass estimator does not preserve the same level of accuracy. In particular, the first pass requires to sample $\mathcal{O}\left(n\mu\varepsilon/(\lambda_{\min} - n\mu\varepsilon)\right)$ columns, which quickly grows above n^2 when $\lambda_{\min}$ becomes small. Finally, Calandriello et al. [3] require that the maximum dictionary size is fixed in advance, which implies some information about the effective dimensions $d_{\text{eff}}(\gamma)_n$, and requires estimating both $\widetilde{\tau}_{t,i}$ and $\widetilde{d}_{\text{eff}}(\gamma)_t$. This extra estimation effort causes an additional $\lambda_{\max}/\gamma$ factor to appear in the space complexity. This factor cannot be easily estimated, and it leads to a space complexity of n^3 in the worst case. Therefore, we can see from the table that SQUEAK achieves the same space complexity (up to constant factors) as knowing the RLS in advance and hence outperforms previous methods.

A recent method by Musco and Musco [10] achieves comparable space and time guarantees as SQUEAK.[3] While they rely on a similar estimator, the two approaches are very different. Their method is batch in nature, as it processes the whole dataset at the same time and it *requires multiple passes* on the data for the estimation of the leverage scores and the sampling. On the other hand, SQUEAK is intrinsically sequential and it *only requires one single pass* on $\mathcal{D}$, as points are "forgotten" once they are dropped from the dictionary. Furthermore, the different structure requires remarkably different tools for the analysis. While the method of Musco and Musco [10] can directly use i.i.d. concentration inequalities (for the price of needing several passes), we need to rely on more sophisticated martingale arguments to consider the sequential stochastic process of SQUEAK. Furthermore, DISQUEAK smoothly extends the ideas of SQUEAK over distributed architectures, while it is not clear whether [10] could be parallelized or distributed.

Future developments Both SQUEAK and DISQUEAK need to know in advance the size of the dataset n to tune $\overline{q}$, which limits their applicability when the algorithm is run on data streams of unknown length.

Notice that nothing in the proof of SQUEAK requires the initial dictionary $\mathcal{I}_0$ to be empty. Therefore, a possible approach to avoid this dependency is to periodically (e.g. every time n doubles) freeze the current dictionary and use it as a fixed (no sample dropping) starting point for a new instance of SQUEAK that will run on the new data with a larger $\overline{q}$. Since by Lem. 3, $2d_{\text{eff}}(\gamma)_{\mathcal{D}\cup\mathcal{D}'} \geq d_{\text{eff}}(\gamma)_{\mathcal{D}} + d_{\text{eff}}(\gamma)_{\mathcal{D}'}$, this doubling trick would only introduce an extra logarithmic dependency on n in the size of the final dictionary. An interesting question is whether it is possible to adaptively increase the $\overline{q}$ parameter at runtime without restarts. This would allow us to continue updating $\mathcal{I}_n$ and indefinitely process new data beyond the initial dataset $\mathcal{D}$, and obtain an algorithm that can be truly applied to lifelong learning tasks.

It is also interesting to see whether SQUEAK could be used in conjunction with existing meta-algorithms (e.g., [8] with model averaging) for kernel matrix approximation that can leverage an accurate sampling scheme as a black-box, and what kind of improvements we could obtain.

Acknowledgements The research presented was supported by French Ministry of Higher Education and Research, Nord-Pas-de-Calais Regional Council and French National Research Agency projects ExTra-Learn (n.ANR-14-CE24-0010-01) and BoB (n.ANR-16-CE23-0003)

References

[1] Ahmed El Alaoui and Michael W. Mahoney. Fast randomized kernel methods with statistical guarantees. In *Neural Information Processing Systems*, 2015.

[2] Francis Bach. Sharp analysis of low-rank kernel matrix approximations. In *Conference on Learning Theory*, 2013.

[3] Daniele Calandriello, Alessandro Lazaric, and Michal Valko. Analysis of Nyström method with sequential ridge leverage scores. In *Uncertainty in Artificial Intelligence*, 2016.

[4] Michael B. Cohen, Cameron Musco, and Jakub Pachocki. Online row sampling. *International Workshop on Approximation, Randomization, and Combinatorial Optimization*, 2016.

[5] Michael B. Cohen, Cameron Musco, and Christopher Musco. Input sparsity time low-rank approximation via ridge leverage score sampling. In *Symposium on Discrete Algorithms*, 2017.

[3] The technical report of Musco and Musco [10] was developed independently from our work.

[6] Yaakov Engel, Shie Mannor, and Ron Meir. The kernel recursive least-squares algorithm. *IEEE Transactions on Signal Processing*, 52(8):2275–2285, 2004.

[7] Alex Gittens and Michael W Mahoney. Revisiting the Nyström method for improved large-scale machine learning. In *International Conference on Machine Learning*, 2013.

[8] Sanjiv Kumar, Mehryar Mohri, and Ameet Talwalkar. Sampling methods for the Nyström method. *Journal of Machine Learning Research*, 13(Apr):981–1006, 2012.

[9] Haim Levy. *Stochastic dominance: Investment decision making under uncertainty.* Springer, 2015.

[10] Cameron Musco and Christopher Musco. Provably useful kernel matrix approximation in linear time. Technical report, 2016. URL `http://arxiv.org/abs/1605.07583`.

[11] Jakub Pachocki. Analysis of resparsification. Technical report, 2016. URL `http://arxiv.org/abs/1605.08194`.

[12] Raajen Patel, Thomas A. Goldstein, Eva L. Dyer, Azalia Mirhoseini, and Richard G. Baraniuk. oASIS: Adaptive column sampling for kernel matrix approximation. Technical report, 2015. URL `http://arxiv.org/abs/1505.05208`.

[13] Alessandro Rudi, Raffaello Camoriano, and Lorenzo Rosasco. Less is more: Nyström computational regularization. In *Neural Information Processing Systems*, 2015.

[14] Bernhard Schölkopf, Alexander J. Smola, and Klaus-Robert Müller. Kernel principal component analysis. In *Advances in kernel methods*, pages 327–352. MIT Press Cambridge, MA, USA, 1999.

[15] Yi Sun, Jürgen Schmidhuber, and Faustino J. Gomez. On the size of the online kernel sparsification dictionary. In *International Conference on Machine Learning*, 2012.

[16] Joel Aaron Tropp. Freedman's inequality for matrix martingales. *Electronic Communications in Probability*, 16:262–270, 2011.

[17] Joel Aaron Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8(1-2):1–230, 2015.

[18] David P Woodruff et al. Sketching as a tool for numerical linear algebra. *Foundations and Trends in Theoretical Computer Science*, 10(1–2):1–157, 2014.

Notation summary

▷ Kernel matrix at time t $\mathbf{K}_t \in \mathbb{R}^{t\times t}$, i-th column $\mathbf{k}_{t,i} \in \mathbb{R}^t$, (i,j)-th entry $k_{i,j} \in \mathbb{R}$
▷ Eigendecomposition $\mathbf{K}_t = \mathbf{U}_t\mathbf{\Lambda}_t\mathbf{U}_t^\mathsf{T} \in \mathbb{R}^{t\times t}$, eigenvector matrix $\mathbf{U}_t \in \mathbb{R}^{t\times t}$ and diagonal SDP eigenvalues matrix $\mathbf{\Lambda}_t \in \mathbb{R}^{t\times t}$.
▷ Kernel matrix at time t in the RKHS, $\mathbf{K}_t = \mathbf{\Phi}_t^\mathsf{T}\mathbf{\Phi}_t$, with $\mathbf{\Phi}_t \in \mathbb{R}^{D\times t}$
▷ SVD decomposition $\mathbf{\Phi}_t = \mathbf{V}_t\mathbf{\Sigma}_t\mathbf{U}_t^\mathsf{T} \in \mathbb{R}^{D\times t}$ with left singular vectors, $\mathbf{V}_t \in \mathbb{R}^{D\times D}$, right singular vector $\mathbf{U}_t$ (same as $\mathbf{K}_t$), and singular values $\mathbf{\Sigma}_t \in \mathbb{R}^{D\times t}$ (σ_i on the main diagonal and zeros under it)
▷ SVD decomposition $\mathbf{\Phi}_t = \mathbf{V}_t\mathbf{\Sigma}_t\mathbf{U}_t^\mathsf{T} \in \mathbb{R}^{D\times t}$ with left singular vectors, $\mathbf{V}_t \in \mathbb{R}^{D\times D}$, right singular vector $\mathbf{U}_t$ (same as $\mathbf{K}_t$), and singular values $\mathbf{\Sigma}_t \in \mathbb{R}^{D\times t}$ (σ_i on the main diagonal and zeros under it)
▷ $\mathbf{P}_t = \mathbf{\Psi}_t\mathbf{\Psi}_t^\mathsf{T}$ with $\mathbf{\Psi}_t = (\mathbf{K}_t + \gamma\mathbf{I}_t)^{-1/2}\mathbf{K}_t^{1/2} \in \mathbb{R}^{t\times t}$
▷ Column dictionary at time t, $\mathcal{I}_t = \{(i, \widetilde{p}_{t,i}, q_{t,i})\}_{i=1}^t$
▷ Selection matrix $\mathbf{S}_t \in \mathbb{R}^{t\times t}$ with $\{\sqrt{\frac{q_{t,i}}{\overline{q}\widetilde{p}_{t,i}}}\}_{i=1}^t$ on the diagonal
▷ $\widetilde{\mathbf{P}}_t = \mathbf{\Psi}_t\mathbf{S}_t\mathbf{S}_t^\mathsf{T}\mathbf{\Psi}_t^\mathsf{T}$ with $\mathbf{S}_t \in \mathbb{R}^{t\times t}$

For node $\{h,l\}$ in the merge tree, given $\nu = |\mathcal{D}_{\{\mathbf{h},\mathbf{l}\}}|$
▷ Similarly defined $\mathbf{K}_{\{h,l\}} \in \mathbb{R}^{\nu\times\nu}, \mathbf{P}_{\{h,l\}} \in \mathbb{R}^{\nu\times\nu}$
▷ Similarly defined $\mathcal{I}_{\{h,l\}}, \mathbf{S}_{\{h,l\}} \in \mathbb{R}^{\nu\times\nu}, \widetilde{\mathbf{P}}_{\{h,l\}} \in \mathbb{R}^{\nu\times\nu}$
▷ For layer h in the merge tree, block diagonal matrices $\mathbf{K}^h, \mathbf{P}^h, \widetilde{\mathbf{P}}^h$ with $\mathbf{K}_{\{h,l\}}, \mathbf{P}_{\{h,l\}}, \widetilde{\mathbf{P}}_{\{h,l\}}$ on the diagonal
▷ $\mathbf{\Psi} = (\mathbf{K}_{\{h,l\}} + \gamma\mathbf{I}_\nu)^{-1/2}\mathbf{K}_{\{h,l\}}^{1/2} \in \mathbb{R}^{\nu\times\nu}$, with $\boldsymbol{\psi}_i$ its i-th column
▷ $\widetilde{\mathbf{P}}_s^{\{h,l\}} = \sum_{i=1}^{\nu\{h,l\}} \frac{q_{s,i}}{\overline{q}\widetilde{p}_{s,i}}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T} \in \mathbb{R}^{\nu\times\nu}$ constructed using weights $\widetilde{p}_{s,i}$ from step s of Alg. 2 and $\boldsymbol{\psi}_i$ from $\{h,l\}$
▷ $\mathbf{Y}_s = \mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_{\{h,l\}} = \mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_h^{\{h,l\}} \in \mathbb{R}^{\nu\times\nu}$ sampling process based on $\mathbf{K}_{\{h,l\}}$, at final step h
▷ $\mathbf{Y}_s = \mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_s^{\{h,l\}} \in \mathbb{R}^{\nu\times\nu}$ sampling process based on $\mathbf{K}_{\{h,l\}}$, at intermediate step s
▷ $\overline{\mathbf{Y}}_s \in \mathbb{R}^{\nu\times\nu}$ sampling process based on $\mathbf{K}_{\{h,l\}}$, with freezing
▷ $\mathbf{W}_h \in \mathbb{R}^{\nu\times\nu}$ total variance of sampling process based on $\mathbf{K}_{\{h,l\}}$, with freezing

A Preliminaries

In this section, we introduce standard matrix results and equivalent definitions for the kernel matrix and the the projection error which use convenient representations exploiting the feature space.

Matrix identity. We often use the following identity.

Proposition 2. *For any symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ and any $\gamma > 0$*

$$\mathbf{A}(\mathbf{A}^\mathsf{T}\mathbf{A} + \gamma\mathbf{I}_m)^{-1}\mathbf{A}^\mathsf{T} = \mathbf{A}\mathbf{A}^\mathsf{T}(\mathbf{A}\mathbf{A}^\mathsf{T} + \gamma\mathbf{I}_n)^{-1}.$$

For any symmetric matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ and diagonal matrix $\mathbf{B} \in \mathbb{R}^{n\times n}$ such that $\mathbf{B}$ has $n - s$ zero entries, and s non-zero entries, define $\mathbf{C} \in \mathbb{R}^{n\times s}$ as the matrix obtained by removing all zero columns in $\mathbf{B}$. Then

$$\mathbf{A}\mathbf{B}(\mathbf{B}\mathbf{A}\mathbf{B} + \gamma\mathbf{I}_m)^{-1}\mathbf{B}\mathbf{A} = \mathbf{A}\mathbf{C}(\mathbf{C}^\mathsf{T}\mathbf{A}\mathbf{C} + \gamma\mathbf{I}_s)^{-1}\mathbf{C}^\mathsf{T}\mathbf{A}.$$

For any appropriately shaped matrix $\mathbf{A}, \mathbf{B}, \mathbf{C}$, with $\mathbf{A}$ and $\mathbf{B}$ invertible, the Woodbury matrix identity states

$$(\mathbf{A} + \mathbf{C}\mathbf{B}\mathbf{C}^\mathsf{T})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{C}\left(\mathbf{C}^\mathsf{T}\mathbf{A}^{-1}\mathbf{C} + \mathbf{B}^{-1}\right)^{-1}\mathbf{C}^\mathsf{T}\mathbf{A}^{-1}.$$

Kernel matrix. Since $\mathbf{K}_t$ is real symmetric matrix, we can eigendecompose it as

$$\mathbf{K}_t = \mathbf{U}_t\mathbf{\Lambda}_t\mathbf{U}_t^\mathsf{T},$$

where $\mathbf{U}_t \in \mathbb{R}^{t\times t}$ is the eigenvector matrix and $\mathbf{\Lambda}_t \in \mathbb{R}^{t\times t}$ is the diagonal eigenvalue matrix, with all non-negative elements since $\mathbf{K}_t$ is PSD. Considering the feature mapping from $\mathcal{X}$ to the RKHS, we can write the kernel matrix as

$$\mathbf{K}_t = \mathbf{\Phi}_t^\mathsf{T}\mathbf{\Phi}_t,$$

where $\mathbf{\Phi}_t \in \mathbb{R}^{D\times t}$ is the feature matrix, whose SVD decomposition is

$$\mathbf{\Phi}_t = \mathbf{V}_t\mathbf{\Sigma}_t\mathbf{U}_t^\mathsf{T},$$

where $\mathbf{V}_t \in \mathbb{R}^{D \times D}$ contains the left singular vectors, $\mathbf{\Sigma}_t \in \mathbb{R}^{D \times t}$ has the singular values σ_i on the main diagonal (followed by zeros), and $\mathbf{U}_t$ contains the right singular vector, which coincide with the eigenvectors of $\mathbf{K}_t$. Furthermore, we have

$$\mathbf{\Lambda}_t = \mathbf{\Sigma}_t^\mathsf{T} \mathbf{\Sigma}_t,$$

and each eigenvalue $\lambda_i = \sigma^i$, with $i = 1, \dots, t$.

Projection error. We derive a convenient lemma on the formulation of the projection error both in terms of kernels and feature space.

Lemma 6. *The following identity holds.*

$$\begin{aligned}\left\|\mathbf{P}_t - \widetilde{\mathbf{P}}_t\right\|_2 &= \left\|(\mathbf{K}_t + \gamma \mathbf{I}_t)^{-1/2} \mathbf{K}_t^{1/2} (\mathbf{I}_t - \mathbf{S}_t \mathbf{S}_t^\mathsf{T}) \mathbf{K}_t^{1/2} (\mathbf{K}_t + \gamma \mathbf{I}_t)^{-1/2}\right\|_2 \\ &= \left\|(\mathbf{\Phi}_t \mathbf{\Phi}_t^\mathsf{T} + \gamma \mathbf{I}_D)^{-1/2} \mathbf{\Phi}_t (\mathbf{I}_t - \mathbf{S}_t \mathbf{S}_t^\mathsf{T}) \mathbf{\Phi}_t^\mathsf{T} (\mathbf{\Phi}_t \mathbf{\Phi}_t^\mathsf{T} + \gamma \mathbf{I}_D)^{-1/2}\right\|_2 .\end{aligned}$$

Proof of Lemma 6. Using the SVD decomposition,

$$\begin{aligned}&\left\|(\mathbf{K}_t + \gamma \mathbf{I}_t)^{-1/2} \mathbf{K}_t^{1/2} (\mathbf{I}_t - \mathbf{S}_t \mathbf{S}_t^\mathsf{T}) \mathbf{K}_t^{1/2} (\mathbf{K}_t + \gamma \mathbf{I}_t)^{-1/2}\right\|_2 \\ &= \left\|(\mathbf{U}_t \mathbf{\Lambda}_t \mathbf{U}^\mathsf{T} + \gamma \mathbf{U}_t \mathbf{I}_t \mathbf{U}_t^\mathsf{T})^{-1/2} \mathbf{U}_t \mathbf{\Lambda}_t^{1/2} \mathbf{U}^\mathsf{T} (\mathbf{I}_t - \mathbf{S}_t \mathbf{S}_t^\mathsf{T}) \mathbf{U}_t \mathbf{\Lambda}_t^{1/2} \mathbf{U}^\mathsf{T} (\mathbf{U}_t \mathbf{\Lambda}_t \mathbf{U}^\mathsf{T} + \gamma \mathbf{U}_t \mathbf{I}_t \mathbf{U}_t^\mathsf{T})^{-1/2}\right\|_2 \\ &= \left\|\mathbf{U}_t (\mathbf{\Lambda}_t + \gamma \mathbf{I}_t)^{-1/2} \mathbf{\Lambda}_t^{1/2} \mathbf{U}^\mathsf{T} (\mathbf{I}_t - \mathbf{S}_t \mathbf{S}_t^\mathsf{T}) \mathbf{U}_t \mathbf{\Lambda}_t^{1/2} (\mathbf{\Lambda}_t + \gamma \mathbf{I}_t)^{-1/2} \mathbf{U}^\mathsf{T}\right\|_2 \\ &= \left\|\mathbf{U}_t (\mathbf{\Lambda}_t + \gamma \mathbf{I}_t)^{-1/2} \mathbf{\Lambda}_t^{1/2} \mathbf{U}^\mathsf{T} (\mathbf{I}_t - \mathbf{S}_t \mathbf{S}_t^\mathsf{T}) \mathbf{U}_t \mathbf{\Lambda}_t^{1/2} (\mathbf{\Lambda}_t + \gamma \mathbf{I}_t)^{-1/2} \mathbf{U}^\mathsf{T}\right\|_2 \\ &= \left\|(\mathbf{\Lambda}_t + \gamma \mathbf{I}_t)^{-1/2} \mathbf{\Lambda}_t^{1/2} \mathbf{U}^\mathsf{T} (\mathbf{I}_t - \mathbf{S}_t \mathbf{S}_t^\mathsf{T}) \mathbf{U}_t \mathbf{\Lambda}_t^{1/2} (\mathbf{\Lambda}_t + \gamma \mathbf{I}_t)^{-1/2}\right\|_2 \\ &= \left\|(\mathbf{\Sigma}_t \mathbf{\Sigma}_t^\mathsf{T} + \gamma \mathbf{I}_D)^{-1/2} \mathbf{\Sigma}_t \mathbf{U}^\mathsf{T} (\mathbf{I}_t - \mathbf{S}_t \mathbf{S}_t^\mathsf{T}) \mathbf{U}_t \mathbf{\Sigma}_t^\mathsf{T} (\mathbf{\Sigma}_t \mathbf{\Sigma}_t^\mathsf{T} + \gamma \mathbf{I}_D)^{-1/2}\right\|_2 \\ &= \left\|\mathbf{V}_t (\mathbf{\Sigma}_t \mathbf{\Sigma}_t^\mathsf{T} + \gamma \mathbf{I}_D)^{-1/2} \mathbf{\Sigma}_t \mathbf{U}^\mathsf{T} (\mathbf{I}_t - \mathbf{S}_t \mathbf{S}_t^\mathsf{T}) \mathbf{U}_t \mathbf{\Sigma}_t^\mathsf{T} (\mathbf{\Sigma}_t \mathbf{\Sigma}_t^\mathsf{T} + \gamma \mathbf{I}_D)^{-1/2} \mathbf{V}_t^\mathsf{T}\right\|_2 \\ &= \left\|\mathbf{V}_t (\mathbf{\Sigma}_t \mathbf{U}_t \mathbf{U}_t^\mathsf{T} \mathbf{\Sigma}_t^\mathsf{T} + \gamma \mathbf{I}_D)^{-1/2} \mathbf{V}_t^\mathsf{T} \mathbf{V}_t \mathbf{\Sigma}_t \mathbf{U}^\mathsf{T} (\mathbf{I}_t - \mathbf{S}_t \mathbf{S}_t^\mathsf{T}) \mathbf{U}_t \mathbf{\Sigma}_t^\mathsf{T} \mathbf{V}^\mathsf{T} \mathbf{V} (\mathbf{\Sigma}_t \mathbf{U}_t \mathbf{U}_t^\mathsf{T} \mathbf{\Sigma}_t^\mathsf{T} + \gamma \mathbf{I}_D)^{-1/2} \mathbf{V}_t^\mathsf{T}\right\|_2 \\ &= \left\|(\mathbf{\Phi}_t \mathbf{\Phi}_t^\mathsf{T} + \gamma \mathbf{I}_D)^{-1/2} \mathbf{\Phi}_t (\mathbf{I}_t - \mathbf{S}_t \mathbf{S}_t^\mathsf{T}) \mathbf{\Phi}_t^\mathsf{T} (\mathbf{\Phi}_t \mathbf{\Phi}_t^\mathsf{T} + \gamma \mathbf{I}_D)^{-1/2}\right\|_2 .\end{aligned}$$

□

B Ridge Leverage Scores and Effective Dimension (Proof of Lemma 1 and 3)

From Calandriello et al. [3, Lem. 1] we know that $\tau_{t,i} \leq \tau_{t-1,i}$ and $d_{\text{eff}}(\gamma)_t \geq d_{\text{eff}}(\gamma)_{t-1}$. We will now prove the lower bound $\tau_{t,i} \geq \tau_{t-1,i}/(\tau_{t-1,1} + 1)$.

Considering the definition of $\tau_{t,i}$ in terms of $\boldsymbol{\phi}_i$ and $\mathbf{\Phi}_t$, and applying the Sherman-Morrison formula we obtain

$$\begin{aligned}\tau_{t,i} &= \boldsymbol{\phi}_i^\mathsf{T} (\mathbf{\Phi}_t \mathbf{\Phi}_t^\mathsf{T} + \gamma \mathbf{I})^{-1} \boldsymbol{\phi}_i = \boldsymbol{\phi}_i^\mathsf{T} (\mathbf{\Phi}_{t-1} \mathbf{\Phi}_{t-1}^\mathsf{T} + \boldsymbol{\phi}_t \boldsymbol{\phi}_t^\mathsf{T} + \gamma \mathbf{I})^{-1} \boldsymbol{\phi}_i \\ &= \boldsymbol{\phi}_i^\mathsf{T} (\mathbf{\Phi}_{t-1} \mathbf{\Phi}_{t-1}^\mathsf{T} + \gamma \mathbf{I})^{-1} \boldsymbol{\phi}_i - \frac{\boldsymbol{\phi}_i^\mathsf{T} (\mathbf{\Phi}_{t-1} \mathbf{\Phi}_{t-1}^\mathsf{T} + \gamma \mathbf{I})^{-1} \boldsymbol{\phi}_t \boldsymbol{\phi}_t^\mathsf{T} (\mathbf{\Phi}_{t-1} \mathbf{\Phi}_{t-1}^\mathsf{T} + \gamma \mathbf{I})^{-1} \boldsymbol{\phi}_i}{1 + \boldsymbol{\phi}_t^\mathsf{T} (\mathbf{\Phi}_{t-1} \mathbf{\Phi}_{t-1}^\mathsf{T} + \gamma \mathbf{I})^{-1} \boldsymbol{\phi}_t} \\ &= \tau_{t-1,i} - \frac{\boldsymbol{\phi}_i^\mathsf{T} (\mathbf{\Phi}_{t-1} \mathbf{\Phi}_{t-1}^\mathsf{T} + \gamma \mathbf{I})^{-1} \boldsymbol{\phi}_t \boldsymbol{\phi}_t^\mathsf{T} (\mathbf{\Phi}_{t-1} \mathbf{\Phi}_{t-1}^\mathsf{T} + \gamma \mathbf{I})^{-1} \boldsymbol{\phi}_i}{1 + \boldsymbol{\phi}_t^\mathsf{T} (\mathbf{\Phi}_{t-1} \mathbf{\Phi}_{t-1}^\mathsf{T} + \gamma \mathbf{I})^{-1} \boldsymbol{\phi}_t} .\end{aligned}$$

Let

$$\mathbf{x} = (\mathbf{\Phi}_{t-1} \mathbf{\Phi}_{t-1}^\mathsf{T} + \gamma \mathbf{I})^{-1/2} \boldsymbol{\phi}_i \quad \text{and} \quad \mathbf{y} = (\mathbf{\Phi}_{t-1} \mathbf{\Phi}_{t-1}^\mathsf{T} + \gamma \mathbf{I})^{-1/2} \boldsymbol{\phi}_t .$$

Then $\tau_{t,i}/\tau_{t-1,i}$ is equal to

$$\frac{\tau_{t,i}}{\tau_{t-1,i}} = 1 - \frac{(\boldsymbol{\phi}_t^\mathsf{T} (\mathbf{\Phi}_{t-1} \mathbf{\Phi}_{t-1}^\mathsf{T} + \gamma \mathbf{I})^{-1} \boldsymbol{\phi}_i)^2}{(1 + \boldsymbol{\phi}_t^\mathsf{T} (\mathbf{\Phi}_{t-1} \mathbf{\Phi}_{t-1}^\mathsf{T} + \gamma \mathbf{I})^{-1} \boldsymbol{\phi}_t) \boldsymbol{\phi}_i^\mathsf{T} (\mathbf{\Phi}_{t-1} \mathbf{\Phi}_{t-1}^\mathsf{T} + \gamma \mathbf{I})^{-1} \boldsymbol{\phi}_i} = 1 - \frac{(\mathbf{y}^\mathsf{T} \mathbf{x})^2}{(1 + \mathbf{y}^\mathsf{T} \mathbf{y})\, \mathbf{x}^\mathsf{T} \mathbf{x}} .$$

Defining the cosine between $\mathbf{y}$ and $\mathbf{x}$ as $\cos(\mathbf{y},\mathbf{x}) = \mathbf{y}^\mathsf{T}\mathbf{x}/(\|\mathbf{x}\|\|\mathbf{y}\|)$, we have that

$$1 - \frac{(\mathbf{y}^\mathsf{T}\mathbf{x})^2}{(1+\mathbf{y}^\mathsf{T}\mathbf{y})\,\mathbf{x}^\mathsf{T}\mathbf{x}} = 1 - \frac{\mathbf{y}^\mathsf{T}\mathbf{y}\mathbf{x}^\mathsf{T}\mathbf{x}\cos(\mathbf{y},\mathbf{x})^2}{(1+\mathbf{y}^\mathsf{T}\mathbf{y})\,\mathbf{x}^\mathsf{T}\mathbf{x}} = 1 - \frac{\|\mathbf{y}\|^2}{1+\|\mathbf{y}\|^2}\cos(\mathbf{y},\mathbf{x})^2,$$

where $\frac{\|\mathbf{y}\|^2}{1+\|\mathbf{y}\|^2}$ depends only on the norm of $\mathbf{y}$ and not its direction, and $\cos(\mathbf{y},\mathbf{x})$ depends only on the direction of $\mathbf{y}$ and is maximized when $\mathbf{y}=\mathbf{x}$. Therefore,

$$\frac{\tau_{t+1,i}}{\tau_{t,i}} = 1 - \frac{(\mathbf{y}^\mathsf{T}\mathbf{x})^2}{(1+\mathbf{y}^\mathsf{T}\mathbf{y})\,\mathbf{x}^\mathsf{T}\mathbf{x}} = 1 - \frac{\|\mathbf{y}\|^2}{1+\|\mathbf{y}\|^2}\cos(\mathbf{y},\mathbf{x})^2 \geq 1 - \frac{\|\mathbf{x}\|^2}{1+\|\mathbf{x}\|^2} = \frac{1}{1+\|\mathbf{x}\|^2} = \frac{1}{1+\tau_{t,i}},$$

which concludes the proof of Lem. 1.

For Lem. 3, the first point $\tau_{i,\mathcal{D}} \geq \tau_{i,\mathcal{D}\cup\mathcal{D}'}$ can be easily proven by choosing $\mathcal{D}$ to construct a kernel matrix $\mathbf{K}_\mathcal{D}$, and then invoke Lem. 1 as we add one sample at a time from $\mathcal{D}'$. Also as easily for $d_{\text{eff}}(\gamma)_\mathcal{D} + d_{\text{eff}}(\gamma)_{\mathcal{D}'}$ we have

$$d_{\text{eff}}(\gamma)_\mathcal{D} + d_{\text{eff}}(\gamma)_{\mathcal{D}'} \leq 2\max\{d_{\text{eff}}(\gamma)_\mathcal{D}; d_{\text{eff}}(\gamma)_{\mathcal{D}'}\} \leq 2\max\{d_{\text{eff}}(\gamma)_{\mathcal{D}\cup\mathcal{D}'}; d_{\text{eff}}(\gamma)_{\mathcal{D}\cup\mathcal{D}'}\} = 2d_{\text{eff}}(\gamma)_{\mathcal{D}\cup\mathcal{D}'}.$$

Finally, we prove the other side of the inequality for $d_{\text{eff}}(\gamma)_\mathcal{D} + d_{\text{eff}}(\gamma)_\mathcal{D}$. Let $\mathbf{\Phi}_\mathcal{D}, \mathbf{\Phi}_{\mathcal{D}'}$ be the matrices constructed using the feature vectors of the samples in $\mathcal{D}$ and $\mathcal{D}'$ respectively. Then,

$$\begin{aligned}
d_{\text{eff}}(\gamma)_\mathcal{D} + d_{\text{eff}}(\gamma)_{\mathcal{D}'} &= \sum_{i\in\mathcal{D}} \tau_{\mathcal{D},i} + \sum_{i\in\mathcal{D}'} \tau_{\mathcal{D}',i} = \sum_{i\in\mathcal{D}} \phi_i^\mathsf{T}(\mathbf{\Phi}_\mathcal{D}\mathbf{\Phi}_\mathcal{D}^\mathsf{T} + \gamma\mathbf{I}_D)^{-1}\phi_i + \sum_{i\in\mathcal{D}'} \phi_i^\mathsf{T}(\mathbf{\Phi}_{\mathcal{D}'}\mathbf{\Phi}_{\mathcal{D}'}^\mathsf{T} + \gamma\mathbf{I}_D)^{-1}\phi_i \\
&\geq \sum_{i\in\mathcal{D}} \phi_i^\mathsf{T}(\mathbf{\Phi}_{\mathcal{D}\cup\mathcal{D}'}\mathbf{\Phi}_{\mathcal{D}\cup\mathcal{D}'}^\mathsf{T} + \gamma\mathbf{I}_D)^{-1}\phi_i + \sum_{i\in\mathcal{D}'} \phi_i^\mathsf{T}(\mathbf{\Phi}_{\mathcal{D}\cup\mathcal{D}'}\mathbf{\Phi}_{\mathcal{D}\cup\mathcal{D}'}^\mathsf{T} + \gamma\mathbf{I}_D)^{-1}\phi_i \\
&= \sum_{i\in\mathcal{D}\cup\mathcal{D}'} \phi_i^\mathsf{T}(\mathbf{\Phi}_{\mathcal{D}\cup\mathcal{D}'}\mathbf{\Phi}_{\mathcal{D}\cup\mathcal{D}'}^\mathsf{T} + \gamma\mathbf{I}_D)^{-1}\phi_i = \sum_{i\in\mathcal{D}\cup\mathcal{D}'} \tau_{\mathcal{D}\cup\mathcal{D}',i} = d_{\text{eff}}(\gamma)_{\mathcal{D}\cup\mathcal{D}'}.
\end{aligned}$$

C Ridge Leverage Scores Estimation (Proof of Lemma 2 and 4)

We begin with a convenient reformulation of the ridge leverage scores,

$$\begin{aligned}
\tau_{t,i} &= \mathbf{e}_{t,i}^\mathsf{T}\mathbf{K}_t(\mathbf{K}_t + \gamma\mathbf{I}_t)^{-1}\mathbf{e}_{t,i} = \mathbf{e}_{t,i}^\mathsf{T}\mathbf{\Phi}_t^\mathsf{T}\mathbf{\Phi}_t(\mathbf{\Phi}_t^\mathsf{T}\mathbf{\Phi}_t + \gamma\mathbf{I}_t)^{-1}\mathbf{e}_{t,i} \\
&= \mathbf{e}_{t,i}^\mathsf{T}\mathbf{\Phi}_t^\mathsf{T}(\mathbf{\Phi}_t\mathbf{\Phi}_t^\mathsf{T} + \gamma\mathbf{I}_D)^{-1}\mathbf{\Phi}_t\mathbf{e}_{t,i} = \phi_i^\mathsf{T}(\mathbf{\Phi}_t\mathbf{\Phi}_t^\mathsf{T} + \gamma\mathbf{I}_D)^{-1}\phi_i.
\end{aligned}$$

This formulation, combined with Def. 1, suggests $\phi_i^\mathsf{T}(\mathbf{\Phi}_t\mathbf{S}_t\mathbf{S}_t^\mathsf{T}\mathbf{\Phi}_t^\mathsf{T} + \gamma\mathbf{I}_D)^{-1}\phi_i$ as an estimator for $\tau_{t,i}$. However, at step t, we only have access to an ε-accurate dictionary w.r.t. $\mathbf{\Phi}_{t-1}$ and not w.r.t. $\mathbf{\Phi}_t$. Therefore, we augment it with $(t,1,\overline{q})$ to construct $\overline{\mathcal{I}}_t$ and the corresponding $\overline{\mathbf{S}}_t$, which will have $[\overline{\mathbf{S}}_t]_{t,t} = 1$. We will now show how to implement this estimator efficiently. From Prop. 2, we apply Woodbury matrix identity with $\mathbf{A} = \gamma\mathbf{I}$, $\mathbf{B} = \mathbf{I}$ and $\mathbf{C} = \mathbf{\Phi}_t\overline{\mathbf{S}}_t$,

$$\begin{aligned}
\widetilde{\tau}_{t,i} &= (1-\varepsilon)\phi_i^\mathsf{T}(\mathbf{\Phi}_t\overline{\mathbf{S}}_t\overline{\mathbf{S}}_t^\mathsf{T}\mathbf{\Phi}_t^\mathsf{T} + \gamma\mathbf{I}_D)^{-1}\phi_i \\
&= (1-\varepsilon)\phi_i^\mathsf{T}(\mathbf{\Phi}_t\overline{\mathbf{S}}_t\mathbf{I}_t\overline{\mathbf{S}}_t^\mathsf{T}\mathbf{\Phi}_t^\mathsf{T} + \gamma\mathbf{I}_D)^{-1}\phi_i \\
(\mathit{Prop.}\ 2) &= (1-\varepsilon)\phi_i^\mathsf{T}\left(\frac{1}{\gamma}\mathbf{I}_D - \frac{1}{\gamma^2}\mathbf{\Phi}_t\overline{\mathbf{S}}_t\left(\frac{1}{\gamma}\overline{\mathbf{S}}_t^\mathsf{T}\mathbf{\Phi}_t^\mathsf{T}\mathbf{\Phi}_t\overline{\mathbf{S}}_t + \mathbf{I}_t\right)^{-1}\overline{\mathbf{S}}_t^\mathsf{T}\mathbf{\Phi}_t^\mathsf{T}\right)\phi_i \\
&= \frac{(1-\varepsilon)}{\gamma}\phi_i^\mathsf{T}\left(\mathbf{I}_D - \mathbf{\Phi}_t\overline{\mathbf{S}}_t\left(\overline{\mathbf{S}}_t^\mathsf{T}\mathbf{\Phi}_t^\mathsf{T}\mathbf{\Phi}_t\overline{\mathbf{S}}_t + \gamma\mathbf{I}_t\right)^{-1}\overline{\mathbf{S}}_t^\mathsf{T}\mathbf{\Phi}_t^\mathsf{T}\right)\phi_i \\
&= \frac{(1-\varepsilon)}{\gamma}\left(\phi_i^\mathsf{T}\phi_i - \phi_i^\mathsf{T}\mathbf{\Phi}_t\overline{\mathbf{S}}_t\left(\overline{\mathbf{S}}_t^\mathsf{T}\mathbf{\Phi}_t^\mathsf{T}\mathbf{\Phi}_t\overline{\mathbf{S}}_t + \gamma\mathbf{I}_t\right)^{-1}\overline{\mathbf{S}}_t^\mathsf{T}\mathbf{\Phi}_t^\mathsf{T}\phi_i\right) \\
&= \frac{(1-\varepsilon)}{\gamma}\left(k_{i,i} - \mathbf{k}_{t,i}\overline{\mathbf{S}}_t(\overline{\mathbf{S}}_t^\mathsf{T}\mathbf{K}_t\overline{\mathbf{S}}_t + \gamma\mathbf{I}_t)^{-1}\overline{\mathbf{S}}_t^\mathsf{T}\mathbf{k}_{t,i}\right),
\end{aligned}$$

which is the estimator defined in Eq. 4.

We can generalize this estimator to the case where instead of using a single dictionary and fresh data to estimate $\tau_{t,i}$, we are using k ε-accurate dictionaries $\mathcal{I}_k$. Given disjoint datasets $\{\mathcal{D}_i\}_{i=1}^k$ with associated feature matrices $\mathbf{\Phi}_i$. From each dataset, construct an ε-accurate dictionary $\mathcal{I}_i$, with its associated selection matrix $\mathbf{S}_i$.

To estimate the RLS τ_i of point i w.r.t. the whole dataset $\mathcal{D} = \cup_{j=1}^k \mathcal{D}_j$, and corresponding feature matrix $\mathbf{\Phi}$, we set the estimator to be

$$\widetilde{\tau}_i = (1-\varepsilon)\phi_i^\mathsf{T} \left(\sum_{j=1}^k \mathbf{\Phi}_j \mathbf{S}_j \mathbf{S}_j^\mathsf{T} \mathbf{\Phi}_j^\mathsf{T} + (1+(k-1)\varepsilon)\gamma \mathbf{I}_D \right)^{-1} \phi_i.$$

Part 1: accuracy of the RLS estimator $\widetilde{\tau}_i$. Since each of the dictionaries $\mathcal{I}_i$ used to generate $\mathbf{S}_i$ is ε-accurate, we can use the equivalence from Lem. 6,

$$\left\| \mathbf{P}_i - \widetilde{\mathbf{P}}_i \right\|_2 = \left\| (\mathbf{\Phi}_i \mathbf{\Phi}_i^\mathsf{T} + \gamma \mathbf{I}_D)^{-1/2} (\mathbf{\Phi}_i \mathbf{\Phi}_i^\mathsf{T} - \mathbf{\Phi}_i \mathbf{S}_i \mathbf{S}_i^\mathsf{T} \mathbf{\Phi}_i^\mathsf{T}) (\mathbf{\Phi}_i \mathbf{\Phi}_i^\mathsf{T} + \gamma \mathbf{I}_D)^{-1/2} \right\|_2 \leq \varepsilon,$$

which implies that

$$(1-\varepsilon)\mathbf{\Phi}_i \mathbf{\Phi}_i^\mathsf{T} - \varepsilon\gamma \mathbf{I}_D \preceq \mathbf{\Phi}_i \mathbf{S}_i \mathbf{S}_i^\mathsf{T} \mathbf{\Phi}_i^\mathsf{T} \preceq (1+\varepsilon)\mathbf{\Phi}_i \mathbf{\Phi}_i^\mathsf{T} + \varepsilon\gamma \mathbf{I}_D.$$

Therefore, we have

$$\begin{aligned}
\widetilde{\tau}_i &= (1-\varepsilon)\phi_i^\mathsf{T} \left(\sum_{j=1}^k \mathbf{\Phi}_j \mathbf{S}_j \mathbf{S}_j^\mathsf{T} \mathbf{\Phi}_j^\mathsf{T} + (1+(k-1)\varepsilon)\gamma \mathbf{I}_D \right)^{-1} \phi_i \\
&\leq (1-\varepsilon)\phi_i^\mathsf{T} \left(\sum_{j=1}^k \left((1-\varepsilon)\mathbf{\Phi}_j \mathbf{\Phi}_j^\mathsf{T} - \varepsilon\gamma \mathbf{I}_D \right) + (1+(k-1)\varepsilon)\gamma \mathbf{I}_D \right)^{-1} \phi_i \\
&\leq (1-\varepsilon)\phi_i^\mathsf{T} \left((1-\varepsilon)\mathbf{\Phi}\mathbf{\Phi}^\mathsf{T} - k\varepsilon\gamma \mathbf{I}_D + (1+(k-1)\varepsilon)\gamma \mathbf{I}_D \right)^{-1} \phi_i \\
&= (1-\varepsilon)\phi_i^\mathsf{T} \left((1-\varepsilon)(\mathbf{\Phi}\mathbf{\Phi}^\mathsf{T} + \gamma \mathbf{I}_D) \right)^{-1} \phi_i = \frac{(1-\varepsilon)}{(1-\varepsilon)} \phi_i^\mathsf{T} \left(\mathbf{\Phi}\mathbf{\Phi}^\mathsf{T} + \gamma \mathbf{I}_D \right)^{-1} \phi_i = \tau_i,
\end{aligned}$$

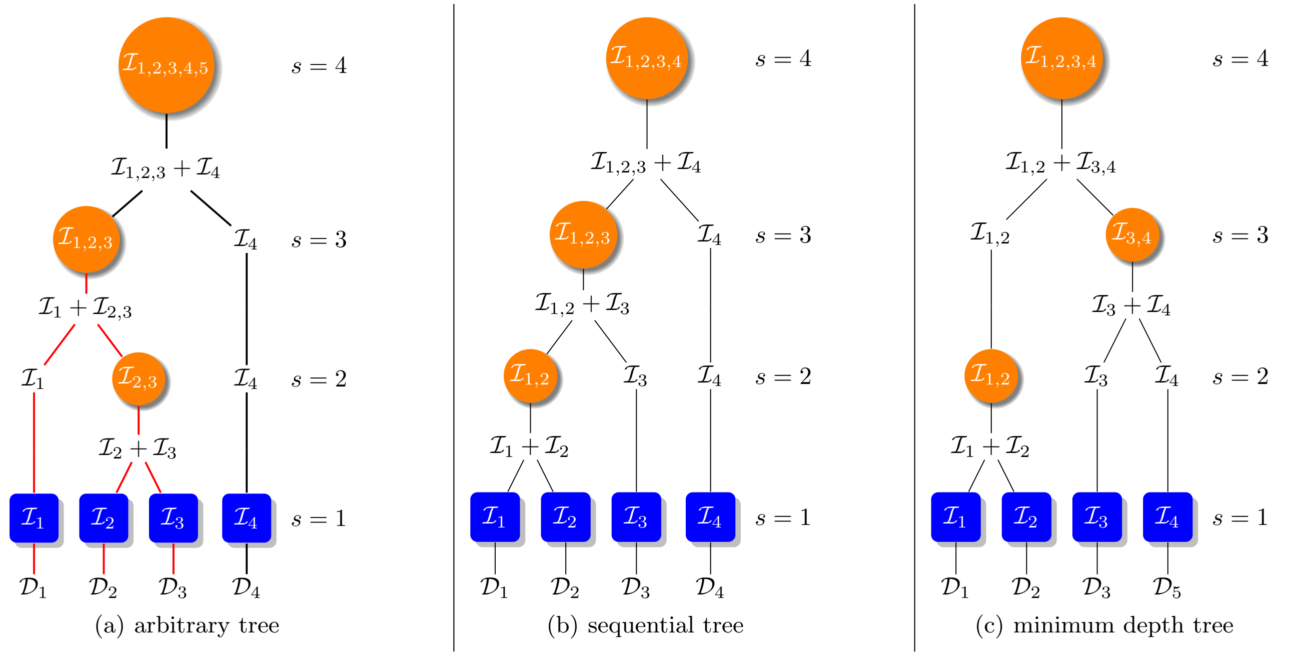
and

$$\begin{aligned}
\widetilde{\tau}_i &= (1-\varepsilon)\phi_i^\mathsf{T} \left(\sum_{j=1}^k \mathbf{\Phi}_j \mathbf{S}_j \mathbf{S}_j^\mathsf{T} \mathbf{\Phi}_j^\mathsf{T} + (1+(k-1)\varepsilon)\gamma \mathbf{I}_D \right)^{-1} \phi_i \\
&\geq (1-\varepsilon)\phi_i^\mathsf{T} \left(\sum_{j=1}^k \left((1+\varepsilon)\mathbf{\Phi}_j \mathbf{\Phi}_j^\mathsf{T} + \varepsilon\gamma \mathbf{I}_D \right) + (1+(k-1)\varepsilon)\gamma \mathbf{I}_D \right)^{-1} \phi_i \\
&= (1-\varepsilon)\phi_i^\mathsf{T} \left((1+\varepsilon)\mathbf{\Phi}\mathbf{\Phi}^\mathsf{T} + k\varepsilon\gamma \mathbf{I}_D + (1+(k-1)\varepsilon)\gamma \mathbf{I}_D \right)^{-1} \phi_i \\
&= (1-\varepsilon)\phi_i^\mathsf{T} \left((1+\varepsilon)\mathbf{\Phi}\mathbf{\Phi}^\mathsf{T} + (1+(2k-1)\varepsilon)\gamma \mathbf{I}_D) \right)^{-1} \phi_i \\
&\geq (1-\varepsilon)\phi_i^\mathsf{T} \left((1+(2k-1)\varepsilon)\mathbf{\Phi}\mathbf{\Phi}^\mathsf{T} + (1+(2k-1)\varepsilon)\gamma \mathbf{I}_D) \right)^{-1} \phi_i \\
&= \frac{(1-\varepsilon)}{(1+(2k-1)\varepsilon)} \phi_i^\mathsf{T} \left(\mathbf{\Phi}\mathbf{\Phi}^\mathsf{T} - \gamma \mathbf{I}_D \right)^{-1} \phi_i = \frac{(1-\varepsilon)}{(1+(2k-1)\varepsilon)} \tau_i.
\end{aligned}$$

Then, we can instantiate this result with $k = 1$ to prove the accuracy claim in Lem. 2, and with $k = 2$ to prove the accuracy claim in Lem. 4.

Part 2: accuracy of $\min\{\widetilde{\tau}_t,\ \widetilde{\tau}_{t-1}\}$. To simplify the notation, for this part of the proof we indicate with $\tau_t \leq \tau_{t-1}$ that for each $i \in \{1, \ldots, t-1\}$ we have $\tau_{t,i} \leq \tau_{t-1,i}$. From Lem. 1, we know that $\tau_{t-1} \geq \tau_t$. Given α-accurate $\widetilde{\tau}_t$ and $\widetilde{\tau}_{t-1}$ we have the upper bound

$$\min\{\widetilde{\tau}_t,\ \widetilde{\tau}_{t-1}\} \leq \min\{\tau_t,\ \tau_{t-1}\} = \tau_t,$$



(a) arbitrary tree

(b) sequential tree

(c) minimum depth tree

Figure 2: Merge trees for Algorithm 2.

and the lower bound,

$$\min\left\{\widetilde{\tau}_t,\ \widetilde{\tau}_{t-1}\right\} \geq \frac{1}{\alpha}\min\left\{\tau_t,\ \tau_{t-1}\right\} = \frac{1}{\alpha}\tau_t,$$

which combined gives us $\frac{1}{\alpha}\tau_t \leq \min\left\{\widetilde{\tau}_t,\ \widetilde{\tau}_{t-1}\right\} \leq \tau_t$ as required by the definition of α-accuracy.

D Proof of Thm. 1 and Thm. 2

From the discussion of Thm. 2, we know that running SQUEAK is equivalent to running DISQUEAK on a specific (fully unbalanced) merge tree. Therefore, we just prove Thm. 2, and invoke it on this tree to prove Thm. 1.

We begin by describing more in detail some notation introduced in the main paper and necessary for this proof.

Merge trees We first formalize the random process induced by Alg. 2.

We partition $\mathcal{D}$ into k disjoint sub-datasets $\mathcal{D}_i$ of size n_i, such that $\mathcal{D} = \cup_{i=1}^{k}\mathcal{D}_i$. For each dataset $\mathcal{D}_i$, we construct an initial dictionary $\mathcal{I}_{\{1,i\}} = \{(j, \widetilde{p}_{0,i} = 1, q_{0,i} = \overline{q}) : j \in \mathcal{D}_i\}$ by inserting all points from $\mathcal{D}_i$ into $\mathcal{I}_{\mathcal{D}_i}$ with weight $\widetilde{p}_{0,i} = 1$ and number of copies $q_{0,i} = \overline{q}$. It is easy to see that $\mathcal{I}_{\{1,i\}}$ is an ε-accurate dictionary, and we can split the data in small enough chunks to make sure that it can be easily stored and manipulated in memory. Alternatively, if we want our initial dictionaries to be small and cannot choose the size of $\mathcal{D}_i$, we can run Alg. 1 on $\mathcal{D}_i$ to generate $\mathcal{I}_{\{1,i\}}$, and the following proof will remain valid. Regardless of construction, the initial dictionaries $\mathcal{I}_{\{1,i\}}$ are included into the dictionary pool $\mathcal{S}_1$.

At iteration h, the inner loop of Alg. 2 arbitrarily chooses two dictionaries from $\mathcal{S}_h$ and merges them into a new dictionary. Any arbitrary sequence of merges can be described by a full binary tree, i.e., a binary tree where each node is either a leaf or has exactly two children. Figure 2 shows several different merge trees corresponding to different choices for the order of the merges. Note that starting from k leaves, a full binary tree will always have exactly $k-1$ internal nodes. Therefore, regardless of the structure of the merge tree, we can always transform it into a tree of depth k, with all the initial dictionaries $\mathcal{I}_{1,i}$ as leaves on its deepest layer. After this transformation, we index the tree nodes using their height (longest path from the node to a leaf, also defined as depth of the tree minus depth of the node), where leaves have height 1 and the root has height k. We can also see that at each layer, there is a single dictionary merge, and the size of $\mathcal{S}_h$ (number of dictionaries present at layer h) is $|\mathcal{S}_h| = k - h + 1$. Therefore, a node corresponding to a dictionary is uniquely identified with two indices $\{h, l\}$, where h is the height of the layer and $l \leq |\mathcal{S}_h|$ is the index of the node in the layer. For example, in Figure 2(a), the node containing $\mathcal{I}_{1,2,3}$ is indexed as $\{3,1\}$, and the highest node containing $\mathcal{I}_4$ is indexed as $\{3,2\}$.

We also define the dataset $\mathcal{D}_{\{h,l\}}$ as the union of all sub-datasets $\mathcal{D}_{l'}$ that are reachable from node $\{h,l\}$ as leaves. For example, in Fig. 2(a), dictionary $\mathcal{I}_{1,2,3}$ in node $\{3,1\}$ is constructed starting from all points in $\mathcal{D}_{\{3,1\}} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$, where we highlight in red the descendant tree. We now define $\mathbf{K}^h$ as the block diagonal kernel matrix where each diagonal block $\mathbf{K}_{\{h,l\}}$ is constructed on $\mathcal{D}_{\{h,l\}}$. Again, from Fig. 2, $\mathbf{K}^3$ is a $n \times n$ matrix with two blocks on the diagonal, a first $(n_1+n_2+n_3) \times (n_1+n_2+n_3)$ block $\mathbf{K}_{3,1}$ constructed on $\mathcal{D}_{\{3,1\}} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \mathcal{D}_3$, and a second $n_4 \times n_4$ block $\mathbf{K}_{3,2}$ constructed on $\mathcal{D}_{\{3,2\}} = \mathcal{D}_4$. Similarly, we can adapt Def. 1 to define $\mathbf{P}^h$ as a block diagonal projection matrix, where block $\mathbf{P}_{\{h,l\}}$ is defined using $\mathbf{K}_{\{h,l\}}$, and block diagonal $\widetilde{\mathbf{P}}^h$, where block $\widetilde{\mathbf{P}}_{\{h,l\}}$ is defined using $\mathbf{K}_{\{h,l\}}$ and $\mathcal{I}_{\{h,l\}}$.

The statement. Since $\mathbf{P}^h - \widetilde{\mathbf{P}}^h$ is block diagonal, we have that a bound on its largest eigenvalue implies an equal bound on each matrix on the diagonal, i.e.,

$$\|\mathbf{P}^h - \widetilde{\mathbf{P}}^h\| = \max_l \|\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_{\{h,l\}}\| \leq \varepsilon \Rightarrow \|\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_{\{h,l\}}\| \leq \varepsilon$$

for all blocks l on the diagonal, and since each block corresponds to a dictionary $\mathcal{I}_{\{h,l\}}$, this means that if $\|\mathbf{P}^h - \widetilde{\mathbf{P}}^h\| \leq \varepsilon$, all dictionaries at layer l are ε-accurate approximation of their respective represented datasets. Our goal is to show

$$\begin{aligned}
&\mathbb{P}\bigg(\exists h \in \{1,\dots,k\} : \|\mathbf{P}^h - \widetilde{\mathbf{P}}^h\|_2 \geq \varepsilon \;\cup\; \max_{l=1,\dots,|\mathcal{S}_h|} |\mathcal{I}_{\{h,l\}}| \geq 3\overline{q} d_{\text{eff}}(\gamma)_{\{h,l\}}\bigg)\\
&= \mathbb{P}\bigg(\exists h \in \{1,\dots,k\} : \underbrace{\bigg(\max_{l=1,\dots,|\mathcal{S}_h|} \|\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_{\{h,l\}}\|_2\bigg) \geq \varepsilon}_{A_h} \;\cup\; \underbrace{\bigg(\max_{l=1,\dots,|\mathcal{S}_h|} |\mathcal{I}_{\{h,l\}}| \geq 3\overline{q} d_{\text{eff}}(\gamma)_{\{h,l\}}\bigg)}_{B_h}\bigg) \leq \delta,
\end{aligned} \tag{9}$$

where event A_h refers to the case when some dictionary $\mathcal{I}_{\{h,l\}}$ at an intermediate layer h fails to accurately approximate $\mathbf{K}_{\{h,l\}}$ and event B_h considers the case when the memory requirement is not met (i.e., too many points are kept in one of the dictionaries $\mathcal{I}_{\{h,l\}}$ at a certain layer h). We can conveniently decompose the previous joint (negative) event into two separate conditions as

$$\begin{aligned}
\mathbb{P}\bigg(\bigcup_{h=1}^k A_h \cup B_h\bigg) &= \mathbb{P}\left(\left\{\bigcup_{h=1}^k A_h\right\} \cup \left\{\bigcup_{h=1}^k B_h\right\}\right) = \mathbb{P}\left(\left\{\bigcup_{h=1}^k A_h\right\}\right) + \mathbb{P}\left(\left\{\bigcup_{h=1}^k B_h\right\} \cap \left\{\bigcup_{h=1}^k A_h\right\}^{\mathsf{C}}\right)\\
&= \mathbb{P}\left(\left\{\bigcup_{h=1}^k A_h\right\}\right) + \mathbb{P}\left(\left\{\bigcup_{h=1}^k B_h\right\} \cap \left\{\bigcap_{h=1}^k A_h^{\mathsf{C}}\right\}\right) = \mathbb{P}\left(\left\{\bigcup_{h=1}^k A_h\right\}\right) + \mathbb{P}\left(\bigcup_{h=1}^k \left\{B_h \cap \left\{\bigcap_{h'=1}^k A_{h'}^{\mathsf{C}}\right\}\right\}\right).
\end{aligned}$$

Applying this reformulation and a union bound we obtain

$$\begin{aligned}
&\mathbb{P}\bigg(\exists h \in \{1,\dots,k\} : \|\mathbf{P}^h - \widetilde{\mathbf{P}}^h\|_2 \geq \varepsilon \;\cup\; \max_{l=1,\dots,|\mathcal{S}_h|} |\mathcal{I}_{\{h,l\}}| \geq 3\overline{q} d_{\text{eff}}(\gamma)_{\{h,l\}}\bigg)\\
&= \mathbb{P}\left(\exists h \in \{1,\dots,k\} : \bigg(\max_{l=1,\dots,|\mathcal{S}_h|} \|\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_{\{h,l\}}\|_2\bigg) \geq \varepsilon\right)\\
&\quad + \mathbb{P}\bigg(\exists h \in \{1,\dots,k\} : \max_{l=1,\dots,|\mathcal{S}_h|} |\mathcal{I}_{\{h,l\}}| \geq 3\overline{q} d_{\text{eff}}(\gamma)_{\{h,l\}} \cap \Big\{\forall h' \in \{1,\dots,h\} : \|\mathbf{P}^{h'} - \widetilde{\mathbf{P}}^{h'}\|_2 \leq \varepsilon\Big\}\bigg)\\
&\leq \sum_{h=1}^k \sum_{l=1}^{|\mathcal{S}_h|} \mathbb{P}\left(\|\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_{\{h,l\}}\|_2 \geq \varepsilon\right)\\
&\quad + \sum_{h=1}^k \sum_{l=1}^{|\mathcal{S}_h|} \mathbb{P}\left(|\mathcal{I}_{\{h,l\}}| \geq 3\overline{q} d_{\text{eff}}(\gamma)_{\{h,l\}} \cap \Big\{\forall h' \in \{1,\dots,h\} : \|\mathbf{P}^{h'} - \widetilde{\mathbf{P}}^{h'}\|_2 \leq \varepsilon\Big\}\right) \leq \delta.
\end{aligned} \tag{10}$$

As discussed in Sect. 3, the accuracy of the dictionary (first term in the previous bound) is guaranteed by the fact that given an ε-accurate dictionary we obtain RLS estimates which are at least a fraction of the true RLS, thus forcing the algorithm to sample each column *enough*. On the other hand, the space complexity bound is achieved by exploiting the fact that RLS estimates are always upper-bounded by the true RLS, thus ensuring that Alg. 2 does not oversample columns w.r.t. the sampling process following the exact RLS.

In the reminder of the proof, we will show that both events happen with probability smaller than $\delta/(2k^2)$. Since $|\mathcal{S}_h| = k - h + 1$, we have

$$\sum_{h=1}^{k}\sum_{l=1}^{|\mathcal{S}_h|}\frac{\delta}{2k^2} = \sum_{h=1}^{k}(k-h+1)\frac{\delta}{2k^2} = k(k+1)\frac{\delta}{4k^2} \leq k^2\frac{\delta}{2k^2} = \delta/2,$$

and the union bound over all events is smaller than δ. The main advantage of splitting the failure probability as we did in Eq. 10 is that we can now analyze the processes that generated each $\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_{\{h,l\}}$ (and each dictionary $\mathcal{I}_{\{h,l\}}$) separately. Focusing on a single node $\{h,l\}$ restricts our problem on a well defined dataset $\mathcal{D}_{\{h,l\}}$, where we can analyze the evolution of $\mathcal{I}_{\{h,l\}}$ sequentially.

Challenges. Due to its sequential nature, the "theoretical" structure of the process generating $\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_{\{h,l\}}$ is considerably complicated, where each step in the sampling process (layer in the merge tree) is highly correlated with the previous steps. This prevents us from using concentration inequalities for i.i.d. processes that are at the basis of the analysis of uniform sampling [2] and the method proposed by Alaoui and Mahoney [1]. As a result, we first show that the project error is a martingale process. The main difficulty technical difficulty in analyzing how the projection error evolves over iterations is that the projection matrices change dimension every time a new point is processed. In fact, all of the matrices $\mathbf{P}_{\{h',l'\}} - \widetilde{\mathbf{P}}_{\{h',l'\}}$ descending from $\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_{\{h,l\}}$ have potentially a different size since they are based on different kernel matrices $\mathbf{K}_{\{h',l'\}}$. This requires a careful definition of the martingale process to still use matrix concentration inequalities for *fixed-size* matrices (see Sect. D.1). Another major technical challenge is to control the variance of the martingale increments. In fact, at each the projection error may increase by a quantity whose cumulative variance can be arbitrarily large. As a result, a direct use of the Freedman matrix inequality in Sect. D.2 would not return an accurate result. In order to provide a tighter bound on the total variance of the martingale process of the projection error, we need to introduce an i.i.d. stochastically dominant process (Sect. D.3 (step 2)), which finally allows us to use an i.i.d. matrix concentration inequality to bound the total variance (Sect. D.3-(step 5)). This finally leads to the bound on the accuracy. The bound on the space complexity (Sect. D.4)follows similar (but simpler) steps.

D.1 Bounding the projection error $\|\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_{\{h,l\}}\|$

The sequential process. Thanks to the union bound in Eq. 10, instead of having to consider the whole merge tree followed by Alg. 2, we can focus on each individual node $\{h,l\}$ and study the sequential process that generated its dictionary $\mathcal{I}_{\{h,l\}}$. We will now map more clearly the actions taken by Alg. 2 to the process that generated $\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_{\{h,l\}}$. We begin by focusing on $\widetilde{\mathbf{P}}_{\{h,l\}}$, which is a random matrix defined starting from the fixed kernel matrix $\mathbf{K}_{\{h,l\}}$ and the random dictionary $\mathcal{I}_{\{h,l\}}$, where the randomness influences both which points are included in $\mathcal{I}_{\{h,l\}}$, and the weight with which they are added.

Consider now a point $i \in \mathcal{D}_{\{h,l\}}$. Since the starting datasets in the leaves are disjoint, there is a single path in the tree, with length h, from the leaves to $\{h,l\}$. This means that for all $s < h$, we can properly define a unique $\widetilde{p}_{s,i}$ and $q_{s,i}$ associated with that point. More in detail, if at layer s point i is present in $\mathcal{D}_{\{s,l'\}}$, it means that either (1) Alg. 2 used $\mathcal{I}_{\{s,l'\}}$ to compute $\widetilde{p}_{s,i}$, and $\widetilde{p}_{s,i}$ to compute $q_{s,i}$, or (2) at layer h Alg. 2 did not have any merge scheduled for point i, and we simply propagate $\widetilde{p}_{s,i} = \widetilde{p}_{s-1,i}$ and $q_{s,i} = q_{s-1,i}$. Consistently with the algorithm, we initialize $\widetilde{p}_{0,i} = 1$ and $q_{0,i} = \overline{q}$.

Denote $\nu_{\{h,l\}} = |\mathcal{D}_{\{h,l\}}|$ so that we can use index $i \in [\nu_{\{h,l\}}]$ to index all points in $\mathcal{D}_{\{h,l\}}$. Given the symmetric matrix $\mathbf{\Psi} = (\mathbf{K}_{\{h,l\}} + \gamma\mathbf{I})^{-1/2}\mathbf{K}_{\{h,l\}}^{1/2}$ with its i-th column $\boldsymbol{\psi}_i = (\mathbf{K}_{\{h,l\}} + \gamma\mathbf{I})^{-1/2}\mathbf{K}_{\{h,l\}}^{1/2}\mathbf{e}_{\nu_{\{h,l\}},i}$, we can rewrite the projection matrix as that $\mathbf{P}_{\{h,l\}} = \mathbf{\Psi}\mathbf{\Psi}^{\mathsf{T}} = \sum_{i=1}^{\nu_{\{h,l\}}}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}$. Note that

$$\|\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\| = \boldsymbol{\psi}_i^{\mathsf{T}}\boldsymbol{\psi}_i = \mathbf{e}_{\nu_{\{h,l\}},i}^{\mathsf{T}}\mathbf{\Psi}^{\mathsf{T}}\mathbf{\Psi}\mathbf{e}_{\nu_{\{h,l\}},i} = \mathbf{e}_{\nu_{\{h,l\}},i}^{\mathsf{T}}\mathbf{\Psi}\mathbf{\Psi}^{\mathsf{T}}\mathbf{e}_{\nu_{\{h,l\}},i} = \mathbf{e}_{\nu_{\{h,l\}},i}^{\mathsf{T}}\mathbf{P}_{\{h,l\}}\mathbf{e}_{\nu_{\{h,l\}},i} = \tau_{\mathcal{D}_{\{h,l\}},i},$$

or, in other words, the norm $\|\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\|$ is equal to the RLS of the i-th sample w.r.t. to dataset $\mathcal{D}_{\{h,l\}}$. Note that since i is present only in node l on layer h, its RLS is uniquely defined w.r.t. $\mathcal{D}_{\{h,l\}}$ and can be shortened as

$\tau_{h,i} = \tau_{\mathcal{D}_{\{h,l\}},i}$. Using $\boldsymbol{\psi}_i$, we can also introduce the random matrix $\widetilde{\mathbf{P}}_s^{\{h,l\}}$ as

$$\widetilde{\mathbf{P}}_s^{\{h,l\}} = \sum_{i=1}^{\nu_{\{h,l\}}} \frac{q_{s,i}}{\overline{q}\widetilde{p}_{s,i}} \boldsymbol{\psi}_i \boldsymbol{\psi}_i^\mathsf{T} = \sum_{i=1}^{\nu_{\{h,l\}}} \sum_{j=1}^{\overline{q}} \frac{z_{s,i,j}}{\overline{q}\widetilde{p}_{s,i}} \boldsymbol{\psi}_i \boldsymbol{\psi}_i^\mathsf{T}.$$

where $z_{s,i,j}$ are $\{0,1\}$ r.v. such that $q_{s,i} = \sum_{j=1}^{\overline{q}} z_{s,i,j}$, or in other words $z_{s,i,j}$ are the Bernoulli random variables that compose the Binomial $q_{s,i}$ associated with point i, with j indexing each individual copy of the point. Note that when $s = h$, we have that $\widetilde{\mathbf{P}}_h^{\{h,l\}} = \widetilde{\mathbf{P}}_{\{h,l\}}$ and we recover the definition of the approximate projection matrix from Alg. 2. But, for a general $s \neq h$ $\widetilde{\mathbf{P}}_s^{\{h,l\}}$ does not have a direct interpretation in the context of Alg. 2. It combines the vectors $\boldsymbol{\psi}_i$, which are defined using $\mathbf{K}_{\{h,l\}}$ at layer h, with the weights $\widetilde{p}_{s,i}$ computed by Alg. 2 across multiple nodes at layer s, which are potentially stored in different machines that cannot communicate. Nonetheless, $\widetilde{\mathbf{P}}_s^{\{h,l\}}$ is a useful tool to analyze Alg. 2.

Taking into account that we are now considering a specific node $\{h,l\}$, we can drop the index from the dataset $\mathcal{D}_{\{h,l\}} = \mathcal{D}$, RLS $\tau_{\mathcal{D}_{\{h,l\}},i} = \tau_{h,i}$, and size $\nu_{\{h,l\}} = \nu$. Using this shorter notation, we can reformulate our objective as bounding $\|\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_{\{h,l\}}\|_2 = \|\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_h^{\{h,l\}}\|_2$, and reformulate the process as a sequence of matrices $\{\mathbf{Y}_s\}_{s=1}^h$ defined as

$$\mathbf{Y}_s = \mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_s^{\{h,l\}} = \frac{1}{\overline{q}} \sum_{i=1}^{\nu} \sum_{j=1}^{\overline{q}} \left(1 - \frac{z_{s,i,j}}{\widetilde{p}_{s,i}}\right) \boldsymbol{\psi}_i \boldsymbol{\psi}_i^\mathsf{T},$$

where $\mathbf{Y}_h = \mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_h^{\{h,l\}} = \mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_{\{h,l\}}$, and $\mathbf{Y}_1 = \mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_0^{\{h,l\}} = \mathbf{0}$ since $\widetilde{p}_{0,i} = 1$ and $q_{0,i} = \overline{q}$.

D.2 Bounding $\mathbf{Y}_h$

We transformed the problem of bounding $\|\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_{\{h,l\}}\|$ into the problem of bounding $\mathbf{Y}_h$, which we modeled as a random matrix process, connected to Alg. 2 by the fact that both algorithm and random process $\mathbf{Y}_h$ make use of the same weight $\widetilde{p}_{s,i}$ and multiplicities $q_{s,i}$. There are two main issues in analyzing the process $\{\mathbf{Y}_s\}_{s=1}^h$:

(1) While the overall algorithm may fail in generating an accurate dictionary at some intermediate iteration and yet return an accurate dictionary at the end, whenever one of the intermediate $\|\mathbf{Y}_s\|$ is larger than ε we lose our guarantees for $\widetilde{p}_{s,i}$ for the whole process, since an inaccurate $\widetilde{p}_{s,i}$ that underestimates too much $p_{s.i}$ will influence all successive $\widetilde{p}_{s',i}$ through the minimum. To solve this, we consider an alternative (more pessimistic) process which is "frozen" as soon as it constructs an inaccurate dictionary. Freezing the probabilities at the first error gives us a process that fails more often, but that provides strong guarantees up to the moment of failure.

(2) While $\|\mathbf{Y}_h\| = \|\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_h^{\{h,l\}}\| \leq \varepsilon$ guarantees that $\mathcal{I}_{\{h,l\}}$ is an ε-accurate dictionary of $\mathbf{K}_{\{h,l\}}$, knowing $\|\mathbf{Y}_s\| = \|\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_s^{\{h,l\}}\| \leq \varepsilon$ for $s < h$ does not guarantee that all descendant $\mathcal{I}_{\{s,l'\}}$ are ε-accurate w.r.t. their $\mathbf{K}_{\{s,l'\}}$. Nonetheless, we will show that $\|\mathbf{Y}_s\| \leq \varepsilon$ is enough to guarantee that the intermediate estimate $\widetilde{p}_{s,i}$ computed in Alg. 2, and used as weights in $\widetilde{\mathbf{P}}_s^{\{h,l\}}$, are never too small.

The frozen process. We will now replace the process $\mathbf{Y}_s$ with an alternative process $\overline{\mathbf{Y}}_s$ defined as

$$\overline{\mathbf{Y}}_s = \mathbf{Y}_{s-1}\mathbb{I}\left\{\|\overline{\mathbf{Y}}_{s-1}\| \leq \varepsilon\right\} + \overline{\mathbf{Y}}_{s-1}\mathbb{I}\left\{\|\overline{\mathbf{Y}}_{s-1}\| \geq \varepsilon\right\}.$$

This process starts from $\overline{\mathbf{Y}}_0 = \mathbf{Y}_0 = \mathbf{0}$, and is identical to $\mathbf{Y}_s$ until a step $\overline{s}$ where for the first time $\|\mathbf{Y}_{\overline{s}}\| \leq \varepsilon$ and $\|\mathbf{Y}_{\overline{s}+1}\| \geq \varepsilon$. After this failure happen the process $\overline{\mathbf{Y}}_s$ is "frozen" at $\overline{s}$ and $\overline{\mathbf{Y}}_s = \mathbf{Y}_{\overline{s}+1}$ for all $\overline{s}+1 \leq s \leq h$. Consequently, if any of the intermediate elements of the sequence violates the condition $\|\mathbf{Y}_s\| \leq \varepsilon$, the last element will violate it too. For the rest, $\overline{\mathbf{Y}}_s$ behaves exactly like $\mathbf{Y}_s$. Therefore,

$$\mathbb{P}\left(\|\mathbf{Y}_h\| \geq \varepsilon\right) \leq \mathbb{P}\Big(\|\overline{\mathbf{Y}}_h\| \geq \varepsilon\Big),$$

and if we can bound $\mathbb{P}\Big(\|\overline{\mathbf{Y}}_h\| \geq \varepsilon\Big)$ we will have a bound for the failure probability of Alg. 2, even though after "freezing" the process $\overline{\mathbf{Y}}_h$ does not make the same choices as the algorithm.

We will see now how to construct the process $\overline{\mathbf{Y}}_s$ starting from $z_{s,i,j}$ and $\widetilde{p}_{s,i,j}$. We recursively define the indicator ($\{0,1\}$) random variable $\overline{z}_{s,i,j}$ as

$$\overline{z}_{s,i,j} = \mathbb{I}\left\{ u_{s,i,j} \leq \frac{\overline{p}_{s,i,j}}{\overline{p}_{s-1,i,j}} \right\} \overline{z}_{s-1,i,j},$$

where $u_{s,i,j} \sim \mathcal{U}(0,1)$ is a $[0,1]$ uniform random variable and $\overline{p}_{s,i,j}$ is defined as

$$\overline{p}_{s,i,j} = \widetilde{p}_{s,i}\mathbb{I}\left\{ \|\overline{\mathbf{Y}}_{s-1}\| \leq \varepsilon \cap z_{s-1,i,j} = 1 \right\} + \overline{p}_{s-1,i,j}\mathbb{I}\left\{ \|\overline{\mathbf{Y}}_{s-1}\| \geq \varepsilon \cup z_{s-1,i,j} = 0 \right\}.$$

This definition of the process satisfies the freezing condition, since if $\|\mathbf{Y}_{\overline{s}+1}\| \geq \varepsilon$ (we have a failure at step $\overline{s}$), for all $s' \geq \overline{s}+1$ we have $\overline{z}_{s',i,j} = \overline{z}_{\overline{s}+1,i,j}$ with probability 1 ($\overline{p}_{\overline{s}+1,i,j}/\overline{p}_{\overline{s},i,j} = \overline{p}_{\overline{s},i,j}/\overline{p}_{\overline{s},i,j} = 1$), and the weights $1/(\overline{q}\overline{p}_{\overline{s}+1,i,j}) = 1/(\overline{q}\overline{p}_{\overline{s},i,j})$ never change.

Introducing a per-copy weight $\overline{p}_{s,i,j}$ and enforcing that $\overline{p}_{s+1,i,j} = \overline{p}_{s,i,j}$ when $z_{s,i,j} = 0$ avoids subtle inconsistencies in the formulation. In particular, not doing so would semantically correspond to reweighting dropped copies. Although this does not directly affect $\mathbf{Y}_s$ (since the ratio $z_{s,i,j}/\widetilde{p}_{s,i}$ is zero for dropped copies), and therefore the relationship $\mathbb{P}\left(\|\mathbf{Y}_h\| \geq \varepsilon\right) \leq \mathbb{P}\Big(\|\overline{\mathbf{Y}}_h\| \geq \varepsilon\Big)$ still holds. We will see later how maintaining consistency helps us bound the second moment of our process.

We can now arrange the indices s, i, and j into a linear index $r = s$ in the range $[1, \ldots, \nu^2\overline{q}]$, obtained as $r = \{s,i,j\} = (s-1)\nu\overline{q} + (i-1)\overline{q} + j$. We also define the difference matrix as

$$\overline{\mathbf{X}}_{\{s,i,j\}} = \frac{1}{\overline{q}}\left(\frac{z_{s-1,i,j}}{\overline{p}_{s-1,i,j}} - \frac{z_{s,i,j}}{\overline{p}_{s,i,j}} \right) \boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T},$$

which allows us to write the cumulative matrix as $\overline{\mathbf{Y}}_{\{s,i,j\}} = \sum_{r=1}^{\{s,i,j\}} \overline{\mathbf{X}}_{\{s,i,j\}}$ where the checkpoints $\{s,\nu,\overline{q}\}$ correspond to $\overline{\mathbf{Y}}_s$,

$$\overline{\mathbf{Y}}_{\{s,\nu,\overline{q}\}} = \overline{\mathbf{Y}}_s = \frac{1}{\overline{q}} \sum_{i=1}^{\nu}\sum_{j=1}^{\overline{q}} \left(1 - \frac{z_{s,i,j}}{\overline{p}_{s,i,j}} \right) \boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T}.$$

Let $\mathcal{F}_s$ be the filtration containing all the realizations of the uniform random variables $u_{s,i,j}$ up to the step s, that is $\mathcal{F}_s = \{u_{s',i',j'}, \forall\{s',i',j'\} \leq s\}$. Again, we notice that $\mathcal{F}_s$ defines the state of the algorithm after completing iteration s because, unless a "freezing" happened, Alg. 2 and $\overline{\mathbf{Y}}_s$ flip coins with the same probability, and generate the same dictionaries. Since $\widetilde{\tau}_{s,i}$ and $\overline{p}_{s,i,j}$ are computed at the beginning of iteration s using the dictionary $\mathcal{I}_{\{s,l'\}}$ (for some l' unique at layer s), they are fully determined by $\mathcal{F}_{s-1}$. Furthermore, since $\mathcal{F}_{s-1}$ also defines the values of all indicator variables $\overline{z}_{s',i,j}$ up to $\overline{z}_{s-1,i,j}$ for any i and j, we have that all the Bernoulli variables $\overline{z}_{s,i,j}$ at iteration s are conditionally independent given $\mathcal{F}_{s-1}$. In other words, we have that for any i', and j' such that $\{s,1,1\} \leq \{s,i',j'\} < s$ the following random variables are equal in distribution,

$$\overline{z}_{s,i,j}\big|\mathcal{F}_{\{s,i',j'\}} = \overline{z}_{s,i,j}\big|\mathcal{F}_{\{s-1,\nu,\overline{q}\}} \sim \mathcal{B}\Big(\frac{\overline{p}_{s,i,j}}{\overline{p}_{s-1,i,j}}\Big), \tag{11}$$

and for any i', and j' such that $\{s,1,1\} \leq \{s,i',j'\} \leq \{s,\nu,\overline{q}\}$ and $s \neq \{s,i',j'\}$ we have the independence

$$\overline{z}_{s,i,j}\big|\mathcal{F}_{\{s-1,\nu,\overline{q}\}} \perp \overline{z}_{s,i',j'}\big|\mathcal{F}_{\{s-1,\nu,\overline{q}\}}. \tag{12}$$

While knowing that $\|\mathbf{Y}_s\| \leq \varepsilon$ is not sufficient to provide guarantees for the approximate probabilities $\widetilde{p}_{s,i}$, we can show that it is enough to prove that the frozen probabilities $\overline{p}_{s,i,j}$ are never too small.

Lemma 7. *Let $\alpha = (1+3\varepsilon)/(1-\varepsilon)$ and $\overline{p}_{s,i,j}$ be the sequence of probabilities generated by the freezing process. Then for any s, i, and j, we have $\overline{p}_{s,i,j} \geq p_{h,i}/\alpha = \tau_{h,i}/\alpha$.*

Proof of Lemma 7. Let $\overline{s}$ be the step where the process freezes ($\overline{s} = h$ if it does not freeze), or, in other words, $\|\mathbf{Y}_{\overline{s}}\| < \varepsilon$ and $\|\mathbf{Y}_{\overline{s}+1}\| \geq \varepsilon$. From the definition of $\overline{p}_{s,i,j}$, we have that

$$\begin{aligned}\overline{p}_{s,i,j} &\geq \overline{p}_{\overline{s},i} = \widetilde{p}_{\overline{s},i} = \max\left\{ \min\left\{ \widetilde{\tau}_{\overline{s},i},\ \widetilde{p}_{\overline{s}-1,i} \right\},\ \widetilde{p}_{\overline{s}-1,i}/2 \right\} \\ &\geq \min\left\{ \widetilde{\tau}_{\overline{s},i},\ \widetilde{p}_{\overline{s}-1,i} \right\} = \min\left\{ \widetilde{\tau}_{\overline{s},i},\ \widetilde{p}_{\overline{s}-2,i} \right\} = \min\left\{ \widetilde{\tau}_{\overline{s},i},\ \widetilde{p}_{\overline{s}-3,i} \right\} \ldots = \min\left\{ \widetilde{\tau}_{\overline{s},i},\ \widetilde{p}_{0,i} \right\} = \widetilde{\tau}_{\overline{s},i},\end{aligned}$$

and therefore $\overline{p}_{s,i,j} \geq \widetilde{\tau}_{\overline{s},i}$. Now let $\{\overline{s}, l'\}$ be the node where $\widetilde{\tau}_{\overline{s},i}$ was computed. We will again drop the $\{h,l\}$ index from $\mathcal{D}_{\{h,l\}}$, and simply refer to it as $\mathcal{D}$. Similarly, we will refer with $\mathcal{D}_i$ to $\mathcal{D}_{\{\overline{s},l'\}}$ (as in, the dataset used to compute $\widetilde{\tau}_{\overline{s},i}$), and with $\overline{\mathcal{D}}_i$ to the samples in $\mathcal{D}$ not contained in $\mathcal{D}_i$ (complement of $\mathcal{D}_i$). Define $\mathbf{A}$ as the $|\mathcal{D}| \times |\mathcal{D}_i|$ matrix that contains the columns of $\mathbf{S}_{\overline{s}}$ related to points in $\mathcal{D}_i$, and similarly define $\mathbf{B}$ as the $|\mathcal{D}| \times |\overline{\mathcal{D}}_i|$ matrix that contains the columns of $\mathbf{S}_{\overline{s}}$ related to points in $\overline{\mathcal{D}}_i$, where $\mathbf{S}_{\overline{s}}$ can be reconstructe by interleaving columns of $\mathbf{A}$ and $\mathbf{B}$. From its definition in Eq.5, we know that $\widetilde{\tau}_{s,i}$ is computed by Alg. 2 as

$$\widetilde{\tau}_{s,i} = (1-\varepsilon)\phi_i^{\mathsf{T}} \left(\mathbf{\Phi}_{\mathcal{D}}\mathbf{A}\mathbf{A}^{\mathsf{T}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} + (1+\varepsilon)\gamma\mathbf{I}_D\right)^{-1}\phi_i,$$

using only the points in $\mathbf{A}$ that are available at node $\{\overline{s}, l'\}$. From Lem.6 we know that

$$\begin{aligned}\|\mathbf{Y}_{\overline{s}}\| = \left\|\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_{\overline{s}}^{\{h,l\}}\right\|_2 &= \left\|(\mathbf{\Phi}_{\mathcal{D}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} + \gamma\mathbf{I}_D)^{-1/2}(\mathbf{\Phi}_{\mathcal{D}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} - \mathbf{\Phi}_{\mathcal{D}}\mathbf{S}_{\overline{s}}\mathbf{S}_{\overline{s}}^{\mathsf{T}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}})(\mathbf{\Phi}_{\mathcal{D}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} + \gamma\mathbf{I}_D)^{-1/2}\right\|_2 \\ &= \left\|(\mathbf{\Phi}_{\mathcal{D}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} + \gamma\mathbf{I}_D)^{-1/2}(\mathbf{\Phi}_{\mathcal{D}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} - \mathbf{\Phi}_{\mathcal{D}}\mathbf{A}\mathbf{A}^{\mathsf{T}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} - \mathbf{\Phi}_{\mathcal{D}}\mathbf{B}\mathbf{B}^{\mathsf{T}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}})(\mathbf{\Phi}_{\mathcal{D}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} + \gamma\mathbf{I}_D)^{-1/2}\right\|_2 \leq \varepsilon\end{aligned}$$

and we know that this implies

$$\mathbf{\Phi}_{\mathcal{D}}\mathbf{A}\mathbf{A}^{\mathsf{T}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} \preceq \mathbf{\Phi}_{\mathcal{D}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} + \varepsilon(\mathbf{\Phi}_{\mathcal{D}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} + \gamma\mathbf{I}_D) - \mathbf{\Phi}_{\mathcal{D}}\mathbf{B}\mathbf{B}^{\mathsf{T}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} \preceq \mathbf{\Phi}_{\mathcal{D}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} + \varepsilon(\mathbf{\Phi}_{\mathcal{D}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} + \gamma\mathbf{I}_D).$$

Plugging it in the initial definition,

$$\begin{aligned}\widetilde{\tau}_{s,i} &= (1-\varepsilon)\phi_i^{\mathsf{T}} \left(\mathbf{\Phi}_{\mathcal{D}}\mathbf{A}\mathbf{A}^{\mathsf{T}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} + (1+\varepsilon)\gamma\mathbf{I}_D\right)^{-1}\phi_i \\ &\geq (1-\varepsilon)\phi_i^{\mathsf{T}}(\mathbf{\Phi}_{\mathcal{D}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} + \varepsilon(\mathbf{\Phi}_{\mathcal{D}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} + \gamma\mathbf{I}_D) + (1+\varepsilon)\gamma\mathbf{I}_D)^{-1}\phi_i \\ &= (1-\varepsilon)\frac{1}{1+2\varepsilon}\phi_i^{\mathsf{T}}(\mathbf{\Phi}_{\mathcal{D}}\mathbf{\Phi}_{\mathcal{D}}^{\mathsf{T}} + \gamma\mathbf{I}_D)^{-1}\phi_i \geq \frac{1-\varepsilon}{1+2\varepsilon}\tau_{h,i} \geq \tau_{h,i}/\alpha.\end{aligned}$$

□

This result is weaker then Lemmas 2 and 4, since we do not provide an upper bound, and only show that $\widetilde{p}_{s,i} \geq p_{h,i}/\alpha$ and not $\widetilde{p}_{s,i} \geq p_{s,i}/\alpha$, but it guarantees that the probabilities used at any intermediate layer s are bigger than a fraction $1/\alpha$ of the exact probabilities that we would use at layer h, which will suffice for our purpose.

We now proceed by studying the process $\{\overline{\mathbf{Y}}_s\}_{s=1}^h$ and showing that it is a bounded martingale. In order to show that $\overline{\mathbf{Y}}_s$ is a martingale, it is sufficient to verify the following (equivalent) conditions

$$\mathbb{E}\left[\overline{\mathbf{Y}}_s \;\middle|\; \mathcal{F}_{s-1}\right] = \overline{\mathbf{Y}}_{s-1} \;\;\Leftrightarrow\;\; \mathbb{E}\left[\overline{\mathbf{X}}_{\{s,i,j\}} \;\middle|\; \mathcal{F}_{s-1}\right] = \mathbf{0}.$$

We begin by inspecting the conditional random variable $\overline{\mathbf{X}}_{\{s,i,j\}}|\mathcal{F}_{s-1}$. Given the definition of $\overline{\mathbf{X}}_{\{s,i,j\}}$, the conditioning on $\mathcal{F}_{s-1}$ determines the values of $\overline{z}_{s-1,i,j}$ and the approximate probabilities $\overline{p}_{s-1,i,j}$ and $\overline{p}_{s,i,j}$. In fact, remember that these quantities are fully determined by the realizations in $\mathcal{F}_{s-1}$ which are contained in $\mathcal{F}_{s-1}$. As a result, the only stochastic quantity in $\overline{\mathbf{X}}_{\{s,i,j\}}$ is the variable $\overline{z}_{s,i,j}$. Specifically, if $\|\overline{\mathbf{Y}}_{s-1}\| \geq \varepsilon$, then we have $\overline{p}_{s,i,j} = \overline{p}_{s-1,i,j}$ and $\overline{z}_{s,i,j} = \overline{z}_{s-1,i,j}$ (the process is stopped), and the martingale requirement $\mathbb{E}\left[\overline{\mathbf{X}}_{\{s,i,j\}} \;\middle|\; \mathcal{F}_{s-1}\right] = \mathbf{0}$ is trivially satisfied. On the other hand, if $\|\overline{\mathbf{Y}}_{s-1}\| \leq \varepsilon$ we have

$$\begin{aligned}&\mathop{\mathbb{E}}_{u_{s,i,j}}\left[\frac{1}{q}\left(\frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s-1,i,j}} - \frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}}\right)\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}} \;\middle|\; \mathcal{F}_{s-1}\right] \\ &\quad= \frac{1}{q}\left(\frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s-1,i,j}} - \frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s,i,j}}\mathbb{E}\left[\mathbb{I}\left\{u_{s,i,j} \leq \frac{\overline{p}_{s,i,j}}{\overline{p}_{s-1,i,j}}\right\} \;\middle|\; \mathcal{F}_{s-1}\right]\right)\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}} \\ &\quad= \frac{1}{q}\left(\frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s-1,i,j}} - \frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s,i,j}}\frac{\overline{p}_{s,i,j}}{\overline{p}_{s-1,i,j}}\right)\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}} = \mathbf{0},\end{aligned}$$

where we use the recursive definition of $\overline{z}_{s,i,j}$ and the fact that $u_{s,i,j}$ is a uniform random variable in $[0,1]$. This proves that $\overline{\mathbf{Y}}_s$ is indeed a martingale. We now compute an upper-bound R on the norm of the values of the difference process as

$$\|\overline{\mathbf{X}}_{\{s,i,j\}}\| = \frac{1}{q}\left|\left(\frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s-1,i,j}} - \frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}}\right)\right|\|\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\| \leq \frac{1}{q}\frac{1}{\overline{p}_{s,i,j}}\|\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\| = \frac{1}{q}\frac{1}{\overline{p}_{s,i,j}}\tau_{h,i} \leq \frac{1}{q}\frac{\alpha}{\tau_{h,i}}\tau_{h,i} = \frac{\alpha}{q} \overset{\text{def}}{=} R,$$

where we used Lemma 7 to bound $\overline{p}_{s,i,j} \leq \tau_{h,i}/\alpha$. If instead, $\|\overline{\mathbf{Y}}_{s-1}\| \geq \varepsilon$, the process is stopped and $\|\overline{\mathbf{X}}_s\| = \|\mathbf{0}\| = 0 \leq R$.

We are now ready to use a Freedman matrix inequality from [16] to bound the norm of $\overline{\mathbf{Y}}$.

Proposition 3 (Tropp [16], Theorem 1.2)**.** *Consider a matrix martingale $\{\mathbf{Y}_k : k = 0, 1, 2, \dots\}$ whose values are self-adjoint matrices with dimension d, and let $\{\mathbf{X}_k : k = 1, 2, 3, \dots\}$ be the difference sequence. Assume that the difference sequence is uniformly bounded in the sense that*

$$\|\mathbf{X}_k\|_2 \leq R \quad \textit{almost surely} \quad \textit{for } k = 1, 2, 3, \dots.$$

Define the predictable quadratic variation process of the martingale as

$$\mathbf{W}_k \overset{\text{def}}{=} \sum_{j=1}^{k} \mathbb{E}\left[\mathbf{X}_j^2 \;\middle|\; \{\mathbf{X}_s\}_{s=0}^{j-1}\right], \quad \textit{for } k = 1, 2, 3, \dots.$$

Then, for all $\varepsilon \geq 0$ and $\sigma^2 > 0$,

$$\mathbb{P}\left(\exists k \geq 0 : \|\mathbf{Y}_k\|_2 \geq \varepsilon \ \cap \ \|\mathbf{W}_k\| \leq \sigma^2\right) \leq 2d \cdot \exp\left\{-\frac{\varepsilon^2/2}{\sigma^2 + R\varepsilon/3}\right\}.$$

In order to use the previous inequality, we develop the probability of error for any fixed h as

$$\begin{aligned}\mathbb{P}\left(\|\mathbf{Y}_h\| \geq \varepsilon\right) \leq \mathbb{P}\left(\|\overline{\mathbf{Y}}_h\| \geq \varepsilon\right) &= \mathbb{P}\left(\|\overline{\mathbf{Y}}_h\| \geq \varepsilon \cap \|\mathbf{W}_h\| \leq \sigma^2\right) + \mathbb{P}\left(\|\overline{\mathbf{Y}}_h\| \geq \varepsilon \cap \|\mathbf{W}_h\| \geq \sigma^2\right)\\ &\leq \underbrace{\mathbb{P}\left(\|\overline{\mathbf{Y}}_h\| \geq \varepsilon \cap \|\mathbf{W}_h\| \leq \sigma^2\right)}_{\text{(a)}} + \underbrace{\mathbb{P}\left(\|\mathbf{W}_h\| \geq \sigma^2\right)}_{\text{(b)}}.\end{aligned}$$

Using the bound on $\|\overline{\mathbf{X}}_{\{s,i,j\}}\|_2$, we can directly apply Proposition 3 to bound (a) for any fixed σ^2. To bound the part (b), we use the following lemma, proved later in Sec. D.3.

Lemma 8 (Low probability of the large norm of the predictable quadratic variation process)**.**

$$\mathbb{P}\left(\|\mathbf{W}_h\| \geq \frac{6\alpha}{\overline{q}}\right) \leq n \cdot \exp\left\{-2\frac{\overline{q}}{\alpha}\right\}$$

Combining Prop. 3 with $\sigma^2 = 6\alpha/\overline{q}$, Lem 8, the fact that $2\varepsilon/3 \leq 1$ and the value used by Alg. 2 $\overline{q} = 39\alpha\log(2n/\delta)/\varepsilon^2$ we obtain

$$\begin{aligned}\mathbb{P}\left(\|\mathbf{P}_{\{h,l\}} - \widetilde{\mathbf{P}}_{\{h,l\}}\|_2 \geq \varepsilon\right) &= \mathbb{P}\left(\|\mathbf{Y}_h\| \geq \varepsilon\right) \leq \mathbb{P}\left(\|\overline{\mathbf{Y}}_h\| \geq \varepsilon \cap \|\mathbf{W}_h\| \leq \sigma^2\right) + \mathbb{P}\left(\|\mathbf{W}_h\| \geq \sigma^2\right)\\ &\leq 2\nu \cdot \exp\left\{-\frac{\varepsilon^2\overline{q}}{\alpha}\left(\frac{1}{12 + 2\varepsilon/3}\right)\right\} + n \cdot \exp\left\{-2\frac{\overline{q}}{\alpha}\right\}\\ &\leq 3n \cdot \exp\left\{-\frac{\varepsilon^2}{13\alpha}\overline{q}\right\} = 3n \cdot \exp\left\{-3\log\left(\frac{2n}{\delta}\right)\right\}\\ &= 3n \cdot \exp\left\{-\log\left(\left(\frac{2n}{\delta}\right)^3\right)\right\} = 3n\frac{\delta^3}{8n^3} \leq \frac{\delta}{2n^2}.\end{aligned}$$

This, combined with the fact that $k \leq n$ since at most we can split our dataset in n parts, concludes this part of the proof.

D.3 Proof of Lemma 8 (bound on predictable quadratic variation)

Step 1 (a preliminary bound). We start by writing out $\mathbf{W}_r$ for the process $\overline{\mathbf{Y}}_s$,

$$\mathbf{W}_r = \frac{1}{\overline{q}^2} \sum_{\{s,i,j\} \leq r} \mathbb{E}\left[\left(\frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s-1,i,j}} - \frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}}\right)^2 \;\middle|\; \mathcal{F}_{\{s,i,j\}-1}\right] \boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}.$$

We rewrite the expectation terms in the equation above as

$$
\begin{aligned}
&\mathbb{E}\left[\left(\frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s-1,i,j}} - \frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}}\right)^2 \,\middle|\, \mathcal{F}_{\{s,i,j\}-1}\right] \\
&= \mathbb{E}\left[\frac{\overline{z}^2_{s-1,i,j}}{\overline{p}^2_{s-1,i,j}} - 2\frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s-1,i,j}}\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}} + \frac{\overline{z}^2_{s,i,j}}{\overline{p}^2_{s,i,j}} \,\middle|\, \mathcal{F}_{\{s,i,j\}-1}\right] \\
&\overset{(a)}{=} \mathbb{E}\left[\frac{\overline{z}^2_{s-1,i,j}}{\overline{p}^2_{s-1,i,j}} - 2\frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s-1,i,j}}\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}} + \frac{\overline{z}^2_{s,i,j}}{\overline{p}^2_{s,i,j}} \,\middle|\, \mathcal{F}_{s-1}\right] \\
&= \frac{\overline{z}^2_{s-1,i,j}}{\overline{p}^2_{s-1,i,j}} - 2\frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s-1,i,j}}\frac{1}{\overline{p}_{s,i,j}}\mathbb{E}\left[\overline{z}_{s,i,j} \mid \mathcal{F}_{s-1}\right] + \frac{1}{\overline{p}^2_{s,i,j}}\mathbb{E}\left[\overline{z}^2_{s,i,j} \mid \mathcal{F}_{s-1}\right] \\
&\overset{(b)}{=} \frac{\overline{z}_{s-1,i,j}}{\overline{p}^2_{s-1,i,j}} - 2\frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s-1,i,j}}\frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s-1,i,j}} + \frac{1}{\overline{p}^2_{s,i,j}}\mathbb{E}\left[\overline{z}_{s,i,j} \mid \mathcal{F}_{s-1}\right] \\
&= \frac{1}{\overline{p}^2_{s,i,j}}\mathbb{E}\left[\overline{z}_{s,i,j} \mid \mathcal{F}_{s-1}\right] - \frac{\overline{z}_{s-1,i,j}}{\overline{p}^2_{s-1,i,j}} \\
&\overset{(c)}{=} \frac{1}{\overline{p}_{s,i,j}}\frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s-1,i,j}} - \frac{\overline{z}_{s-1,i,j}}{\overline{p}^2_{s-1,i,j}} = \frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s-1,i,j}}\left(\frac{1}{\overline{p}_{s,i,j}} - \frac{1}{\overline{p}_{s-1,i,j}}\right),
\end{aligned}
$$

where in (a) we use the fact that the approximate probabilities $\overline{p}_{s-1,i,j}$ and $\overline{p}_{s,i,j}$ and $\overline{z}_{s-1,i,j}$ are fixed at the end of the previous iteration, while in (b) and (c) we use the fact that $\overline{z}_{s,i,j}$ is a Bernoulli of parameter $\overline{p}_{s,i,j}/\overline{p}_{s-1,i,j}$ (whenever $\overline{z}_{s-1,i,j}$ is equal to 1). Therefore, we can write $\mathbf{W}_r$ at the end of the process as

$$
\mathbf{W}_h = \mathbf{W}_{\{h,m,\overline{q}\}} = \frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\sum_{s=1}^{h}\frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s-1,i,j}}\left(\frac{1}{\overline{p}_{s,i,j}} - \frac{1}{\overline{p}_{s-1,i,j}}\right)\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T}.
$$

We can now upper-bound $\mathbf{W}_h$ as

$$
\begin{aligned}
\mathbf{W}_h &\preceq \frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\sum_{s=1}^{h}\frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s-1,i,j}}\left(\frac{1}{\overline{p}_{s,i,j}} - \frac{1}{\overline{p}_{s-1,i,j}}\right)\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T} \\
&= \frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\left(\frac{\overline{z}_{h,i,j}}{\overline{p}^2_{h,i,j}} - \frac{\overline{z}_{h,i,j}}{\overline{p}^2_{h,i,j}} + \sum_{s=1}^{h}\frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s-1,i,j}}\left(\frac{1}{\overline{p}_{s,i,j}} - \frac{1}{\overline{p}_{s-1,i,j}}\right)\right)\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T} \\
&= \frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\left(\frac{\overline{z}_{h,i,j}}{\overline{p}^2_{h,i,j}} + \left(\sum_{s=1}^{h} -\frac{\overline{z}_{s,i,j}}{\overline{p}^2_{s,i,j}} + \frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s,i,j}\overline{p}_{s-1,i,j}}\right) - \frac{\overline{z}_{0,i,j}}{\overline{p}^2_{0,i,j}}\right)\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T} \\
&\preceq \frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\left(\frac{\overline{z}_{h,i,j}}{\overline{p}^2_{h,i,j}} + \left(\sum_{s=1}^{h}\frac{\overline{z}_{s-1,i,j}}{\overline{p}_{s,i,j}\overline{p}_{s-1,i,j}} - \frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}\overline{p}_{s-1,i,j}}\right)\right)\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T} \\
&= \frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\left(\frac{\overline{z}_{h,i,j}}{\overline{p}^2_{h,i,j}} + \sum_{s=1}^{h}\frac{\overline{z}_{s-1,i,j}(1-\overline{z}_{s,i,j})}{\overline{p}_{s,i,j}\overline{p}_{s-1,i,j}}\right)\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T},
\end{aligned}
$$

where in the inequality we use the fact $\overline{p}_{s,i,j} \leq \overline{p}_{s-1,i,j}$. From the definition of $\overline{p}_{s,i,j}$, we know that when $\overline{z}_{s,i,j} = 0$, $\overline{p}_{s,i,j} = \overline{p}_{s-1,i,j}$. Therefore $\frac{\overline{z}_{s-1,i,j}(1-\overline{z}_{s,i,j})}{\overline{p}_{s,i,j}\overline{p}_{s-1,i,j}} = \frac{\overline{z}_{s-1,i,j}(1-\overline{z}_{s,i,j})}{\overline{p}^2_{s-1,i,j}}$, since the term is non-zero only when $\overline{z}_{s,i,j} = 0$.

Finally, we see that only one of the $\overline{z}_{s-1,i,j}(1-\overline{z}_{s,i,j})$ terms can be active for $s \in [h]$ and thus

$$\begin{aligned}\mathbf{W}_h &\preceq \frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\left(\frac{\overline{z}_{h,i,j}}{\overline{p}_{h,i,j}^2} + \sum_{s=1}^{h}\frac{\overline{z}_{s-1,i,j}(1-\overline{z}_{s,i,j})}{\overline{p}_{s-1,i,j}^2}\right)\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\\ &= \frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\left(\max\left\{\max_{s=1,\dots,h}\left\{\frac{\overline{z}_{s-1,i,j}(1-\overline{z}_{s,i,j})}{\overline{p}_{s-1,i,j}^2}\right\};\frac{\overline{z}_{h,i,j}}{\overline{p}_{h,i,j}^2}\right\}\right)\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\\ &= \frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\left(\max_{s=0,\dots,h}\left\{\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}^2}\right\}\right).\end{aligned} \tag{13}$$

Step 2 (introduction of a stochastically dominant process). We want to study $\max_{s=0,\dots,h}\left\{\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}^2}\right\}$. To simplify notation, we will consider $\max_{s=0,\dots,h}\left\{\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}}\right\}$, where we removed the square, which will be re-added in the end. We know trivially that this quantity is larger or equal than, 1 because $\overline{z}_{0,i,j}/\overline{p}_{0,i,j} = 1$, but upper-bounding this quantity is not trivial as the evolution of the various $\overline{p}_{s,i,j}$ depends in a complex way on the interaction between the random variables $\overline{z}_{s,i,j}$. Nonetheless, whenever $\overline{p}_{s,i,j}$ is significantly smaller than $\overline{p}_{s-1,i,j}$, the probability of keeping a copy of point i at iteration s (i.e., $\overline{z}_{s,i,j} = 1$) is also very small. As a result, we expect the ratio $\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}}$ to be still small with high probability.

Unfortunately, due to the dependence between different copies of the point at different iterations, it seems difficult to exploit this intuition directly to provide an overall high-probability bound on $\mathbf{W}_h$. For this reason, we simplify the analysis by replacing each of the (potentially dependent) chains $\{\overline{z}_{s,i,j}/\overline{p}_{s,i,j}\}_{s=0}^{h}$ with a set of (independent) random variables $w_{0,i,j}$ that will stochastically dominate them.

We define the random variable $w_{s,i,j}$ using the following conditional distribution,[4]

$$\mathbb{P}\left(\frac{1}{w_{s,i,j}} \leq a \;\middle|\; \mathcal{F}_s\right) = \begin{cases} 0 & \text{for} \quad a < 1/\overline{p}_{s,i,j} \\ 1 - \frac{1}{\overline{p}_{s,i,j}a} & \text{for} \quad 1/\overline{p}_{s,i,j} \leq a < \alpha/p_{h,i} \\ 1 & \text{for} \quad \alpha/p_{h,i} \leq a \end{cases}.$$

To show that this distribution is well defined, we use Lem. 7 to guarantee that $1/\overline{p}_{s,i,j} \leq a < \alpha/p_{h,i}$. Note that the distribution of $\frac{1}{w_{s,i,j}}$ conditioned on $\mathcal{F}_s$ is determined by only $\overline{p}_{s,i,j}$, $p_{h,i}$, and α, where $p_{h,i}$ and α are fixed. Remembering that $\overline{p}_{s,i,j}$ is a function of $\mathcal{F}_{s-1}$ (computed using the previous iteration), we have that

$$\mathbb{P}\left(\frac{1}{w_{s,i,j}} \leq a \;\middle|\; \mathcal{F}_s\right) = \mathbb{P}\left(\frac{1}{w_{s,i,j}} \leq a \;\middle|\; \mathcal{F}_{s-1}\right).$$

Notice that in the definition of $w_{s,i,j}$, none of the other $w_{s',i',j'}$ (for any different s', i', or j') appears and $\overline{p}_{s,i,j}$ is a function of $\mathcal{F}_{s-1}$. It follows that given $\mathcal{F}_{s-1}$, $w_{s,i,j}$ is independent from all other $w_{s',i',j'}$ (for any different s', i', or j'). This is easier to see in the probabilistic graphical model reported in Fig. 3, which illustrates the dependence between the various variables.

Finally for the special case $w_{0,i,j}$ the definition above reduces to

$$\mathbb{P}\left(\frac{1}{w_{0,i,j}} \leq a\right) = \begin{cases} 0 & \text{for} \quad a < 1 \\ 1 - \frac{1}{a} & \text{for} \quad 1 \leq a < \alpha/p_{h,i} \\ 1 & \text{for} \quad \alpha/p_{h,i} \leq a \end{cases}, \tag{14}$$

since $\overline{p}_{0,i,j} = 1$ by definition. From this definition, $w_{0,i,j}$ and $w_{0,i',j'}$ are all independent, and this will allow us to use stronger concentration inequalities for independent random variables.

[4] Notice that unlike $\overline{z}_{s,i,j}$, $w_{s,i,j}$ is no longer $\mathcal{F}_s$-measurable but it is $\mathcal{F}'_s$-measurable, where

$$\mathcal{F}'_{\{s,i,j\}} = \left\{u_{s',i',j'},\ \forall\{s',i',j'\} \leq \{s,i,j\}\right\} \cup \{w_{s,i,j}\} = \mathcal{F}_{\{s,i,j\}} \cup \{w_{s,i,j}\}.$$

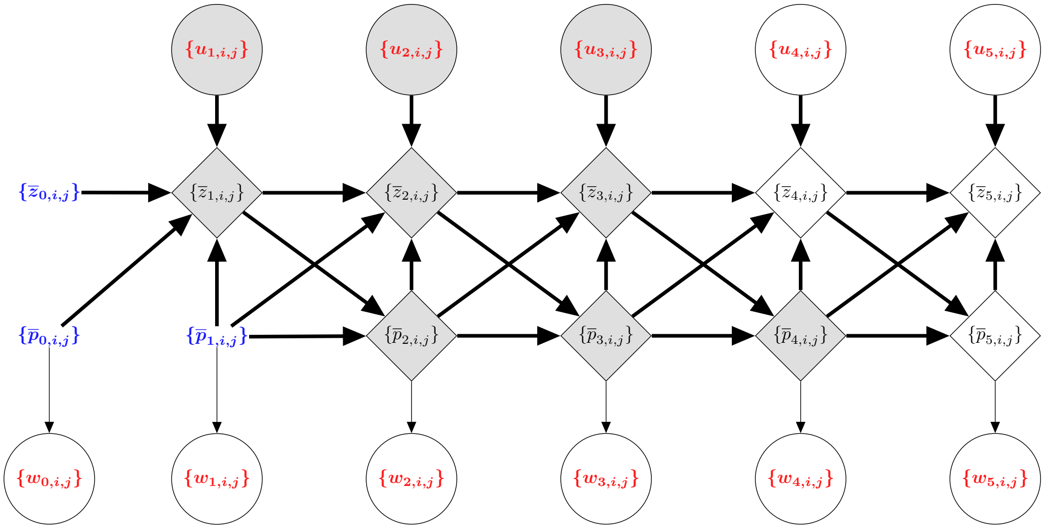

$\{u_{1,i,j}\}$
$\{u_{2,i,j}\}$
$\{u_{3,i,j}\}$
$\{u_{4,i,j}\}$
$\{u_{5,i,j}\}$
$\{\overline{z}_{0,i,j}\}$
$\{\overline{z}_{1,i,j}\}$
$\{\overline{z}_{2,i,j}\}$
$\{\overline{z}_{3,i,j}\}$
$\{\overline{z}_{4,i,j}\}$
$\{\overline{z}_{5,i,j}\}$
$\{\overline{p}_{0,i,j}\}$
$\{\overline{p}_{1,i,j}\}$
$\{\overline{p}_{2,i,j}\}$
$\{\overline{p}_{3,i,j}\}$
$\{\overline{p}_{4,i,j}\}$
$\{\overline{p}_{5,i,j}\}$
$\{w_{0,i,j}\}$
$\{w_{1,i,j}\}$
$\{w_{2,i,j}\}$
$\{w_{3,i,j}\}$
$\{w_{4,i,j}\}$
$\{w_{5,i,j}\}$

Figure 3: The dependence graph of the considered variables. **Red** variables are **random**. Black variables are deterministically computed using their input (a function of their input), with bold lines indicating the deterministic (functional) relation. **Blue** variables are **constants**. A grey filling indicates that a random variable is observed or a function of observed variables.

Step 3 Proving the dominance. We remind the reader that a random variable A stochastically dominates random variable B, if for all values a the two equivalent conditions are verified,

$$\mathbb{P}(A \geq a) \geq \mathbb{P}(B \geq a) \Leftrightarrow \mathbb{P}(A \leq a) \leq \mathbb{P}(B \leq a).$$

As a consequence, if A dominates B, the following implication holds,

$$\mathbb{P}(A \geq a) \geq \mathbb{P}(B \geq a) \implies \mathbb{E}[A] \geq \mathbb{E}[B],$$

while the reverse (A dominates B, if $\mathbb{E}[A] \geq \mathbb{E}[B]$) is not true in general. Following this definition of stochastic dominance, our goal is to prove

$$\mathbb{P}\left(\max_{s=0}^{h} \frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}} \leq a\right) \geq \mathbb{P}\left(\frac{1}{w_{0,i,j}} \leq a\right).$$

We prove this inequality by proceeding backwards with a sequence of conditional probabilities. We first study the distribution of the maximum conditional to the state of the algorithm at the end of iteration h, i.e., $\mathcal{F}_h$. From the definition of $w_{h,i,j}$, we know that, w.p. 1, $1/\overline{p}_{h,i} \leq 1/w_{h,i,j}$. Therefore,

$$\mathbb{P}\left(\max_{s=0,\ldots,h} \frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}} \leq a\right) \geq \mathbb{P}\left(\max\left\{\max_{s=0,\ldots,h-1} \frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}}; \frac{\overline{z}_{h,i,j}}{w_{h,i,j}}\right\} \leq a\right).$$

Now focus on an arbitrary intermediate step $1 \leq k \leq h$, where we fix $\mathcal{F}_{k-1}$. Since $u_{k,i,j}$ and $w_{k,i,j}$ are independent

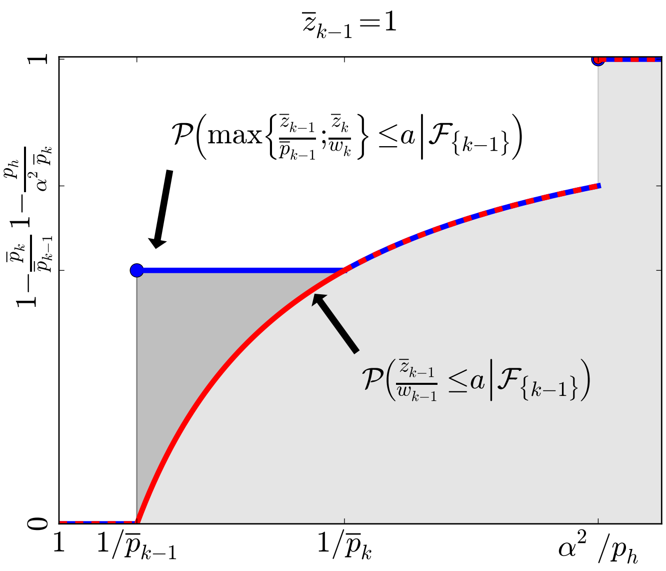


Figure 4: C.d.f. of $\max\left\{\overline{z}_{k-1,i,j}/\overline{p}_{t,i,j};\overline{z}_{k,i,j}/\overline{p}_{k,i,j}\right\}$ and $\overline{z}_{k-1,i,j}/w_{k-1,i,j}$ conditioned on $\mathcal{F}_{\{k-1\}}$. For conciseness, we omit the i,j indices.

given $\mathcal{F}_{k-1}$, we have

$$\begin{aligned}
\mathbb{P}\left(\frac{\overline{z}_{k,i,j}}{w_{k,i,j}}\le a\;\middle|\;\mathcal{F}_{k-1}\right) &= \mathbb{P}\left(\mathbb{I}\left\{u_{k,i,j}\le\frac{\overline{p}_{k,i,j}}{\overline{p}_{k-1,i,j}}\right\}\frac{1}{w_{k,i,j}}\le a\;\middle|\;\mathcal{F}_{k-1}\right)\\
&=\begin{cases}
0 & \text{for}\quad a\le 0\\
1-\frac{\overline{p}_{k,i,j}}{\overline{p}_{k-1,i,j}} & \text{for}\quad 0\le a<1/\overline{p}_{k,i,j}\\
1-\frac{\overline{p}_{k,i,j}}{\overline{p}_{k-1,i,j}}+\frac{\overline{p}_{k,i,j}}{\overline{p}_{k-1,i,j}}\left(1-\frac{1}{\overline{p}_{k,i,j}a}\right)=1-\frac{1}{\overline{p}_{k-1,i,j}a} & \text{for}\quad 1/\overline{p}_{k,i,j}\le a<\alpha/p_{h,i}\\
1 & \text{for}\quad \alpha/p_{h,i}\le a
\end{cases}\\
&\ge\begin{cases}
0 & \text{for}\quad a<1/\overline{p}_{k-1,i,j}\\
1-\frac{1}{\overline{p}_{k-1,i,j}a} & \text{for}\quad 1/\overline{p}_{k-1,i,j}\le a<1/\overline{p}_{k,i,j}\\
1-\frac{1}{\overline{p}_{k-1,i,j}a} & \text{for}\quad 1/\overline{p}_{k,i,j}\le a<\alpha/p_{h,i}\\
1 & \text{for}\quad \alpha/p_{h,i}\le a
\end{cases} \qquad (15)\\
&=\mathbb{P}\left(\frac{1}{w_{k-1,i,j}}\le a\;\middle|\;\mathcal{F}_{k-2}\right)=\mathbb{P}\left(\frac{1}{w_{k-1,i,j}}\le a\;\middle|\;\mathcal{F}_{k-1}\right),
\end{aligned}$$

where the inequality is also represented in Fig. 4. We now proceed by peeling off layers from the end of the chain one by one, taking advantage of the dominance we just proved. Fig. 4 visualizes one step of the peeling when $\overline{z}_{k-1,i,j}=1$ (note that the peeling is trivially true when $\overline{z}_{k-1,i,j}=0$ since the whole chain terminated at step

$\overline{z}_{k-1,i,j}$). We show how to move from an iteration $k \leq h$ to $k-1$.

$$\begin{aligned}
&\mathbb{P}\left(\max\left\{\max_{s=0\dots k-1}\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}};\frac{\overline{z}_{k,i,j}}{w_{k,i,j}}\right\}\leq a\right) = \mathop{\mathbb{E}}_{\mathcal{F}_{k-1}}\left[\mathbb{P}\left(\max\left\{\max_{s=0\dots k-1}\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}};\frac{\overline{z}_{k,i,j}}{w_{k,i,j}}\right\}\leq a \;\middle|\; \mathcal{F}_{k-1}\right)\right]\\
&\overset{(a)}{\geq} \mathop{\mathbb{E}}_{\mathcal{F}_{k-1}}\left[\mathbb{P}\left(\max\left\{\max_{s=0\dots k-1}\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}};\frac{\overline{z}_{k-1,i,j}}{w_{k-1,i,j}}\right\}\leq a \;\middle|\; \mathcal{F}_{k-1}\right)\right]\\
&= \mathop{\mathbb{E}}_{\mathcal{F}_{k-1}}\left[\mathbb{P}\left(\max\left\{\max_{s=0\dots k-2}\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}};\frac{\overline{z}_{k-1,i,j}}{\overline{p}_{k-1,i,j}};\frac{\overline{z}_{k-1,i,j}}{w_{k-1,i,j}}\right\}\leq a \;\middle|\; \mathcal{F}_{k-1}\right)\right]\\
&= \mathop{\mathbb{E}}_{\mathcal{F}_{k-1}}\left[\mathbb{P}\left(\max\left\{\max_{s=0\dots k-2}\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}};\overline{z}_{k-1,i,j}\max\left\{\frac{1}{\overline{p}_{k-1,i,j}};\frac{1}{w_{k-1,i,j}}\right\}\right\}\leq a \;\middle|\; \mathcal{F}_{k-1}\right)\right]\\
&\overset{(b)}{=} \mathop{\mathbb{E}}_{\mathcal{F}_{k-1}}\left[\mathbb{P}\left(\max\left\{\max_{s=0\dots k-2}\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}};\frac{\overline{z}_{k-1,i,j}}{w_{k-1,i,j}}\right\}\leq a \;\middle|\; \mathcal{F}_{k-1}\right)\right] = \mathbb{P}\left(\max\left\{\max_{s=0\dots k-2}\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}};\frac{\overline{z}_{k-1,i,j}}{w_{k-1,i,j}}\right\}\leq a\right),
\end{aligned}$$

where in (a), given $\mathcal{F}_{k-1}$, everything is fixed except $u_{k,i,j}$ and $w_{k,i,j}$ and we can use the stochastic dominance in (15), and in (b) we use the fact that the inner maximum is always attained by $1/w_{k,i,j}$ since by definition $1/w_{k-1,i,j}$ is lower-bounded by $1/\overline{p}_{k-1,i,j}$. Applying the inequality recursively from $k = h$ to $k = 1$ removes all $\overline{z}_{s,i,j}$ from the maximum and we are finally left with only $w_{0,i,j}$ as we wanted,

$$\mathbb{P}\left(\max_{s=0,\dots,h}\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}}\leq a\right) \geq \mathbb{P}\left(\max\left\{\frac{\overline{z}_{0,i,j}}{\overline{p}_{0,i,j}};\frac{\overline{z}_{0,i,j}}{w_{0,i,j}}\right\}\leq a\right) \geq \mathbb{P}\left(\frac{1}{w_{0,i,j}}\leq a\right),$$

where in the last inequality we used that $\overline{z}_{0,i,j} = 1$ from the definition of the algorithm and $\overline{p}_{0,i,j} = 1$ while $w_{0,i,j} \leq 1$ by (14).

Step 4 (stochastic dominance on $\mathbf{W}_h$). Now that we proved the stochastic dominance of $1/w_{0,i,j}$, we plug this result in the definition of $\mathbf{W}_h$. For the sake of notation, we introduce the term $\overline{p}^{\max}_{h',i,j}$ to indicate the maximum over the first h' step of copy i, j such that

$$\max_{s=0,\dots,h'}\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}} = \frac{1}{\overline{p}^{\max}_{h',i,j}}.$$

We first notice that while $\overline{\mathbf{Y}}_h$ is not necessarily PSD, $\mathbf{W}_h$ is a sum of PSD matrices. Introducing the function $\Lambda(\{1/\overline{p}^{\max}_{h,i,j}\}_{i,j})$ we can restate Eq. 13 as

$$\|\mathbf{W}_h\| = \lambda_{\max}(\mathbf{W}_h) \leq \Lambda(\{1/\overline{p}^{\max}_{h,i,j}\}_{i,j}) \stackrel{\text{def}}{=} \lambda_{\max}\left(\frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\left(\frac{1}{\overline{p}^{\max}_{h,i,j}}\right)^2 \boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\right).$$

In Step 4, we showed that $1/\overline{p}^{\max}_{h,i,j}$ is stochastically dominated by $1/w_{0,i,j}$ for every i and j. In order to bound $\Lambda(\{1/\overline{p}^{\max}_{h,i,j}\}_{i,j})$, we need to show that this dominance also applies to the summation over all columns inside the matrix norm. We can reformulate $\Lambda(\{1/\overline{p}^{\max}_{h,i,j}\}_{i,j})$ as

$$\begin{aligned}
&\lambda_{\max}\left(\frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\left(\frac{1}{\overline{p}^{\max}_{h,i,j}}\right)^2 \boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\right) = \max_{\mathbf{x}:\|\mathbf{x}\|=1}\mathbf{x}^{\mathsf{T}}\left(\frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\left(\frac{1}{\overline{p}^{\max}_{h,i,j}}\right)^2 \boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\right)\mathbf{x}\\
&= \max_{\mathbf{x}:\|\mathbf{x}\|=1}\frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\left(\frac{1}{\overline{p}^{\max}_{h,i,j}}\right)^2 \|\boldsymbol{\psi}_i\|_2^2\mathbf{x}^{\mathsf{T}}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\mathbf{x} = \max_{\mathbf{x}:\|\mathbf{x}\|=1}\frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\left(\frac{1}{\overline{p}^{\max}_{h,i,j}}\right)^2 \left(\|\boldsymbol{\psi}_i\|_2\boldsymbol{\psi}_i^{\mathsf{T}}\mathbf{x}\right)^2.
\end{aligned}$$

From this reformulation, it is easy to see that, because $1/\overline{p}^{\max}_{h,i,j}$ is strictly positive, the function $\Lambda(\{1/\overline{p}^{\max}_{h,i,j}\}_{i,j})$ is monotonically increasing w.r.t. the individual $1/\overline{p}^{\max}_{h,i,j}$, or in other words that increasing an $1/\overline{p}^{\max}_{h,i,j}$ without decreasing the others can only increase the maximum. Introducing $\Lambda(\{1/w_{0,i,j}\}_{i,j})$ as

$$\Lambda(\{1/w_{0,i,j}\}_{i,j}) \stackrel{\text{def}}{=} \max_{\mathbf{x}:\|\mathbf{x}\|=1}\frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\left(\frac{1}{w_{0,i,j}}\right)^2\left(\|\boldsymbol{\psi}_i\|_2\boldsymbol{\psi}_i^{\mathsf{T}}\mathbf{x}\right)^2,$$

we now need to prove the stochastic dominance of $\Lambda(\{1/w_{0,i,j}\}_{i,j})$ over $\Lambda(\{1/\overline{p}^{\max}_{h,i,j}\}_{i,j})$. Using the definition of $1/\overline{p}^{\max}_{h,i,j}$, $w_{h,i,j}$, and the monotonicity of Λ we have

$$\begin{aligned}\mathbb{P}\left(\Lambda\left(\left\{\frac{1}{\overline{p}^{\max}_{h,i,j}}\right\}_{i,j}\right) \leq a\right) &= \mathbb{P}\left(\Lambda\left(\left\{\max\left\{\max_{s=0,\dots,h-1}\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}}; \frac{\overline{z}_{h,i,j}}{\overline{p}_{h,i,j}}\right\}\right\}_{i,j}\right) \leq a\right)\\ &\geq \mathbb{P}\left(\Lambda\left(\left\{\max\left\{\max_{s=0,\dots,h-1}\frac{\overline{z}_{s,i,j}}{\overline{p}_{s,i,j}}; \frac{\overline{z}_{h,i,j}}{w_{h,i,j}}\right\}\right\}_{i,j}\right) \leq a\right).\end{aligned}$$

Now pick $1 \leq k \leq h$, for a fixed $\mathcal{F}_{k-1}$, $\frac{1}{\overline{p}^{\max}_{k-1,i,j}}$ is a constant and $\max\left\{\frac{1}{\overline{p}^{\max}_{k,i,j}}; x\right\}$ is a monotonically increasing function in x, making $\Lambda\left(\max\left\{\frac{1}{\overline{p}^{\max}_{k,i,j}}; x\right\}\right)$ also an increasing function. Therefore, we have

$$\begin{aligned}&\mathbb{P}\left(\Lambda\left(\left\{\max\left\{\frac{1}{\overline{p}^{\max}_{k-1,i,j}}; \frac{\overline{z}_{k,e,j}}{w_{k,i,j}}\right\}\right\}_{i,j}\right) \leq a\right) = \mathop{\mathbb{E}}_{\mathcal{F}_{k-1}}\left[\mathbb{P}\left(\Lambda\left(\left\{\max\left\{\frac{1}{\overline{p}^{\max}_{k-1,i,j}}; \frac{\overline{z}_{k,e,j}}{w_{k,i,j}}\right\}\right\}_{i,j}\right) \leq a \,\middle|\, \mathcal{F}_{k-1}\right)\right]\\ &\overset{(a)}{\geq} \mathop{\mathbb{E}}_{\mathcal{F}_{k-1}}\left[\mathbb{P}\left(\Lambda\left(\left\{\max\left\{\frac{1}{\overline{p}^{\max}_{k-1,i,j}}; \frac{\overline{z}_{k-1,i,j}}{w_{k-1,i,j}}\right\}\right\}_{i,j}\right) \leq a \,\middle|\, \mathcal{F}_{k-1}\right)\right]\\ &\overset{(b)}{=} \mathop{\mathbb{E}}_{\mathcal{F}_{k-1}}\left[\mathbb{P}\left(\Lambda\left(\left\{\max\left\{\frac{1}{\overline{p}^{\max}_{k-2,i,j}}; \frac{\overline{z}_{k-1,i,j}}{w_{k-1,i,j}}\right\}\right\}_{i,j}\right) \leq a \,\middle|\, \mathcal{F}_{k-1}\right)\right],\end{aligned}$$

where inequality (a) follows from the fact that stochastic dominance is preserved by monotonically increasing functions [9], such as Λ, combined with the fact that for a fixed $\mathcal{F}_{k-1}$ the variables $\overline{z}_{k,i,j}$ and $w_{k,i,j}$ are all independent and (b) from the definition of $1/\overline{p}^{\max}_{k-1,i,j}$ and the fact that by definition $1/w_{k-1,i,j}$ is lower-bounded by $1/\overline{p}_{k-1,i,j}$. We can iterate this inequality to obtain the desired result

$$\mathbb{P}(\|\mathbf{W}_h\| \geq \sigma^2) \leq \mathbb{P}\left(\Lambda\left(\left\{\frac{1}{\overline{p}^{\max}_{h,i,j}}\right\}_{i,j}\right) \geq \sigma^2\right) \leq \mathbb{P}\left(\lambda_{\max}\left(\frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\left(\frac{1}{w_{0,i,j}}\right)^2 \boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\right) \geq \sigma^2\right).$$

Step 5 (concentration inequality). Since all $w_{0,i,j}$ are (unconditionally) independent from each other, we can apply the following theorem.

Proposition 4 (Tropp [17], Theorem 5.1.1). *Consider a finite sequence $\{\mathbf{X}_k : k = 1, 2, 3, \dots\}$ whose values are independent, random, PSD Hermitian matrices with dimension d. Assume that each term in the sequence is uniformly bounded in the sense that*

$$\lambda_{\max}(\mathbf{X}_k) \leq L \quad \textit{almost surely} \quad \textit{for } k = 1, 2, 3, \dots.$$

Introduce the random matrix $\mathbf{V} \overset{\text{def}}{=} \sum_k \mathbf{X}_k$, and the maximum eigenvalue of its expectation

$$\mu_{\max} \overset{\text{def}}{=} \lambda_{\max}(\mathbb{E}[\mathbf{V}]) = \lambda_{\max}\left(\sum_k \mathbb{E}[\mathbf{X}_k]\right).$$

Then, for all $h \geq 0$,

$$\begin{aligned}\mathbb{P}(\lambda_{\max}(\mathbf{V}) \geq (1+h)\mu_{\max}) &\leq d \cdot \left[\frac{e^h}{(1+h)^{1+h}}\right]^{\frac{\mu_{\max}}{L}}\\ &\leq d \cdot \exp\left\{-\frac{\mu_{\max}}{L}((h+1)\log(h+1) - h)\right\}.\end{aligned}$$

In our case, we have

$$\mathbf{X}_{\{i,j\}} = \frac{1}{\overline{q}^2}\frac{1}{w_{0,i,j}}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}} \preceq \frac{1}{\overline{q}^2}\frac{\alpha^2}{p_{h,i}^2}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}} \preceq \frac{1}{\overline{q}^2}\frac{\alpha^2}{p_{h,i}^2}\|\boldsymbol{\psi}_i\boldsymbol{\psi}_i^{\mathsf{T}}\|^2\mathbf{I} \preceq \frac{\alpha^2}{\overline{q}^2}\mathbf{I},$$

where the first inequality follows from the definition of $w_{0,i,j}$ in Eq. 14, the second from the PSD ordering, and the third from the definition of $\|\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T}\|$.

Therefore, we can use $L \overset{\text{def}}{=} \alpha^2/\overline{q}^2$ for the purpose of Prop. 4. We need now to compute $\mathbb{E}\left[\mathbf{X}_k\right]$, that we can use in turn to compute $\mu_{\max}$. We begin by computing the expected value of $1/w_{0,i,j}$. Let us denote the c.d.f. of $1/w_{0,i,j}^2$ as

$$F_{1/w_{0,i,j}^2}(a) = \mathbb{P}\left(\frac{1}{w_{0,i,j}^2} \leq a\right) = \mathbb{P}\left(\frac{1}{w_{0,i,j}} \leq \sqrt{a}\right) = \begin{cases} 0 & \text{for} \quad a < 1 \\ 1 - \frac{1}{\sqrt{a}} & \text{for} \quad 1 \leq a < \alpha^2/p_{h,i}^2 \\ 1 & \text{for} \quad \alpha^2/p_{h,i}^2 \leq a \end{cases}.$$

Since $\mathbb{P}\left(1/w_{0,i,j}^2 \geq 0\right) = 1$, we have that

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{w_{0,i,j}}\right] &= \int_{a=0}^{\infty} \left[1 - F_{1/w_{0,i,j}}(a)\right] \mathrm{d}a \\
&= \int_{a=0}^{1} \left(1 - F_{1/w_{0,i,j}}(a)\right) \mathrm{d}a + \int_{a=1}^{\alpha^2/p_{h,i}^2} \left(1 - F_{1/w_{0,i,j}}(a)\right) \mathrm{d}a + \int_{a=\alpha^2/p_{h,i}^2}^{\infty} \left(1 - F_{1/w_{0,i,j}}(a)\right) \mathrm{d}a \\
&= \int_{a=0}^{1} (1-0)\,\mathrm{d}a + \int_{a=1}^{\alpha^2/p_{h,i}^2} \left(1 - \left(1 - \frac{1}{\sqrt{a}}\right)\right) \mathrm{d}a + \int_{a=\alpha^2/p_{h,i}^2}^{\infty} (1-1)\,\mathrm{d}a \\
&= \int_{a=0}^{1} \mathrm{d}a + \int_{a=1}^{\alpha^2/p_{h,i}^2} \frac{1}{\sqrt{a}}\,\mathrm{d}a = 1 + [2\sqrt{a}]_1^{\alpha^2/p_{h,i}^2} = 2\alpha/p_{h,i} - 1.
\end{aligned}$$

Therefore,

$$\begin{aligned}
\mu_{\max} = \lambda_{\max}(\mathbb{E}\left[\mathbf{V}\right]) = \lambda_{\max}\Big(\sum_{\{i,j\}} \mathbb{E}\left[\mathbf{X}_{\{i,j\}}\right]\Big) &= \lambda_{\max}\left(\frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\mathbb{E}\left[\frac{1}{w_{0,i,j}^2}\right]\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T}\right) \\
= \lambda_{\max}\left(\frac{1}{\overline{q}}\sum_{i=1}^{\nu}\left(\frac{2\alpha}{p_{h,i}} - 1\right)p_{h,i}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T}\right) &\leq \lambda_{\max}\left(\frac{2\alpha}{\overline{q}}\sum_{i=1}^{\nu}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T}\right) = \frac{2\alpha}{\overline{q}}\lambda_{\max}\left(\mathbf{P}\right) \leq \frac{2\alpha}{\overline{q}} \overset{\text{def}}{=} L.
\end{aligned}$$

Therefore, selecting $h = 2$, $\sigma^2 = 6\alpha/\overline{q}$ and applying Prop. 4 we have

$$\begin{aligned}
\mathbb{P}\left(\|\mathbf{W}_h\| \geq \sigma^2\right) &\leq \mathbb{P}\left(\lambda_{\max}\left(\frac{1}{\overline{q}^2}\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\frac{1}{w_{0,i,j}^2}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T}\boldsymbol{\psi}_i\boldsymbol{\psi}_i^\mathsf{T}\right) \geq (1+2)\frac{2\alpha}{\overline{q}}\right) \\
&\leq \nu \cdot \exp\left\{-\frac{2\alpha}{\overline{q}}\frac{\overline{q}^2}{\alpha^2}(3\log(3) - 2)\right\} \leq n \cdot \exp\left\{-\frac{2\overline{q}}{\alpha}\right\}.
\end{aligned}$$

D.4 Space complexity bound

Denote with A the event $A = \left\{\forall h' \in \{1, \ldots, h\} : \|\mathbf{P}^{h'} - \widetilde{\mathbf{P}}^{h'}\|_2 \leq \varepsilon\right\}$, and again $\nu = |\mathcal{D}_{\{h,l\}}|$. Letting $q = |\mathcal{I}_{\{h,l\}}| = \sum_{i=1}^{\nu} q_{h,i} = \sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} z_{h,i,j}$ be the random number of points in $\mathcal{I}_{\{h,l\}}$, we reformulate

$$\begin{aligned}
&\mathbb{P}\left(|\mathcal{I}_{\{h,l\}}| \geq 3\overline{q}d_{\text{eff}}(\gamma)_{\{h,l\}} \cap \left\{\forall h' \in \{1, \ldots, h\} : \left(\|\mathbf{P}^{h'} - \widetilde{\mathbf{P}}^{h'}\|_2 \leq \varepsilon\right) \leq \varepsilon\right\}\right) \\
&= \mathbb{P}\left(|\mathcal{I}_{\{h,l\}}| \geq 3\overline{q}d_{\text{eff}}(\gamma)_{\{h,l\}} \cap A\right) = \mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} z_{h,i,j} \geq 3\overline{q}d_{\text{eff}}(\gamma)_{\{h,l\}} \cap A\right) \\
&= \mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} z_{h,i,j} \geq 3\overline{q}d_{\text{eff}}(\gamma)_{\{h,l\}} \;\middle|\; A\right)\mathbb{P}\left(A\right).
\end{aligned}$$

While we do know that the $z_{h,i,j}$ are Bernoulli random variables (since they are either 0 or 1), it is not easy to compute the success probability of each $z_{h,i,j}$, and in addition there could be dependencies between $z_{h,i,j}$ and $z_{h,i',j'}$. Similarly to Lem. 8, we are going to find a stochastic variable to dominate $z_{h,i,j}$. Denoting with $u'_{s,i,j} \sim \mathcal{U}(0,1)$ a uniform random variable, we will define $w'_{s,i,j}$ as

$$w'_{s,i,j}|\mathcal{F}_{\{s,i',j'\}} = w'_{s,i,j}|\mathcal{F}_{s-2} \stackrel{\text{def}}{=} \mathbb{I}\left\{u'_{s,i,j} \leq \frac{p_{h,i}}{\widetilde{p}_{s-1,i}}\right\} \sim \mathcal{B}\left(\frac{p_{h,i}}{\widetilde{p}_{s-1,i}}\right)$$

for any i' and j' such that $\{s,1,1\} \leq \{s,i',j'\} < \{s,i,j\}$. Note that $w'_{s,i,j}$, unlike $z_{s,i,j}$, does not have a recursive definition, and its only dependence on any other variable comes from $\widetilde{p}_{s-1,i}$. First, we peel off the last step

$$\begin{aligned}
&\mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} z_{h,i,j} \geq g \;\middle|\; A\right) = \mathop{\mathbb{E}}_{\mathcal{F}_{t-1}|A}\left[\mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} \mathbb{I}\left\{u_{h,i,j} \leq \frac{\widetilde{p}_{h,i}}{\widetilde{p}_{t-1,i}}\right\} z_{h-1,i,j} \geq g \;\middle|\; \mathcal{F}_{t-1}\cap A\right)\right]\\
&\leq \mathop{\mathbb{E}}_{\mathcal{F}_{t-1}|A}\left[\mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} \mathbb{I}\left\{u'_{h,i,j} \leq \frac{p_{h,i}}{\widetilde{p}_{t-1,i}}\right\} z_{h-1,i,j} \geq g \;\middle|\; \mathcal{F}_{t-1}\cap A\right)\right] = \mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} w'_{h,i,j} z_{h-1,i,j} \geq g \;\middle|\; A\right),
\end{aligned}$$

where we used the fact that conditioned on A, $\mathcal{I}_{\{h,l\}}$ is accurate w.r.t. $\mathbf{K}_{\{h,l\}}$, which guarantees that $\widetilde{p}_{h,i} \leq p_{h,i}$. Plugging this in the previous bound,

$$\mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} z_{h,i,j} \geq g \;\middle|\; A\right)\mathbb{P}(A) \leq \mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} w'_{h,i,j} z_{h-1,i,j} \geq g \cap A\right) \leq \mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} w'_{h,i,j} z_{h-1,i,j} \geq g\right).$$

We now proceed by peeling off layers from the end of the chain one by one. We show how to move from an iteration $s \leq h$ to $s-1$.

$$\begin{aligned}
&\mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} w'_{s,i,j} z_{s-1,i,j} \geq g\right) = \mathop{\mathbb{E}}_{\mathcal{F}_{s-2}}\left[\mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} \mathbb{I}\left\{u'_{s,i,j} \leq \frac{p_{h,i}}{\widetilde{p}_{s-1,i}}\right\} z_{s-1,i,j} \geq g \;\middle|\; \mathcal{F}_{s-2}\right)\right]\\
&= \mathop{\mathbb{E}}_{\mathcal{F}_{s-2}}\left[\mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} \mathbb{I}\left\{u'_{s,i,j} \leq \frac{p_{h,i}}{\widetilde{p}_{s-1,i}}\right\}\mathbb{I}\left\{u_{s-1,i,j} \leq \frac{\widetilde{p}_{s-1,i}}{\widetilde{p}_{s-2,i}}\right\} z_{s-2,i,j} \geq g \;\middle|\; \mathcal{F}_{s-2}\right)\right]\\
&= \mathop{\mathbb{E}}_{\mathcal{F}_{s-2}}\left[\mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} \mathbb{I}\left\{u'_{s-1,i,j} \leq \frac{p_{h,i}}{\widetilde{p}_{s-2,i}}\right\} z_{s-2,i,j} \geq g \;\middle|\; \mathcal{F}_{s-2}\right)\right] = \mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} w'_{s-1,i,j} z_{s-2,i,j} \geq g\right)
\end{aligned}$$

Applying this repeatedly from $s = h$ to $s = 2$ we have,

$$\mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} w'_{h,i,j} z_{h-1,i,j} \geq g\right) = \mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} w'_{1,i,j} z_{0,i,j} \geq g\right) = \mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} w'_{1,i,j} \geq g\right).$$

Now, all the $w'_{1,i,j}$ are independent Bernoulli random variables, and we can bound their sum with a Hoeffding-like bound using Markov inequality,

$$\begin{aligned}
&\mathbb{P}\left(\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu} w'_{1,i,j} \geq g\right) = \inf_{\theta>0} \mathbb{P}\left(e^{\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\theta w'_{1,i,j}} \geq e^{\theta g}\right)\\
&\leq \inf_{\theta>0} \frac{\mathbb{E}\left[e^{\sum_{j=1}^{\overline{q}}\sum_{i=1}^{\nu}\theta w'_{1,i,j}}\right]}{e^{\theta g}} = \inf_{\theta>0} \frac{\mathbb{E}\left[\prod_{j=1}^{\overline{q}}\prod_{i=1}^{\nu} e^{\theta w'_{1,i,j}}\right]}{e^{\theta g}} = \inf_{\theta>0} \frac{\prod_{j=1}^{\overline{q}}\prod_{i=1}^{\nu}\mathbb{E}\left[e^{\theta w'_{1,i,j}}\right]}{e^{\theta g}}\\
&= \inf_{\theta>0} \frac{\prod_{j=1}^{\overline{q}}\prod_{i=1}^{\nu}(p_{h,i}e^{\theta} + (1-p_{h,i}))}{e^{\theta g}} = \inf_{\theta>0} \frac{\prod_{j=1}^{\overline{q}}\prod_{i=1}^{\nu}(1 + p_{h,i}(e^{\theta}-1))}{e^{\theta g}}\\
&\leq \inf_{\theta>0} \frac{\prod_{j=1}^{\overline{q}}\prod_{i=1}^{\nu} e^{p_{h,i}(e^{\theta}-1)}}{e^{\theta g}} \leq \inf_{\theta>0} \frac{e^{\overline{q}(e^{\theta}-1)\sum_{i=1}^{\nu}p_{h,i}}}{e^{\theta g}} = \inf_{\theta>0} e^{(d_{\text{eff}}(\gamma)_{\{h,l\}}\overline{q}(e^{\theta}-1)-\theta g)} \leq \inf_{\theta>0} e^{(d_{\text{eff}}(\gamma)_{\{h,l\}}\overline{q}(e^{\theta}-1)-\theta g)},
\end{aligned}$$

where we use the fact that $1 + x \leq e^x$, $w'_{1,i,j} \sim \mathcal{B}(p_{h,i})$ and by Def. 2, $\sum_{i=1}^{\nu} p_{h,i} = \sum_{i=1}^{\nu} \tau_{h,i} = d_{\text{eff}}(\gamma)_{\{h,l\}}$. The choice of θ minimizing the previous expression is obtained as

$$\frac{d}{d\theta} e^{\left(\overline{q} d_{\text{eff}}(\gamma)_{\{h,l\}}(e^{\theta}-1)-\theta g\right)} = e^{\left(\overline{q} d_{\text{eff}}(\gamma)_{\{h,l\}}(e^{\theta}-1)-\theta g\right)} \left(\overline{q} d_{\text{eff}}(\gamma)_{\{h,l\}} e^{\theta} - g\right) = 0,$$

and thus $\theta = \log(g/(\overline{q} d_{\text{eff}}(\gamma)_{\{h,l\}}))$. Plugging this in the previous bound,

$$\begin{aligned}\inf_{\theta} \exp\left\{\overline{q} d_{\text{eff}}(\gamma)_{\{h,l\}}(e^{\theta}-1) - \theta g\right\} &= \exp\left\{g - \overline{q} d_{\text{eff}}(\gamma)_{\{h,l\}} - g \log\left(\frac{g}{\overline{q} d_{\text{eff}}(\gamma)_{\{h,l\}}}\right)\right\} \\ &= \exp\left\{-g\left(\log\left(\frac{g}{\overline{q} d_{\text{eff}}(\gamma)_{\{h,l\}}}\right) - 1\right)\right\} e^{-\overline{q} d_{\text{eff}}(\gamma)_{\{h,l\}}},\end{aligned}$$

and choosing $g = 3\overline{q} d_{\text{eff}}(\gamma)_{\{h,l\}}$, we conclude our proof.

E Applications

Proof of Lemma 5.

$$\begin{aligned}&\widetilde{\mathbf{K}}_n = \mathbf{K}_n \mathbf{S}_n (\mathbf{S}_n^{\mathsf{T}} \mathbf{K}_n \mathbf{S}_n + \gamma \mathbf{I}_n)^{-1} \mathbf{S}_n^{\mathsf{T}} \mathbf{K}_n \\ &= \mathbf{\Phi}_n^{\mathsf{T}} \mathbf{\Phi}_n \mathbf{S}_n (\mathbf{S}_n^{\mathsf{T}} \mathbf{\Phi}_n^{\mathsf{T}} \mathbf{\Phi}_n \mathbf{S}_n + \gamma \mathbf{I}_n)^{-1} \mathbf{S}_n^{\mathsf{T}} \mathbf{\Phi}_n^{\mathsf{T}} \mathbf{\Phi}_n \\ &= \mathbf{\Phi}_n^{\mathsf{T}} \mathbf{\Phi}_n \mathbf{S}_n \mathbf{S}_n^{\mathsf{T}} \mathbf{\Phi}_n^{\mathsf{T}} (\mathbf{\Phi}_n \mathbf{S}_n \mathbf{S}_n^{\mathsf{T}} \mathbf{\Phi}_n^{\mathsf{T}} + \gamma \mathbf{I}_D)^{-1} \mathbf{\Phi}_n \\ &= \mathbf{\Phi}_n^{\mathsf{T}} (\mathbf{\Phi}_n \mathbf{S}_n \mathbf{S}_n^{\mathsf{T}} \mathbf{\Phi}_n^{\mathsf{T}} + \gamma \mathbf{I}_D - \gamma \mathbf{I}_D)(\mathbf{\Phi}_n \mathbf{S}_n \mathbf{S}_n^{\mathsf{T}} \mathbf{\Phi}_n^{\mathsf{T}} + \gamma \mathbf{I}_D)^{-1} \mathbf{\Phi}_n \\ &= \mathbf{\Phi}_n^{\mathsf{T}} (\mathbf{I}_D - \gamma (\mathbf{\Phi}_n \mathbf{S}_n \mathbf{S}_n^{\mathsf{T}} \mathbf{\Phi}_n^{\mathsf{T}} + \gamma \mathbf{I}_D)^{-1}) \mathbf{\Phi}_n \\ &= \mathbf{\Phi}_n^{\mathsf{T}} \mathbf{I}_D \mathbf{\Phi}_n - \gamma \mathbf{\Phi}_n^{\mathsf{T}} (\mathbf{\Phi}_n \mathbf{S}_n \mathbf{S}_n^{\mathsf{T}} \mathbf{\Phi}_n^{\mathsf{T}} + \gamma \mathbf{I}_D)^{-1} \mathbf{\Phi}_n \\ &= \mathbf{K}_n - \gamma \mathbf{\Phi}_n^{\mathsf{T}} (\mathbf{\Phi}_n \mathbf{S}_n \mathbf{S}_n^{\mathsf{T}} \mathbf{\Phi}_n^{\mathsf{T}} + \gamma \mathbf{I}_D)^{-1} \mathbf{\Phi}_n\end{aligned}$$

From Lem. 6, and the fact that $\mathcal{I}_n$ is ε-approximate we have that

$$\left\|(\mathbf{\Phi}_t \mathbf{\Phi}_t^{\mathsf{T}} + \gamma \mathbf{I}_D)^{-1/2} \mathbf{\Phi}_t (\mathbf{I}_t - \mathbf{S}_s \mathbf{S}_s^{\mathsf{T}}) \mathbf{\Phi}_t^{\mathsf{T}} (\mathbf{\Phi}_t \mathbf{\Phi}_t^{\mathsf{T}} + \gamma \mathbf{I}_D)^{-1/2}\right\|_2 \leq \varepsilon,$$

which implies

$$\mathbf{\Phi}_t \mathbf{\Phi}_t^{\mathsf{T}} - \mathbf{\Phi}_t \mathbf{S}_s \mathbf{S}_s^{\mathsf{T}} \mathbf{\Phi}_t^{\mathsf{T}} \preceq \varepsilon (\mathbf{\Phi}_t \mathbf{\Phi}_t^{\mathsf{T}} + \gamma \mathbf{I}_D)$$

and

$$\mathbf{\Phi}_t \mathbf{\Phi}_t^{\mathsf{T}} - \varepsilon (\mathbf{\Phi}_t \mathbf{\Phi}_t^{\mathsf{T}} + \gamma \mathbf{I}_D) \preceq \mathbf{\Phi}_t \mathbf{S}_s \mathbf{S}_s^{\mathsf{T}} \mathbf{\Phi}_t^{;} \mathsf{T}$$

Therefore,

$$\begin{aligned}\mathbf{K}_n - \widetilde{\mathbf{K}}_n &= \gamma \mathbf{\Phi}_n^{\mathsf{T}} (\mathbf{\Phi}_n \mathbf{S}_n \mathbf{S}_n^{\mathsf{T}} \mathbf{\Phi}_n^{\mathsf{T}} + \gamma \mathbf{I}_D)^{-1} \mathbf{\Phi}_n \preceq \gamma \mathbf{\Phi}_n^{\mathsf{T}} (\mathbf{\Phi}_t \mathbf{\Phi}_t^{\mathsf{T}} - \varepsilon (\mathbf{\Phi}_t \mathbf{\Phi}_t^{\mathsf{T}} + \gamma \mathbf{I}_D) + \gamma \mathbf{I}_D)^{-1} \mathbf{\Phi}_n \\ &= \frac{\gamma}{1-\varepsilon} \mathbf{\Phi}_n^{\mathsf{T}} (\mathbf{\Phi}_t \mathbf{\Phi}_t^{\mathsf{T}} + \gamma \mathbf{I}_D)^{-1} \mathbf{\Phi}_n = \frac{\gamma}{1-\varepsilon} \mathbf{K}_t (\mathbf{K}_t + \gamma \mathbf{I})^{-1} \preceq \frac{\gamma}{1-\varepsilon} \mathbf{I}.\end{aligned}$$

□