DYNAMIC ESTIMATION OF SLOWLY VARYING SEQUENCES

Anonymous authors
Paper under double-blind review

ABSTRACT

We consider the problem of sequentially approximating a target function of each element in a slowly-varying sequence, i.e. one where the magnitude α_i of the difference between the elements at positions i and $i-1$, measured in an appropriate norm, is small. Recent work on implicit trace estimation shows that when α_i is small in Frobenius norm, reusing queries to past sequence elements can reduce the overall cost (Dharangutte and Musco, 2021; Woodruff et al., 2022). We introduce a framework that generalizes this approach to linear and nonlinear functions on normed vector spaces, obtaining sequential estimation results for matrix powers, spectral densities, Monte Carlo integration, and certain boundary value problems from partial differential equations (PDEs). Furthermore, we develop an algorithm that locally scales the estimation budget with α_i, obtaining sharper path-length-style variation bounds of form $\mathcal{O}\left(\frac{1}{\epsilon^2}\sum_{i=1}^{m}\alpha_i\right)$ on the cost of estimating a sequence of length m to accuracy ϵ. This improves upon the earlier implicit trace estimation dependence on $\frac{m}{\epsilon^2}\max_i \alpha_i$ (Dharangutte and Musco, 2021), which results from fixing the query budget using the worst-case step size α_i and is thus inefficient for stable sequences with rare bursts. Furthermore, while prior theoretical guarantees assume a known bound on the step sizes, we show in certain cases how to estimate them on the fly with (nearly) no added cost. Lastly, we complement our theoretical results with experiments demonstrating improved accuracy–cost trade-offs over prior baselines on trace estimation and Hessian tracking. In summary, our framework makes sequential approximation general-purpose and adaptive while sharpening state-of-the-art guarantees for dynamic trace estimation.

1 INTRODUCTION

In many areas of theory and practice, we face a slowly evolving system and wish to estimate its properties given only limited, structured access to its state. Examples of this problem include estimating evolving probability distributions given sample access (Mazzetto and Upfal, 2023; Mazzetto, 2024), tracking connectivity and centrality metrics in evolving social networks given only local subgraphs (Estrada and Hatano, 2008; Wang et al., 2021; Chan et al., 2014; Chen et al., 2018; Beckermann et al., 2018), maintaining statistics of dynamic data streams (Zhu and Shasha, 2002; Jayram et al., 2008; Frahling et al., 2005), or tracking the loss curvature during neural network optimization (Li et al., 2018; Yao et al., 2020; Ghorbani et al., 2019).

We propose a general framework for such estimation problems when the underlying sequence consists of m elements of a vector space and the distance between consecutive elements $i-1$ and i is α_i. This setting is inspired by past work on estimating traces given only matrix-vector product queries and a bound on the maximum Frobenius norm between consecutive matrices (Dharangutte and Musco, 2021; Woodruff et al., 2022); our framework enables us to find direct applicability to other vector spaces (e.g. L^∞ functions), derive results for certain nonlinear maps (e.g. eigenvalue distributions), develop a novel algorithm whose cost scales with $\sum_i \alpha_i$ (instead of prior works' looser bound of $m \cdot \max_i \alpha_i$), and design an efficient mechanism that estimates α_i on-the-fly (instead of assuming a known global bound). To achieve this, we require only a linear estimator that has sub-exponential concentration around its target property for every element of the vector space, with concentration parameters that scale with the element's norm. Such estimators exist in multiple settings and yield results for both linear and nonlinear properties, as evidenced in our contributions:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1. **General framework:** We identify the existence of what we term a *norm-concentrated estimator* of the target quantity, one whose concentration scales with the norm of the input, as a key property that enabled dynamic trace estimation in past work. This leads to a meta-algorithm and associated guarantees for sequential stochastic approximation that can be applied to a wide range of both linear and nonlinear problems, including traces of matrix powers, spectral densities, function integrals, and solutions to certain boundary value PDEs.
2. **Efficiency:** Our algorithm dynamically adjusts the budget given to the norm-concentrated estimator based on the local change α_i between consecutive elements, resulting in a total cost that scales with the sum $\sum_{i=1}^{m} \alpha_i$ of the local changes rather than the worst case change $m \cdot \max_i \alpha_i$. This leads to a direct improvement in the query complexity of dynamic trace estimation relative to past work (Dharangutte and Musco, 2021).
3. **Adaptivity:** We identify *auto-concentration*, an extension of norm-concentration, as a property that allows us to estimate the local change α_i on-the-fly with (nearly) no added cost. This enables us to remove the assumption of a known bound on the step sizes α_i in dynamic trace estimation while remaining provably accurate, another improvement over past work (Dharangutte and Musco, 2021).
4. **Empirical results:** Using experiments on synthetic matrix sequences and Hessians from neural network optimization trajectories, we show that our algorithm requires significantly fewer samples than prior methods for dynamic trace estimation. Our approach is simple to implement and can be applied to other estimators as well.

Related Work. A representative problem in this setting is trace estimation in the implicit matrix model. Here, we have **matrix-vector product** (MVP) access to a sequence of matrices $A_1, \ldots, A_m$ (for simplicity, assume $\|A_i\|_F \leq 1$ after scaling), and we wish to maintain an approximation of $\text{tr}(A_i)$ up to additive ϵ error every step while minimizing the total number of MVPs. This problem has received a lot of recent attention due to its importance in machine learning and data science, such as when analyzing moments of the Hessian spectral density during neural network optimization (Yao et al., 2020; Ghorbani et al., 2019), counting triangles in massive, dynamically changing graphs (Avron, 2010; Estrada and Hatano, 2008), while training Gaussian processes (Dong et al., 2017; Fitzsimons et al., 2017), and so on. The assumption that consecutive matrices are close is natural in these settings: a training-loss Hessian evolves gradually, a graph receives a few edge updates per round, a covariance matrix receives rank-one modifications, etc. In such smoothly evolving settings, naively applying a static estimator, such as Hutchinson's estimator for traces (Hutchinson, 1989), at each step leads to an undesired multiplicative $O(m)$ overhead. Recent work has shown how to dramatically improve on this (Dharangutte and Musco, 2021; Woodruff et al., 2022), obtaining query complexity that scales with m times the largest local change, $m \cdot \max_i \alpha_i$, where $\alpha_i := \|A_i - A_{i-1}\|_F$. We obtain query complexity scaling with $\sum_i \alpha_i$, a natural measure of sequence variation similar to path-length-style bounds in online learning (Auer et al., 2002; Zinkevich, 2003), and show how their approach can be broadly generalized to other sequential estimation problems.

2 The dynamic estimation problem

We first set up the general problem and specify our assumptions regarding the static estimator. Let V be a normed vector space with norm $\|\cdot\|_V$, and consider a linear map $L : V \to \mathbb{R}$ that we *cannot* compute directly. We are interested in estimating $L(v_i)$ for each element in a sequence $v_1, \ldots, v_m \in V$ given some setting-specific access model (e.g. MVPs for $V = \mathbb{R}^{n \times n}$) that can be used by some randomized estimator that we will discuss shortly. Our goal is to adapt to sequences that change relatively slowly between consecutive steps, which we will quantify by the local **step size**

$$\alpha_i = \|v_i - v_{i-1}\|_V \qquad \text{for} \qquad i = 1, \ldots, m \quad \text{and} \quad v_0 = 0 \tag{1}$$

We treat these local step sizes as known in Section 2.1 and generalize to the *unknown* case in Section 2.2. As our query model, we assume access to a randomized estimator $\mathcal{E}(v, k)$ of $L(v)$ that queries elements $v \in V$ with budget $k > 0$ using the setting-specific access model. Then the **dynamic estimation problem** is to output an estimate $\tilde{L}_i$ of $L(v_i)$ at each time $i = 1, \ldots, m$ using only queries of form $\mathcal{E}(u_i, k_i)$, where $k_i > 0$ is the query budget at time i and $u_i \in V$ is a linear combination of $v_1, \ldots, v_i$ with query cost $q > 0$; the goal is to minimize the total query cost $q \sum_{i=1}^{m} k_i$ while guaranteeing w.p. $\geq 1 - \delta$ that $|\tilde{L}_i - L(v_i)| \leq \epsilon$ simultaneously $\forall\, i \in [m]$. Note that the actual input $u_i \in V$ to the estimator can be *implicit*, i.e. we rely on the estimator $\mathcal{E}(u_i, k_i)$ being computable given only structured query access to the components $v_1, \ldots, v_i$ of u_i.

Table 1: Overview of the dynamic estimation applications we consider and their framework-specific details. While nonlinear in the underlying object, both traces of matrix powers and spectral density estimation can be reduced to dynamic trace estimation over A_i^p; the Dirichlet solution can be represented as linear in g_i. The norm column gives the norm defining the step sizes α_i for each application.

Application	Underlying Object	Map	Norm	Estimator
Trace Estimation (§3)	matrix $A_i \in \mathbb{R}^{n\times n}$	$\operatorname{tr}(A_i)$	$\|\cdot\|_F$	Hutchinson's
– Matrix Power (§3.1)	matrix $A_i \in \mathbb{R}^{n\times n}$	$\operatorname{tr}(A_i^p)$	$\|\cdot\|_F$	Hutchinson's
– Spectral Density (§3.2)	symmetric $A_i \in \mathbb{R}^{n\times n}$	$\frac{1}{\sqrt{n}}\operatorname{tr}(A_i^p), p = 1, \ldots, \mathcal{O}(\frac{1}{\epsilon})$	$\|\cdot\|_F$	Hutchinson's
Function Integration (§4.1)	$f_i : \Omega \mapsto \mathbb{R}, \|f_i\|_\infty \leq 1$	$\mathbb{E}_{X\sim P}[f_i(X)]$	$\|\cdot\|_2$	Monte Carlo
– Dirichlet Problem (§4.2)	$g_i : \partial\Omega \mapsto \mathbb{R}, \|g_i\|_\infty \leq 1$	$u(x)$ s.t. $\Delta u\|_\Omega = 0$ & $u\|_{\partial\Omega} = g_i$	$\|\cdot\|_2$	Monte Carlo

To understand what we need from $\mathcal{E}$ to solve this problem, we consider the concrete instantiation studied by Dharangutte and Musco (2021), in which $V = \mathbb{R}^{n\times n}$ has the Frobenius norm and L is the trace operator. Their algorithm, **DeltaShift**, outputs a sequence of estimates of the traces $L(v_i)$ that are linear combinations of Hutchinson's estimates of the traces $L(u_i)$ of their *damped differences* $u_i = v_i - (1-\gamma)v_{i-1} \in V$ for $\gamma = \alpha$, where $\alpha = \max_i \alpha_i$. To combine these estimates, they rely on the sub-exponential concentration of Hutchinson's estimator around its mean, a property we will also require of $\mathcal{E}$. However, most crucial for their result is the fact that the estimator's variance scales not only with (the reciprocal of) the number of queries but also with the norm of the input u_i, which is why estimating the differences $L(u_i)$ allows them to adapt to small α.[1] We formalize this in the following definition, which underlies the key assumption of our framework.

Definition 1 (Norm-Concentrated Estimator)**.** *A randomized estimator $\mathcal{E}(v,k)$ is* **norm-concentrated with power** $p > 0$ *if there exist constants $c_1, c_2, c_3 \geq 0$ such that for all elements $v \in V$ and integer budget parameters $k \geq 1$ the random variable $\mathcal{E}(v,k)$ is subexponentially distributed with parameters ν and β satisfying*

$$\nu \leq \frac{c_1\|v\|_V^p}{\sqrt{k}} \quad \textit{and} \quad \beta \leq \frac{c_2\|v\|_V^p + c_3}{k} \tag{2}$$

i.e. $\mathbb{E}\left[\exp\left(\lambda(\mathcal{E}(v,k) - \mu)\right)\right] \leq \exp\left(\frac{1}{2}\lambda^2\nu^2\right)\ \forall\ |\lambda| < 1/\beta$*, where* $\mu = \mathbb{E}[\mathcal{E}(v,k)]$*. Unless otherwise specified, we will use "norm-concentrated" to refer to the case* $p = 1$*.*

While Hutchinson's trace estimator is the motivating example of a norm-concentrated estimator, another basic example is the Monte Carlo estimator for integrals. As we show later, quantities as diverse as spectral densities and boundary value solutions can also be estimated by extending these approaches. The assumption is also reminiscent of the strong growth (or relative noise) condition in stochastic optimization, under which an unbiased estimator $g(x,\xi)$ of the gradient $\nabla f(x)$ satisfies $\mathbb{E}_\xi\|g(x,\xi)\|^2 \leq C\|\nabla f(x)\|^2$ for constant $C \geq 1$ (Schmidt and Le Roux, 2013; Vaswani et al., 2019). Both conditions require the stochastic variability to shrink with the magnitude of the underlying input.

2.1 Dynamic estimation with known step sizes

The naive approach to solving the dynamic estimation problem is to compute $\mathcal{E}(v_i,k)$ at every step i with a sufficiently large fixed budget k to guarantee $|\mathcal{E}(v_i,k) - L(v_i)| \leq \epsilon$ w.p. $\geq 1 - \frac{\delta}{m}$ and then apply a union bound, which incurs a total budget of $\mathcal{O}\left(\frac{m}{\epsilon^2}\log\frac{m}{\delta}\right)$. For traces, DeltaShift improves on this by aggregating estimates $\mathcal{E}(u_i,k)$ of the damped differences $u_i = v_i - (1-\gamma)v_{i-1}$ described before while using a fixed budget $k = \mathcal{O}\left(\frac{\alpha}{\epsilon^2}\log\frac{m}{\delta}\right)$ queries at every step after initialization, thus attaining a total cost $\mathcal{O}\left(\frac{1+m\alpha}{\epsilon^2}\log\frac{m}{\delta}\right)$. This approach is conceptually related to certain recurrent variance-reduction estimators in stochastic optimization (Nguyen et al., 2017; Fang et al., 2018), which similarly maintain a state and estimate only a small residual at each step.

We now improve upon this even further by dynamically rescaling both the damping factor and the query budget at each time step to account for the local step size α_i, as described in Algorithm 1; this yields the following path-length-style bound on the total cost:

[1] Observe that $\|u_i\|_V \leq \|v_i - v_{i-1}\|_V + \gamma\|v_{i-1}\|_V \leq \max_i\|v_i - v_{i-1}\|_V + \gamma = 2\alpha$.

Algorithm 1: AdaDeltaShift

Input: sequence $v_1, \dots, v_m$, step sizes $\alpha_1, \dots, \alpha_m$, target accuracy ϵ, failure probability $\delta > 0$, norm-concentrated estimator $\mathcal{E}$ with constants c_1, c_2, c_3

$N \leftarrow \frac{\epsilon^2}{2\log(2m/\delta)}, B \leftarrow \frac{\epsilon}{2\log(2m/\delta)}$ // set concentration thresholds

$v_0 \leftarrow 0, \tilde{L}_0 \leftarrow 0$ // initialize running estimate

for $i = 1, \dots, m$ **do**

- $\gamma_i \leftarrow \min(1, \alpha_i)$ // set damping factor
- $u_i \leftarrow v_i - (1-\gamma_i)v_{i-1}$ // define implicit damped residual
- $k_i \leftarrow \left\lceil \max\left(\frac{4c_1^2\gamma_i}{N}, \frac{2c_2\gamma_i + c_3 1_{\gamma_i > 0}}{B}\right)\right\rceil$ // set query budget
- $\hat{L}_i \leftarrow \mathcal{E}(u_i, k_i)$ // estimate residual map
- $\tilde{L}_i \leftarrow (1-\gamma_i)\tilde{L}_{i-1} + \hat{L}_i$ // update running estimate

Output: sequence of running estimates $\tilde{L}_1, \dots, \tilde{L}_m$

Theorem 1 (Dynamic Sample Complexity, known step sizes)**.** *Suppose in Algorithm 1 we have $\|v_i\|_V \le 1$ and $\alpha_i \ge \|v_i - v_{i-1}\|_V$ for all $i \in [m]$, and let $\mathcal{E}$ be a norm-concentrated unbiased estimator of L that Algorithm 1 invokes with independent randomness at every step i. Then w.p. $\ge 1-\delta$ its estimates $\tilde{L}_1, \dots, \tilde{L}_m$ satisfy $|\tilde{L}_i - L(v_i)| \le \epsilon$ simultaneously for all $i \in [m]$ and the total budget spent querying $\mathcal{E}$ (with queries consisting of linear combinations of at most two elements of V) is bounded by*

$$\mathcal{O}\left(\left(\left(\frac{c_1^2}{\epsilon^2} + \frac{c_2}{\epsilon}\right)\sum_{i=1}^{m}\alpha_i + \frac{c_3 m}{\epsilon}\right)\log\frac{m}{\delta} + \mathrm{nnz}_i(\alpha_i)\right) \tag{3}$$

calls to $u_i = v_i - (1-\gamma_i)v_{i-1}$, where $\mathrm{nnz}_i(\alpha_i) \le m$ is the number of nonzero step size bounds α_i.[2]

Note that the non-dynamic baseline has sample complexity $\tilde{\mathcal{O}}(\frac{m}{\epsilon^2})$, so Theorem 1 improves upon it whenever $\sum_{i=1}^m \alpha_i = o(m)$. More importantly, the total cost in Theorem 1 scales with the sum of local changes $\sum_i \alpha_i$, rather than the worst-case change $m \cdot \max_i \alpha_i$. This yields a direct improvement in the query complexity of dynamic trace estimation over prior work (Dharangutte and Musco, 2021). Below we prove this result by induction on the sub-exponential parameters of the estimates at each time, before proceeding to handle the case of unknown step sizes via on-the-fly norm estimation.

Proof. Let $e_i = \tilde{L}_i - L(v_i)$. We prove inductively that e_i is sub-exponential with parameters $\nu_i^2 \le N = \frac{\epsilon^2}{2\log(2m/\delta)}$ and $\beta_i \le B = \frac{\epsilon}{2\log(2m/\delta)}$. By standard sub-exponential tail bounds (Lemma 1), this guarantees $\mathbb{P}[|e_i| \ge \epsilon] \le 2\exp\left(-\min\left(\frac{\epsilon^2}{2N}, \frac{\epsilon}{2B}\right)\right) \le \frac{\delta}{m}$. A union bound over m steps then yields the global success probability of at least $1-\delta$.

The base case follows from $e_0 = \tilde{L}_0 - L(v_0) = -L(v_0)$ being a constant, which is sub-exponential with parameters $\nu_0^2 = \beta_0 = 0$. For $i > 0$, linearity of L gives $e_i = (1-\gamma_i)e_{i-1} + \eta_i$, where $\eta_i = \hat{L}_i - L(u_i)$. If $\alpha_i \le 1$, then $\gamma_i = \alpha_i$ and

$$\|u_i\|_V = \|v_i - v_{i-1} + \gamma_i v_{i-1}\|_V \le \|v_i - v_{i-1}\|_V + \gamma_i\|v_{i-1}\|_V \le \alpha_i + \alpha_i = 2\alpha_i \tag{4}$$

Otherwise, if $\alpha_i > 1$, then $\gamma_i = 1$ so $\|u_i\|_V = \|v_i\|_V \le 1 \le 2\gamma_i$. Thus $\|u_i\|_V \le 2\gamma_i$ in both cases, so η_i is sub-exponential with parameters $\nu_{\mathrm{fresh}}^2 \le \frac{4c_1^2\gamma_i^2}{k_i} \le \gamma_i N$ and $\beta_{\mathrm{fresh}} \le \frac{2c_2\gamma_i + c_3}{k_i} \le B$. Since η_i is independent of e_{i-1}, by composition of sub-exponential random variables (Lemma 2) we have that $e_i = (1-\gamma_i)e_{i-1} + \eta_i$ is sub-exponential with parameters

$$\nu_i^2 \le (1-\gamma_i)^2 N + \gamma_i N \le N \qquad \text{and} \qquad \beta_i \le \max\{(1-\gamma_i)B, B\} = B \tag{5}$$

as desired. Lastly, to count the number of queries to $u_i = v_i - (1-\gamma_i)v_{i-1}$, we bound k_i to get

$$\sum_{i=1}^{m} k_i \le \sum_{i=1}^{m} \frac{4c_1^2\gamma_i}{N} + \frac{2c_2\gamma_i + c_3 1_{\gamma_i > 0}}{B} + 1_{\gamma_i > 0} \tag{6}$$

Since $\gamma_i \le \alpha_i$, substituting the definitions of N and B yields the stated bound (3). □

[2]The low-order (ϵ-independent) term $\mathrm{nnz}_i(\alpha_i) = \mathcal{O}(m)$ is present due to the need to ensure integrality of k_i; it should also appear (but was omitted) in the DeltaShift bound (Dharangutte and Musco, 2021, Theorem 1.1).

2.2 Dynamic estimation with unknown step sizes

In the preceding analysis, we assumed explicit knowledge of the local step sizes $\alpha_i = \|v_i - v_{i-1}\|_V$ to set our parameters. While in practice α_i is often unknown, as we show next it can often be efficiently estimated on-the-fly. In fact, if distances between elements can be approximated with the following specific kind of norm-concentrated estimator, then Algorithm 1 can be run with cheap over-estimates of α_i that suffice to (nearly) preserve the guarantees of Theorem 1.

Definition 2 (Auto-Concentrated Estimator)**.** *A nonnegative, unbiased estimator $\mathcal{E}_V(v, k)$ of the squared norm $\|v\|_V^2$ is* **auto-concentrated** *if there exist constants $a_1, a_2 > 0$ such that $\mathcal{E}_V$ is norm-concentrated with power $p = 2$ and constants $a_1, a_2, 0$ (corresponding to c_1, c_2, c_3 in Definition 1).*

The natural case of an auto-concentrated estimator is Hutchinson's estimator for the squared Frobenius norm, which is derived from Hutchinson's trace estimator. Such estimators give probabilistic upper bounds on α_i at each step i by fixing some $\rho \in (0, 1)$ and calling $\mathcal{E}_V(v_i - v_{i-1}, r)$ with $r = \lceil 2\max\{\frac{a_1^2}{\rho^2}, \frac{a_2}{\rho}\} \log \frac{4m}{\delta} \rceil$ queries to $v_i - v_{i-1}$. Sub-exponential concentration (Lemma 1) then implies that for $\alpha_i > 0$ we have

$$\mathbb{P}\left[|\mathcal{E}_V(v_i - v_{i-1}, r) - \alpha_i^2| > \rho\alpha_i^2\right] \leq 2\exp\left(-\min\left\{\frac{\rho^2 r}{2a_1^2}, \frac{\rho r}{2a_2}\right\}\right) \leq \frac{\delta}{2m} \tag{7}$$

while for $\alpha_i = 0$ we have by unbiasedness and nonnegativity that $\mathcal{E}_V(v_i - v_{i-1}, r) = 0$ a.s. Thus w.p. $\geq 1 - \frac{\delta}{2}$ we have $|\mathcal{E}_V(v_i - v_{i-1}, r) - \alpha_i^2| \leq \rho\alpha_i^2$ simultaneously for all i, which is equivalent to

$$\alpha_i\sqrt{1-\rho} \leq \sqrt{\mathcal{E}_V(v_i - v_{i-1}, r)} \leq \alpha_i\sqrt{1+\rho} \tag{8}$$

This allows us to run Algorithm 1 with cheap estimates $\tilde{\alpha}_i$ that are probabilistically guaranteed to be a small multiplicative factor greater than the true step sizes α_i, as described in the following result:

Theorem 2 (Dynamic Sample Complexity, unknown step sizes)**.** *Under the assumptions of Theorem 1, suppose we additionally have an estimator $\mathcal{E}_V$ of $\|\cdot\|_V^2$ that is auto-concentrated with constants a_1, a_2 and that we invoke it independently of $\mathcal{E}$ at every step $i = 1, \ldots, m$ to run Algorithm 1 but with failure probability $\frac{\delta}{2}$ and setting step size bounds α_i via the following estimates for fixed $\rho \in (0, 1)$:*

$$\tilde{\alpha}_i = \sqrt{\frac{\mathcal{E}_V(v_i - v_{i-1}, r)}{1-\rho}} \qquad \text{for} \qquad r = \left\lceil 2\max\left\{\frac{a_1^2}{\rho^2}, \frac{a_2}{\rho}\right\} \log\frac{4m}{\delta}\right\rceil \tag{9}$$

Then w.p. $\geq 1 - \delta$, the method succeeds (i.e. $|\tilde{L}_i - L(v_i)| \leq \epsilon\ \forall\ i \in [m]$) and uses a query budget of

$$\mathcal{O}\left(\left(\left(\frac{c_1^2}{\epsilon^2} + \frac{c_2}{\epsilon}\right)\sqrt{\frac{1+\rho}{1-\rho}}\sum_{i=1}^{m}\alpha_i + \frac{c_3 m}{\epsilon}\right)\log\frac{m}{\delta} + \mathrm{nnz}_i(\alpha_i)\right) \tag{10}$$

for $\mathcal{E}$ at $v_i - (1-\gamma_i)v_{i-1}$ and a total query budget of $\mathcal{O}\left(\left(\frac{a_1^2}{\rho^2} + \frac{a_2}{\rho}\right) m \log\frac{m}{\delta}\right)$ for $\mathcal{E}_V$ at $v_i - v_{i-1}$.

For constant a_1, a_2, ρ, Theorem 2 only has an extra overhead of $\mathcal{O}(m \log \frac{m}{\delta})$ compared to Theorem 1, with no added dependency on ϵ. Thus, given an auto-concentrated estimator of the norm of the sequence elements, we obtain adaptivity to unknown step sizes nearly for free. The parameter ρ controls a tradeoff between the two query budgets: smaller ρ gives tighter upper bounds on the step sizes but requires more queries to $\mathcal{E}_V$, while larger ρ reduces the norm-estimation budget at the cost of a larger factor $\sqrt{(1+\rho)/(1-\rho)}$ multiplying the adaptive term in the budget for $\mathcal{E}$. In our applications, we can think of ρ as a fixed constant.

Proof. Success follows from the union of (a) Equation 8's guarantee that $\tilde{\alpha}_i \geq \alpha_i \geq \|v_i - v_{i-1}\|_V$ holds for all $i \in [m]$ w.p. $\geq 1 - \frac{\delta}{2}$ with (b) Theorem 1's guarantee with failure probability $\frac{\delta}{2}$. The number of queries to $\mathcal{E}$ is given in Equation 3 followed by the bound $\gamma_i \leq \tilde{\alpha}_i \leq \alpha_i\sqrt{(1+\rho)/(1-\rho)}$, while the number of queries to $\mathcal{E}_V$ is simply the number of queries mr to $v_i - v_{i-1}$. □

3 Application: Dynamic estimation of matrix properties

We first apply our general framework to obtain improved guarantees for dynamic matrix trace estimation in the matrix-vector product model, as introduced in Section 1. In this application, the vector space $V = \mathbb{R}^{n\times n}$ is equipped with the Frobenius norm and the linear map is $L(A) = \mathrm{tr}(A)$. Afterwards, we work in the same MVP model to show results for dynamic estimation of two related nonlinear quantities: traces of matrix powers and spectral densities.

To query matrix $A \in \mathbb{R}^{n\times n}$, we will use the standard randomized algorithm for implicit trace estimation, Hutchinson's estimator (Hutchinson, 1989). For a budget of k queries, it is defined as

$$H_k(A) := \frac{1}{k}\sum_{j=1}^{k} g_j^\top A g_j \qquad \text{for i.i.d. Rademacher vectors} \qquad g_1,\dots,g_k \in \{\pm 1\}^n \tag{11}$$

It is well-known that Hutchinson's estimator is unbiased and sub-exponential (Rudelson and Vershynin, 2013) with parameters $\nu \le c_1\|A\|_F/\sqrt{k}$ and $\beta \le c_2\|A\|_F/k$ for some constants $c_1, c_2 \ge 0$, making it a norm-concentrated estimator by Definition 1.[3]

In addition to its use for traces, Hutchinson's is also well-known to yield an unbiased estimator of the squared Frobenius norm by applying it to the positive semi-definite (PSD) Gram matrix $A^\top A$ and using $\|A\|_F^2 = \mathrm{tr}(A^\top A)$. The Gram matrix being PSD means its Frobenius norm is bounded by its trace and thus $\|A^\top A\|_F \le \mathrm{tr}(A^\top A) = \|A\|_F^2$, so $H_k(A^\top A)$ is *also* sub-exponential but with parameters $\nu \le c_1\|A\|_F^2/\sqrt{k}$ and $\beta \le c_2\|A\|_F^2/k$. With the nonnegativity of $H_k(A^\top A)$ (which also follows from PSD), $H_k(A^\top A)$ is thus an auto-concentrated estimator of $\|A\|_F^2$ by Definition 2.

For the trace-estimation and matrix-power bounds below, we fix $\rho \in (0,1)$, and the hidden constants may depend on ρ. We can thus run Algorithm 1 in the manner described in Theorem 2, with H_k applied to both the damped differences $U_i = A_i - (1-\gamma_i)A_{i-1}$ and to the Gram matrices $\Delta_i^\top \Delta_i$ of the differences $\Delta_i = A_i - A_{i-1}$. Since both $H_k(U_i)$ and $H_k(\Delta_i^\top \Delta_i)$ cost $2k$ MVPs each, their respective norm-concentration and auto-concentration imply the following corollary:

Corollary 3 (Dynamic Trace Estimation, unknown step sizes)**.** *Suppose we are given a sequence $A_1,\dots,A_m \in \mathbb{R}^{n\times n}$ with $A_0 = 0$, $\|A_i\|_F \le 1$ for all $i \in [m]$, and unknown $\alpha_i = \|A_i - A_{i-1}\|_F$ and we run Algorithm 1 as described in Theorem 2, with Hutchinson's trace and squared-norm estimators as $\mathcal{E}$ and $\mathcal{E}_V$, respectively. Then w.p. $\ge 1-\delta$ the resulting trace estimates $\tilde{T}_1 = \tilde{L}_1,\dots,\tilde{T}_m = \tilde{L}_m$ satisfy $|\tilde{T}_i - \mathrm{tr}(A_i)| \le \epsilon$ simultaneously for all $i \in [m]$ and the total number of MVPs is bounded by $\mathcal{O}\left(\left(m + \frac{1}{\epsilon^2}\sum_{i=1}^m \alpha_i\right)\log\frac{m}{\delta}\right)$.*

The bound of Dharangutte and Musco (2021) for known steps is $\mathcal{O}\left(\frac{1}{\epsilon^2}\log\frac{m}{\delta}\,(1 + m\cdot\max_i \alpha_i)\right)$. As noted earlier, our result replaces the $m\cdot\max_i\alpha_i$ term with the sharper quantity $\sum_{i=1}^m \alpha_i$. Moreover, it provides a way to estimate the step sizes on the fly with only an ϵ-independent $m\log\frac{m}{\delta}$ overhead.

3.1 Traces of matrix powers

We now show that our framework naturally extends to tracking the trace of matrix powers, which we do in the same MVP setting over $V = \mathbb{R}^{n\times n}$ equipped with the Frobenius norm. Fix an integer $p \ge 1$. While at first glance $\mathrm{tr}(A^p)$ is a nonlinear function of A, we can instead apply the previous result (Corollary 3) to the sequence $A_1^p,\dots,A_m^p$ to get an ϵ-accuracy guarantee and a query cost that scales with $\sum_{i=1}^m \|A_i^p - A_{i-1}^p\|_F$ (rather than the desired $\sum_{i=1}^m \alpha_i = \sum_{i=1}^m \|A_i - A_{i-1}\|_F$). The remaining questions are thus (1) whether we can get a result in terms of the Frobenius differences $\alpha_i = \|A_i - A_{i-1}\|_F$ instead and (2) the MVP cost of applying the Hutchinson estimators to $A_i^p - (1-\gamma_i)A_{i-1}^p$ and the Gram matrix of $A_i^p - A_{i-1}^p$. For the latter, it is easy to see that H_k can be applied to both with $2pk$ MVPs each by simply applying the matrices sequentially, i.e. $A_i^p g = A_i(A_i(\cdots(A_i g)))$. This also avoids explicitly forming potentially dense matrix powers by requiring only matrix-vector access to A_i. For the former, we can use the fact that the change in the pth power can be written as $A_i^p - A_{i-1}^p = \sum_{j=0}^{p-1} A_i^j(A_i - A_{i-1})A_{i-1}^{p-1-j}$, so by the sub-multiplicativity of the Frobenius norm ($\|XY\|_F \le \|X\|_2\|Y\|_F$) and using $\|A_i\|_2 \le \|A_i\|_F \le 1$, we obtain

$$\begin{aligned}\|A_i^p - A_{i-1}^p\|_F &\le \min\left\{\|A_i^p\|_F + \|A_{i-1}^p\|_F, \sum_{j=0}^{p-1}\|A_i^j\|_2\|A_i - A_{i-1}\|_F\|A_{i-1}^{p-1-j}\|_2\right\} \\ &\le \min\left\{\|A_i^{p-1}\|_2\|A_i\|_F + \|A_{i-1}^{p-1}\|_2\|A_{i-1}\|_F, p\alpha_i\right\} \le \min\{2, p\alpha_i\}\end{aligned} \tag{12}$$

We thus obtain the following result directly from Corollary 3, with $\sum_{i=1}^m \alpha_i$ replaced by $\min\{m, p\sum_{i=1}^m \alpha_i\}$ by (1), and an additional factor of p in the MVP cost by (2):

[3] Specifically, use Claim B.3 of Dharangutte and Musco (2021), then bound the spectral norm by the Frobenius.

Corollary 4 (Dynamic Estimation of Traces of Matrix Powers, unknown step sizes)**.** *Suppose we are given a sequence $A_1, \dots, A_m \in \mathbb{R}^{n\times n}$ with $A_0 = 0$, $\|A_i\|_F \le 1$ for all $i \in [m]$, and unknown $\alpha_i = \|A_i - A_{i-1}\|_F$ and we run Algorithm 1 as described in Theorem 2 on the matrix powers $A_1^p, \dots, A_m^p$, with Hutchinson's trace and squared-norm estimators as $\mathcal{E}$ and $\mathcal{E}_V$, respectively. Then w.p. $\ge 1-\delta$ the resulting trace power estimates $\tilde{T}_1 = \tilde{L}_1, \dots, \tilde{T}_m = \tilde{L}_m$ satisfy $|\tilde{T}_i - \mathrm{tr}(A_i^p)| \le \epsilon$ simultaneously for all $i \in [m]$ and the total number of MVPs is bounded by $\mathcal{O}\left(\left(m + \frac{1}{\epsilon^2}\min\left\{m, p\sum_{i=1}^m \alpha_i\right\}\right) p \log\frac{m}{\delta}\right)$.*

Thus for small path-lengths $\sum_{i=1}^m \alpha_i = \mathcal{O}(m\epsilon^2/p)$, the cost (up to log factors) is at most $\mathcal{O}(pm)$, while recovering the naive bound of $\mathcal{O}(mp/\epsilon^2)$ otherwise.

3.2 Spectral density estimation

We next use estimates of traces of matrix powers to track the spectral distribution of a changing symmetric matrix A with spectral radius at most one, again in the MVP setting. This application, called *spectral density estimation* (SDE), arises in dynamic settings such as tracking Hessian spectra during neural network training (Ghorbani et al., 2019). The spectral distribution itself, which corresponds to the uniform distribution over the eigenvalues of A weighted by multiplicity, is nonlinear in A; however, we can use a well-known reduction of SDE to computing normalized traces $\mathrm{tr}(A^p)/n$ of matrix powers, which correspond to the pth moments. In particular, Kong and Valiant (2017) show that if we have ϵ'-accurate estimates $\hat{m}_1, \dots, \hat{m}_K$ of the first $K = \Omega(1/\epsilon)$ moments, i.e. $|\hat{m}_p - \mathrm{tr}(A^p)/n| \le \epsilon'\ \forall\, p \in [K]$, then if $\epsilon' = \exp(-\Omega(1/\epsilon))$ we can use them to construct a linear program (23) with $\mathcal{O}(1/\varepsilon' + K)$ variables and $\mathcal{O}(K)$ constraints returning a distribution that is ϵ-close in Wasserstein-1 distance to the true spectral density. See Theorem 8 in Appendix B for details.

With the problem reduced to estimating traces of multiple matrix powers, for the sequential case it is natural to take an approach similar to the one in Section 3.1, adjusted for setting-specific assumptions. In particular, while we again consider a sequence of matrices $A_1, \dots, A_m$ with $\|A_i - A_{i-1}\|_F \le \alpha_i$, we now assume that they are symmetric and have spectral radius at most one, i.e. that their eigenvalues are in $[-1, 1]$. This implies that $\|A_i^p\|_F^2 = \sum_{j=1}^n |\lambda_j(A_i)|^{2p} \le n$ for every $p \ge 1$, so the rescaled powers $B_{i,p} = A_i^p/\sqrt{n}$ satisfy the requirement $\|B_{i,p}\|_F \le 1$ of Corollary 3. Furthermore, it suffices to release $\epsilon'\sqrt{n}$-accurate estimates $\tilde{L}_{i,p}$ of $\mathrm{tr}(B_{i,p})$ to guarantee that the approximate moments $\hat{m}_{i,p} = \tilde{L}_{i,p}/\sqrt{n}$ satisfy the required accuracy:

$$\left|\hat{m}_{i,p} - \frac{\mathrm{tr}(A_i^p)}{n}\right| = \frac{|\tilde{L}_{i,p} - \mathrm{tr}(B_{i,p})|}{\sqrt{n}} \le \epsilon' \tag{13}$$

Lastly, analogously to Equation 12, we can get $\|B_{i,p} - B_{i-1,p}\|_F \le \min\{2, p\alpha_i/\sqrt{n}\}$, so using the argument in Corollary 4 and then taking the maximum over $p \in [K]$ yields the following:

Corollary 5 (Dynamic SDE, unknown step sizes)**.** *Consider a sequence of symmetric matrices $A_1, \dots, A_m \in \mathbb{R}^{n\times n}$ with spectral radii at most one, $A_0 = 0$, and unknown $\alpha_i = \|A_i - A_{i-1}\|_F$. Suppose for some $K = \Theta(1/\epsilon)$, $\epsilon' = \exp(-\Theta(C/\epsilon))$, and $C = \mathcal{O}(1)$ we run Algorithm 1 as described in Theorem 2 on each of the K sequences $A_1^p/\sqrt{n}, \dots, A_m^p/\sqrt{n}$ for $p = 1, \dots, K$, using target accuracy $\epsilon'\sqrt{n}$ and failure probability δ/K. Then if at each i we reuse randomness and MVPs across all instantiations $p = 1, \dots, K$ of Algorithm 1, the total number of MVPs is bounded by*

$$\mathcal{O}\left(\left(m + \frac{\exp(C/\epsilon)}{n}\min\left\{m, \frac{1}{\epsilon\sqrt{n}}\sum_{i=1}^m \alpha_i\right\}\right)\frac{1}{\epsilon}\log\frac{m}{\epsilon\delta}\right) \tag{14}$$

and w.p. $\ge 1-\delta$ we obtain moment estimates $\hat{m}_{i,p} = \tilde{L}_{i,p}/\sqrt{n}$ such that for each i the linear program (23) induced by the moment estimates $\hat{m}_{i,1}, \dots, \hat{m}_{i,K}$ yields a distribution that is ϵ-close in Wasserstein-1 distance to the spectral distribution of A_i.

Observe that exponential dependence on $1/\epsilon$ is multiplied by the path-length $\sum_i \alpha_i$, so if the matrices vary slowly the leading-order term is much smaller than the $\tilde{\mathcal{O}}(\frac{m}{n}\exp(C/\epsilon))$ MVPs used by a non-dynamic algorithm in the same setting. We note that more efficient methods are available for *non-dynamic* SDE: Chebyshev moment matching (Braverman et al., 2022; Musco et al., 2025) and stochastic Lanczos quadrature (Chen et al., 2021) achieve Wasserstein-1 error ϵ using a number of MVPs polynomial in $1/\epsilon$. Adapting these methods to exploit small changes between successive matrices in the dynamic setting is an interesting direction for future research.

4 APPLICATION: DYNAMIC ESTIMATION OF LINEAR FUNCTIONALS

We next consider a setting where dynamic estimation has yet to be studied: approximating maps from function spaces to the real line, i.e. *functionals*. The specific applications are (1) dynamic function integration and (2) tracking solutions to Laplace's equation with changing boundary conditions. Both can be reduced to estimating expectations under a fixed distribution, so if we assume the ability to evaluate the functions in the sequence at any point then we can apply the well-known Monte Carlo estimator. This section demonstrates the utility of our framework by showing how, once this estimator is shown to be norm-concentrated, it is easy to then also show guarantees for dynamic estimation.

We assume a fixed probability distribution P on a domain Ω and work with the vector space $V = L^2(\Omega, P)$ of measurable functions on Ω equipped with the norm $\|f\|_2 = \sqrt{\mathbb{E}_{X\sim P}[f(X)^2]}$. Our goal will be to estimate the linear functional $L(f) = \mathbb{E}_{X\sim P}[f(X)]$ applied to a sequence of functions f in this space; we further assume these functions have supremum norm at most one, which also bounds their $\|\cdot\|_2$ norms by one. To estimate $L(f)$, we will use the standard Monte Carlo estimator

$$\mathrm{MC}_k(f) := \frac{1}{k}\sum_{j=1}^{k} f(X_j), \qquad X_1, \ldots, X_k \overset{\text{i.i.d.}}{\sim} P. \tag{15}$$

It is easy to see that this estimator is unbiased and that its variance is at most $\|f\|_2^2/k$. For bounded functions as in our applications, Bernstein's inequality (Bernstein, 1924; Vershynin, 2018) further implies that it is sub-exponential with parameters $\nu \le \frac{c_1\|f\|_2}{\sqrt{k}}, \beta \le \frac{c_3}{k}$ for absolute constants $c_1, c_3 > 0$. Thus, $\mathrm{MC}_k(f)$ is a norm-concentrated estimator with power 1 (and $c_2 = 0$).

4.1 FUNCTION INTEGRATION

We first consider a sequence of integrands $f_1, \ldots, f_m$, where the goal is to release estimates of their integrals over a fixed domain. We can run Algorithm 1 using MC_k to estimate the damped differences $f_i - (1-\gamma_i)f_{i-1}$. Each sample evaluates the two consecutive functions at the same point, so a call with budget k costs at most $2k$ function evaluations. The preceding concentration bounds and Theorem 1 therefore give the following bound on the dynamic sample complexity. Compared with estimating each integral independently, it replaces the leading $\frac{m}{\epsilon^2}$ by $\frac{1}{\epsilon^2}\sum_{i=1}^{m}\alpha_i$. Notably, when $\sum_{i=1}^{m}\alpha_i = \mathcal{O}(m\epsilon)$, the cost is $\mathcal{O}((m/\epsilon)\log(m/\delta))$, while the baseline is $\mathcal{O}((m/\epsilon^2)\log(2m/\delta))$.

Corollary 6 (Dynamic Monte Carlo Integration, known step sizes)**.** *Suppose we have sample access to a fixed distribution P and evaluation access to a sequence $f_1, \ldots, f_m \in V$ with $f_0 = 0$, $\|f_i\|_\infty \le 1$, and known $\alpha_i = \|f_i - f_{i-1}\|_2$. For $\epsilon, \delta \in (0,1)$, run Algorithm 1 with $\mathcal{E}(f,k) = \mathrm{MC}_k(f)$. Then w.p. $\ge 1-\delta$ the resulting estimates $\tilde{L}_i$ of the integrals of f_i satisfy $|\tilde{L}_i - \mathbb{E}_{X\sim P}[f_i(X)]| \le \epsilon$ simultaneously for all $i \in [m]$, and the total number of samples and function evaluations is bounded by $\mathcal{O}\left(\left(\frac{1}{\epsilon^2}\sum_{i=1}^{m}\alpha_i + \frac{m}{\epsilon}\right)\log\frac{m}{\delta}\right)$.*

4.2 DIRICHLET EVALUATION

We next use dynamic Monte Carlo integration to track solutions to a classical boundary-value PDE known as the Dirichlet problem. Given a continuous boundary function $g : \partial\Omega \to \mathbb{R}$ on the boundary of a fixed bounded domain $\Omega \subset \mathbb{R}^d$ satisfying a standard boundary regularity condition, namely the exterior cone condition at every boundary point, the goal is to find its unique harmonic extension $u \in C(\overline{\Omega}) \cap C^2(\Omega)$, i.e. the solution to

$$\Delta u = 0 \text{ in } \Omega \qquad \text{and} \qquad u = g \text{ on } \partial\Omega, \tag{16}$$

where $\Delta = \sum_i \partial_{ii}^2$ is the Laplacian operator. For a fixed interior point $x \in \Omega$, Kakutani's representation (Kakutani, 1944; Mörters and Peres, 2010) expresses the solution as

$$u(x) = \int_{\partial\Omega} g(y) P_x(dy) = \mathbb{E}_{Y\sim P_x}[g(Y)], \tag{17}$$

where the *harmonic measure* P_x over $\partial\Omega$ is the distribution of the first boundary point reached by Brownian motion started at x. Crucially, this distribution depends only on the domain and the evaluation point, not on the boundary function. Thus, as the boundary condition evolves through $g_1, \ldots, g_m$, tracking the corresponding solutions $u_1(x), \ldots, u_m(x)$ is exactly the preceding integration problem with $P = P_x$ and $f_i = g_i$. Applying Corollary 6 gives:

Corollary 7 (Dynamic Dirichlet Evaluation, known step sizes)**.** *Fix Ω and x as above, and let $g_1, \ldots, g_m$ be continuous boundary functions with $g_0 = 0$, $\|g_i\|_\infty \le 1$, and known $\alpha_i = \|g_i - g_{i-1}\|_2$. Suppose we have sample access to P_x and evaluation access to these functions. For $\epsilon, \delta \in (0,1)$, run Algorithm 1 as in Corollary 6, with $P = P_x$ and $f_i = g_i$. Then w.p. $\ge 1 - \delta$ the resulting estimates $\tilde{u}_i(x) = \tilde{L}_i$ satisfy $|\tilde{u}_i(x) - u_i(x)| \le \epsilon$ simultaneously for all $i \in [m]$, and the total number of boundary samples and boundary-function evaluations is bounded by $\mathcal{O}\left(\left(\frac{1}{\epsilon^2}\sum_{i=1}^m \alpha_i + \frac{m}{\epsilon}\right)\log\frac{m}{\delta}\right)$.*

While mechanically very similar to the previous result, the Dirichlet evaluation result illustrates the wide-ranging applicability of the dynamic estimation problem. Note also that, in practice, we can use the walk-on-spheres (WoS) algorithm (Muller, 1956) to approximately sample boundary points from P_x and then evaluate g at those points. While they can be expensive on complex geometries, WoS boundary outputs can often be reused across boundary functions, so the reduction in evaluations due to our method would be mainly useful when the boundary function itself is expensive to compute, e.g. when each evaluation requires neural network inference or solving a different PDE.

5 Experiments

To validate our improved bounds, we evaluate our adaptive algorithm against the parameter-free DeltaShift baseline of Dharangutte and Musco (2021). Figure 1 exhibits our improved path-length bounds, dramatically reducing the number of MVPs needed to estimate the traces of neural network Hessians during training with restarts (Loshchilov and Hutter, 2017). See Appendix C for full details.

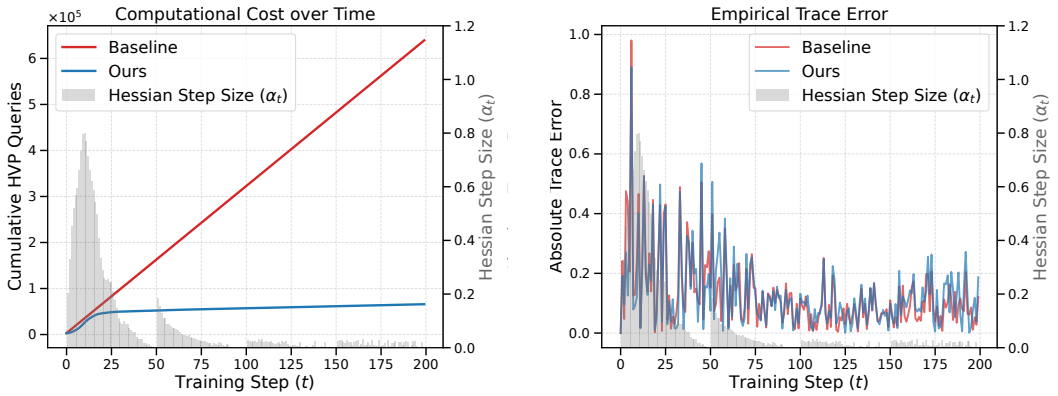


Figure 1: **Hessian Trace Estimation during SGDR.** (Left) Cumulative queries over time, with the shaded background showing the Hessian step size (α_i), which spikes heavily every 50 steps due to SGDR restarts. (Right) Empirical trace error, showing that both algorithms maintain roughly the same error, even though our adaptive approach requires far fewer Hessian-vector product queries.

6 Conclusion

Inspired by prior work, we introduced a general framework for dynamic estimation of slowly-varying sequences in a normed space, identifying *norm-concentration* as a key property that yields sequential estimation results across a wide range of old and new estimation problems for both linear and nonlinear quantities, including traces of matrix powers, spectral densities, function integrals, and solutions to certain boundary value PDEs. Our algorithm dynamically rescales the query budget to account for the local change α_i, obtaining sharper path-length-style bounds that replace the $m \cdot \max_i \alpha_i$ dependence with $\sum_{i=1}^m \alpha_i$ in the aforementioned applications. Given an auto-concentrated estimator of the squared norm, our framework can also estimate the local changes on-the-fly. Experiments on synthetic matrix sequences and Hessians from neural network optimization trajectories demonstrate improved accuracy–cost trade-offs over prior baselines for sequences that are slowly varying. Promising directions for future work include further empirical investigations, e.g. tracking evaluation metrics during pretraining via dynamic function integration, and theoretical extensions to high-dimensional or biased estimators: the latter could enable dynamic estimation results for other problems with existing estimators exhibiting norm-concentration-like properties, such as quantum shadow tomography (Huang et al., 2020) and differentially private covariance estimation (Amin et al., 2023).

7 AI Disclosure Statement

We used large language models to polish the writing, simplify some proofs, assist with conducting experiments, generate experimental code, and support literature review and search.

References

Kareem Amin, Travis Dick, Mikhail Khodak, and Sergei Vassilvitskii. Private algorithms with private predictions. Theory and Practice of Differential Privacy (TPDP) workshop manuscript, 2023. URL https://tpdp.journalprivacyconfidentiality.org/2023/papers/main-6.pdf.

Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, 32(1):48–77, 2002.

Haim Avron. Counting triangles in large graphs using randomized matrix trace estimation. In *Proceedings of the Workshop on Large-Scale Data Mining: Theory and Applications (KDD-LDMTA)*, 2010.

Bernhard Beckermann, Daniel Kressner, and Marcel Schweitzer. Low-rank updates of matrix functions. *SIAM Journal on Matrix Analysis and Applications*, 39(1):539–565, 2018. doi: 10.1137/17M1140108. URL https://doi.org/10.1137/17M1140108.

Sergei N Bernstein. On a modification of Chebyshev's inequality and of the error formula of Laplace. *Ann. Sci. Inst. Sav. Ukraine, Sect. Math*, 1(4):38–49, 1924.

Vladimir Braverman, Aditya Krishnan, and Christopher Musco. Sublinear time spectral density estimation. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1144–1157. Association for Computing Machinery, 2022. doi: 10.1145/3519935.3520009. URL https://arxiv.org/abs/2104.03461.

Hau Chan, Leman Akoglu, and Hanghang Tong. Make it or break it: Manipulating robustness in large networks. In *Proceedings of the 2014 SIAM International Conference on Data Mining (SDM)*, pages 325–333, 2014. doi: 10.1137/1.9781611973440.37. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611973440.37.

Chen Chen, Ruiyue Peng, Lei Ying, and Hanghang Tong. Network connectivity optimization: Fundamental limits and effective algorithms. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, pages 1167–1176, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220019. URL https://doi.org/10.1145/3219819.3220019.

Tyler Chen, Thomas Trogdon, and Shashanka Ubaru. Analysis of stochastic Lanczos quadrature for spectrum approximation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 1728–1739. PMLR, 2021. URL https://proceedings.mlr.press/v139/chen21s.html.

David Cohen-Steiner, Weihao Kong, Christian Sohler, and Gregory Valiant. Approximating the spectrum of a graph. *arXiv preprint arXiv:1712.01725*, 2017.

Prathamesh Dharangutte and Christopher Musco. Dynamic trace estimation. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 30088–30099. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/fcdf698a5d673435e0a5a6f9ffea05ca-Paper.pdf.

Kun Dong, David Eriksson, Hannes Nickisch, David Bindel, and Andrew Gordon Wilson. Scalable log determinants for Gaussian process kernel learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NeurIPS '17, pages 6330–6340, Red Hook, NY, USA, 2017. Curran Associates, Inc. ISBN 9781510860964.

Ernesto Estrada and Naomichi Hatano. Communicability in complex networks. *Phys. Rev. E*, 77: 036111, Mar 2008. doi: 10.1103/PhysRevE.77.036111. URL https://link.aps.org/doi/10.1103/PhysRevE.77.036111.

Cong Fang, Chris Junchi Li, Zhouchen Lin, and Tong Zhang. SPIDER: Near-optimal non-convex optimization via stochastic path-integrated differential estimator. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://papers.nips.cc/paper_files/paper/2018/hash/1543843a4723ed2ab08e18053ae6dc5b-Abstract.html.

Jack Fitzsimons, Diego Granziol, Kurt Cutajar, Michael Osborne, Maurizio Filippone, and Stephen Roberts. Entropic trace estimates for log determinants. In Michelangelo Ceci, Jaakko Hollmén, Ljupčo Todorovski, Celine Vens, and Sašo Džeroski, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 323–338. Springer International Publishing, 2017. ISBN 978-3-319-71249-9.

Gereon Frahling, Piotr Indyk, and Christian Sohler. Sampling in dynamic data streams and applications. In *Proceedings of the Twenty-First Annual Symposium on Computational Geometry*, SCG '05, pages 142–149, New York, NY, USA, 2005. Association for Computing Machinery. ISBN 1581139918. doi: 10.1145/1064092.1064116. URL https://doi.org/10.1145/1064092.1064116.

Behrooz Ghorbani, Shankar Krishnan, and Ying Xiao. An investigation into neural net optimization via Hessian eigenvalue density. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2232–2241. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/ghorbani19b.html.

Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16:1050–1057, 2020. doi: 10.1038/s41567-020-0932-7.

M. F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Communications in Statistics—Simulation and Computation*, 18(3):1059–1076, 1989. doi: 10.1080/03610918908812806.

T. S. Jayram, Andrew McGregor, S. Muthukrishnan, and Erik Vee. Estimating statistical aggregates on probabilistic data streams. *ACM Trans. Database Syst.*, 33(4), December 2008. ISSN 0362-5915. doi: 10.1145/1412331.1412338. URL https://doi.org/10.1145/1412331.1412338.

Shizuo Kakutani. Two-dimensional Brownian motion and harmonic functions. *Proceedings of the Imperial Academy*, 20(10):706–714, 1944. doi: 10.3792/pia/1195572706. URL https://doi.org/10.3792/pia/1195572706.

Weihao Kong and Gregory Valiant. Spectrum estimation from samples. *The Annals of Statistics*, 45 (5):2218–2247, 2017. doi: 10.1214/16-AOS1525.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NeurIPS '18, pages 6391–6401, Red Hook, NY, USA, 2018. Curran Associates, Inc.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Alessio Mazzetto. An improved algorithm for learning drifting discrete distributions. In *Proceedings of the 27th International Conference on Artificial Intelligence and Statistics*, 2024.

Alessio Mazzetto and Eli Upfal. Nonparametric density estimation under distribution drift. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Peter Mörters and Yuval Peres. *Brownian Motion*. Cambridge University Press, 2010. doi: 10.1017/CBO9780511750489.

Mervin E. Muller. Some continuous Monte Carlo methods for the Dirichlet problem. *The Annals of Mathematical Statistics*, 27(3):569–589, 1956. doi: 10.1214/aoms/1177728169. URL https://doi.org/10.1214/aoms/1177728169.

Cameron Musco, Christopher Musco, Lucas Rosenblatt, and Apoorv Vikram Singh. Sharper bounds for Chebyshev moment matching, with applications. In *Proceedings of Thirty Eighth Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pages 4309–4358. PMLR, 2025. URL https://proceedings.mlr.press/v291/musco25a.html.

Lam M. Nguyen, Jie Liu, Katya Scheinberg, and Martin Takáč. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2613–2621. PMLR, 2017. URL https://proceedings.mlr.press/v70/nguyen17b.html.

Barak A. Pearlmutter. Fast exact multiplication by the Hessian. *Neural Computation*, 6(1):147–160, 1994. doi: 10.1162/neco.1994.6.1.147.

Mark Rudelson and Roman Vershynin. Hanson-Wright inequality and sub-Gaussian concentration. *Electronic Communications in Probability*, 18:1–9, 2013.

Mark Schmidt and Nicolas Le Roux. Fast convergence of stochastic gradient descent under a strong growth condition. *arXiv preprint arXiv:1308.6370*, 2013.

Sharan Vaswani, Francis Bach, and Mark Schmidt. Fast and faster convergence of SGD for over-parameterized models and an accelerated perceptron. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*, volume 89 of *Proceedings of Machine Learning Research*, pages 1195–1204. PMLR, 2019.

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

Sheng Wang, Yuan Sun, Christopher Musco, and Zhifeng Bao. Public transport planning: When transit network connectivity meets commuting demand. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, pages 1906–1919, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383431. doi: 10.1145/3448016.3457247. URL https://doi.org/10.1145/3448016.3457247.

David Woodruff, Fred Zhang, and Richard Zhang. Optimal query complexities for dynamic trace estimation. *Advances in Neural Information Processing Systems*, 35:35049–35060, 2022.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W. Mahoney. PyHessian: Neural networks through the lens of the Hessian. In *Proceedings of the 2020 IEEE International Conference on Big Data (Big Data)*, pages 581–590. IEEE, 2020. doi: 10.1109/BigData50022.2020.9378171.

Yunyue Zhu and Dennis Shasha. StatStream: Statistical monitoring of thousands of data streams in real time. In Philip A. Bernstein, Yannis E. Ioannidis, Raghu Ramakrishnan, and Dimitris Papadias, editors, *VLDB '02: Proceedings of the 28th International Conference on Very Large Databases*, chapter 32, pages 358–369. Morgan Kaufmann, San Francisco, 2002. ISBN 978-1-55860-869-6. doi: 10.1016/B978-155860869-6/50039-1. URL https://www.sciencedirect.com/science/article/pii/B9781558608696500391.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning*, 2003.

A Preliminaries on sub-exponential random variables

Definition 3 (Sub-Exponential Random Variable). *A scalar random variable X with mean μ is sub-exponential with variance parameter $\nu^2 \geq 0$ and scale parameter $\beta \geq 0$ if*

$$\mathbb{E}\left[e^{\lambda(X-\mu)}\right] \leq e^{\lambda^2\nu^2/2} \quad \textit{for all } |\lambda| < 1/\beta. \tag{18}$$

For $\beta = 0$, the condition is required for every $\lambda \in \mathbb{R}$. Throughout, $1/0$ is interpreted as $+\infty$.

We use the following two standard facts about sub-exponential random variables.

Lemma 1 (Sub-Exponential Concentration). *Let X be a sub-exponential random variable with mean μ, variance parameter ν^2, and scale parameter β. Then, for any $\epsilon > 0$,*

$$\mathbb{P}[|X - \mu| \geq \epsilon] \leq 2\exp\left(-\min\left\{\frac{\epsilon^2}{2\nu^2}, \frac{\epsilon}{2\beta}\right\}\right). \tag{19}$$

Consequently, it is sufficient for failure probability at most δ that $\nu^2 \leq \epsilon^2/(2\log(2/\delta))$ and $\beta \leq \epsilon/(2\log(2/\delta))$.

Lemma 2 (Linear Combinations of Sub-Exponentials). *Let $X_1, \ldots, X_k$ be independent sub-exponential random variables with respective parameters (ν_i^2, β_i). For $a_1, \ldots, a_k \in \mathbb{R}$, the sum $Y = \sum_{i=1}^k a_i X_i$ is sub-exponential with parameters*

$$\nu_*^2 = \sum_{i=1}^k a_i^2 \nu_i^2 \qquad \textit{and} \qquad \beta_* = \max_i |a_i|\beta_i. \tag{20}$$

B Spectral reconstruction from approximate moments

We now explain how to obtain an approximate spectral distribution of a symmetric matrix $A \in \mathbb{R}^{n\times n}$ from estimates of normalized traces $\mathrm{tr}(A^p)/n$ of its powers $p \geq 1$. This relies on the observation that these normalized traces are the pth moments of the spectral distribution, so we can approximate it by seeking a probability distribution whose moments are close to these estimates. Note that the results in this section are not novel, but rather they summarize existing literature (Kong and Valiant, 2017; Cohen-Steiner et al., 2017) in our notation. Throughout this section, we use $W_1(\mu, \nu)$ to denote the 1-Wasserstein distance, which is the minimum expected distance $\mathbb{E}[|X - Y|]$ over all joint distributions of (X, Y) with marginals μ and ν. This distance is widely used to measure error in SDE (Braverman et al., 2022; Chen et al., 2021; Musco et al., 2025).

To see why this suffices, consider two distributions on $[-1, 1]$ that have the same first K moments. For any degree-K polynomial $P(x) = a_0 + \sum_{p=1}^K a_p x^p$, its expectation under either distribution is determined by these moments. Thus, if two distributions agree in their first K moments, they also agree on the expectation of every polynomial of degree at most K.

To relate this observation to the Wasserstein distance W_1, recall its equivalent definition as the supremum, over all 1-Lipschitz functions f, of the absolute difference between the expectations of f under the two distributions. It is also known from Jackson's approximation theorem that for every such function f on $[-1, 1]$ there is a polynomial P_f of degree at most K satisfying

$$\sup_{x\in[-1,1]} |f(x) - P_f(x)| \leq \frac{C}{K}, \tag{21}$$

where C is an absolute constant; see Kong and Valiant (2017, Section 4.1). Replacing f by P_f therefore changes its expectation under either distribution by at most C/K. Since the expectations of P_f agree, the expectations of f differ by at most $2C/K$. This holds for every 1-Lipschitz function, so the two distributions are $\mathcal{O}(1/K)$ apart in W_1.

In our setting, the moments are only approximately known. To handle this, Kong and Valiant (2017, Proposition 1) extend the above argument to bound the W_1 distance in terms of both K and the discrepancies between the moments. It therefore suffices to find a distribution whose moments fit the estimates closely enough. The reconstruction procedure of Kong and Valiant (2017, Algorithm 2, Eq. (2)), also used by Cohen-Steiner et al. (2017, Section 3.3), does this by choosing probability

weights on a fine grid to minimize the total absolute difference between the resulting moments and their estimates.

Given moment estimates $\widehat{m}_1, \ldots, \widehat{m}_K$ and a grid parameter $\epsilon' \in (0, 1)$, set

$$N = \left\lceil \frac{2}{\epsilon'} \right\rceil, \qquad x_j = -1 + \frac{2j}{N}, \qquad j = 0, \ldots, N. \tag{22}$$

Introducing variables t_p for the absolute moment discrepancies, we solve the linear program

$$\begin{aligned} \min_{w \geq 0,\, t \geq 0} \quad & \sum_{p=1}^{K} t_p \\ \text{subject to} \quad & \sum_{j=0}^{N} w_j = 1, \\ & -t_p \leq \sum_{j=0}^{N} w_j x_j^p - \widehat{m}_p \leq t_p, \qquad p \in [K]. \end{aligned} \tag{23}$$

For an optimal solution (w^*, t^*), let $\widehat{\mu}$ be the distribution assigning probability w_j^* to x_j. Combining this reconstruction with Kong and Valiant (2017, Proposition 1) gives the following consequence, stated here in our notation for completeness; see also (Cohen-Steiner et al., 2017, Lemma 6).

Theorem 8 (SDE from approximate traces; Kong and Valiant, 2017)**.** *There exist universal constants $C_1, C_2 > 0$ such that the following holds. Let $A \in \mathbb{R}^{n \times n}$ be a symmetric matrix with all eigenvalues in $[-1, 1]$, and let μ be its spectral distribution. For any $\epsilon \in (0, 1)$, set*

$$K = \left\lceil \frac{C_1}{\epsilon} \right\rceil, \qquad \epsilon' = \exp(-C_2/\epsilon). \tag{24}$$

Given moment estimates $\widehat{m}_1, \ldots, \widehat{m}_K$ satisfying

$$\left| \widehat{m}_p - \frac{\operatorname{tr}(A^p)}{n} \right| \leq \epsilon', \qquad p \in [K], \tag{25}$$

solving the linear program (23) *with grid parameter ϵ' gives a probability distribution $\widehat{\mu}$ on $[-1, 1]$ satisfying $W_1(\widehat{\mu}, \mu) \leq \epsilon$.*

Applying this reconstruction at each time step gives the spectral estimates used in Section 3.2, without any additional MVPs.

C Experiments

Cost across a sequence and ablations. In our first experiment, we generate a synthetic sequence of $2{,}000 \times 2{,}000$ matrices over $m = 200$ time steps. We introduce rare, large perturbations to simulate a stable sequence punctuated by bursts. As shown in Figure 2, the baseline method chooses its fixed query budget using the worst-case step size $\max_i \alpha_i$. Our adaptive algorithm instead scales down its query budget during stable periods, resulting in a lower total MVP cost. Figure 4 shows that this reduction persists across a wide range of error tolerances ϵ and failure probabilities δ.

Cost vs. Empirical Error Pareto Frontier. In our second experiment, we evaluate the trade-off between total computational cost (MVPs) and empirical error. To generate a single point on the trade-off graph, we first run our adaptive method for a specific target error tolerance and record its total budget. We then divide that budget evenly across all time steps for the non-adaptive baseline. Repeating this procedure across error tolerances yields the Pareto plots in Figure 3, which report maximum absolute error, mean absolute error, and a weighted mean error that emphasizes large matrix changes. Our method performs better on all three metrics.

Hessian Trace Tracking During Neural Network Training. We evaluate dynamic Hessian trace estimation on a neural network training trajectory. We train a $2{,}410$-parameter multilayer perceptron for 200 steps using an SGDR (stochastic gradient descent with warm restarts) learning-rate schedule (Loshchilov and Hutter, 2017) with period $T_0 = 50$. We compute Hessian-vector products by backpropagation using Pearlmutter's trick (Pearlmutter, 1994). Figure 1 compares cumulative queries and trace error.

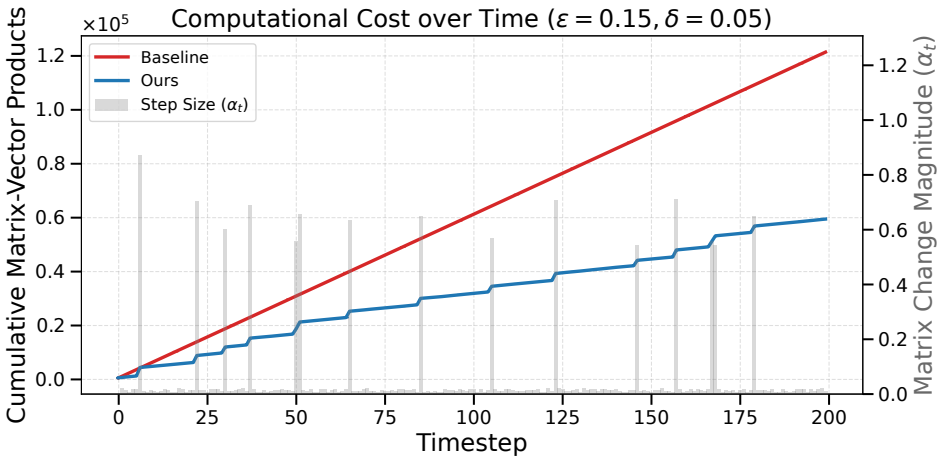


Figure 2: Our adaptive method requires fewer MVPs to estimate traces of a sequence of matrices.

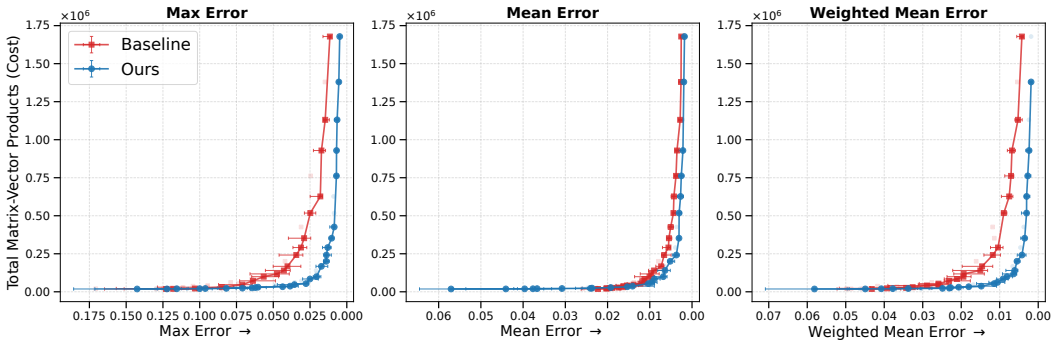


Figure 3: **Cost vs. Empirical Error Pareto Frontier.** At equivalent computational budgets, the adaptive algorithm achieves lower errors across all three metrics in these experiments.

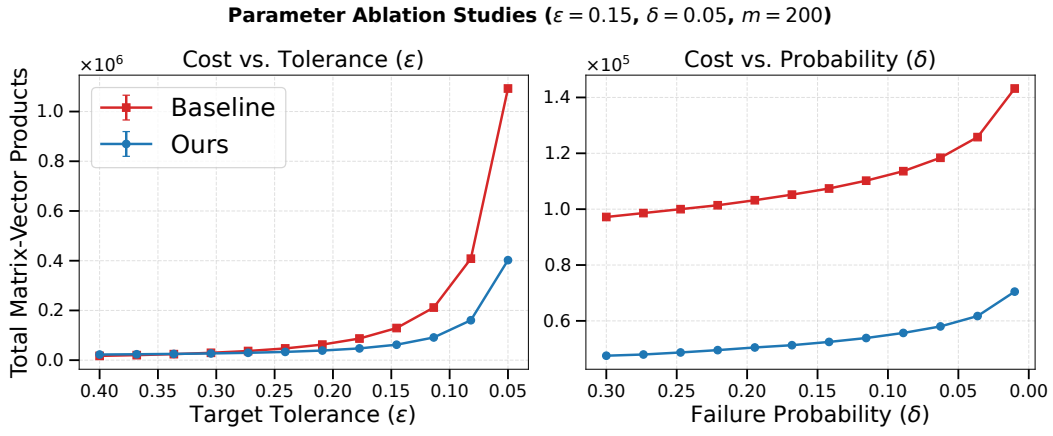


Figure 4: **Ablations.** Our method uses fewer queries than the baseline across the tested target tolerances ϵ and failure probabilities δ.