XTRACE: MAKING THE MOST OF EVERY SAMPLE IN STOCHASTIC TRACE ESTIMATION*

ETHAN N. EPPERLY†, JOEL A. TROPP†, AND ROBERT J. WEBBER†

Abstract. The implicit trace estimation problem asks for an approximation of the trace of a square matrix, accessed via matrix–vector products (matvecs). This paper designs new randomized algorithms, XTRACE and XNYSTRACE, for the trace estimation problem by exploiting both variance reduction and the exchangeability principle. For a fixed budget of matvecs, numerical experiments show that the new methods can achieve errors that are orders of magnitude smaller than existing algorithms, such as the Girard–Hutchinson estimator or the HUTCH++ estimator. A theoretical analysis confirms the benefits by offering a precise description of the performance of these algorithms as a function of the spectrum of the input matrix. The paper also develops an exchangeable estimator, XDIAG, for approximating the diagonal of a square matrix using matvecs.

Key words. Trace estimation, low-rank approximation, exchangeability, variance reduction, randomized algorithm.

AMS subject classifications. 65C05, 65F30, 68W20

1. Introduction.

Over the past three decades, researchers have developed *randomized algorithms* for linear algebra problems such as trace estimation [14, 19, 26], low-rank approximation [15], and over-determined least squares [3, 30]. Many of these algorithms collect information by judicious random sampling of the problem data. As a consequence, we can design better algorithms using techniques from the theory of statistical estimation, such as variance reduction and the exchangeability principle. This paper explores how the exchangeability principle leads to faster randomized algorithms for trace estimation.

Suppose that we wish to compute a quantity $Q(\boldsymbol{A})$ associated with a matrix $\boldsymbol{A}$. A typical randomized algorithm might proceed as follows.

1. **Collect information** about the matrix $\boldsymbol{A}$ by computing matrix–vector products $\boldsymbol{A\omega}_1, \dots, \boldsymbol{A\omega}_k$ with random test vectors $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_k$.
2. **Form an estimate** of $Q(\boldsymbol{A})$ from the samples $\boldsymbol{A\omega}_1, \dots, \boldsymbol{A\omega}_k$.

The question arises: *Given the data $\boldsymbol{A\omega}_1, \dots, \boldsymbol{A\omega}_k$, what is an optimal estimator for $Q(\boldsymbol{A})$?* One property an optimal estimator must obey is the exchangeability principle:

> **Exchangeability principle:** If the test vectors $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_k$ are exchangeable, the minimum-variance unbiased estimator for $Q(\boldsymbol{A})$ is always a symmetric function of $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_k$.

*Date: 18 January 2023.

Funding: ENE acknowledges support from the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Department of Energy Computational Science Graduate Fellowship under Award Number DE-SC0021110. JAT and RJW acknowledge support from the Office of Naval Research through BRC Award N00014-18-1-2363 and from the National Science Foundation through FRG Award 1952777.

†Division of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125 USA (eepperly@caltech.edu, jtropp@caltech.edu, rwebber@caltech.edu).

"Exchangeability" means that the family $(\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_k)$ has the same distribution as the permuted family $(\boldsymbol{\omega}_{\sigma(1)}, \dots, \boldsymbol{\omega}_{\sigma(k)})$ for every permutation σ in the symmetric group S_k. In particular, an independent and identically distributed (iid) family is exchangeable.

The implication of the exchangeability principle is that our estimators should be symmetric functions of the samples, whenever possible. This idea is attributed to Halmos [16], and it plays a central role in the theory of U-statistics [21].

This paper will demonstrate that the exchangeability principle can lead to new randomized algorithms for linear algebra problems. As a case study, we will explore the problem of *implicit trace estimation*:

> **Implicit trace estimation problem:** Given access to a square matrix $\boldsymbol{A}$ via the matrix–vector product (matvec) operation $\boldsymbol{\omega} \mapsto \boldsymbol{A}\boldsymbol{\omega}$, estimate the trace of $\boldsymbol{A}$.

Trace estimation plays a role in a wide range of areas, including computational statistics, statistical mechanics, and network analysis. See the survey [35] for more applications.

As we will see, it is natural to design randomized algorithms for trace estimation that use matvecs between the input matrix and random test vectors. At present, the state-of-the-art trace estimators do not satisfy the exchangeability principle. By pursuing this insight, we will develop better trace estimators. Given a fixed budget of matvecs, the new algorithms can reduce the variance of the trace estimate by several orders of magnitude. This case study highlights the importance of enforcing exchangeability in design of randomized algorithms.

1.1. Stochastic trace estimators. In this section, we outline the classic approach to randomized trace estimation based on Monte Carlo approximation. Then we introduce a more modern approach that incorporates a variance reduction strategy.

1.1.1. The Girard–Hutchinson estimator. The first randomized algorithm for trace estimation was proposed by Girard [14] and extended by Hutchinson [19].

Let $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ be a square input matrix. Consider an isotropic random vector $\boldsymbol{\omega} \in \mathbb{R}^N$:

$$\mathbb{E}[\boldsymbol{\omega}\boldsymbol{\omega}^*] = \mathbf{I}. \tag{1.1}$$

For example, we may take a random sign vector $\boldsymbol{\omega} \sim \textsc{uniform}\{\pm 1\}^N$. By isotropy,

$$\mathbb{E}[\boldsymbol{\omega}^* (\boldsymbol{A}\boldsymbol{\omega})] = \operatorname{tr} \boldsymbol{A}.$$

The symbol * denotes the transpose. This relation suggests a Monte Carlo method.

Accordingly, the Girard–Hutchinson trace estimator takes the form

$$\widehat{\operatorname{tr}}_{\mathrm{GH}} := \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{\omega}_i^* (\boldsymbol{A}\boldsymbol{\omega}_i) \quad \text{where the } \boldsymbol{\omega}_i \text{ are iid copies of } \boldsymbol{\omega}. \tag{1.2}$$

This estimator is exchangeable, and it is unbiased: $\mathbb{E}\left[\widehat{\operatorname{tr}}_{\mathrm{GH}}\right] = \operatorname{tr} \boldsymbol{A}$. We can measure the quality of the estimator using the variance, $\operatorname{Var}\left[\widehat{\operatorname{tr}}_{\mathrm{GH}}\right]$. The variance depends on the matrix $\boldsymbol{A}$ and the distribution of $\boldsymbol{\omega}$, but it converges to zero at the Monte Carlo rate $\Theta(m^{-1})$ as we increase the number m of samples. See the survey [25, §4] for more discussion.

1.1.2. The Hutch++ estimator. To improve on the Girard–Hutchinson estimator, several papers [12, 23, 26, 31] have advocated variance reduction techniques. The key idea is to form a low-rank approximation of the input matrix. We can compute the trace of the approximation exactly (as a control variate), so we only need to estimate the trace of the residual. This approach can attain lower variance than the Monte Carlo method.

Algorithm 1.1 HUTCH++ [26]

Input: Matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and number m of matvecs, where m is divisible by 3
Output: Trace estimate $\widehat{\mathrm{tr}} \approx \operatorname{tr} \boldsymbol{A}$

Draw iid isotropic $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_{2m/3} \in \mathbb{R}^N$ ▷ For example, $\boldsymbol{\omega}_i \sim \text{UNIFORM}\{\pm 1\}^N$
$\boldsymbol{Y} \leftarrow \boldsymbol{A}\begin{bmatrix}\boldsymbol{\omega}_{m/3+1} & \cdots & \boldsymbol{\omega}_{2m/3}\end{bmatrix}$ ▷ Use matvecs
$\boldsymbol{Q} \leftarrow \operatorname{orth}(\boldsymbol{Y})$
$\boldsymbol{G} \leftarrow \begin{bmatrix}\boldsymbol{\omega}_1 & \cdots & \boldsymbol{\omega}_{m/3}\end{bmatrix} - \boldsymbol{Q}\boldsymbol{Q}^*\begin{bmatrix}\boldsymbol{\omega}_1 & \cdots & \boldsymbol{\omega}_{m/3}\end{bmatrix}$
$\widehat{\mathrm{tr}} \leftarrow \operatorname{tr}\big(\boldsymbol{Q}^*(\boldsymbol{A}\boldsymbol{Q})\big) + (m/3)^{-1}\operatorname{tr}(\boldsymbol{G}^*(\boldsymbol{A}\boldsymbol{G}))$ ▷ Use matvecs

The HUTCH++ estimator of Meyer, Musco, Musco, and Woodruff [26] crystallizes the variance reduction strategy. Let $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ be a square input matrix. Given a fixed budget of m matvecs, with m divisible by 3, HUTCH++ proceeds as follows:

1. *Sample* iid isotropic vectors $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_{2m/3} \in \mathbb{R}^N$ as in (1.1).
2. *Sketch* $\boldsymbol{Y} = \boldsymbol{A}\begin{bmatrix}\boldsymbol{\omega}_{m/3+1} & \boldsymbol{\omega}_{m/3+2} & \cdots & \boldsymbol{\omega}_{2m/3}\end{bmatrix}$
3. *Orthonormalize* $\boldsymbol{Q} = \operatorname{orth}(\boldsymbol{Y})$.
4. *Output* the estimate

$$\widehat{\mathrm{tr}}_{\mathrm{H++}} := \operatorname{tr}(\boldsymbol{Q}^*(\boldsymbol{A}\boldsymbol{Q})) + \frac{1}{m/3}\sum_{i=1}^{m/3} \boldsymbol{\omega}_i^*(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^*)\big(\boldsymbol{A}(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^*)\boldsymbol{\omega}_i\big). \tag{1.3}$$

See Algorithm 1.1 for efficient HUTCH++ pseudocode.

To illustrate how HUTCH++ takes advantage of low-rank approximation, we first observe that $\widehat{\boldsymbol{A}} = \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}$ is a low-rank approximation of the matrix $\boldsymbol{A}$. Indeed, the matrix $\widehat{\boldsymbol{A}}$ coincides with the randomized SVD [15] formed from the test matrix $\begin{bmatrix}\boldsymbol{\omega}_{m/3+1} & \cdots & \boldsymbol{\omega}_{2m/3}\end{bmatrix}$. HUTCH++ computes the trace of the low-rank approximation, which is

$$\operatorname{tr}\widehat{\boldsymbol{A}} = \operatorname{tr}\big(\boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}\big) = \operatorname{tr}\big(\boldsymbol{Q}^*\boldsymbol{A}\boldsymbol{Q}\big).$$

Afterward, HUTCH++ applies the Girard–Hutchinson estimator to estimate the trace of the residual

$$\operatorname{tr}\big(\boldsymbol{A} - \widehat{\boldsymbol{A}}\big) = \operatorname{tr}\big((\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^*)\boldsymbol{A}\big) = \operatorname{tr}\big((\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^*)\boldsymbol{A}(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^*)\big).$$

Like the Girard–Hutchinson trace estimator, the HUTCH++ estimator is unbiased. In contrast to the $\Theta(m^{-1})$ variance of Girard–Hutchinson, the variance of HUTCH++ is no greater than $\mathcal{O}(m^{-2})$. In practice, the reduction in variance is conspicuous. On the other hand, the HUTCH++ estimator violates the exchangeability principle, so we recognize an opportunity to design a better algorithm.

1.2. New exchangeable trace estimators. The HUTCH++ estimator is not exchangeable because it uses some test vectors to perform low-rank approximation, while it uses other test vectors to estimate the trace of the residual. Although it might seem natural to symmetrize HUTCH++ over all splits of the test vectors, this approach is both wasteful and computationally infeasible.

Instead, we will develop a new family of exchangeable trace estimators that make the most out of every test vector. We will use *all* of the test vectors for low-rank approximation, and we will use *all* of the test vectors for estimating the trace of the residual. The key to this strategy is a leave-one-out technique that can be implemented at the same computational cost as HUTCH++. This innovation can reduce the variance by several orders of magnitude.

Algorithm 1.2 XTRACE: Naïve implementation

Input: Matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and number m of matvecs, where m is even
Output: Trace estimate $\widehat{\mathrm{tr}} \approx \mathrm{tr}\,\boldsymbol{A}$ and error estimate $\widehat{\mathrm{err}} \approx |\widehat{\mathrm{tr}} - \mathrm{tr}\,\boldsymbol{A}|$

Draw iid isotropic $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_{m/2} \in \mathbb{R}^N$ ▷ See subsection 2.3
$\boldsymbol{Y} \leftarrow \boldsymbol{A}\begin{bmatrix}\boldsymbol{\omega}_1 & \cdots & \boldsymbol{\omega}_{m/2}\end{bmatrix}$ ▷ Use matvecs
for $i = 1, 2, 3, \dots, m/2$ **do**
  $\boldsymbol{Q}_{(i)} \leftarrow \mathrm{orth}(\boldsymbol{Y}_{-i})$ ▷ Remove ith column of $\boldsymbol{Y}$
  $\widehat{\mathrm{tr}}_i \leftarrow \mathrm{tr}\big(\boldsymbol{Q}_{(i)}^*(\boldsymbol{A}\boldsymbol{Q}_{(i)})\big) + \boldsymbol{\omega}_i^*\big(\mathbf{I} - \boldsymbol{Q}_{(i)}\boldsymbol{Q}_{(i)}^*\big)\big(\boldsymbol{A}\big(\mathbf{I} - \boldsymbol{Q}_{(i)}\boldsymbol{Q}_{(i)}^*\big)\big)\boldsymbol{\omega}_i$ ▷ Use matvecs
end for
$\widehat{\mathrm{tr}} \leftarrow (m/2)^{-1}\sum_{i=1}^{m/2}\widehat{\mathrm{tr}}_i$
$\widehat{\mathrm{err}}^2 \leftarrow ((m/2)(m/2-1))^{-1}\sum_{i=1}^{m/2}(\widehat{\mathrm{tr}}_i - \widehat{\mathrm{tr}})^2$

1.2.1. The XTRACE estimator. Our first method, called XTRACE, is an exchangeable trace estimator designed for general square matrices. It computes a family of variance-reduced trace estimators. Each estimator uses all but one test vector to form a low-rank approximation, and it uses the remaining test vector to estimate the trace of the residual. XTRACE then averages the basic estimators together to obtain an exchangeable trace estimator.

Let us give a more detailed description. Fix a square input matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$. The parameter m is the number of matvecs, where m is an even number. Draw an iid family $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_{m/2} \in \mathbb{R}^N$ of isotropic test vectors, and define the test matrix

$$\boldsymbol{\Omega} = \begin{bmatrix}\boldsymbol{\omega}_1 & \boldsymbol{\omega}_2 & \boldsymbol{\omega}_3 & \dots & \boldsymbol{\omega}_{m/2}\end{bmatrix}.$$

Construct the orthonormal matrices

$$\boldsymbol{Q}_{(i)} = \mathrm{orth}(\boldsymbol{A}\boldsymbol{\Omega}_{-i}) \quad \text{for each } i = 1, \dots, m/2, \tag{1.4}$$

where $\boldsymbol{\Omega}_{-i}$ is the test matrix with the ith column removed. Compute the basic trace estimators

$$\widehat{\mathrm{tr}}_i := \mathrm{tr}\big(\boldsymbol{Q}_{(i)}^*\big(\boldsymbol{A}\boldsymbol{Q}_{(i)}\big)\big) + \boldsymbol{\omega}_i^*\big(\mathbf{I} - \boldsymbol{Q}_{(i)}\boldsymbol{Q}_{(i)}^*\big)\big(\boldsymbol{A}\big(\mathbf{I} - \boldsymbol{Q}_{(i)}\boldsymbol{Q}_{(i)}^*\big)\boldsymbol{\omega}_i\big) \quad \text{for } i = 1, \dots, m/2. \tag{1.5}$$

The XTRACE estimator averages these basic estimators:

$$\widehat{\mathrm{tr}}_{\mathrm{X}} := \frac{1}{m/2}\sum_{i=1}^{m/2}\widehat{\mathrm{tr}}_i. \tag{1.6}$$

The XTRACE method gives an unbiased, exchangeable estimate for the trace. Theorem 1.1 provides a detailed prior bound for the variance. We can also obtain a posterior estimate for the error using the formula

$$\widehat{\mathrm{err}}_{\mathrm{X}}^2 := \frac{1}{(m/2)(m/2-1)}\sum_{i=1}^{m/2}(\widehat{\mathrm{tr}}_i - \widehat{\mathrm{tr}}_{\mathrm{X}})^2.$$

Subsection 3.1 contains further discussion of the error estimate.

This procedure requires exactly m matvecs with the input matrix $\boldsymbol{A}$. See Algorithm 1.2 for direct XTRACE pseudocode. With careful attention to the linear algebra, we can develop an implementation whose computational cost is comparable with a single estimator of the form (1.5); see subsection 2.1 for details.

Algorithm 1.3 XNysTrace: Naïve implementation

Input: Psd matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and number m of matvecs
Output: Trace estimate $\widehat{\mathrm{tr}} \approx \operatorname{tr} \boldsymbol{A}$ and error estimate $\widehat{\mathrm{err}} \approx |\widehat{\mathrm{tr}} - \operatorname{tr} \boldsymbol{A}|$

Draw iid isotropic $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_m \in \mathbb{R}^N$ ▷ See subsection 2.3
$\boldsymbol{\Omega} \leftarrow \begin{bmatrix} \boldsymbol{\omega}_1 & \dots & \boldsymbol{\omega}_m \end{bmatrix}$
$\boldsymbol{Y} \leftarrow \boldsymbol{A}\boldsymbol{\Omega}$ ▷ Use matvecs
for $i = 1, 2, 3, \dots, m$ **do**
  $\widehat{\boldsymbol{A}}_i \leftarrow \boldsymbol{Y}_{-i}(\boldsymbol{\Omega}_{-i}^* \boldsymbol{Y}_{-i})^\dagger \boldsymbol{Y}_{-i}^*$ ▷ Remove ith column of $\boldsymbol{Y}$ and $\boldsymbol{\Omega}$
  $\widehat{\mathrm{tr}}_i \leftarrow \operatorname{tr} \widehat{\boldsymbol{A}}_i + \boldsymbol{\omega}_i^*((\boldsymbol{A} - \widehat{\boldsymbol{A}}_i)\boldsymbol{\omega}_i)$ ▷ Use matvecs
end for
$\widehat{\mathrm{tr}} \leftarrow m^{-1} \sum_{i=1}^m \widehat{\mathrm{tr}}_i$
$\widehat{\mathrm{err}}^2 \leftarrow (m(m-1))^{-1} \sum_{i=1}^m (\widehat{\mathrm{tr}}_i - \widehat{\mathrm{tr}})^2$

1.2.2. The XNysTrace estimator. Our second method, called XNysTrace, is an exchangeable trace estimator designed for positive-semidefinite (psd) matrices. Rather than using a randomized SVD to reduce the variance, this estimator uses a Nyström approximation [25, §14] of the psd matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$. This approximation takes the form

$$\boldsymbol{A}\langle \boldsymbol{X} \rangle := \boldsymbol{A}\boldsymbol{X}(\boldsymbol{X}^*\boldsymbol{A}\boldsymbol{X})^\dagger(\boldsymbol{A}\boldsymbol{X})^* \quad \text{for a test matrix } \boldsymbol{X} \in \mathbb{R}^{N\times k}. \tag{1.7}$$

The Nyström method requires only k matvecs to compute a rank-k approximation, while the randomized SVD requires $2k$ matvecs.

Let us summarize the XNysTrace method. Draw iid isotropic test vectors $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_m$, and form the test matrix $\boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{\omega}_1 & \dots & \boldsymbol{\omega}_m \end{bmatrix}$. The basic estimators take the form

$$\widehat{\mathrm{tr}}_i := \operatorname{tr} \boldsymbol{A}\langle \boldsymbol{\Omega}_{-i} \rangle + \boldsymbol{\omega}_i^*\big((\boldsymbol{A} - \boldsymbol{A}\langle \boldsymbol{\Omega}_{-i} \rangle)\boldsymbol{\omega}_i\big) \quad \text{for } i = 1, \dots, m. \tag{1.8}$$

As usual, $\boldsymbol{\Omega}_{-i}$ denotes the test matrix with the ith column removed. To obtain the XNysTrace estimator and an error estimate, we use the formulas

$$\widehat{\mathrm{tr}}_{\mathrm{XN}} := \frac{1}{m}\sum_{i=1}^m \widehat{\mathrm{tr}}_i \quad \text{and} \quad \widehat{\mathrm{err}}_{\mathrm{XN}} := \frac{1}{m(m-1)}\sum_{i=1}^m (\widehat{\mathrm{tr}}_i - \widehat{\mathrm{tr}}_{\mathrm{XN}})^2. \tag{1.9}$$

The XNysTrace estimator is unbiased and exchangeable. Theorem 1.1 provides a bound for the variance. See Algorithm 1.3 for naïve XNysTrace pseudocode and subsection 2.2 for a more efficient approach.

The recent paper [28] describes an estimator called Nyström++ that uses a Nyström approximation to perform reduced-variance trace estimation. The Nyström++ method violates the exchangeability principle, while XNysTrace repairs this weakness.

1.2.3. Stochastic diagonal estimators. As an extension of XTrace, we also propose the XDiag algorithm for estimating the *diagonal* of an implicitly defined matrix. We will discuss this approach in subsection 2.4.

1.3. Numerical experiments. To highlight the advantages of the XTrace and XNysTrace estimators, we present some motivating numerical experiments. Section 4 contains further numerical work.

1.3.1. Exploiting spectral decay. Our first experiment uses a synthetic input matrix to illustrate how the exchangeable estimators wring more information out of the samples.

We apply several trace estimators to a psd matrix with exponentially decreasing eigenvalues; see subsection 4.1 for the details of the matrix. Figure 1 reports the average error

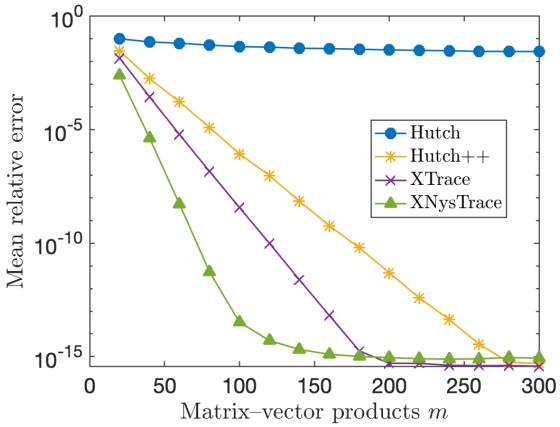


Fig. 1: **Exploiting spectral decay.** Average error of trace estimators applied to a (synthetic) psd matrix with exponentially decreasing eigenvalues. See subsection 1.3.1.

over 1000 trials. The Girard–Hutchinson estimator (HUTCH) converges at the Monte Carlo rate, whereas the newer estimators all converge much faster. This improvement comes from variance reduction techniques that exploit the spectral decay. Observe that XTRACE and XNYSTRACE converge exponentially fast at 1.5× and 3× the rate of HUTCH++, until reaching machine precision. For a fixed budget of m matvecs, XTRACE and XNYSTRACE can reduce the error by several orders of magnitude compared to HUTCH++. Strikingly, the reduction in variance from enforcing exchangeability is almost as significant as the reduction in variance from using a low-rank approximation as a control variate.

1.3.2. Computing partition functions. Our second experiment shows how the advantages of using exchangeable estimators persist in a scientific application.

We apply several trace estimators to compute the partition function for a quantum system

$$Z(\beta) := \operatorname{tr} \exp(-\beta \boldsymbol{H}),$$

where $\boldsymbol{H}$ is a symmetric Hamiltonian matrix and $\beta > 0$ is an inverse temperature. Specifically, we consider the Hamiltonian matrix $\boldsymbol{H}$ for the tranverse-field Ising model on 18 sites, which has dimension $N = 2^{18} = 262\,144$. See subsection 4.3 for details on the matrix $\boldsymbol{H}$ and the partition function $Z(\beta)$. To evaluate matvecs with $\exp(-\beta \boldsymbol{H})$, we can use the code of Higham [18] that implements an adaptive polynomial approximation [2].

Figure 2 reports the mean estimation error over 100 trials. With just $m = 10$ matvecs, all the variance-reduced methods achieve errors that are 5 orders of magnitude smaller than the Girard–Hutchinson estimator. Furthermore, XTRACE and XNYSTRACE are faster and more precise than HUTCH++. For example, with $m = 40$ matvecs, **XTRACE is 240× more accurate**, and **XNYSTRACE is 2400× more accurate**.

1.4. Theoretical guarantees. To explain the excellent performance of the exchangeable estimators, we establish detailed theoretical guarantees. For theoretical convenience, our analysis uses standard normal test vectors. As a consequence, we can deliver explicit constants that allow us to make meaningful comparisons among the methods.

THEOREM 1.1 (Variance bounds). *Let $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ be a square matrix. Fix the number*

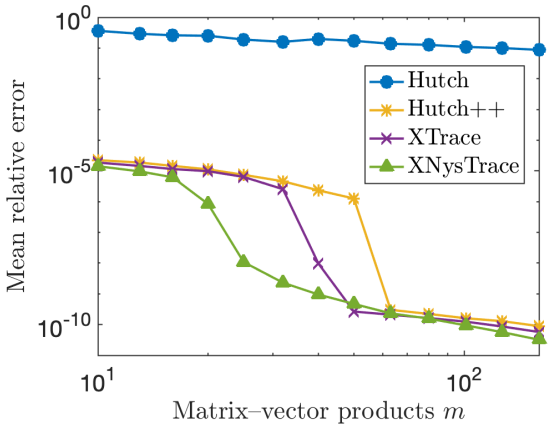


Fig. 2: **Computing a partition function.** Average error of trace estimators applied to the partition function of the transverse-field Ising model. See subsection 1.3.2.

of matvecs: $m \geq 8$. The HUTCH++, XTRACE, *and* XNYSTRACE *estimators are unbiased estimators of the trace. With standard normal test vectors $\boldsymbol{\omega} \sim \text{NORMAL}(\mathbf{0}, \mathbf{I})$, these estimators satisfy the variance bounds*

$$\left(\mathbb{E}\left|\widehat{\text{tr}}_{\text{H++}} - \operatorname{tr} \boldsymbol{A}\right|^2\right)^{1/2} \leq \min_{r \leq m/3-2} \left(\sqrt{2} \frac{\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|_{\text{F}}}{\sqrt{m/3 - r - 1}} \right);$$

$$\left(\mathbb{E}\left|\widehat{\text{tr}}_{\text{X}} - \operatorname{tr} \boldsymbol{A}\right|^2\right)^{1/2} \leq \sqrt{m} \min_{r \leq m/2-4} \left(2 \frac{\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|}{\sqrt{m/2 - r - 3}} + 2\mathrm{e} \frac{\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|_{\text{F}}}{m/2 - r - 3} \right);$$

$$\left(\mathbb{E}\left|\widehat{\text{tr}}_{\text{XN}} - \operatorname{tr} \boldsymbol{A}\right|^2\right)^{1/2} \leq m \min_{r \leq m-6} \left(\sqrt{8} \frac{\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|}{m - r - 5} + \sqrt{2} \frac{\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|_{\text{F}}}{(m - r - 5)^{3/2}} + 5\mathrm{e}^2 \frac{\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|_*}{(m - r - 5)^2} \right).$$

These formulas involve the Frobenius norm $\|\cdot\|_{\text{F}}$, the spectral norm $\|\cdot\|$, and the trace norm $\|\cdot\|_*$. The matrix $[\![\boldsymbol{A}]\!]_r$ is a simultaneous best rank-r approximation of $\boldsymbol{A}$ in these norms.

In addition, for each of these three methods, it suffices to use $m = \mathcal{O}(\eta^{-1/2})$ matvecs to achieve the variance bound

$$\text{Var}[\widehat{\text{tr}}] = \mathbb{E}\left[|\widehat{\text{tr}} - \operatorname{tr} \boldsymbol{A}|^2\right] \leq \eta \|\boldsymbol{A}\|_*^2 \quad \text{for } \eta \in (0, 1). \tag{1.10}$$

The proof of Theorem 1.1 appears in section 5.

As the number m of matvecs increases, Theorem 1.1 ensures that the variance of XTRACE, XNYSTRACE, and HUTCH++ decreases at a rate of $\mathcal{O}(1/m^2)$. Therefore, these algorithms are all superior to the Girard–Hutchinson estimator, whose variance decreases at the Monte Carlo rate $\Theta(1/m)$.

Theorem 1.1 also demonstrates the advantage of XTRACE and XNYSTRACE for matrices whose singular values decay rapidly. This benefit is visible from the error bounds because they allow for larger values r of the approximation rank. As an example, consider a psd matrix $\boldsymbol{A}$ whose eigenvalues have exponential decay with rate $\alpha \in (0, 1)$:

$$\lambda_i(\boldsymbol{A}) \leq \alpha^i \quad \text{for } i = 1, 2, 3, \ldots.$$

The errors of HUTCH++, XTRACE, and XNYSTRACE decay like

$$\begin{aligned}\left(\mathbb{E}\left|\widehat{\mathrm{tr}}_{\mathrm{H++}} - \operatorname{tr} \boldsymbol{A}\right|^2\right)^{1/2} &\leq \quad C_1(\alpha)\,\alpha^{m/3};\\ \left(\mathbb{E}\left|\widehat{\mathrm{tr}}_{\mathrm{X}} - \operatorname{tr} \boldsymbol{A}\right|^2\right)^{1/2} &\leq \sqrt{m}\,C_2(\alpha)\,\alpha^{m/2};\\ \left(\mathbb{E}\left|\widehat{\mathrm{tr}}_{\mathrm{XN}} - \operatorname{tr} \boldsymbol{A}\right|^2\right)^{1/2} &\leq \quad m\,C_3(\alpha)\,\alpha^{m}.\end{aligned}$$

For this class of matrix, XTRACE converges exponentially fast at 1.5× the rate of HUTCH++, and XNYSTRACE converges exponentially fast at 3× the rate of HUTCH++. This is precisely the behavior we observe in Figure 1.

1.5. Benefits. In summary, the XTRACE and XNYSTRACE estimators have several desirable features as compared with previous approaches.

1. **Higher accuracy:** For a fixed budget of m matvecs, XTRACE and XNYSTRACE often yield errors that are *orders of magnitude smaller* than HUTCH++.
2. **Efficient algorithms:** We have designed optimized implementations of XTRACE and XNYSTRACE that only require m matvecs plus $\mathcal{O}(m^2 N)$ arithmetic operations, which is the same computational cost as HUTCH++.
3. **Error estimation:** We can equip the XTRACE and XNYSTRACE estimators with reliable error estimates.

Altogether, these benefits make a compelling case that XTRACE and XNYSTRACE should be the algorithms of choice for trace estimation.

1.6. A brief history of stochastic trace estimation. Girard wrote the first paper [14] on stochastic trace estimation, in which he proposed the estimator (1.2) with test vectors drawn uniformly from a Euclidean sphere. His goal was to develop an efficient way to perform generalized cross-validation for smoothing splines. Hutchinson [19] suggested using random sign vectors instead: $\boldsymbol{\omega} \sim \mathrm{UNIFORM}\{\pm 1\}^N$. See [25, §4] for further details.

In the last five years, researchers have developed far more efficient methods for trace estimation by incorporating variance reduction techniques. In 2017, Saibaba, Alexanderian, and Ipsen [31] proposed a biased estimator that outputs the trace of a low-rank approximation as a surrogate for $\operatorname{tr} \boldsymbol{A}$. Around the same time, Gambhir, Stathopoulos, and Orginos [12] proposed a hybrid estimator, similar to HUTCH++, that outputs the trace of a low-rank approximation, $\operatorname{tr} \widehat{\boldsymbol{A}}$, plus a Girard–Hutchinson estimate for $\operatorname{tr}(\boldsymbol{A} - \widehat{\boldsymbol{A}})$. The paper [23] of Lin contains related ideas.

In 2021, Meyer et al. [26] distilled the ideas from [12, 23] to develop the HUTCH++ algorithm. They proved that HUTCH++ satisfies a worst-case variance bound of $\mathcal{O}(1/m^2)$. Meyer et al. also proposed a version of HUTCH++ that needs only a single pass over the input matrix. The follow-up paper [20] sharpens the analysis of the single-pass algorithm.

Persson, Cortinovis, and Kressner [28] have introduced several refinements to the HUTCH++ estimator. Their first improvement adaptively apportions test vectors between approximating the matrix and estimating the trace of the residual in order to meet an error tolerance. Their second contribution is NYSTRÖM++, a version of HUTCH++ for psd matrices that uses Nyström approximation.

XTRACE and XNYSTRACE build on the previous strategies of variance reduction using low-rank approximation. However, XTRACE and XNYSTRACE take a step forward by also enforcing the exchangeability principle. These algorithms push the ideas of HUTCH++ and NYSTRÖM++ to their limit by using *all* the test vectors for low-rank approximation and *all* the test vectors for residual trace estimation.

To conclude, let us mention two techniques designed for computing the trace of a standard matrix function (that is, $\operatorname{tr} f(\boldsymbol{A})$). First, stochastic Lanczos quadrature [34] approximates the spectral density of $\boldsymbol{A}$, from which estimates of $\operatorname{tr} f(\boldsymbol{A})$ for any function $f(\cdot)$

are immediately accessible. See [8] for a recent overview of stochastic Lanczos quadrature and related ideas. As a second approach, when the matvecs $\boldsymbol{\omega} \mapsto f(\boldsymbol{A})\boldsymbol{\omega}$ are computed using a Krylov subspace method [17, §13.2], the paper [7] recommends reuse of the matvecs from the Krylov subspace method for the purpose of trace estimation.

1.7. Reproducible research. Optimized MATLAB R2022b implementations of our algorithms as well as code to reproduce the experiments in this paper can be found online at https://github.com/eepperly/XTrace.

1.8. Outline. The balance of the paper is organized as follows. Section 2 describes efficient implementations for XTrace, XNysTrace, and the diagonal estimator XDiag. Section 3 discusses error estimation and adaptive stopping, section 4 presents numerical experiments, and section 5 proves our theoretical results.

1.9. Notation. Matrices and vectors are denoted by capital and lowercase bold letters. The ith column of $\boldsymbol{B}$ is expressed as $\boldsymbol{b}_i$, and the (i,j)th entry of $\boldsymbol{B}$ is b_{ij}. For a matrix $\boldsymbol{B}$, we form a matrix $\boldsymbol{B}_{-i}$ by deleting the ith column from $\boldsymbol{B}$. Similarly, we form $\boldsymbol{B}_{-ij}$ by deleting the ith and jth columns. We work with the spectral norm $\|\cdot\|$, the Frobenius norm $\|\cdot\|_{\mathrm{F}}$, and the trace norm $\|\cdot\|_*$. The symbol $[\![\boldsymbol{B}]\!]_r$ denotes a (simultaneous) best rank-r approximation of $\boldsymbol{B}$ with respect to any unitarily invariant norm.

2. Efficient implementation of exchangeable trace estimators. This section works through some issues that arise in the implementation of the XTrace and XNysTrace estimators. Subsections 2.1 and 2.2 show how to implement the new estimators efficiently using insights from numerical linear algebra. Subsection 2.3 discusses a method of renormalizing the test vectors that improves the accuracy of XTrace and XNysTrace. Subsection 2.4 develops the XDiag estimator.

2.1. Computing XTrace. In this section, we develop an efficient implementation of the XTrace estimator from subsection 1.2.1. Recall that $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is a general square matrix, and introduce the test matrix $\boldsymbol{\Omega} = \begin{bmatrix}\boldsymbol{\omega}_1 & \dots & \boldsymbol{\omega}_{m/2}\end{bmatrix} \in \mathbb{R}^{N\times(m/2)}$.

First, we form the matrix product $\boldsymbol{A\Omega}$ and compute the orthogonal decomposition $\boldsymbol{A\Omega} = \boldsymbol{QR}$. Following [10, App. A], we make the critical observation that the basis matrix $\boldsymbol{Q}_{(i)} = \mathrm{orth}(\boldsymbol{A\Omega}_{-i})$ is related to the full basis matrix $\boldsymbol{Q}$ by a rank-one update:

$$\boldsymbol{Q}_{(i)}\boldsymbol{Q}_{(i)}^* = \boldsymbol{Q}\big(\mathbf{I} - \boldsymbol{s}_i\boldsymbol{s}_i^*\big)\boldsymbol{Q}^* \quad \text{where} \quad \boldsymbol{R}_{-i}^*\boldsymbol{s}_i = \mathbf{0} \quad \text{and} \quad \|\boldsymbol{s}_i\|_{\ell_2} = 1. \tag{2.1}$$

Thus, the rank-one update requires a unit vector $\boldsymbol{s}_i \in \mathbb{R}^N$ in the null space of $\boldsymbol{R}_{-i}^*$.

Let us exhibit an efficient algorithm that simultaneously computes all the vectors $\boldsymbol{s}_i$ for $1 \le i \le m/2$. We argue that the matrix $\boldsymbol{S} = \begin{bmatrix}\boldsymbol{s}_1 & \dots & \boldsymbol{s}_{m/2}\end{bmatrix}$ can be represented as

$$\boldsymbol{S} = (\boldsymbol{R}^*)^{-1}\boldsymbol{D}, \tag{2.2}$$

where the diagonal matrix $\boldsymbol{D}$ enforces the normalization of the columns of $\boldsymbol{S}$. Indeed, since $\boldsymbol{R}^*\boldsymbol{S} = \boldsymbol{D}$ is diagonal, the ith column of $\boldsymbol{R}_{-i}^*\boldsymbol{S}$ is the zero vector. We reach the desired conclusion $\boldsymbol{R}_{-i}^*\boldsymbol{s}_i = \mathbf{0}$.

In summary, given the full basis $\boldsymbol{Q}$, we can use (2.2) to compute all the vectors $\boldsymbol{s}_i$ needed to construct the orthogonal projectors $\boldsymbol{Q}_{(i)}\boldsymbol{Q}_{(i)}^*$ for $1 \le i \le m/2$ appearing in (2.1). This calculation requires just $\mathcal{O}(m^3)$ operations, which is dominated by the cost of solving $m/2$ triangular linear systems. It follows that the XTrace estimator can be computed in just $\mathcal{O}(m^2N)$ operations, which is the same asymptotic cost as Hutch++. For full details, see the efficient MATLAB implementation in the supplementary materials Program SM2.1.

2.2. Computing XNysTrace. We can design a similar method to compute the XNysTrace estimator (1.9) efficiently. Let $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ be a psd matrix and define the test matrix $\boldsymbol{\Omega} = \begin{bmatrix}\boldsymbol{\omega}_1 & \dots & \boldsymbol{\omega}_m\end{bmatrix}$.

As before, we compute the orthogonal decomposition $\boldsymbol{A\Omega} = \boldsymbol{QR}$. We can express the Nyström approximation (1.7) in the form

$$\boldsymbol{A}\langle\boldsymbol{\Omega}\rangle = \boldsymbol{QR}\boldsymbol{H}^{-1}\boldsymbol{R}^*\boldsymbol{Q}^*,$$

where $\boldsymbol{H} = \boldsymbol{\Omega}^*\boldsymbol{A\Omega}$. After deleting the ith column from $\boldsymbol{\Omega}$, the resulting Nyström approximation satisfies

$$\boldsymbol{A}\langle\boldsymbol{\Omega}_{-i}\rangle = \boldsymbol{QR}_{-i}\boldsymbol{H}_{(i)}^{-1}\boldsymbol{R}_{-i}^*\boldsymbol{Q}^*, \tag{2.3}$$

where $\boldsymbol{H}_{(i)}$ is $\boldsymbol{H}$ upon deletion of its ith row and column. To compute $\boldsymbol{A}\langle\boldsymbol{\Omega}_{-i}\rangle$ efficiently, we recognize that (2.3) can be expressed as a rank-one update:

$$\boldsymbol{A}\langle\boldsymbol{\Omega}_{-i}\rangle = \boldsymbol{QR}\left(\boldsymbol{H}^{-1} - \frac{\boldsymbol{H}^{-1}\mathbf{e}_i\mathbf{e}_i^*\boldsymbol{H}^{-1}}{\mathbf{e}_i^*\boldsymbol{H}^{-1}\mathbf{e}_i}\right)\boldsymbol{R}^*\boldsymbol{Q}^*. \tag{2.4}$$

Taking advantage of the rank-one update formula (2.4), we can form the XNysTrace estimator using just m matvecs and $\mathcal{O}(m^2N)$ post-processing operations. An efficient MATLAB implementation appears in the supplementary materials Program SM2.2, which incorporates several additional tricks taken from [22, 33] to improve its numerical stability.

2.3. Normalization of test vectors. For the best performance of XTrace and XNysTrace, we recommend the following normalization of the test vectors. First, we draw the test vectors from a spherically symmetric distribution, such as $\boldsymbol{\omega} \sim \text{NORMAL}(\mathbf{0}, \mathbf{I})$. We use these test vectors to form the matrix $\boldsymbol{Q}_{(i)}$ or Nyström approximation $\boldsymbol{A}\langle\boldsymbol{\Omega}_{-i}\rangle$. Second, when computing the basic trace estimates, we **normalize** the test vectors. In XTrace, we compute

$$\boldsymbol{\mu}_i \coloneqq \big(\mathbf{I} - \boldsymbol{Q}_{(i)}\boldsymbol{Q}_{(i)}^*\big)\boldsymbol{\omega}_i \quad \text{and} \quad \boldsymbol{\nu}_i \coloneqq \sqrt{N - \operatorname{rank}(\boldsymbol{Q}_{(i)})}\cdot \boldsymbol{\mu}_i / \big\|\boldsymbol{\mu}_i\big\|_{\ell_2}.$$

To obtain the ith trace estimate, we form

$$\widehat{\operatorname{tr}}_i \coloneqq \operatorname{tr}\big(\boldsymbol{Q}_{(i)}^*(\boldsymbol{AQ}_{(i)})\big) + \boldsymbol{\nu}_i^*\big(\boldsymbol{A\nu}_i\big).$$

For XNysTrace, we set

$$\boldsymbol{P}_{(i)} \coloneqq \operatorname{orth}\boldsymbol{\Omega}_{-i} \quad \text{and} \quad \boldsymbol{\mu}_i \coloneqq (\mathbf{I} - \boldsymbol{P}_{(i)}\boldsymbol{P}_{(i)}^*)\boldsymbol{\omega}_i \quad \text{and} \quad \boldsymbol{\nu}_i \coloneqq \sqrt{N - \operatorname{rank}(\boldsymbol{P}_{(i)})}\cdot \boldsymbol{\mu}_i / \big\|\boldsymbol{\mu}_i\big\|_{\ell_2}.$$

Then define the basic trace estimates

$$\widehat{\operatorname{tr}}_i \coloneqq \operatorname{tr}\boldsymbol{A}\langle\boldsymbol{\Omega}_{-i}\rangle + \boldsymbol{\nu}_i^*\big(\boldsymbol{A\nu}_i\big).$$

The normalization removes a source of variance related to the random lengths of the vectors $\boldsymbol{\mu}_i$, improving the accuracy compared to unnormalized Gaussian test vectors or uniform random vectors on the sphere. We compare this normalization approach against alternative distributions for test vectors in subsection 4.2.

2.4. Diagonal estimation. In the spirit of Girard and Hutchinson, the paper [5] of Bekas, Kokiopoulou, and Saad (BKS) develops an estimator for the diagonal of an implicit matrix:

$$\widehat{\mathbf{diag}}_{\text{BKS}} = \frac{\sum_{i=1}^m \boldsymbol{\omega}_i \odot (\boldsymbol{A\omega}_i)}{\sum_{i=1}^m \boldsymbol{\omega}_i \odot \boldsymbol{\omega}_i}. \tag{2.5}$$

Here, $\odot$ denotes the entrywise product and the division is performed entrywise. The recent paper [4] proposes a biased estimator for the diagonal, called DIAG++, that is inspired by BKS and HUTCH++.

We have observed that the exchangeability principle leads to an unbiased diagonal estimator with lower variance. Our diagonal estimator, XDIAG, takes the form

$$\widehat{\mathbf{diag}}_{\mathrm{X}} = \frac{1}{m/2} \sum_{i=1}^{m/2} \left[\operatorname{diag}\big(\boldsymbol{Q}_{(i)}(\boldsymbol{Q}_{(i)}^* \boldsymbol{A})\big) + \frac{\boldsymbol{\omega}_i \odot \big(\mathbf{I} - \boldsymbol{Q}_{(i)}\boldsymbol{Q}_{(i)}^*\big)(\boldsymbol{A}\boldsymbol{\omega}_i)}{\boldsymbol{\omega}_i \odot \boldsymbol{\omega}_i} \right],$$

where $\boldsymbol{Q}_{(i)}$ is defined in (1.4). In contrast to XTRACE, the XDIAG estimator requires matvecs with $\boldsymbol{A}^*$ in addition to matvecs with $\boldsymbol{A}$. The same ideas from subsection 2.1 allow us to implement XDIAG in $\mathcal{O}(m^2 N)$ operations; an implementation is provided in the supplementary materials (Program SM2.3).

3. Error estimation and adaptive stopping. Our exchangeable estimators depend on averaging over a family of basic estimators, and we can reuse the basic estimators to compute a reliable posterior approximation for the error (subsection 3.1). The error estimate allows us to develop adaptive methods for selecting the number m of matvecs to achieve a specified error tolerance (subsection 3.2). These refinements are very important for practical implementations.

3.1. Error estimation. The XTRACE and XNYSTRACE estimators are both formed as averages of individual trace estimates $\widehat{\mathrm{tr}}_1, \dots, \widehat{\mathrm{tr}}_\ell$ where

$$\begin{aligned} \widehat{\mathrm{tr}}_i &= \operatorname{tr}\big(\boldsymbol{Q}_{(i)}^* \boldsymbol{A} \boldsymbol{Q}_{(i)}\big) + \boldsymbol{\omega}_i^* \big(\mathbf{I} - \boldsymbol{Q}_{(i)}\boldsymbol{Q}_{(i)}^*\big)\boldsymbol{A}\big(\mathbf{I} - \boldsymbol{Q}_{(i)}\boldsymbol{Q}_{(i)}^*\big)\boldsymbol{\omega}_i, \qquad \ell := \frac{m}{2} \qquad \text{(XTRACE)}; \\ \widehat{\mathrm{tr}}_i &= \operatorname{tr}(\boldsymbol{A}\langle \boldsymbol{\Omega}_{-i} \rangle) + \boldsymbol{\omega}_i^* (\boldsymbol{A} - \boldsymbol{A}\langle \boldsymbol{\Omega}_{-i} \rangle)\boldsymbol{\omega}_i, \qquad \ell := m \qquad \text{(XNYSTRACE)}. \end{aligned}$$

The scaled variance of the individual trace estimates $\widehat{\mathrm{tr}}_i$ provides a useful estimate for the squared error in the trace estimate:

$$\widehat{\mathrm{err}}^2 := \frac{1}{\ell(\ell-1)} \sum_{i=1}^{\ell} \big|\widehat{\mathrm{tr}}_i - \widehat{\mathrm{tr}}\big|^2 \approx \big|\operatorname{tr}(\boldsymbol{A}) - \widehat{\mathrm{tr}}\big|^2 \quad \text{where} \quad \widehat{\mathrm{tr}} = \frac{1}{\ell} \sum_{i=1}^{\ell} \widehat{\mathrm{tr}}_i. \tag{3.1}$$

The next result contains an analysis of this posterior error estimator.

PROPOSITION 3.1 (Error estimate). *The error estimate* (3.1) *satisfies*

$$\mathbb{E}\,\widehat{\mathrm{err}}^2 = \frac{1 - \operatorname{Cor}(\widehat{\mathrm{tr}}_1, \widehat{\mathrm{tr}}_2)}{1 + (\ell - 1)\operatorname{Cor}(\widehat{\mathrm{tr}}_1, \widehat{\mathrm{tr}}_2)} \cdot \mathbb{E}\big|\operatorname{tr}(\boldsymbol{A}) - \widehat{\mathrm{tr}}\big|^2.$$

We have written $\operatorname{Cor}(\cdot,\cdot)$ *for the correlation of two random variables.*

The proof and some additional discussion of the correlation $\operatorname{Cor}(\widehat{\mathrm{tr}}_1, \widehat{\mathrm{tr}}_2)$ appears in subsection 5.6. In practice, we find that the individual trace estimators have a small positive correlation, so we typically *underestimate* the true error by a small amount. Thus, the posterior error estimate is a valuable tool. For an illustration, see Figure 5a in subsection 4.3.

3.2. Adaptive stopping. In practice, we often wish to choose the number m of matvecs adaptively to estimate $\operatorname{tr} \boldsymbol{A}$ up to a prescribed accuracy level:

$$|\operatorname{tr}(\boldsymbol{A}) - \widehat{\mathrm{tr}}| \le \varepsilon \cdot |\operatorname{tr} \boldsymbol{A}| \quad \text{for } \varepsilon \in (0, 1).$$

One simple way to achieve this tolerance is through a *doubling strategy*:

1. To initialize, collect m_0 matvecs $\boldsymbol{A\omega}_1, \dots, \boldsymbol{A\omega}_{m_0}$, and set $j \leftarrow 0$.
2. Use $\boldsymbol{A\omega}_1, \dots, \boldsymbol{A\omega}_{m_j}$ to form a trace estimate $\widehat{\mathrm{tr}}^{(j)}$ and an error estimate $\widehat{\mathrm{err}}^{(j)}$.
3. If $\widehat{\mathrm{err}}^{(j)} \le \varepsilon \cdot |\widehat{\mathrm{tr}}^{(j)}|$, then stop.
4. Collect m_j additional matvecs $\boldsymbol{A\omega}_{m_j+1}, \dots, \boldsymbol{A\omega}_{2m_j}$. Set $j \leftarrow j+1$ and $m_j \leftarrow 2m_{j-1}$. Go to step 2.

The doubling strategy requires at most twice the optimal number of matvecs to meet the tolerance and maintains the $\mathcal{O}(m^2N)$ computational cost of XTRACE and XNYSTRACE. We implement this approach in our experiments to produce Figure 5b.

4. Numerical experiments. This section presents a numerical evaluation of XTRACE, XNYSTRACE, and XDIAG. Subsection 4.1 compares different trace estimators on synthetic matrices, subsection 4.2 evaluates XNYSTRACE with different distributions for the test vectors, subsection 4.3 applies XTRACE and XNYSTRACE to computations in quantum statistical physics, and subsection 4.4 applies XDIAG to computations in network science. Code to reproduce the experiments can be found at https://github.com/eepperly/XTrace.

4.1. Comparison of trace estimators. The first experiment is designed to compare the accuracy of six trace estimators that each use m matvecs:

- HUTCH: The Girard–Hutchinson estimator (1.2).
- LRA: The Saibaba et al. [31] estimator (without additional subspace iteration): $\mathrm{tr}\,\widehat{\boldsymbol{A}}$ where $\widehat{\boldsymbol{A}} = \boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}$ and $\boldsymbol{Q} = \mathrm{orth}(\boldsymbol{A\Omega})$.
- The HUTCH++ estimator (1.3).
- The NYSTRÖM++ estimator [28]. We use the implementation provided by the authors of [28], modified to the test vector $\boldsymbol{\omega} \sim \mathrm{UNIFORM}\{\pm 1\}^N$.
- The XTRACE estimator (1.6).
- The XNYSTRACE estimator (1.9).

To create a fair comparison, we apply all six estimators using a test matrix whose entries are uniformly random signs: $\boldsymbol{\Omega} \sim \mathrm{UNIFORM}\{\pm 1\}^{N\times \ell}$, as was used in HUTCH++. The additional benefit for XTRACE and XNYSTRACE of using normalized, spherically symmetric test vectors is explored in subsection 4.2. The supplementary materials (section SM1) contain additional comparisons with the adaptive HUTCH++ algorithm of [28]; this comparison requires a more complicated experimental setup.

We apply each of these estimators to randomly generated matrices of the form

$$\boldsymbol{A}(\boldsymbol{\lambda}) = \boldsymbol{U}\,\mathrm{diag}(\boldsymbol{\lambda})\boldsymbol{U}^*$$

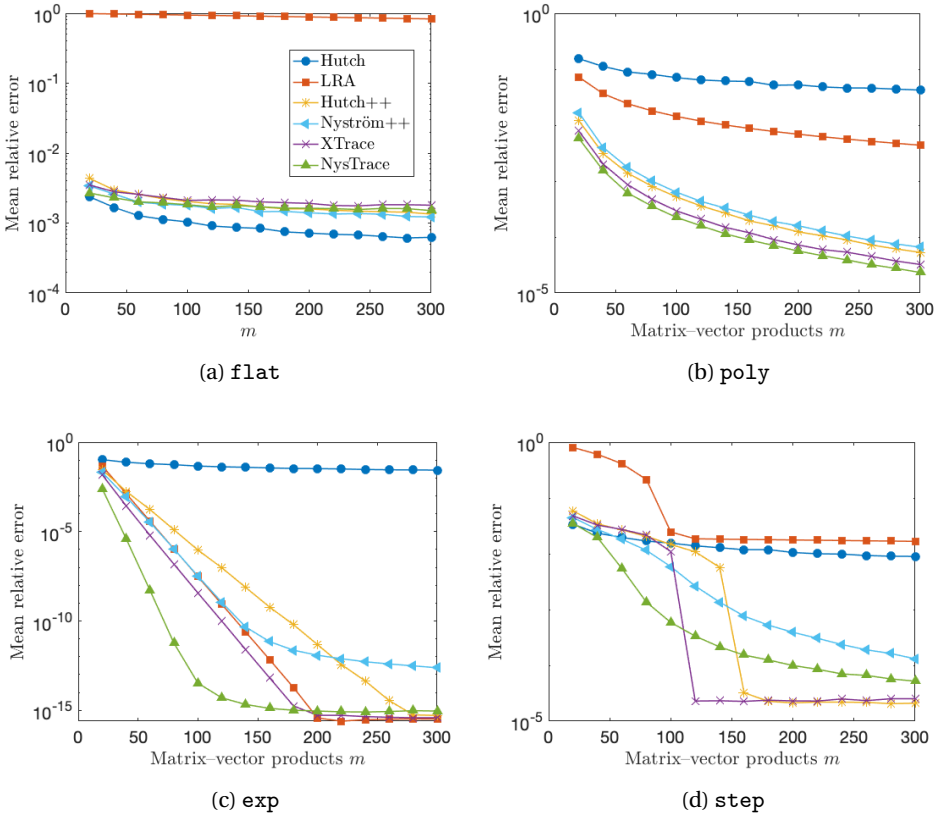
where $\boldsymbol{U}$ is a Haar random orthogonal matrix. We use four choices for the eigenvalues $\boldsymbol{\lambda}$:

- `flat`: $\boldsymbol{\lambda} = (3 - 2(i-1)/(N-1) : i = 1, 2, \dots, N)$.
- `poly`: $\boldsymbol{\lambda} = (i^{-2} : i = 1, 2, \dots, N)$.
- `exp`: $\boldsymbol{\lambda} = (0.7^i : i = 0, 1, \dots, N-1)$.
- `step`: $\boldsymbol{\lambda} = (\underbrace{1, \dots, 1}_{50 \text{ times}}, \underbrace{10^{-3}, \dots, 10^{-3}}_{N-50 \text{ times}})$.

We fix the matrix dimension $N = 1000$, and we report the relative error $|\mathrm{tr}(\boldsymbol{A}) - \widehat{\mathrm{tr}}|/\mathrm{tr}(\boldsymbol{A})$ averaged over 1000 trials.

Discussion. The variance-reduced trace estimators dramatically outperform HUTCH, except on the `flat` instance (Figure 3a). The implication is that HUTCH only makes sense when estimating the trace of a matrix with a nearly flat spectrum. For the `flat` instance, the performance of LRA is especially poor because LRA is a biased estimator that substantially underestimates the trace.

Across all the instances, XTRACE produces smaller errors than HUTCH++, sometimes by orders of magnitude. For the `exp` instance (Figure 3c), the error of XTRACE decays expo-



(a) `flat` (b) `poly`

(c) `exp` (d) `step`

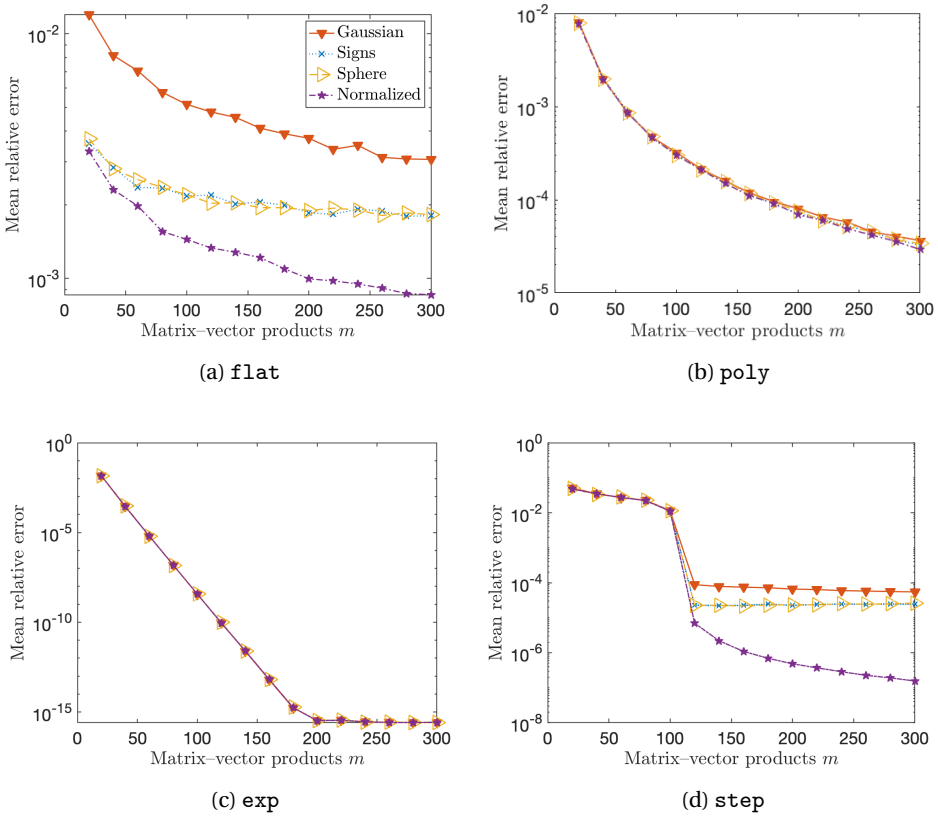
Fig. 3: **Synthetic instances.** Average relative error of trace estimators for matrices with particular spectral profiles using random sign test vectors. See subsection 4.1 and Figure 4.

nentially fast at a rate 1.5× faster than HUTCH++. The superiority of XTRACE is also visible for the `step` instance (Figure 3d) where XTRACE achieves accuracy 10^{-4} with just $m \approx 120$ matvecs, as compared to $m \approx 160$ for HUTCH++.

XNYSTRACE is frequently the most accurate of the trace estimation methods. For the `exp` instance (Figure 3c), XNYSTRACE converges at a rate 2× faster than XTRACE and NYSTRÖM++ and 3× faster than HUTCH++. However, XNYSTRACE (and NYSTRÖM++) can exhibit poor performance for matrices that possess a long tail of slowly decreasing eigenvalues (see the `step` instance in Figure 3d). To understand this phenomenon, observe that the error bounds for XNYSTRACE depend on the trace norm, which is sensitive to slow eigenvalue decay (Theorem 1.1). We can improve the performance by using the normalization approach of subsection 2.3, as we detail in the next section.

4.2. Choice of test vectors. In subsection 2.3, we recommended an implementation of XTRACE and XNYSTRACE that uses rotationally invariant test vectors for low-rank approximation and normalizes the distinguished test vector used for trace estimation.

This section shows how this method can improve over estimators that lack the normalization step. We compare against test vectors from the random sign distribution $\boldsymbol{\omega} \sim \text{UNIFORM}\{\pm 1\}^N$, the standard normal distribution $\boldsymbol{\omega} \sim \text{NORMAL}(\mathbf{0}, \mathbf{I})$, and the uniform distribution on the sphere $\boldsymbol{\omega} \sim \text{UNIFORM}\{\boldsymbol{x} \in \mathbb{R}^N : \|\boldsymbol{x}\|_{\ell_2} = \sqrt{N}\}$. The differences among the



(a) `flat` (b) `poly`

(c) `exp` (d) `step`

Fig. 4: **Normalization of test vectors.** The average relative error of the XTRACE estimator using normalized test vectors (subsection 2.3) as compared with alternative (unnormalized) test vector distributions. See subsection 4.2.

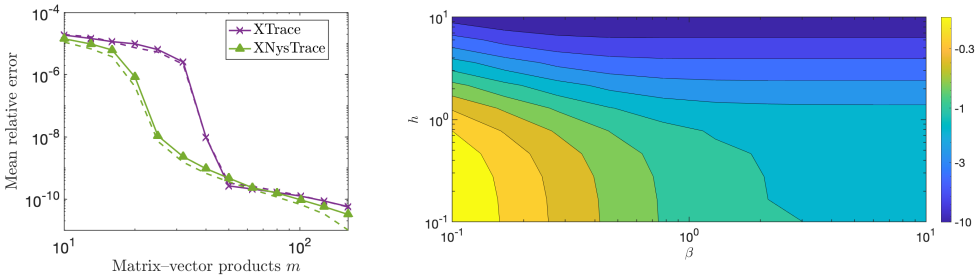
distributions are only visible for matrices whose spectrum has flat segments, as in the `flat` and `step` examples. For these examples, the normalization strategy is conspicuously the best, followed by the uniform sign and uniform sphere distributions, with the standard normal distribution lagging well behind.

4.3. Application: Quantum statistical mechanics. Our next experiment shows the benefits of using XTRACE and XNYSTRACE for an application in quantum physics. To compute a phase diagram, we must evaluate a large number of trace estimators. Our exchangeable estimators reduce the number of matvecs needed to achieve a desired tolerance, and we can use the posterior error estimator to adaptively determine the minimum number m of matvecs.

The average energy of a quantum system with a symmetric Hamiltonian matrix $\boldsymbol{H} \in \mathbb{R}^{N\times N}$ at inverse temperature $\beta > 0$ is

$$E(\beta) = \frac{1}{Z(\beta)} \operatorname{tr}[\boldsymbol{H}\exp(-\beta\boldsymbol{H})] \quad \text{where } Z(\beta) = \operatorname{tr}\exp(-\beta\boldsymbol{H}).$$

The quantity $Z(\beta)$ is the partition function, introduced in subsection 1.3. We observe that $\operatorname{tr}[\boldsymbol{H}\exp(-\beta\boldsymbol{H})]$ and $\operatorname{tr}\exp(-\beta\boldsymbol{H})$ are ideal candidates for estimation using XTRACE and



(a) (b)

Fig. 5: **Quantum statistical mechanics.** *Left*: Mean relative error (solid lines) and mean posterior error estimates (dashed lines) when computing the partition function $Z(\beta = 0.6, h = 10)$. *Right*: Average energy $E(\beta, h)/n$ per site for $\beta, h \in [10^{-1}, 10^{1}]$ as computed by XNysTrace. See subsection 4.3.

XNysTrace, since the matrix exponential leads to rapidly decaying eigenvalues.

We apply XTrace and XNysTrace to compute the partition function and energy for the transverse field Ising model (TFIM) for a periodic 1D chain [29], which is specified by the Hamiltonian matrix

$$\boldsymbol{H} = -\sum_{i=1}^{n} \boldsymbol{Z}_i \boldsymbol{Z}_{i+1} - h \sum_{i=1}^{n} \boldsymbol{X}_i \in \mathbb{R}^{2^n \times 2^n}. \tag{4.1}$$

Here, $\boldsymbol{X}_i$ and $\boldsymbol{Z}_i$ denote Pauli operators acting on the ith site; that is,

$$\boldsymbol{X}_i = \mathbf{I}_{2\times 2}^{\otimes(n-i-1)} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes \mathbf{I}_{2\times 2}^{\otimes(n-i)}, \quad \boldsymbol{Z}_i = \mathbf{I}_{2\times 2}^{\otimes(n-i-1)} \otimes \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \otimes \mathbf{I}_{2\times 2}^{\otimes(n-i)},$$

and $\boldsymbol{Z}_{n+1} = \boldsymbol{Z}_1$ by periodicity. The eigenvalues of $\boldsymbol{H}$ are known exactly [24, eqs. (16)–(17)], which allows us to precisely evaluate the error of stochastic estimates of $Z(\beta) = Z(\beta, h)$ and $E(\beta) = E(\beta, h)$. Before applying stochastic trace estimation, we shift the Hamiltonian matrix by a constant $b = (1 + h)n$ so that $\boldsymbol{H} + b\mathbf{I}$ is positive semidefinite.

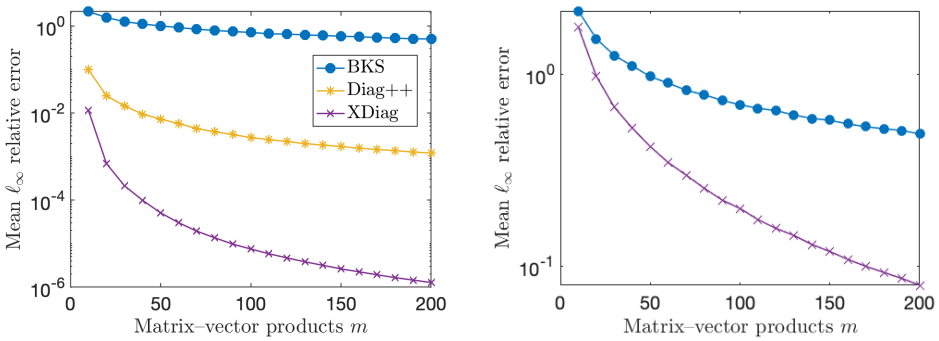
Figure 5a shows the errors of XTrace and XNysTrace when computing the partition function $Z(\beta, h)$ of the TFIM with $n = 18$, $h = 10$, and $\beta = 0.6$; this is the same setting as in Figure 2. The thick lines indicate the average errors over 10 trials, while the dashed lines indicate the average error estimates introduced in subsection 3.1. We observe that the error estimates closely track the true errors, differing by a factor of at most 3.2.

Figure 5b shows the average energy $E(\beta, h)/n$ per site for parameters $\beta, h \in [10^{-1}, 10^{1}]$, up to a relative error of 10^{-3}. We compute the energy by using XNysTrace, together with the doubling strategy from subsection 3.2. To ensure the robustness of the doubling strategy, we use a slightly stricter tolerance $\varepsilon = 10^{-4}$ than our desired accuracy of 10^{-3}.

4.4. Application: Networks. One of the basic problems in network science is to measure the centrality of each node in a graph. We focus on two centrality measures, which can be defined in terms of the adjacency matrix $\boldsymbol{M}$:

1. The *number of triangles* [1] incident on node i is given by $\triangle_i(\boldsymbol{M}) = \frac{1}{2}(\boldsymbol{M}^3)_{ii}$.
2. The *subgraph centrality* [11] of node i is defined as $\mathrm{SC}_i := (\exp(\boldsymbol{M}))_{ii}$.

Both centrality measures are the diagonal entries of functions of the adjacency matrix.



(a) Subgraph centralities $\mathrm{SC}_i(\boldsymbol{M})$ (b) Numbers of triangles $\triangle_i(\boldsymbol{M})$

Fig. 6: **Networks.** Mean error of BKS diagonal estimator and XDIAG for the subgraph centralities (*left*) and triangle numbers (*right*) for the `yeast` graph. See subsection 4.4.

Using the BKS, DIAG++, and XDIAG diagonal estimators, we estimate these centrality measures for the protein–protein interaction network for budding yeast [6], available in the SuiteSparse collection [9]. (Following [4], DIAG++ is omitted for the triangle problem because $\boldsymbol{M}^3/2$ is not psd.) We evaluate the quality of our estimates using the relative ℓ_∞ error

$$\mathrm{error}(\widehat{\mathbf{diag}}) := \frac{\max_{1\le i\le N} |a_{ii} - \widehat{\mathrm{diag}}_i|}{\max_{1\le i\le N} |a_{ii}|},$$

averaged over 1000 trials. Figure 6 shows the results. For the subgraph centrality problem after $m = 200$ matvecs, XDIAG is more accurate than BKS by five orders of magnitude and more accurate than DIAG++ by three orders of magnitude.

5. Theoretical analysis. In this final section, we prove Theorem 1.1, which provides refined error bounds for three trace estimators, and we prove Proposition 3.1, which describes the behavior of the posterior error estimator.

5.1. XTRACE variance bound. To begin, let us establish an initial variance bound for the XTRACE estimator. This result shows that the variance depends on the error in a low-rank approximation of the input matrix. Later, we will bound these errors using standard results for the randomized SVD.

PROPOSITION 5.1 (XTRACE error). *Fix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, and consider the* XTRACE *estimator $\widehat{\mathrm{tr}}_{\mathrm{X}}$ defined in (1.6) with a test matrix $\boldsymbol{\Omega} = \begin{bmatrix}\boldsymbol{\omega}_1 & \dots & \boldsymbol{\omega}_{m/2}\end{bmatrix}$ consisting of $m/2$ standard normal test vectors. The estimator is unbiased: $\mathbb{E}\,\widehat{\mathrm{tr}}_{\mathrm{X}} = \mathrm{tr}\,\boldsymbol{A}$. Moreover, the variance satisfies*

$$\mathrm{Var}[\widehat{\mathrm{tr}}_{\mathrm{X}}] = \mathbb{E}\left|\widehat{\mathrm{tr}}_{\mathrm{X}} - \mathrm{tr}\,\boldsymbol{A}\right|^2 \le \frac{2}{m/2}\mathbb{E}\left\|\left(\mathbf{I} - \boldsymbol{Q}_{(1)}\boldsymbol{Q}_{(1)}^*\right)\boldsymbol{A}\right\|_{\mathrm{F}}^2 + 4\,\mathbb{E}\left\|\left(\mathbf{I} - \boldsymbol{Q}_{(12)}\boldsymbol{Q}_{(12)}^*\right)\boldsymbol{A}\right\|^2,$$

where $\boldsymbol{Q}_{(i)} = \mathrm{orth}(\boldsymbol{A}\boldsymbol{\Omega}_{-i})$ and $\boldsymbol{Q}_{(ij)} = \mathrm{orth}(\boldsymbol{A}\boldsymbol{\Omega}_{-ij})$.

Proof. For all indices $1 \le i, j \le m/2$, we abbreviate the orthogonal projectors $\boldsymbol{\Pi}_i := \boldsymbol{Q}_{(i)}\boldsymbol{Q}_{(i)}^*$ and $\boldsymbol{\Pi}_{ij} := \boldsymbol{Q}_{(ij)}\boldsymbol{Q}_{(ij)}^*$. Note that $\boldsymbol{\omega}_i$ is independent from $\boldsymbol{\Pi}_i$, while $(\boldsymbol{\omega}_i, \boldsymbol{\omega}_j)$ is independent from $\boldsymbol{\Pi}_{ij}$.

For each $1 \le i \le m/2$, we can check that the basic estimator $\widehat{\mathrm{tr}}_i$ defined in (1.5) is unbiased. To do so, condition on $\boldsymbol{\Omega}_{-i}$ and average over $\boldsymbol{\omega}_i$ to arrive at the identity

$$\begin{aligned}\mathbb{E}\big[\widehat{\mathrm{tr}}_i \,\big|\, \boldsymbol{\Omega}_{-i}\big] &= \mathbb{E}\big[\mathrm{tr}(\boldsymbol{\Pi}_i \boldsymbol{A} \boldsymbol{\Pi}_i) + \boldsymbol{\omega}_i^* (\mathbf{I} - \boldsymbol{\Pi}_i) \boldsymbol{A} (\mathbf{I} - \boldsymbol{\Pi}_i) \boldsymbol{\omega}_i \,\big|\, \boldsymbol{\Omega}_{-i}\big] \\ &= \mathbb{E}\big[\mathrm{tr}(\boldsymbol{\Pi}_i \boldsymbol{A} \boldsymbol{\Pi}_i) + \mathrm{tr}\big((\mathbf{I} - \boldsymbol{\Pi}_i) \boldsymbol{A} (\mathbf{I} - \boldsymbol{\Pi}_i)\big) \,\big|\, \boldsymbol{\Omega}_{-i}\big] = \mathrm{tr}\, \boldsymbol{A}.\end{aligned} \tag{5.1}$$

The second equality uses the fact that each test vector $\boldsymbol{\omega}_i$ is isotropic and independent from $\boldsymbol{\Pi}_i$, which is a function of $\boldsymbol{\Omega}_{-i}$. The third equality follows when we cycle the traces and invoke the fact that the projector $\mathbf{I} - \boldsymbol{\Pi}_i$ is idempotent. We confirm that $\widehat{\mathrm{tr}}_i$ is unbiased by applying the tower law to take the total expectation of (5.1). The full estimator $\widehat{\mathrm{tr}}_{\mathrm{X}}$ is unbiased because it is an average of the unbiased estimators $\widehat{\mathrm{tr}}_i$.

Next, to bound the variance, we use the exchangeability of $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_{m/2}$ to compute

$$\mathrm{Var}\big[\widehat{\mathrm{tr}}_{\mathrm{X}}\big] = \mathrm{Var}\bigg[\frac{1}{m/2} \sum_{i=1}^{m/2} \widehat{\mathrm{tr}}_i\bigg] = \bigg(\frac{1}{m/2}\bigg) \underbrace{\mathrm{Var}\big[\widehat{\mathrm{tr}}_1\big]}_{\mathsf{A}} + \bigg(1 - \frac{1}{m/2}\bigg) \underbrace{\mathrm{Cov}\big[\widehat{\mathrm{tr}}_1, \widehat{\mathrm{tr}}_2\big]}_{\mathsf{B}}.$$

Hence, $\mathrm{Var}\big[\widehat{\mathrm{tr}}_{\mathrm{X}}\big]$ is the weighted average of a variance term A and a covariance term B.

To evaluate the variance term A, condition on $\boldsymbol{\Omega}_{-1}$ and average over $\boldsymbol{\omega}_1$. Thus,

$$\begin{aligned}\mathsf{A} = \mathrm{Var}\big[\widehat{\mathrm{tr}}_1\big] &= \mathbb{E}\big[\mathrm{Var}\big[\widehat{\mathrm{tr}}_1 \,\big|\, \boldsymbol{\Omega}_{-1}\big]\big] + \mathrm{Var}\big[\mathbb{E}\big[\widehat{\mathrm{tr}}_1 \,\big|\, \boldsymbol{\Omega}_{-1}\big]\big] \\ &= \mathbb{E}\big[\mathrm{Var}\big[\mathrm{tr}(\boldsymbol{\Pi}_1^* \boldsymbol{A} \boldsymbol{\Pi}_1) + \boldsymbol{\omega}_1^* (\mathbf{I} - \boldsymbol{\Pi}_1) \boldsymbol{A} (\mathbf{I} - \boldsymbol{\Pi}_i) \boldsymbol{\omega}_1 \,\big|\, \boldsymbol{\Omega}_{-1}\big]\big] \\ &= \mathbb{E}\big[\mathrm{Var}\big[\boldsymbol{\omega}_1^* (\mathbf{I} - \boldsymbol{\Pi}_1) \boldsymbol{A} (\mathbf{I} - \boldsymbol{\Pi}_1) \boldsymbol{\omega}_1 \,\big|\, \boldsymbol{\Omega}_{-1}\big]\big] \le 2\,\mathbb{E}\big\|(\mathbf{I} - \boldsymbol{\Pi}_1) \boldsymbol{A} (\mathbf{I} - \boldsymbol{\Pi}_1)\big\|_{\mathrm{F}}^2.\end{aligned}$$

The first relation is the chain rule for the variance. To pass to the second line, we invoke the fact (5.1) that the conditional expectation is constant. To pass to the third line, we drop the trace, which is conditionally constant. The last relation follows from a direct calculation using the facts that $\boldsymbol{\omega}_1$ is standard normal and independent from $\boldsymbol{\Pi}_1$.

To bound the covariance term B, it is helpful to isolate the part of the covariance that only depends on $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$. We rely on the following observation. For any (random) matrix $\boldsymbol{X} \in \mathbb{R}^{N \times N}$ that is independent from $\boldsymbol{\omega}_1$ and $\boldsymbol{\omega}_2$, we may calculate that

$$\mathbb{E}\big[\big(\widehat{\mathrm{tr}}_1 - \mathrm{tr}\, \boldsymbol{A} + \mathrm{tr}\, \boldsymbol{X} - \boldsymbol{\omega}_1^* \boldsymbol{X} \boldsymbol{\omega}_1\big)\big(\widehat{\mathrm{tr}}_2 - \mathrm{tr}\, \boldsymbol{A} + \mathrm{tr}\, \boldsymbol{X} - \boldsymbol{\omega}_2^* \boldsymbol{X} \boldsymbol{\omega}_2\big)\big] \tag{5.2}$$

$$= \mathbb{E}\big[\big(\widehat{\mathrm{tr}}_1 - \mathrm{tr}\, \boldsymbol{A} + \mathrm{tr}\, \boldsymbol{X} - \boldsymbol{\omega}_1^* \boldsymbol{X} \boldsymbol{\omega}_1\big)\big(\widehat{\mathrm{tr}}_2 - \mathrm{tr}\, \boldsymbol{A}\big)\big] \tag{5.3}$$

$$= \mathbb{E}\big[\big(\widehat{\mathrm{tr}}_1 - \mathrm{tr}\, \boldsymbol{A}\big)\big(\widehat{\mathrm{tr}}_2 - \mathrm{tr}\, \boldsymbol{A}\big)\big] = \mathrm{Cov}\big[\widehat{\mathrm{tr}}_1, \widehat{\mathrm{tr}}_2\big]. \tag{5.4}$$

To pass to (5.3), we condition on $\boldsymbol{\Omega}_{-1}$ and average over $\boldsymbol{\omega}_1$, exploiting the fact (5.1) that $\widehat{\mathrm{tr}}_1$ is an unbiased estimator of $\mathrm{tr}\, \boldsymbol{A}$, conditional on $\boldsymbol{\Omega}_{-1}$. To pass to (5.4), condition on $\boldsymbol{\Omega}_{-2}$ and average over $\boldsymbol{\omega}_2$.

To continue, select the particular random matrix $\boldsymbol{X} = (\mathbf{I} - \boldsymbol{\Pi}_{12}) \boldsymbol{A} (\mathbf{I} - \boldsymbol{\Pi}_{12})$. Applying the Cauchy–Schwarz inequality, we find that

$$\begin{aligned}\mathsf{B} = \mathrm{Cov}\big[\widehat{\mathrm{tr}}_1, \widehat{\mathrm{tr}}_2\big] &= \mathbb{E}\big[\big(\widehat{\mathrm{tr}}_1 - \mathrm{tr}\, \boldsymbol{A} + \mathrm{tr}\, \boldsymbol{X} - \boldsymbol{\omega}_1^* \boldsymbol{X} \boldsymbol{\omega}_1\big)\big(\widehat{\mathrm{tr}}_2 - \mathrm{tr}\, \boldsymbol{A} + \mathrm{tr}\, \boldsymbol{X} - \boldsymbol{\omega}_2^* \boldsymbol{X} \boldsymbol{\omega}_2\big)\big] \\ &\le \mathbb{E}\big|\widehat{\mathrm{tr}}_1 - \mathrm{tr}\, \boldsymbol{A} + \mathrm{tr}\, \boldsymbol{X} - \boldsymbol{\omega}_1^* \boldsymbol{X} \boldsymbol{\omega}_1\big|^2 \\ &= \mathbb{E}\,\mathrm{Var}\big[\boldsymbol{\omega}_1^* \big[(\mathbf{I} - \boldsymbol{\Pi}_1) \boldsymbol{A} (\mathbf{I} - \boldsymbol{\Pi}_1) - \boldsymbol{X}\big] \boldsymbol{\omega}_1 \,\big|\, \boldsymbol{\Omega}_{-1}\big] \\ &\le 2\,\mathbb{E}\big\|(\mathbf{I} - \boldsymbol{\Pi}_1) \boldsymbol{A} (\mathbf{I} - \boldsymbol{\Pi}_1) - \boldsymbol{X}\big\|_{\mathrm{F}}^2.\end{aligned}$$

Since $\mathrm{range}(\boldsymbol{Q}_{(12)}) \subseteq \mathrm{range}(\boldsymbol{Q}_{(1)})$, we have the relations $\boldsymbol{\Pi}_{12} = \boldsymbol{\Pi}_1 \boldsymbol{\Pi}_{12}$ and $\mathbf{I} - \boldsymbol{\Pi}_1 = (\mathbf{I} - \boldsymbol{\Pi}_1)(\mathbf{I} -$

$\boldsymbol{\Pi}_{12}$). Since $\boldsymbol{X} = (\mathbf{I} - \boldsymbol{\Pi}_{12})\boldsymbol{A}(\mathbf{I} - \boldsymbol{\Pi}_{12})$, it follows that

$$\begin{aligned}\left\|(\mathbf{I}-\boldsymbol{\Pi}_1)\boldsymbol{A}(\mathbf{I}-\boldsymbol{\Pi}_1) - \boldsymbol{X}\right\|_{\mathrm{F}}^2 &= \left\|(\mathbf{I}-\boldsymbol{\Pi}_1)\boldsymbol{X}(\mathbf{I}-\boldsymbol{\Pi}_1) - \boldsymbol{X}\right\|_{\mathrm{F}}^2 = \left\|(\mathbf{I}-\boldsymbol{\Pi}_1)\boldsymbol{X}\boldsymbol{\Pi}_1 + \boldsymbol{\Pi}_1\boldsymbol{X}\right\|_{\mathrm{F}}^2 \\ &= \left\|(\mathbf{I}-\boldsymbol{\Pi}_1)\boldsymbol{X}\boldsymbol{\Pi}_1\right\|_{\mathrm{F}}^2 + \left\|\boldsymbol{\Pi}_1\boldsymbol{X}\right\|_{\mathrm{F}}^2 \le \|\boldsymbol{X}\boldsymbol{\Pi}_1\|_{\mathrm{F}}^2 + \|\boldsymbol{\Pi}_1\boldsymbol{X}\|_{\mathrm{F}}^2 \\ &= \left\|\boldsymbol{X}(\boldsymbol{\Pi}_1 - \boldsymbol{\Pi}_{12})\right\|_{\mathrm{F}}^2 + \left\|(\boldsymbol{\Pi}_1 - \boldsymbol{\Pi}_{12})\boldsymbol{X}\right\|_{\mathrm{F}}^2 \\ &= \left\|\boldsymbol{X}(\boldsymbol{\Pi}_1 - \boldsymbol{\Pi}_{12})\right\|^2 + \left\|(\boldsymbol{\Pi}_1 - \boldsymbol{\Pi}_{12})\boldsymbol{X}\right\|^2 \le 2\|\boldsymbol{X}\|^2.\end{aligned}$$

We invoke the Pythagorean theorem to pass to the second line. To reach the third line, exploit the representations $\boldsymbol{X} = \boldsymbol{X}(\mathbf{I} - \boldsymbol{\Pi}_{12})$ and $\boldsymbol{X} = (\mathbf{I} - \boldsymbol{\Pi}_{12})\boldsymbol{X}$. To pass to the fourth line, note that $\boldsymbol{\Pi}_1 - \boldsymbol{\Pi}_{12}$ is a rank-one orthogonal projector; the Frobenius norm and spectral norm coincide for rank-one matrices. Combining the last two displays, we deduce that

$$\mathsf{B} = \operatorname{Cov}[\widehat{\mathrm{tr}}_1, \widehat{\mathrm{tr}}_2] \le 4\,\mathbb{E}\left\|(\mathbf{I} - \boldsymbol{\Pi}_{12})\boldsymbol{A}(\mathbf{I} - \boldsymbol{\Pi}_{12})\right\|^2.$$

Combining the estimates for A and B, we achieve the stated bound for the variance. ∎

5.2. Hutch++ variance bound. By a similar argument, we can obtain an initial variance bound for the Hutch++ estimator. This result is more elementary because it does not require us to account for interactions between the simple estimators.

Proposition 5.2 (Hutch++ error). *Fix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, and consider the* Hutch++ *estimator $\widehat{\mathrm{tr}}_{\mathrm{H++}}$ defined in (1.3) with $2m/3$ standard normal test vectors. The estimator is unbiased: $\mathbb{E}\,\widehat{\mathrm{tr}}_{\mathrm{H++}} = \operatorname{tr}\boldsymbol{A}$. Moreover, the variance satisfies*

$$\mathbb{E}\left|\widehat{\mathrm{tr}}_{\mathrm{H++}} - \operatorname{tr}\boldsymbol{A}\right|^2 \le \frac{2}{m/3}\mathbb{E}\|(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^*)\boldsymbol{A}\|_{\mathrm{F}}^2,$$

where $\boldsymbol{Q} = \operatorname{orth}(\boldsymbol{A}\boldsymbol{\Omega})$ and $\boldsymbol{\Omega} = \begin{bmatrix}\boldsymbol{\omega}_{m/3+1} & \dots & \boldsymbol{\omega}_{2m/3}\end{bmatrix}$.

Proof. The idea is to condition on the low-rank approximation $\boldsymbol{Q}\boldsymbol{Q}^*\boldsymbol{A}$ and invoke the chain rule for the variance, as in the proof of Proposition 5.1. See the argument in [27, Thm. 10], which was supplied by the second author of this paper. ∎

5.3. XNysTrace variance bound. Last, we establish an initial variance bound for the XNysTrace estimator. This result shows how the variance depends on the error in a randomized Nyström approximation. Later, we will use recent results for the Nyström approximation to obtain a complete variance bound.

Proposition 5.3 (XNysTrace error). *Let $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ be psd. Consider the* XNysTrace *estimator $\widehat{\mathrm{tr}}_{NX}$ with m standard normal test vectors as defined in (1.9). The estimator is unbiased: $\mathbb{E}\,\widehat{\mathrm{tr}}_{\mathrm{NX}} = \operatorname{tr}\boldsymbol{A}$. Moreover, the variance satisfies the bound*

$$\mathbb{E}\left|\widehat{\mathrm{tr}}_{\mathrm{XN}} - \operatorname{tr}\boldsymbol{A}\right|^2 \le \frac{2}{m}\mathbb{E}\left\|\boldsymbol{A} - \boldsymbol{A}\langle\boldsymbol{\Omega}_{-1}\rangle\right\|_{\mathrm{F}}^2 + 2\,\mathbb{E}\left\|\boldsymbol{A} - \boldsymbol{A}\langle\boldsymbol{\Omega}_{-12}\rangle\right\|^2.$$

Proof. The proof resembles the proof of Proposition 5.1 but is slightly simpler. The unbiasedness of XNysTrace follows from a short computation similar to (5.1).

To control the variance, we calculate that

$$\operatorname{Var}[\widehat{\mathrm{tr}}_{\mathrm{XN}}] = \operatorname{Var}\left[\frac{1}{m}\sum_{i=1}^{m}\widehat{\mathrm{tr}}_i\right] = \left(\frac{1}{m}\right)\underbrace{\operatorname{Var}[\widehat{\mathrm{tr}}_1]}_{\mathsf{A}} + \left(1 - \frac{1}{m}\right)\underbrace{\operatorname{Cov}[\widehat{\mathrm{tr}}_1, \widehat{\mathrm{tr}}_2]}_{\mathsf{B}}.$$

The variance term is exactly

$$\mathsf{A} = \operatorname{Var}[\widehat{\mathrm{tr}}_1] = 2\mathbb{E}\left\|\boldsymbol{A} - \boldsymbol{A}\langle\boldsymbol{\Omega}_{-1}\rangle\right\|_{\mathrm{F}}^2.$$

To bound the covariance term B, we set $\boldsymbol{X} = \boldsymbol{A} - \boldsymbol{A}\langle\boldsymbol{\Omega}_{-12}\rangle$ in (5.2). Applying the Cauchy–Schwarz inequality, we find that

$$\begin{aligned}
\mathsf{B} = \operatorname{Cov}[\widehat{\mathrm{tr}}_1, \widehat{\mathrm{tr}}_2] &= \mathbb{E}\big[\big(\widehat{\mathrm{tr}}_1 - \operatorname{tr}\boldsymbol{A} + \operatorname{tr}\boldsymbol{X} - \boldsymbol{\omega}_1^*\boldsymbol{X}\boldsymbol{\omega}_1\big)\big(\widehat{\mathrm{tr}}_2 - \operatorname{tr}\boldsymbol{A} + \operatorname{tr}\boldsymbol{X} - \boldsymbol{\omega}_2^*\boldsymbol{X}\boldsymbol{\omega}_2\big)\big] \\
&\le \mathbb{E}\big|\widehat{\mathrm{tr}}_1 - \operatorname{tr}\boldsymbol{A} + \operatorname{tr}\boldsymbol{X} - \boldsymbol{\omega}_1^*\boldsymbol{X}\boldsymbol{\omega}_1\big|^2 \\
&= \mathbb{E}\operatorname{Var}\big[\boldsymbol{\omega}_1^*\big[\boldsymbol{A}\langle\boldsymbol{\Omega}_{-12}\rangle - \boldsymbol{A}\langle\boldsymbol{\Omega}_{-1}\rangle\big]\boldsymbol{\omega}_1 \,\big|\, \boldsymbol{\Omega}_{-1}\big] \\
&= 2\mathbb{E}\big\|\boldsymbol{A}\langle\boldsymbol{\Omega}_{-1}\rangle - \boldsymbol{A}\langle\boldsymbol{\Omega}_{-12}\rangle\big\|_{\mathrm{F}}^2.
\end{aligned}$$

The psd matrix $\boldsymbol{A}\langle\boldsymbol{\Omega}_{-1}\rangle - \boldsymbol{A}\langle\boldsymbol{\Omega}_{-12}\rangle$ has rank one, and it is bounded above by $\boldsymbol{A} - \boldsymbol{A}\langle\boldsymbol{\Omega}_{-12}\rangle$ in the psd order. Therefore,

$$\mathsf{B} = \operatorname{Cov}[\widehat{\mathrm{tr}}_1, \widehat{\mathrm{tr}}_2] \le 2\mathbb{E}\big\|\boldsymbol{A} - \boldsymbol{A}\langle\boldsymbol{\Omega}_{-12}\rangle\big\|^2.$$

Combine the displays to complete the proof. ∎

5.4. Error bounds for low-rank approximations. To prove the main result, Theorem 1.1, we need two auxiliary lemmas. First, we present error bounds for randomized SVD and randomized Nyström approximation, drawn from the recent paper [32].

LEMMA 5.4 (Randomized SVD and randomized Nyström error). *Fix a matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$, and draw a standard normal matrix $\boldsymbol{\Omega} \in \mathbb{R}^{N\times k}$. For any $r \le k-2$, the randomized SVD error is bounded by*

$$\begin{aligned}
\mathbb{E}\|(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^*)\boldsymbol{A}\|^2 &\le \frac{k+r-1}{k-r-1}\bigg(\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|^2 + \frac{\mathrm{e}^2}{k-r}\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|_{\mathrm{F}}^2\bigg), \\
\mathbb{E}\|(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^*)\boldsymbol{A}\|_{\mathrm{F}}^2 &\le \frac{k-1}{k-r-1}\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|_{\mathrm{F}}^2,
\end{aligned}$$

where $\boldsymbol{Q} = \operatorname{orth}(\boldsymbol{A}\boldsymbol{\Omega})$.

Assume that $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is a psd matrix. For any $r \le k-4$, the randomized Nyström error is bounded by

$$\begin{aligned}
\big(\mathbb{E}\|\boldsymbol{A} - \boldsymbol{A}\langle\boldsymbol{\Omega}\rangle\|^2\big)^{1/2} &\le \frac{k+r-1}{k-r-3}\bigg(\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\| + \frac{\sqrt{3}\mathrm{e}^2}{k-r}\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|_*\bigg), \\
\big(\mathbb{E}\|\boldsymbol{A} - \boldsymbol{A}\langle\boldsymbol{\Omega}\rangle\|_{\mathrm{F}}^2\big)^{1/2} &\le \frac{k-2}{k-r-3}\bigg(\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|_{\mathrm{F}} + \frac{1}{\sqrt{k-r}}\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|_*\bigg).
\end{aligned}$$

Second, we report a standard fact about the decay rate of the singular values, which is also exploited in [27, Lem. 13] and [13, Lem. 7]. We omit the easy proof.

FACT 5.5. *For any matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ and any $r \ge 1$,*

$$\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\| \le \frac{\|\boldsymbol{A}\|_*}{r+1}, \qquad \|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|_{\mathrm{F}} \le \frac{\|\boldsymbol{A}\|_*}{2\sqrt{r}}.$$

5.5. The complete variance bound. To establish the main result, Theorem 1.1, we begin with the initial variance bounds and introduce the results from Lemma 5.4 and Fact 5.5.

Proof of Theorem 1.1. We recognize that all the terms in the error formulas in Propositions 5.1 to 5.3 reflect the squared approximation error in a randomized SVD or a randomized Nyström approximation. Therefore, we can apply the error bounds in Lemma 5.4 to obtain more explicit error representations. For HUTCH++, when $r \le m/3 - 2$,

$$\mathbb{E}\big|\widehat{\mathrm{tr}}_{\mathrm{H++}} - \operatorname{tr}\boldsymbol{A}\big|^2 \le \frac{2}{m/3}\mathbb{E}\|(\mathbf{I} - \boldsymbol{Q}\boldsymbol{Q}^*)\boldsymbol{A}\|_{\mathrm{F}}^2 \le \frac{2}{m/3 - r - 1}\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|_{\mathrm{F}}^2.$$

For XTRACE, when $r \le m/2 - 4$,

$$\begin{aligned}\mathbb{E}\left|\widehat{\mathrm{tr}}_{\mathrm{X}} - \operatorname{tr} \boldsymbol{A}\right|^2 &\le \frac{2}{m/2}\mathbb{E}\left\|\left(\mathbf{I} - \boldsymbol{Q}_{(1)}\boldsymbol{Q}_{(1)}^*\right)\boldsymbol{A}\right\|_{\mathrm{F}}^2 + 4\mathbb{E}\left\|\left(\mathbf{I} - \boldsymbol{Q}_{(12)}\boldsymbol{Q}_{(12)}^*\right)\boldsymbol{A}\right\|^2 \\ &\le \frac{4m}{m/2 - r - 3}\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|^2 + \frac{4\mathrm{e}^2 m}{(m/2 - r - 3)^2}\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|_{\mathrm{F}}^2.\end{aligned}$$

Last, for XNYSTRACE, when $r \le m - 6$,

$$\begin{aligned}\left(\mathbb{E}\left|\widehat{\mathrm{tr}}_{\mathrm{XN}} - \operatorname{tr} \boldsymbol{A}\right|^2\right)^{1/2} &\le \frac{\sqrt{2}}{\sqrt{m}}\left(\mathbb{E}\left\|\boldsymbol{A} - \boldsymbol{A}\langle\boldsymbol{\Omega}_{-1}\rangle\right\|_{\mathrm{F}}^2\right)^{1/2} + \sqrt{2}\left(\mathbb{E}\left\|\boldsymbol{A} - \boldsymbol{A}\langle\boldsymbol{\Omega}_{-12}\rangle\right\|^2\right)^{1/2} \\ &\le \frac{\sqrt{8}m}{m - r - 5}\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\| + \frac{\sqrt{2}m}{(m - r - 5)^{3/2}}\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|_{\mathrm{F}} + \frac{5\mathrm{e}^2 m}{(m - r - 5)^2}\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|_*.\end{aligned}$$

Thus, we confirm the detailed error bounds in Theorem 1.1.

All that remains is to show that each trace estimator $\widehat{\mathrm{tr}}$ satisfies

$$\left(\mathbb{E}|\widehat{\mathrm{tr}} - \operatorname{tr} \boldsymbol{A}|^2\right)^{1/2} \le \frac{C}{m}\|\boldsymbol{A}\|_*, \tag{5.5}$$

for an absolute constant C. To that end, apply Fact 5.5 to bound $\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|$ and $\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|_{\mathrm{F}}$ in terms of $\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_r\|_*$. For HUTCH++, we set $r = \lfloor m/6 \rfloor - 1$. For XTRACE, we set $r = \lfloor m/4 \rfloor - 2$. For XNYSTRACE, we set $r = \lfloor m/2 \rfloor - 3$. Simplifying yields (5.5) for each estimator. ∎

5.6. Proof of Proposition 3.1. Last, we must argue that the posterior error estimator $\widehat{\mathrm{err}}$ defined in (3.1) reflects the actual error. We instate the notation from Proposition 3.1.

For both XTRACE and XNYSTRACE, each individual trace estimate $\widehat{\mathrm{tr}}_i$ is unbiased. As a consequence, the variance takes the form

$$\mathbb{E}\left|\widehat{\mathrm{tr}} - \operatorname{tr} \boldsymbol{A}\right|^2 = \operatorname{Var}(\widehat{\mathrm{tr}}) = \frac{1}{\ell^2}\sum_{i,j=1}^{\ell}\operatorname{Cov}\left(\widehat{\mathrm{tr}}_i, \widehat{\mathrm{tr}}_j\right).$$

A short calculation yields

$$\mathbb{E}\,\widehat{\mathrm{err}}^2 = \frac{1}{\ell^2(\ell - 1)}\sum_{i,j=1}^{\ell}\left[\operatorname{Var}\left(\widehat{\mathrm{tr}}_j\right) - \operatorname{Cov}\left(\widehat{\mathrm{tr}}_i, \widehat{\mathrm{tr}}_j\right)\right].$$

Since the samples $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_\ell$ are exchangeable, the variance is the same for each j and the covariance is the same for all $i \ne j$. Therefore,

$$\begin{aligned}\mathbb{E}\left|\widehat{\mathrm{tr}} - \operatorname{tr}(\boldsymbol{A})\right|^2 &= \frac{1}{\ell}\operatorname{Var}\left(\widehat{\mathrm{tr}}_1\right) + \frac{\ell - 1}{\ell}\operatorname{Cov}\left(\widehat{\mathrm{tr}}_1, \widehat{\mathrm{tr}}_2\right), \\ \mathbb{E}\,\widehat{\mathrm{err}}^2 &= \frac{1}{\ell}\operatorname{Var}\left(\widehat{\mathrm{tr}}_1\right) - \frac{1}{\ell}\operatorname{Cov}\left(\widehat{\mathrm{tr}}_1, \widehat{\mathrm{tr}}_2\right).\end{aligned}$$

The result follows when we take the ratio of these two quantities and simplify. ∎

As a final comment, we observe that the calculations in subsection 5.1 show for *symmetric matrices* that the XTRACE correlations are bounded by

$$\operatorname{Cor}(\widehat{\mathrm{tr}}_1, \widehat{\mathrm{tr}}_2) \le 2\frac{\mathbb{E}\left\|(\mathbf{I} - \boldsymbol{Q}_{(12)}\boldsymbol{Q}_{(12)}^*)\boldsymbol{A}(\mathbf{I} - \boldsymbol{Q}_{(12)}\boldsymbol{Q}_{(12)}^*)\right\|^2}{\mathbb{E}\left\|(\mathbf{I} - \boldsymbol{Q}_{(1)}\boldsymbol{Q}_{(1)}^*)\boldsymbol{A}(\mathbf{I} - \boldsymbol{Q}_{(1)}\boldsymbol{Q}_{(1)}^*)\right\|_{\mathrm{F}}^2},$$

where $\boldsymbol{Q}_{(1)}$ and $\boldsymbol{Q}_{(12)}$ are defined in Proposition 5.1. These correlations are small for matrices with slow rates of singular value decay, i.e., when $\|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_{m/2-1}\|_{\mathrm{F}} \gg \|\boldsymbol{A} - [\![\boldsymbol{A}]\!]_{m/2-2}\|$.

In practice, we observe the correlations to be small even for matrices with singular values which decay more quickly. As an example, for the matrix with exponentially decaying eigenvalues in Figure 1, the XTRACE correlations (measured over 10^4 independent runs of the algorithm) are no higher than 0.06 and the average error estimate is correct up to a factor of 1.2.

Acknowledgments. We thank Eitan Levin for helpful discussions regarding the fast implementation of XTRACE.

REFERENCES

[1] M. AL HASAN AND V. S. DAVE, *Triangle counting in large networks: A review*, WIREs Data Mining and Knowledge Discovery, 8 (2018), p. e1226, https://doi.org/10.1002/widm.1226. 15

[2] A. H. AL-MOHY AND N. J. HIGHAM, *Computing the action of the matrix exponential, with an application to exponential integrators*, SIAM Journal on Scientific Computing, 33 (2011), pp. 488–511, https://doi.org/10.1137/100788860. 6

[3] H. AVRON, P. MAYMOUNKOV, AND S. TOLEDO, *Blendenpik: Supercharging LAPACK's least-squares solver*, SIAM Journal on Scientific Computing, 32 (2010), pp. 1217–1236, https://doi.org/10.1137/090767911. 1

[4] R. A. BASTON AND Y. NAKATSUKASA, *Stochastic diagonal estimation: Probabilistic bounds and an improved algorithm*, Jan. 2022, https://arxiv.org/abs/2201.10684. 11, 16

[5] C. BEKAS, E. KOKIOPOULOU, AND Y. SAAD, *An estimator for the diagonal of a matrix*, Applied Numerical Mathematics, 57 (2007), pp. 1214–1229, https://doi.org/10.1016/j.apnum.2007.01.003. 10

[6] D. BU, Y. ZHAO, L. CAI, H. XUE, X. ZHU, H. LU, J. ZHANG, S. SUN, L. LING, N. ZHANG, G. LI, AND R. CHEN, *Topological structure analysis of the protein–protein interaction network in budding yeast*, Nucleic Acids Research, 31 (2003), pp. 2443–2450, https://doi.org/10.1093/nar/gkg340. 16

[7] T. CHEN AND E. HALLMAN, *Krylov-aware stochastic trace estimation*, May 2022, https://arxiv.org/abs/2205.01736. 9

[8] T. CHEN, T. TROGDON, AND S. UBARU, *Randomized matrix-free quadrature for spectrum and spectral sum approximation*, Apr. 2022, https://arxiv.org/abs/2204.01941. 9

[9] T. DAVIS AND Y. HU, *The University of Florida sparse matrix collection*, ACM Transactions on Mathematical Software, 38 (2011), pp. 1–25, https://doi.org/10.1145/2049662.2049663. 16

[10] E. N. EPPERLY AND J. A. TROPP, *Jackknife variability estimation for randomized matrix computations*, July 2022, https://arxiv.org/abs/2207.06342v2. 9

[11] E. ESTRADA, *The many facets of the Estrada indices of graphs and networks*, SeMA Journal, 79 (2022), pp. 57–125, https://doi.org/10.1007/s40324-021-00275-w. 15

[12] A. S. GAMBHIR, A. STATHOPOULOS, AND K. ORGINOS, *Deflation as a method of variance reduction for estimating the trace of a matrix inverse*, SIAM Journal on Scientific Computing, 39 (2017), pp. A532–A558, https://doi.org/10.1137/16M1066361. 2, 8

[13] A. C. GILBERT, M. J. STRAUSS, J. A. TROPP, AND R. VERSHYNIN, *One sketch for all: Fast algorithms for compressed sensing*, in Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing, June 2007, pp. 237–246, https://doi.org/10.1145/1250790.1250824. 19

[14] A. GIRARD, *A fast"Monte-Carlo cross-validation" procedure for large least squares problems with noisy data*, Numerische Mathematik, 56 (1989), pp. 1–23, https://doi.org/10.1007/BF01395775. 1, 2, 8

[15] N. HALKO, P.-G. MARTINSSON, AND J. A. TROPP, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Review, 53 (2011), pp. 217–288, https://doi.org/10.1137/090771806. 1, 3

[16] P. R. HALMOS, *The theory of unbiased estimation*, The Annals of Mathematical Statistics, 17 (1946), pp. 34–43, https://doi.org/10.1214/aoms/1177731020. 2

[17] N. J. HIGHAM, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, 2008, https://doi.org/10.1137/1.9780898717778. 9

[18] N. J. HIGHAM, *Matrix exponential times a vector*, Nov. 2010. Available at https://www.mathworks.com/matlabcentral/fileexchange/29576-matrix-exponential-times-a-vector (accessed 10/25/2022). 6

[19] M. F. HUTCHINSON, *A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines*, Communications in Statistics - Simulation and Computation, 18 (1989), pp. 1059–1076, https://doi.org/10.1080/03610918908812806. 1, 2, 8

[20] S. JIANG, H. PHAM, D. P. WOODRUFF, AND Q. ZHANG, *Optimal sketching for trace estimation*, in 35th Conference on Neural Information Processing Systems., 2021, p. 13. 8

[21] V. S. KOROLJUK AND Y. V. BOROVSKICH, *Theory of U-Statistics*, Springer Netherlands, Dordrecht, 1994, https://doi.org/10.1007/978-94-017-3515-5. 2

[22] H. LI, G. C. LINDERMAN, A. SZLAM, K. P. STANTON, Y. KLUGER, AND M. TYGERT, *Algorithm 971: An implementation of a randomized algorithm for principal component analysis*, ACM Transactions On Mathematical Software, 43 (2017), https://doi.org/10.1145/3004053. 10

[23] L. LIN, *Randomized estimation of spectral densities of large matrices made accurate*, Numerische Mathematik, 136 (2017), pp. 183–213, https://doi.org/10.1007/s00211-016-0837-7. 2, 8

[24] C. LITENS, *Transverse Field Ising Model with Different Boundary Conditions*, Bachelor's Thesis, Stockholm University, Feb. 2019. Available at http://staff.fysik.su.se/~ardonne/files/theses/bachelor-thesis_christopher-litens.pdf. 15

[25] P.-G. MARTINSSON AND J. A. TROPP, *Randomized numerical linear algebra: Foundations and algorithms*, Acta Numerica, 29 (2020), pp. 403–572, https://doi.org/10.1017/S0962492920000021. 2, 5, 8

[26] R. A. MEYER, C. MUSCO, C. MUSCO, AND D. P. WOODRUFF, *Hutch++: Optimal stochastic trace estimation*, in Symposium on Simplicity in Algorithms, SIAM, Jan. 2021. 1, 2, 3, 8

[27] R. A. MEYER, C. MUSCO, C. MUSCO, AND D. P. WOODRUFF, *Hutch++: Optimal Stochastic Trace Estimation*, June 2021, https://arxiv.org/abs/2002.11457. 18, 19

[28] D. PERSSON, A. CORTINOVIS, AND D. KRESSNER, *Improved variants of the Hutch++ algorithm for trace estimation*, SIAM Journal on Matrix Analysis and Applications, (2022), pp. 1162–1185, https://doi.org/10.1137/21M1447623. 5, 8, 12, 23

[29] P. PFEUTY, *The one-dimensional Ising model with a transverse field*, Annals of Physics, 57 (1970), pp. 79–90, https://doi.org/10.1016/0003-4916(70)90270-8. 15

[30] V. ROKHLIN AND M. TYGERT, *A fast randomized algorithm for overdetermined linear least-squares regression*, Proceedings of the National Academy of Sciences, 105 (2008), pp. 13212–13217, https://doi.org/10.1073/pnas.0804869105. 1

[31] A. K. SAIBABA, A. ALEXANDERIAN, AND I. C. F. IPSEN, *Randomized matrix-free trace and log-determinant estimators*, Numerische Mathematik, 137 (2017), pp. 353–395, https://doi.org/10.1007/s00211-017-0880-z. 2, 8, 12

[32] J. A. TROPP AND R. J. WEBBER, *Randomized algorithms for low-rank matrix approximation: Design, analysis, and applictions*, Manuscript in preparation, (2023). 19

[33] J. A. TROPP, A. YURTSEVER, M. UDELL, AND V. CEVHER, *Fixed-rank approximation of a positive-semidefinite matrix from streaming data*, in Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 1225–1234. 10

[34] S. UBARU, J. CHEN, AND Y. SAAD, *Fast estimation of* $\mathrm{tr}(f(A))$ *via stochastic Lanczos quadrature*, SIAM Journal on Matrix Analysis and Applications, 38 (2017), pp. 1075–1099, https://doi.org/10.1137/16M1104974. 8

[35] S. UBARU AND Y. SAAD, *Applications of trace estimation techniques*, in High Performance Computing in Science and Engineering, 2017, https://doi.org/10.1007/978-3-319-97136-0_2. 2

SUPPLEMENTARY MATERIAL

SM1. Comparison with Adaptive HUTCH++. The adaptive HUTCH++ algorithm of [28] flexibly apportions test vectors between low-rank approximation and residual trace estimation, based on the estimated singular values of the input matrix. This algorithm interpolates between the Girard–Hutchinson estimator, the HUTCH++ estimator, and a purely low-rank approximation-based trace estimate. This algorithm also estimates the total number of matvecs m to meet an (absolute) error tolerance:

$$\left|\widehat{\mathrm{tr}} - \mathrm{tr}(\boldsymbol{A})\right| \leq \varepsilon_{\mathrm{abs}} \quad \text{except with probability } 1 - 2\delta,$$

where δ is a parameter chosen by the user.

We compare XTRACE with adaptive HUTCH++ in Figure SM1. To produce these plots, we first run adaptive HUTCH++ with error tolerances $\varepsilon_{\mathrm{abs}} = \varepsilon \cdot \mathrm{tr}\, \boldsymbol{A}$ and failure probability parameter $\delta := 1/10$. For each value of the error tolerance ε, we plot the mean error (solid line) and desired accuracy ε against the mean number m of matvecs required by the algorithm. We then run XTRACE with a variable number of matvecs m and report the mean error over 1000 trials. Overall, we find that XTRACE performs similarly adaptive HUTCH++ or better than HUTCH++ by up to an order of magnitude.

SM2. MATLAB implementations. We present MATLAB R2022b implementations of XTRACE, XNYSTRACE, and XDIAG in Programs SM2.1 to SM2.3.

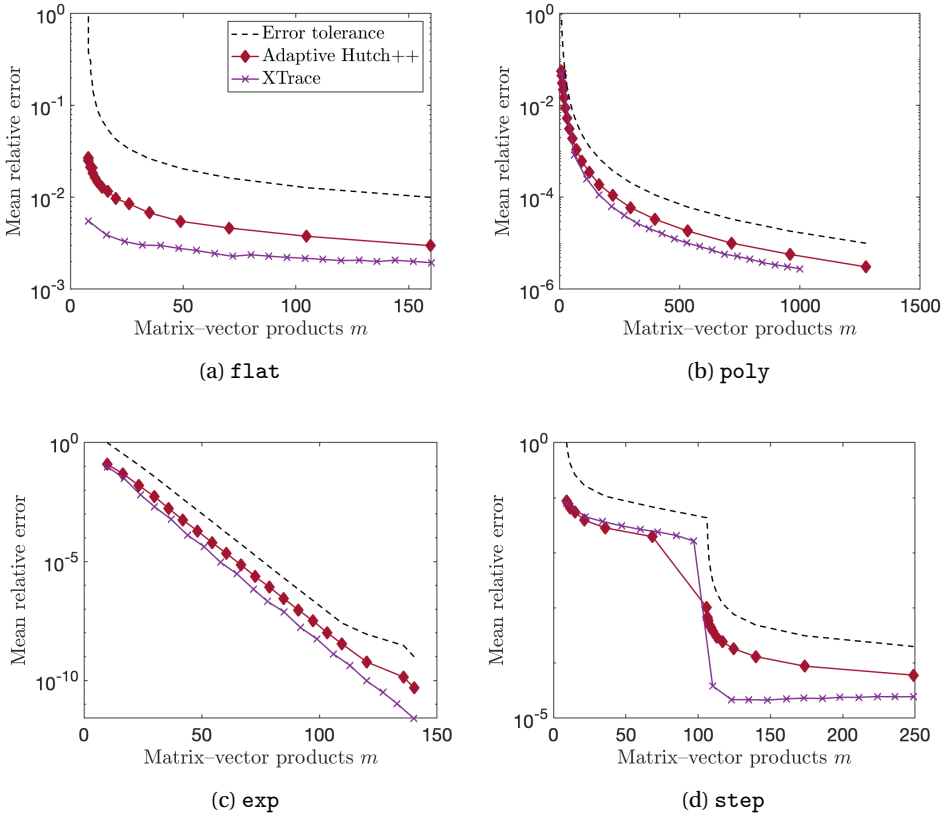


Fig. SM1: **Comparison with adaptive HUTCH++.** Average relative error using XTRACE and adaptive HUTCH++ (solid lines) and user-chosen error tolerance for Adaptive HUTCH++ (dashed lines). See section SM1.

```
function [t,err] = xtrace(A, m)
N = size(A,1); m = floor(m/2);

%% Helper functions
cnormc = @(M) M ./ vecnorm(M,2,1);
diag_prod = @(A,B) sum(conj(A).*B,1).'; % computes diag(A'*B)

%% Randomized SVD
Om = sqrt(N)*cnormc(randn(N, m));
Y = A*Om;
[Q,R] = qr(Y,0);

%% Normalization
W = Q'*Om;
S = cnormc(inv(R)');
scale = (N - m + 1) ./ (N - (vecnorm(W)').^2 ...
        + abs(diag_prod(S,W) .* vecnorm(S)').^2);

%% Quantities needed for trace estimation
Z = A*Q; H = Q'*Z; HW = H*W; T=Z'*Om;
dSW = diag_prod(S, W); dSHS = diag_prod(S, H*S);
dTW = diag_prod(T, W); dWHW = diag_prod(W, HW);
dSRmHW = diag_prod(S, R-HW); dTmHRS = diag_prod(T-H'*W,S);

%% Trace estimate
ests = trace(H)*ones(m,1) - dSHS + (dWHW - dTW + dTmHRS .* dSW...
    + abs(dSW).^2 .* dSHS + conj(dSW) .* dSRmHW) .* scale;
t = mean(ests);
err = std(ests)/sqrt(m);
end
```

Program SM2.1: MATLAB 2022b implementation for XTRACE

```
function [t,err] = xnystrace(A, m)
N = size(A,1);

%% Helper functions
cnormc = @(M) M ./ vecnorm(M,2,1);
diag_prod = @(A,B) sum(conj(A).*B,1).'; % computes diag(A'*B)

%% Nystrom
Om = sqrt(N) * cnormc(randn(N,m));
Y = A*Om;
nu = eps*norm(Y,'fro')/sqrt(N);
Y = Y + nu*Om; % Shift for numerical stability
[Q,R] = qr(Y,0);
H = Om'*Y; C = chol((H+H')/2);
B = R/C; % Nystrom approx is Q*B*B'*Q'

%% Normalization
[QQ,RR] = qr(Om,0);
WW = QQ'*Om;
SS = cnormc(inv(RR)');
scale = (N - m + 1) ./ (N - vecnorm(WW).^2 ...
    + abs(diag_prod(SS,WW)' .* vecnorm(SS)).^2);

%% Trace estimate
W = Q'*Om; S = (B/C') .* (diag(inv(H))').^(-0.5);
dSW = diag_prod(S, W).';
ests = norm(B,'fro')^2 - vecnorm(S).^2 + abs(dSW).^2 .* scale...
    - nu*N;
t = mean(ests);
err = std(ests)/sqrt(m);
end
```

Program SM2.2: MATLAB 2022b implementation for XNYSTRACE

```
function d = xdiag(A, m)
N = size(A,1); m = floor(m/2);

%% Helper functions
cnormc = @(M) M ./ vecnorm(M,2,1);
diag_prod = @(A,B) sum(conj(A).*B,1).'; % computes diag(A'*B)

%% Randomized SVD
Om = -3 + 2*randi(2,n,m); % Random signs
Y = A*Om;
[Q,R] = qr(Y,0);

% Quantities needed for trace estimation
Z = A'*Q; T=Z'*Om;
S = cnormc(inv(R)');
dQZ = diag_prod(Q',Z'); dQSSZ = diag_prod((Q*S)',(Z*S)');
dOmQT = diag_prod(Om',(Q*T).'); dOmY = diag_prod(Om',Y.');
dOmQSST = diag_prod(Om',(Q*S*diag(diag_prod(S,T))).');

% Trace estimate
d = dQZ + (-dQSSZ+dOmY-dOmQT+dOmQSST)/m;
end
```

Program SM2.3: MATLAB 2022b implementation for XDIAG