Computationally Efficient Approximations for Matrix-Based Rényi's Entropy

Tieliang Gong, Yuxin Dong, Shujian Yu, and Bo Dong

***Abstract*—The recently developed matrix-based Rényi's α-order entropy enables measurement of information in data simply using the eigenspectrum of symmetric positive semi-definite (PSD) matrices in reproducing kernel Hilbert space, without estimation of the underlying data distribution. This intriguing property makes this new information measurement widely adopted in multiple statistical inference and learning tasks. However, the computation of such quantity involves the trace operator on a PSD matrix G to power α (i.e., $\mathrm{tr}(G^\alpha)$), with a normal complexity of nearly $\mathcal{O}(n^3)$, which severely hampers its practical usage when the number of samples (i.e., n) is large. In this work, we present computationally efficient approximations to this new entropy functional that can reduce its complexity to even significantly less than $\mathcal{O}(n^2)$. To this end, we leverage the recent progress on Randomized Numerical Linear Algebra, developing Taylor, Chebyshev and Lanczos approximations to $\mathrm{tr}(G^\alpha)$ for arbitrary values of α by converting it into a matrix-vector multiplication problem. We also establish the connection between the matrix-based Rényi's entropy and PSD matrix approximation, which enables exploiting both clustering and block low-rank structure of G to further reduce the computational cost. We theoretically provide approximation accuracy guarantees and illustrate the properties for different approximations. Large-scale experimental evaluations on both synthetic and real-world data corroborate our theoretical findings, showing promising speedup with negligible loss in accuracy.**

***Index Terms*—Matrix-based Rényi's entropy, randomized numerical linear algebra, trace estimation, information bottleneck, mutual information.**

Manuscript received 21 June 2022; revised 1 November 2022; accepted 27 December 2022. Date of publication 4 January 2023; date of current version 11 January 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ali Tajer. This work was supported in part by the Key Research and Development Program of China under Grant 2021ZD0110700, in part by the The National Natural Science Foundation of China under Grant 62106191, in part by the The Key Research and Development Program of Shaanxi Province under Grant 2021GXLH-Z-095, in part by The Innovative Research Group of the National Natural Science Foundation of China under Grant 61721002, in part by The Consulting Research Project of the Chinese Academy of Engineering (The Online and Offline Mixed Educational Service System for The Belt and Road Training in MOOC China), in part by the Project of China Knowledge Centre for Engineering Science and Technology, and in part by The innovation team from the Ministry of Education under Grant IRT_17R86. *(Tieliang Gong and Yuxin Dong contributed equally to this work.) (Corresponding author: Tieliang Gong.)*

Tieliang Gong, Yuxin Dong, and Bo Dong are with the School of Computer Science and Technology, Xi'an Jiaotong University and Shaanxi Provincial Key Laboratory of Big Data Knowledge Engineering, Xi'an 710049, China (e-mail: adidasgtl@gmail.com; dongyuxin@stu.xjtu.edu.cn; dong.bo@mail.xjtu.edu.cn).

Shujian Yu is with the Machine Learning Group, UiT - The Arctic University of Norway, 9019 Tromsø, Norway, and also with the Department of Computer Science, Vrije Universiteit Amsterdam, 1081 HV Amsterdam, The Netherlands (e-mail: yusj9011@gmail.com).

Digital Object Identifier 10.1109/TSP.2022.3233724

I. Introduction

Information-theoretic measures (such as entropy, mutual information, synergy) and principles (such as information bottleneck [1] and maximum entropy [2], [3]) have a long track record of usefulness in machine learning and neuroscience [4], [5], [6]. Notable examples include feature selection by maximizing the mutual information between selected feature subset and class labels [7] and the utilization of partial information decomposition (PID) framework [8] to quantify the neural goal functions [9]. In the deep learning area, information-theoretic methods have become the workhorse of several impressive deep learning breakthroughs in recent years, ranging from the representation learning by variational information bottleneck [10] or information maximization (InfoMax) [11] to theoretical investigations on the generalization bound [12].

The combination of information theory with machine learning and neuroscience usually involves an exact access to different information-theoretic measures defined over the probability space. However, the accurate probability density function (PDF) estimation is computationally infeasible in high-dimensional space [13], which impedes more widespread adoption of information-theoretic methods in data-driven science.

In [14], [15] and later [16], a novel information measure named the matrix-based Rényi's α-order entropy was proposed, enabling us to quantify the information of a single variable or complex interactions across multiple variables directly from given samples. Distinct from the classical Shannon entropy [17] and Rényi's entropy [18], this new family of information measures is defined on the eigenspectrum of a Gram matrix constructed by projecting data in reproducing kernel Hilbert space (RKHS), thus avoiding the expensive evaluation of underlying distributions. This elegant property makes the matrix-based entropy functional being a reliable choice in lots of machine learning and neuroscience applications, which include but are not limited to similarity measurement [19], feature reduction [20], compressing deep neural networks by neuron pruning [21], identifying the most informative regions in the brain (i.e., subgraph) to psychiatric disorders (such as depression and autism) [22], inferring the effective connectivity across different brain areas [23].

Despite of its practical utility, exactly computing matrix-based Rényi's entropy involves calculating eigenvalues of a Gram matrix G of size $n \times n$ (n is the sample size, please refer to Definition 1), which requires $\mathcal{O}(n^3)$ computational complexity with traditional techniques such as Singular Value

Decomposition (SVD) [24], [25], CUR decomposition [26], [27] or QR Factorization [28], [29]. This drawback poses great challenges for both storage and computing in practice. We therefore seek for computationally efficient methods with statistical guarantees to approximate the matrix-based Rényi's entropy. To this end, we borrow the idea of stochastic trace estimation [30] for integer α-order entropy approximation, where the trace estimation problem is transferred into matrix-vector multiplications. We then employ polynomial approximation techniques to estimate the power of G for fractional-order α. Furthermore, we establish the connection between matrix-based Rényi's entropy estimation and Gram matrix approximation, where the k-means algorithm is employed to discover the block structure of G and a low-rank decomposition is conducted to approximate the off-diagonal blocks. In summary, our contributions are four-fold:

- We develop Taylor, Chebyshev and Lanczos approximations for arbitrary α-order (integer & fractional) matrix-based Rényi's entropy that transforms the original trace estimation problem into matrix-vector multiplications, reducing the overall complexity to roughly $\mathcal{O}(n^2 s)$, where $s \ll n$ is the number of queried random vector.
- We additionally reveal the intrinsic connection between eigenspectrum estimation and Gram matrix approximation. To this end, we investigate the spectral structure of the Gram matrix and design a novel block low-rank approximation method, which enables us to further reduce the computational burden to $\mathcal{O}(n^2/c + nck)$, where $c \ll n$ is the number of clusters and $k \ll n$ is the order of rank.
- Theoretically, we conduct statistical analysis for each of the estimators above, establishing accuracy guarantees and revealing their intrinsic properties and connections.
- We evaluate the effectiveness of the proposed approximations on large-scale simulation and real-world datasets, showing remarkable acceleration on various information-related tasks with no significant loss in accuracy.

II. Preliminaries

Entropy measures the uncertainty of random variables using a single scalar quantity [31]. For a continuous random variable X with PDF $p(\mathbf{x})$ defined on the finite set $\mathcal{X}$, the α-order Rényi's entropy ($\alpha > 0$ and $\alpha \neq 1$) $\mathbf{H}_\alpha(X)$ is defined as

$$\mathbf{H}_\alpha(X) = \frac{1}{1-\alpha} \log \int_{\mathcal{X}} p^\alpha(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{1}$$

The limit case of $\alpha \to 1$ yields the well known Shannon's entropy, i.e. $\mathbf{H}(X) = -\int_{\mathcal{X}} p(\mathbf{x}) \log p(\mathbf{x})\, \mathrm{d}\mathbf{x}$. It is easy to see that Rényi's entropy is a natural extension of Shannon entropy by introducing a hyper-parameter α. In real-world applications, the choice of α is task-specific: on the one hand, α should be less than 2 or even 1 when the learning task requires estimating tails of the distribution or multiple modalities; on the other hand, α is suggested to be greater than 2 to emphasize mode behavior when we aim to characterize the mean behavior [4].

It is worth noting that calculating Rényi α-order entropy requires the prior about the PDF of data, which prevents its more widespread adoption in data-driven science. To alleviate this issue, an alternative measure, namely the matrix-based Rényi's α-order entropy was recently proposed, which resembles quantum Rényi's entropy in terms of the eigenspectrum of a normalized Hermitian matrix constructed by projecting data in RKHS. It is defined as:

Definition 1: [14] Let $\varphi : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ be a real valued positive kernel that is also infinitely divisible [32]. Given $\{\mathbf{x}_i\}_{i=1}^n \subset \mathcal{X}$, each $\mathbf{x}_i$ being a real-valued scalar or vector, and the Gram matrix K obtained from $K_{ij} = \varphi(\mathbf{x}_i, \mathbf{x}_j)$, a matrix-based analogue to Rényi's α-entropy can be defined as:

$$\mathbf{S}_\alpha(G) = \frac{1}{1-\alpha} \log_2(\mathrm{tr}(G^\alpha)) = \frac{1}{1-\alpha} \log_2 \left[\sum_{i=1}^n \lambda_i^\alpha(G) \right], \tag{2}$$

where $G_{ij} = \frac{1}{n} \frac{K_{ij}}{\sqrt{K_{ii} K_{jj}}}$ is a normalized kernel matrix and $\lambda_i(G)$ denotes the i-th eigenvalue of G.

Let Δ_n^+ be the set of positive semi-definite matrices of size $n \times n$ whose elements take real values. From [14] we know that $\mathbf{S}_\alpha(PGP^*) = \mathbf{S}_\alpha(G)$ for any orthonormal matrix P, and that $\mathbf{S}_\alpha(\frac{1}{n}I) = \log_2(n)$ for identity matrix $I \in \Delta_n^+$ takes the maximum entropy value among all $n \times n$ kernel matrices. We denote the maximum and minimum eigenvalue of G as $\lambda_{\max}$ and $\lambda_{\min}$ respectively, the condition number is then $\kappa = \lambda_{\max}/\lambda_{\min}$.

Definition 2: [16] Given a collection of n samples $\{\mathbf{s}_i = (\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^L)\}_{i=1}^n$, each sample contains L ($L \geq 2$) measurements $\mathbf{x}^1 \in \mathcal{X}^1$, $\mathbf{x}^2 \in \mathcal{X}^2$, $\cdots$, $\mathbf{x}^L \in \mathcal{X}^L$ obtained from the same realization, and the positive definite kernels $\varphi_1 : \mathcal{X}^1 \times \mathcal{X}^1 \mapsto \mathbb{R}$, $\varphi_2 : \mathcal{X}^2 \times \mathcal{X}^2 \mapsto \mathbb{R}$, $\cdots$, $\varphi_L : \mathcal{X}^L \times \mathcal{X}^L \mapsto \mathbb{R}$, a matrix-based analogue to Rényi's α-order joint-entropy among L variables can be defined as:

$$\mathbf{S}_\alpha(G^1, G^2, \ldots, G^L) = \mathbf{S}_\alpha \left(\frac{G^1 \circ G^2 \circ \cdots \circ G^L}{\mathrm{tr}(G^1 \circ G^2 \circ \cdots \circ G^L)} \right), \tag{3}$$

where $(G^1)_{ij} = \varphi_1(\mathbf{x}_i^1, \mathbf{x}_j^1)$, $(G^2)_{ij} = \varphi_2(\mathbf{x}_i^2, \mathbf{x}_j^2)$, $\cdots$, $(G^L)_{ij} = \varphi_L(\mathbf{x}_i^L, \mathbf{x}_j^L)$, and $\circ$ denotes the Hadamard product.

Given Eq. (2) and (3), the matrix-based Rényi's α-order mutual information $I_\alpha(\cdot;\cdot)$ and total correlation $T_\alpha(\cdot)$ amongst $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^L$ and $\mathbf{y}$ in analogy of Shannon's definitions can be derived accordingly:

$$\begin{aligned} \mathbf{I}_\alpha(G^1, \ldots, G^L; G) = \ & \mathbf{S}_\alpha(G^1, \ldots, G^L) + \mathbf{S}_\alpha(G) \\ & - \mathbf{S}_\alpha(G^1, \ldots, G^L, G). \end{aligned} \tag{4}$$

$$\mathbf{T}_\alpha(G^1, \ldots, G^L) = \left[\sum_{i=1}^L \mathbf{S}_\alpha(G^i) \right] - \mathbf{S}_\alpha(G^1, \ldots, G^L). \tag{5}$$

It is simple to verify that both of them are permutation-invariant to the ordering of kernel matrices $G^1, \ldots, G^L$. Furthermore, the matrix-based entropy functionals are independent of the specific dimensions of variables $\mathbf{x}^1, \ldots, \mathbf{x}^L$, thus avoiding estimation of the underlying data distributions and making them suitable to be applied on high-dimensional data with either discrete or continuous distributions.

Algorithm 1: Integer Order Matrix-Based Rényi's Entropy Estimation.

```
1: Input: Kernel matrix $G \in \mathbb{R}^{n\times n}$, number of random
   vectors $s$, integer order $\alpha \geq 2$.
2: Output: Approximation to $S_\alpha(G)$.
3: Generate $s$ independent random standard Gaussian
   vectors $\mathbf{g}_i, i = 1, \ldots, s$.
4: Return: $\tilde{\mathbf{S}}_\alpha(G) = \frac{1}{1-\alpha}\log_2(\frac{1}{s}\sum_{i=1}^{s}\mathbf{g}_i^\top G^\alpha \mathbf{g}_i)$.
```

Algorithm 2: Fractional Order Matrix-Based Rényi's Entropy Estimation via Taylor Series.

```
1: Input: Kernel matrix $G \in \mathbb{R}^{n\times n}$, number of random
   vectors $s$, fractional order $\alpha > 0$, polynomial degree
   $m > \alpha$.
2: Output: Approximation to $S_\alpha(G)$.
3: Calculate $\lambda_{\max}$ by power iteration.
4: Generate $s$ independent random standard Gaussian
   vectors $\mathbf{g}_i, i = 1, \ldots, s$.
5: Return: $\tilde{\mathbf{S}}_\alpha(G) = \frac{1}{1-\alpha}\cdot$
   $\log_2\left(\frac{\lambda_{\max}^\alpha}{s}\sum_{i=1}^{s}\sum_{n=0}^{m}\binom{\alpha}{n}\mathbf{g}_i^\top(\frac{G}{\lambda_{\max}} - I_n)^n \mathbf{g}_i\right)$.
```

III. Approximating Matrix-Based Rényi's Entropy: Randomized Linear Algebra Approaches

In this section, we present computationally efficient approaches with statistical guarantees for matrix-based Rényi's entropy approximation from the perspective of randomized numerical linear algebra, where the general idea is to approximate the trace operator by random matrix-vector multiplications to avoid the expensive eigenvalue decomposition.

A. Randomized Approximation

Inspired by the work on randomized trace estimation [30], we adopt random algorithms for calculating the matrix-based Rényi's entropy. The following lemma [33] characterizes an algorithm that approximates the trace of any symmetric positive semi-definite (PSD) matrix by computing inner products of the matrix with Gaussian random vectors:

Lemma 1: [30] Let $G \in \mathbb{R}^{n\times n}$ be a symmetric positive semi-definite matrix. If $\mathbf{g}_1, \mathbf{g}_2, \cdots, \mathbf{g}_s \in \mathbb{R}^n$ are i.i.d random standard Gaussian vectors, then for $s = \lceil 8\ln(2/\delta)/\epsilon^2 \rceil$, with probability at least $1-\delta$,

$$\left| \mathrm{tr}(G) - \frac{1}{s}\sum_{i=1}^{s}\mathbf{g}_i^\top G \mathbf{g}_i \right| \leq \epsilon \cdot \mathrm{tr}(G).$$

Lemma 1 immediately implies Algorithm 1 for integer order matrix-based Rényi's entropy estimation. With conventional eigenvalue-based approaches, exactly calculating the matrix-based Rényi's entropy generally requires $\mathcal{O}(n^3)$ time complexity. Algorithm 1 successfully transforms the eigenvalue problem into matrix-vector multiplications whose complexity is $\mathcal{O}(\alpha s \cdot \mathbf{nnz}(G))$, where $s \ll n$ is the number of random vector queries and $\mathbf{nnz}(G)$ denotes the number of non-zero elements in G ($\mathbf{nnz}(G) \approx n^2$ for dense matrix G). Theorem 1 provides the quality-of-approximation result for Algorithm 1.

Theorem 1: Let $G \in \mathbb{R}^{n\times n}$ be a normalized PSD kernel matrix and $\tilde{\mathbf{S}}_\alpha(G)$ be the output of Algorithm 1 with $s = \lceil 8\ln(2/\delta)/\epsilon^2 \rceil$. Then with confidence at least $1-\delta$,

$$\left| \mathbf{S}_\alpha(G) - \tilde{\mathbf{S}}_\alpha(G) \right| \leq \left| \frac{1}{1-\alpha}\log_2(1-\epsilon) \right|.$$

Proof: Let $\tilde{\mathrm{tr}}(G^\alpha) = \frac{1}{s}\sum_{i=1}^{s}\mathbf{g}_i^\top G^\alpha \mathbf{g}_i$, then

$$\begin{aligned}\left| \mathbf{S}_\alpha(G) - \tilde{\mathbf{S}}_\alpha(G) \right| &= \left| \frac{1}{1-\alpha}\log_2\left(\frac{\tilde{\mathrm{tr}}(G^\alpha)}{\mathrm{tr}(G^\alpha)}\right) \right| \\ &= \left| \frac{1}{1-\alpha}\log_2\left(1 - \frac{\mathrm{tr}(G^\alpha) - \tilde{\mathrm{tr}}(G^\alpha)}{\mathrm{tr}(G^\alpha)}\right) \right| \\ &\leq \left| \frac{1}{1-\alpha}\log_2\left(1 - \left|\frac{\mathrm{tr}(G^\alpha) - \tilde{\mathrm{tr}}(G^\alpha)}{\mathrm{tr}(G^\alpha)}\right|\right) \right| \\ &\leq \left| \frac{1}{1-\alpha}\log_2(1-\epsilon) \right|.\end{aligned}$$

Where the first inequality follows by the fact that $|\log_2(1-x)| \leq -\log_2(1-|x|)$ for all $x \in (-1, 1)$, and the second inequality follows by Lemma 1. ■

Remark 1: Theorem 1 theoretically proves the feasibility of approximating matrix-based Rényi's entropy through matrix-vector multiplication operations, which leads to substantially lower computational cost than eigenvalue-based methods. The approximation quality depends on the number of queried random vectors s and hyper-parameter α. In particular, with the increase of α, the approximation error decreases for integer $\alpha \geq 2$.

B. Taylor Series Approximation

The above approach could only handle integer α-orders. For fractional α cases, the result of matrix-vector multiplications $G^\alpha \cdot \mathbf{g}_i$ cannot be directly acquired in the same manner. To address this issue, we apply a Taylor series expansion on the matrix α-power functional, where the implementing details are presented in Algorithm 2. It requires extra estimation of $\lambda_{\max}$, the dominant eigenvalue of G. Here, we adopt power iteration [34], an estimator for extreme matrix eigenvalues via random vector queries. As a result, matrix-vector multiplication is still the only operator that directly accesses the kernel matrix G in our algorithms, enabling further optimizations with matrix approximation techniques. The total computational complexity of Algorithm 2 is $\mathcal{O}(ms \cdot \mathbf{nnz}(G))$, where m is the degree of the Taylor series.

Theorem 2: Let $G \in \mathbb{R}^{n\times n}$ be a normalized PSD kernel matrix and $\tilde{\mathbf{S}}_\alpha(G)$ be the output of Algorithm 2 with $s = \lceil 8\ln(2/\delta)/\epsilon^2 \rceil$ and

$$m = \left\lceil \alpha + \begin{cases} W_0\left(\kappa\beta \sqrt[\alpha+1]{\frac{\Gamma(\alpha+1)}{\epsilon\pi}}\right) \Big/ \beta & \text{if } \lambda_{\min} > 0 \\ \sqrt[\alpha]{\frac{n\Gamma(\alpha+1)}{\epsilon\pi}} & \text{if } \lambda_{\min} = 0 \end{cases} \right\rceil. \tag{6}$$

Then with confidence at least $1-\delta$,

$$\left|\mathbf{S}_\alpha(G) - \tilde{\mathbf{S}}_\alpha(G)\right| \leq \left|\frac{1}{1-\alpha}\log_2(1-3\epsilon)\right|,$$

where $\beta = -\log(1-1/\kappa)/(\alpha+1)$ and W_0 is the principal branch of Lambert W function.

Proof: In Algorithm 2, we use binomial series, the Taylor expansion for function $f(x) = (x+1)^\alpha$:

$$(1+x)^\alpha = \sum_{i=0}^{\infty} \binom{\alpha}{i} x^i. \tag{7}$$

Binomial series converge absolutely for symmetric matrices G with eigenvalues in $[-1,1)$ and $\alpha > 0$. Taking $\lambda_{\max}$, the maximum eigenvalue of G, the eigenvalues of $\frac{G}{\lambda_{\max}} - I$ are within $[-1,0]$, which means the series above converge absolutely. Therefore, for each eigenvalue λ of G:

$$\lambda^\alpha = \lambda_{\max}^\alpha \cdot \left(\frac{\lambda}{\lambda_{\max}}\right)^\alpha = \lambda_{\max}^\alpha \sum_{i=0}^{\infty} \binom{\alpha}{i} \left(\frac{\lambda}{\lambda_{\max}} - 1\right)^i. \tag{8}$$

Let $f_m(\lambda) = \lambda_{\max}^\alpha \sum_{i=0}^{m} \binom{\alpha}{i} (\lambda/\lambda_{\max} - 1)^i$ denote the Taylor series above with degree m. When $\lambda_{\min} > 0$, we have

$$\begin{aligned} |\lambda^\alpha - f_m(\lambda)| &= \lambda_{\max}^\alpha \left| \sum_{i=m+1}^{\infty} \binom{\alpha}{i} \left(\frac{\lambda}{\lambda_{\max}} - 1\right)^i \right| \\ &\leq \lambda_{\max}^\alpha \left| \binom{\alpha}{m+1} \sum_{i=m+1}^{\infty} \left(\frac{\lambda}{\lambda_{\max}} - 1\right)^i \right| \\ &= \lambda_{\max}^\alpha \left| \frac{\lambda_{\max}}{\lambda} \binom{\alpha}{m+1} \left(\frac{\lambda}{\lambda_{\max}} - 1\right)^{m+1} \right| \\ &\leq \lambda_{\max}^\alpha \kappa \left| \binom{\alpha}{m+1} \left(\frac{1}{\kappa} - 1\right)^{m+1} \right|. \end{aligned} \tag{9}$$

Recall that the binomial term satisfies $\binom{\alpha}{i} = \prod_{j=1}^{i} \frac{\alpha-j+1}{j}$. When $i > \alpha$, $\binom{\alpha}{i+1} = \binom{\alpha}{i}\frac{\alpha-i}{i+1}$, which means $\binom{\alpha}{i+1} \cdot \binom{\alpha}{i} \leq 0$. Combining with the fact that $\lambda/\lambda_{\max} - 1 \leq 0$, we know that the coefficients $\binom{\alpha}{i}(\frac{\lambda}{\lambda_{\max}} - 1)^i$ above share the same sign for $i > \alpha$. By assuming $m > \alpha$, (9) follows by noticing that $|\binom{\alpha}{i}| = |\binom{\alpha}{m+1} \prod_{j=m+2}^{i} \frac{\alpha-j+1}{j}| \leq |\binom{\alpha}{m+1}|$, $\forall i \geq m+1$, since $|\frac{\alpha-j+1}{j}| \leq 1$ when $j \geq \alpha + 1$.

To upper bound the binomial term $\binom{\alpha}{m+1}$, we introduce the following lemma:

Lemma 2: [35] Let $\Gamma(x)$ be the gamma function and let $R(x,y) = \Gamma(x+y)/\Gamma(x)$, then

$$R(x,y) \geq x^y \qquad \text{for } 1 \leq y \leq 2,$$

$$R(x,y) \geq x(x+1)^{y-1} \qquad \text{for } y \geq 2.$$

Then we have the following upper bound:

$$\begin{aligned} \left|\binom{\alpha}{m+1}\right| &= \left|\frac{\Gamma(\alpha+1)}{\Gamma(m+2)\Gamma(\alpha-m)}\right| \\ &\leq \left|\frac{\Gamma(\alpha+1)}{\Gamma(m-\alpha+1)\Gamma(\alpha-m)}\right| \end{aligned}$$

$$\cdot \begin{cases} \frac{1}{(m-\alpha+1)^{\alpha+1}} & 0 < \alpha < 1 \\ \frac{1}{(m-\alpha+1)(m-\alpha+2)^\alpha} & \alpha > 1 \end{cases} \tag{10}$$

$$\leq \frac{\Gamma(\alpha+1)}{\pi(m-\alpha+1)} \frac{1}{(m-\alpha+1)^\alpha}$$

$$\leq \frac{\Gamma(\alpha+1)}{\pi(m-\alpha+1)^{\alpha+1}}. \tag{11}$$

(10) follows by applying Lemma 2 on $R(m-\alpha+1, \alpha+1)$. (11) follows by Euler's reflection formula that for any non-integer number z, $\Gamma(z)\Gamma(1-z) = \pi/\sin(\pi z)$. By limiting the choice of m, we would like to ensure that

$$\begin{aligned} |\lambda^\alpha - f_m(\lambda)| &\leq \lambda_{\max}^\alpha \kappa \left| \frac{\Gamma(\alpha+1)}{\pi(m-\alpha+1)^{\alpha+1}} \left(\frac{1}{\kappa} - 1\right)^{m+1} \right| \\ &\leq \epsilon \lambda_{\min}^\alpha \leq \epsilon \lambda^\alpha. \end{aligned}$$

Solving the inequation above yields:

$$m \geq \alpha + W_0\left(\kappa\beta \sqrt[\alpha+1]{\frac{\Gamma(\alpha+1)}{\epsilon\pi}}\right) \Big/ \beta\,,$$

where $\beta = -\log(1-1/\kappa)/(\alpha+1)$ and W_0 is the principal branch of Lambert W function. Combining the results above, we have that with confidence at least $1-\delta$:

$$\left|\sum_{i=1}^{s} \mathbf{g}_i^\top f_m(G)\mathbf{g}_i - \mathbf{tr}(G^\alpha)\right|$$

$$\leq \left|\sum_{i=1}^{s} \mathbf{g}_i^\top f_m(G)\mathbf{g}_i - \mathbf{tr}(f_m(G))\right| + |\mathbf{tr}(f_m(G)) - \mathbf{tr}(G^\alpha)|$$

$$\leq \epsilon \cdot \mathbf{tr}(f_m(G)) + |\mathbf{tr}(f_m(G)) - \mathbf{tr}(G^\alpha)| \tag{12}$$

$$\leq \epsilon \cdot \mathbf{tr}(G^\alpha) + (1+\epsilon) \cdot |\mathbf{tr}(f_m(G)) - \mathbf{tr}(G^\alpha)|$$

$$= \epsilon \cdot \mathbf{tr}(G^\alpha) + (1+\epsilon) \cdot \sum_{i=1}^{n} |\lambda_i^\alpha - f_m(\lambda_i)|$$

$$\leq \epsilon \cdot \mathbf{tr}(G^\alpha) + \epsilon(1+\epsilon) \cdot \sum_{i=1}^{n} \lambda_i^\alpha$$

$$\leq 3\epsilon \cdot \mathbf{tr}(G^\alpha), \tag{13}$$

where (12) follows by applying Lemma 1 on matrix $f_m(G)$. This finishes the proof for the case $\lambda_{\min} > 0$ by adopting similar steps in the proof of Theorem 1.

Elsewise when $\lambda_{\min} = 0$, notice that

$$\left(1 + \frac{\lambda}{\lambda_{\max}} - 1\right) \sum_{i=m}^{\infty} \binom{\alpha}{i} \left(\frac{\lambda}{\lambda_{\max}} - 1\right)^i$$

$$= \sum_{i=m}^{\infty} \binom{\alpha}{i} \left(\frac{\lambda}{\lambda_{\max}} - 1\right)^i + \sum_{i=m+1}^{\infty} \binom{\alpha}{i-1} \left(\frac{\lambda}{\lambda_{\max}} - 1\right)^i$$

$$= \binom{\alpha}{m} \left(\frac{\lambda}{\lambda_{\max}} - 1\right)^m + \sum_{i=m+1}^{\infty} \binom{\alpha+1}{i} \left(\frac{\lambda}{\lambda_{\max}} - 1\right)^i, \tag{14}$$

where the last step follows by the property of binomial terms that for any $\alpha > 0$ and integer $i > 1$, $\binom{\alpha}{i-1} + \binom{\alpha}{i} = \binom{\alpha+1}{i}$. Then by setting $\lambda = 0$ in the equation above, we have

$$\binom{\alpha}{m}(-1)^m + \sum_{i=m+1}^{\infty} \binom{\alpha+1}{i}(-1)^i = 0.$$

Through similar steps as the proof above, we have

$$\begin{aligned} |\lambda^\alpha - f_m(\lambda)| &= \lambda_{\max}^\alpha \left| \sum_{i=m+1}^{\infty} \binom{\alpha}{i} \left(\frac{\lambda}{\lambda_{\max}} - 1 \right)^i \right| \\ &\leq \lambda_{\max}^\alpha \left| \sum_{i=m+1}^{\infty} \binom{\alpha}{i} (-1)^i \right| \\ &= \lambda_{\max}^\alpha \left| - \binom{\alpha-1}{m} (-1)^m \right| \\ &\leq \frac{\lambda_{\max}^\alpha \Gamma(\alpha)}{\pi} \cdot \frac{1}{(m-\alpha+1)^\alpha}. \end{aligned}$$

Then by choosing m as:

$$m \geq \alpha + \sqrt[\alpha]{\frac{n\Gamma(\alpha+1)}{\epsilon\pi}},$$

we have $|\lambda^\alpha - f_m(\lambda)| \leq \frac{\epsilon}{n}\lambda_{\max}^\alpha$ and

$$\sum_{i=1}^{n} |\lambda_i^\alpha - f_m(\lambda_i)| \leq \epsilon \lambda_{\max}^\alpha \leq \epsilon \cdot \text{tr}(G^\alpha),$$

which finishes the proof for the case $\lambda_{\min} = 0$. ■

Remark 2: Theorem 2 establishes the statistical guarantee for Algorithm 2, whose approximation error relies on the condition number κ of G, the number of random vectors s and the degree of Taylor series m. Note that $s = \mathcal{O}(1/\epsilon^2)$ is required for both full-rank and rank-deficient cases, but the requirement for m is slightly different. As shown in (6), $m = \mathcal{O}(\alpha + \kappa \log(\frac{\Gamma(\alpha+1)}{\epsilon(\alpha+1)^{\alpha+1}} + 1))$ (by the fact that $W_0(z) \leq \log(1+z), z \geq 0$) is sufficient to yield a meaningful approximation for full rank cases, less than the requirement of rank-deficient cases where $m = \mathcal{O}(\alpha + (\frac{n\Gamma(\alpha+1)}{\epsilon})^{\frac{1}{\alpha}})$.

C. Chebyshev Series Approximation

Chebyshev series leverage orthogonal polynomials to approximate arbitrary analytic functions. It serves as an alternative approach for polynomial approximation, which usually enjoys faster convergence rates compared to Taylor series. Let $T_k(x) = \cos(k \arccos(2x/\lambda_{\max} - 1))$ with $x \in [0, \lambda_{\max}]$ be the Chebyshev polynomials of the first kind for any integer $k \geq 0$, and let $f_m(x) = c_0 T_0(x)/2 + \sum_{k=1}^m c_k T_k(x)$ with

$$c_k = \frac{2\lambda_{\max}^\alpha \Gamma(\alpha + 1/2)(\alpha)_k}{\sqrt{\pi}\Gamma(\alpha+1)(\alpha+k)_k}, \tag{15}$$

where $(\alpha)_k = \alpha \cdot (\alpha-1) \cdots (\alpha-k+1)$ is the falling factorial. Algorithm 3 summarizes the procedure of approximating fractional order matrix-based Rényi's entropy through Chebyshev series, in which Clenshaw's algorithm [36] is employed to calculate $F_i(G) = \mathbf{g}_i^\top f_m(G)\mathbf{g}_i$.

Algorithm 3: Fractional Order Matrix-Based Rényi's Entropy Estimation via Chebyshev Series.

```
Input: Kernel matrix $G \in \mathbb{R}^{n\times n}$, number of random vectors $s$, fractional order $\alpha > 0$, polynomial degree $m > \alpha$.
Output: Approximation to $S_\alpha(G)$.
Calculate $\lambda_{\max}$ by power iteration.
Calculate Chebyshev coefficients $c_k$, $k \in [0, m]$ by Eq. (15).
Generate $s$ independent random standard Gaussian vectors $\mathbf{g}_i, i = 1, \ldots, s$.
for $i = 1, 2, \ldots, s$ do
  Set $\mathbf{y}_{m+2} = \mathbf{y}_{m+1} = \mathbf{0}$.
  for $k = m, m-1, \ldots, 0$ do
    $\mathbf{y}_k = c_k \mathbf{g}_i + 4G\mathbf{y}_{k+1}/\lambda_{\max} - 2\mathbf{y}_{k+1} - \mathbf{y}_{k+2}$.
  end for
  Calculate $F_i(G) = c_0 \mathbf{g}_i^\top \mathbf{g}_i/4 + \mathbf{g}_i^\top(\mathbf{y}_0 - \mathbf{y}_2)/2$.
end for
Return: $\tilde{\mathbf{S}}_\alpha(G) = \frac{1}{1-\alpha}\log_2(\frac{1}{s}\sum_{i=1}^s F_i(G))$.
```

Theorem 3: Let $G \in \mathbb{R}^{n\times n}$ be a normalized PSD kernel matrix and $\tilde{\mathbf{S}}_\alpha(G)$ be the output of Algorithm 3 with $s = \lceil 8\ln(2/\delta)/\epsilon^2 \rceil$ and

$$m = \left\lceil \alpha + \begin{cases} \sqrt{\kappa}\sqrt[2\alpha]{\frac{\Gamma(\alpha+\frac{1}{2})\Gamma(\alpha)}{\epsilon\pi^{3/2}}} & \text{if } \lambda_{\min} > 0 \\ \sqrt[2\alpha]{\frac{n\Gamma(\alpha+\frac{1}{2})\Gamma(\alpha)}{\epsilon\pi^{3/2}}} & \text{if } \lambda_{\min} = 0 \end{cases} \right\rceil.$$

Then with confidence at least $1 - \delta$,

$$\left| \mathbf{S}_\alpha(G) - \tilde{\mathbf{S}}_\alpha(G) \right| \leq \left| \frac{1}{1-\alpha}\log_2(1-3\epsilon) \right|.$$

Please refer to the appendix for the proof.

Remark 3: Theorem 3 establishes the statistical guarantees of Algorithm 3 in terms of s, κ and m. Similar with Taylor approximation, $s = \mathcal{O}(1/\epsilon^2)$ is required for both full rank and rank-deficient cases, where the corresponding m is $\mathcal{O}(\alpha + \sqrt{\kappa}(\frac{\Gamma(2\alpha)}{\epsilon})^{\frac{1}{2\alpha}})$ and $\mathcal{O}(\alpha + (\frac{n\Gamma(2\alpha)}{\epsilon})^{\frac{1}{2\alpha}})$ respectively. Compared with Theorem 2, Chebyshev approximation requires substantially lower degree than Taylor approximation especially when κ is large.

D. Lanczos Quadrature Approximation

Besides explicit polynomial expansions, an alternative approach for estimating implicit matrix-vector multiplications is the Lanczos method [37]. It could be regarded as an adaptive polynomial approximation technique, where the coefficients of each $\mathbf{g}_i, G \cdot \mathbf{g}_i, \ldots, G^m \cdot \mathbf{g}_i$ are chosen according to the kernel matrix G and random vector g. We summarize such approximation in Algorithm 4. The Lanczos method is shown to achieve faster convergence rate compared to explicit approximations such as Taylor or Chebyshev series, as shown in Theorem 4.

Theorem 4: Let $G \in \mathbb{R}^{n\times n}$ be a normalized PSD kernel matrix and $\tilde{\mathbf{S}}_\alpha(G)$ be the output of Algorithm 4 with $s =$

Algorithm 4: Fractional Order Matrix-Based Rényi's Entropy Estimation via Lanczos Iteration

```
Input: Kernel matrix $G \in \mathbb{R}^{n\times n}$, number of random vectors $s$, fractional order $\alpha > 0$, Lanczos steps $m$.
Output: Approximation to $S_\alpha(G)$.
Generate $s$ i.i.d. random Rademacher vectors $\mathbf{g}_i$, $i = 1, \ldots, s$, i.e. $\mathbb{P}\{(\mathbf{g}_i)_j = 1\} = \mathbb{P}\{(\mathbf{g}_i)_j = -1\} = \frac{1}{2}$.
for $i = 1, 2, \ldots, s$ do
  Set $\mathbf{q}_0 = 0, \beta_0 = 0, \mathbf{q}_1 = \mathbf{g}_i/\|\mathbf{g}_i\|$.
  for $j = 1, 2, \ldots, m$ do
    $\hat{\mathbf{q}}_{j+1} = A\mathbf{q}_j - \beta_{j-1}\mathbf{q}_{j-1}$, $\gamma_j = \langle \hat{\mathbf{q}}_{j+1}, \mathbf{q}_j \rangle$.
    $\hat{\mathbf{q}}_{j+1} = \hat{\mathbf{q}}_{j+1} - \gamma_j \mathbf{q}_j$.
    Orthogonalize $\hat{\mathbf{q}}_{j+1}$ against $\mathbf{q}_1, \ldots, \mathbf{q}_{j-1}$.
    $\beta_j = \|\hat{\mathbf{q}}_{j+1}\|$, $\mathbf{q}_{j+1} = \hat{\mathbf{q}}_{j+1}/\beta_j$.
  end for
  Let $\mathbf{p}$ be the first column of $T^\alpha$, where
    $T = \begin{vmatrix} \gamma_1 & \beta_1 & & \\ \beta_1 & \gamma_2 & \ddots & \\ & \ddots & \ddots & \beta_{m-1} \\ & & \beta_{m-1} & \gamma_m \end{vmatrix}$.
  Calculate $F_i = \mathbf{g}_i^T \sum_{k=1}^m (\mathbf{p})_k \mathbf{q}_k$.
end for
Return: $\tilde{\mathbf{S}}_\alpha(G) = \frac{1}{1-\alpha}\log_2(\frac{\sqrt{n}}{s}\sum_{i=1}^s F_i)$.
```

$\lceil 24\ln(2/\delta)/\epsilon^2 \rceil$ and

$$m = \left\lceil \frac{1}{4}\sqrt{\kappa}\log\left(\frac{\kappa^{\alpha+\frac{1}{2}}}{\epsilon}\right)\right\rceil.$$

Then with confidence at least $1-\delta$,

$$\left|\mathbf{S}_\alpha(G) - \tilde{\mathbf{S}}_\alpha(G)\right| \leq \left|\frac{1}{1-\alpha}\log_2(1-\epsilon)\right|.$$

Remark 4: Theorem 4 shows that $s = \mathcal{O}(1/\epsilon^2)$ and $m = \mathcal{O}(\max(\alpha, 1)\sqrt{\kappa}\log(\kappa/\epsilon))$ are sufficient to obtain a good approximation. Note that m in Algorithm 4 is the number of Lanczos steps, which can also be regarded as the degree of the Lanczos method. Compared with $m = \mathcal{O}(\kappa\log(1/\kappa\epsilon))$ of Theorem 2 and $m = \mathcal{O}(\sqrt[2\alpha]{1/\epsilon})$ of Theorem 3, Theorem 4 only needs $m = \mathcal{O}(\sqrt{\kappa}\log(\kappa/\epsilon))$ to achieve the desired approximation accuracy in consideration of all these approximations require $s = \mathcal{O}(1/\epsilon^2)$. This implies that the Lanczos method enjoys better theoretical properties than explicit polynomial approximation methods. However, it requires extra reorthogonalization in limited-precision computation models, which causes higher computational cost.

Remark 5: It worth noting that in most of the derived error bounds, the selection of ϵ is independent of n, which means that s and m do not need to scale up along with n to guarantee the same level of accuracy. Therefore, for a large enough n, we can achieve arbitrary level of accuracy by selecting increasingly larger s and m under the precondition that $s \ll n$ and $m \ll n$. For some of the other bounds, e.g. the $\lambda_{\min} = 0$ case in Theorem 2, 3, the upper bounds involve n so that m needs to scale up with n. When $\alpha > 1$, we have $m = O(\sqrt[\alpha]{n/\epsilon})$ in Theorem 2 (or $m = O(\sqrt[2\alpha]{n/\epsilon})$ in Theorem 3) so that the increment of m is slower than n (i.e. $m = o(n)$), which results in the same conclusion as above. Otherwise if $\alpha < 1$ which is rarely the case, there exists an upper limit for the achievable level of approximation accuracy (i.e. a lower bound for ϵ) using the developed algorithms under the precondition that $s \ll n$ and $m \ll n$. However, our simulation studies show that the achieved accuracy is still desirable in this situation (please refer to the appendix).

IV. Approximating Matrix-Based Rényi's Entropy: From the Viewpoint of Matrix Approximation

Previous section establishes the efficient approximations from the viewpoint of randomized linear algebra, in which the main computation cost comes from matrix-vector multiplications required in both power iteration and trace estimation, leading to $\mathcal{O}(ms \cdot \mathbf{nnz}(G))$ time complexity, where $m = \alpha$ for integer α-orders and $\mathbf{nnz}(G)$ denotes the cost of calculating matrix-vector multiplication using the straightforward approach. In view of the fact that the kernel matrix is usually dense in practice, $\mathbf{nnz}(G)$ is proportional to $\mathcal{O}(n^2)$, which is still unaffordable in large-scale data science tasks. We hence expect to take full advantage of the structure of the kernel matrix G to further reduce the computation burden. To this end, we first establish the connection between matrix-based Rényi's entropy approximation and kernel matrix approximation in the following theorem:

Theorem 5: For any symmetric PSD matrix G and its symmetric approximation $\tilde{G}$, the matrix-based Rényi's α-order entropy of $\tilde{G}$ is bounded by

$$\left|\mathbf{S}_\alpha(G) - \mathbf{S}_\alpha(\tilde{G})\right| \leq \left|\frac{\alpha}{1-\alpha}\log_2\left(1 - \sqrt{n}\left\|G^{-1}\right\|_2\left\|G - \tilde{G}\right\|_2\right)\right|.$$

Proof: Noticing that

$$\begin{aligned}\left|\mathbf{S}_\alpha(G) - \mathbf{S}_\alpha(\tilde{G})\right| &= \left|\frac{1}{1-\alpha}\log_2\frac{\sum_{i=1}^n \lambda_i^\alpha(\tilde{G})}{\sum_{i=1}^n \lambda_i^\alpha(G)}\right| \\ &\leq \left|\frac{1}{1-\alpha}\log_2\max_i\frac{\lambda_i^\alpha(\tilde{G})}{\lambda_i^\alpha(G)}\right| \\ &= \left|\frac{\alpha}{1-\alpha}\log_2\max_i\frac{\lambda_i(\tilde{G})}{\lambda_i(G)}\right|. \end{aligned} \tag{16}$$

Note that there is no restriction on the order of eigenvalues, i.e. we can reorder the eigenvalues $\lambda_i(G)$ and $\lambda_i(\tilde{G})$ arbitrarily for $i = 1, \ldots, n$. To upper bound the eigenvalue ratio above, we introduce the following lemma:

Lemma 3: [38] there exists a permutation τ of $\{1, \ldots, n\}$ such that for normal non-singular matrix A,

$$\max\left|\frac{\mu_{\tau(i)} - \lambda_i}{\lambda_i}\right| \leq \sqrt{n(n-s+1)}\left\|A^{-1}\right\|_2\|E\|_2. \tag{17}$$

where λ_i and μ_i are eigenvalues of A and $\tilde{A} = A + E$ respectively. s comes from assuming that there exists a unitary matrix

U such that $U^* AU = \mathbf{diag}(A_1, \ldots, A_s)$, where $1 \leq s \leq n$, and A_i is an upper triangular matrix for $i = 1, \ldots, s$.

When the approximation $\tilde{G}$ is symmetric, $\tilde{G}$ is also a normal matrix and $s = n$ is guaranteed. Combining (16) and (17) yields the final result. ■

Theorem 5 verifies the availability of reducing the computational burden by approximating the kernel matrix. A common solution is to build approximations using low-rank algorithms e.g. the Nyström method, which are shown to be an effective technique to tackle large-scale datasets with no significant decrease in performance [39]. However, computing matrix-based Rényi's entropy involves every eigenvalue of G, which is contradictory to the behavior of these algorithms to keep only larger eigenvalues and ignore smaller ones. This results in losing a large portion of the information contained in the eigenspectrum, and can greatly harm the performance for matrix-based Rényi's entropy approximation.

Motivated by the recent progress on kernel approximation [40], we consider exploring both the clustering and low-rank structure of kernel matrices. In case of the most widely used shift-invariant kernels e.g. Gaussian kernel and Laplacian kernel, we can rewrite their functions as $\varphi(\mathbf{x}_i, \mathbf{x}_j) = g(\sigma(\mathbf{x}_i - \mathbf{x}_j))$, where $g : \mathbb{R}^d \to \mathbb{R}$ is a measurable function and σ is the scale parameter. When σ is large enough, the off-diagonal entries in G become small in magnitude compared to diagonal elements. In such cases, the majority of information is stored in the diagonal blocks, so it is natural to adopt low-rank approximation to reduce computational cost. However, for relatively small σ, the impacts of off-diagonal blocks cannot be ignored since the kernel matrix remains dense. To accommodate a wider range of σ values, we employ a block low-rank structure to reduce the computational burden. Take some partition $\mathcal{V}_1, \ldots, \mathcal{V}_c$ of the given samples, where $\mathcal{V}_1 \cup \cdots \cup \mathcal{V}_c = \{n\}$ and $\mathcal{V}_s \cap \mathcal{V}_t = \phi$ for any $1 \leq s < t \leq c$, the kernel matrix G can be approximated by $\tilde{G}$ with both block and low-rank structure as given below:

$$G \approx \tilde{G} = \begin{bmatrix} G^{(1,1)} & G_k^{(1,2)} & \cdots & G_k^{(1,c)} \\ G_k^{(2,1)} & G^{(2,2)} & \cdots & G_k^{(2,c)} \\ \vdots & \vdots & \ddots & \vdots \\ G_k^{(c,1)} & G_k^{(c,2)} & \cdots & G^{(c,c)} \end{bmatrix}, \tag{18}$$

where $G_k^{(s,t)}$ denotes the rank-k approximation of the sub-matrix $G^{(s,t)}$ constructed by rows $\mathcal{V}_s$ and columns $\mathcal{V}_t$ of the original kernel matrix G. Observing that

$$\|G - \tilde{G}\|_F^2 \leq \sum_{i,j} \varphi(\mathbf{x}_i, \mathbf{x}_j)^2 - \sum_{s=1}^{c} \sum_{i,j \in \mathcal{V}_s} \varphi(\mathbf{x}_i, \mathbf{x}_j)^2,$$

therefore, minimizing the difference between G and $\tilde{G}$ is equivalent to maximizing the second term:

$$\min \|G - \tilde{G}\|_F^2 \Leftrightarrow \max \sum_{s=1}^{c} \sum_{i,j \in \mathcal{V}_s} \varphi(\mathbf{x}_i, \mathbf{x}_j)^2 := D^{kernel}. \tag{19}$$

However, directly maximizing D^{kernel} can result in all the data being assigned to the same cluster. A common way to solve this problem is to normalize D by each cluster's size $|\mathcal{V}_s|$. This leads to the spectral clustering objective:

$$D^{kernel}(\{\mathcal{V}_s\}_{s=1}^c) = \sum_{s=1}^{c} \frac{1}{|\mathcal{V}_s|} \sum_{i,j \in \mathcal{V}_s} \varphi(\mathbf{x}_i, \mathbf{x}_j)^2. \tag{20}$$

Directly optimizing (20) is computational infeasible since we have to iterate all possibilities of the partition $\{\mathcal{V}_s\}_{s=1}^c$. For applicable minimization in practice, we adopt a lower bound for our objective $D^{kernel}(\{\mathcal{V}_s\}_{s=1}^c)$:

Theorem 6: For any shift-invariant Lipschitz continuous kernel function φ,

$$D^{kernel}(\{\mathcal{V}_s\}_{s=1}^c) \geq \frac{1}{2n} - R^2 D^{kmeans}(\{\mathcal{V}_s\}_{s=1}^c),$$

where R is a constant depending on the kernel function, and

$$D^{kmeans}(\{\mathcal{V}_s\}_{s=1}^c) \equiv \sum_{s=1}^{c} \frac{1}{|\mathcal{V}_s|} \sum_{i,j \in \mathcal{V}_s} \|x_i - x_j\|_2^2$$

is the k-means objective function.

Proof: For any shift-invariant kernel function, there exists a real valued function $f \in \mathbb{R}^d \mapsto \mathbb{R}$ so that for any data points $\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^d$,

$$\varphi(\boldsymbol{x}_i, \boldsymbol{x}_j) = f(\boldsymbol{x}_i - \boldsymbol{x}_j) \geq f(\mathbf{0}) - L \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2,$$

where L is the Lipschitz constant of $f(\cdot)$. Then the elements of matrix G satisfies:

$$\begin{aligned} G_{ij} &= \frac{1}{n} \frac{\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sqrt{\kappa(\boldsymbol{x}_i, \boldsymbol{x}_i)\kappa(\boldsymbol{x}_j, \boldsymbol{x}_j)}} = \frac{f(\boldsymbol{x}_i - \boldsymbol{x}_j)}{nf(\mathbf{0})} \\ &\geq \frac{1}{nf(\mathbf{0})} \left(f(\mathbf{0}) - L \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \right) \\ &= \frac{1}{n} - \frac{L}{nf(\mathbf{0})} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2. \end{aligned}$$

Let $R \equiv L/nf(\mathbf{0})$, we have

$$G_{ij} + R \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \geq \frac{1}{n}.$$

Taking the square of both sides

$$G_{ij}^2 + R^2 \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 + 2G_{ij}R \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \geq \frac{1}{n^2}.$$

From the classical arithmetic and geometric mean inequality, we get the following bound:

$$2G_{ij}R \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 \leq G_{ij}^2 + R^2 \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2,$$

and therefore:

$$G_{ij}^2 + R^2 \|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 \geq \frac{1}{2n^2}.$$

Algorithm 5: Block Low-Rank Kernel Matrix Approximation.

```
Input: Shift-invariant kernel matrix $G \in \mathbb{R}^{n \times n}$, rank $k$, number of clusters $c$.
Output: Approximation to $G$.
Obtain a partition $\mathcal{V}_1, \ldots, \mathcal{V}_c$ by the $k$-means algorithm with $c$ clusters.
Rearrange the kernel matrix $G$ as a block matrix by the partition $\mathcal{V}_1, \ldots, \mathcal{V}_c$.
Obtain the best rank-$k$ approximation of each off-diagonal block.
Return: Construct $\tilde{G}$ by (18).
```

According to the definition of D^{kernel}:

$$\begin{aligned} D^{kernel}(\{\mathcal{V}_s\}_{s=1}^c) &= \sum_{s=1}^c \frac{1}{|\mathcal{V}_s|} \sum_{i,j \in \mathcal{V}_s} G_{ij}^2 \\ &\geq \sum_{s=1}^c \frac{1}{|\mathcal{V}_s|} \sum_{i,j \in \mathcal{V}_s} \left(\frac{1}{2n^2} - R^2 \left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|_2^2 \right) \\ &= \frac{1}{2n} - R^2 \sum_{s=1}^c \frac{1}{|\mathcal{V}_s|} \sum_{i,j \in V_s} \left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|_2^2 \\ &= \frac{1}{2n} - R^2 D^{kmeans}(\{\mathcal{V}_s\}_{s=1}^c), \end{aligned}$$

which finishes the proof. ■

Theorem 6 suggests that the k-means algorithm is an ideal choice for selecting the partition $\mathcal{V}_1, \ldots, \mathcal{V}_c$. The progress of block low-rank kernel approximation is summarized in Algorithm 5, where the k largest singular values and the corresponding singular vectors are calculated by Randomized SVD algorithm [41] in practice, which takes $\mathcal{O}(n_r n_c k)$ arithmetic operations for a $n_r \times n_c$ sub-matrix. Through the block low-rank approximation, the complexity of matrix-vector multiplication is reduced to $\mathcal{O}(n^2/c + nck)$. The following proposition gives the error bound of kernel approximation using the given strategy:

Proposition 1: Given samples $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^d$ with a partition $\mathcal{V}_1, \ldots, \mathcal{V}_c$. Let $\tilde{G}$ be the output of Algorithm 5 and let the radius of partition $\mathcal{V}_i$ be r_i for $1 \leq i \leq c$. Assuming $r_1 \leq r_2 \leq \ldots \leq r_c$, then for any shift-invariant Lipschitz continuous kernel function, we have

$$\left\| G - \tilde{G} \right\|_F \leq 4Lk^{-\frac{1}{d}} \sqrt{2r}, \tag{21}$$

where L is the Lipschitz constant of the kernel function, and $r = \sum_{i=1}^c r_i^2 |\mathcal{V}_i| \sum_{j=i+1}^c |\mathcal{V}_j|$.

Proof: Denote $G^{(s,t)}$ as one block of matrix G with rows $\mathcal{V}_s$ and columns $\mathcal{V}_t$. By Theorem 2 in [40], for any shift-invariant Lipschitz continuous kernel function with Lipschitz constant L, we have the following bound:

$$\left\| G^{(s,t)} - G_k^{(s,t)} \right\|_F \leq 4Lk^{-\frac{1}{d}} \sqrt{|\mathcal{V}_s|\,|\mathcal{V}_t|} \min(r_s, r_t).$$

Summing up the inequality above for all off-diagonal blocks yields the final result:

$$\begin{aligned} \left\| G - \tilde{G} \right\|_F &= \sqrt{\sum\nolimits_{s,t=1\ldots c, s<t} 2 \left\| G^{(s,t)} - G_k^{(s,t)} \right\|_F^2} \\ &\leq 4Lk^{-\frac{1}{d}} \sqrt{\sum\nolimits_{s,t=1\ldots c, s<t} 2\,|\mathcal{V}_s|\,|\mathcal{V}_t| \min{}^2(r_s, r_t)} \\ &= 4Lk^{-\frac{1}{d}} \sqrt{2 \sum\nolimits_{i=1}^c r_i^2\, |\mathcal{V}_i| \sum\nolimits_{j=i+1}^c |\mathcal{V}_j|}. \end{aligned}$$

■

Combining Theorem 5 and Proposition 1 together yields the following corollary on approximation error of matrix-based Rényi's entropy with block low-rank approximation technique:

Corollary 1: Under the same conditions in Proposition 1, the approximation error of $\mathbf{S}_\alpha(G)$ is bounded by

$$\left| \mathbf{S}_\alpha(G) - \mathbf{S}_\alpha(\tilde{G}) \right| \leq \left| \frac{\alpha}{1-\alpha} \log_2 \left(1 - 4\sqrt{2rn}Lk^{-\frac{1}{d}} \left\| G^{-1} \right\|_2 \right) \right|.$$

Remark 6: Obviously, there is a trade-off in the choice of k and c between time complexity and approximation error. On the one hand, small k or large c may cause higher approximation error, or even worse that the approximated kernel matrix loses its positive semi-definite property. On the other hand, kernel approximation loses its running time improvement for large k or small c. Empirically, we suggest $c \approx \sqrt[4]{n}$ and $k \approx \sqrt{n}$ for the best performance.

A. Approximation for Matrix-Based Rényi's Mutual Information and Total Correlation

In the discussion above, we developed randomized approximations through both stochastic trace estimators and block low-rank kernel approximations for matrix-based Rényi's entropy. As natural extensions of the Rényi's α-order entropy functional, matrix-based Rényi's mutual information (Eq. (4)) and total correlation (Eq. (5)) can also be directly approximated by the established algorithms, since they are just linear combinations of the original definition. Moreover, mutual information and total correlation are fundamental information statistics for most downstream machine learning and neuroscience tasks, such as measuring dependence or causality, feature selection and regularizations in deep neural training. Comprehensive experiments in Section V-B demonstrate the excellent performance of the proposed approximation algorithms in multiple real-world data science tasks.

V. Experimental Results

In this section, we evaluate the performance of the proposed approximation algorithms on large-scale simulation and real-world datasets. All numerical studies are conducted with an Intel i7-10700 (2.90 GHz) CPU, an RTX 2080Ti GPU and 64 GB of RAM. Approximation algorithms are implemented in C++ and Python, with k-means algorithm provided by OpenCV [42], fundamental linear algebra functions provided by Eigen [43] and Pytorch respectively.

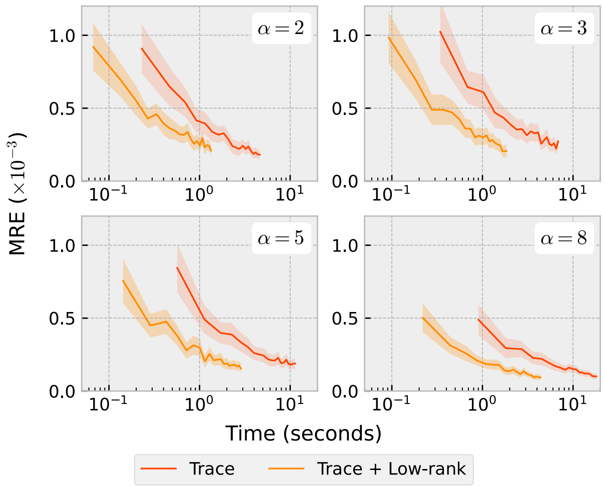

Fig. 1. Time vs. MRE curves for integer α-order Rényi's entropy estimation.

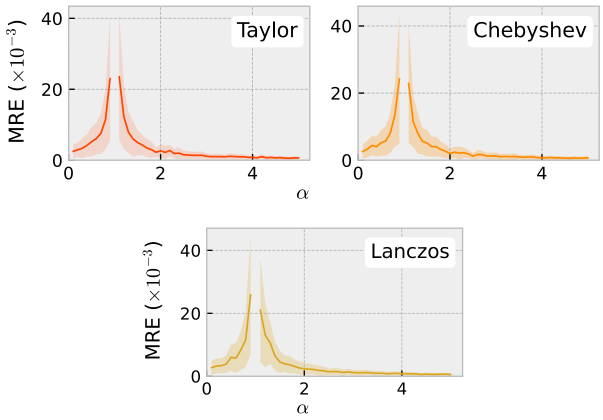

Fig. 2. α vs. MRE curves for fractional α-order algorithms.

A. Simulation Studies

In our experiments, the simulation data are generated by mixture of Gaussian distribution $\frac{1}{2}N(-1, \mathbf{I}_d) + \frac{1}{2}N(1, \mathbf{I}_d)$ with $n = 10,000$ and $d = 10$, the size of the kernel matrix is then $10,000 \times 10,000$. We choose Gaussian kernel $\varphi(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/2\sigma^2)$ with $\sigma = 1$ in kernel matrix construction. We set order of rank $k = 80$ and number of clusters $c = 20$ in block low-rank approximation experiments. To reduce the impact of randomness, we run each test for $K = 100$ times and report the mean relative error (MRE) and corresponding standard deviation (SD). The oracle $\mathbf{S}_\alpha(G)$ is computed through the trivial $\mathcal{O}(n^3)$ eigenvalue approach. We use the high resolution clock provided by the C++ standard library, whose precision reaches nanoseconds. For comparison, the straightforward eigenvalue approach takes 219.8 seconds for a 10000×10000 kernel matrix.

1) Randomized Approximation: We examine the performance of the trace estimation algorithm "Trace" and its combination with block low-rank approximation "Trace + Low-rank" for Rényi's entropy with integer orders. We choose $\alpha = \{2, 3, 5, 8\}$ in Algorithm 1 and the number of random Gaussian vectors s ranges from 20 to 200 in increments of 20. The Time vs. MRE curves for different α are shown in Fig. 1, in which the shaded area indicates ± 0.25 SD. It is easy to observe that when s is large enough, "Trace + Low-rank" runs faster than "Trace" while yielding comparable estimation error.

2) Fractional Order Approximation: We further evaluate the approximation effect for fractional α orders. The impact of α on MRE for Taylor, Chebyshev and Lanczos polynomial approximations are reported in Fig. 2. It can be seen that the approximation error grows with the increase of α for $\alpha < 1$ and decreases otherwise, which supports our claims in Theorems 2, 3 and 4 in which the term $|\frac{1}{1-\alpha}|$ dominates the approximation error for fractional α-orders.

Next, we test the approximation precision of Algorithms 2, 3 and with or without block low-rank kernel approximation with polynomial degree $m = 30$. For Algorithm 4, we use the naive implementation where the Lanczos vectors are reorthogonalized every iteration with $m = 15$, since it achieves a faster convergence rate. Again, the number of random Gaussian vectors s ranges from 20 to 200 in increments of 20, and the results are shown in Fig. 3. It can be observed that the Lanczos method runs slower than Taylor or Chebyshev. For a wide range of different α values, all of the three methods produce similar MRE. Besides, the block low-rank methods save half of the running time while only introducing negligible error in the approximation.

Next, we demonstrate that the proposed algorithms are applicable to different kinds of kernel functions. Considering the prerequisite that matrix-based Rényi's entropy supports only infinitely divisible kernels, we choose the following widely-used ones to constitute our benchmark:

- Polynomial kernel: $\varphi(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + r)^p$ with $r = 1$ and $p = 2, 4$.
- Gaussian kernel: $\varphi(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|_2^2/2\sigma^2)$ with $\sigma = 0.5, 1$.

We set $\alpha = 1.5$ and $m = 50$, while s ranges from 10 to 100 in increments of 10. We keep the previous settings and for Gaussian kernel, we add further comparisons with block low-rank approximation due to its shift-invariant property. The evaluation results are presented in Fig. 4 . It can be seen that all methods achieve consistent approximations under different kernel settings, which proves that our algorithm can accommodate different kernel functions.

3) Effect of Block Low-Rank Approximation: Lastly, we examine the effect of block low-rank kernel approximation with different number of clusters c and different ranks k. We use the Gaussian kernel same as above with $\alpha = 2.5$, $s = 100$ and $m = 40$. To evaluate the impact of c and k respectively, we take a grid search for c varying from 2 to 20 with step 2 and k varying from 10 to 100 with step 10. The results are shown in Fig. 5 using Chebyshev approximation. It can be seen that with reasonably high c and k values, block low-rank approximation achieves nearly the same accuracy as the original method, while saving nearly half of running time. This phenomenon demonstrates that our approximations can reach a satisfactory trade-off in a suitable range of both c and k. The result also provides some insights on how to appropriately select the most suitable approximation

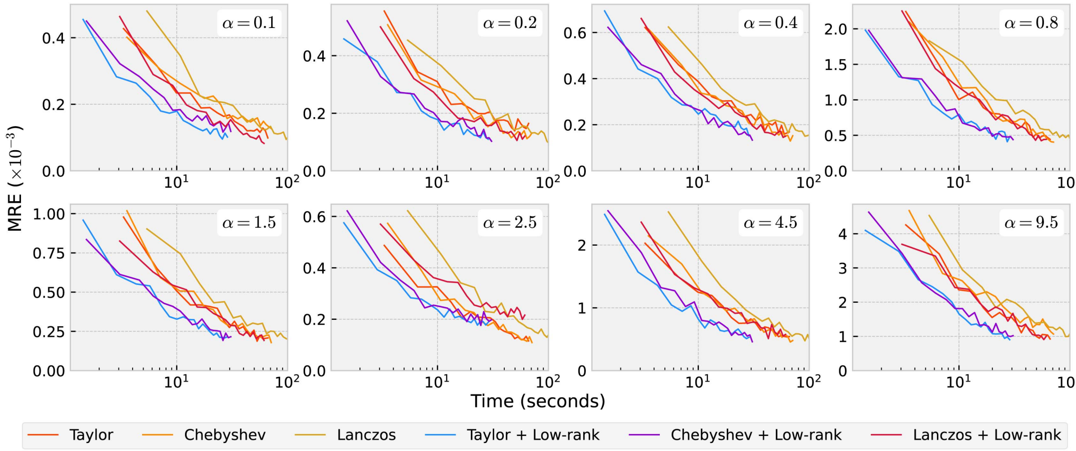

Fig. 3. Time vs. MRE curves for fractional α-order Rényi's entropy estimation.

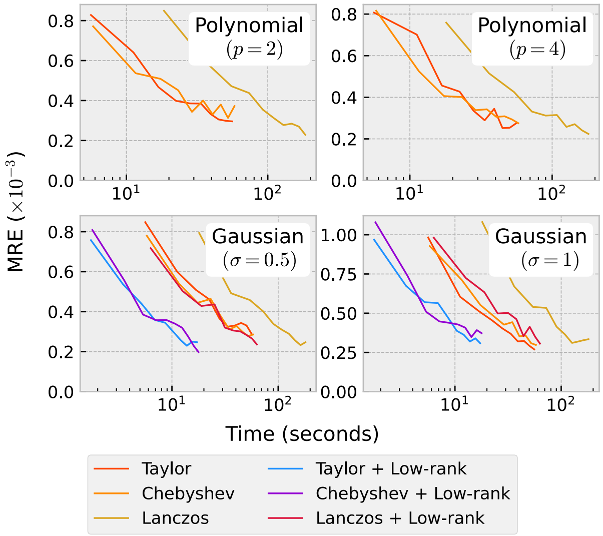

Fig. 4. Time vs. MRE curves evaluated on different kernels.

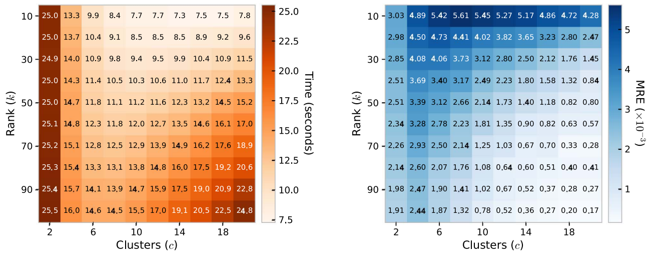

Fig. 5. Impact of c and k on Time and MRE in block low-rank kernel approximation. For comparison, the Chebyshev algorithm without low-rank approximation yields MRE 0.19×10^{-3} in 46.9 seconds.

in real applications: the running time is decreasing with c at first along with the decrease of MRE, which is due to the reduced complexity $\mathcal{O}(n^2/c + nck)$ of matrix-vector multiplication. In this sense, we recommend a range of c between 10 and 20 and a range of k between 50 and 100 to balance running time and approximation precision.

Finally, we evaluate such a hyper-parameter combination for a 50000×50000 Gram matrix. "Chebyshev + low-rank" approximations take less than 8 minutes to obtain the entropy value, whereas the original estimator requires more than 8 hours. This further validates the remarkable performance of our approximation algorithms.

B. Real-World Data Studies

Following the Venn diagram relation for Shannon entropies [44], the matrix-based Rényi's α-order mutual information and total correlation are defined accordingly as shown Eqs. (4) and (5). In real-world scenarios, these extended information quantities have much more widespread applications including feature selection, dimension reduction and information-based clustering. In this section, we apply our algorithms on three representative information-based learning tasks, namely information bottleneck, multi-view information bottleneck and feature selection, to demonstrate the acceleration effect of both matrix-based Rényi's entropy and mutual information. We select Gaussian kernel with $\sigma = 1$ and $\alpha = 2$ in the following experiments for simplicity.

1) Information Bottleneck for Robust Deep Learning: The Information Bottleneck (IB) objective was firstly introduced by [1] and has recently achieved great success in preventing overfitting in deep network training [45], [46], [47]. IB aims to learn a representation $\mathbf{T}$ by maximizing $\mathbf{I}(\mathbf{Y}, \mathbf{T})$ and minimizing $\mathbf{I}(\mathbf{X}, \mathbf{T})$ simultaneously, forcing the network to ignore irrelevant information of $\mathbf{X}$ about $\mathbf{Y}$ and thus improving both robustness and generalization. IB is formulated as:

$$\mathcal{L}_{IB} = \mathbf{I}(\mathbf{Y}, \mathbf{T}) - \beta \mathbf{I}(\mathbf{X}, \mathbf{T}). \tag{22}$$

In deep neural networks, $\mathbf{X}$ denotes the input variable, $\mathbf{Y}$ denotes the desired output (e.g., class labels) and $\mathbf{T}$ refers to the latent representation of one hidden layer. Usually, this can be done by optimizing the IB Lagrangian (22) via a classic cross-entropy (CE) loss regularized by a differentiable mutual information term $\mathbf{I}(\mathbf{X}; \mathbf{T})$. In practice, $\mathbf{I}(\mathbf{X}; \mathbf{T})$ can be measured by variational approximations [10], [48], mutual information

TABLE I
CLASSIFICATION ACCURACY (%) AGAINST ADVERSARIAL ATTACK, AND TIME SPENT ON CALCULATING IB OBJECTIVE (LEFT) AND TRAINING NETWORK (RIGHT) FOR DIFFERENT METHODS ON CIFAR-10 (THE SECOND-BEST PERFORMANCES ARE UNDERLINED)

Method	Perturbation Amount (ϵ)							Time (h)
	0.00	0.05	0.10	0.15	0.20	0.25	0.30	
Cross Entropy	93.10	63.19	55.82	51.51	45.90	41.60	38.61	- / 2.27
VIB [10]	93.77	65.33	58.14	53.02	47.93	44.01	40.37	0.03 / 2.30
NIB [48]	94.07	68.45	61.51	56.32	52.71	46.09	39.85	0.11 / 2.38
RIB [45] ($\alpha = 1.01$)	**94.36**	**69.68**	**62.45**	**57.76**	52.92	47.93	41.78	1.13 / 3.40
RIB [45] ($\alpha = 2$)	94.28	69.48	61.64	56.89	52.79	47.56	41.19	
ARIB ($\alpha = 2$)	94.30	69.33	61.47	57.08	**52.97**	**48.41**	**43.19**	0.23 / 2.50

The bold entities denote the best performance.

neural estimator (MINE) [49], [50] or the matrix-based Rényi's entropy [45].

Following the experiment settings in [10], [45], we select VGG16 [51] as the baseline network and CIFAR-10 as the classification dataset. We choose the last fully connected layer ahead softmax as the bottleneck layer. All models are trained for 400 epochs with 100 batch size where the learning rate is 0.1 initially and reduced by a factor of 10 every 100 epochs. We compare the performance of different IB methods, including Variational IB (VIB) [10], Nonlinear IB (NIB) [48], matrix-based Rényi's IB (RIB) [45] and our Approximated RIB (ARIB). We follow the suggestion of [45] and set $\beta = 0.01$. For ARIB, we set number of random vectors $s = 20$. To test the robustness of different methods against adversarial attacks, we adopt the Fast Gradient Sign Attack (FGSM) criterion, in which the adversarial examples are generated by:

$$\hat{x} = x + \epsilon \cdot sign(\nabla_x \mathcal{L}(\theta, x, y)),$$

where x denotes the original input, $\hat{x}$ denotes the attack input, $\mathcal{L}(\theta, x, y)$ is the loss with respect to the input x and ϵ is the perturbation amount. The final classification accuracy and time spent on network training are reported in Table I. In [45], the authors choose $\alpha = 1.01$ to recover Shannon's entropy, while the performance with $\alpha = 2$ was not tested. For completeness, we compare the performance of RIB with $\alpha = 1.01$ and $\alpha = 2$, which shows that $\alpha = 2$ is also capable of high performance. The final results demonstrate that our ARIB achieves remarkable speedup without sacrificing prediction ability compared to directly applying RIB. In particular, our approximation methods lead to 5 times accelerating effect, making the training time comparable to VIB or NIB.

2) Information Bottleneck for Multi-View Image Classification: Recently, the IB principle has been extended to multi-view learning tasks to minimize redundant information between multiple input views [52], [53]. Given different views of features $\mathbf{X}_1, \ldots, \mathbf{X}_k$ and the target label $\mathbf{Y}$, a typical multi-view deep network contains k separate encoders, each maps one specific view of input data to a latent representation $\mathbf{T}_i, i \in [1, k]$. These intermediate representations are then fused to a joint one $\mathbf{Z}$ by the classifier to acquire the final classification result. To this end, the multi-view IB objective can be defined as [52]:

$$\mathcal{L}_{MIB} = \mathbf{I}(\mathbf{Y}; \mathbf{Z}) - \sum_{i=1}^{k} \beta_i \mathbf{I}(\mathbf{X}_i; \mathbf{Z}_i),$$

where $\beta_i, i \in [1, k]$ balances the trade-off between representability and robustness in the i-th view.

In this experiment, we follow the settings in [54] and choose MNIST as our benchmark, where the $k = 2$ views are constructed by randomly rotating input images with an angle in $[\pi/4, \pi/4]$, and adding $[-0.1, 0.1]$ uniformly distributed background noise after normalization. The classification network is built upon an MLP of form 512-512-512 for each view, 512-256-10 for classifier and ReLU activation layers. All models are trained for 60 epochs with 100 batch size. We compare the performance of matrix-based Rényi's multi-view IB (MEIB) [54], our approximated MEIB (AMEIB) and Deep Multi-view IB (Deep IB) [52]. We select the values of $\beta_1 = 0.001$ and $\beta_2 = 0.01$ through cross-validation. The final results of classification accuracy and training time are reported in Table II. Again, it can be seen that with our approximation methods, we achieved a 3 times speedup, which is in the same level of training time compared to Deep IB, while bringing negligible drop in classification accuracy. This improvement could be further enlarged with larger batch sizes, which is a common choice in modern fine-tuning techniques [55].

TABLE II
CLASSIFICATION ACCURACY FOR MULTI-VIEW IMAGE CLASSIFICATION, AND TIME SPENT ON CALCULATING IB OBJECTIVE (LEFT) AND TRAINING NETWORK (RIGHT) FOR DIFFERENT METHODS ON MNIST (THE SECOND-BEST PERFORMANCES ARE UNDERLINED)

Method	Accuracy (%)	Time (h)
Cross Entropy	96.93 ± 0.10	- / 1.22
Deep IB [52]	97.22 ± 0.14	0.06 / 1.28
MEIB [54] ($\alpha = 1.01$)	**97.50 ± 0.08**	0.21 / 1.43
MEIB [54] ($\alpha = 2$)	97.46 ± 0.07	
AMEIB ($\alpha = 2$)	97.47 ± 0.13	0.07 / 1.29

The bold entities denote the best performance.

TABLE III
DATASETS USED IN FEATURE SELECTION TASKS, AND THE TOTAL RUNNING TIME OF RMI AND ARMI

Dataset	# Instance	# Feature	# Class	Type	Running Time (h)	
					RMI	ARMI
Covid	5434	21	2	discrete	2.56	0.47
Optdigits	5620	65	10	discrete	10.68	1.25
Statlog	6435	37	6	discrete	8.48	1.71
Gesture	11674	65	4	discrete	86.03	8.19
Spambase	4601	57	2	continuous	4.88	1.15
Waveform	5000	40	3	continuous	4.43	0.77
Galaxy	9150	16	2	continuous	8.64	1.29
Beans	13611	17	7	continuous	30.35	3.11

TABLE IV
FEATURE SELECTION RESULTS IN TERMS OF CLASSIFICATION ERROR (%)

Method	Dataset								Average Ranking
	Covid	Optdigits	Statlog	Gesture	Spambase	Waveform	Galaxy	Beans	
MIFS [7]	5.96	9.13	13.07	39.67	22.06	25.04	12.85	**7.07**	5.5
FOU [59]	3.59	14.04	12.32	39.34	19.17	20.34	12.86	7.69	5.0
MIM [60]	4.31	12.49	13.58	33.70	9.93	20.84	0.74	9.10	5.3
MRMR [61]	4.36	9.13	12.91	34.42	9.87	16.66	12.87	8.26	4.1
JMI [62]	4.32	11.01	13.36	39.03	9.98	**16.18**	0.72	8.96	4.5
CMIM [63]	4.4	**8.43**	13.77	40.30	**9.80**	17.3	5.98	7.59	4.6
RMI [16]	**3.86**	9.50	**13.35**	20.28	9.82	19.98	**0.51**	7.16	**2.8**
ARMI (Ours)	4.36	8.91	13.36	**20.06**	10.71	20.98	**0.51**	7.32	3.8

The bold entities denote the best performance.

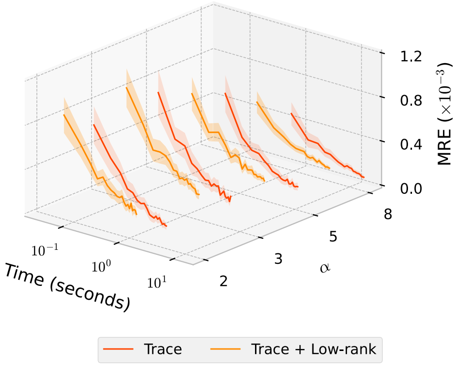

Fig. 6. Time vs. MRE curves for integer α-order Rényi's entropy estimation.

3) Mutual Information for Feature Selection: Given a set of features $S = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$, the feature selection task aims to find the smallest subset of S while trying to maximize its relevance about the labels $\mathbf{Y}$. In the view of information theory, the goal is to maximize $\mathbf{I}(\{\mathbf{X}_{i_1}, \ldots, \mathbf{X}_{i_k}\}; \mathbf{Y})$, where $i_1, \ldots, i_k$ indicate the selected features. However, the curse of high dimension causes this objective extremely hard to estimate. By employing the extension of multivariate matrix-based Rényi's mutual information [16], we are finally granted direct measure to this information quantity. We evaluate the performance of matrix-based Rényi's mutual information (RMI) and RMI approximated by our algorithms (ARMI) on 8 well-known classification datasets used in previous works [16], [56], [57], [58] which cover a wide range of instance-feature ratio, number of classes and data source domains, as shown in Table III. For completeness, we compare with state-of-the-art information based feature selection methods, including Mutual Information-based Feature Selection (MIFS) [7], First-Order Utility (FOU) [59], Mutual Information Maximization (MIM) [60], Maximum-Relevance Minimum-Redundancy (MRMR) [61], Joint Mutual Information (JMI) [62] and Conditional Mutual Information Maximization (CMIM) [63].

In our experiments, continuous features are discretized into 5 bins by equal-width strategy [64] for non-Rényi methods. We choose the Support Vector Machine algorithm implemented by libSVM [65] with RBF kernel ($\sigma = 1$) as the classifier and employ 10-fold cross-validation. In our observation, classification error tends to stop decreasing after incrementally selecting the first $k = 8$ features, so we select the first 8 most informative features step by step through a greedy criterion and report the best validation error achieved by each method. Experiment results are shown in Table IV in terms of classification error, with corresponding running time reported in Table III for RMI and ARMI. As we can see, ARMI achieves 5 to 10 times speedup compared to original RMI on all datasets while introducing only less than 1% error in classification experiments, which is still significantly lower than other methods in comparison. This demonstrates the great potential of the proposed approximating algorithms in machine learning applications. It is worth noting that the Galaxy dataset contains an extremely informative feature named "redshift," which leads to higher than 95% classification accuracy solely in identifying the class of a given star. Both RMI and ARMI can pick it out in the first place, but other methods even fail to choose it as the second high-weighted candidate.

VI. CONCLUSION

In this paper, we develop computationally efficient approximations for matrix-based Rényi's entropy. From the viewpoint of randomized linear algebra, we design random Gaussian approximation for integer-order matrix-based Rényi's entropy and use Taylor and Chebyshev series to approximate fractional matrix-based Rényi's entropy. Moreover, we exploit the structure of kernel matrices to design a memory-saving block low-rank approximation for calculating the matrix-based Rényi's entropy. Statistical guarantees are established for the proposed approximation algorithms. We also demonstrate their practical usage in real-world applications including feature selection and (multi-view) information bottleneck. In the future, we will try to acquire tighter theoretical upper bounds, and establish theoretical lower bounds for matrix-based Rényi's entropy. We will also continue to explore more practical applications of our fast matrix-based Rényi's entropy in both machine learning and neuroscience.

APPENDIX
SUPPLEMENTARY EXPERIMENT RESULTS

Additionally, we report the results for integer and fractional α-order Rényi's entropy estimation with 3D plots in Figs. 6 and 7 for a more intuitive comparison.

A. Low-Rank Approximation

We then evaluate the performance of the proposed block low-rank approximation method compared with other low-rank approximation algorithms. We use the strategy above to generate the kernel matrix G of size $1,000 \times 1,000$ with varying σ in Gaussian kernel. For a fair comparison, we fix $c = 10$

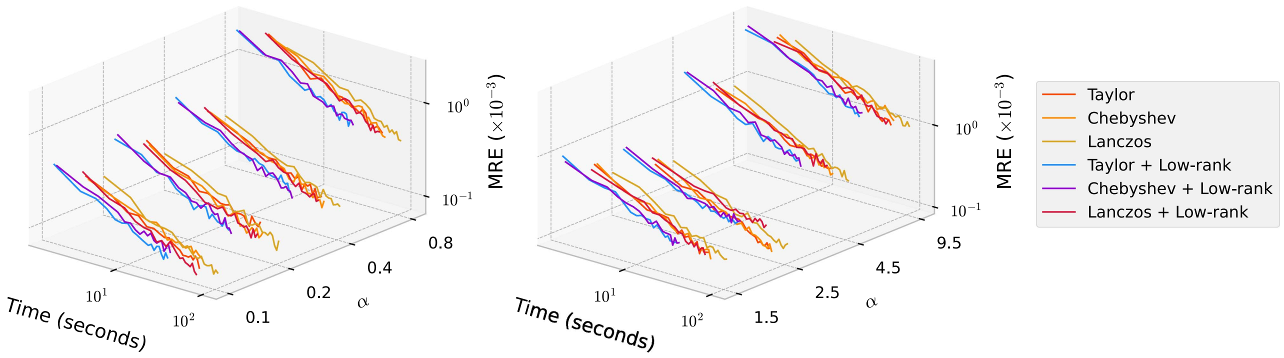

Fig. 7. Time vs. MRE curves for fractional α-order Rényi's entropy estimation.

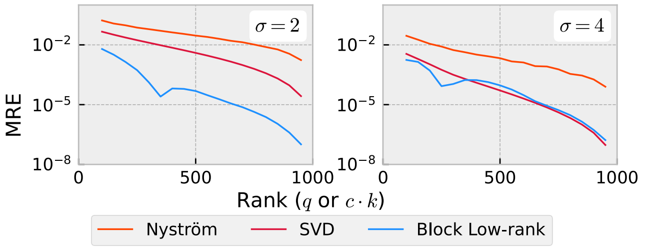

Fig. 8. Rank vs. MRE curves for different low-rank approximation methods.

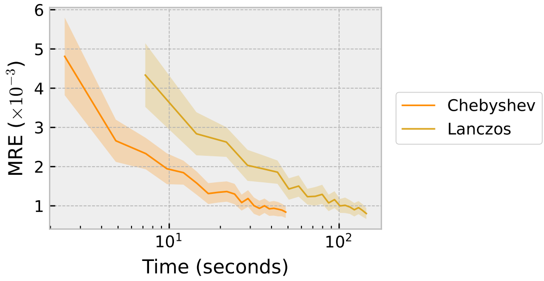

Fig. 9. Time vs. MRE curves for rank deficient Rényi's entropy estimation.

and let k varies from 10 to 100, and set the rank for other low-rank approximation methods as $q = c \cdot k$. We compare with the widely-adopted Nyström method [39] (Nyström) and the best rank-q approximation of G obtained by SVD (SVD). The mean relative error (MRE) achieved for $\alpha = 1.5$ is shown in Fig. 8. As can be seen, our block low-rank approach achieves significantly higher accuracy than the Nyström method, and is competitive with the best rank-q approximation. This verifies the advantage of using our block low-rank method to approximate matrix-based Rényi's entropy.

B. Rank Deficient Approximation

We further test the performance of Chebyshev and Lanczos approaches for fractional α-order entropy estimation with $\lambda_{\min} = 0$ and $\alpha = 0.5$, for which there exists a lower bound for ϵ. To generate rank-deficient kernel matrices, we use the polynomial kernel $\varphi(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j + 1)^2$ and set the dimension of data points $d = 138$, resulting in nearly 3% of the eigenvalues being zero. We keep the other hyper-parameters the same as our previous fractional α-order experiment, and the final Time vs. MRE curves are shown in Fig. 9. As we can see, our approximation algorithms can still achieve satisfactory level of approximation accuracy even under this extreme circumstance.

Proof of Theorem 3 and 4

C. Proof of Theorem 3

Proof: Chebyshev series of the first kind is defined as $T_n(x) = \cos(n \arccos x)$ for $n \in \mathbb{N}$ on interval $[-1, 1]$. Since the eigenvalues of G all lie in $[0, \lambda_{\max})$, the affine transformation

$$F(x) = \frac{\lambda_{\max}}{2}(x+1), \qquad \hat{T}_n(x) = T_n \circ F.$$

allows us to define Chebyshev series on interval $[0, \lambda_{\max}]$. The coefficients of Chebyshev series $\hat{T}_n$ can be calculated by:

$$\begin{aligned} c_n &= \frac{2}{\pi}\int_0^\pi F^\alpha(cos\theta)\cos(n\theta)\,\mathrm{d}\theta \\ &= \frac{2}{\pi}\int_0^\pi \left(\frac{\lambda_{\max}}{2}(\cos\theta + 1)\right)^\alpha \cos(n\theta)\,\mathrm{d}\theta \\ &= \frac{2\lambda_{\max}^\alpha \Gamma(\alpha+\frac{1}{2})(\alpha)_n}{\sqrt{\pi}\Gamma(\alpha+1)(\alpha+n)_n}, \end{aligned}$$

where $(\alpha)_n$ is the falling factorial: $(\alpha)_n = \alpha \cdot (\alpha - 1) \cdot \ldots \cdot (\alpha - n + 1)$. Let $f_m(x) = c_0/2 + \sum_{i=1}^m c_i \hat{T}_i(x)$ denote the Chebyshev series with degree m, then for each eigenvalue λ of G:

$$|\lambda^\alpha - f_m(\lambda)| = \left|\sum_{i=m+1}^\infty c_i \tilde{T}_i(\lambda)\right|$$

$$\leq \sum_{i=m+1}^\infty |c_i| = \sum_{i=m+1}^\infty \left|\frac{2\lambda_{\max}^\alpha \Gamma(\alpha+\frac{1}{2})(\alpha)_i}{\sqrt{\pi}\Gamma(\alpha+1)(\alpha+i)_i}\right| \tag{23}$$

$$= \frac{2\lambda_{\max}^\alpha}{\sqrt{\pi}} \sum_{i=m+1}^\infty \left|\frac{\Gamma(\alpha+\frac{1}{2})\Gamma(\alpha+1)}{\Gamma(\alpha+i+1)\Gamma(\alpha-i+1)}\right|$$

$$\leq \frac{2\lambda_{\max}^\alpha}{\sqrt{\pi}} \sum_{i=m+1}^\infty \left|\frac{\Gamma(\alpha+\frac{1}{2})\Gamma(\alpha+1)}{\Gamma(i-\alpha)\Gamma(\alpha-i+1)(i-\alpha)^{2\alpha+1}}\right| \tag{24}$$

$$\leq \frac{2\lambda_{\max}^\alpha \Gamma(\alpha+\frac{1}{2})\Gamma(\alpha+1)}{\pi^{3/2}} \sum_{i=m+1}^\infty \left|\frac{1}{(i-\alpha)^{2\alpha+1}}\right| \tag{25}$$

$$\leq \frac{2\lambda_{\max}^\alpha \Gamma(\alpha+\frac{1}{2})\Gamma(\alpha+1)}{\pi^{3/2}} \int_m^\infty \frac{1}{(x-\alpha)^{2\alpha+1}}\,\mathrm{d}x$$

$$= \frac{2\lambda_{\max}^{\alpha}\Gamma(\alpha+\frac{1}{2})\Gamma(\alpha+1)}{\pi^{3/2}} \frac{1}{2\alpha(m-\alpha)^{2\alpha}}$$

$$= \frac{\lambda_{\max}^{\alpha}\Gamma(\alpha+\frac{1}{2})\Gamma(\alpha)}{\pi^{3/2}(m-\alpha)^{2\alpha}}. \tag{26}$$

In (23) follows by noticing that $\hat{T}_n(x) \in [-1,1]$ for any $x \in [0, \lambda_{\max}]$. In (24) follows by applying Lemma 2 on $R(i-\alpha, 2\alpha+1)$ similar to (10). In (25) follows by Euler's reflection formula similar to (11). In (26) follows by noticing that $n^{-k} \leq \int_{n-1}^{n} x^{-k}\,\mathrm{d}x$ for $n > 1$ and $k > 1$. If G has full rank, i.e, $\lambda_{\min} > 0$, then by choosing m as:

$$m \geq \alpha + \sqrt{\kappa}\sqrt[2\alpha]{\frac{\Gamma(\alpha+\frac{1}{2})\Gamma(\alpha)}{\epsilon\pi^{3/2}}},$$

we have $|\lambda^{\alpha} - f_m(\lambda)| \leq \epsilon\lambda_{\min}^{\alpha} \leq \epsilon\lambda^{\alpha}$, which yields the same upper bound as (13).

Similarly, if G is rank deficient, i.e, $\lambda_{\min} = 0$, by choosing m as:

$$m \geq \alpha + \sqrt[2\alpha]{\frac{n\Gamma(\alpha+\frac{1}{2})\Gamma(\alpha)}{\epsilon\pi^{3/2}}},$$

we have $|\lambda^{\alpha} - f_m(\lambda)| \leq \frac{\epsilon}{n}\lambda_{\max}^{\alpha}$, which yields the same error upper bound as above. ■

D. Proof of Theorem 4

Proof: We apply the result of [37] to establish the error bound of Lanczos iteration, as shown in the following lemma:

Lemma 4: [37] Consider a symmetric positive definite matrix $A \in \mathbb{R}^{n\times n}$ with eigenvalues in $[\lambda_{\min}, \lambda_{\max}]$ and condition number $\kappa = \lambda_{\max}/\lambda_{\min}$, and let f be a function that is analytic and either positive or negative inside its interval, and whose absolute maximum and minimum values in the interval are M_ρ and m_ρ respectively. Let ϵ, η be constants in the interval $(0,1)$. Then for stochastic Lanczos quadrature parameters satisfying:

1) $s \geq \frac{24}{\epsilon^2}\ln(2/\eta)$ number of random Rademacher vectors, and
2) $m \geq \frac{\sqrt{\kappa}}{4}\log(K/\epsilon)$ number of Lanczos steps,

where $K = \frac{(\lambda_{\max}-\lambda_{\min})(\sqrt{\kappa}-1)^2 M_\rho}{\sqrt{\kappa}m_\rho}$, the output Γ is such that:

$$\mathbb{P}[|\mathrm{tr}(f(A)) - \Gamma| \leq \epsilon|\mathrm{tr}(f(A))|] \geq 1-\eta.$$

For our problem, $\lambda_{\min} \geq 0$ and $\lambda_{\max} \leq 1$, thus $\lambda_{\max} - \lambda_{\min} \leq 1$. The function f to be approximated is the α-power function $f(A) = A^{\alpha}$, therefore $M_\rho/m_\rho = \lambda_{\max}^{\alpha}/\lambda_{\min}^{\alpha} = \kappa^{\alpha}$. From the analysis above, we have the final upper bound for the number of Lanczos steps: $m = \left\lceil \frac{\sqrt{\kappa}}{4}\log(\kappa^{\alpha+\frac{1}{2}}/\epsilon)\right\rceil$. ■

References

[1] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *Proc. 37th Annu. Allerton Conf. Commun., Control Comput.*, 1999, pp. 368–377.

[2] E. T. Jaynes, "Information theory and statistical mechanics," *Phys. Rev.*, vol. 106, no. 4, pp. 620–630, 1957.

[3] E. T. Jaynes, "Information theory and statistical mechanics. II," *Phys. Rev.*, vol. 108, no. 2, pp. 171–190, 1957.

[4] J. C. Principe, *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. Berlin, Germany: Springer, 2010.

[5] J. D. Victor, "Approaches to information-theoretic analysis of neural activity," *Biol. Theory*, vol. 1, no. 3, pp. 302–316, 2006.

[6] N. M. Timme and C. Lapish, "A tutorial for information theory in neuroscience," *Eneuro*, vol. 5, no. 3, 2018, doi: 10.1523/ENEURO.0052-18.2018.

[7] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Trans. Neural Netw.*, vol. 5, no. 4, pp. 537–550, Jul. 1994.

[8] P. L. Williams and R. D. Beer, "Nonnegative decomposition of multivariate information," 2010, *arXiv:1004.2515*.

[9] M. Wibral, V. Priesemann, J. W. Kay, J. T. Lizier, and W. A. Phillips, "Partial information decomposition as a unified approach to the specification of neural goal functions," *Brain Cogn.*, vol. 112, pp. 25–38, 2017.

[10] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *Proc. Int. Conf. Learn. Representation*, 2017.

[11] R. D. Hjelm et al., et al., "Learning deep representations by mutual information estimation and maximization," in *Proc. Int. Conf. Learn. Representation*, 2010.

[12] A. Xu and M. Raginsky, "Information-theoretic analysis of generalization capability of learning algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2525–2534.

[13] J. Fan and R. Li, "Statistical challenges with high dimensionality," in *Proc. Int. Congr. Mathematicians*, 2006, pp. 595–622.

[14] L. G. S. Giraldo, M. Rao, and J. C. Principe, "Measures of entropy from data using infinitely divisible kernels," *IEEE Trans. Inf. Theory*, vol. 61, no. 1, pp. 535–548, Jan. 2015.

[15] L. G. S. Giraldo and J. C. Principe, "Information theoretic learning with infinitely divisible kernels," 2013, *arXiv:1301.3551*.

[16] S. Yu, L. G. S. Giraldo, R. Jenssen, and J. C. Principe, "Multivariate extension of matrix-based rényi alpha-order entropy functional," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 11, pp. 2960–2966, Nov. 2020.

[17] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948.

[18] A. Rényi, "On measures of entropy and information," in *Proc. 4th Berkeley Symp. Math. Statist. Probability, Volume 1: Contributions to the Theory of Statist.*, vol. 4. University of California Press, 1961, pp. 547–562.

[19] A. J. Brockmeier, T. Mu, S. Ananiadou, and J. Y. Goulermas, "Quantifying the informativeness of similarity measurements," *J. Mach. Learn. Res.*, vol. 18, pp. 1–61, 2017.

[20] A. M. Alvarez-Meza, J. A. Lee, M. Verleysen, and G. Castellanos-Dominguez, "Kernel-based dimensionality reduction using renyi's α-entropy measures of similarity," *Neurocomputing*, vol. 222, 2016, pp. 36–46, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2016.10.004

[21] C. Sarvani, M. Ghorai, S. R. Dubey, and S. S. Basha, "HRel: Filter pruning based on High Relevance between activation maps and class labels," *Neural Netw.*, vol. 147, pp. 186–197, 2022.

[22] K. Zheng, S. Yu, B. Li, R. Jenssen, and B. Chen, "Brainib: Interpretable brain network-based psychiatric diagnosis with graph information bottleneck," 2022, *arXiv:2205.03612*.

[23] I. De La Pava Panche, A. M. Alvarez-Meza, and A. Orozco-Gutierrez, "A data-driven measure of effective connectivity based on renyi's α-entropy," *Front. Neurosci.*, vol. 13, 2019, Art. no. 1277.

[24] M. Elad, *Sparse and redundant representations: From theory to applications in signal and image processing*. Berlin, Germany: Springer, 2010.

[25] M. Li, W. Bi, J. T. Kwok, and B.-L. Lu, "Large-scale Nyström kernel matrix approximation using randomized SVD," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 1, pp. 152–164, Jan. 2015.

[26] M. W. Mahoney and P. Drineas, "Cur matrix decompositions for improved data analysis," *Proc. Nat. Acad. Sci.*, vol. 106, no. 3, pp. 697–702, 2009.

[27] M. W. Mahoney, "Randomized algorithms for matrices and data," *Foundations Trends Mach. Learn.*, vol. 3, no. 2, pp. 123–224, 2011.

[28] D. S. Watkins, "The QR algorithm revisited," *SIAM Rev.*, vol. 50, no. 1, pp. 133–145, 2008.

[29] D. S. Watkins, *Fundamentals of matrix computations*. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2004, vol. 64.

[30] H. Avron and S. Toledo, "Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix," *J. ACM (JACM)*, vol. 58, no. 2, pp. 1–34, 2011.

[31] G. J. Klir and M. J. Wierman, *Uncertainty-Based Information: Elements of Generalized Information Theory*, vol. 15. Springer, 1999.

[32] R. Bhatia, "Infinitely divisible matrices," *Amer. Math. Monthly*, vol. 113, no. 3, pp. 221–235, 2006.

[33] F. Roosta-Khorasani and U. Ascher, "Improved bounds on sample size for implicit matrix trace estimators," *Foundations Comput. Math.*, vol. 15, no. 5, pp. 1187–1212, 2015. [Online]. Available: http://dx.doi.org/10.1007/s10208-014-9220-1

[34] C. Boutsidis, P. Drineas, P. Kambadur, E.-M. Kontopoulou, and A. Zouzias, "A randomized algorithm for approximating the log determinant of a symmetric positive definite matrix," *Linear Algebra Appl.*, vol. 533, pp. 95–117, 2017.

[35] S. Das, "Inequalities for Q-gamma function ratios," *Anal. Math. Phys.*, vol. 9, no. 1, pp. 313–321, 2019.

[36] C. W. Clenshaw, "A note on the summation of chebyshev series," *Math. Comput.*, vol. 9, no. 51, pp. 118–120, 1955.

[37] S. Ubaru, J. Chen, and Y. Saad, "Fast estimation of TR(F(A)) via stochastic lanczos quadrature," *SIAM J. Matrix Anal. Appl.*, vol. 38, no. 4, pp. 1075–1099, 2017.

[38] W. Li and W. Sun, "The perturbation bounds for eigenvalues of normal matrices," *Numer. Linear Algebra Appl.*, vol. 12, no. 2/3, pp. 89–94, 2005.

[39] C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 682–688.

[40] S. Si, C.-J. Hsieh, and I. S. Dhillon, "Memory efficient kernel approximation," *J. Mach. Learn. Res.*, vol. 18, no. 20, pp. 1–32, 2017.

[41] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–288, 2011.

[42] G. Bradski, "The OpenCV library," *Dr Dobb's J. Softw. Tools Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.

[43] G. Guennebaud and B. Jacob, "Eigen v3," 2010. [Online]. Available: http://eigen.tuxfamily.org

[44] R. W. Yeung, "A new outlook on shannon's information measures," *IEEE Trans. Inf. Theory*, vol. 37, no. 3, pp. 466–474, May 1991.

[45] X. Yu, S. Yu, and J. C. Principe, "Deep deterministic information bottleneck with matrix-based entropy functional," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2021, pp. 3160–3164.

[46] L. Ardizzone, R. Mackowiak, C. Rother, and U. Köthe, "Training normalizing flows with the information bottleneck for competitive generative classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 7828–7840.

[47] T. Wu, H. Ren, P. Li, and J. Leskovec, "Graph information bottleneck," in*Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 20437–20448 .

[48] A. Kolchinsky, B. D. Tracey, and D. H. Wolpert, "Nonlinear information bottleneck," *Entropy*, vol. 21, no. 12, 2019, Art. no. 1181.

[49] M. I. Belghazi et al., et al., "Mutual information neural estimation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 531–540.

[50] A. Elad, D. Haviv, Y. Blau, and T. Michaeli, "Direct validation of the information bottleneck principle for deep nets," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops*, 2019, pp. 758–762.

[51] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015, pp. 1–14.

[52] Q. Wang, C. Boudreau, Q. Luo, P.-N. Tan, and J. Zhou, "Deep multi-view information bottleneck," in *Proc. SIAM Int. Conf. Data Mining*, 2019, pp. 37–45.

[53] I. E. Aguerri and A. Zaidi, "Distributed variational representation learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 120–138, Jan. 2021.

[54] Q. Zhang, S. Yu, J. Xin, and B. Chen, "Multi-view information bottleneck without variational approximation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2022, pp. 4318–4322.

[55] S. L. Smith, P.-J. Kindermans, and Q. V. Le, "Don't decay the learning rate, increase the batch size," in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=B1Yy1BxCZ

[56] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[57] M. Koklu and I. A. Ozkan, "Multiclass classification of dry beans using computer vision and machine learning techniques," *Comput. Electron. Agriculture*, vol. 174, 2020, Art. no. 105507. [Online]. Available: https://doi.org/10.1016/j.compag.2020.105507

[58] N. X. Vinh, S. Zhou, J. Chan, and J. Bailey, "Can high-order dependencies improve mutual information based feature selection?," *Pattern Recognit.*, vol. 53, pp. 46–58, 2016.

[59] G. Brown, "A new perspective for information theoretic feature selection," *J. Mach. Learn. Res.*, vol. 5, no. 1, pp. 49–56, 2009.

[60] D. D. Lewis, "Feature selection and feature extraction for text categorization," in *Proc. Workshop Speech Natural Lang.*, 1992, pp. 212–217.

[61] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, Aug. 2005.

[62] H. Yang, "Data visualization and feature selection: New algorithms for nongaussian data," in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 687–693 .

[63] F. Fleuret, "Fast binary feature selection with conditional mutual information," *J. Mach. Learn. Res.*, vol. 5, no. 9, pp. 1531–1555, 2004.

[64] N. X. Vinh, J. Chan, and J. Bailey, "Reconsidering mutual information based feature selection: A statistical significance view," *Proc. AAAI Conf. Artif. Intell.*, vol. 28, no. 1, pp. 2092–2098, 2014.

[65] C.-C. Chang and C.-J. Lin, "LibSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–27, 2011.

Tieliang Gong received the Ph.D. degree from Xi'an Jiaotong University, Xi'an China, in 2018. From September 2018 to October 2020, he was a Postdoctoral Researcher with the Department of Mathematics and Statistics, University of Ottawa, Ottawa, ON, Canada. He is currently an Assistant Professor with the School of Computer Science and Technology, Xi'an Jiaotong University. His research interests include statistical learning theory, machine learning, and information theory.

Yuxin Dong received the B.S. degree from Xi'an Jiaotong, Xi'an, China, in 2019. He is currently working toward the Ph.D. degree with the School of Computer Science and Technology, Xi'an Jiaotong University. His research interests include information theory, statistical learning theory, and bioinformatics.

Shujian Yu received the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2019, with the Ph.D. minor in statistics. He will join the Department of Computer Science with the Vrije Universiteit Amsterdam, Amsterdam, The Netherlands, in 2023, as a Tenure-Track Assistant Professor. He is also affiliated with the Department of Physics and Technology with the UiT - The Arctic University of Norway, Troms, Norway. From 2019 to 2021, he was a machine learning Research Scientist with the NEC Labs Europe, Heidelberg, Germany. He was the recipient of the 2020 International Neural Networks Society Aharon Katzir Young Investigator Award for the contribution on the development of novel information theoretic measures for analysis and training of deep neural networks. He is also selected for the 2023 AAAI New Faculty Highlights.

Bo Dong received the Ph.D. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2014. From 2014 to 2017, he did Postdoctoral research with the MOE Key Lab for Intelligent Networks and Network Security, Xi'an Jiaotong University. He is currently the Research Director with the School of Continuing Education, Xi'an Jiaotong University. His research interests include data mining and intelligent e-Learning.