Electronic Colloquium on Computational Complexity, Revision 5 of Report No. 97 (2024)

Near-Optimal Averaging Samplers and Matrix Samplers

Zhiyang Xun*
Department of Computer Science
The University of Texas at Austin
zxun@cs.utexas.edu

David Zuckerman†
Department of Computer Science
The University of Texas at Austin
diz@utexas.edu

August 15, 2025

Abstract

We present the first efficient averaging sampler that achieves asymptotically optimal randomness complexity and near-optimal sample complexity. For any $\delta < \varepsilon$ and any constant $\alpha > 0$, our sampler uses $m + O(\log(1/\delta))$ random bits to output $t = O((\frac{1}{\varepsilon^2}\log\frac{1}{\delta})^{1+\alpha})$ samples $Z_1, \ldots, Z_t \in \{0,1\}^m$ such that for any function $f : \{0,1\}^m \to [0,1]$,

$$\Pr\left[\left|\frac{1}{t}\sum_{i=1}^{t} f(Z_i) - \mathbb{E}[f]\right| \le \varepsilon\right] \ge 1 - \delta.$$

The randomness complexity is optimal up to a constant factor, and the sample complexity is optimal up to the $O((\frac{1}{\varepsilon^2}\log\frac{1}{\delta})^{\alpha})$ factor.

Our technique generalizes to matrix samplers. A matrix sampler is defined similarly, except that $f : \{0,1\}^m \to \mathbb{C}^{d\times d}$ and the absolute value is replaced by the spectral norm. Our matrix sampler achieves randomness complexity $m + \widetilde{O}(\log(d/\delta))$ and sample complexity $O((\frac{1}{\varepsilon^2}\log\frac{d}{\delta})^{1+\alpha})$ for any constant $\alpha > 0$, both near-optimal with only a logarithmic factor in randomness complexity and an additional α exponent on the sample complexity.

We use known connections with randomness extractors and list-decodable codes to give applications to these objects. Specifically, we give the first extractor construction with optimal seed length up to an arbitrarily small constant factor above 1, when the min-entropy $k = \beta n$ for a large enough constant $\beta < 1$. Finally, we generalize the definition of averaging sampler to any normed vector space.

1 Introduction

Randomization plays a crucial role in computer science, offering significant benefits across various applications. However, obtaining true randomness can be challenging. It's therefore natural to study whether we can achieve the benefits of randomization while using few random bits.

One of the most basic uses of randomness is sampling. Given oracle access to an arbitrary function $f : \{0,1\}^m \to [0,1]$ on a large domain, our goal is to estimate its average value. By drawing $t = O(\log(1/\delta)/\varepsilon^2)$ independent random samples $Z_1, \ldots, Z_t \in \{0,1\}^m$, the Chernoff bound guarantees that the average value $\left|\frac{1}{t}\sum_{i=1}^{t} f(Z_i) - \mathbb{E}\, f\right| \le \varepsilon$ with probability at least $1 - \delta$. This method uses full independence in sampling, but more efficient strategies can be pursued. This leads to the following definition:

*Supported by NSF Grant CCF-2312573, a Simons Investigator Award (#409864, David Zuckerman), NSF award CCF-2008868 and the NSF AI Institute for Foundations of Machine Learning (IFML).

†Supported by NSF Grant CCF-2312573 and a Simons Investigator Award (#409864).

ISSN 1433-8092

Definition 1 ([BR94]). *A function* $\mathsf{Samp} : \{0,1\}^n \to (\{0,1\}^m)^t$ *is a* (δ, ε) *averaging sampler with* t *samples using* n *random bits if for every function* $f : \{0,1\}^m \to [0,1]$*, we have*

$$\Pr_{(Z_1,\ldots,Z_t)\sim\mathsf{Samp}(U_n)}\left[\left|\frac{1}{t}\sum_i f(Z_i) - \mathbb{E}\, f\right| \le \varepsilon\right] \ge 1-\delta.$$

The goal is to construct explicit samplers using a small number of random bits that have sample complexity close to the optimal. Researchers have made significant progress toward this goal, and a summary is presented in Table 1. Bellare and Rompel [BR94] suggested that interesting choices of parameters are $\delta = \exp(-\operatorname{poly}(m))$ and $\varepsilon = 1/\operatorname{poly}(m)$, which allow us to use $\operatorname{poly}(m)$ random bits and generate $\operatorname{poly}(m)$ samples. For simplicity, we assume $\delta \le \varepsilon$ throughout the paper (see Remark 10 for further discussion).

Reference	Method	Random Bits	Sample Complexity
[CEG95]	Lower Bound	$m + \log(1/\delta) - \log(O(t))$	$\Omega(\log(1/\delta)/\varepsilon^2)$
[CEG95]	Non-Explicit	$m + 2\log(2/\delta) + \log\log(1/\varepsilon)$	$2\log(4/\delta)/\varepsilon^2$
Standard	Full Independence	$O(m\log(1/\delta)/\varepsilon^2)$	$O(\log(1/\delta)/\varepsilon^2)$
[CG89]	Pairwise Independence	$O(m + \log(1/\delta))$	$O(1/(\delta\varepsilon^2))$
[Gil98]	Expander Walks	$m + O(\log(1/\delta)/\varepsilon^2)$	$O(\log(1/\delta)/\varepsilon^2)$
[BR94]	Iterated Sampling	$O(m + (\log m)\log(1/\delta))$	$\operatorname{poly}(1/\varepsilon, \log(1/\delta), \log m)$
[Zuc97]	Hash-Based Extractors	$(1+\alpha)(m + \log(1/\delta))$	$\operatorname{poly}(1/\varepsilon, \log(1/\delta), m)$
[RVW00]	Zig-Zag Extractors	$m + (1+\alpha)\log(1/\delta)$	$\operatorname{poly}(1/\varepsilon, \log(1/\delta))$
Theorem 1	Compose [RVW00] With Almost ℓ-wise Ind.	$m + O(\log(1/\delta))$	$O((\log(1/\delta)/\varepsilon^2)^{1+\alpha})$

Table 1: Comparison of averaging samplers, α any positive constant.

The best existing randomness-efficient averaging sampler comes from the equivalence between averaging samplers and extractors [Zuc97], which we will elaborate on later in the paper. Improving Zuckerman's construction, Reingold, Vadhan, and Wigderson [RVW00] provided a (δ, ε) averaging sampler for domain $\{0,1\}^m$ that uses $m + (1+\alpha)\log(1/\delta)$ random bits for any positive constant α. This almost matches the lower bound in [CEG95]. However, a notable gap remains in sample complexity: the existing construction's complexity $\operatorname{poly}(1/\varepsilon, \log(1/\delta))$ does not align with the optimal $O(\log(1/\delta)/\varepsilon^2)$. This raised the following open problem (see, e.g., [Vad12, Open Problem 4.24], [Gol11, Section 6]):

Problem 1. *Can we explicitly design a* (δ, ε) *averaging sampler for domain* $\{0,1\}^m$ *that uses* $O(m + \log(1/\delta))$ *random bits and only* $O(\log(1/\delta)/\varepsilon^2)$ *samples?*

We note that such algorithms do exist for general samplers, which query f and estimate $\mathbb{E}\, f$ through a more complicated computation than taking the average [BGG93]. However, many applications require the use of averaging samplers, such as the original use in interactive proofs [BR94]. Beyond these applications, averaging samplers act as a fundamental combinatorial object that relate to other notions such as randomness extractors, expander graphs, and list-decodable codes [Zuc97; Vad07].

1.1 Our Averaging Sampler

In this paper, we construct a polynomial-time computable (δ, ε) averaging sampler with near-optimal sample complexity using an asymptotically optimal number of random bits. In fact, the

sampler we constructed is a *strong* sampler, defined as follows:

Definition 2. *A (δ, ε) averaging sampler* Samp *is* strong *if for every sequence of t functions $f_1, \ldots, f_t : \{0,1\}^m \to [0,1]$, we have*

$$\Pr_{(Z_1,\ldots,Z_t)\sim\mathsf{Samp}(U_n)}\left[\left|\frac{1}{t}\sum_i (f_i(Z_i) - \mathbb{E}\, f_i)\right| \le \varepsilon\right] \ge 1-\delta.$$

We now state our main theorems about averaging samplers, which follow from a more general theorem that is slightly more complicated to state, Theorem 21.

Theorem 1. *For every constant $\alpha > 0$, there exists an efficient strong (δ, ε) averaging sampler for domain $\{0,1\}^m$ that uses $m + O(\log(1/\delta))$ random bits and $O((\frac{1}{\varepsilon^2}\log\frac{1}{\delta})^{1+\alpha})$ samples.*

This nearly resolves Problem 1. We also give a sampler with asymptotically optimal sample complexity but a worse randomness complexity.

Theorem 2. *There exists an efficient strong (δ, ε) averaging sampler for domain $\{0,1\}^m$ that uses $m + O(\log\frac{1}{\delta}(\log\frac{1}{\varepsilon} + \log\log\frac{1}{\delta}))$ random bits and $O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ samples.*

1.2 Matrix Samplers

A natural generalization of the classic Chernoff bound is the Matrix Chernoff Bound [Rud99; AW02; Tro12]. Suppose we wish to estimate $\mathbb{E}\, f$ for a matrix-valued function $f : \{0,1\}^m \to \mathbb{C}^{d\times d}$ satisfying $\|f(x)\| \le 1$. By drawing $t = O(\log(d/\delta)/\varepsilon^2)$ independent random samples $Z_1, \ldots, Z_t \in \{0,1\}^m$, the Matrix Chernoff Bound guarantees that

$$\Pr\left[\left\|\frac{1}{t}\sum_{i=1}^t f(Z_i) - \mathbb{E}\, f\right\| \le \varepsilon\right] \ge 1-\delta,$$

where $\|\cdot\|$ denotes the spectral norm. As in the real-valued case, we wish to derandomize this process without increasing the sample complexity too much. To address this, Wigderson and Xiao [WX05] initiated the study of randomness-efficient matrix samplers:

Definition 3. *A function $\mathsf{Samp} : \{0,1\}^n \to (\{0,1\}^m)^t$ is a d-dimensional (δ, ε) matrix sampler with t samples using n random bits if the following holds: For any function $f : \{0,1\}^m \to \mathbb{C}^{d\times d}$ such that $\|f(x)\| \le 1$ for all $x \in \{0,1\}^m$, we have*

$$\Pr_{(Z_1,\ldots,Z_t)\sim\mathsf{Samp}(U_n)}\left[\left\|\frac{1}{t}\sum_i f(Z_i) - \mathbb{E}\, f\right\| \le \varepsilon\right] \ge 1-\delta.$$

Extending the construction of non-explicit standard averaging samplers [CEG95], we can show that there exists a non-explicit matrix sampler that requires only an additional $2\log d$ bits of randomness compared to averaging samplers while achieving asymptotically optimal sample complexity.

Proposition 4. *There exists a (non-explicit) d-dimensional (δ, ε) matrix sampler for domain $\{0,1\}^m$ using $O(\frac{1}{\varepsilon^2}\log\frac{d}{\delta})$ samples and $m + 2\log\frac{1}{\delta} + 2\log d + \log\log\frac{d}{\varepsilon}$ random bits.*

Reference	Method	Random Bits	Sample Complexity
Proposition 4	Non-Explicit	$m + 2\log(1/\delta) + 2\log d$	$O(\log(d/\delta)/\varepsilon^2)$
[AW02]	Matrix Chernoff Bound	$O(m\log(d/\delta)/\varepsilon^2)$	$O(\log(d/\delta)/\varepsilon^2)$
[WX05]	Union Bound Over Entries	$m + O(\log(d/\delta))$	$O((d/\varepsilon)^{2+\alpha} \cdot \log^{1+\alpha}(1/\delta))$
[GLSS18]	Expander Walks	$m + O((1/\varepsilon^2) \cdot \log(d/\delta))$	$O(\log(d/\delta)/\varepsilon^2)$
Theorem 3	Iterated Sampler Composition	$m + O(\log(1/\delta) + \log d \log\log d)$	$O((\log(d/\delta)/\varepsilon^2)^{1+\alpha})$

Table 2: Comparison of matrix samplers, α any positive constant, $\varepsilon = 1/\mathrm{poly}(m)$, $\delta = \exp(-\mathrm{poly}(m))$, ignoring lower order terms. The complexity of the union bound sampler depends on the complexity of the "base" averaging sampler, and we use the bound in Theorem 1 here.

However, explicitly constructing randomness-efficient matrix samplers turns out to be very challenging. While a union bound over matrix entries suggests that a randomness-optimal averaging sampler directly implies a randomness-optimal matrix sampler (see Lemma 17), this method incurs an unavoidable d^2 factor in sample complexity, making the dependence on d exponentially worse than optimal. This raises an open question: can we construct a matrix sampler with (nearly) optimal randomness complexity and polynomial sample complexity, analogous to the averaging samplers in [BR94] and [Zuc97]?

Problem 2. *Can we explicitly design a d-dimensional (δ, ε) matrix sampler for domain $\{0,1\}^m$ that uses $O(m + \log(d/\delta))$ random bits and $\mathrm{poly}(1/\varepsilon, \log(1/\delta), \log d)$ samples?*

We summarize prior matrix sampler constructions in Table 2. The best existing construction, a matrix analog of the expander walks sampler, was provided by Garg, Lee, Song, and Srivastava [GLSS18]. Similar to expander walks for real-valued sampling, this construction gives asymptotically optimal sample complexity, but the randomness complexity is worse than optimal by a $\mathrm{poly}(1/\varepsilon)$ factor. We note that even if we allow the the matrix sampler to be non-averaging, no known better construction is currently known.

In this work, we construct a polynomial-time computable (δ, ε) matrix sampler with near-optimal randomness and sample complexity. The randomness complexity is optimal up to a logarithmic factor, and the sample complexity is within a $(\frac{1}{\varepsilon^2}\log\frac{d}{\delta})^{\alpha}$ factor of optimal for arbitrarily small constant $\alpha > 0$. This brings us close to resolving Problem 2.

Theorem 3. *For any constant $\alpha > 0$: There exists an efficient d-dimensional (δ, ε) matrix sampler for domain $\{0,1\}^m$ that uses $m + O(\log(1/\delta) + \log(d/\varepsilon)\log\log d)$ random bits and $O((\frac{1}{\varepsilon^2}\log\frac{d}{\delta})^{1+\alpha})$ samples.*

Additionally, we construct a matrix sampler achieving asymptotically optimal randomness complexity, though at the cost of increased sample complexity. This breaks the d^2 barrier in sample complexity for randomness-optimal matrix samplers.

Theorem 4. *For any constant $\alpha > 0$, there exists an efficient d-dimensional (δ, ε) matrix sampler for domain $\{0,1\}^m$ that uses $m + O(\log(d/\delta))$ random bits and $O(\frac{d^{\alpha}}{\varepsilon^{2+\alpha}}\log^{1+\alpha}\frac{1}{\delta})$ samples.*

1.3 Samplers for General Normed Vector Space

Apart from spectral norms of matrices, it is natural to study the averaging-sampling problem in other normed vector spaces V.

Definition 5. *A function* $\mathsf{Samp} : \{0,1\}^n \to (\{0,1\}^m)^t$ *is a* $(V, \mathcal{F})$*-sampler for a normed space* V *and a class of functions* $\mathcal{F} \subseteq \{f : \{0,1\}^m \to V\}$ *if, for every* $f \in \mathcal{F}$*,*

$$\Pr_{(Z_1,\ldots,Z_t)\sim\mathsf{Samp}(U_n)}\left[\left\|\frac{1}{t}\sum_i f(Z_i) - \mathbb{E}\, f\right\| \leq \varepsilon\right] \geq 1-\delta.$$

We call Samp *a* V*-sampler when* $\mathcal{F} = \{f : \{0,1\}^m \to V \mid \|f(x)\| \leq 1 \text{ for all } x\}$.

Under this definition, a d-dimensional matrix sampler is precisely a $(\mathbb{C}^{d\times d}, \|\cdot\|_2)$-sampler. Previous work also studied $(\mathbb{R}, \mathcal{F})$-samplers for a broader class of $\mathcal{F}$, such as subgaussian or subexponential real-valued functions [Bła19; Agr19]. Extending our construction to other normed spaces and broader function classes remains an interesting direction for future research.

1.4 Randomness Extractors

Our sampler construction has implications for randomness extractors. A randomness extractor is a function that extracts almost-uniform bits from a low-quality source of randomness. We define the quality of a random source as its min-entropy.

Definition 6. *The min-entropy of a random variable* X *is*

$$H_\infty(X) := \min_{x\in\mathrm{supp}(X)} \log\left(\frac{1}{\Pr[X=x]}\right).$$

An (n,k)*-source is a random variable on* n *bits with min-entropy at least* k*.*

Then a randomness extractor is defined as:

Definition 7 ([NZ96])**.** *A function* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is a* (k,ε) *extractor if for every* (n,k)*-source* X*, the distribution* $\mathsf{Ext}(X, U_d) \approx_\varepsilon U_m$*. We say* Ext *is a* strong (k,ε) *extractor if for every* (n,k)*-source* X*, the distribution* $(\mathsf{Ext}(X,Y), Y) \approx_\varepsilon U_{m+d}$*, where* Y *is chosen from* U_d*.*

Randomness extractors are essential tools in theoretical computer science. However, there has been little study of explicit extractors with the right dependence on ε for vanishing ε. This is a particular concern in cryptography, where extractors are widely used as building blocks and security requirements demand superpolynomially small ε [Lu02; Vad03; CDHKS00; DS02; KLRZ08; KLR09; DW09]. Existentially, there are extractors with seed length $d = \log(n-k) + 2\log(1/\varepsilon) + O(1)$, and there is a matching lower bound [RT00].

Zuckerman [Zuc97] showed that averaging samplers are essentially equivalent to extractors. Specifically, an extractor $\mathsf{Ext} : \{0,1\}^n \times [2^d] \to \{0,1\}^m$ can be seen as a sampler that generates $\mathsf{Ext}(X,i)$ as its i-th sample point using the random source X. Using this equivalence, we give the first extractor construction with optimal seed length up to an arbitrarily small constant factor bigger than 1, when the min-entropy $k = \beta n$ for a large enough constant $\beta < 1$.

Theorem 5. *For every constant* $\alpha > 0$*, there exists constant* $\beta < 1$ *such that for all* $\varepsilon > 0$ *and* $k \geq \beta n$*, there is an efficient strong* (k,ε) *extractor* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *with* $m = \Omega(k) - \log(1/\varepsilon)$ *and* $d = (1+\alpha)\log(n-k) + (2+\alpha)\log(1/\varepsilon) + O(1)$*.*

Prior to our work, extractors with a seed length dependence on ε achieving $2\log(1/\varepsilon)$ or close to it were based on the leftover hash lemma [ILL89; BBR88; IZ89; HILL99] and expander random walks [Gil98; Zuc07]. Extractors using the leftover hash lemma have a seed length of $n + 2\log(1/\varepsilon)$, which is far from optimal. Expander random walks give a (k,ε) extractor with $k > (1-\Omega(\varepsilon^2))n$

and an optimal seed length of $\log(n-k)+2\log(1/\varepsilon)+O(1)$. Our extractor is better than expander walks for all vanishing ε by allowing smaller entropy k.

In fact, if we aim to remove the α and achieve the optimal seed length of $\log(n-k)+2\log(1/\varepsilon)+O(1)$ to match expander random walks, we can set $s=1$ in Theorem 21 and get the following extractor for entropy rate $1-O(1/\log n)$ for $\varepsilon \geq 1/\mathrm{poly}(n)$:

Theorem 6. *There exists constant $\beta<1$ such that for all $\varepsilon>0$ and $k \geq (1-\frac{\beta}{\log n+\log(1/\varepsilon)})n$, there is an efficient strong (k,ε) extractor $\mathsf{Ext}:\{0,1\}^n\times\{0,1\}^d\to\{0,1\}^m$ with $m=\Omega(k)-\log^2(1/\varepsilon)$ and $d=\log(n-k)+2\log(1/\varepsilon)+O(1)$.*

This is better than expander random walks' entropy rate of $1-O(\varepsilon^2)$ for all $\varepsilon \leq o(1/\sqrt{\log n})$.

1.5 List-Decodable Codes

Another perspective on averaging samplers is its connection to error-correcting codes. Ta-Shma and Zuckerman [TZ04] showed that strong randomness extractors are equivalent to codes with good soft-decision decoding, which is related to list recovery. From this perspective, the composition scheme in our construction is similar to code concatenation.

For codes over the binary alphabet, soft decision decoding amounts to list decodability, which we focus on here.

We give good list-decodable codes without using the composition. That is, by just applying our almost ℓ-wise independence sampler on the binary alphabet, we can get a binary list-decodable code with rate $\Omega(\varepsilon^{2+\alpha})$ and non-trivial list size, although the list size is still exponential.

Theorem 7. *For every constant $\alpha>0$: there exists an explicit binary code with rate $\Omega(\varepsilon^{2+\alpha})$ that is $(\rho=\frac{1}{2}-\varepsilon, L)$ list-decodable with list size $L=2^{(1-c)n}$ for some constant $c=c(\alpha)>0$.*

Prior to our work, the best known code rate was $\Omega(\varepsilon^3)$ by Guruswami and Rudra [GR08]. We emphasize that their code achieved a list size of $L=\mathrm{poly}(n)$, while our list size is exponentially large, making our code unlikely to be useful.

1.6 Techniques

1.6.1 Averaging Samplers

Our construction of the averaging sampler is very simple, and is based on two observations:

1. Rather than querying every sample point produced by a sampler Samp, we can use an inner sampler Samp_{in} to select a subset of samples for querying. This sub-sampling approach has been utilized in previous sampler constructions [BR94; Gol11]. Although Samp_{in} incurs an additional randomness cost, the final sample complexity depends only on Samp_{in}, leading to reduced overall sample complexity. Since the domain of Samp_{in} is much smaller than the original domain, we can leverage more efficient sampling strategies.

2. The bottleneck of generating an almost ℓ-wise independent sequence over a large domain $\{0,1\}^m$ lies in sampling ℓ independent random points, which costs ℓm random bits. Since we can only afford $O(m)$ random bits, we are restricted to generating constant-wise independent samples. However, for a much smaller domain, we can use few random bits to generate an almost ℓ-wise independent sequence for large ℓ.

Our construction is outlined as follows. Let $\mathsf{Samp}_E : \{0,1\}^n \times [t'] \to \{0,1\}^m$ be the extractor-based sampler in [RVW00]. Let $Y_1, \ldots, Y_t$ be an almost ℓ-wise independent sequence over domain $[t']$, thinking of $t \ll t'$. Our sampler is then defined by

$$\mathsf{Samp} := (\mathsf{Samp}_E(X, Y_1), \mathsf{Samp}_E(X, Y_2), \ldots, \mathsf{Samp}_E(X, Y_t)).$$

In this construction, we use the almost ℓ-wise independent sequence to sub-sample from the extractor-based sampler. This can be viewed as a composition, similar to other cases such as Justesen codes [Jus72] and the first PCP theorem [ALMSS98], where the goal is to optimize two main parameters simultaneously by combining two simpler schemes, each optimizing one parameter without significantly compromising the other.

Previous works have applied almost ℓ-wise independence in extractor constructions. Srinivasan and Zuckerman [SZ99] proved a randomness-efficient leftover hash lemma by sampling an almost ℓ-wise independent function f using uniform seeds U_d and output $f(X)$, where X is the weak random source. From an extractor perspective, our inner sampler takes an inverse approach: we use X to pick a function f in the space of almost ℓ-wise independent functions, and then output $f(U_d)$. Furthermore, Raz's two-source extractor [Raz05] follows a more general framework, where two weak random sources to sample are used – one to sample an almost ℓ-wise independent function and the other as its input. However, directly applying Raz's error bound in our analysis (Lemma 20) results in a sample complexity that is off by a $\log(1/\delta)$ factor.

1.6.2 Matrix Samplers

Using the connection between averaging samplers and matrix samplers (see Lemma 17), our averaging sampler directly implies a (δ, ε) matrix sampler using $m + O(\log(d/\delta))$ random bits and $O((\frac{d^2}{\varepsilon^2} \log \frac{1}{\delta})^{1+\alpha})$ samples. This already gives the best randomness-optimal matrix sampler to date; however, its sample complexity has exponentially worse dependence on d than optimal.

Our sub-sampling technique using almost ℓ-wise independence offers a way to further reduce sample complexity. The composition of samplers only depends on the triangle inequality, which also applies to spectral norms. The remaining task is to verify that almost ℓ-wise independence also provides good concentration bounds for matrix sampling, which is straightforward given the extensive literature on moment inequalities for random matrices [CGT12; LT13; Tro+15].

Applying this composition, we get a (δ, ε) matrix sampler using $m + O(\log(d/\delta))$ random bits and $O((\frac{d^\alpha}{\varepsilon^{2+\alpha}} \log^{1+\alpha} \frac{1}{\delta}))$ samples, as described in Theorem 4. This is close to optimal for cases where $d < \mathrm{poly}(1/\varepsilon, \log(1/\delta))$, though it is not yet sufficient for larger d.

However, we can apply composition recursively. By repeating the composition $O(\log \log d)$ times, the dependence on d becomes $d^{\alpha^{O(\log \log d)}} = O(1)$. Each round of composition costs an additional $O(\log(d/\delta))$ random bits, resulting in a (δ, ε) matrix sampler using $m + O(\log(d/\delta) \log \log d)$ random bits and $O((\frac{1}{\varepsilon^2} \log \frac{d}{\delta})^{1+\alpha})$ samples. This already gives a matrix sampler using $m + \widetilde{O}(\log(d/\delta))$ random bits and near-optimal sample complexity.

To further improve the dependence on δ in randomness complexity and achieve the bound in Theorem 3, we introduce an alternative way of composing samplers:

Proposition 8. *Suppose we are given two efficient matrix samplers:*

- *Let* $\mathsf{Samp}_{out} : \{0,1\}^{n_1} \times [t_1] \to \{0,1\}^m$ *be a* $(\delta_1, \varepsilon_1)$ *matrix sampler.*
- *Let* $\mathsf{Samp}_{in} : \{0,1\}^{n_2} \times [t_2] \to \{0,1\}^{n_1}$ *be a* $(\delta_2, \varepsilon_2)$ ***averaging*** *sampler.*

Then, for uniformly random sources $X \sim U_{n_2}$,

$$\mathsf{Samp}(X) := (\mathsf{Samp}_{out}(\mathsf{Samp}_{in}(X, i), j))_{i\in[t_2], j\in[t_1]}$$

is an efficient $(\delta_2, 2\delta_1 + 2\varepsilon_2 + \varepsilon_1)$ matrix sampler for domain $\{0,1\}^m$ with $t_1 \cdot t_2$ samples using n_2 random bits.

This essentially says, composing a good $(\varepsilon, \varepsilon)$ matrix sampler Samp_{out} and a good (δ, ε) standard averaging sampler Samp_{in} would give a good $(\delta, O(\varepsilon))$ matrix sampler. Although this slightly increases the sample complexity, we can use our sub-sampling technique to reduce it later.

Unlike sub-sampling, in the composition of Proposition 8, Samp_{in} generates multiple random seeds for Samp_{out}, and we query all the samples it produces. This approach effectively reduces the error probability of Samp_{out} from ε to δ. The key idea is that only an $O(\varepsilon)$ fraction of the seeds generated by Samp_{in} lead to failure in Samp_{out}, contributing only a tolerable $O(\varepsilon)$ additive error in the estimate of $\mathbb{E}\, f$. The reasoning is straightforward: at most an ε fraction of all possible seeds for Samp_{out} cause failure, and with probability $1 - \delta$, Samp_{in} does not oversample these failure seeds by more than an additional ε proportion. As a result, the final proportion of failure seeds remains bounded by $O(\varepsilon)$.

Remark 9. *We can also define* strong matrix samplers *as a matrix analog of strong averaging samplers. All results for matrix samplers in this paper would hold for strong matrix samplers as well, with proofs following similar arguments. However, for simplicity, we present our results in the non-strong case only.*

2 Preliminaries

Notation. We use $[t]$ to represent set $\{1, \ldots, t\}$. For integer m, U_m is a random variable distributed uniformly over $\{0,1\}^m$. For random variables X and Y, we use $X \approx_\varepsilon Y$ to represent the statistical distance (total variation distance) between X and Y is at most ε, i.e.,

$$\max_{T\subseteq \mathrm{supp}(X)} \left| \Pr_{x\sim X}[x \in T] - \Pr_{y\sim Y}[y \in T] \right| \le \varepsilon.$$

We refer to an algorithm as "efficient" if it is polynomial-time computable. For simplicity, we omit domain sizes for samplers and matrix dimensions when context permits. Unless otherwise specified, statements such as "there exists a (δ, ε) sampler" mean that for all $0 < \delta \le \varepsilon < 1$, there exists a (δ, ε) sampler with the stated properties.

Remark 10. *The condition $\delta \le \varepsilon$ is very mild and holds in nearly all applications. This requirement can be relaxed to $\delta \le \varepsilon^\alpha$ for averaging samplers, and to $\delta \le d\varepsilon^\alpha$ for matrix samplers, where α is an arbitrarily small positive constant. Such relaxations do not alter the results.*

In the extreme case where $\delta > \varepsilon^\alpha$ for every constant $\alpha > 0$, pairwise independence is already a near-optimal averaging sampler (see Lemma 12). Specifically, this yields an efficient strong sampler with $O(1/(\delta\varepsilon^2)) \le O(1/\varepsilon^{2+\alpha})$ samples, using only $O(m + \log(1/\varepsilon))$ random bits. Similarly, for matrix samplers under the condition $\delta > d\varepsilon^\alpha$ for all $\alpha > 0$, pairwise independence also achieves near-optimal efficiency with $O(1/\varepsilon^{2+\alpha})$ samples and $O(m + \log(1/\varepsilon))$ random bits.

2.1 Extractor-Based Sampler

As mentioned above, averaging samplers are equivalent to extractors. We will introduce this in detail in Section 5.1. Reingold, Vadhan, and Wigderson used this equivalence to achieve the following:

Theorem 11 ([RVW00, Corollary 7.3], see also [Gol11, Theorem 6.1]). *For every constant $\alpha > 0$, there exists an efficient (δ, ε) averaging sampler over $\{0,1\}^m$ with $\mathrm{poly}(1/\varepsilon, \log(1/\delta))$ samples using $m + (1+\alpha)\cdot\log_2(1/\delta)$ random bits.*

For ease of presentation, we often denote an extractor-based averaging sampler by $\mathsf{Samp}_E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$, where $\mathsf{Samp}_E(X,i)$ is the i-th output sample point of the sampler using randomness input X. Therefore, the sample complexity of Samp_E is 2^d.

2.2 Almost ℓ-wise Independence

A sequence $Z_1, \ldots, Z_t$ is pairwise independent if the marginal distribution of every pair (Z_{i_1}, Z_{i_2}) is uniformly random. Chor and Goldreich [CG89] proved that using pairwise independence, we can have a sampler using few random bits but with unsatisfying sample complexity.

Lemma 12 ([CG89]). *For all $\delta, \varepsilon > 0$, there exists an efficient strong (δ, ε) averaging sampler for domain $\{0,1\}^m$ sampler with $O(1/(\delta\varepsilon^2))$ samples using $O(m + \log(1/\delta) + \log(1/\varepsilon))$ random bits.*

Generalizing pairwise independence, an almost ℓ-wise independent sequence is a sequence of random variables such that the marginal distribution of every ℓ of them is close to uniform.

Definition 13 ([NN93]). *A sequence of random variables $Z_1, \ldots, Z_t \in \{0,1\}^m$ is said to be γ-almost ℓ-wise independent if for all subsets $S \subseteq [t]$ such that $|S| \leq \ell$,*

$$(Z_i)_{i\in[S]} \approx_\gamma U_{m\times|S|}.$$

In particular, the pairwise independent sequence mentioned above is a 0-almost 2-wise independent sequence. Naor and Naor proved that such sequences can be randomness-efficiently generated.

Lemma 14 ([AGHP92]). *There exists an efficient algorithm that uses $(2+o(1))(\frac{\ell m}{2} + \log\log t) + 2\log\frac{1}{\gamma}$ random bits to generate a γ-almost ℓ-wise independent sequence $z_1, \ldots, z_t \in \{0,1\}^m$.*

Using standard techniques, we have the following concentration bound for almost ℓ-wise independent sequences (see Appendix B for the proof). Similar bounds for exact ℓ-wise independent sequences have been shown in [BR94; Dod00].

Lemma 15. *Let $Z_1, \ldots, Z_t \in \{0,1\}^m$ be a sequence of γ-almost ℓ-wise independent variables for an even integer ℓ. Then for every sequence of functions $f_1, \ldots, f_t : \{0,1\}^m \to [0,1]$,*

$$\Pr\left[\left|\frac{1}{t}\sum_{i=1}^{t}(f_i(Z_i) - \mathbb{E}\, f_i)\right| \leq \varepsilon\right] \geq 1 - \left(\frac{25\ell}{\varepsilon^2 t}\right)^{\ell/2} - \frac{\gamma}{\varepsilon^\ell}.$$

2.3 Composition of Samplers

The idea of composing samplers has been studied before. More specifically, Goldreich proved the following proposition.

Proposition 16 ([Gol11]). *Suppose we are given two efficient samplers:*

- *A (δ, ε) averaging sampler for domain $\{0,1\}^m$ with t_1 samples using n_1 random bits.*
- *A (δ', ε') averaging sampler for domain $\{0,1\}^{\log t_1}$ with t_2 samples using n_2 random bits.*

Then, there exists an efficient $(\delta + \delta', \varepsilon + \varepsilon')$ averaging sampler for domain $\{0,1\}^m$ with t_2 samples using $O(n_1 + n_2)$ random bits.

2.4 Averaging Samplers Imply Matrix Samplers

When Wigderson and Xiao first introduced matrix samplers, they observed that an averaging sampler also functions as a matrix sampler with weaker parameters, though they did not provide a formal proof. We formalize this observation below:

Lemma 17. *A (δ, ε) averaging sampler is a d-dimensional $(2d^2\delta, 2d\varepsilon)$ matrix sampler.*

The proof is presented in Appendix C.

3 Construction of Averaging Samplers

Our construction is based on a reduction lemma that constructs a sampler for domain $\{0,1\}^m$ based on a sampler for domain $\{0,1\}^{O(\log(1/\varepsilon)+\log\log(1/\delta))}$. We exploit the fact that when composing averaging samplers, the final sample complexity depends on only one of the samplers. Our strategy is:

- Apply the extractor sampler in Theorem 11 as a $(\delta/2, \varepsilon/2)$ sampler over domain $\{0,1\}^m$. This uses $m + O(\log(1/\delta))$ random bits and generates $\mathrm{poly}(1/\varepsilon, \log(1/\delta))$ samples.
- By Proposition 16, we only need to design a $(\delta/2, \varepsilon/2)$ averaging sampler over domain $\{0,1\}^{O(\log(1/\varepsilon)+\log\log(1/\delta))}$ using $O(\log(1/\delta))$ random bits. The total sample complexity will be equal to the sample complexity of this sampler. For this sampler, we use almost ℓ-wise independence.

Following the idea of Proposition 16, we first prove that composing samplers maintains the properties of a strong sampler.

Lemma 18 (Strong Composition)**.** *Suppose we are given two efficient averaging samplers:*

- *Let $\mathsf{Samp}_{out} : \{0,1\}^{n_1} \times [t_1] \to \{0,1\}^m$ be a (δ, ε) sampler.*
- *Let $\mathsf{Samp}_{in} : \{0,1\}^{n_2} \times [t_2] \to \{0,1\}^{\log t_1}$ be a strong (δ', ε') sampler.*

Then, for uniformly random sources $X_1 \sim U_{n_1}$ and $X_2 \sim U_{n_2}$,

$$\mathsf{Samp}(X_1 \circ X_2) := (\mathsf{Samp}_{out}(X_1, \mathsf{Samp}_{in}(X_2, i)))_{i \in [t_2]}$$

is an efficient $(\delta + \delta', \varepsilon + \varepsilon')$ strong averaging sampler for domain $\{0,1\}^m$ with t_2 samples using $n_1 + n_2$ random bits.

Proof. Let $f_1, \ldots, f_{t_2} : \{0,1\}^m \to [0,1]$ be an arbitrary sequence of functions, and define $f_{\mathrm{avg}} := \frac{1}{t_2}\sum_{i=1}^{t_2} f_i$. Since Samp_{out} is a (δ, ε) averaging sampler and f_{avg} is bounded in $[0,1]$, we have

$$\Pr_{X_1 \sim U_{n_1}}\left[\left|\mathop{\mathbb{E}}_{Y \sim U_{\log t_1}} f_{\mathrm{avg}}(\mathsf{Samp}_{out}(X_1, Y)) - \mathbb{E}\, f_{\mathrm{avg}}\right| \le \varepsilon\right] \ge 1 - \delta.$$

Equivalently, we can express this as

$$\Pr_{X_1 \sim U_{n_1}} \left[\left| \frac{1}{t_2} \sum_{i=1}^{t_2} \mathop{\mathbb{E}}_{Y \sim U_{\log t_1}} f_i(\mathsf{Samp}_{out}(X_1, Y)) - \frac{1}{t_2} \sum_{i=1}^{t_2} \mathbb{E}\, f_i \right| \leq \varepsilon \right] \geq 1 - \delta. \tag{1}$$

For an arbitrary x, view $f_i(\mathsf{Samp}_{out}(x, \cdot))$ as a Boolean function on domain $\{0,1\}^{\log t_1}$. Therefore, since $\mathsf{Samp}_{in}(X_2, 1), \ldots, \mathsf{Samp}_{in}(X_2, t_2)$ are generated by a strong (δ, ε) sampler,

$$\Pr_{X_2} \left[\left| \frac{1}{t_2} \sum_{i=1}^{t_2} \left(f_i(\mathsf{Samp}_{out}(x, \mathsf{Samp}_{in}(X_2, i))) - \mathop{\mathbb{E}}_{Y \sim U_{\log t_1}} f_i(\mathsf{Samp}_{out}(x, Y)) \right) \right| \leq \varepsilon' \right] \geq 1 - \delta'. \tag{2}$$

By the triangle inequality and a union bound over equations (1) and (2), we have

$$\Pr_{X_1, X_2} \left[\left| \frac{1}{t_2} \sum_{i=1}^{t_2} \left(f_i(\mathsf{Samp}_{out}(x, \mathsf{Samp}_{in}(X_2, i))) - \mathbb{E}\, f_i \right) \right| \leq \varepsilon' + \varepsilon \right] \geq 1 - \delta' - \delta.$$

This proves that the sampler we constructed is a strong $(\delta + \delta', \varepsilon + \varepsilon')$ averaging sampler. □

Instantiating Lemma 18 with the extractor-based sampler from Theorem 11 gives:

Lemma 19 (Reduction Lemma)**.** *For any $\alpha > 0$: For a sufficiently large constant $C > 0$, suppose there exists an efficient (δ', ε') averaging sampler Samp_{base} for domain $\{0,1\}^{C(\log(1/\varepsilon) + \log\log(1/\delta))}$ with t samples using n random bits. Then there exists an efficient $(\delta + \delta', \varepsilon + \varepsilon')$ averaging sampler Samp for domain $\{0,1\}^m$ with t samples using $m + (1+\alpha)\log(1/\delta) + n$ random bits. Moreover, if Samp_{base} is a strong sampler, then Samp is also strong.*

Proof. By Theorem 11, there exists an explicit (δ, ε) averaging sampler $\mathsf{Samp}_E : \{0,1\}^{n'} \times \{0,1\}^d \to \{0,1\}^m$ with

$$n' = m + (1+\alpha)(\log(1/\delta)) \quad \text{and} \quad d = \log(\mathrm{poly}(1/\varepsilon, \log(1/\delta))) \leq C(\log\log(1/\delta) + \log(1/\varepsilon))$$

when C is large enough. Therefore, Samp_{base} can work for domain $\{0,1\}^d$.

This enables us to apply Lemma 18. The total number of random bits used is $n' + n = m + (1+\alpha)\log(1/\delta) + n$, and a total of t samples are used.

□

Next, we show that for domain $\{0,1\}^m$ with $m \leq O(\log(1/\varepsilon) + \log\log(1/\delta))$, we can use an almost ℓ-wise independent sequence to design a strong averaging sampler with near-optimal sample complexity.

Lemma 20. *For any $2 \leq s < 1/\delta$, there exists an efficient strong (δ, ε) averaging sampler for domain $\{0,1\}^m$ with $O(\frac{s}{\varepsilon^2}\log\frac{1}{\delta})$ samples using $\frac{(2+o(1))(m+2\log(1/\varepsilon))\log(1/\delta)}{\log s} + 2\log(1/\delta)$ random bits.*

Proof. We begin by setting $\ell = \frac{2\log(2/\delta)}{\log s}$, $\gamma = \frac{\delta\varepsilon^\ell}{2}$, and $t = \frac{50s\log(2/\delta)}{\varepsilon^2 \log s}$. We then define our sampler by outputting a γ-almost ℓ-wise independent sequence $Z_1, \ldots, Z_t \in \{0,1\}^m$. Taking the parameters of Lemma 15, observe

$$\left(\frac{25\ell}{\varepsilon^2 t}\right)^{\ell/2} = \left(\frac{1}{s}\right)^{\ell/2} = \left(\frac{1}{s}\right)^{\frac{\log(2/\delta)}{\log s}} = \frac{\delta}{2},$$

and

$$\frac{\gamma}{\varepsilon^\ell} = \frac{\delta}{2}.$$

Therefore, for every sequence of functions $f_1, \ldots, f_t : \{0,1\}^m \to [0,1]$,

$$\Pr\left[\left|\frac{1}{t}\sum_{i=1}^{t}(f_i(Z_i) - \mathbb{E}\, f_i)\right| \le \varepsilon\right] \ge 1 - \delta.$$

Furthermore, Lemma 14 shows that we have an efficient algorithm that uses only

$$(2+o(1))\left(\frac{\ell m}{2} + \log\log t\right) + 2\cdot\log\frac{1}{\gamma} = \frac{(2+o(1))(m+2\log(1/\varepsilon))\log(1/\delta)}{\log s} + 2\cdot\log\frac{1}{\delta}$$

random bits to generate this γ-almost ℓ-wise independent sequence. □

We remark that the construction in Lemma 20 can be replaced with a perfectly ℓ-wise independent sequence. This yields a slightly weaker sampler that uses $O\left(\left(\frac{m+\log(1/\varepsilon)+\log\log(1/\delta)}{\log s}+1\right)\log\frac{1}{\delta}\right)$ random bits and $O\left(\frac{s}{\varepsilon^2}\cdot\log\frac{1}{\delta}\right)$ samples. This looser construction is already sufficient to establish Theorem 1. However, our tighter construction in Lemma 20 applies more broadly, particularly when m is small—for example, in Lemma 36.

Combining Lemma 19 and Lemma 20, we can prove our main result about averaging samplers:

Theorem 21. *For every constant $\alpha > 0$, and for any $m \ge 1$, $\delta, \varepsilon > 0$, and $2 \le s \le 1/\delta$, there exists an efficient strong (δ, ε) averaging sampler for domain $\{0,1\}^m$ that uses*

$$m + O\left(\frac{\log(1/\varepsilon)+\log\log(1/\delta)}{\log s}\cdot\log\frac{1}{\delta}\right) + (3+\alpha)\log\frac{1}{\delta}$$

random bits and $O((s/\varepsilon^2)\cdot\log(1/\delta))$ samples.

Proof. By Lemma 19, our goal is to design an efficient strong $(\delta/2, \varepsilon/2)$ averaging sampler Samp_{base} for domain $\{0,1\}^{C(\log(1/\varepsilon)+\log\log(1/\delta))}$ for some large enough constant C. By Lemma 20, for any $2 \le s < 1/\delta$, such sampler exists using $O(\frac{s}{\varepsilon^2}\log\frac{1}{\delta})$ samples and

$$O\left(\frac{(\log(1/\varepsilon)+\log\log(1/\delta))\log(1/\delta)}{\log s}\right) + 2\cdot\log\frac{1}{\delta}$$

Taking these into Lemma 19 gives us the desired bounds. □

For an arbitrarily small constant α, by setting $s = \varepsilon^{-2\alpha}\log^{\alpha}(1/\delta)$ in Theorem 21, we get Theorem 1 as a corollary:

Theorem 1. *For every constant $\alpha > 0$, there exists an efficient strong (δ, ε) averaging sampler for domain $\{0,1\}^m$ that uses $m + O(\log(1/\delta))$ random bits and $O((\frac{1}{\varepsilon^2}\log\frac{1}{\delta})^{1+\alpha})$ samples.*

We can also set $s = 2$ in Theorem 21 and get the following sampler with asymptotically optimal sample complexity but a worse randomness complexity.

Theorem 2. *There exists an efficient strong (δ, ε) averaging sampler for domain $\{0,1\}^m$ that uses $m + O(\log\frac{1}{\delta}(\log\frac{1}{\varepsilon} + \log\log\frac{1}{\delta}))$ random bits and $O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ samples.*

4 Construction of Matrix Samplers

Before moving further, we note that non-explicitly, a good matrix sampler exists. This generalizes the non-explicit sampler given in [CEG95], with the proof deferred to Appendix A.

Proposition 4. *There exists a (non-explicit) d-dimensional (δ, ε) matrix sampler for domain $\{0,1\}^m$ using $O(\frac{1}{\varepsilon^2}\log\frac{d}{\delta})$ samples and $m + 2\log\frac{1}{\delta} + 2\log d + \log\log\frac{d}{\varepsilon}$ random bits.*

Our improved averaging sampler directly implies the best randomness-optimal matrix sampler to date. Applying Lemma 17 to our sampler in Theorem 1 gives:

Lemma 22. *For every constant $\alpha > 0$, there exists an efficient d-dimensional (δ, ε) matrix sampler for domain $\{0,1\}^m$ using $m + O(\log(d/\delta))$ random bits and $O((\frac{d^2}{\varepsilon^2}\log\frac{1}{\delta})^{1+\alpha})$ samples.*

However, compared to the optimal sample complexity given in the non-explicit construction, our dependence on d is exponentially worse. As d is potentially very large, our goal is to utilize our composition to reduce the sample complexity while not increasing the randomness complexity too much.

4.1 One-Layer Composition

It is easy to verify that the composition lemma holds for matrices:

Lemma 23 (Matrix Composition)**.** *Suppose we are given two efficient matrix samplers:*

- *Let $\mathsf{Samp}_{out} : \{0,1\}^{n_1} \times [t_1] \to \{0,1\}^m$ be a $(\delta_1, \varepsilon_1)$ matrix sampler.*
- *Let $\mathsf{Samp}_{in} : \{0,1\}^{n_2} \times [t_2] \to \{0,1\}^{\log t_1}$ be a $(\delta_2, \varepsilon_2)$ matrix sampler.*

Then, for uniformly random sources $X_1 \sim U_{n_1}$ and $X_2 \sim U_{n_2}$,

$$\mathsf{Samp}(X_1 \circ X_2) := (\mathsf{Samp}_{out}(X_1, \mathsf{Samp}_{in}(X_2, i)))_{i\in[t_2]}$$

is an efficient $(\delta_1 + \delta_2, \varepsilon_1 + \varepsilon_2)$ matrix sampler for domain $\{0,1\}^m$ with t_2 samples using $n_1 + n_2$ random bits.

The proof is essentially the same as the proof of Lemma 18, since the triangle inequality applies to spectral norms, but since we are not dealing with the strong case we only have to do a union bound for two events, a bad sample from Samp_{out} and a bad sample from Samp_{in}.

The following lemma is the matrix version of Lemma 20, and we delay its proof to Section 4.4.

Lemma 24. *For any $2 \le s < d/\delta$, there exists an efficient d-dimensional (δ, ε) matrix sampler for domain $\{0,1\}^m$ with $O(\frac{s}{\varepsilon^2}\log\frac{d}{\delta})$ samples using $O(\frac{(m+\log(1/\varepsilon))\log(d/\delta)}{\log s} + \log(d/\delta))$ random bits.*

Composing Lemma 22 with Lemma 24 gives us the next theorem.

Lemma 25. *Suppose we have an efficient d-dimensional $(\delta_1, \varepsilon_1)$ matrix sampler for domain $\{0,1\}^m$ with t samples using n bits. For any constant $\alpha > 0$ such that $(t/\varepsilon_2)^\alpha \le d/\delta_2$, we can construct an efficient d-dimensional $(\delta_1 + \delta_2, \varepsilon_1 + \varepsilon_2)$ matrix sampler for domain $\{0,1\}^m$ with $O(\frac{t^\alpha}{\varepsilon_2^{2+\alpha}}\log\frac{d}{\delta_2})$ samples using $n + O(\log(d/\delta_2))$ bits.*

Proof. When $(t/\varepsilon_2)^\alpha \le d/\delta$, by setting $s = (t/\varepsilon_2)^\alpha$ in Lemma 24, we have a strong $(\delta_2, \varepsilon_2)$ matrix sampler for domain $\{0,1\}^{\log t}$ with $O(\frac{t^\alpha}{\varepsilon_2^{2+\alpha}} \log \frac{1}{\delta_2})$ samples using

$$O\left(\frac{(\log t + \log(1/\varepsilon_2))\log(d/\delta_2)}{\log s} + \log(d/\delta_2)\right) = O(\log(d/\delta_2))$$

random bits. Then by Lemma 23, we have the theorem we want. □

Theorem 4. *For any constant $\alpha > 0$, there exists an efficient d-dimensional (δ, ε) matrix sampler for domain $\{0,1\}^m$ that uses $m + O(\log(d/\delta))$ random bits and $O(\frac{d^\alpha}{\varepsilon^{2+\alpha}} \log^{1+\alpha} \frac{1}{\delta})$ samples.*

Proof. Set $\delta_1 = \delta_2 = \delta/2$ and $\varepsilon_1 = \varepsilon_2 = \varepsilon/2$ in Lemma 25 and apply it to Lemma 22 will give the result. □

4.2 Iterated Composition

Lemma 26. *There's an efficient d-dimensional (δ, ε) matrix sampler for domain $\{0,1\}^m$ using $m + O(\log(d/\delta)\log\log d)$ random bits and $O((\frac{\log\log d}{\varepsilon})^5 \log^2 \frac{d}{\delta})$ samples.*

Proof. Let $r = \log\log d$. We will prove that there exists a constant $C > 0$ such that for each $i \in [r]$, there exists an efficient $(\frac{i\delta}{r}, \frac{i\varepsilon}{r})$ matrix sampler using at most $m + i \cdot C \log \frac{d}{\delta}$ random bits and t_i samples, where

$$t_i = d^{2^{3-i}} \cdot C \cdot \frac{r^5}{\varepsilon^5} \log^2 \frac{d}{\delta}.$$

The $i = r$ case proves the lemma.

We will prove by induction on i from 1 to r.

Base Case ($i = 1$): By Lemma 22, there exists an efficient d-dimensional $(\frac{\delta}{r}, \frac{\varepsilon}{r})$ matrix sampler using $m + C_1(\log \frac{d}{\delta/r})$ random bits and $C_2 \frac{d^3}{(\varepsilon/r)^3} \log^{1.5} \frac{r}{\delta}$ samples for some constants $C_1, C_2 > 0$. When $C \ge 2C_1$ and $C \ge C_2$, we have

$$m + C_1\left(\log \frac{d}{\delta/r}\right) \le m + C\left(\log \frac{d}{\delta}\right)$$

and

$$C_2 \frac{d^3}{(\varepsilon/r)^3} \log^{1.5} \frac{r}{\delta} \le d^{2^2} \cdot C \cdot \frac{r^5}{\varepsilon^5} \log^2 \frac{d}{\delta} \le t_1.$$

Inductive Step: Assume that for some $i \in [1, r-1]$, there exists an efficient $(\frac{i\delta}{r}, \frac{i\varepsilon}{r})$ matrix sampler using $m + i \cdot C \log \frac{d}{\delta}$ random bits and t_i samples. By choosing some constant $\alpha < 1/2$ such that $(t_i r/\varepsilon)^\alpha < dr/\delta$ in Lemma 25, we have a $\left(\frac{(i+1)\delta}{r}, \frac{(i+1)\varepsilon}{r}\right)$ matrix sampler using $m + i \cdot C \log \frac{d}{\delta} + C_3 \log \frac{d}{\delta/r}$ random bits and $C_4 \frac{\sqrt{t_i}}{(\varepsilon/r)^{2.5}} \log \frac{d}{\delta/r}$ samples for some constants C_3 and C_4. When $C \ge 2C_3$ and $\sqrt{C} \ge 2C_4$, we have

$$m + i \cdot C \log \frac{d}{\delta} + C_3 \log \frac{d}{\delta/r} \le m + (i+1) \cdot C \log \frac{d}{\delta}$$

and

$$C_4 \frac{{t_i}^\alpha}{(\varepsilon/r)^{2+\alpha}} \log \frac{d}{\delta/r} \le 2C_4 \cdot \sqrt{C} \cdot \sqrt{d^{2^{3-i}}} \cdot \frac{r^5}{\varepsilon^5} \log^2 \frac{d}{\delta} \le d^{2^{3-(i+1)}} \cdot C \cdot \frac{r^5}{\varepsilon^5} \log^2 \frac{d}{\delta} \le t_{i+1}.$$

This finishes the induction and proves the lemma. □

Using Lemma 25, we have the following lemma:

Lemma 27. *For any constant $\alpha > 0$: There exists an efficient d-dimensional (δ, ε) matrix sampler for domain $\{0,1\}^m$ using $m + O(\log(d/\delta)\log\log d)$ random bits and $O((\frac{1}{\varepsilon^2}\log\frac{d}{\delta})^{1+\alpha})$ samples.*

Proof. Set $\delta_1 = \delta_2 = \delta/2$ and $\varepsilon_1 = \varepsilon_2 = \varepsilon/2$ in Lemma 25 and apply it to Lemma 26 will give the result. □

4.3 Another Composition Scheme

To further reduce the number of random bits used in Lemma 27, we introduce another way of composing matrix samplers. Instead of sub-sampling the samples, we sample the random seeds here.

Proposition 8. *Suppose we are given two efficient matrix samplers:*

- *Let $\mathsf{Samp}_{out} : \{0,1\}^{n_1} \times [t_1] \to \{0,1\}^m$ be a $(\delta_1, \varepsilon_1)$ matrix sampler.*
- *Let $\mathsf{Samp}_{in} : \{0,1\}^{n_2} \times [t_2] \to \{0,1\}^{n_1}$ be a $(\delta_2, \varepsilon_2)$ **averaging** sampler.*

Then, for uniformly random sources $X \sim U_{n_2}$,

$$\mathsf{Samp}(X) := (\mathsf{Samp}_{out}(\mathsf{Samp}_{in}(X, i), j))_{i\in[t_2], j\in[t_1]}$$

is an efficient $(\delta_2, 2\delta_1 + 2\varepsilon_2 + \varepsilon_1)$ matrix sampler for domain $\{0,1\}^m$ with $t_1 \cdot t_2$ samples using n_2 random bits.

Proof. Let $f : \{0,1\}^m \to \mathbb{C}^{d\times d}$ be a function such that $\|f(x)\| \leq 1$ for all $x \in \{0,1\}^m$. We define $h_f : \{0,1\}^{n_1} \to \{0,1\}$ as follows:

$$h_f(x) := \mathbf{1}\left[\left\|\frac{1}{t_1}\sum_{i=1}^{t_1} f(\mathsf{Samp}_{out}(x,i)) - \mathbb{E}\, f\right\| > \varepsilon_1\right].$$

Then $\mathbb{E}\, h_f < \delta_1$. One good property of h_f is that

$$\left\|\frac{1}{t_1}\sum_{i=1}^{t_1} f(\mathsf{Samp}_{out}(x,i)) - \mathbb{E}\, f\right\| \leq 2h_f(x) + \varepsilon_1.$$

For $Z_1, \ldots, Z_{t_2}$ the output of Samp_{in}, we have

$$\Pr\left[\left|\frac{1}{t_2}\sum_i h_f(Z_i) - \mathbb{E}\, h_f\right| \leq \varepsilon_2\right] \geq 1 - \delta_2.$$

Therefore, with $1 - \delta_2$ probability,

$$\frac{1}{t_2}\sum_i h_f(Z_i) < \delta_1 + \varepsilon_2.$$

Then we have

$$\begin{aligned}\left\|\frac{1}{t_1 t_2}\sum_{i=1}^{t_2}\sum_{j=1}^{t_1} f(\mathsf{Samp}_{out}(Z_i, j)) - \mathbb{E}\, f\right\| &\leq \frac{1}{t_2}\sum_{i=1}^{t_2}\left\|\frac{1}{t_1}\sum_{j=1}^{t_1} f(\mathsf{Samp}_{out}(Z_i, j)) - \mathbb{E}\, f\right\| \\ &\leq \frac{1}{t_2}\sum_{i=1}^{t_2}(2h_f(Z_i) + \varepsilon_1) \\ &\leq \varepsilon_1 + \frac{2}{t_2}\sum_i h_f(Z_i).\end{aligned}$$

This show that, with probability $1-\delta_2$, the error of Samp is at most $\varepsilon_1 + 2\delta_1 + 2\varepsilon_2$. □

Theorem 3. *For any constant $\alpha > 0$: There exists an efficient d-dimensional (δ, ε) matrix sampler for domain $\{0,1\}^m$ that uses $m + O(\log(1/\delta) + \log(d/\varepsilon)\log\log d)$ random bits and $O((\frac{1}{\varepsilon^2}\log\frac{d}{\delta})^{1+\alpha})$ samples.*

Proof. We apply Proposition 8 with the following choices:

- Samp_{out}: Use the $(\varepsilon/5, \varepsilon/5)$ matrix sampler for domain $\{0,1\}^m$ in Lemma 27 by choosing $\alpha = 0.5$. This uses $m + O(\log(d/\varepsilon)\log\log d)$ random bits and $O((\frac{1}{\varepsilon^2}\log\frac{d}{\varepsilon})^{1.5})$ samples.
- Samp_{in}: Use the $(\delta, \varepsilon/5)$ averaging sampler for domain $\{0,1\}^{m+O(\log(d/\varepsilon)\log\log d)}$ in Theorem 1 by choosing $\alpha = 0.5$. This uses $m + O(\log(1/\delta) + \log(d/\varepsilon)\log\log d)$ random bits and $O((\frac{1}{\varepsilon^2}\log\frac{1}{\delta})^{1.5})$ samples.

This gives as an efficient (δ, ε) matrix sampler using $m + O(\log(1/\delta) + \log(d/\varepsilon)\log\log d)$ random bits and $O((\frac{1}{\varepsilon^2}\log\frac{d}{\varepsilon})^{1.5} \cdot (\frac{1}{\varepsilon^2}\log\frac{1}{\delta})^{1.5})$ samples.

Set $\delta_1 = \delta_2 = \delta/2$ and $\varepsilon_1 = \varepsilon_2 = \varepsilon/2$ in Lemma 25 and apply it to this sampler will reduce the sample complexity to $O((\frac{1}{\varepsilon^2}\log\frac{d}{\delta})^{1+\alpha})$ for any constant $\alpha > 0$. □

4.4 Proof of Lemma 24

4.4.1 Concentration of Random Matrices

The goal of this section is to prove Lemma 30, an analog of Lemma 15 for random Hermitian matrices. Our approach follows standard techniques from the random matrix literature [Tom74; CGT12; LT13].

Lemma 28. *Let $\{X_i\}_{i\in[t]}$ be a sequence of independent, mean-zero, self-adjoint random matrices of size $d \times d$ such that $\|X_i\| < 1$ for every $i \in [t]$. Then, for every even interger $q \geq 2$, we have*

$$\mathbb{E}\left[\mathrm{Tr}\left(\left(\sum_{i=1}^{t}\varepsilon_i X_i\right)^q\right)\right] \leq d(qt)^{q/2},$$

where the sequence $\{\varepsilon_i\}$ consists of independent Rademacher random variables.

Proof. Note that

$$\mathbb{E}\left[\mathrm{Tr}\left(\left(\sum_{i=1}^{t}\varepsilon_i X_i\right)^q\right)\right] = \sum_{v=(v_1,\ldots,v_q)\in[t]^q}\mathbb{E}\left[\mathrm{Tr}(\varepsilon_{v_1}X_{v_1}\ldots\varepsilon_{v_q}X_{v_q})\right]$$

When some index appears an odd number of times in the word $v = (v_1, \ldots, v_q)$, the corresponding term vanishes. Hence we may restrict the sum to

$$\mathcal{T} \;=\; \{v \in [t]^q : \text{every index in } v \text{ appears an even number of times}\},$$

and we have

$$\mathbb{E}\left[\mathrm{Tr}\left(\left(\sum_{i=1}^{t}\varepsilon_i X_i\right)^q\right)\right] = \sum_{v\in\mathcal{T}}\mathbb{E}\left[\mathrm{Tr}(\varepsilon_{v_1}X_{v_1}\ldots\varepsilon_{v_q}X_{v_q})\right] \leq \sum_{v\in\mathcal{T}} d\,\mathbb{E}\left[\|X_{v_1}\ldots X_{v_q}\|\right] \leq \sum_{v\in\mathcal{T}} d,$$

where the last inequality comes from the fact that $\|X_i\| \leq 1$ always holds for every i.

It remains to bound $|\mathcal{T}|$. Since each index appearing in $v \in \mathcal{T}$ must occur an even number of times, one can group the q positions into exactly $q/2$ unordered pairs so that the two entries in each pair carry the same index. The number of ways to pair is bounded by $q^{q/2}$. For each pairing there are t choices of index per pair. Therefore,

$$|\mathcal{T}| \leq q^{q/2} \cdot t^{q/2} \leq (qt)^{q/2}.$$

Therefore,

$$\mathbb{E}\left[\mathrm{Tr}\left(\left(\sum_{i=1}^{t} \varepsilon_i X_i\right)^q\right)\right] \leq d(qt)^{q/2} \leq d|\mathcal{T}| \leq d(qt)^{q/2},$$

which proves the lemma. □

Using a standard symmetrization trick, we get the following lemma, which is an analog of Proposition 39:

Lemma 29. *Let $\{X_i\}_{i\in[t]}$ be a sequence of independent, mean-zero, self-adjoint random matrices of size $d \times d$ such that $\|X_i\| < 1$ for every $i \in [t]$. Then, for every even interger $q \geq 2$, we have*

$$\mathbb{E}\left[\mathrm{Tr}\left(\left(\sum_{i=1}^{t} X_i\right)^q\right)\right] \leq d\,(4qt)^{q/2}\,.$$

Proof. Let us write

$$S := \sum_{i=1}^{t} X_i, \qquad S' := \sum_{i=1}^{t} X_i',$$

where $\{X_i'\}_{i=1}^t$ is an independent copy of $\{X_i\}_{i=1}^t$, independent also of the original X_i's. Since each X_i has mean zero, we have

$$\mathbb{E}\left[\mathrm{Tr}\left(S^q\right)\right] = \mathbb{E}\left[\mathrm{Tr}\left((S - \mathbb{E}[S'])^q\right)\right] \leq \mathbb{E}\left[\mathrm{Tr}\left((S - S')^q\right)\right],$$

where the last inequality follows from Jensen's inequality.

Next, observe that

$$S - S' = \sum_{i=1}^{t} \left(X_i - X_i'\right).$$

Introduce a fresh sequence of Rademacher random variables $\{\varepsilon_i\}_{i=1}^t$, independent of everything else. Then, conditional on $\{X_i, X_i'\}_{i=1}^t$, the random matrix

$$\sum_{i=1}^{t} \varepsilon_i \left(X_i - X_i'\right)$$

has the same distribution as $S - S'$. Consequently,

$$\mathbb{E}\left[\mathrm{Tr}\left((S - S')^q\right)\right] = \mathop{\mathbb{E}}_{X,X'} \mathop{\mathbb{E}}_{\varepsilon}\left[\mathrm{Tr}\left(\left(\sum_{i=1}^{t} \varepsilon_i (X_i - X_i')\right)^q\right)\right].$$

Note that for each fixed (X, X', ε),

$$\mathrm{Tr}\left(\left(\sum_{i=1}^{t} \varepsilon_i (X_i - X_i')\right)^q\right) \leq 2^{q-1}\left(\mathrm{Tr}\left((\sum_{i=1}^{t} \varepsilon_i X_i)^q\right) + \mathrm{Tr}\left((\sum_{i=1}^{t} \varepsilon_i X_i')^q\right)\right).$$

Taking $\mathbb{E}_\varepsilon$, then $\mathbb{E}_{X,X'}$, and using that $\{X_i'\}$ has the same law as $\{X_i\}$ gives

$$\mathbb{E}\left[\mathrm{Tr}\left((S-S')^q\right)\right] = \mathop{\mathbb{E}}_{X,X'}\mathop{\mathbb{E}}_{\varepsilon}\left[\mathrm{Tr}\left((\sum_{i=1}^t \varepsilon_i(X_i - X_i'))^q\right)\right] \le 2^q \mathop{\mathbb{E}}_{X}\mathop{\mathbb{E}}_{\varepsilon}\left[\mathrm{Tr}\left((\sum_{i=1}^t \varepsilon_i X_i)^q\right)\right].$$

In combination with the symmetrization bound $\mathbb{E}[\mathrm{Tr}(S^q)] \le \mathbb{E}[\mathrm{Tr}((S-S')^q)]$, this yields

$$\mathbb{E}\left[\mathrm{Tr}(S^q)\right] \le 2^q \,\mathbb{E}\left[\mathrm{Tr}\left((\sum_{i=1}^t \varepsilon_i X_i)^q\right)\right].$$

Taking in the bound in Lemma 28, we have

$$\mathbb{E}\left[\mathrm{Tr}\left(\left(\sum_{i=1}^t X_i\right)^q\right)\right] \le 2^q \cdot d(qt)^{\frac{q}{2}} = d\left(2^2 qt\right)^{\frac{q}{2}} = d\,(4qt)^{\frac{q}{2}}.$$

□

Lemma 30. *Let $Z_1, \ldots, Z_t \in \{0,1\}^m$ be a sequence of γ-almost ℓ-wise independent variables for a positive even integer ℓ. Then for any $f : \{0,1\}^m \to \mathbb{C}^{d\times d}$ such that for all $x \in \{0,1\}^m$, $f(x)$ is Hermitian and $\|f(x)\| \le 1$, we have*

$$\Pr\left[\left\|\frac{1}{t}\sum_{i=1}^t f(Z_i) - \mathbb{E}\, f\right\| \le \varepsilon\right] \ge 1 - d\left(\frac{16\ell}{\varepsilon^2 t}\right)^{\ell/2} - \frac{2^\ell \gamma d}{\varepsilon^\ell}.$$

Proof. Let $W_i := (f(Z_i) - \mathbb{E}\, f)/2$. We have

$$\Pr\left[\left\|\frac{1}{t}\sum_{i=1}^t f(Z_i) - \mathbb{E}\, f\right\| > \varepsilon\right] = \Pr\left[\left\|\sum_{i=1}^t W_i\right\| > \frac{t\varepsilon}{2}\right] \le \frac{\mathbb{E}\left[\left\|\sum_{i=1}^t W_i\right\|^\ell\right]}{(t\varepsilon/2)^\ell}.$$

Note that each W_i is always a Hermitian matrix, so their sum is always Hermitian and therefore normal. Then we have

$$\left\|\sum_{i=1}^t W_i\right\|^\ell = \left\|\left(\sum_{i=1}^t W_i\right)^\ell\right\|.$$

Moreover, since ℓ is a positive even integer, $(\sum_{i=1}^t W_i)^\ell$ is positive semidefinite. Therefore,

$$\left\|\left(\sum_{i=1}^t W_i\right)^\ell\right\| \le \mathrm{Tr}\left(\left(\sum_{i=1}^t W_i\right)^\ell\right) = \mathrm{Tr}\left(\sum_{i_1,\ldots,i_\ell\in[t]} W_{i_1}W_{i_2}\ldots W_{i_\ell}\right) = \sum_{i_1,\ldots,i_\ell\in[t]} \mathrm{Tr}\left(W_{i_1}W_{i_2}\ldots W_{i_\ell}\right).$$

Let $W_1', \ldots, W_t'$ be a sequence of independent random variables where $W_i' := f_i(U_{\{0,1\}^m}) - \mathbb{E}\, f_i$.

Since the W_i's are γ-almost ℓ-wise independent and $|W_i| \le 1$, we have

$$\begin{aligned}
\mathbb{E}\left[\left\|\sum_{i=1}^t W_i\right\|^\ell\right] &\le \mathbb{E}\left[\sum_{i_1,\ldots,i_\ell\in[t]} \mathrm{Tr}\left(W_{i_1}W_{i_2}\ldots W_{i_\ell}\right)\right] \\
&= \sum_{i_1,\ldots,i_\ell\in[t]} \mathbb{E}\left[\mathrm{Tr}\left(W_{i_1}W_{i_2}\ldots W_{i_\ell}\right)\right] \\
&\le \sum_{i_1,\ldots,i_\ell\in[t]} \mathbb{E}\left[\mathrm{Tr}\left(W_{i_1}'W_{i_2}'\ldots W_{i_\ell}'\right)\right] + \gamma d t^\ell \\
&= \mathbb{E}\left[\mathrm{Tr}\left(\left(\sum_{i=1}^t W_i'\right)^\ell\right)\right] + \gamma d t^\ell.
\end{aligned}$$

Therefore, by Lemma 29, we have

$$\mathbb{E}\left[\left\|\sum_{i=1}^{t} W_i\right\|^{\ell}\right] \le \mathbb{E}\left[\operatorname{Tr}\left(\left(\sum_{i=1}^{t} W_i'\right)^{\ell}\right)\right] + \gamma d t^{\ell} \le d\,(4\ell t)^{\ell/2} + \gamma d t^{\ell}.$$

Hence,

$$\Pr\left[\left\|\sum_{i=1}^{t} W_i\right\| > \frac{t\varepsilon}{2}\right] \le \frac{\mathbb{E}\left[\left\|\sum_{i=1}^{t} W_i\right\|^{\ell}\right]}{(t\varepsilon/2)^{\ell}} \le d\left(\frac{16\ell}{\varepsilon^2 t}\right)^{\ell/2} + \frac{2^{\ell}\gamma d}{\varepsilon^{\ell}}.$$

□

4.4.2 Almost ℓ-wise independence for small domains

We first prove that the concentration analysis for Hermitian matrices directly implies the general case.

Lemma 31. *Let* $\mathsf{Samp} : \{0,1\}^n \to (\{0,1\}^m)^t$ *be a function. Suppose for any* $f : \{0,1\}^m \to \mathbb{C}^{2d\times 2d}$ *such that for all* $x \in \{0,1\}^m$, $f(x)$ *is Hermitian and* $\|f(x)\| \le 1$,

$$\Pr_{(Z_1,\ldots,Z_t)\sim\mathsf{Samp}(U_n)}\left[\left\|\frac{1}{t}\sum_i f(Z_i) - \mathbb{E}\, f\right\| \le \varepsilon\right] \ge 1-\delta.$$

Then Samp *is a* d*-dimensional* (δ,ε) *matrix sampler.*

Proof. Let $(Z_1,\ldots,Z_t) \sim \mathsf{Samp}(U_n)$. We are going to prove that for any function $f : \{0,1\}^m \to \mathbb{C}^{d\times d}$ such that $\|f(x)\| \le 1$ for all $x \in \{0,1\}^m$, we have

$$\Pr\left[\left\|\frac{1}{t}\sum_i f(Z_i) - \mathbb{E}\, f\right\| \le \varepsilon\right] \ge 1-\delta.$$

For any matrix $A \in \mathbb{C}^{d\times d}$, its Hermitian dilation $\mathcal{H}(A) \in \mathbb{C}^{2d\times 2d}$ is defined by

$$\mathcal{H}(A) := \begin{bmatrix} 0 & A \\ A^* & 0 \end{bmatrix}.$$

It is easy to verify that $\|A\| = \|\mathcal{H}(A)\|$. Then, for function $g : x \mapsto \mathcal{H}(f(x))$, we have

$$\Pr\left[\left\|\frac{1}{t}\sum_i g(Z_i) - \mathbb{E}\, g\right\| \le \varepsilon\right] \ge 1-\delta.$$

Note that we have

$$\left\|\frac{1}{t}\sum_i f(Z_i) - \mathbb{E}\, f\right\| = \left\|\mathcal{H}\left(\frac{1}{t}\sum_i f(Z_i) - \mathbb{E}\, f\right)\right\| = \left\|\frac{1}{t}\sum_i g(Z_i) - \mathbb{E}\, g\right\|.$$

Hence,

$$\Pr\left[\left\|\frac{1}{t}\sum_i f(Z_i) - \mathbb{E}\, f\right\| \le \varepsilon\right] \ge 1-\delta.$$

□

Now we are ready to prove Lemma 24.

Lemma 24. *For any $2 \le s < d/\delta$, there exists an efficient d-dimensional (δ, ε) matrix sampler for domain $\{0,1\}^m$ with $O(\frac{s}{\varepsilon^2}\log\frac{d}{\delta})$ samples using $O(\frac{(m+\log(1/\varepsilon))\log(d/\delta)}{\log s} + \log(d/\delta))$ random bits.*

Proof. We begin by setting $\ell = \frac{2\log(2d/\delta)}{\log s}$, $\gamma = \frac{\delta\varepsilon^\ell}{2^{\ell+1}d}$, and $t = \frac{16\ell s}{\varepsilon^2}$. We then define our sampler by outputting a γ-almost ℓ-wise independent sequence $Z_1, \ldots, Z_t \in \{0,1\}^m$. Taking the parameters of Lemma 30, observe

$$d\left(\frac{16\ell}{\varepsilon^2 t}\right)^{\ell/2} \le d\left(\frac{1}{s}\right)^{\ell/2} = d\left(\frac{1}{s}\right)^{\frac{\log(2d/\delta)}{\log s}} = \frac{\delta}{2},$$

and

$$\frac{2^\ell \gamma d}{\varepsilon^\ell} = \frac{\delta}{2}.$$

Let $\mathsf{Samp} : \{0,1\}^n \to (\{0,1\}^m)^t$ be a function. Suppose for any $f : \{0,1\}^m \to \mathbb{C}^{2d\times 2d}$ such that for all $x \in \{0,1\}^m$, $f(x)$ is Hermitian and $\|f(x)\| \le 1$,

$$\Pr_{(Z_1,\ldots,Z_t)\sim\mathsf{Samp}(U_n)}\left[\left\|\frac{1}{t}\sum_i f(Z_i) - \mathbb{E}\, f\right\| \le \varepsilon\right] \ge 1-\delta.$$

Then Samp is a d-dimensional (δ,ε) matrix sampler by Lemma 31. Therefore, by Lemma 30, we have

$$\Pr\left[\left|\frac{1}{t}\sum_{i=1}^t (f(Z_i) - \mathbb{E}\, f)\right| \le \varepsilon\right] \ge 1-\delta.$$

Our sampler uses

$$t = O\left(\frac{\ell s}{\varepsilon^2}\right) = O\left(\frac{s}{\varepsilon^2\log s}\log\frac{d}{\delta}\right)$$

samples. Furthermore, Lemma 14 shows that we have an efficient algorithm that uses only

$$\begin{aligned} O(\ell m + \log(1/\gamma) + \log\log t) &= O\left(\ell m + \ell\log(1/\varepsilon) + \log(d/\delta) + \log\log s\right) \\ &= O\left(\frac{(m+\log(1/\varepsilon))\log(d/\delta)}{\log s} + \log(d/\delta)\right) \end{aligned}$$

random bits to generate this γ-almost ℓ-wise independent sequence. □

Remark 32. *The initial work by Wigderson and Xiao [WX05] on matrix samplers focused on Hermitian matrices, where each $f(x)$ was assumed to be Hermitian. Nevertheless, as shown in Lemma 31, any sampler that works for Hermitian matrices can naturally be applied to general matrices as well.*

5 Applications to Extractors and Codes

5.1 Applications to Extractors

Zuckerman showed that averaging samplers are equivalent to randomness extractors [Zuc97]. Here we state the only direction that we need.

Lemma 33 ([Zuc97])**.** *An efficient strong (δ,ε) averaging sampler $\mathsf{Samp} : \{0,1\}^n \to (\{0,1\}^m)^t$ gives an efficient strong $(n - \log(1/\delta) + \log(1/\varepsilon), 2\varepsilon)$ extractor $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^{\log t} \to \{0,1\}^m$.*

Applying Lemma 33 on Theorem 1 gives Theorem 5:

Theorem 5. *For every constant $\alpha > 0$, there exists constant $\beta < 1$ such that for all $\varepsilon > 0$ and $k \geq \beta n$, there is an efficient strong (k, ε) extractor* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *with $m = \Omega(k) - \log(1/\varepsilon)$ and $d = (1+\alpha)\log(n-k) + (2+\alpha)\log(1/\varepsilon) + O(1)$.*

Proof. By Theorem 1, for any positive constant $\alpha > 0$, there exists a constant $\lambda > 1$ such that there exists an efficient strong (δ, ε) averaging sampler for domain $\{0,1\}^m$ with $O(\frac{1}{\varepsilon^{2+\alpha}} \log^{1+\alpha} \frac{1}{\delta})$ samples using $\lambda(m + \log \frac{1}{\delta})$ random bits.

To construct the required strong (k, ε) extractor for every n, we set δ such that $\log(1/\delta) = \frac{n}{2\lambda} + \log(1/\varepsilon)$. Then, we construct an efficient strong (δ, ε) sampler Samp for domain $\{0,1\}^m$ where

$$m = \frac{n}{\lambda} - \log(1/\delta) > \frac{n}{2\lambda} - \log(1/\varepsilon) = \Omega(n) - \log(1/\varepsilon).$$

By the above, Samp uses n random bits and generates $O(\frac{1}{\varepsilon^{2+\alpha}} \log^{1+\alpha} \frac{1}{\delta})$ samples.

By Lemma 33, Samp implies an efficient strong $(n - \log(1/\delta) + \log(1/\varepsilon), 2\varepsilon)$ extractor $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d \leq (1+\alpha)\log(n-k) + (2+\alpha)\log(1/\varepsilon) + O(1)$. It is only left to verify that $n - \log(1/\delta) + \log(1/\varepsilon) \leq \beta n$ for some constant $\beta < 1$. We have

$$n - \log(1/\delta) + \log(1/\varepsilon) = n - \frac{n}{2\lambda} \leq \frac{2\lambda - 1}{2\lambda} n.$$

This proves the theorem. □

If we would like an extractor with the optimal seed length of $d = \log(n-k) + 2\log(1/\varepsilon) + O(1)$, we can have the following extractor using Theorem 2.

Theorem 6. *There exists constant $\beta < 1$ such that for all $\varepsilon > 0$ and $k \geq (1 - \frac{\beta}{\log n + \log(1/\varepsilon)})n$, there is an efficient strong (k, ε) extractor* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *with $m = \Omega(k) - \log^2(1/\varepsilon)$ and $d = \log(n-k) + 2\log(1/\varepsilon) + O(1)$.*

Proof. By Theorem 2, there exists a constant $\lambda > 1$ such that there exists an efficient strong (δ, ε) averaging sampler for domain $\{0,1\}^m$ with $O(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$ samples using $m + \lambda \log \frac{1}{\delta}(\log\log(1/\delta) + \log(1/\varepsilon))$ random bits.

To construct the required strong (k, ε) extractor for every n, we set δ such that $\log(1/\delta) = \frac{1}{2\lambda}(\frac{n}{\log n + \log(1/\varepsilon)}) + \log(1/\varepsilon)$. Then, we construct an efficient strong (δ, ε) sampler Samp for domain $\{0,1\}^m$ where

$$\begin{aligned} m &= n - \lambda \log \frac{1}{\delta}(\log\log(1/\delta) + \log(1/\varepsilon)) \\ &\geq n - \frac{n}{2} \frac{\log\log(1/\delta) + \log(1/\varepsilon)}{\log n + \log(1/\varepsilon)} - \log^2(1/\varepsilon) - \log(1/\varepsilon)\log n \\ &\geq \Omega(n) - \log^2(1/\varepsilon). \end{aligned}$$

By the above, Samp uses n random bits and generates $O(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$ samples.

By Lemma 33, Samp implies an efficient strong $(n - \log(1/\delta) + \log(1/\varepsilon), 2\varepsilon)$ extractor $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = \log(n-k) + 2\log(1/\varepsilon) + O(1)$. It is only left to verify that $n - \log(1/\delta) + \log(1/\varepsilon) \leq (1 - \frac{\beta}{\log n + \log(1/\varepsilon)})n$ for some constant $\beta < 1$. We have

$$n - \log(1/\delta) + \log(1/\varepsilon) = n - \frac{1}{2\lambda}(\frac{n}{\log n + \log(1/\varepsilon)}) \leq (1 - \frac{1}{2\lambda(\log n + \log(1/\varepsilon))})n.$$

This proves the theorem. □

5.2 Application to List-Decodable Codes

Error-correcting codes are combinatorial objects that enable messages to be accurately transmitted, even when parts of the data get corrupted. Codes have been extensively studied and have proven to be extremely useful in computer science. Here we focus on the combinatorial property of list-decodability, defined below.

Definition 34. *A code* $\mathsf{ECC} : \{0,1\}^n \to (\{0,1\}^m)^t$ *is* (ρ, L) *list-decodable if for every received message* $r \in (\{0,1\}^m)^t$*, there are at most* L *messages* $x \in \{0,1\}^n$ *such that* $d_H(\mathsf{ECC}(x), r) \le \rho t$*, where* d_H *denotes the Hamming distance. A code is binary if* $m = 1$*.*

We focus on the binary setting, i.e., $m = 1$.

Lemma 35 ([TZ04])**.** *An efficient strong* (δ, ε) *averaging sampler* $\mathsf{Samp} : \{0,1\}^n \to \{0,1\}^t$ *over the binary domain gives an efficient binary code that is* $(\rho = \frac{1}{2} - \varepsilon, \delta 2^n)$ *list-decodable with code rate* $R = n/t$*.*

To construct our codes, we will use our almost ℓ-wise independence sampler in Lemma 20 directly.

Lemma 36. *For all constant* $\alpha > 0$*, there exists an efficient strong* (δ, ε) *averaging sampler for binary domain with* $O(\frac{1}{\varepsilon^{2+\alpha}} \log \frac{1}{\delta})$ *samples using* $n = C \log(1/\delta)$ *random bits for some constant* $C \ge 1$*.*

Proof. By setting $s = 1/\varepsilon^\alpha$ and $m = 1$ in Lemma 20, we have that whenever $1/\varepsilon^\alpha \le 1/\delta$, we have a strong (δ, ε) sampler with $O(\frac{1}{\varepsilon^2} \log \frac{1}{\delta})$ samples using $O(\log(1/\delta))$ random bits. When $1/\varepsilon^\alpha > 1/\delta$. Using the pairwise independence sampler in Lemma 12 for binary domain will satisfy the condition. □

Applying Lemma 35 to Lemma 36 gives Theorem 7:

Theorem 7. *For every constant* $\alpha > 0$*: there exists an explicit binary code with rate* $\Omega(\varepsilon^{2+\alpha})$ *that is* $(\rho = \frac{1}{2} - \varepsilon, L)$ *list-decodable with list size* $L = 2^{(1-c)n}$ *for some constant* $c = c(\alpha) > 0$*.*

Proof. We use the (δ, ε) sampler in Lemma 36, where we choose δ such that $n = C \log(1/\delta)$. Applying Lemma 35 to this sampler implies Theorem 7, where $c(\alpha) = 1/C$ here. □

6 Open Problems

Our work raises interesting open problems.

- Comparing to the sampler in [RVW00] which uses $m + (1+\alpha)\log(1/\delta)$ random bits, our averaging sampler requires $m + O(\log(1/\delta))$ random bits. Can we improve our randomness efficiency while maintaining a good sample complexity?
- Is there a way to eliminate the additional α in the sample complexity? For $\varepsilon = 1/\mathrm{poly}(m)$ and $\delta = \exp(-\mathrm{poly}(m))$, can we design an efficient averaging sampler that is asymptotically optimal in both randomness and sample complexity?
- Can we further improve the randomness complexity of our matrix samplers to fully resolve Problem 2?
- Is it possible to reduce the list size of the list-decodable codes in Theorem 7 to $\mathrm{poly}(n)$ using the structure of the list?
- Can we construct randomness-efficient V-samplers on other normed spaces V?

Acknowledgements

We thank Kuan Cheng for introducing us to the matrix sampler problem. We thank Shravas Rao for simplifying and slightly improving Lemma 30 (although this didn't improve our final result). We thank Oded Goldreich, Dana Moshkovitz, Amnon Ta-Shma, Salil Vadhan, and anonymous reviewers for helpful comments.

References

[AGHP92] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. "Simple constructions of almost k-wise independent random variables". In: *Random Structures & Algorithms* 3.3 (1992), pp. 289–304.

[Agr19] Rohit Agrawal. "Samplers and Extractors for Unbounded Functions". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*. Ed. by Dimitris Achlioptas and László A. Végh. Vol. 145. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019, 59:1–59:21. ISBN: 978-3-95977-125-2. DOI: 10.4230/LIPIcs.APPROX-RANDOM.2019.59. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.APPROX-RANDOM.2019.59.

[ALMSS98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. "Proof verification and the hardness of approximation problems". In: *J. ACM* 45.3 (May 1998), pp. 501–555. ISSN: 0004-5411. DOI: 10.1145/278298.278306. URL: https://doi.org/10.1145/278298.278306.

[AW02] Rudolf Ahlswede and Andreas Winter. "Strong converse for identification via quantum channels". In: *IEEE Transactions on Information Theory* 48.3 (2002), pp. 569–579.

[BBR88] Charles H Bennett, Gilles Brassard, and Jean-Marc Robert. "Privacy amplification by public discussion". In: *SIAM journal on Computing* 17.2 (1988), pp. 210–229.

[BGG93] Mihir Bellare, Oded Goldreich, and Shafi Goldwasser. "Randomness in interactive proofs". In: *Computational Complexity* 3 (1993), pp. 319–354.

[Bła19] Jarosław Błasiok. "Optimal streaming and tracking distinct elements with high probability". In: *ACM Transactions on Algorithms (TALG)* 16.1 (2019), pp. 1–28.

[BR94] Mihir Bellare and John Rompel. "Randomness-efficient oblivious sampling". In: *Proceedings 35th Annual Symposium on Foundations of Computer Science*. IEEE. 1994, pp. 276–287.

[CDHKS00] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. "Exposure-Resilient Functions and All-or-Nothing Transforms". In: *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceedings*. Springer, 2000, pp. 453–469.

[CEG95] Ran Canetti, Guy Even, and Oded Goldreich. "Lower bounds for sampling algorithms for estimating the average". In: *Information Processing Letters* 53.1 (1995), pp. 17–25.

[CG89] Benny Chor and Oded Goldreich. "On the power of two-point based sampling". In: *Journal of Complexity* 5.1 (1989), pp. 96–106.

[CGT12] Richard Y Chen, Alex Gittens, and Joel A Tropp. "The masked sample covariance estimator: an analysis using matrix concentration inequalities". In: *Information and Inference: A Journal of the IMA* 1.1 (2012), pp. 2–20.

[Dod00] Yevgeniy Dodis. *PhD thesis: Exposure-resilient cryptography*. English (US). Massachusetts Institute of Technology, Sept. 2000.

[DS02] Yevgeniy Dodis and Joel Spencer. "On the (Non)Universality of the One-Time Pad". In: *43rd Symposium on Foundations of Computer Science (FOCS 2002), 16-19 November 2002, Vancouver, BC, Canada, Proceedings*. IEEE Computer Society, 2002, pp. 376–385.

[DW09] Yevgeniy Dodis and Daniel Wichs. "Non-Malleable Extractors and Symmetric Key Cryptography from Weak Secrets". In: *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*. ACM, 2009, pp. 601–610.

[Gil98] David Gillman. "A Chernoff bound for random walks on expander graphs". In: *SIAM Journal on Computing* 27.4 (1998), pp. 1203–1220.

[GLSS18] Ankit Garg, Yin Tat Lee, Zhao Song, and Nikhil Srivastava. "A matrix expander chernoff bound". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*. 2018, pp. 1102–1114.

[Gol11] Oded Goldreich. "A sample of samplers: A computational perspective on sampling". In: *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation: In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*. Springer, 2011, pp. 302–332.

[GR08] Venkatesan Guruswami and Atri Rudra. "Explicit Codes Achieving List Decoding Capacity: Error-Correction With Optimal Redundancy". In: *IEEE Transactions on Information Theory* 54.1 (2008), pp. 135–150. DOI: 10.1109/TIT.2007.911222.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A Levin, and Michael Luby. "A pseudorandom generator from any one-way function". In: *SIAM Journal on Computing* 28.4 (1999), pp. 1364–1396.

[ILL89] R. Impagliazzo, L. A. Levin, and M. Luby. "Pseudo-random generation from one-way functions". In: *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*. STOC '89. Seattle, Washington, USA: Association for Computing Machinery, 1989, pp. 12–24. ISBN: 0897913078. DOI: 10.1145/73007.73009. URL: https://doi.org/10.1145/73007.73009.

[IZ89] Russell Impagliazzo and David Zuckerman. "How to recycle random bits". In: *FOCS*. Vol. 30. 1989, pp. 248–253.

[Jus72] Jørn Justesen. "Class of constructive asymptotically good algebraic codes". In: *IEEE Transactions on information theory* 18.5 (1972), pp. 652–656.

[KLR09] Yael Tauman Kalai, Xin Li, and Anup Rao. "2-Source Extractors under Computational Assumptions and Cryptography with Defective Randomness". In: *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*. IEEE Computer Society, 2009, pp. 617–626.

[KLRZ08] Yael Tauman Kalai, Xin Li, Anup Rao, and David Zuckerman. "Network Extractor Protocols". In: *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*. IEEE Computer Society, 2008, pp. 654–663.

[LT13] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

[Lu02] Chin-Laung Lu. "Hyper-encryption against space-bounded adversaries from on-line strong extractors". In: *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 2002, Proceedings*. Springer, 2002, pp. 257–271.

[NN93] Joseph Naor and Moni Naor. "SMALL-BIAS PROBABILITY SPACES-EFFICIENT CONSTRUCTIONS AND APPLICATIONS". In: *SIAM Journal on Computing* 22.4 (1993), pp. 838–856.

[NZ96] Noam Nisan and David Zuckerman. "Randomness is linear in space". In: *Journal of Computer and System Sciences* 52.1 (1996), pp. 43–52.

[Raz05] Ran Raz. "Extractors with weak random seeds". In: *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*. 2005, pp. 11–20.

[RL01] Yao-Feng Ren and Han-Ying Liang. "On the best constant in Marcinkiewicz–Zygmund inequality". In: *Statistics & probability letters* 53.3 (2001), pp. 227–233.

[RT00] Jaikumar Radhakrishnan and Amnon Ta-Shma. "Bounds for dispersers, extractors, and depth-two superconcentrators". In: *SIAM Journal on Discrete Mathematics* 13.1 (2000), pp. 2–24.

[Rud99] Mark Rudelson. "Random vectors in the isotropic position". In: *Journal of Functional Analysis* 164.1 (1999), pp. 60–72.

[RVW00] Omer Reingold, Salil Vadhan, and Avi Wigderson. "Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors". In: *Proceedings 41st Annual Symposium on Foundations of Computer Science*. IEEE. 2000, pp. 3–13.

[SZ99] Aravind Srinivasan and David Zuckerman. "Computing with very weak random sources". In: *SIAM Journal on Computing* 28.4 (1999), pp. 1433–1459.

[Tom74] Nicole Tomczak-Jaegermann. "The moduli of smoothness and convexity and the Rademacher averages of the trace classes $S_{\{}p\}$ $(1 \leq p < \infty)$". eng. In: *Studia Mathematica* 50.2 (1974), pp. 163–182. URL: http://eudml.org/doc/217886.

[Tro+15] Joel A Tropp et al. "An introduction to matrix concentration inequalities". In: *Foundations and Trends® in Machine Learning* 8.1-2 (2015), pp. 1–230.

[Tro12] Joel A Tropp. "User-friendly tail bounds for sums of random matrices". In: *Foundations of computational mathematics* 12 (2012), pp. 389–434.

[TZ04] Amnon Ta-Shma and David Zuckerman. "Extractor codes". In: *IEEE transactions on information theory* 50.12 (2004), pp. 3015–3025.

[Vad03] Salil P. Vadhan. "On constructing locally computable extractors and cryptosystems in the bounded storage model". In: *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*. Springer, 2003, pp. 61–77.

[Vad07] Salil Vadhan. "The unified theory of pseudorandomness: guest column". In: *ACM SIGACT News* 38.3 (2007), pp. 39–54.

[Vad12] Salil P. Vadhan. "Pseudorandomness". In: *Foundations and Trends® in Theoretical Computer Science* 7.1–3 (2012), pp. 1–336.

[WX05] Avi Wigderson and David Xiao. "A randomness-efficient sampler for matrix-valued functions and applications". In: *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS'05)*. IEEE. 2005, pp. 397–406.

[Zuc07] David Zuckerman. "Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number". In: *Theory of Computing* 3.6 (2007), pp. 103–128. DOI: 10.4086/toc.2007.v003a006. URL: https://theoryofcomputing.org/articles/v003a006.

[Zuc97] David Zuckerman. "Randomness-optimal oblivious sampling". In: *Random Structures & Algorithms* 11.4 (1997), pp. 345–367.

A Proof of Proposition 4

In this section, we extend the non-explicit averaging sampler construction from [CEG95] to matrix samplers.

Lemma 37. *Let* Samp *be a d-dimensional (δ, ε) matrix sampler for domain $\{0,1\}^m$ using t samples. Then there exists a d-dimensional $(2\delta, 3\varepsilon)$ matrix sampler for domain $\{0,1\}^m$ using $m + 2\log\frac{1}{\delta} + 2\log d + \log\log\frac{d}{\varepsilon} + 1$ random bits and t samples.*

Proof. Our goal is to construct a $(2\delta, 3\varepsilon)$ matrix sampler Samp' based on Samp.

Output Approximation via Discretization. We construct a discretization grid $G \subset \mathbb{C}$ by

$$G := \{a\Delta + b\Delta i \mid a, b \in \mathbb{Z} \text{ and } |a|^2 + |b|^2 \leq \Delta^{-2}\},$$

where $\Delta = \frac{\varepsilon}{d}$. For each $x \in \{0,1\}^m$, define an approximation function f' that rounds each entry of $f(x)$ to the nearest point in G, yielding $f' : \{0,1\}^m \to G^{d\times d}$. Since each entry in $f(x)$ differs from $f'(x)$ by at most Δ, the total approximation error per matrix (in spectral norm) is bounded by $d\Delta \leq \varepsilon$ according to Proposition 40. Thus, f' has an average that approximates the average of f within ε, and the set of all such approximations f' forms a finite function class, which we denote F.

Bounding the Size of F. Each entry of a matrix in $G^{d\times d}$ has at most $1/\Delta^2$ possible values. The number of possible matrices is therefore bounded by

$$\left(\frac{1}{\Delta^2}\right)^{d^2} = \left(\frac{d}{\varepsilon}\right)^{2d^2},$$

so the total number of functions in F is

$$|F| \leq \left(\left(\frac{d}{\varepsilon}\right)^{2d^2}\right)^{2^m} = 2^{2^{m+1}d^2\log\frac{d}{\varepsilon}}.$$

Probabilistic Reduction of Random Bits. For each function $f' \in F$, let a random seed be called *bad* if the estimate of Samp deviates from the true average of f' by more than ε. Since Samp is a (δ, ε)-sampler, the fraction of bad random seeds for any $f' \in F$ is at most δ. By Hoeffding's inequality, if we select k random seeds independently at random, the probability that more than $2\delta k$ of them are bad is at most $2e^{-2\delta^2 k}$. Applying a union bound over all $f' \in F$, the probability that there exists any f' with more than $2\delta k$ bad seeds is at most $|F| \cdot 2e^{-2\delta^2 k}$.

Choosing k and Applying Probabilistic Method. Set

$$k = \frac{\ln|F| + \ln 2.01}{2\delta^2} \leq \frac{2^m d^2 \log \frac{d}{\varepsilon} + 1}{\delta^2}$$

so that $|F| \cdot 2e^{-2\delta^2 k} < 1$. With this choice, there exists a set K of k random seeds such that, for all $f' \in F$, the fraction of bad seeds in K is at most 2δ. The number of random bits required to select a sequence $\rho \in K$ is

$$\log k \leq m + 2\log\frac{1}{\delta} + 2\log d + \log\log\frac{d}{\varepsilon} + 1.$$

Defining the New Sampler Samp'. We define Samp' as follows: Select a random seed $\rho \in K$, and run $\mathsf{Samp}(\rho)$ to get samples $Z_1, \ldots, Z_t \in \{0,1\}^m$. With $1 - 2\delta$ probability, we have

$$\left\| \frac{1}{t}\sum_{i=1}^{t} f(Z_i) - \mathbb{E}\, f \right\| \leq \left\| \frac{1}{t}\sum_{i=1}^{t} \left((f(Z_i) - f'(Z_i)) + (f'(Z_i) - \mathbb{E}\, f') + (\mathbb{E}\, f' - \mathbb{E}\, f) \right) \right\| \leq 3\varepsilon.$$

Samp' is then a $(2\delta, 3\varepsilon)$ matrix sampler, using only

$$m + 2\log\frac{1}{\delta} + 2\log d + \log\log\frac{d}{\varepsilon} + 1$$

random bits. This completes the proof. □

Theorem 38 (Matrix Chernoff Bound, see [Tro+15])**.** *Let $X_1, \ldots, X_k$ be independent $d \times d$ complex random matrices. Suppose $\|X_i\| \leq 1$ for all $i \in [k]$. Then, for any $\varepsilon > 0$, the following inequality holds:*

$$\Pr\left(\left\| \frac{1}{t}\sum_{i=1}^{t} (X_i - \mathbb{E}[X_i]) \right\| > \varepsilon \right) \leq 2d \cdot \exp\left(-\frac{3}{8} t\varepsilon^2 \right).$$

Applying matrix chernoff bound, we can prove Proposition 4.

Proposition 4. *There exists a (non-explicit) d-dimensional (δ, ε) matrix sampler for domain $\{0,1\}^m$ using $O(\frac{1}{\varepsilon^2}\log\frac{d}{\delta})$ samples and $m + 2\log\frac{1}{\delta} + 2\log d + \log\log\frac{d}{\varepsilon}$ random bits.*

Proof. By Theorem 38, taking $t = \frac{24}{\varepsilon^2}\log\frac{4d}{\delta}$ independent samples in $\{0,1\}^m$ would give a d-dimensional $(\delta/2, \varepsilon/3)$ matrix sampler for domain $\{0,1\}^m$ using t samples and tm random bits. Applying Lemma 37, we get a d-dimensional (δ, ε) matrix sampler for domain $\{0,1\}^m$ using $O(\frac{1}{\varepsilon^2}\log\frac{d}{\delta})$ samples and $m + 2\log\frac{1}{\delta} + 2\log d + \log\log\frac{d}{\varepsilon}$ random bits. □

B Proof of Lemma 15

Proposition 39 (Marcinkiewicz–Zygmund inequality [RL01])**.** *Let $\{X_i, i \geq 1\}$ be a sequence of independent random variables with $\mathbb{E}\, X_i = 0$, $\mathbb{E}|X_i|^p < \infty$. Then for $p \geq 2$:*

$$\mathbb{E}\left|\sum_{i=1}^{n} X_i\right|^p \leq C(p) n^{p/2-1} \sum_{i=1}^{n} \mathbb{E}|X_i|^p,$$

where $C(p) \leq (3\sqrt{2})^p p^{p/2}$.

Lemma 15. *Let $Z_1, \ldots, Z_t \in \{0,1\}^m$ be a sequence of γ-almost ℓ-wise independent variables for an even integer ℓ. Then for every sequence of functions $f_1, \ldots, f_t : \{0,1\}^m \to [0,1]$,*

$$\Pr\left[\left|\frac{1}{t}\sum_{i=1}^{t}(f_i(Z_i) - \mathbb{E}\, f_i)\right| \leq \varepsilon\right] \geq 1 - \left(\frac{25\ell}{\varepsilon^2 t}\right)^{\ell/2} - \frac{\gamma}{\varepsilon^\ell}.$$

Proof. Let $W_i := f_i(Z_i) - \mathbb{E}\, f_i$. We have

$$\Pr\left[\left|\sum_{i=1}^{t} W_i\right| > t\varepsilon\right] \leq \frac{\mathbb{E}\left[\left|\sum_{i=1}^{t} W_i\right|^\ell\right]}{(t\varepsilon)^\ell}.$$

Let $W_1', \ldots, W_t'$ be a sequence of independent random variables where $W_i' := f_i(U_{\{0,1\}^m}) - \mathbb{E}\, f_i$. Since the W_i's are γ-almost ℓ-wise independent and $|W_i| \leq 1$, we have

$$\mathbb{E}\left[\left|\sum_{i=1}^{t} W_i\right|^\ell\right] = \mathbb{E}\left[\left(\sum_{i=1}^{t} W_i\right)^\ell\right] \leq \mathbb{E}\left[\left(\sum_{i=1}^{t} W_i'\right)^\ell\right] + \gamma t^\ell = \mathbb{E}\left[\left|\sum_{i=1}^{t} W_i'\right|^\ell\right] + \gamma t^\ell.$$

Since $\mathbb{E}\, W_i' = 0$ and $|W_i'| \leq 1$, they satisfy the conditions for Marcinkiewicz–Zygmund inequality. We have

$$\mathbb{E}\left[\left|\sum_{i=1}^{t} W_i'\right|^\ell\right] \leq (3\sqrt{2})^\ell \ell^{\ell/2} t^{\ell/2-1} \sum_{i=1}^{t} \mathbb{E}|W_i'|^\ell \leq (5\sqrt{\ell t})^\ell.$$

Therefore,

$$\frac{\mathbb{E}\left[\left|\sum_{i=1}^{t} W_i\right|^\ell\right]}{(t\varepsilon)^\ell} \leq \left(\frac{25\ell}{\varepsilon^2 t}\right)^{\ell/2} + \frac{\gamma}{\varepsilon^\ell}.$$

□

C Proof of Lemma 17

To prove Lemma 17, we need the following property of matrix norms:

Proposition 40. *Let $A \in \mathbb{C}^{d\times d}$ and define*

$$r = \max_{i,j} |A_{ij}|.$$

Then the spectral norm of A satisfies

$$r \leq \|A\| \leq dr.$$

Proof. Select standard basis vectors $e_i, e_j \in \mathbb{C}^d$ such that $|A_{ij}| = r$. Then,

$$\|A\| \geq \frac{|e_i^* A e_j|}{\|e_i\|_2 \|e_j\|_2} = |A_{ij}| = r.$$

We also have

$$\|A\| \leq \|A\|_F = \sqrt{\sum_{i=1}^{d} \sum_{j=1}^{d} A_{ij}^2} \leq dr.$$

□

Lemma 17. *A (δ, ε) averaging sampler is a d-dimensional $(2d^2\delta, 2d\varepsilon)$ matrix sampler.*

Proof. Let $Z_1, \ldots, Z_t$ be the sampler's output. We define

$$A := \frac{1}{t} \sum_{i=1}^{t} f(Z_i).$$

Now we fix some $i, j \in [d]$. For all $x \in \{0,1\}^m$, we have $|f(x)_{ij}| \leq \|f(x)\| \leq 1$ by Proposition 40. Then, since $Z_i's$ are the output of a (δ, ε) averaging sampler, we have

$$\Pr\left[|\mathrm{Re}(A_{ij}) - \mathrm{Re}((\mathbb{E}\, f)_{ij})| \leq \varepsilon\right] \geq 1 - \delta \quad \text{and} \quad \Pr\left[|\mathrm{Im}(A_{ij}) - \mathrm{Im}((\mathbb{E}\, f)_{ij})| \leq \varepsilon\right] \geq 1 - \delta,$$

where $\mathrm{Re}(x)$ and $\mathrm{Im}(x)$ are the functions that extract the real part and imaginary part of x respectively. Take a union bound, we have with $1 - 2\delta$ probability,

$$|A_{ij} - (\mathbb{E}\, f)_{ij}| \leq |\mathrm{Re}(A_{ij} - (\mathbb{E}\, f)_{ij})| + |\mathrm{Im}(A_{ij} - (\mathbb{E}\, f)_{ij})| \leq 2\varepsilon.$$

By a union bound over all entries, with $1 - 2d^2\delta$ probability, all entries have an additive error bounded by 2ε, and this implies that $\|A - \mathbb{E}\, f\| \leq 2d\varepsilon$ by Proposition 40. □

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il