$\mathsf{BPL} \subseteq \mathsf{L\text{-}AC}^1$

Kuan Cheng
Center on Frontiers of Computing Studies, School of Computer Science, Peking University, Beijing, China

Yichuan Wang
Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China

Abstract

Whether $\mathsf{BPL} = \mathsf{L}$ (which is conjectured to be equal) or even whether $\mathsf{BPL} \subseteq \mathsf{NL}$, is a big open problem in theoretical computer science. It is well known that $\mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{L\text{-}AC}^1$. In this work we show that $\mathsf{BPL} \subseteq \mathsf{L\text{-}AC}^1$ also holds. Our proof is based on a new iteration method for boosting precision in approximating matrix powering, which is inspired by the Richardson Iteration method developed in a recent line of work [1, 28, 10, 17, 12, 25, 8]. We also improve the algorithm for approximate counting in low-depth $\mathsf{L\text{-}AC}$ circuits from an *additive error* setting to a *multiplicative error* setting.

2012 ACM Subject Classification Theory of computation → Pseudorandomness and derandomization; Theory of computation → Computational complexity and cryptography

Keywords and phrases Randomized Space Complexity, Circuit Complexity, Derandomization

Digital Object Identifier 10.4230/LIPIcs.CCC.2024.32

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2024/048/`

Acknowledgements We thank anonymous reviewers for helpful comments.

1 Introduction

BPL is the class of languages that can be computed by randomized logspace Turing Machines (TM) with error probability $\leq 1/3$. Here by *randomized* we mean that the TM has read-once access to a random tape. We also require that the TM halts on any input randomness. Whether $\mathsf{BPL} \overset{?}{=} \mathsf{L}$ is a big open problem of *space-bounded derandomization* in theoretical computer science. Most believe that $\mathsf{L} = \mathsf{BPL}$ is true. Different from *time-bounded derandomization*, we even do not know whether $\mathsf{L} = \mathsf{NL}$ implies $\mathsf{L} = \mathsf{BPL}$. But on the other hand, there is no known barrier for proving $\mathsf{L} = \mathsf{BPL}$. The seminal work by Saks and Zhou [29] shows that $\mathsf{BPL} \subseteq \mathsf{L}^{3/2}$ and this was improved to be $\mathsf{BPL} \subseteq \mathsf{SPACE}\left[O\left((\log n)^{3/2}/\sqrt{\log\log n}\right)\right]$ by Hoza [17].

The relation between (Randomized) small space-bounded computation and uniform low-depth circuits is also an interesting topic. It is well known that $\mathsf{L\text{-}NC}^1 \subseteq \mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{L\text{-}AC}^1$, where $\mathsf{L\text{-}NC}^1$ and $\mathsf{L\text{-}AC}^1$ are complexity classes of logspace-uniform $O(\log n)$-depth NC and AC circuits. But for BPL, there is still an interesting question: is BPL also a subset of $\mathsf{L\text{-}AC}^1$? If the conjecture $\mathsf{L} = \mathsf{BPL}$ is true, or even if $\mathsf{BPL} \subseteq \mathsf{NL}$, then immediately $\mathsf{BPL} \subseteq \mathsf{L\text{-}AC}^1$. But without these assumptions, it becomes a challenge. In this work, we will unconditionally prove that $\mathsf{BPL} \subseteq \mathsf{L\text{-}AC}^1$. On the other hand, we mention that the inclusion $\mathsf{BPL} \subseteq \mathsf{AC}^1$ for non-uniform AC^1, is obvious via non-uniform derandomization techniques.[1] See Figure 1 for a visualization of the known relations between the complexity classes.

[1] There are two ways to prove $\mathsf{BPL} \subseteq$ non-uniform AC^1: (1) By $\mathsf{L} \subseteq \mathsf{AC}^1$ we know BPL can be computed by randomized AC^1 circuits, then apply the non-uniform derandomization for AC in [3] we know $\mathsf{BPL} \subseteq \mathsf{AC}^1$; (2) First by a standard non-uniform derandomization argument we have $\mathsf{BPL} \subseteq \mathsf{L}_{/\mathsf{poly}}$, then by $\mathsf{L} \subseteq \mathsf{AC}^1$ we know $\mathsf{BPL} \subseteq \mathsf{L}_{/\mathsf{poly}} \subseteq \mathsf{AC}^1$.

39th Computational Complexity Conference (CCC 2024).
Editor: Rahul Santhanam; Article No. 32; pp. 32:1–32:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
COMPUTATIONAL COMPLEXITY CONFERENCE

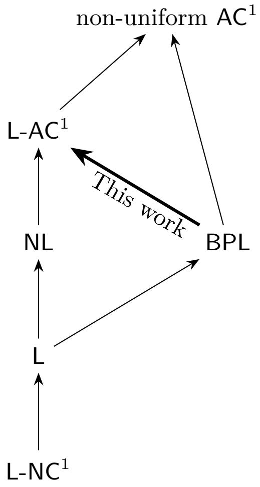

Figure 1 Relation of Complexity Classes. A → B means A $\subseteq$ B.

One may view derandomizing BPL as the problem of approximating powers of substochastic matrices. For a TM with s bits of memory, one can label all its states by elements in $[2^s]$. We can define $\mathbf{A} \in \mathbb{R}^{2^s \times 2^s}$ to be its transition matrix in the sense that $\mathbf{A}_{i,j}$ is the probability that the machine moves from state i to state j in one step. Note that it must arrive at an *accept* or *reject* state in 2^s steps, so we only need to approximate $\mathbf{A}^{2^s}$. Saks and Zhou [29] showed that approximating $\mathbf{A}^n$ for $\mathbf{A} \in \mathbb{R}^{w\times w}$ can be done in space $O\left((\log n)^{3/2} + \sqrt{\log n} \cdot \log w\right)$. Hoza [17] gave a logarithmic improvement in the $n = \mathit{poly}(w)$ regime, attaining $O(\frac{1}{\sqrt{\log\log n}}(\log n)^{3/2})$ space. Cohen, Doron, Sberlo and Ta-shma[12], and also Putterman and Pyne [25] independently improved [29]'s result to $\widetilde{O}(\log n+\sqrt{\log n}\cdot\log w)$. We mention that a closely related problem setting is to approximate the multiplication of many distinct matrices, also called the iterated matrix multiplication (IMM) setting. This corresponds to the *read-once branching program* (ROBP) model. IMM asks one to approximate $\mathbf{A}_1\mathbf{A}_2\cdots\mathbf{A}_n$ for a given sequence $\mathbf{A}_1,\cdots,\mathbf{A}_n \in \mathbb{R}^{w\times w}$. [29, 17, 12, 25] can also work for IMM, attaining the same parameters respectively as their results for matrix powering. In [29] and [17], this is done via a simple block-box reduction from the powering setting. While in [12] and [25], a more careful analysis is applied. In the rest of our paper, we mainly consider matrix powering since this is enough for our main result and IMM is also in BPL. A key idea in [17, 12, 25] is to use Richardson Iteration to boost precision, which is developed in a line of work [1, 28, 10, 12, 25, 8]. We briefly recall this method here. Consider the problem of approximating $\mathbf{X}^{-1}$ for an invertible matrix $\mathbf{X}$. Assume we already have a matrix $\mathbf{Y}$, which is an approximation of $\mathbf{X}^{-1}$ such that $\|\mathbf{I}-\mathbf{Y}\mathbf{X}\| < \varepsilon$. Then we can rewrite $\mathbf{X}\mathbf{X}^{-1} = \mathbf{I}$ as

$$\mathbf{X}^{-1} = (\mathbf{I}-\mathbf{Y}\mathbf{X})\mathbf{X}^{-1} + \mathbf{Y}.$$

Start from $\mathbf{Y}^{(0)} = \mathbf{Y}$, by taking the iteration

$$\mathbf{Y}^{(i+1)} := (\mathbf{I}-\mathbf{Y}\mathbf{X})\mathbf{Y}^{(i)} + \mathbf{Y},$$

we can reduce $\left\|\mathbf{Y}^{(i)} - \mathbf{X}^{-1}\right\|$ very quickly. Then in the application of approximating $\mathbf{A}^1,\cdots,\mathbf{A}^n$, we can take

$$\mathbf{X} := \begin{pmatrix} \mathbf{I} & & & & & \\ -\mathbf{A} & \mathbf{I} & & & & \\ & -\mathbf{A} & \mathbf{I} & & & \\ & & & \ddots & & \\ & & & & -\mathbf{A} & \mathbf{I} \end{pmatrix}, \quad \mathbf{X}^{-1} = \begin{pmatrix} \mathbf{I} & & & & & \\ \mathbf{A} & \mathbf{I} & & & & \\ \mathbf{A}^2 & \mathbf{A} & \mathbf{I} & & & \\ & & & & & \\ \mathbf{A}^{n-1} & & & \cdots & \mathbf{I} & \\ \mathbf{A}^n & \mathbf{A}^{n-1} & & & \mathbf{A} & \mathbf{I} \end{pmatrix}.$$

In this way, approximating $\mathbf{A}, \mathbf{A}^2, \cdots, \mathbf{A}^n$ do not necessarily need to conduct approximating for their inverse matrices. Although Richardson Iteration is a powerful method, an interesting question is: are there any other iteration methods that have different or more powerful effects? In fact we develop a more efficient iteration algorithm for boosting precision for our setting in Section 4, which is the main ingredient of our proof of $\mathsf{BPL} \subseteq \mathsf{L\text{-}AC}^1$. The new iteration keeps using the idea of boosting precision via numerical analysis techniques, and it does not rely on approximating matrix inversions even in its analysis.

Another side of proving $\mathsf{BPL} \subseteq \mathsf{L\text{-}AC}^1$ is on the power of $\mathsf{L\text{-}AC}$ circuits. A key tool here is the approximate counting computable by $\mathsf{L\text{-}AC}$ circuits. Specifically, the task is to decide whether an n-bit string contains $\leq a$ or $\geq b$ 1's by $\mathrm{poly}(n)$-size low depth $\mathsf{L\text{-}AC}$ circuits. The $\mathsf{L\text{-}AC}^0$ algorithm for distinguishing $\geq 2n/3$ 1's and $\leq n/3$ 1's was developed by Ajtai [2]. A line of work [3, 30, 31, 13] further studies this question, achieving depth $O\left(\frac{\log \frac{n}{b-a}}{\log\log n} + 1\right)$ (also see our Lemma 17). We will show that this can actually be done by $O\left(\frac{\log \frac{b}{b-a}}{\log\log n} + 1\right)$-depth poly-size $\mathsf{L\text{-}AC}$ circuits. This can be viewed as improving the previous results from an *additive error* setting to a *multiplicative error* setting. But even so, one can see still that $\mathsf{L\text{-}AC}$ circuits are good at aggregating on *many* inputs, but not good at *high precision*. This triggers one to think of some steps of boosting precision potentially by iteration methods.

1.1 Our Result

▶ **Theorem 1** (Main Theorem, see also Corollary 21). $\mathsf{BPL} \subseteq \mathsf{L\text{-}AC}^1$.

▶ **Theorem 2** (Multiplicative Approximate Counting in AC, see also Theorem 14). *Let $n, a, b \in \mathbb{N}$ such that $0 \leq a < b \leq n$. Then there exists a* $\mathrm{poly}(n)$*-size* $O\left(\frac{\log \frac{b}{b-a}}{\log\log n} + 1\right)$*-depth* L*-uniform* AC *circuit family* $\{\mathcal{C}_{n,a,b}\}$ *that computes* $\mathsf{GapMaj}[a, b]$ *on n bits.*[2]

1.2 Related Work

We investigate some more about related work on derandomizing BPL. For other results not covered, we refer to these surveys [18][16].

A remarkable line of work [5, 22, 20, 24, 4, 6, 19, 7, 11, 27, 17] develops PRGs, weighted PRGs, and Hitting-set generators for ROBPs, which directly provide black-box derandomizations for BPL and its related classes. Among them, the celebrated work [22] by Nisan presents a logspace computable pseudorandom generator with seed length $O(\log n \log \frac{nw}{\varepsilon})$, error ε, for length n width w ROBPs. Based on some special properties of this generator, Saks and Zhou [29] showed $\mathsf{BPL} \subseteq \mathsf{L}^{3/2}$, and Nisan [23] showed that $\mathsf{BPL} \subseteq \mathsf{TISP}[\mathrm{poly}(n), O((\log n)^2)]$. Nisan and Zuckerman [24] showed a PRG for ROBPs with large widths but very short lengths

[2] $\mathsf{GapMaj}[a, b]$ is the promise problem that asks us to distinguish whether the number of 1's in an n-bit string is $\geq b$ or $\leq a$. See Definition 11 for a formal definition.

i.e. $n = \text{poly}(\log w)$, attaining seed length $O(\log w)$ with error $2^{-\log^{0.99} w}$. Armoni [4] gave an improved construction by interpolating [22] and [24]. The recent improvement for PRGs [6, 19, 7, 11, 27, 17] focus on achieving seed length optimal in the error parameter. Several iteration-type methods are applied by these work, including the use of Richardson Iteration developed by Ahmadinejad, Kelner, Murtagh, Peebles, Sidford, and Vadhan [1], and then further developed in [28, 10, 17, 12, 25, 8].

There is also a sequence of work studying derandomizing BPL under assumptions. Klivans and van Melkebeek [21] showed that under the assumption that SPACE$[O(n)]$ requires $2^{\Omega(n)}$ circuit size, one can have L $=$ BPL. Cheng and Hoza [9] showed that under the assumption that there exists a black-box hitting-set generator computable in logspace, one can have L $=$ BPL. Some recent works [15, 26, 14] further study upon this line with various and enhanced requirements for derandomization.

1.3 Proof Overview

We sketch the proof of BPL $\subseteq$ L-AC^1 and discuss the organization of our paper.

In Section 2, we describe some basic concepts and tools for our main proof.

In Section 3, we prove that deciding whether n bits contains $\leq a$ or $\geq b$ 1's can be done in poly(n)-size $O\left(\frac{\log \frac{b}{b-a}}{\log \log n} + 1\right)$-depth, see Theorem 14. This will be a building block for approximating matrix operations. The main idea to prove Theorem 14 is to reduce the general GapMaj$[a, b]$ to the special case GapMaj$[n/3, 2n/3]$ via pairwise independent hash functions, and then apply [2]'s algorithm for GapMaj$[n/3, 2n/3]$.

One can see that low-depth L-AC circuits are good at aggregating on *many* inputs, but without a *high precision*. This motivates us to consider a step of boosting precision for matrix powerings. However, we must be very careful since we cannot pay too much in depth.

In Section 4, we give the core iteration step. This is a depth-efficient iteration algorithm for boosting precision in matrix powerings, which is the main ingredient of our proof.

▶ **Theorem 3** (see also Theorem 18). *Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be a substochastic matrix and $k, t \in \mathbb{Z}^+$ such that $\log n \geq k \geq t$. Suppose substochastic matrices $\mathbf{B}_0, \cdots, \mathbf{B}_{k-1}$ are approximations of $\mathbf{A}^{2^0}, \cdots, \mathbf{A}^{2^{k-1}}$ such that $\left\|\mathbf{B}_i - \mathbf{A}^{2^i}\right\|_1 \leq \varepsilon_i$ for $i = 1, 2, \cdots, k-1$. Define* [3]

$$\mathbf{C} := -\sum_{i=1}^{t-1} \sum_{\substack{\{j_1<\cdots<j_p\}\uplus\{j'_1<\cdots<j'_q\} \\ =\{k-1,k-2,\cdots,k-i+1\}}} \mathbf{B}_{j_p}\cdots\mathbf{B}_{j_1}\mathbf{B}_{k-i}^2\mathbf{B}_{j'_1}\cdots\mathbf{B}_{j'_q}$$

$$+ \sum_{\substack{\{j_1<\cdots<j_p\}\uplus\{j'_1<\cdots<j'_q\} \\ =\{k-1,k-2,\cdots,k-t+1\}}} \mathbf{B}_{j_p}\cdots\mathbf{B}_{j_1}\mathbf{B}_{k-t}^2\mathbf{B}_{j'_1}\cdots\mathbf{B}_{j'_q}.$$

Then

$$\left\|\mathbf{C} - \mathbf{A}^{2^k}\right\|_1 \leq \sum_{i=1}^{t-1} 2^{i-1}\varepsilon_{k-i}^2 + 2^t\varepsilon_{k-t}.$$

[3] Here $\sum_{\substack{\{j_1<\cdots<j_p\}\uplus\{j'_1<\cdots<j'_q\} \\ =\{k-1,k-2,\cdots,k-i+1\}}}$ means taking the sum over all possible two-partitions of the set $\{k-1, k-2, \cdots, k-i+1\}$. Each two-partition partitions $\{k-1, k-2, \cdots, k-i+1\}$ into two disjoint subsets $\{j_1, \cdots, j_p\}, \{j'_1, \cdots, j'_q\}$. Here set elements are sorted in increasing order, i.e., $j_1 < \cdots < j_p$ and $j'_1 < \cdots < j'_q$. Therefore this $\sum$ is sum of 2^{i-1} terms.
When $i = 1$, $\{k-1, k-2, \cdots, k-i+1\}$ represents the empty set.

Intuitively speaking, we can obtain a good approximation of $\mathbf{A}^{2^k}$ given these $\mathbf{B}_{k-1}, \cdots, \mathbf{B}_0$, which either has lower accuracy or is an approximation of $\mathbf{A}^{2^{k'}}$ for much smaller k'. We will prove that the iteration step can be easily computed by low-depth L-AC circuits in Theorem 19 which crucially uses our depth-efficient approximate counting in Section 3.

In Section 5 we present the complete algorithm. We compute intermediate matrices $\mathbf{M}(k, t)$ for $k, t \le O(\log n)$, where $\mathbf{M}(k, t)$ is a $1/2^t$-approximation of $\mathbf{A}^{2^k}$ (i.e., $\left\|\mathbf{M}(k,t) - \mathbf{A}^{2^k}\right\|_1 \le 1/2^t$). We will use the iteration step developed in Section 4 to show that, for any $k, t \le O(\log n)$, given all $\mathbf{M}(k-i, \lceil t/2 \rceil + 2i)$'s (for $i = 1, 2, \cdots$), we can compute a valid $\mathbf{M}(k, t)$ in $O(t)$-depth. Then we can compute a valid $\mathbf{M}(\log n, \log n)$ in $O(\log n)$-depth.

Finally in Section 6 we will discuss some open problems.

2 Preliminaries

2.1 Matrix Approximation

▶ **Definition 4** (L_1-norm). *Define the L_1-norm of a vector $(x_1, \cdots, x_n)^\top \in \mathbb{R}^n$ to be*

$$\left\|(x_1, \cdots, x_n)^\top\right\|_1 := |x_1| + \cdots + |x_n|.$$

Define the L_1-norm of a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ to be

$$\|\mathbf{A}\|_1 := \sup_{\mathbf{x} \in \mathbb{R}^n} \frac{\|\mathbf{A}\mathbf{x}\|_1}{\|\mathbf{x}\|_1} = \max_{1 \le j \le n} \{|\mathbf{A}_{1,j}| + |\mathbf{A}_{2,j}| + \cdots + |\mathbf{A}_{n,j}|\}.$$

▶ **Theorem 5.** *For any $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times n}$, we have:*

1. $\|\mathbf{A} + \mathbf{B}\|_1 \le \|\mathbf{A}\|_1 + \|\mathbf{B}\|_1$*;*
2. $\|\mathbf{A}\mathbf{B}\|_1 \le \|\mathbf{A}\|_1 \|\mathbf{B}\|_1$*;*
3. *If $\|\mathbf{A}\|_1, \|\mathbf{B}\|_1 \le 1$, then for any $p \in \mathbb{Z}^+$, $\|\mathbf{A}^p - \mathbf{B}^p\|_1 \le p \|\mathbf{A} - \mathbf{B}\|_1$.*

▶ **Definition 6** (Non-negative Matrix). *We say a matrix is non-negative if each of its entry is non-negative.*

▶ **Definition 7** (Substochastic Matrix). *We say a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a substochastic matrix if $\mathbf{A}$ is non-negative and $\|\mathbf{A}\|_1 \le 1$.*

For simplicity, we always assume that the size of a substochastic matrix is a power of 2. To represent a substochastic matrix, we independently represent each entry in binary, accurate to $100 \log n$ decimal places.

2.2 L-uniform AC Circuit Family and Approximate Counting

▶ **Definition 8** (AC circuit). AC *circuit is a circuit with input gates, NOT gates, unbounded fan-in AND/OR gates, and (possibly more than one) output gates. The size of a circuit is defined by the number of AND/OR gates. The depth of a circuit is defined by the largest number of AND/OR gates on any path from an input gate to an output gate.*

▶ **Definition 9** (L-uniform AC circuit family). *For functions $S, d\colon \mathbb{Z}^+ \to \mathbb{R}^+$, we say a collection of circuits $\{\mathcal{C}_n\}_{n \in \mathbb{Z}^+}$ is an S-size d-depth* L*-uniform* AC *circuit family, if each $\mathcal{C}_n$ has size $\le S(n)$ and depth $\le d(n)$, and given the binary representation of n, the description of $\mathcal{C}_n$ can be computed in uniform $O(\log n)$-space.*

We need to mention that the number of input gates in $\mathcal{C}_n$ is not necessarily n. Also note that since we can encode a tuple of $O(1)$ many integers to a single integer, we can also consider circuit collections with a tuple of integers as an index.

▶ **Definition 10** (Complexity Class L-AC^1). *We say a language L is in the class* L-AC^1 *if there exists a* $\text{poly}(n)$*-size* $O(\log n)$*-depth* L*-uniform* AC *circuit family* $\{\mathcal{C}_n\}$ *such that* $\mathcal{C}_n$ *computes* L *on* n*-bit inputs.*

▶ **Definition 11** (GapMaj). *For $n \in \mathbb{Z}^+$ and $a, b \in \mathbb{R}$ such that $0 \leq a < b \leq n$, define the promise problem* GapMaj$[a,b]$ *on n bits as follow:*

$$\mathsf{GapMaj}[a,b](x_1,\cdots,x_n) := \begin{cases} \text{YES} & \textit{if } x_1,\cdots,x_n \textit{ contains} \geq b \text{ 1's} \\ \text{NO} & \textit{if } x_1,\cdots,x_n \textit{ contains} \leq a \text{ 1's} \\ \perp & \textit{otherwise} \end{cases}$$

2.3 Tool: Pairwise Independent Hash Function

We will use pairwise independent hash function as a tool for approximate counting in AC. We shall use the following construction based on convolution, which was also used in [22].

▶ **Definition 12** (Convolution-Based Pairwise Independent Hash Function). *Suppose m is a power of 2. Define $H_m\colon [m^3] \times [m] \to [m]$ by: for $(k,x) \in [m^3] \times [m]$, let $x_1 \cdots x_{\log m}$ be the binary representation of $x-1$, let $a_1 \cdots a_{2\log m} b_1 \cdots b_{\log m}$ be the binary representation of $k-1$, let $y_j := \left(\sum_{i=1}^{\log m} a_{i+j}x_i + b_j\right) \bmod 2$ for $j \in [\log m]$, then define $H_m(k,x)$ by letting $y_1 \cdots y_{\log m}$ be the binary representation of $H_m(k,x)-1$.*

▶ **Theorem 13.** *H_m is Pairwise Independent Hash Function in the following sense: for any $1 \leq i < j \leq m$, when k is sampled from the uniform distribution over $[m^3]$, the joint distribution of $(H_m(k,i), H_m(k,j))$ is identical to the uniform distribution over $[m] \times [m]$.*

3 Approximate Counting in AC

The goal of this Section is to prove Theorem 14, which will be a building block for the proof of BPL $\subseteq$ L-AC^1.

▶ **Theorem 14.** *Let $n, a, b \in \mathbb{N}$ such that $0 \leq a < b \leq n$. Then there exists a* $\text{poly}(n)$*-size* $O\left(\frac{\log \frac{b}{b-a}}{\log\log n} + 1\right)$*-depth* L*-uniform* AC *circuit family* $\{\mathcal{C}_{n,a,b}\}$ *that computes* GapMaj$[a,b]$ *on n bits.*

The proof depends on the next few Lemmas.

▶ **Lemma 15** ([2]). *Let $n \in \mathbb{Z}^+$. Then there exists* $\text{poly}(n)$*-size* $O(1)$*-depth* L*-uniform* AC *circuit family* $\{\mathcal{C}_n^{(0)}\}$ *that computes* GapMaj$[n/3, 2n/3]$ *on n bits.*

▶ **Lemma 16** (Exact Counting). *Let $n, \ell \in \mathbb{Z}^+$ such that $n \geq \ell$. Then there exists a* $\text{poly}(n)$*-size* $O\left(\frac{\log \ell}{\log\log n} + 1\right)$*-depth* L*-uniform* AC *circuit family* $\{\mathcal{E}_{n,\ell}\}$ *such that on ℓ bits of input, $\mathcal{E}_{n,\ell}$ outputs the exact number of 1's over the input bits, in binary form.*

We remark that in Lemma 16, n is only used to bound the size of the circuit. Also, n, ℓ are not necessarily polynomially related.

Proof. We only need to show how to compute sum of $O(\sqrt{\log n})$ many $O(\log n)$-bit[4] non-negative integers in $O(1)$-depth, then by divide-and-conquer we can compute sum of ℓ bits in $O\left(\frac{\log \ell}{\log\log n}+1\right)$-depth.

View the $O(\log n)$-bit integers as $2^{\left\lceil\sqrt{\log n}\right\rceil}$-base $O(\sqrt{\log n})$-digit integers. We use the grade-school algorithm to sum $O(\sqrt{\log n})$ integers as follow. We first guess the result and all *carry-bits*, which involve at most $O(\sqrt{\log n})\cdot O\left(\log\left(\sqrt{\log n}\cdot 2^{\left\lceil\sqrt{\log n}\right\rceil}\right)\right)=O(\log n)$ bits, and thus has at most $\mathrm{poly}(n)$ choices. Then we can apply a local check on each digit, each local check involves at most $O(\log n)$ bits, and thus deciding whether all local checks are passed can be computed in $O(1)$-depth. Then we can take the result of the only guess that passes all local checks. The total cost is $O(1)$-depth. ◀

▶ **Lemma 17.** *Let $n,a,b\in\mathbb{N}$ such that $0\le a<b\le n$. Then there exists a* $\mathrm{poly}(n)$*-size* $O\left(\frac{\log\frac{n}{b-a}}{\log\log n}+1\right)$*-depth* L*-uniform* AC *circuit family* $\{\mathcal{C}^{(1)}_{n,a,b}\}$ *that computes* $\mathsf{GapMaj}[a,b]$ *on n bits.*

Proof. Only consider the case that n is a power of 2, otherwise we can use a simple padding argument. By Lemma 15, it suffices to show how to reduce $\mathsf{GapMaj}[a,b]$ on n bits to $\mathsf{GapMaj}[n^3/3,2n^3/3]$ on n^3 bits, via a $\mathrm{poly}(n)$-size $O\left(\frac{\log\frac{n}{b-a}}{\log\log n}+1\right)$-depth L-uniform AC circuit.

If $b-a\le 4\sqrt{n}$ then we can directly compute the number of 1's exactly via Lemma 16. Below we only consider $b-a>4\sqrt{n}$.

Let $\ell:=\left\lceil\frac{12n^2}{(b-a)^2}\right\rceil$. Suppose the $\mathsf{GapMaj}[a,b]$ instance is $x_1,x_2,\cdots,x_n$. Let H_n be the hash function defined in Definition 12. Define $y_1,\cdots,y_{n^3}$ as follow: for $i\in[n^3]$, let y_i be 1 if at least $\frac{a+b}{2n}$ fraction of $x_{H_n(i,1)},\cdots,x_{H_n(i,\ell)}$ is 1, otherwise let y_i be 0. Note that $y_1,\cdots,y_{n^3}$ can be computed via a $\mathrm{poly}(n)$-size $O\left(\frac{\log\ell}{\log\log n}+1\right)$-depth L-uniform AC circuit, by Lemma 16. Here $O\left(\frac{\log\ell}{\log\log n}+1\right)=O\left(\frac{\log\frac{n}{b-a}}{\log\log n}+1\right)$.

Let's do some simple calculations. Assume p fraction of $x_1,\cdots,x_n$ is 1. Let S_i be number of 1's in $x_{H_n(i,1)},\cdots,x_{H_n(i,\ell)}$. Then we have $\mathbb{E}_{i\sim[n^3]}[S_i]=p\ell$ and $\mathrm{Var}_{i\sim[n^3]}[S_i]\le\ell$. So if $p\le\frac{a}{n}$, then $\Pr_{i\sim[n^3]}\left[S_i\ge\ell\cdot\frac{(a+b)}{2n}\right]\le\frac{\ell}{\left(\ell\cdot\frac{(b-a)}{2n}\right)^2}=\frac{4n^2}{\ell(b-a)^2}\le\frac{1}{3}$. Similarly if $p\ge\frac{b}{n}$ then $\Pr_{i\sim[n^3]}\left[S_i\le\ell\cdot\frac{(a+b)}{2n}\right]\le\frac{1}{3}$. This means if $x_1,\cdots,x_n$ is YES/NO instance of $\mathsf{GapMaj}[a,b]$, then $y_1,\cdots,y_{n^3}$ is YES/NO instance of $\mathsf{GapMaj}[n^3/3,2n^3/3]$. The reduction is completed. ◀

Proof of Theorem 14. We will try to reduce to Lemma 17. Suppose the $\mathsf{GapMaj}[a,b]$ instance is $x_1,x_2,\cdots,x_n$. We only consider the case n is a power of 2, otherwise use a simple padding argument. We only consider the case $10\left(\frac{b}{b-a}\right)^2<\frac{n}{b-a}$ (or equivalently, $n(b-a)>10b^2$), otherwise we can directly apply Lemma 17.

Let $\ell:=\left\lceil\frac{n(b-a)}{2b^2}\right\rceil$. For $i\in[n^3]$, let $y_i:=x_{H_n(i,1)}\vee\cdots\vee x_{H_n(i,\ell)}$, here H_n is the hash function defined in Definition 12. Then $y_1,\cdots,y_{n^3}$ can be computed via $\mathrm{poly}(n)$-size $O(1)$-depth L-uniform AC circuit.

Assume p fraction of $x_1,\cdots,x_n$ is 1. Let S_i be number of 1's in $x_{H_n(i,1)},\cdots,x_{H_n(i,\ell)}$. Then we have $\mathbb{E}_{i\sim[n^3]}[S_i]=p\ell$ and $\mathbb{E}_{i\sim[n^3]}[S_i^2]=\ell(\ell-1)p^2+\ell p\le\ell p+\ell^2p^2$. Thus by

$$\frac{\mathbb{E}_{i\sim[n^3]}[S_i]^2}{\mathbb{E}_{i\sim[n^3]}[S_i^2]}\le\Pr_{i\sim[n^3]}[S_i\ge 1]\le\mathop{\mathbb{E}}_{i\sim[n^3]}[S_i]$$

[4] Here "$O(\log n)$-bit integers" refers to integers which has $O(\log n)$-bits in its binary representation.

we know: if $p \le \frac{a}{n}$, then $\Pr_{i\sim[n^3]}[S_i \ge 1] \le \frac{\ell a}{n}$; if $p \ge \frac{b}{n}$, then $\Pr_{i\sim[n^3]}[S_i \ge 1] \ge \frac{\left(\frac{\ell b}{n}\right)^2}{\frac{\ell b}{n}+\left(\frac{\ell b}{n}\right)^2} \ge \frac{\ell b}{n} - \left(\frac{\ell b}{n}\right)^2$. To summarize, if $x_1, \cdots, x_n$ is YES/NO instance of $\mathsf{GapMaj}[a,b]$, then $y_1, \cdots, y_{n^3}$ is YES/NO instance of $\mathsf{GapMaj}\left[\left\lfloor n^3 \cdot \frac{\ell a}{n}\right\rfloor, \left\lceil n^3 \cdot \left(\frac{\ell b}{n} - \left(\frac{\ell b}{n}\right)^2\right)\right\rceil\right]$.

Finally we observe that $\left(\frac{\ell b}{n} - \left(\frac{\ell b}{n}\right)^2\right) - \frac{\ell a}{n} = \ell \cdot \left(\frac{b-a}{n} - \frac{\ell b^2}{n^2}\right) \ge \frac{n(b-a)}{3b^2} \cdot \frac{b-a}{2n} = \frac{(b-a)^2}{6b^2}$. Thus by Lemma 17, $\mathsf{GapMaj}\left[\left\lfloor n^3 \cdot \frac{\ell a}{n}\right\rfloor, \left\lceil n^3 \cdot \left(\frac{\ell b}{n} - \left(\frac{\ell b}{n}\right)^2\right)\right\rceil\right]$ over n^3 bits can be computed via a poly(n)-size $O\left(\frac{\log \frac{b}{b-a}}{\log\log n} + 1\right)$-depth L-uniform AC circuit. ◀

4 The Iteration Method

In this section, we will introduce the iteration step, which is the core of our proof of BPL $\subseteq$ L-AC^1.

▶ **Theorem 18** (The Iteration). *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a substochastic matrix and $k, t \in \mathbb{Z}^+$ such that $\log n \ge k \ge t$. Suppose substochastic matrices $\mathbf{B}_0, \cdots, \mathbf{B}_{k-1}$ are approximations of $\mathbf{A}^{2^0}, \cdots, \mathbf{A}^{2^{k-1}}$ such that $\left\|\mathbf{B}_i - \mathbf{A}^{2^i}\right\|_1 \le \varepsilon_i$ for $i = 1, 2, \cdots, k-1$. Define*

$$\mathbf{C} := -\sum_{i=1}^{t-1} \sum_{\substack{\{j_1<\cdots<j_p\}\uplus\{j'_1<\cdots<j'_q\}\\ =\{k-1,k-2,\cdots,k-i+1\}}} \mathbf{B}_{j_p}\cdots\mathbf{B}_{j_1}\mathbf{B}_{k-i}^2\mathbf{B}_{j'_1}\cdots\mathbf{B}_{j'_q}$$
$$+ \sum_{\substack{\{j_1<\cdots<j_p\}\uplus\{j'_1<\cdots<j'_q\}\\ =\{k-1,k-2,\cdots,k-t+1\}}} \mathbf{B}_{j_p}\cdots\mathbf{B}_{j_1}\mathbf{B}_{k-t}^2\mathbf{B}_{j'_1}\cdots\mathbf{B}_{j'_q}.$$

Then

$$\left\|\mathbf{C} - \mathbf{A}^{2^k}\right\|_1 \le \sum_{i=1}^{t-1} 2^{i-1}\varepsilon_{k-i}^2 + 2^t\varepsilon_{k-t}.$$

Proof. Note that

$$\mathbf{C} - \mathbf{A}^{2^k} = -\sum_{i=1}^{t-1} \sum_{\substack{\{j_1<\cdots<j_p\}\uplus\{j'_1<\cdots<j'_q\}\\ =\{k-1,k-2,\cdots,k-i+1\}}} \mathbf{B}_{j_p}\cdots\mathbf{B}_{j_1}\left(\mathbf{A}^{2^{k-i}} - \mathbf{B}_{k-i}\right)^2\mathbf{B}_{j'_1}\cdots\mathbf{B}_{j'_q}$$
$$- \sum_{\substack{\{j_1<\cdots<j_p\}\uplus\{j'_1<\cdots<j'_q\}\\ =\{k-1,k-2,\cdots,k-t+1\}}} \mathbf{B}_{j_p}\cdots\mathbf{B}_{j_1}\left(\mathbf{A}^{2^{k-t+1}} - \mathbf{B}_{k-t}^2\right)\mathbf{B}_{j'_1}\cdots\mathbf{B}_{j'_q}.$$

So by Theorem 5,

$$\left\|\mathbf{C} - \mathbf{A}^{2^k}\right\|_1 \le \sum_{i=1}^{t-1} 2^{i-1}\left\|\mathbf{A}^{2^{k-i}} - \mathbf{B}_{k-i}\right\|_1^2 + 2^t\left\|\mathbf{A}^{2^{k-t}} - \mathbf{B}_{k-t}\right\|_1$$
$$\le \sum_{i=1}^{t-1} 2^{i-1}\varepsilon_{k-i}^2 + 2^t\varepsilon_{k-t}.$$ ◀

▶ **Theorem 19** (Computing the Iteration). *Let $n, k, t, \mathbf{A}, \mathbf{B}_0, \cdots, \mathbf{B}_{k-1}, \varepsilon_0, \cdots, \varepsilon_{k-1}, \mathbf{C}$ be as defined in Theorem 18. Let d be an integer such that $4\log n \geq d \geq t/10$. Then there exists a* $\mathrm{poly}(n)$*-size* $O(d)$*-depth* L*-uniform* AC *circuit family $\{\mathcal{I}_{n,k,t,d}\}$ such that on inputs $\mathbf{B}_{k-t}, \cdots, \mathbf{B}_{k-1}$, if*

$$\sum_{i=1}^{t-1} 2^{i-1}\varepsilon_{k-i}^2 + 2^t\varepsilon_{k-t} \leq \frac{1}{2^{d+2}}$$

is satisfied, then $\mathcal{I}_{n,k,t,d}$ outputs a substochastic matrix $\mathbf{C}'$ such that $\left\|\mathbf{C}' - \mathbf{A}^{2^k}\right\|_1 \leq 1/2^d$.

The intuition behind Theorem 19 is that to approximately compute $\mathbf{C}$, all arithmetic operations only need a multiplicative accuracy of $1/2^{\Theta(d)}$. This can be done efficiently by L-uniform AC circuit by Theorem 14.

Proof of Theorem 19. We observe that $\mathbf{C}$ is the sum of 2^{t-1} "+" terms and $2^{t-1} - 1$ "−" terms, and each term is a multiplication of not more than $t + 1$ substochastic matrices. We will first show how to approximate the multiplication of substochastic matrices and then show how to approximate their sum.

To approximate $\mathbf{Z} := \mathbf{X}\mathbf{Y}$ for two substochastic matrices $\mathbf{X}, \mathbf{Y}$, we only need to approximate $\sum_{r=1}^n \mathbf{X}_{i,r}\mathbf{Y}_{r,j}$ for each pair $(i, j) \in [n]^2$. We first represent each entry $\mathbf{X}_{i,r}, \mathbf{Y}_{r,j}$ using n^{100} bits such that fraction of 1's in these n^{100} bits is equal to the entry, then use a layer of AND gate to represent each $\mathbf{X}_{i,r}\mathbf{Y}_{r,j}$ using fraction of 1's in n^{200} bits, and then represent each $\frac{1}{n}\sum_{r=1}^n \mathbf{X}_{i,r}\mathbf{Y}_{r,j}$ using fraction of 1's in n^{201} bits. Then we invoke $\mathcal{C}_{n^{201},\ell,\lceil \ell(1+1/2^{20d+10})\rceil}$ (as defined in Theorem 14, which has depth $\leq O\left(\frac{d}{\log\log n} + 1\right) \leq O\left(\frac{d}{\log(t+1)}\right))$[5] for $\ell = 1, 2, \cdots, n^{200}$ over these n^{201} bits. Suppose ℓ_0 is the smallest index such that $\mathcal{C}_{n^{201},\ell_0,\lceil \ell_0(1+1/2^{20d+10})\rceil}$ outputs 0, then we have

$$\frac{\ell_0 - 1}{n^{200}} < \mathbf{Z}_{i,j} < \frac{\ell_0\left(1 + \frac{1}{2^{20d+10}}\right)}{n^{200}}$$

and thus[6]

$$\frac{\mathbf{Z}_{i,j}}{1 + \frac{1}{2^{20d+10}}} - \frac{1}{n^{100}} \leq \frac{1}{n^{100}}\left[\frac{\ell_0}{n^{100}}\right] \leq \mathbf{Z}_{i,j}.$$

Use $[\ell_0/n^{100}]/n^{100}$ as an approximation of $\mathbf{Z}_{i,j}$, then we obtain an approximation $\widetilde{\mathbf{Z}}$ of $\mathbf{Z}$ such that $\mathbf{Z} - \widetilde{\mathbf{Z}}$ is non-negative and $\widetilde{\mathbf{Z}}$ is substochastic and $\left\|\mathbf{Z} - \widetilde{\mathbf{Z}}\right\|_1 \leq 1/2^{20d+10} + 1/n^{99}$. We need to be careful that here we need a *multiplicative* small error on each entry and thus we need to strengthen Lemma 17 to Theorem 14.

Then multiplication of not more than $t + 1$ substochastic matrices can be computed via $O(\log(t + 1))$ layers of multiplication of two matrices. Recall that multiplying two matrices uses $O\left(\frac{d}{\log(t+1)}\right)$-depth and has additive error $1/2^{20d+10} + 1/n^{99}$. So the total depth for computing multiplication of not more than $t+1$ substochastic matrices is $O(d)$ and the total error is $\leq t(1/2^{20d+10} + 1/n^{99}) \leq 1/2^{19d+5}$.

[5] In Theorem 14 we take $(a, b) = (\ell, \lceil \ell(1 + 1/2^{20d+10})\rceil)$, and then $\log\frac{b}{b-a} \leq O(d)$.
[6] Since $n^{200}\mathbf{Z}_{i,j}$ is an integer, we have $\frac{\ell_0-1}{n^{200}} < \mathbf{Z}_{i,j} \implies \frac{\ell_0}{n^{200}} \leq \mathbf{Z}_{i,j}$.

To summarize, suppose $\mathbf{C} = -\sum_{i=1}^{2^{t-1}-1} \mathbf{D}_i + \sum_{i=1}^{2^{t-1}} \mathbf{D}'_i$, here each $\mathbf{D}_i, \mathbf{D}'_i$ is multiplication of some substochastic matrices. Then we can compute their approximations $\widetilde{\mathbf{D}_i}, \widetilde{\mathbf{D}'_i}$ in $O(d)$ depth such that $\left\|\mathbf{D}_i - \widetilde{\mathbf{D}_i}\right\|_1 \le 1/2^{19d+5}$ and $\left\|\mathbf{D}'_i - \widetilde{\mathbf{D}'_i}\right\|_1 \le 1/2^{19d+5}$.

We approximate $\frac{1}{2^{t-1}}\sum_{i=1}^{2^{t-1}-1} \widetilde{\mathbf{D}_i}$ and $\frac{1}{2^{t-1}}\sum_{i=1}^{2^{t-1}} \widetilde{\mathbf{D}'_i}$. Use the similar idea as summing $\frac{1}{n}\sum_{r=1}^{n} \mathbf{X}_{i,r}\mathbf{Y}_{r,j}$, we can compute substochastic matrices $\mathbf{C}^-, \mathbf{C}^+$ using $O(d)$-depth, such that

$$\left\|\mathbf{C}^- - \frac{1}{2^{t-1}}\sum_{i=1}^{2^{t-1}-1}\widetilde{\mathbf{D}_i}\right\|_1 \le \frac{1}{2^{19d+5}},$$

$$\left\|\mathbf{C}^+ - \frac{1}{2^{t-1}}\sum_{i=1}^{2^{t-1}}\widetilde{\mathbf{D}'_i}\right\|_1 \le \frac{1}{2^{19d+5}}.$$

Then $2^{t-1}(\mathbf{C}^+ - \mathbf{C}^-)$ is a good approximation of $\mathbf{A}^{2^k}$ since

$$\begin{aligned}
\left\|2^{t-1}(\mathbf{C}^+ - \mathbf{C}^-) - \mathbf{A}^{2^k}\right\|_1 &\le 2^{t-1}\left\|\mathbf{C}^- - \frac{1}{2^{t-1}}\sum_{i=1}^{2^{t-1}-1}\widetilde{\mathbf{D}_i}\right\|_1 + 2^{t-1}\left\|\mathbf{C}^+ - \frac{1}{2^{t-1}}\sum_{i=1}^{2^{t-1}}\widetilde{\mathbf{D}'_i}\right\|_1 \\
&\quad + \sum_{i=1}^{2^{t-1}-1}\left\|\mathbf{D}_i - \widetilde{\mathbf{D}_i}\right\|_1 + \sum_{i=1}^{2^{t-1}}\left\|\mathbf{D}'_i - \widetilde{\mathbf{D}'_i}\right\|_1 \\
&\quad + \left\|-\sum_{i=1}^{2^{t-1}-1}\mathbf{D}_i + \sum_{i=1}^{2^{t-1}}\mathbf{D}'_i - \mathbf{A}^{2^k}\right\|_1 \\
&\le \frac{2^{t-1}}{2^{19d+5}} + \frac{2^{t-1}}{2^{19d+5}} + \frac{2^{t-1}}{2^{19d+5}} + \frac{2^{t-1}}{2^{19d+5}} + \left\|\mathbf{C} - \mathbf{A}^{2^k}\right\|_1 \\
&\le \frac{1}{2^{9d+4}} + \left(\sum_{i=1}^{t-1} 2^{i-1}\varepsilon_{k-i}^2 + 2^t\varepsilon_{k-t}\right) \\
&\le \frac{1}{2^{9d+4}} + \frac{1}{2^{d+2}}.
\end{aligned}$$

Here the last step is from the statement of Theorem 19.

Finally we compute a substochastic matrix $\mathbf{C}'$ which is a good approximation of $\mathbf{A}^{2^k}$ and $2^{t-1}(\mathbf{C}^+ - \mathbf{C}^-)$. Here we need to be careful that $\mathbf{C}$ and $2^{t-1}(\mathbf{C}^+ - \mathbf{C}^-)$ are not necessarily non-negative or substochastic (but $\mathbf{A}^{2k}$ is guaranteed substochastic). Let

$$\begin{aligned}
\mathbf{C}''_{i,j} &:= \max\{2^{t-1}(\mathbf{C}^+_{i,j} - \mathbf{C}^-_{i,j}), 0\}, \\
\mathbf{C}'_{i,j} &:= \frac{1}{n^{100}}\left\lfloor \mathbf{C}''_{i,j}\left(1 - \frac{1}{2^{d+1}}\right)\cdot n^{100}\right\rfloor.
\end{aligned}$$

We can compute $\mathbf{C}'$ given $\mathbf{C}^+, \mathbf{C}^-$ by hardwiring the map $(\mathbf{C}^+_{i,j}, \mathbf{C}^-_{i,j}) \mapsto \mathbf{C}'_{i,j}$, which is L-uniform. Obviously $\mathbf{C}'$ is non-negative. Note that $\mathbf{C}''$ is entrywise closer to $\mathbf{A}^{2^k}$ than $2^{t-1}(\mathbf{C}^+ - \mathbf{C}^-)$ and hence

$$\left\|\mathbf{C}'' - \mathbf{A}^{2^k}\right\|_1 \le \left\|2^{t-1}(\mathbf{C}^+ - \mathbf{C}^-) - \mathbf{A}^{2^k}\right\|_1 \le \frac{1}{2^{9d+4}} + \frac{1}{2^{d+2}}$$

Therefore $\mathbf{C}'$ is substochastic since

$$\|\mathbf{C}'\|_1 \le \left(1 - \frac{1}{2^{d+1}}\right)\|\mathbf{C}''\|_1 \le \left(1 - \frac{1}{2^{d+1}}\right)\left(1 + \frac{1}{2^{9d+4}} + \frac{1}{2^{d+2}}\right) \le 1.$$

Also note that

$$\begin{aligned}\left\|\mathbf{C}'-\mathbf{A}^{2^k}\right\|_1 &\le \|\mathbf{C}'-\mathbf{C}''\|_1+\left\|\mathbf{C}''-\mathbf{A}^{2^k}\right\|_1\\ &\le \frac{1}{n^{99}}+\frac{1}{2^{d+1}}\|\mathbf{C}''\|_1+\left\|\mathbf{C}''-\mathbf{A}^{2^k}\right\|_1\\ &\le \frac{1}{n^{99}}+\frac{1}{2^{d+1}}\left(1+\frac{1}{2^{9d+4}}+\frac{1}{2^{d+2}}\right)+\frac{1}{2^{9d+4}}+\frac{1}{2^{d+2}}\\ &\le \frac{1}{2^d}.\end{aligned}$$

To summarize, we can output a valid $\mathbf{C}'$ in $O(d)$-depth. And the circuit is poly(n)-size and L-uniform. ◀

5 The Complete Algorithm

▶ **Theorem 20.** *Let n be a power of 2. Then there exists a* poly(n)*-size* $O(\log n)$*-depth* L*-uniform* AC *circuit family* $\{\mathcal{M}_n\}$[7] *such that on input a substochastic matrix* $\mathbf{A}\in\mathbb{R}^{n\times n}$, $\mathcal{M}_n$ *outputs a substochastic matrix* $\mathbf{M}\in\mathbb{R}^{n\times n}$ *such that* $\|\mathbf{M}-\mathbf{A}^n\|_1\le 1/n$.

Proof. Only consider $\log n\ge 10$. For $k,t\in\mathbb{N}$ such that $k\le\log n$ and $1\le t\le 3\log n-2k$, we wish to compute a substochastic matrix $\mathbf{M}(k,t)$, which is an approximation of $\mathbf{A}^{2^k}$, such that $\left\|\mathbf{M}(k,t)-\mathbf{A}^{2^k}\right\|_1\le 1/2^t$. Then $\mathbf{M}:=\mathbf{M}(\log n,\log n)$ is the desired matrix.

For $k=0$, we can trivially let $\mathbf{M}(0,t):=\mathbf{A}$. Now we show how to recursively compute $\mathbf{M}(k_0,t_0)$ for $k_0=1,2,\cdots,\log n$.

In Theorem 18, take the same $n,\mathbf{A}$ and take $k:=k_0$, take $\mathbf{B}_{k-i}:=\mathbf{M}(k-i,\lceil t_0/2\rceil+2i)$ for $1\le i\le k$. Then we can take $\varepsilon_{k-i}:=1/2^{\lceil t_0/2\rceil+2i}$ for $1\le i\le k-1$ and $\varepsilon_0=0$. Now we will invoke Theorem 18, 19 by choosing parameter t properly according to the following two cases.

Case 1. $k\le 2t_0+2$.
Take the parameter t in Theorem 18 to be $t:=k$. Then

$$\sum_{i=1}^{k-1}2^{i-1}\varepsilon_{k-i}^2+2^k\varepsilon_0=\sum_{i=1}^{k-1}\frac{1}{2^{2\lceil t_0/2\rceil+3i+1}}\le\frac{1}{2^{t_0+2}}.$$

In Theorem 19 take $d:=t_0$. It is easy to verify that $\log n\ge k\ge t$ and $4\log n\ge d\ge t/10$ hold when we invoke Theorem 18, 19. Given $\mathbf{B}_{k-1},\cdots,\mathbf{B}_0$, use $\mathcal{I}_{n,k_0,k_0,t_0}$ (defined in Theorem 19) we can compute a substochastic matrix $\mathbf{C}'$ such that $\left\|\mathbf{C}'-\mathbf{A}^{2^k}\right\|_1\le 1/2^{t_0}$.

Case 2. $k\ge 2t_0+3$.
Take $t:=2t_0+3$ in Theorem 18. Then

$$\sum_{i=1}^{2t_0+2}2^{i-1}\varepsilon_{k-i}^2+2^{2t_0+3}\varepsilon_{k-2t_0-3}\le\sum_{i=1}^{2t_0+2}\frac{1}{2^{2\lceil t_0/2\rceil+3i+1}}+\frac{1}{2^{\lceil t_0/2\rceil+2t_0+3}}\le\frac{1}{2^{t_0+2}}.$$

In Theorem 19 take $d:=t_0$. Given $\mathbf{B}_{k-1},\cdots,\mathbf{B}_0$, use $\mathcal{I}_{n,k_0,2t_0+3,t_0}$ we can compute a substochastic matrix $\mathbf{C}'$ such that $\left\|\mathbf{C}'-\mathbf{A}^{2^k}\right\|_1\le 1/2^{t_0}$.

[7] We require that given n, description of $\mathcal{M}_n$ can be computed in space $O(\log n)$.

To summarize, take $\mathbf{M}(k_0, t_0) := \mathbf{C}'$, we can compute $\mathbf{M}(k_0, t_0)$ given $\mathbf{M}(k_0 - i, [t_0/2] + 2i)$ for $1 \le i \le k_0$, via a poly(n)-size $O(t_0)$-depth L-uniform AC circuit.

Let $\gamma > 0$ be a concrete constant such that we can compute $\mathbf{M}(k_0, t_0)$ given $\mathbf{M}(k_0 - i, [t_0/2] + 2i)$'s via a poly$(n)$-size γt_0-depth L-uniform AC circuit. Note that if $\mathbf{M}(k_0 - i, [t_0/2] + 2i)$ can be computed in $2\gamma(2(k_0 - i) + ([t_0/2] + 2i))$-depth for $1 \le i \le k_0$, then $\mathbf{M}(k_0, t_0)$ can be computed in

$$\gamma t_0 + \max_{1 \le i \le k_0} \{2\gamma(2(k_0 - i) + ([t_0/2] + 2i))\} \le 2\gamma(2k_0 + t_0)$$

-depth. Also note that $\mathbf{M}(0, t_0)$'s are just the inputs, so by induction we know $\mathbf{M}(k_0, t_0)$ can be computed in $2\gamma(2k_0 + t_0)$-depth. Specially, $\mathbf{M}(\log n, \log n)$ (which is the desired output) can be computed in $6\gamma \log n \le O(\log n)$-depth. Also note that we use "compute $\mathbf{M}(k_0, t_0)$ given $\mathbf{M}(k_0 - i, [t_0/2] + 2i)$" $O((\log n)^2)$ many times, so the total circuit size for computing $\mathbf{M}(\log n, \log n)$ is still poly(n). ◀

▶ **Corollary 21.** BPL $\subseteq$ L-AC^1.

6 Open Problems

1. Our algorithm based on the improved iteration can be thought of as low-depth of *precision requirement*. Can this method be applied to obtain other interesting results in derandomizing BPL? It seems that the space-bounded model or nondeterministic space-bounded model cannot deal with low accuracy aggregating on many bits at low cost, as in the AC circuit model.
2. Our algorithm involves a "$\times O(\log \log n)$" step when multiplying $O(\log n)$ matrices and a "$/O(\log \log n)$" step in approximate counting in AC, which seems *coincidentally* achieves $O(\log n)$-depth. Can we improve the algorithm to obtain an $O\left(\frac{\log n}{\log \log n}\right)$-depth AC circuit for approximating powers of substochastic matrices? We need to mention that this does not imply BPL can be computed by $O\left(\frac{\log n}{\log \log n}\right)$-depth AC circuits since we do not know whether L can be computed by $O\left(\frac{\log n}{\log \log n}\right)$-depth AC circuits.

References

1 AmirMahdi Ahmadinejad, Jonathan A. Kelner, Jack Murtagh, John Peebles, Aaron Sidford, and Salil P. Vadhan. High-precision estimation of random walks in small space. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1295–1306. IEEE, 2020. doi:10.1109/FOCS46700.2020.00123.

2 Miklós Ajtai. Approximate counting with uniform constant-depth circuits. In Jin-Yi Cai, editor, *Advances In Computational Complexity Theory, Proceedings of a DIMACS Workshop, New Jersey, USA, December 3-7, 1990*, volume 13 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 1–20. DIMACS/AMS, 1990. doi:10.1090/DIMACS/013/01.

3 Miklós Ajtai and Michael Ben-Or. A theorem on probabilistic constant depth computations. In Richard A. DeMillo, editor, *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1984, Washington, DC, USA*, pages 471–474. ACM, 1984. doi:10.1145/800057.808715.

4 Roy Armoni. On the derandomization of space-bounded computations. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 47–59. Springer, 1998.

5 László Babai, Noam Nisant, and Márió Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, 45(2):204–232, 1992.

6 Mark Braverman, Gil Cohen, and Sumegha Garg. Pseudorandom pseudo-distributions with near-optimal error for read-once branching programs. *SIAM Journal on Computing*, 49(5):STOC18–242, 2019.

7 Eshan Chattopadhyay and Jyun-Jie Liao. Optimal error pseudodistributions for read-once branching programs. In *35th Computational Complexity Conference*, 2020.

8 Lijie Chen, William M. Hoza, Xin Lyu, Avishay Tal, and Hongxun Wu. Weighted pseudorandom generators via inverse analysis of random walks and shortcutting. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 1224–1239. IEEE, 2023. doi:10.1109/FOCS57990.2023.00072.

9 Kuan Cheng and William M. Hoza. Hitting sets give two-sided derandomization of small space. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 10:1–10:25. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.CCC.2020.10.

10 Gil Cohen, Dean Doron, Oren Renard, Ori Sberlo, and Amnon Ta-Shma. Error reduction for weighted prgs against read once branching programs. In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPIcs*, pages 22:1–22:17. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.CCC.2021.22.

11 Gil Cohen, Dean Doron, Oren Renard, Ori Sberlo, and Amnon Ta-Shma. Error reduction for weighted prgs against read once branching programs. In *Proceedings of the 36th Computational Complexity Conference*, page 1, 2021.

12 Gil Cohen, Dean Doron, Ori Sberlo, and Amnon Ta-Shma. Approximating iterated multiplication of stochastic matrices in small space. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 35–45. ACM, 2023. doi:10.1145/3564246.3585181.

13 Joshua Cook. Size bounds on low depth circuits for promise majority. In Nitin Saxena and Sunil Simon, editors, *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2020, December 14-18, 2020, BITS Pilani, K K Birla Goa Campus, Goa, India (Virtual Conference)*, volume 182 of *LIPIcs*, pages 19:1–19:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPICS.FSTTCS.2020.19.

14 Dean Doron, Edward Pyne, and Roei Tell. Opening up the distinguisher: A hardness to randomness approach for BPL = L that uses properties of BPL. *Electron. Colloquium Comput. Complex.*, TR23-208, 2023. arXiv:TR23-208.

15 Dean Doron and Roei Tell. Derandomization with minimal memory footprint. In Amnon Ta-Shma, editor, *38th Computational Complexity Conference, CCC 2023, July 17-20, 2023, Warwick, UK*, volume 264 of *LIPIcs*, pages 11:1–11:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.CCC.2023.11.

16 Pooya Hatami and William Hoza. Theory of unconditional pseudorandom generators. *Electron. Colloquium Comput. Complex.*, TR23-019, 2023. arXiv:TR23-019.

17 William M. Hoza. Better pseudodistributions and derandomization for space-bounded computation. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pages 28:1–28:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPICS.APPROX/RANDOM.2021.28.

18 William M. Hoza. Recent progress on derandomizing space-bounded computation. *Bull. EATCS*, 138, 2022. URL: http://eatcs.org/beatcs/index.php/beatcs/articℓe/view/728.

19 William M Hoza and David Zuckerman. Simple optimal hitting sets for small-success rl. *SIAM Journal on Computing*, 49(4):811–820, 2020.

20 Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 356–364, 1994.

21 Adam R. Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31(5):1501–1526, 2002. `doi:10.1137/S0097539700389652`.

22 Noam Nisan. Pseudorandom generators for space-bounded computation. *Comb.*, 12(4):449–461, 1992. `doi:10.1007/BF01305237`.

23 Noam Nisan. $\mathsf{RL} \subseteq \mathsf{SC}$. *Comput. Complex.*, 4:1–11, 1994. `doi:10.1007/BF01205052`.

24 Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.

25 Aaron (Louie) Putterman and Edward Pyne. Near-optimal derandomization of medium-width branching programs. In Barna Saha and Rocco A. Servedio, editors, *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 23–34. ACM, 2023. `doi:10.1145/3564246.3585108`.

26 Edward Pyne, Ran Raz, and Wei Zhan. Certified hardness vs. randomness for log-space. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*, pages 989–1007. IEEE, 2023. `doi:10.1109/FOCS57990.2023.00061`.

27 Edward Pyne and Salil Vadhan. Pseudodistributions that beat all pseudorandom generators. In *36th Computational Complexity Conference (CCC 2021)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.

28 Edward Pyne and Salil P. Vadhan. Pseudodistributions that beat all pseudorandom generators (extended abstract). In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPIcs*, pages 33:1–33:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPICS.CCC.2021.33`.

29 Michael E. Saks and Shiyu Zhou. $\mathsf{BP_H SPACE}(S) \subseteq \mathsf{DSPACE}(S^{3/2})$. *J. Comput. Syst. Sci.*, 58(2):376–403, 1999. `doi:10.1006/JCSS.1998.1616`.

30 Emanuele Viola. On approximate majority and probabilistic time. In *22nd Annual IEEE Conference on Computational Complexity (CCC 2007), 13-16 June 2007, San Diego, California, USA*, pages 155–168. IEEE Computer Society, 2007. `doi:10.1109/CCC.2007.16`.

31 Emanuele Viola. Randomness buys depth for approximate counting. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 230–239. IEEE Computer Society, 2011. `doi:10.1109/FOCS.2011.19`.