Electronic Colloquium on Computational Complexity, Report No. 48 (2024)

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

Kuan Cheng* Yichuan Wang†

2024-03-04

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 $\mathsf{L}\text{-}\mathsf{NC}^1 \subseteq \mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{L}\text{-}\mathsf{AC}^1$. In this work we will show that $\mathsf{BPL} \subseteq \mathsf{L}\text{-}\mathsf{AC}^1$, which was not known before. Our proof is based on modifying the Richardson Iteration method for boosting precision in approximating matrix powering, which was developed in a line of works [AKM+20][PV21][CDR+21][CDST22][PP22][CHL+23]. We also improve the algorithm for approximating counting in low-depth L-uniform AC circuit from *additive error* setting to *multiplicative error* setting.

*Center on Frontiers of Computing Studies, Peking University. ckkcdh@pku.edu.cn

†Institute for Interdisciplinary Information Sciences, Tsinghua University. yichuan-21@mails.tsinghua.edu.cn.

ISSN 1433-8092

1 Introduction

BPL is the class of languages that can be computed by a randomized logspace Turing Machine with error probability $\leq 1/3$, here by *randomized* we mean the TM has read-once access to a random tape. We also require that the TM halts on any random tape. Whether $\mathsf{BPL} \overset{?}{=} \mathsf{L}$, or *space-bounded derandomization*, is a big open problem in theoretical computer science. Most believe that $\mathsf{L} = \mathsf{BPL}$ is true. Different from the time-bounded derandomization, we even do not know whether $\mathsf{L} = \mathsf{NL}$ can imply $\mathsf{L} = \mathsf{BPL}$. But on the other hand, there is no known barrier for proving $\mathsf{L} = \mathsf{BPL}$. The current optimal upper-bound for BPL against space-bounded computation is $\mathsf{BPL} \subseteq \mathsf{SPACE}\left[(\log n)^{3/2}/\sqrt{\log\log n}\right]$ [Hoz21].

We also consider consider the relation between L and L-uniform low-depth circuit complexity classes. It is well known that $\mathsf{L\text{-}NC}^1 \subseteq \mathsf{L} \subseteq \mathsf{NL} \subseteq \mathsf{L\text{-}AC}^1$, here $\mathsf{L\text{-}NC}^1$ and $\mathsf{L\text{-}AC}^1$ are complexity classes of logspace-uniform $O(\log n)$-depth NC and AC circuits. We observe that under the conjectured $\mathsf{L} = \mathsf{BPL}$, or even weaker, $\mathsf{BPL} \subseteq \mathsf{NL}$, we should have $\mathsf{BPL} \subseteq \mathsf{L\text{-}AC}^1$. In this work, we will unconditionally prove that $\mathsf{BPL} \subseteq \mathsf{L\text{-}AC}^1$, which was unknown before. See Figure 1 for a visualization of the known relations between the complexity classes. On the other hand, we mention that the inclusion $\mathsf{BPL} \subseteq \mathsf{AC}^1$ for nonuniform AC^1 is obvious. By $\mathsf{L} \subseteq \mathsf{AC}^1$ we know $\mathsf{BPL} \subseteq \mathsf{BP} \cdot \mathsf{AC}^1$, then apply the nonuniform derandomize for AC in [AB84] we know $\mathsf{BPL} \subseteq \mathsf{BP} \cdot \mathsf{AC}^1 = \mathsf{AC}^1$.

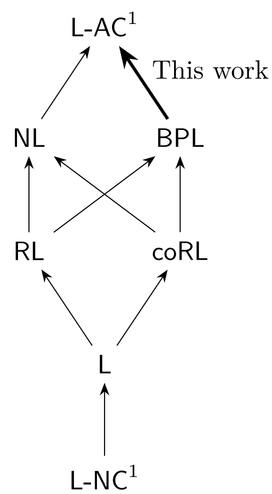

Figure 1: Relation of Complexity Classes. $\mathsf{A} \to \mathsf{B}$ means $\mathsf{A} \subseteq \mathsf{B}$.

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: let $\mathbf{A}_{i,j}$ be the probability that on state i, goes to state j in one step. Note that we must arrive at *accept* or *reject* state in 2^s steps, so we only need to approximate $\mathbf{A}^{2^s}$. [SZ99] use this idea to prove that $\mathsf{BPL} \subseteq \mathsf{L}^{3/2}$. More generally, approximating $\mathbf{A}^n$ for $\mathbf{A} \in \mathbb{R}^{w \times w}$ can be done in space $O\left((\log n)^{3/2} + \sqrt{\log n} \cdot \log w\right)$.

[CDST22] and [PP22] independently discovered how to improve [SZ99]'s result to $\widetilde{O}(\log n + \sqrt{\log n} \cdot \log w)$. The main idea in [CDST22][PP22] is using Richardson Iteration to boost precision. Consider the problem of approximating $\mathbf{X}^{-1}$ for invertible matrix $\mathbf{X}$. Assume we already have a matrix $\mathbf{Y}$, which is an approximation of $\mathbf{X}^{-1}$ such that $\|\mathbf{I} - \mathbf{YX}\| < \varepsilon$. Then we can rewrite $\mathbf{XX}^{-1} = \mathbf{I}$ as

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

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

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

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

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

Needless to say, approximating $\mathbf{A}, \mathbf{A}^2, \cdots, \mathbf{A}^n$ does not necessarily need to go through the framework of approximating the inversion of a matrix. We developed a more efficient iteration algorithm for boosting precision in Section 4, which is the main ingredient of our proof of $\mathsf{BPL} \subseteq \mathsf{L}\text{-}\mathsf{AC}^1$. Our new iteration keeps using the idea of boosting precision via numerical analysis techniques, but does not rely on the framework of approximating inversion of matrix.

We need to mention another setting which considers the multiplication of many distinct matrices, i.e., iterated matrix multiplication. This corresponds to the *read-once branching program* model. Iterated matrix multiplication asks us to approximate $\mathbf{A}_1\mathbf{A}_2 \cdots \mathbf{A}_n$ for given $\mathbf{A}_1, \cdots, \mathbf{A}_n \in \mathbb{R}^{w \times w}$. Actually the methods in [SZ99][CDST22][PP22] can also work for iterated matrix multiplications. [CDST22] and [PP22] again achieve space complexity $\widetilde{O}(\log n + \sqrt{\log n} \cdot \log w)$ for this setting. In the rest of our paper we will only consider matrix powering.

Another side of $\mathsf{BPL} \subseteq \mathsf{L}\text{-}\mathsf{AC}^1$ is on the power of $\mathsf{L}\text{-}\mathsf{AC}$ circuits. Our main tool is approximating counting in $\mathsf{L}\text{-}\mathsf{AC}$ circuits, we will show 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 3.1). This improves the previous results in [AB84][Ajt90][Vio07][Vio10][Coo20] from *additive error* to *multiplicative error*. The version of *additive error* guarantees $O\left(\frac{\log \frac{n}{b-a}}{\log\log n} + 1\right)$-depth, see Lemma 3.4. Intuitively speaking, $\mathsf{L}\text{-}\mathsf{AC}$ circuit is good at aggregating on *many* inputs, but not good at *high precision*, this is why we need a step of boosting precision.

1.1 Our Result

Theorem 1.1. (Main Theorem) (see also Corollary 5.2) $\mathsf{BPL} \subseteq \mathsf{L}\text{-}\mathsf{AC}^1$.

Theorem 1.2. (Multiplicative Approximate Counting in AC) (see also Theorem 3.1) *Let $n, a, b \in \mathbb{N}$ such that $0 \leq a < b \leq n$. Then there exists a* 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.*

1.2 Related Works

Derandomizing BPL

We investigate some progress towards $\mathsf{BPL} = \mathsf{L}$. For more results not covered, we refer to these surveys [Hoz22][HH23].

[Nis92b] presented a logspace computable pseudorandom generator with seed length $O((\log n)^2)$, which can be used to show $\mathsf{BPL} \subseteq \mathsf{TISP}[\mathrm{poly}(n), O((\log n)^2)]$ [Nis92a]. Later [SZ99] gave an algorithm to balance the "logspace computable" and "seed length $O((\log n)^2)$" and show that $\mathsf{BPL} \subseteq \mathsf{L}^{3/2}$. [Hoz21] improved this upper-bound to $\mathsf{SPACE}\left[(\log n)^{3/2}/\sqrt{\log\log n}\right]$. More generally, [SZ99] showed that approximating $\mathbf{A}^n$ for $\mathbf{A} \in \mathbb{R}^{w\times w}$ can be done in space $O\left((\log n)^{3/2} + \sqrt{\log n}\cdot\log w\right)$. [CDST22][PP22] improves [SZ99]'s result to $\widetilde{O}\left(\log n + \sqrt{\log n}\cdot \log w\right)$ via Richardson Iteration. The usage of Richardson Iteration was developed in a line of works [AKM+20][PV21][CDR+21][CDST22][PP22][CHL+23].

[Pyn23] showed that $\mathsf{BPL} \subseteq \mathsf{CSPACE}[O(\log n), O((\log n)^2)]$ in the catalytic space computation model.

[KvM02] showed that under the assumption that $\mathsf{SPACE}[O(n)]$ requires $2^{\Omega(n)}$ circuit size, we have $\mathsf{L} = \mathsf{BPL}$. [CH20] showed that under the assumption that there exists a black-box hitting-set generator computable in logspace, we have $\mathsf{L} = \mathsf{BPL}$. [DT23][PRZ23][DPT23] further improved the derandomization of BPL under assumptions, for different purposes.

Approximating Counting in AC

Algorithms for approximate counting in AC has been studies in a line of work [AB84][Ajt90][Vio07][Vio10][Coo20]. These previous works focused on distinguishing whether n bits contains $\geq \left(\frac{1}{2}+\varepsilon\right)n$ 1's or $\leq \left(\frac{1}{2}-\varepsilon\right)n$ 1's, which can be thought as *additive error*. The $\mathsf{L}\text{-}\mathsf{AC}^0$ algorithm for distinguishing $\geq 2n/3$ 1's and $\leq n/3$ 1's was developed in [Ajt90].

1.3 Proof Sketch

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

In Section 3 we will prove that deciding whether n bits contains $\leq a$ or $\geq b$ 1's can be done in $\mathrm{poly}(n)$-size $O\left(\frac{\log\frac{b}{b-a}}{\log\log n}+1\right)$-depth, see Theorem 3.1. This will be a building block for approximating matrix operations.

In Section 4 we will develop the core iteration step.

Theorem 1.3. (see also Theorem 4.1) *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a substochastic matrix and $k, t \in \mathbb{N}^*$ 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* [1]

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

Then

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

[1] 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.

Intuitively speaking, we can obtain a good approximation of $\mathbf{A}^{2^k}$ only given these $\mathbf{B}_{k-1}, \cdots, \mathbf{B}_0$, which either has lower accuracy or is approximation of $\mathbf{A}^{2^{k'}}$ for much smaller k'. We will prove that the iteration step can be easily computed in L-AC in Theorem 4.2. We need to mention that only use the original form of Richardson Iteration does not suffice to prove $\mathsf{BPL} \subseteq \mathsf{L}\text{-}\mathsf{AC}^1$.

In Section 5 we will present the complete algorithm. We wish to compute some intermediate matrices $\mathbf{M}(k,t)$ for $k, t \le O(\log n)$, here $\mathbf{M}(k,t)$ is a $1/2^t$-approximation of $\mathbf{A}^{2^k}$. We will use the iteration step developed in Section 4 to show that, given all $\mathbf{M}(k-i, [t/2]+2i)$'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 2.1. **(l1-norm)** *Define the l1-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 l1-norm of a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$ to be

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

Theorem 2.2. *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 $p \in \mathbb{N}^*$, $\|\mathbf{A}^p - \mathbf{B}^p\|_1 \le p \|\mathbf{A}-\mathbf{B}\|_1$.*

Definition 2.3. **(Non-negative Matrix)** *We say a matrix is non-negative if each of its entry is non-negative.*

Definition 2.4. **(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 2.5. **(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 2.6. **(L-uniform AC circuit family)** *For functions $S, d\colon \mathbb{N}^* \to \mathbb{R}^+$, we say a collection of circuits $\{\mathcal{C}_n\}_{n \in \mathbb{N}^*}$ 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 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 2.7. (Complexity Class $\mathsf{L\text{-}AC}^1$) *We say a language L is in class $\mathsf{L\text{-}AC}^1$ if there exists a $\mathrm{poly}(n)$-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 2.8. (GapMaj) *For $n \in \mathbb{N}^*$ and $a, b \in \mathbb{R}$ such that $0 \le a < b \le n$, define $\mathsf{GapMaj}[a, b]$ on n bits as follow:*

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

2.3 Tool: Pairwise Independent Hash Function

We will use pairwise independent hash function as a tool for approximating counting in AC. We shall use the following construction based on convolution, which was also used in [Nis92b].

Definition 2.9. (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 binary representation of $x - 1$, let $a_1 \cdots a_{2\log m} b_1 \cdots b_{\log m}$ be 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 binary representation of $H_m(k, x) - 1$.*

Theorem 2.10. *H_m is Pairwise Independent Hash Function in the following sense: for any $1 \le i < j \le m$, 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 uniform over $[m] \times [m]$.*

3 Approximate Counting in AC

The goal of this Section is to prove Theorem 3.1, which will be a building block for the proof of $\mathsf{BPL} \subseteq \mathsf{L\text{-}AC}^1$.

Theorem 3.1. *Let $n, a, b \in \mathbb{N}$ such that $0 \le a < b \le n$. Then there exists a $\mathrm{poly}(n)$-size $O\left(\frac{\log \frac{b}{b-a}}{\log\log n} + 1\right)$-depth L-uniform AC circuit family $\{\mathcal{C}_{n,a,b}\}$ that computes $\mathsf{GapMaj}[a, b]$ on n bits.*

The proof depends on the next few Lemmas.

Lemma 3.2. [Ajt90] *Let $n \in \mathbb{N}^*$. Then there exists $\mathrm{poly}(n)$-size $O(1)$-depth L-uniform AC circuit family $\{\mathcal{C}_n^{(0)}\}$ that computes $\mathsf{GapMaj}[n/3, 2n/3]$ on n bits.*

Lemma 3.3. (Exact Counting) *Let $n, l \in \mathbb{N}^*$ such that $n \ge l$. Then there exists a $\mathrm{poly}(n)$-size $O\left(\frac{\log l}{\log\log n} + 1\right)$-depth L-uniform AC circuit family $\{\mathcal{E}_{n,l}\}$ such that on l bits of input, $\mathcal{E}_{n,l}$ outputs the exact number of 1's over the input bits, in binary form.*

Proof.

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

View the $O(\log n)$-bit integers as $2^{\left\lceil\sqrt{\log n}\right\rceil}$-base $O(\sqrt{\log n})$-digit integers. Use the grade-school algorithm to sum $O(\sqrt{\log n})$ integers. 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. $\square$

Lemma 3.4. *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 3.2, 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 3.3. Below we only consider $b-a>4\sqrt{n}$.

Let $l:=\left\lceil\frac{12n^2}{(b-a)^2}\right\rceil$. 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 2.9. 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 $x_{H_n(i,1)},\cdots,x_{H_n(i,l)}$ 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 $O\left(\frac{\log l}{\log\log n}+1\right)$-depth L-uniform AC circuit, by Lemma 3.3. Here $O\left(\frac{\log l}{\log\log n}+1\right)=O\left(\frac{\log\frac{n}{b-a}}{\log\log n}+1\right)$.

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 $x_{H_n(i,1)},\cdots,x_{H_n(i,l)}$. Then we have $\mathbb{E}_{i\sim[n^3]}[S_i]=pl$ and $\mathrm{Var}_{i\sim[n^3]}[S_i]\le l$. So if $p\le\frac{a}{n}$, then $\Pr_{i\sim[n^3]}\left[S_i\ge l\cdot\frac{(a+b)}{2n}\right]\le\frac{l}{\left(l\cdot\frac{(b-a)}{2n}\right)^2}=\frac{4n^2}{l(b-a)^2}\le\frac{1}{3}$. Similarly if $p\ge\frac{b}{n}$ then $\Pr_{i\sim[n^3]}\left[S_i\le l\cdot\frac{(a+b)}{2n}\right]\le\frac{1}{3}$. 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. $\square$

Proof of Theorem 3.1.

We will try to reduce to Lemma 3.4. 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 3.4.

Let $l:=\left\lceil\frac{n(b-a)}{2b^2}\right\rceil$. For $i\in[n^3]$, let $y_i:=x_{H_n(i,1)}\vee\cdots\vee x_{H_n(i,l)}$, here H_n is the hash function defined in Definition 2.9. 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 $x_{H_n(i,1)},\cdots,x_{H_n(i,l)}$. Then we have $\mathbb{E}_{i\sim[n^3]}[S_i]=pl$ and $\mathbb{E}_{i\sim[n^3]}[S_i^2]=l(l-1)p^2+lp\le lp+l^2p^2$. Thus by

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

we know: if $p \le \frac{a}{n}$, then $\Pr_{i\sim[n^3]}[S_i \ge 1] \le \frac{la}{n}$; if $p \ge \frac{b}{n}$, then $\Pr_{i\sim[n^3]}[S_i \ge 1] \ge \frac{\left(\frac{lb}{n}\right)^2}{\frac{lb}{n}+\left(\frac{lb}{n}\right)^2} \ge \frac{lb}{n} - \left(\frac{lb}{n}\right)^2$. 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 $\mathsf{GapMaj}\left[\left\lceil n^3 \cdot \frac{la}{n} \right\rceil, \left\lceil n^3 \cdot \left(\frac{lb}{n} - \left(\frac{lb}{n}\right)^2\right) \right\rceil\right]$.

Finally we observe that $\left(\frac{lb}{n} - \left(\frac{lb}{n}\right)^2\right) - \frac{la}{n} = l \cdot \left(\frac{b-a}{n} - \frac{lb^2}{n^2}\right) \ge \frac{n(b-a)}{3b^2} \cdot \frac{b-a}{2n} = \frac{(b-a)^2}{6b^2}$. Thus by Lemma 3.4, $\mathsf{GapMaj}\left[\left\lceil n^3 \cdot \frac{la}{n} \right\rceil, \left\lceil n^3 \cdot \left(\frac{lb}{n} - \left(\frac{lb}{n}\right)^2\right) \right\rceil\right]$ 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. $\square$

4 The Iteration Method

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

Theorem 4.1. (The Iteration) *Let $\mathbf{A} \in \mathbb{R}^{n\times n}$ be a substochastic matrix and $k, t \in \mathbb{N}^*$ 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

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

$\square$

Theorem 4.2. (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 4.1. Let $4\log n \geq d \geq t/10$. Then there exists a* poly(n)*-size $O(d)$-depth* L*-uniform* AC *circuit family $\{\mathcal{I}_{n,k,t,d}\}$ 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 4.2 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 3.1.

Proof of Theorem 4.2.

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 $\mathcal{C}_{n^{201}, l, \lceil l(1+1/2^{20d+10})\rceil}$ (as defined in Theorem 3.1, which has depth $\leq O\left(\frac{d}{\log\log n} + 1\right) \leq O\left(\frac{d}{\log(t+1)}\right))^2$ for $l = 1, 2, \cdots, n^{200}$ over these n^{201} bits. Suppose l_0 is the smallest index such that $\mathcal{C}_{n^{201}, l_0, \lceil l_0(1+1/2^{20d+10})\rceil}$ outputs 0, then we have

$$\frac{l_0 - 1}{n^{200}} < \mathbf{Z}_{i,j} < \frac{l_0\left(1 + \frac{1}{2^{20d+10}}\right)}{n^{200}}$$

and thus[3]

$$\frac{\mathbf{Z}_{i,j}}{1 + \frac{1}{2^{20d+10}}} - \frac{1}{n^{100}} \leq \frac{1}{n^{100}}\left\lfloor\frac{l_0}{n^{100}}\right\rfloor \leq \mathbf{Z}_{i,j}.$$

Use $\lfloor l_0/n^{100}\rfloor/n^{100}$ 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 3.4 to Theorem 3.1.

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 $\leq t(1/2^{20d+10} + 1/n^{99}) \leq 1/2^{19d+5}$.

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 $\left\|\mathbf{D}_i - \widetilde{\mathbf{D}_i}\right\|_1 \leq 1/2^{19d+5}$ and $\left\|\mathbf{D}'_i - \widetilde{\mathbf{D}'_i}\right\|_1 \leq 1/2^{19d+5}$.

---

[2] In Theorem 3.1 we take $(a,b) = (l, \lceil l(1+1/2^{20d+10})\rceil)$, and then $\frac{b}{b-a} \leq O(d)$.

[3] Since $n^{200}\mathbf{Z}_{i,j}$ is an integer, we have $\frac{l_0-1}{n^{200}} < \mathbf{Z}_{i,j} \implies \frac{l_0}{n^{200}} \leq \mathbf{Z}_{i,j}$.

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 4.2.

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

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

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

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

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

5 The Complete Algorithm

Theorem 5.1. *Let n be a power of 2. Then there exists a $\mathrm{poly}(n)$-size $O(\log n)$-depth L-uniform AC circuit family $\{\mathcal{M}_n\}$[4] 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 $\left\|\mathbf{M}-\mathbf{A}^n\right\|_1\le 1/n$.*

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 4.1, take the same $n,\mathbf{A}$ and take $k:=k_0$, take $\mathbf{B}_{k-i}:=\mathbf{M}(k-i,[t_0/2]+2i)$ for $1\le i\le k$. Then we can take $\varepsilon_{k-i}:=1/2^{[t_0/2]+2i}$ for $1\le i\le k-1$ and $\varepsilon_0=0$. Now we will invoke Theorem 4.1, 4.2 by choosing parameter t properly according to the following two cases.

Case 1. $k\le 2t_0+2$.
Take the parameter t in Theorem 4.1 to be $t:=k$. Then

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

In Theorem 4.2 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 4.1, 4.2. Given $\mathbf{B}_{k-1},\cdots,\mathbf{B}_0$, use $\mathcal{I}_{n,k_0,k_0,t_0}$ (defined in Theorem 4.2) 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 4.1. Then

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

In Theorem 4.2 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 $\left\|\mathbf{C}'-\mathbf{A}^{2^k}\right\|_1\le 1/2^{t_0}$. ☐

[4] 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 $\mathbf{M}(k_0 - i, [t_0/2] + 2i)$ 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). $\square$

Corollary 5.2. $\mathsf{BPL} \subseteq \mathsf{L}\text{-}\mathsf{AC}^1$.

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 approximating 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 circuit?

References

[AB84] 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.

[Ajt90] 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.

[AKM+20] 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.

[CDR+21] 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.

[CDST22] Gil Cohen, Dean Doron, Ori Sberlo, and Amnon Ta-Shma. Approximating iterated multiplication of stochastic matrices in small space. *Electron. Colloquium Comput. Complex.*, TR22-149, 2022.

[CH20] Kuan Cheng and William Hoza. Hitting sets give two-sided derandomization of small space. *Electron. Colloquium Comput. Complex.*, TR20-016, 2020.

[CHL+23] 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.

[Coo20] Joshua Cook. Size bounds on low depth circuits for promise majority. *Electron. Colloquium Comput. Complex.*, TR20-122, 2020.

[DPT23] 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.

[DT23] 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.

[HH23] Pooya Hatami and William Hoza. Theory of unconditional pseudorandom generators. *Electron. Colloquium Comput. Complex.*, TR23-019, 2023.

[Hoz21] William Hoza. Better pseudodistributions and derandomization for space-bounded computation. *Electron. Colloquium Comput. Complex.*, TR21-048, 2021.

[Hoz22] William Hoza. Recent progress on derandomizing space-bounded computation. *Electron. Colloquium Comput. Complex.*, TR22-121, 2022.

[KvM02] 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.

[Nis92a] Noam Nisan. $\mathsf{RL} \subseteq \mathsf{SC}$. In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, pages 619–623. ACM, 1992.

[Nis92b] Noam Nisan. Pseudorandom generators for space-bounded computation. *Comb.*, 12(4):449–461, 1992.

[PP22] Aaron (Louie) Putterman and Edward Pyne. Near-optimal derandomization of medium-width branching programs. *Electron. Colloquium Comput. Complex.*, TR22-150, 2022.

[PRZ23] 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.

[PV21] Edward Pyne and Salil P. Vadhan. Pseudodistributions that beat all pseudorandom generators. *Electron. Colloquium Comput. Complex.*, TR21-019, 2021.

[Pyn23] Edward Pyne. Time-space tradeoffs for BPL via catalytic computation. *Electron. Colloquium Comput. Complex.*, TR23-168, 2023.

[SZ99] Michael E. Saks and Shiyu Zhou. $\mathsf{BP_HSPACE}(S) \subseteq \mathsf{DSPACE}(S^{3/2})$. *J. Comput. Syst. Sci.*, 58(2):376–403, 1999.

[Vio07] 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.

[Vio10] Emanuele Viola. Randomness buys depth for approximate counting. *Electron. Colloquium Comput. Complex.*, TR10-175, 2010.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il