I have a great interest in understanding the power of different computation resources and the connections between them, and during my doctoral study and research I am particularly interested in space-bounded computation. The major reason is that quantum computers nowadays (and in the foreseeable near future) are built on physical systems with limited number of controllable qubits, and thus have very small quantum memory size compared to the classical RAMs. There is hence a crucial need for understanding what can and cannot be done with memory-bounded quantum computers, and whether our previous knowledge about quantum computation still holds in the small-space settings. From the perspective of a theorist, space complexity is interesting in its own right, and is closely related to many aspects in both classical and quantum complexity theory.

During my PhD years, I worked with my collaborators to show that space-bounded computation has unexpected powers. We proved in [3] that intermediate measurements can be eliminated in $\text{BQL}$ (bounded-error quantum logspace) circuits, without adding too much ancilla qubits to the quantum space. We also proved in [4] that every problem in $\text{BPL}$ (bounded-error probabilistic logspace) needs only $O(\log n)$ truly random bits, which has further implications on pseudorandom generators and derandomization of logspace computation.

In the sections below, I will present more results we proved, and questions I would like to answer during my fellowship, that may lead to unconditionally provable quantum advantages for space-bounded computation.

### Learning with Quantum Memory

Since Raz’s breakthrough result on time-space lower bound for learning [7], there are many follow-up works that generalize and strengthen this result to different learning models and broader classes of learning problems. This line of works open up the possibility of showing unconditionally provable quantum advantage for space-bounded learning, by answering the following question:

**Question 1.** Find a classical learning problem with sample size $n$, that can be learned using a quantum computer of space $S$ within $\text{poly}(n)$ samples, while any classical computer of space $\tilde{O}(S)$ requires exponentially many samples to learn.

Such separation was shown in [2] for intrinsic quantum learning problems like shadow tomography. However, it is not suitable for demonstrating quantum advantage, as the classical computers can not even access full information from a sample in the form of a quantum state.

In [3], we showed that Question 1 is probably hard in the regime of $S = O(\log n)$, in the sense that such a separation leads to $\text{promiseBQL} \neq \text{promiseBPL}$. For larger space regimes, we also proved in [6] that parity learning (learning an unknown $x \in \{0, 1\}^n$ with samples of form $(a, a \cdot x)$ for random $a \in \{0, 1\}^n$) with a quantum learner requires either $\Omega(n^2)$ classical memory, or $\Omega(n)$ quantum memory, or $2^{\Omega(n)}$ samples. We conjecture that it can be improved to $\Omega(n^2)$ quantum memory, so that it would not be a candidate problem for Question 1. This conjecture
is an interesting technical side-quest in the pursuit of answer to Question 1, as it implies that we should look for answers outside extractor-based learning problems.

**The Coupon Collector Model**

In an attempt to attack the notoriously hard problem of proving time-space tradeoff lower bounds for decision problems, we proposed in [8] a new model called the coupon collector model. In this model, for a computation task on input $x \in \{0,1\}^n$, in each step instead of querying $x_i$ on a specific index $i$, one receives a uniformly random index $i \in [n]$ along with $x_i$. We showed that to compute any function of high sensitivity with *zero-error*, the space $S$ and time $T$ must satisfy $ST \geq \Omega(n^2)$.

Our results in [8] covers functions like AND which is easy to compute with bounded error in the coupon collector model. On the other hand, we conjecture that the XOR function is still hard with bounded error.

**Question 2.** Show that computing the XOR function on $n$ bits in the coupon collector model with bounded error requires the space $S$ and time $T$ to satisfy $ST \geq \tilde{\Omega}(n^2)$.

We expect to give an affirmative answer to Question 2 in the near future. The natural next step is to examine quantum time-space tradeoffs in the coupon collector model. Apart from proving quantum lower bounds in line with Question 2, another exciting possibility is to answer the following question, after handling the techniques of proving classical bounded-error lower bounds:

**Question 3.** Find a decision problem that exhibits polynomial separation between bounded-error classical computation and bounded-error quantum computation, in the coupon collector model.

**Multi-output Functions**

For computing a multi-output function $f : \{0,1\}^n \to \{0,1\}^m$, their time-space lower bounds are all proved using the method by Borodin and Cook [1]. As the method is a probabilistic argument, it intrinsically provides the same lower bound for deterministic and randomized branching programs by Yao’s minimax principle. This leads to the following question:

**Question 4.** Find a total multi-output function that shows polynomial separation between time-space complexity for deterministic and randomized branching programs.

In an ongoing work with Huacheng Yu, we give a partial answer to Question 4, where we find such a problem and prove the separation in a restricted oblivious query model.

Interestingly, such separation was already known between randomized and quantum computation, presented by the SORTING problem [5]. On the other hand, there is no unified framework for proving quantum time-space tradeoff lower bounds for multi-output function. The few existing lower bounds are proved using ad-hoc methods and the applications are limited, leaving the following question open, which I would like to make contribution to:

**Question 5.** Find a unified framework for, or prove tighter quantum time-space tradeoffs on multi-output functions.

**References**


