An introduction to the recording method

Anand Natarajan

September 17, 2026

Based on notes by Niels Kornerup and many conversations with Chinmay Nirkhe, with input from Tina Zhang and Aparna Gupte.

The compressed oracle technique was introduced by Zhandry [Zha19], who used it to show several quantum query lower bounds with respect to a random oracle that were difficult or impossible to achieve with previously existing techniques. Essentially, Zhandry's key insight was to record oracle locations that are accessed by a quantum query algorithm by purifying the random oracle and measuring it in a special "recording basis" (essentially the Fourier basis for a uniformly random oracle). In this basis, every location in the oracle starts out in a known state $|\perp\rangle$ (the uniform superposition state in the case of uniformly random oracles); whether an oracle location remains in the $|\perp\rangle$ state is a diagnostic for whether it has been meaningfully queried by the algorithm. Using this technique, Zhandry was able to show lower bounds for finding collisions in random functions, that previously were only known in the worst case using delicate polynomial-method techniques.

In this note we present the essence of this technique following the exposition of Hamoudi [Ham23]. Hamoudi dispenses with the "compressed" aspect of Zhandry's technique, which is only relevant for cryptographic applications, and hence prefers to call it the "recording method." He also allows for more general choices of recording bases, which will allow us to handle oracles that are not uniformly random. We will focus in particular on the Grover search problem, representing the random oracle as a biased random distribution over a binary alphabet.

1 The basic idea: purifying a random oracle

Traditionally, an oracle encoding an exponentially long string $x \in \{0,1\}^N$ in a quantum algorithm is modeled by a unitary, e.g. the controlled phase oracle

$$O_x|i\rangle|b\rangle = (-1)^{b \cdot x_i}|i\rangle|b\rangle.$$

To keep track of "how much" a certain index i in the oracle is accessed by the algorithm, a typical argument might use the L_2 weight of the portion of the algorithm's state where the query register is set to i. For instance, this is how the classic BBBV analysis of unstructured search proceeds. In that analysis, one ends up considering a distribution over oracles, and this is done in the straightforward way of treating O as a random variable.

The key idea behind the recording technique is to model randomness of the oracle in a different way, which will give us a much cleaner way to track the way the algorithm accesses the oracle. Specifically, we will *purify* the randomness in the oracle: we will add an ancillary "oracle register" of N qubits containing the truth table of the oracle, which is not accessible to the algorithm except through applications of the oracle unitary O:

$$O|i\rangle|b\rangle|x\rangle = (-1)^{b \cdot x_i}|i\rangle|b\rangle|x\rangle.$$

A random oracle now corresponds to a *coherent superposition* in the oracle register, which starts out unentangled with the algorithm's working register. To track the algorithm's queries to the oracle, instead of looking at the algorithm's state directly, we will look at the *entanglement* between the algorithm and the oracle register.

2 Lower bounds for unstructured search

To illustrate these ideas, let's reprove the BBBV bound for unstructured search, where the oracle is modeled as a random string $x \in \{0,1\}^N$, where each coordinate x_i is indepedently set to 0 with probability $1-p$ and 1 with probability p. Define two orthogonal states

$$|\perp\rangle = \sqrt{1-p}|0\rangle + \sqrt{p}|1\rangle \tag{2.1}$$

$$|\top\rangle = \sqrt{p}|0\rangle - \sqrt{1-p}|1\rangle, \tag{2.2}$$

so that the purified random oracle corresponds to on oracle register in the state $|\perp\rangle^{\otimes N}$. For simplicity, we'll work with the non-controlled version of the oracle, so we have

$$O|i\rangle|x\rangle = (-1)^{x_i}|i\rangle|x\rangle.$$

It will be easy to modify the calculations we do to work for the controlled case, with essentially no change.

A quantum query algorithm can be written as

$$|\psi^T\rangle = U_T O U_{T_1} O \dots O U_0 |0\dots0\rangle|\perp\rangle^{\otimes N}. \tag{2.3}$$

Define the state at time t to be

$$|\psi^t\rangle = U_t O U_{t-1} O \dots O U_0 |0\dots0\rangle|\perp\rangle^{\otimes N}, \tag{2.4}$$

and define Π to be the projector onto states where at least one oracle register is orthogonal to $\perp$. We will use the quantity

$$\|\Pi|\psi^t\rangle\| \tag{2.5}$$

to measure the amount of progress the algorithm has made towards finding a solution. The first claim we will show is that this progress measure does not grow much after each query.

Claim 2.1.

$$\|\Pi|\psi^t\rangle\| \leq \|\Pi|\psi^{t-1}\rangle\| + 2\sqrt{p(1-p)}. \tag{2.6}$$

Proof.

$$\begin{aligned}
\|\Pi|\psi^t\rangle\| &= \|\Pi U_t O|\psi^{t-1}\rangle\| && (2.7)\\
&= \|U_t \Pi O|\psi^{t-1}\rangle\| && (2.8)\\
&= \|\Pi O|\psi^{t-1}\rangle\| && (2.9)\\
&\leq \|\Pi O \Pi|\psi^{t-1}\rangle\| + \|\Pi O(I-\Pi)|\psi^{t-1}\rangle\| && (2.10)\\
&\leq \|\Pi|\psi^{t-1}\rangle\| + \|\Pi O(I-\Pi)|\psi^{t-1}\rangle\| && (2.11)
\end{aligned}$$

Now, the subnormalized state $(I-\Pi)|\psi^{t-1}\rangle$ can be written as

$$(I-\Pi)|\psi^{t-1}\rangle = \sum_i \alpha_i |\phi_i\rangle|i\rangle|\perp\rangle^{\otimes N}, \tag{2.12}$$

where each $|\phi_i\rangle$ is a unit vector and $\sum_i |\alpha_i|^2 \leq 1$. To compute what happens when we apply ΠO to this, let's first compute ΠO's action on a basis state:

$$\begin{aligned}
\Pi O|i\rangle|\perp\rangle^{\otimes N} &= \Pi O|i\rangle|\perp\rangle_1 \ldots |\perp\rangle_{i-1}|\perp\rangle_i|\perp\rangle_{i+1}\ldots|\perp\rangle_N && (2.13)\\
&= \Pi|i\rangle|\perp\rangle_1 \ldots |\perp\rangle_{i-1}\left(\sqrt{1-p}|0\rangle - \sqrt{p}|1\rangle\right)|\perp\rangle_{i+1}\ldots|\perp\rangle_N && (2.14)\\
&= \Pi|i\rangle|\perp\rangle_1 \ldots |\perp\rangle_{i-1}\left((1-2p)|\perp\rangle + 2\sqrt{p(1-p)}|\top\rangle\right)|\perp\rangle_{i+1}\ldots|\perp\rangle_N && (2.15)\\
&= 2\sqrt{p(1-p)}|i\rangle|\perp\rangle_1 \ldots |\perp\rangle_{i-1}|\top\rangle_i|\perp\rangle_{i+1}\ldots|\perp\rangle_N. && (2.16)
\end{aligned}$$

Thus, we obtain that

$$\|\Pi O(I-\Pi)|\psi^{t-1}\rangle\| = 2\sqrt{p(1-p)}\sqrt{\sum_i |\alpha_i|^2} \leq 2\sqrt{p(1-p)}, \tag{2.17}$$

which proves the Claim. □

Now, observe that

$$\|\Pi|\psi^0\rangle\| = 0. \tag{2.18}$$

Thus, by induction, we immediately obtain

$$\|\Pi|\psi^T\rangle\| \leq 2T\sqrt{p(1-p)}. \tag{2.19}$$

It remains to relate this quantity to the success probability of a quantum search algorithm. Define Π_{succ} to be the projector onto states of the algorithm and oracle together, in which the query register contains index i and the ith oracle register is set to 1. Without loss of generality, suppose the algorithm writes its output in the query register. Then the success probability of the algorithm is just

$$p_{succ} = \|\Pi_{succ}|\psi^T\rangle\|^2. \tag{2.20}$$

Now we may calculate.

$$\|\Pi_{succ}|\psi^T\rangle\| \leq \|\Pi_{succ}\Pi|\psi^T\rangle\| + \|\Pi_{succ}(I-\Pi)|\psi^T\rangle\| \tag{2.21}$$

$$\leq 2T\sqrt{p(1-p)} + \|\Pi_{succ}(I-\Pi)|\psi^T\rangle\|. \tag{2.22}$$

To calculate the second term, as before, write

$$(I-\Pi)|\psi^T\rangle = \sum_i \beta_i|\phi_i\rangle|i\rangle|\perp\rangle^{\otimes N}, \tag{2.23}$$

where each $|\phi_i\rangle$ is a unit vector and $\sum_i |\beta_i|^2 \leq 1$. Then we have

$$\Pi_{succ}|i\rangle|\perp\rangle^{\otimes N} = \sqrt{p}|i\rangle|\perp\rangle_1 \ldots |\perp\rangle_{i-1}|1\rangle_i|\perp\rangle_{i+1}\ldots|\perp\rangle_N. \tag{2.24}$$

And therefore,

$$\|\Pi_{succ}(I-\Pi)|\psi^T\rangle\| = \sqrt{p}\cdot\sqrt{\sum_i |\beta_i|^2} \leq \sqrt{p}. \tag{2.25}$$

Putting it all together, we have

$$p_{succ} \leq \left(2T\sqrt{p(1-p)} + \sqrt{p}\right)^2 \tag{2.26}$$

3 Searching for multiple marked items

We will now study a problem which is *not* as easy to analyze with standard techniques, which is the search problem for multiple marked items in Grover search. This is loosely based on a recent work of Hamoudi and Magniez [HM23].

Theorem 3.1. *Any T query algorithm succeeds in finding k distinct preimages with probability at most*

$$\left(2\sqrt{p}\left(\frac{eT}{k}\right)\right)^{2k}.$$

Proof. We will proceed roughly as in the Grover search case,

Let $\Pi_{\geq k}$ be the projector onto oracle states containing at least k entries that are orthogonal to $\perp$.

Claim 3.2.

$$\|\Pi_{\geq k}|\psi^t\rangle\| \leq \|\Pi_{\geq k}|\psi^{t-1}\rangle\| + 2\sqrt{p(1-p)}\|\Pi_{\geq k-1}|\psi^{t-1}\rangle\|. \tag{3.1}$$

Proof.

$$\|\Pi_{\geq k}|\psi^t\rangle\| = \|\Pi_{\geq k}U_t O|\psi^{t-1}\rangle\| \tag{3.2}$$

$$= \|U_t\Pi_{\geq k}O|\psi^{t-1}\rangle\| \tag{3.3}$$

$$= \|\Pi_{\geq k}O|\psi^{t-1}\rangle\| \tag{3.4}$$

$$\leq \|\Pi_{\geq k}R\Pi_{\geq k}|\psi^{t-1}\rangle\| + \|\Pi_{\geq k}O(I-\Pi_{\geq k})|\psi^{t-1}\rangle\| \tag{3.5}$$

$$\leq \|\Pi_{\geq k}|\psi^{t-1}\rangle\| + \|\Pi_{\geq k}O(I-\Pi_{\geq k})|\psi^{t-1}\rangle\| \tag{3.6}$$

The subnormalized state $(I - \Pi_{\geq k})|\psi^{t-1}\rangle$ is supported on oracle states that have at most $k-1$ entries not equal to $\perp$. Using our immense cleverness we notice that a single invocation of O can only increase the number of entries equal 1 to by one. Thus

$$\Pi_{\geq k} O = \Pi_{\geq k} O \Pi_{\geq k-1},$$

so

$$\Pi_{\geq k} O (I - \Pi_{\geq k}) = \Pi_{\geq k} O \Pi_{=k-1}. \tag{3.7}$$

A similar calculation to what we had earlier lets us bound

$$\|\Pi_{\geq k} O \Pi_{=k-1} |\psi^{t-1}\rangle\| \leq 2\sqrt{p(1-p)} \|\Pi_{\geq k-1} |\psi^{t-1}\rangle\|. \tag{3.8}$$

This proves the claim. □

Now, observe that in the initial state of the algorithm we have

$$\|\Pi_{\geq 0} |\psi^0\rangle\| = 1,$$

and

$$\|\Pi_{\geq k} |\psi^0\rangle\| = 0$$

for all $k \geq 1$. Thus, we can recursively apply the claim to deduce the bound

$$\|\Pi_{\geq k} |\psi^T\rangle\| \leq \binom{T}{k} (2\sqrt{p(1-p)})^k. \tag{3.9}$$

(Intuitively, we have to "step down" k times in the recursion , incurring a factor of $2\sqrt{p(1-p)}$ each time, and there are $\binom{T}{k}$ choices for where to place the steps.)

Now, to finish up the proof, we need to use these quantities to bound the success probability of the algorithm. Define Π_{succ} to be the projector onto states of the algorithm where the output register contains a list of $\geq k$ distinct indices, and each oracle register corresponding to an index is in the state $|1\rangle$ in the *standard* basis. The success probability of the algorithm is

$$p_{succ} = \|\Pi_{succ} |\psi^T\rangle\|^2.$$

As before, we calculate.

$$\|\Pi_{succ} |\psi^T\rangle\| \leq \|\Pi_{succ} \Pi_{\geq k} |\psi^T\rangle\| + \|\Pi_{succ} (I - \Pi_{\geq k}) |\psi^T\rangle\| \tag{3.10}$$

$$\leq \binom{T}{k} (2\sqrt{p(1-p)})^k + \underbrace{\|\Pi_{succ} (I - \Pi_{\geq k}) |\psi^T\rangle\|}_{\sqrt{p}}. \tag{3.11}$$

The second term is essentially the chance that the algorithm is correct "by chance," given that at least one of the indices in its output corresponds to an oracle register that is in the $|\perp\rangle$ state. We see that this is at most $\sqrt{p}$. Putting everything together, we would get

$$p_{succ} \leq \left(\binom{T}{k} (2\sqrt{p(1-p)})^k + \sqrt{p} \right)^2. \tag{3.12}$$

Now, this bound is *true*, but it's not great, since it doesn't scale exponentially in k. In fact, we were very loose with the second term above.

$$\begin{aligned}
&\|\Pi_{succ}(I-\Pi_{\geq k})|\psi^T\rangle\| \\
&\quad = \|\Pi_{succ}\sum_{j=0}^{k-1}\Pi_{=j}|\psi^T\rangle\| && (3.13)\\
&\quad \leq \|\Pi_{succ}\Pi_{=0}|\psi^T\rangle\| + \sum_{j=1}^{k-1}\|\Pi_{succ}\Pi_{\geq j}|\psi^T\rangle\| && (3.14)\\
&\quad \leq (\sqrt{p})^k + \sum_{j=1}^{k-1}(\sqrt{p})^{k-j}\binom{T}{j}(2\sqrt{p(1-p)})^j && (3.15)\\
&\quad = (\sqrt{p})^k\left(1+\sum_{j=1}^{k-1}\binom{T}{j}(2\sqrt{1-p})^j\right) && (3.16)\\
&\quad \leq (\sqrt{p})^k\cdot 2^{k-1}\cdot\left(\sum_{j=0}^{k-1}\binom{T}{j}\right) && (3.17)\\
&\quad \leq (\sqrt{p})^k\cdot\left(2^{k-1}\left(\frac{eT}{k}\right)^k\right). && (3.18)
\end{aligned}$$

Here in the last step we used the standard[1] bound

$$\sum_{j=0}^{k}\binom{N}{j} \leq \left(\frac{eN}{k}\right)^k, \tag{3.19}$$

and also "inflated" the sum over j to range up to k rather than $k-1$, to make our final expression a little cleaner. This leaves us with a total bound of

$$\begin{aligned}
p_{succ} &\leq \left(\binom{T}{k}(2\sqrt{p(1-p)})^k + \left((\sqrt{p})^k\cdot(2^{k-1}\left(\frac{eT}{k}\right)^k\right)\right)^2 && (3.20)\\
&\leq \left((\sqrt{p})^k 2^k\left(\frac{eT}{k}\right)^k\right)^2 && (3.21)\\
&\leq \left(2\sqrt{p}\left(\frac{eT}{k}\right)\right)^{2k} && (3.22)
\end{aligned}$$

□

[1] I only knew the RHS as a bound for $\binom{N}{k}$ but a proof that this bounds the whole sum may be found in the comment by user "Joe Dohn" at https://mathoverflow.net/questions/17202/sum-of-the-first-k-binomial-coefficients-for-fixed-n.

4 Collision

References

[Ham23] Yassine Hamoudi. PCMI graduate summer school: Quantum query complexity. https://yassine-hamoudi.github.io/pcmi2023/, 2023.

[HM23] Yassine Hamoudi and Frédéric Magniez. Quantum time–space tradeoff for finding multiple collision pairs. *ACM Transactions on Computation Theory*, 15(1-2):1–22, 2023, arXiv:2002.08944.

[Zha19] Mark Zhandry. How to record quantum queries, and applications to quantum indifferentiability. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*, pages 239–268. Springer, 2019. doi:10.1007/978-3-030-26951-7_9.