Our Results

- **Theorem**: For $p > 2$, there is an algorithm using **optimal** $\tilde{O}(n^{1-2/p})$ bits of space and an **update time of** $O(1)$ to approximate $F_p(x)$ up to constant factors

- Improves on poly$(\log n)$ update time of earlier works such as [Andoni, Krauthgamer, Onak '10]

- **Theorem**: For $0 < p < 2$, can approximate $F_p(x)$ up to $1 \pm \varepsilon$ using **optimal** $O(\varepsilon^{-2} \log n)$ bits of space and $O(\log n)$ **update time**

- Valid only for $\varepsilon < 1/n^c$

- Improves on $O(\log^2 n \log\log n)$ update time of [KNPW '11]

14

$$F_p(x) = \sum_i |x[i]|^p$$

Our Results

$$F_p(x) = \sum_i |x[i]|^p$$

- **Theorem**: For $p > 2$, there is an algorithm using **optimal** $\tilde{O}(n^{1-2/p})$ bits of space and an **update time of** $O(1)$ to approximate $F_p(x)$ up to constant factors
  - Improves on poly$(\log n)$ update time of earlier works such as [Andoni, Krauthgamer, Onak '10]
- **Theorem**: For $0 < p < 2$, can approximate $F_p(x)$ up to $1 \pm \varepsilon$ using **optimal** $O(\varepsilon^{-2} \log n)$ bits of space and $O(\log n)$ **update time**
  - Valid only for $\varepsilon < 1/n^c$
  - Improves on $O(\log^2 n \log \log n)$ update time of [KNPW '11]

Our Results