FASTER, FAULT-RESILIENT SUBLINEAR ALGORITHMS*

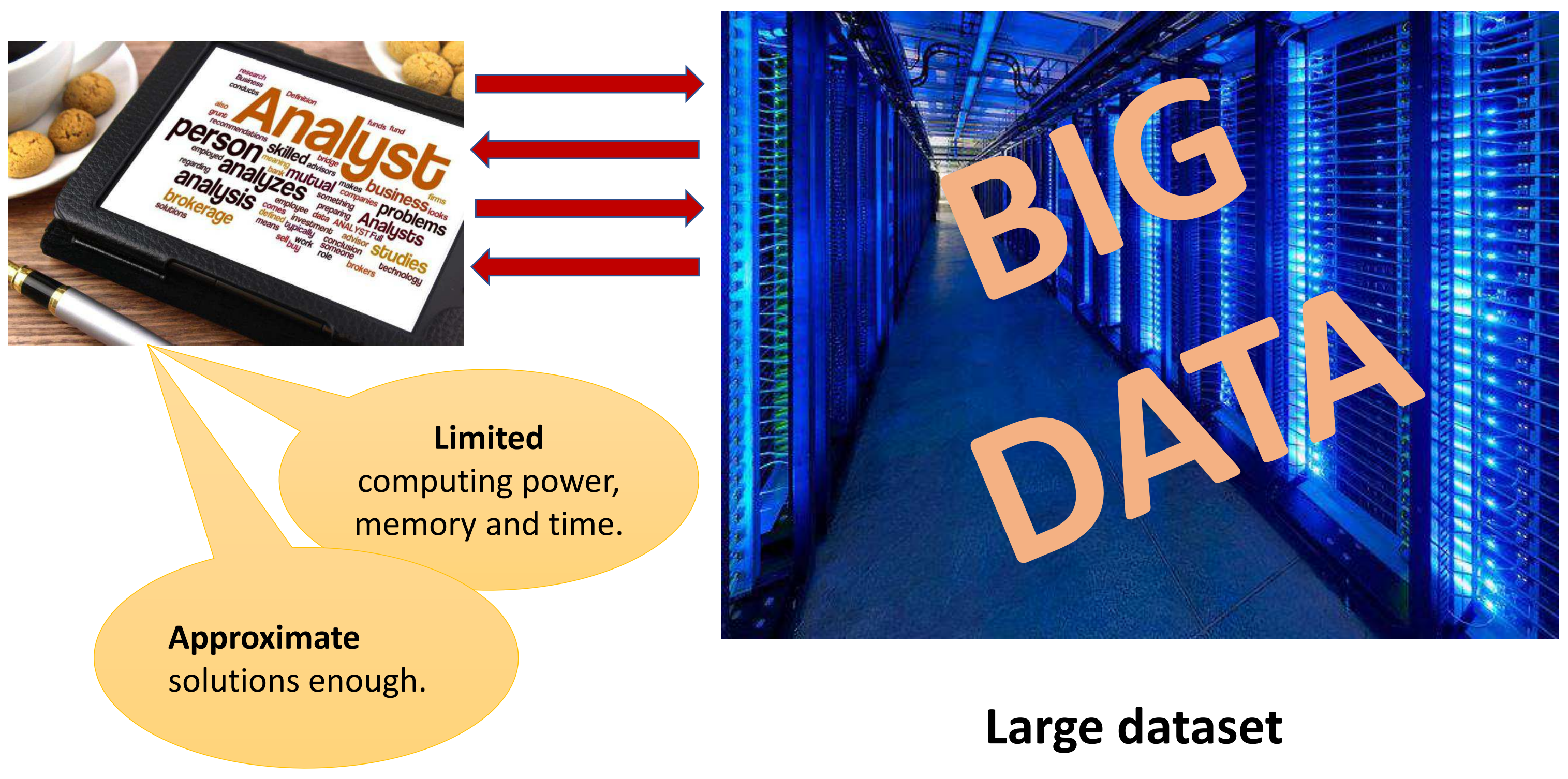
NITHIN M. VARMA

Department of Computer Science, Boston University

Collaborators: Kashyap Dixit, Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, Abhradeep Thakurta

*Part of the work done at the Pennsylvania State University.

Sublinear Algorithms

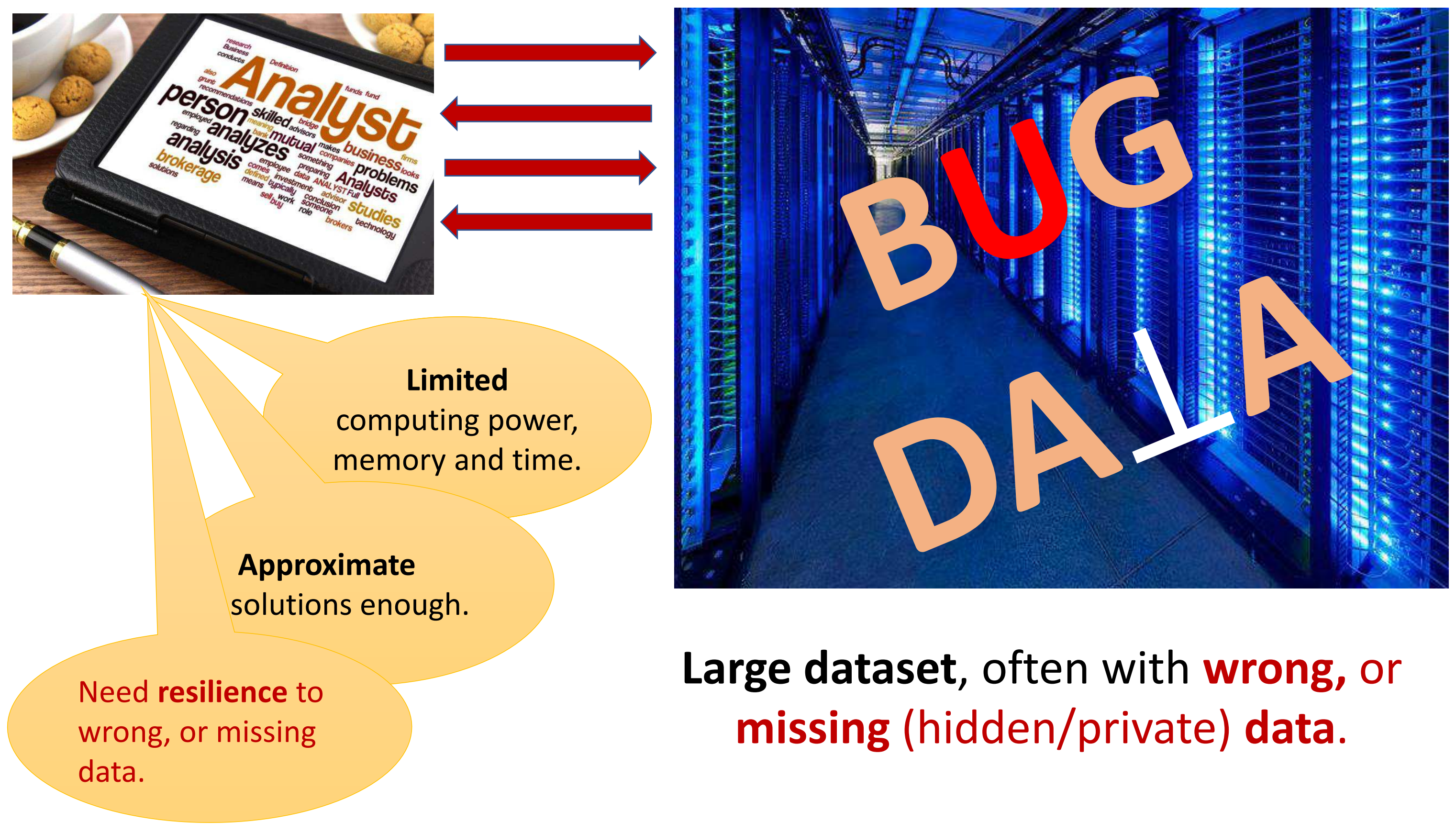


Limited computing power, memory and time.

Approximate solutions enough.

Large dataset

Fault-resilient Sublinear Algorithms: Why?



Limited computing power, memory and time.

Approximate solutions enough.

Need resilience to wrong, or missing data.

Large dataset, often with **wrong,** or **missing** (hidden/private) **data**.

Erasure-resilient Testing: Main Results

Efficient Erasure-resilient Testers

Design of erasure-resilient testers for several important properties with only a $\Theta\left(\frac{1}{1-\alpha}\right)$ factor overhead in the sample complexity (in comparison to standard testers), where α denotes the upper bound on the fraction of worst-case erasures in the input.

Property	Input object	Sample complexity of testing	Sample complexity of erasure-resilient testing
Monotonicity and Lipschitz properties	$f:[n]\to\mathbb{R}$ (real-valued arrays)	$\Theta\left(\frac{\log n}{\epsilon}\right)$ [EKKRV00,F04]	$\Theta\left(\frac{1}{1-\alpha}\cdot\frac{\log n}{\epsilon}\right)$ [DRTV16]
	$f:\ [n]^d\to\mathbb{R}$	$\Theta\left(\frac{d\cdot\log n}{\epsilon}\right)$ [GGLRS00, DGLRRS99,…,CS13, CS14]	$O\left(\frac{1}{1-\alpha}\cdot\frac{d\cdot\log n}{\epsilon}\right)$ for $\alpha=O\left(\frac{\epsilon}{d}\right)$ [DRTV16]
Convexity	$f:[n]\to\mathbb{R}$	$\Theta\left(\frac{\log n}{\epsilon}\right)$ [PRR03]	$\Theta\left(\frac{1}{1-\alpha}\cdot\frac{\log n}{\epsilon}\right)$ [DRTV16]
Monotonicity	$f:N\to\mathbb{R}$, where N is a partial order	$O\left(\left\{\frac{\lvert N\rvert}{\epsilon}\right\}^{1/2}\right)$ [FLNRRS02]	$O\left(\frac{1}{1-\alpha}\cdot\left\{\frac{\lvert N\rvert}{\epsilon}\right\}^{1/2}\right)$ [DRTV16]

Erasure-resilient Testing is Hard in General

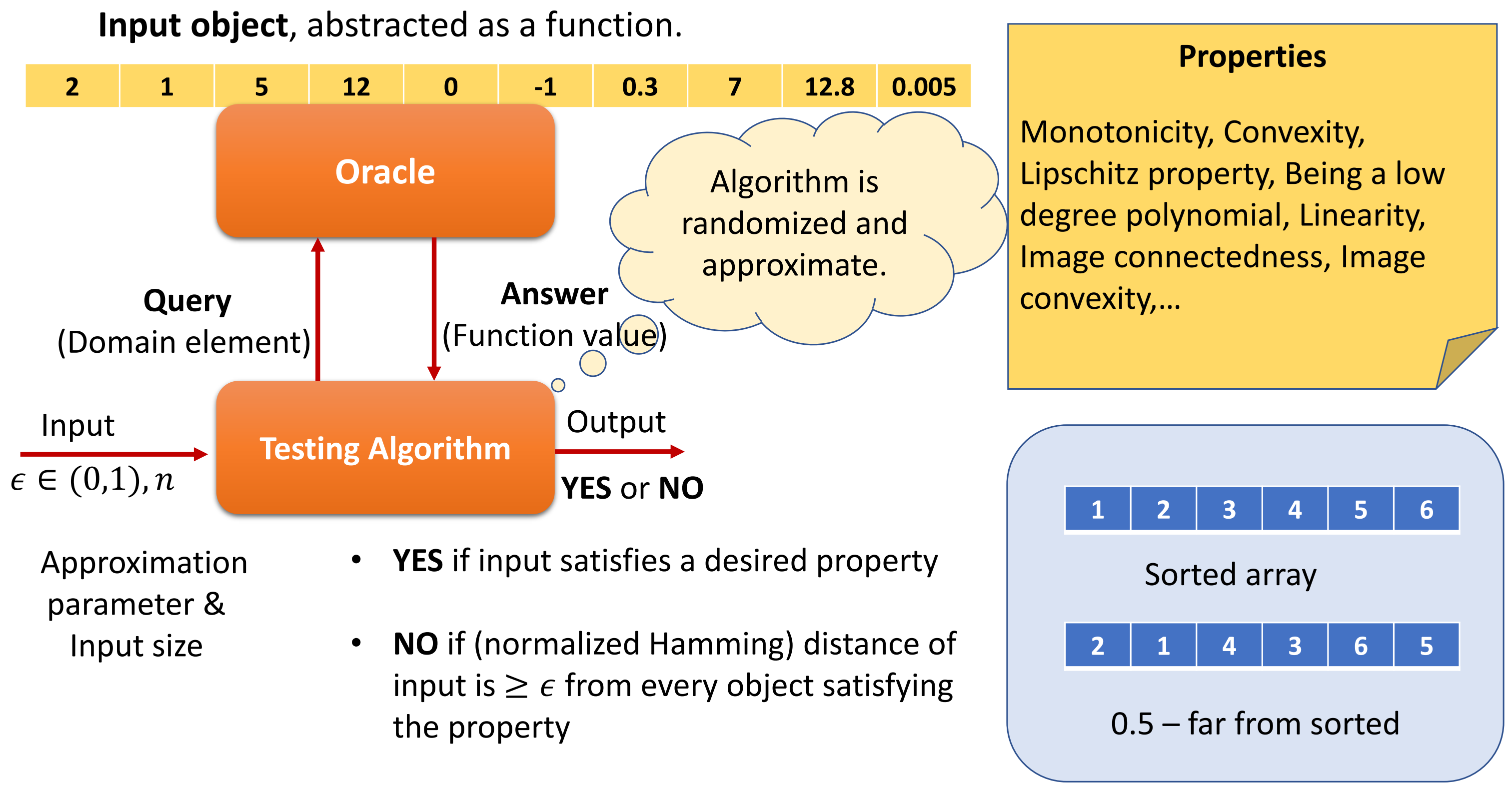
Proof of existence of a property that has a constant sample complexity tester in the absence of erasures but requires polynomially many samples to test in the presence of even a small fraction of worst-case erasures [DRTV16]

(Published in the proceedings of ICALP 2016. Joint work with Kashyap Dixit, Sofya Raskhodnikova and Abhradeep Thakurta.)

Ongoing Work and Future Directions

- Compare the complexity of tolerant testing against erasure-resilient testing with worst-case erasures.
- Compare the complexity of testing with worst-case erasures and testing random erasures.
- Investigate other models of erasures and corruptions.
- Design erasure-resilient testers for other important properties such as that of being a low-degree polynomial and dictatorship.

Property Testing [Rubinfeld, Sudan '96, Goldreich, Goldwasser, Ron '98]


Input object, abstracted as a function.

- **YES** if input satisfies a desired property
- **NO** if (normalized Hamming) distance of input is $\geq\epsilon$ from every object satisfying the property

Properties

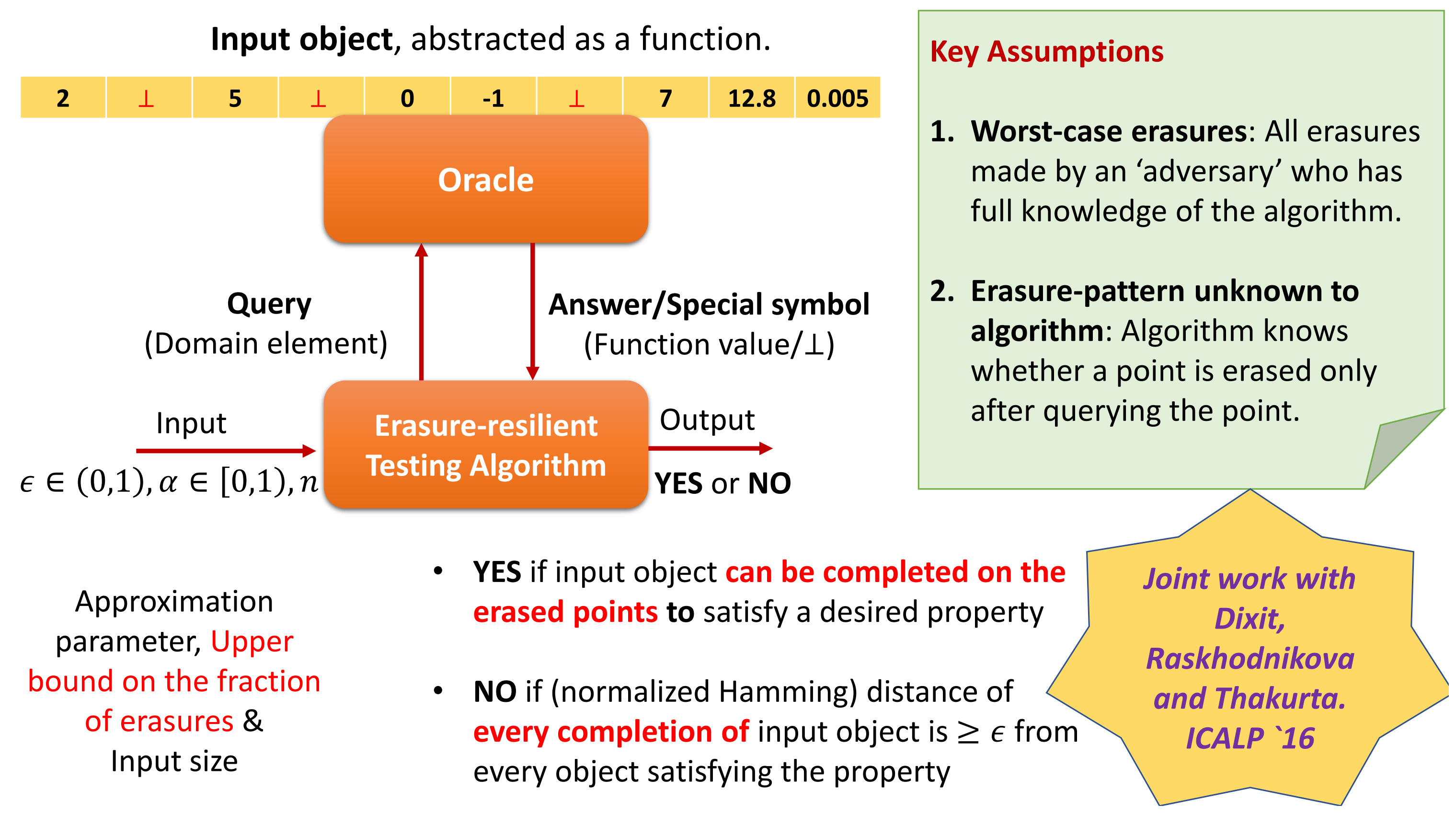
Monotonicity, Convexity, Lipschitz property, Being a low degree polynomial, Linearity, Image connectedness, Image convexity,…

Useful as a routine to efficiently check, before learning (in the PAC model), whether an input satisfies the properties of the classifier that we are trying to learn.

Tolerant Property Testing [Parnas, Ron, Rubinfeld 2006]

Accept inputs that are 'close' to a property and reject inputs that are 'far' from the property. Framework designed to handle 'corruptions' in the input data.

Erasure-resilient Property Testing


Input object, abstracted as a function.

Key Assumptions

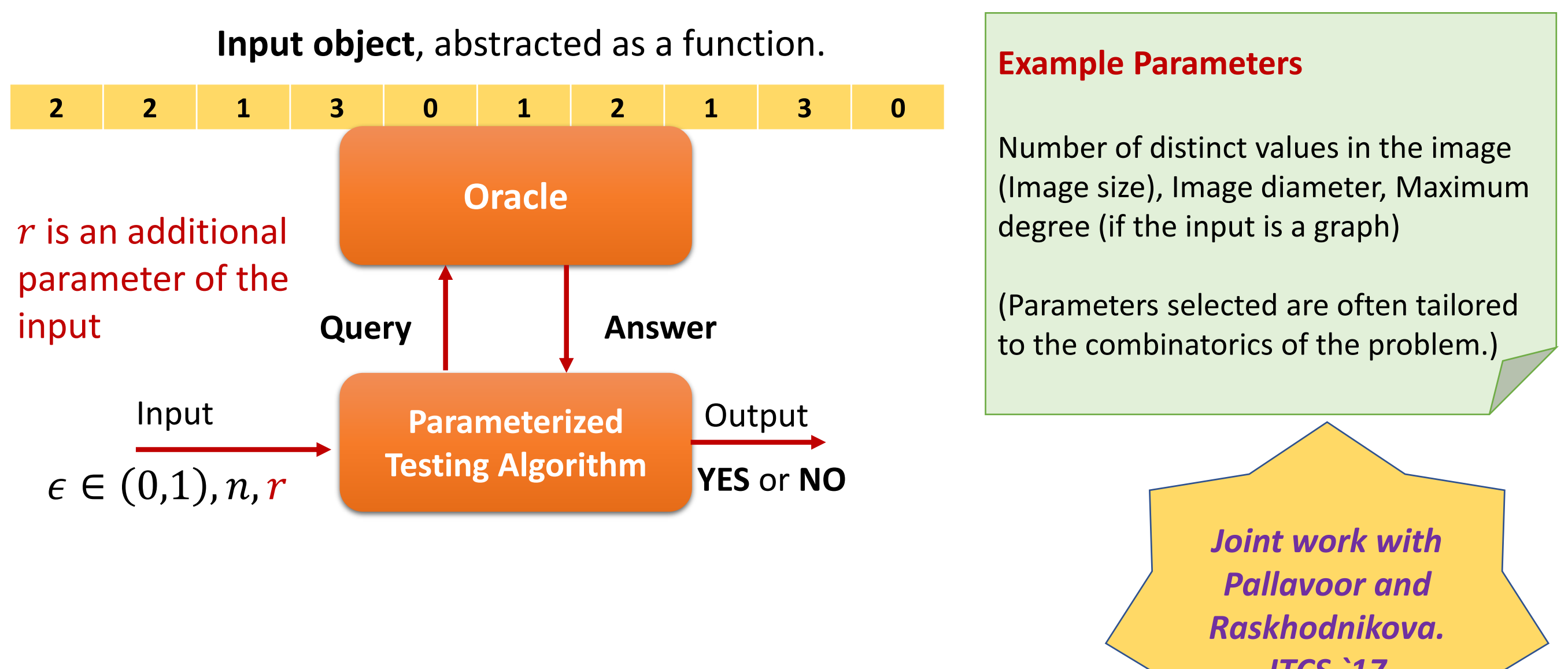
1. **Worst-case erasures**: All erasures made by an 'adversary' who has full knowledge of the algorithm.
2. **Erasure-pattern unknown to algorithm**: Algorithm knows whether a point is erased only after querying the point.

- **YES** if input object **can be completed on the erased points to** satisfy a desired property
- **NO** if (normalized Hamming) distance of **every completion of** input object is $\geq\epsilon$ from every object satisfying the property

Joint work with Dixit, Raskhodnikova and Thakurta. ICALP `16

Faster Sublinear Algorithms? Parameterize!

BIG IDEA: Measure the complexity with respect to parameters other than the input size.


Input object, abstracted as a function.

Example Parameters

Number of distinct values in the image (Image size), Image diameter, Maximum degree (if the input is a graph)

(Parameters selected are often tailored to the combinatorics of the problem.)

Joint work with Pallavoor and Raskhodnikova. ITCS `17

Studied the effectiveness of the image size of a function as the parameter in testing Monotonicity and Convexity!

Property	Sample complexity of testing	Sample complexity of testing parametrized by image size
Sortedness Of n-length arrays with $\leq r$ distinct values	$\Theta\left(\frac{\log n}{\epsilon}\right)$ [EKKRV00,F04]	$O\left(\frac{\log r}{\epsilon}\right)$
	$\Theta(\sqrt{n})$ (using uniform and independent samples)	$\Theta(\sqrt{r})$
Monotonicity Of $f:[n]^d\mapsto\mathbb{R},\ \lvert\mathrm{Im}(f)\rvert\leq r$	$\Theta\left(\frac{d\log n}{\epsilon}\right)$ [GGLRS00, DGLRRS99, …., CS13, CS14]	$\tilde{O}\left(\frac{d\log r}{\epsilon}\right)$
Convexity Of $f:[n]\mapsto\mathbb{R},\ \lvert\mathrm{Im}(f)\rvert\leq r$	$O\left(\frac{\log n}{\epsilon}\right)$ [PRR03]	$O(1)$ (for small r)

(Published in the proceedings of ITCS 2017. Joint work with Ramesh Krishnan S. Pallavoor and Sofya Raskhodnikova.)