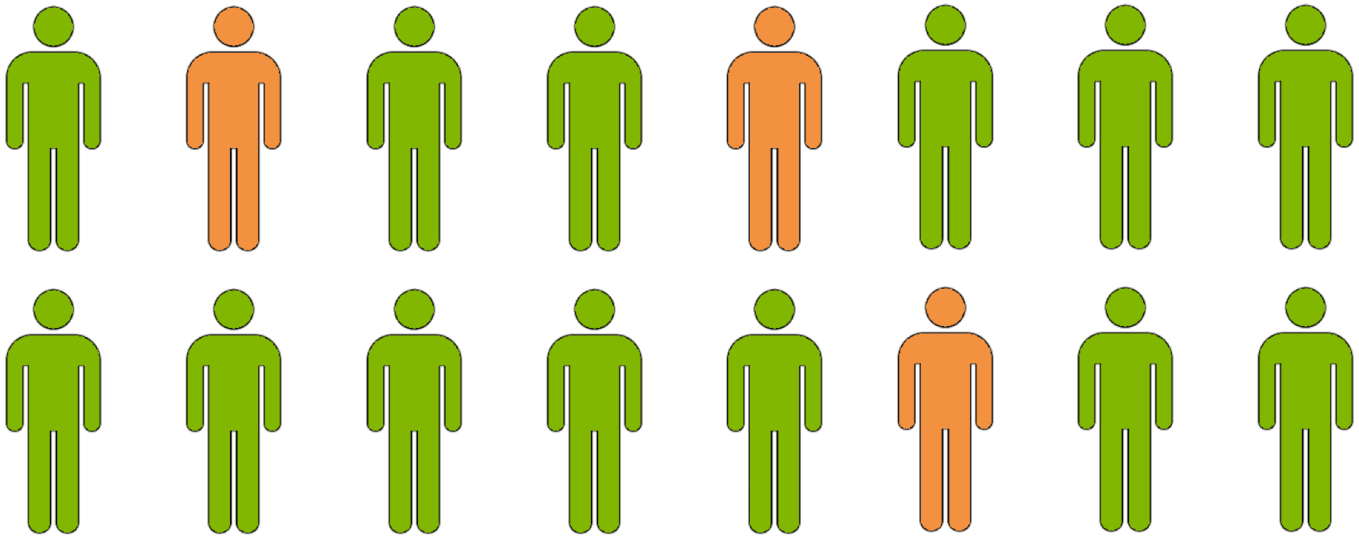
Group Testing

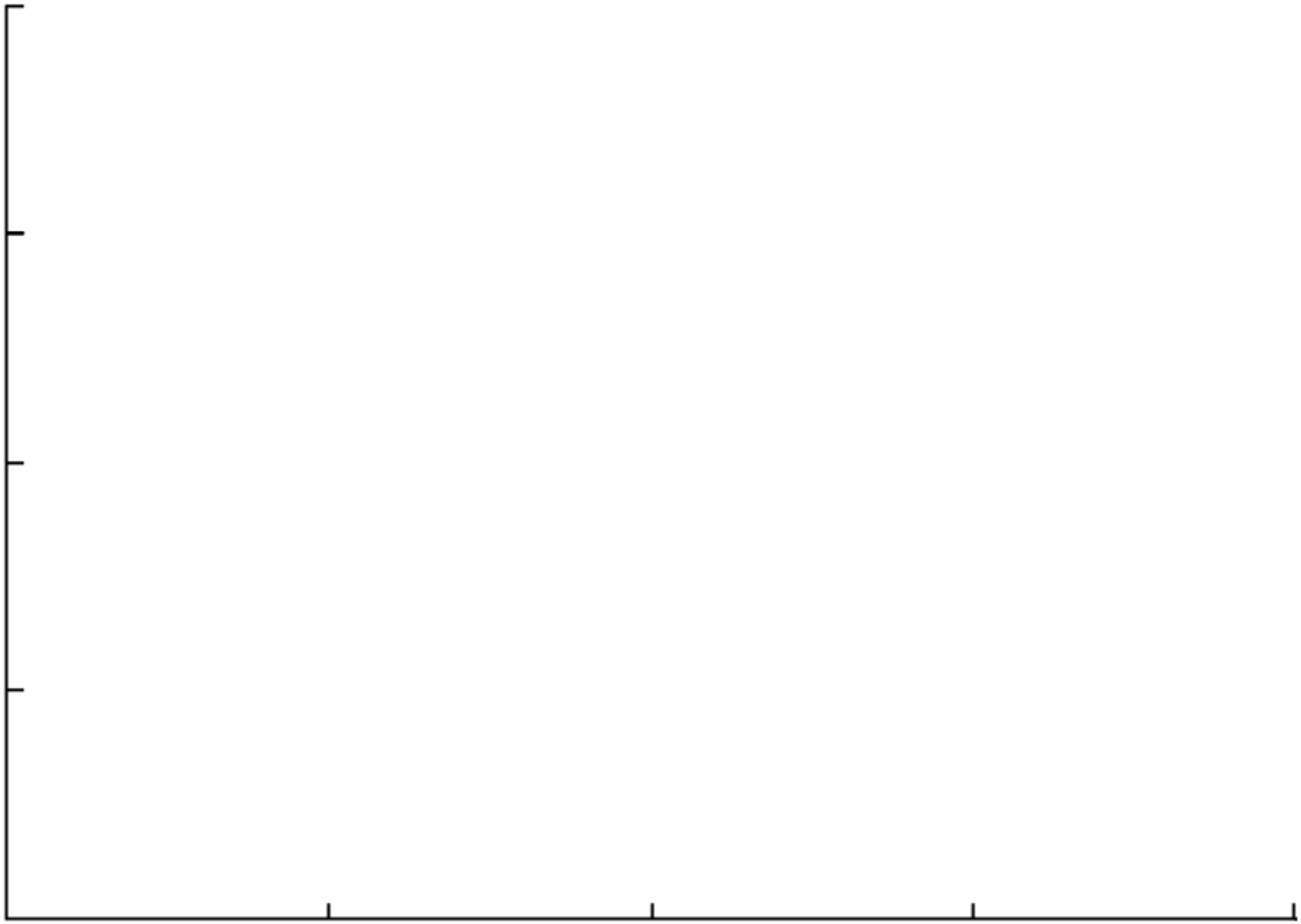
A nonlinear sparse inference problem



Matthew Aldridge
University of Leeds

with Oliver Johnson, Jonathan Scarlett, and Leonardo Baldassini

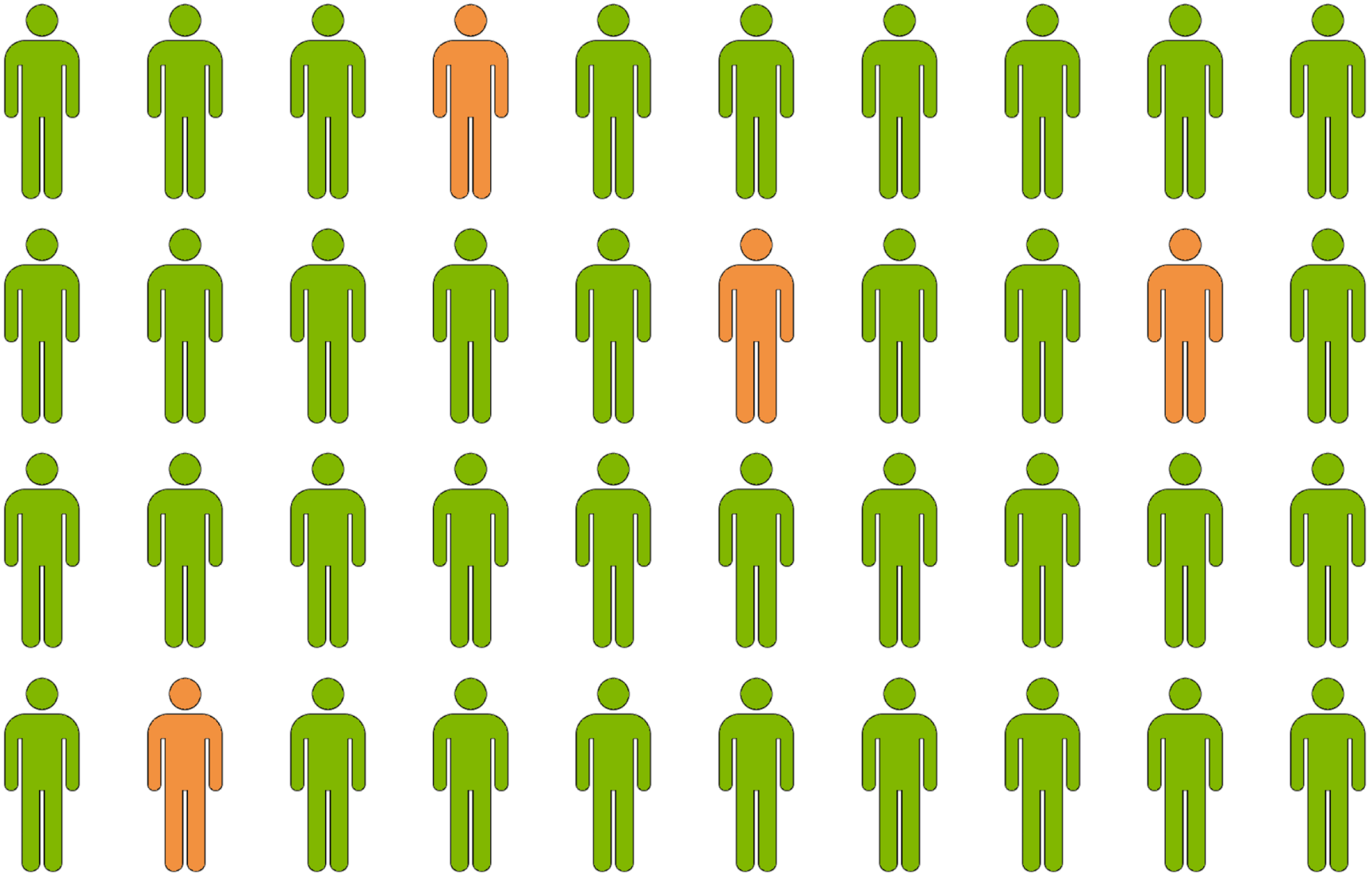
Durham University
December 2018



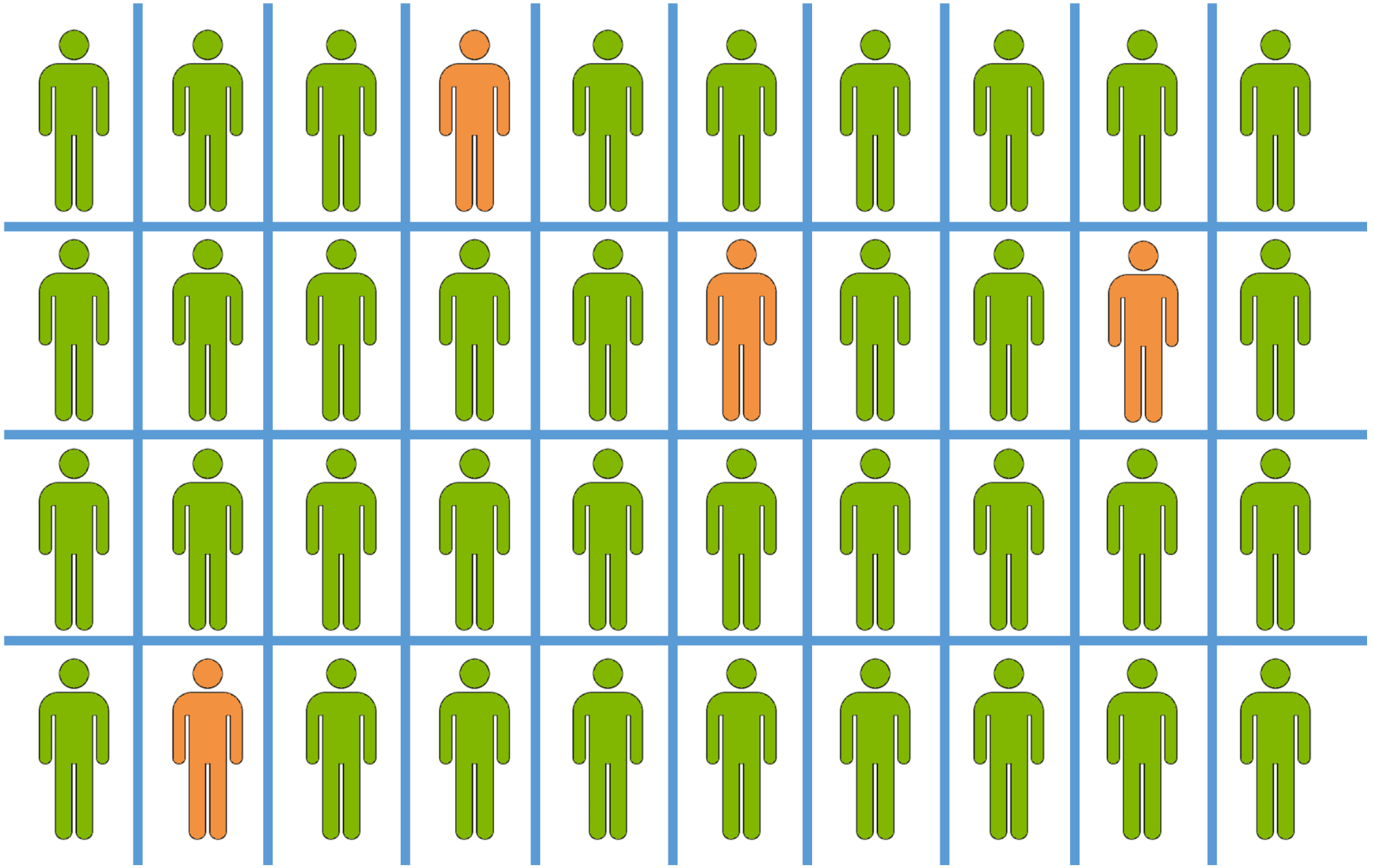
1

What is group testing?

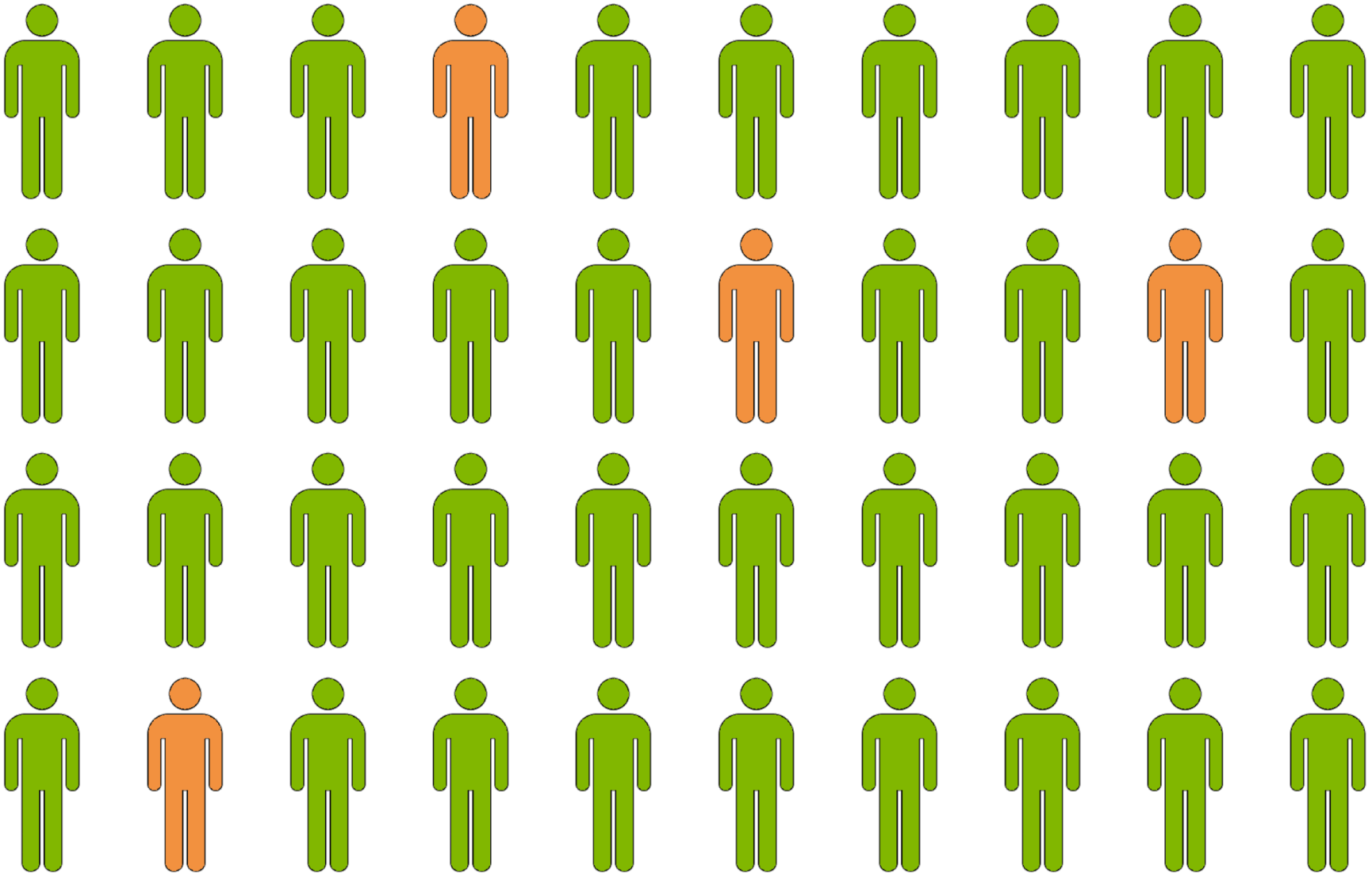
Group testing



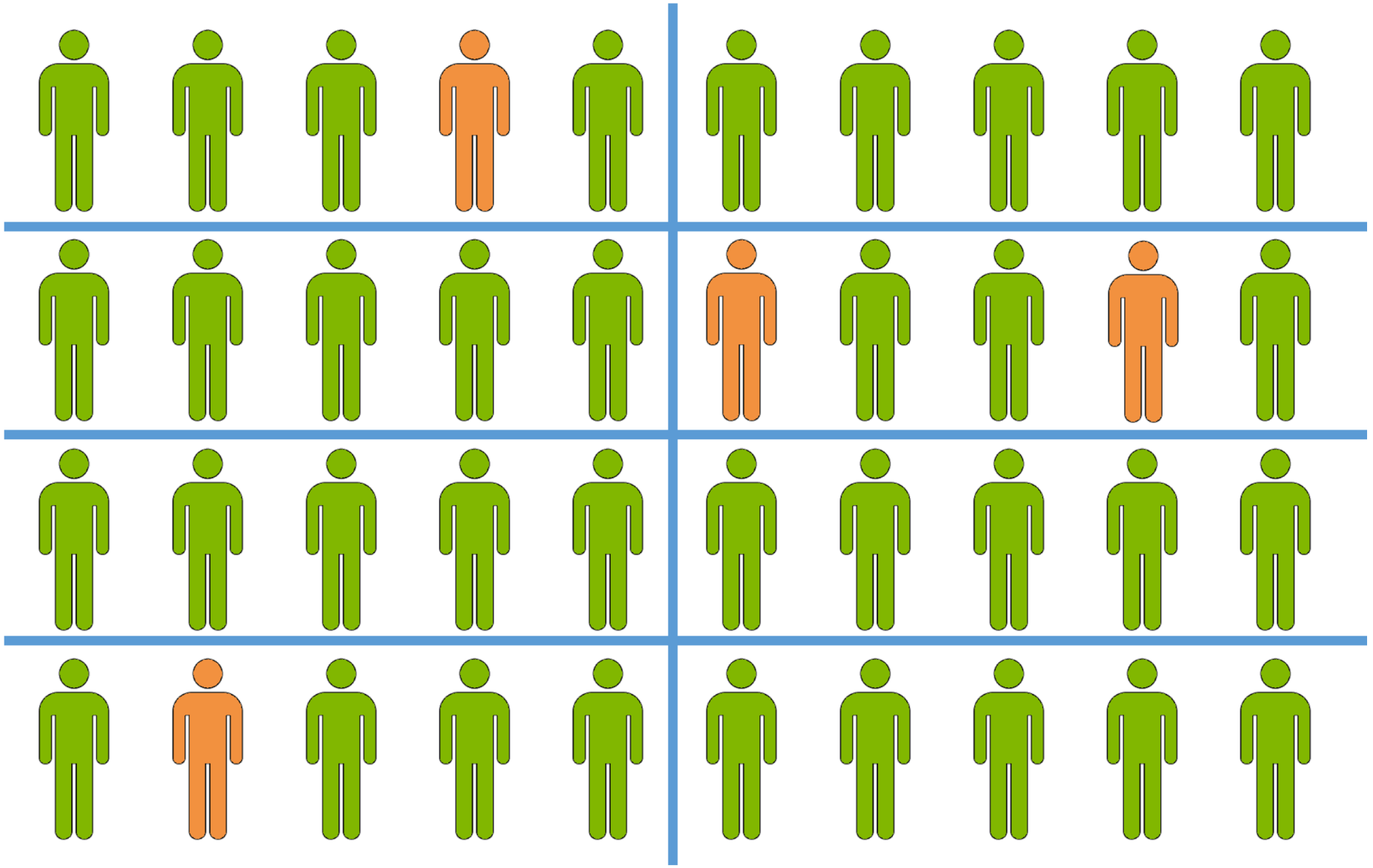
Group testing



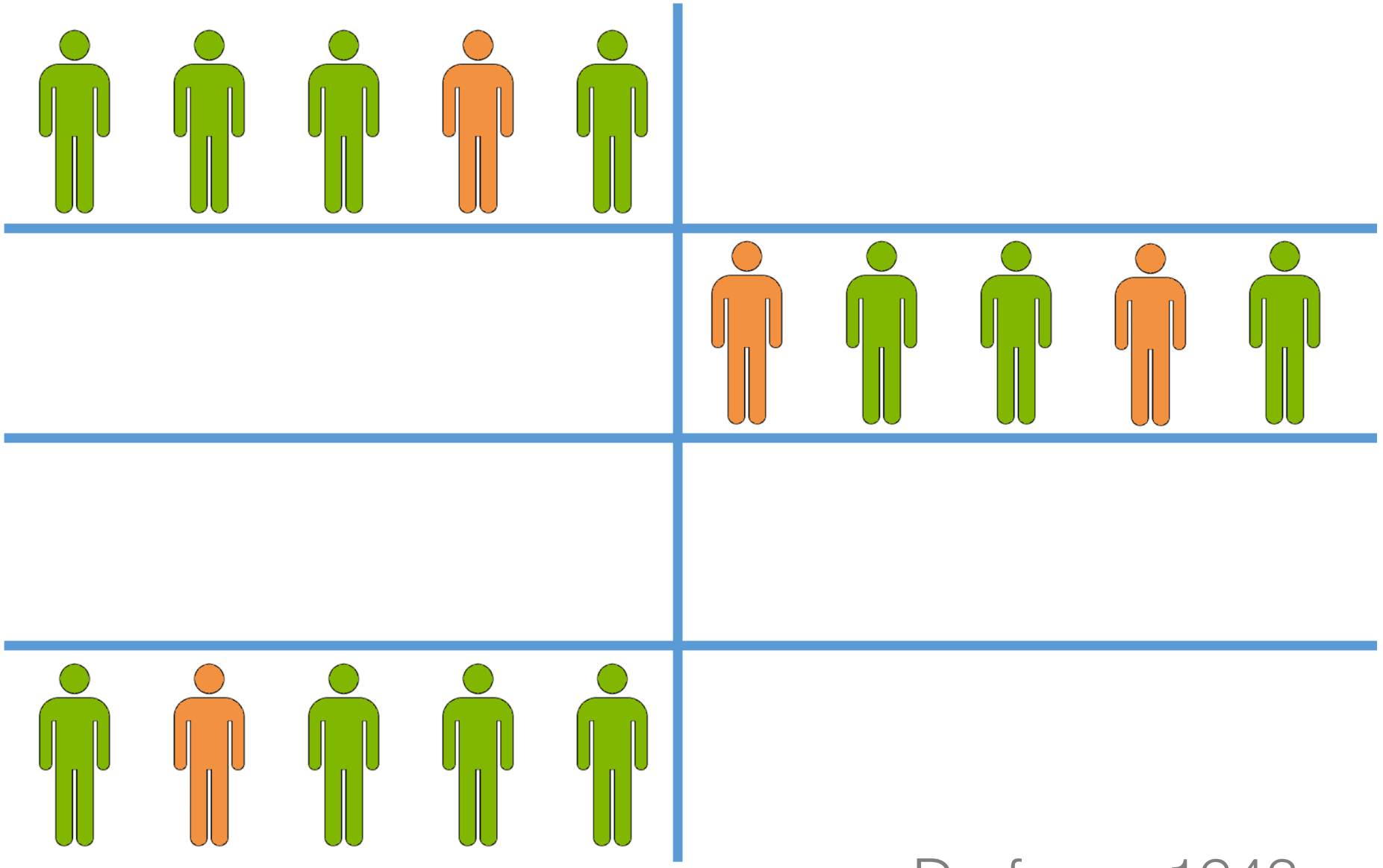
Group testing



Group testing



Group testing



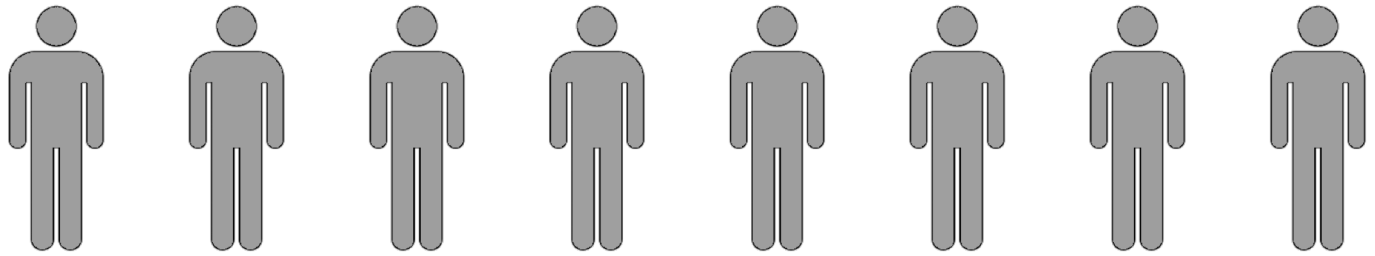
Dorfman, 1943

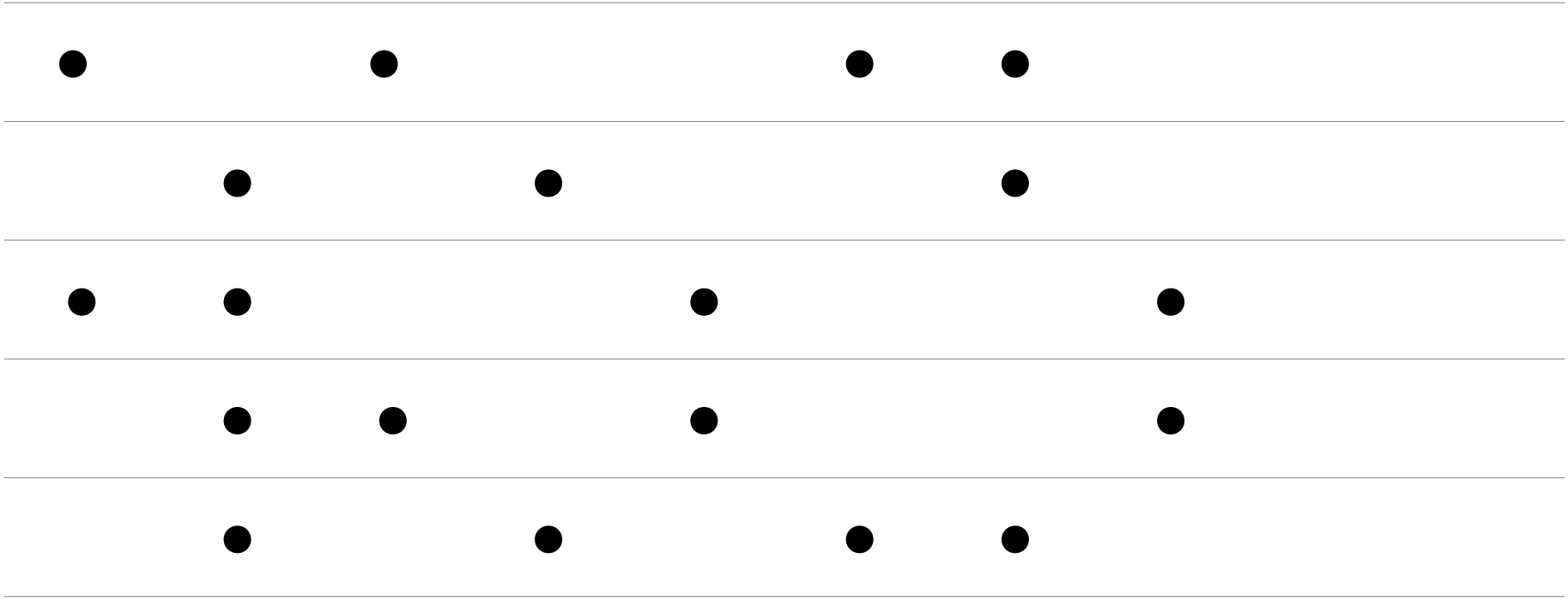
Types of problem

Adaptive
look at previous tests
before designing the next

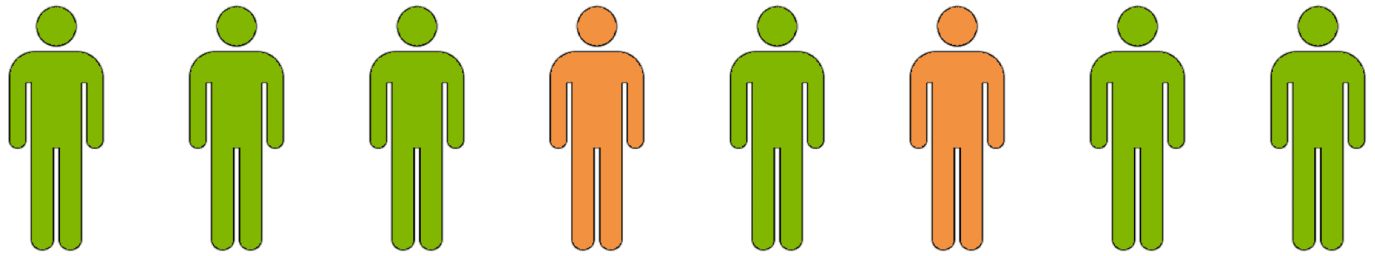
Nonadaptive
all tests designed
in advance

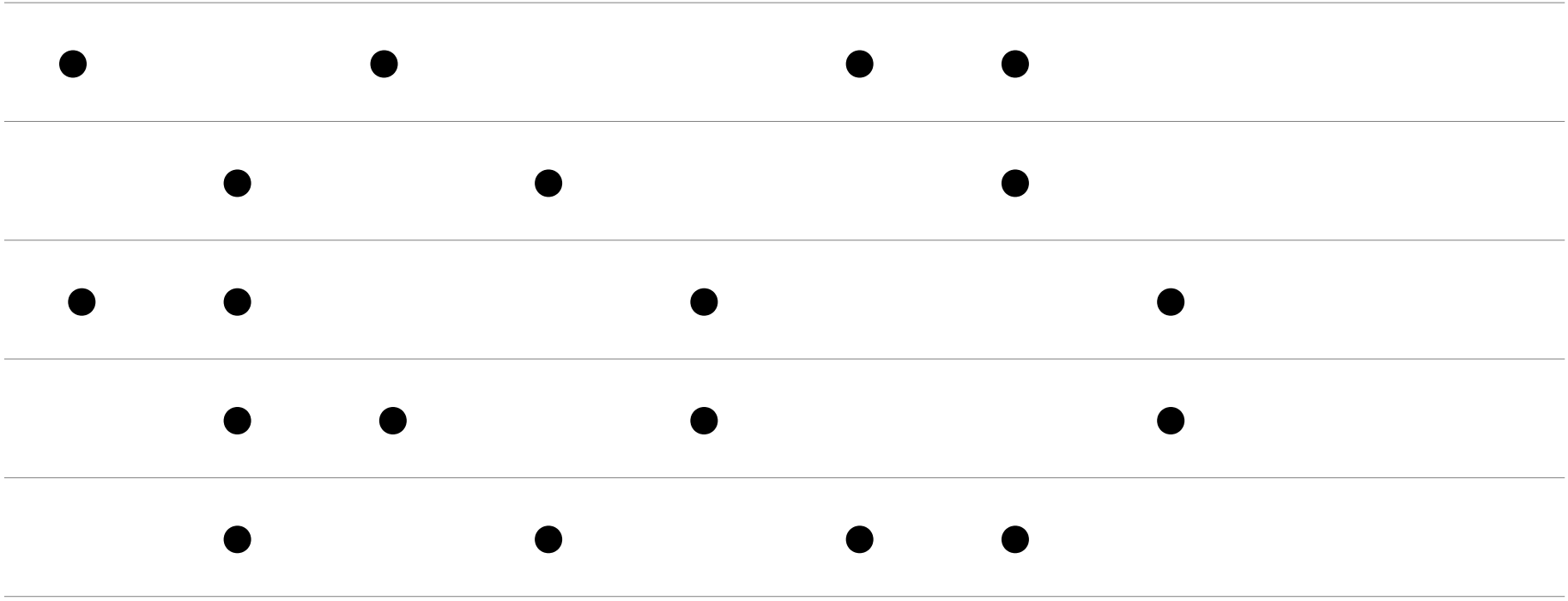
Group testing



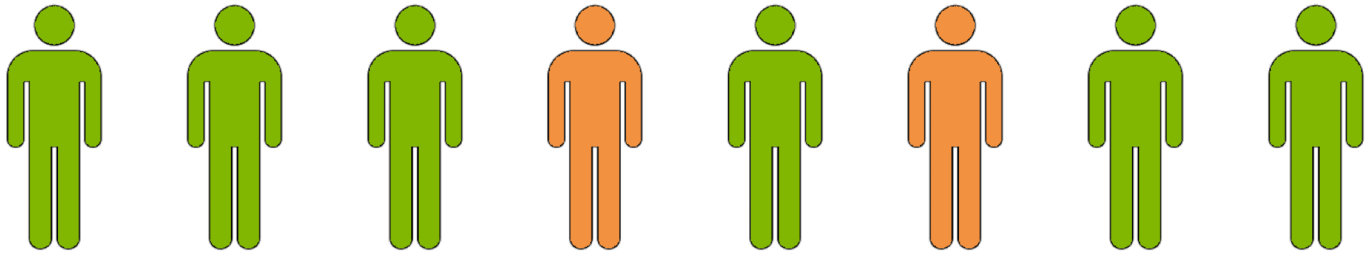


Group testing



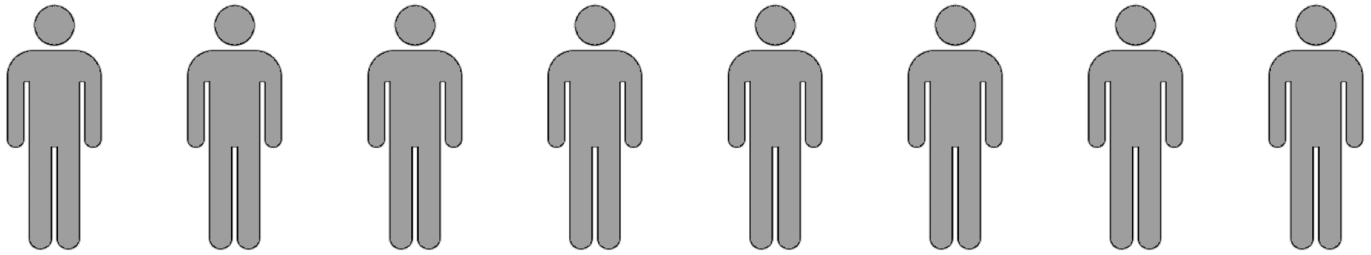


Group testing



● ● ● ●	pos
● ● ● ●	pos
● ● ● ●	neg
● ● ● ●	neg
● ● ● ●	pos

Group testing



•			•			•	•	pos
	•			•			•	pos
•	•				•			• neg
	•	•			•			• neg
	•			•		•	•	pos

Group testing

n items (soldiers)

k defective items (soldiers with syphilis)

T tests: *"Does this group of items contain* at least one *defective item?"* (blood tests)

Main problem

n items

k defective items

T tests

Given n and k,
how big does T have to be
to reliably work out
which items were defective?

Why should I care?

Why should I care?

Applications

Testing soldiers for syphilis
DNA screening
Management of wireless networks
Database management
Data compression
Cybersecurity
Graph learning
The counterfeit coin problem
...

Why should I care?

Applications

Concrete example of more general problems

Sparse inference, $p > n$ statistics

Nonlinear models

Search problems

Inverse problems

Why should I care?

Applications

Concrete example of more general problems

A fun problem in its own right

Probability
Statistics
Computer science
Information theory
Combinatorics

2

Counting bound and rate

Main problem

n items

k defective items

T tests

Given n and k,
how big does T have to be
to reliably work out
which items were defective?

Counting bound

$$T \geq \log_2 \binom{n}{k}$$

Counting bound

$$T \geq \log_2 \binom{n}{k}$$

There are $\binom{n}{k}$ sets of size k.

We need $\log_2 \binom{n}{k}$ bits of information to identify the defective set.

We learn at most **1** bit per test.

Rate

$$T \geq \log_2 \binom{n}{k}$$

So if we use

$c \log_2 \binom{n}{k}$ tests,

(and have error probability → 0)

we have learned at a **rate** of

$\frac{1}{c}$ bits per test

Rate

Rate ≤ 1

So if we use

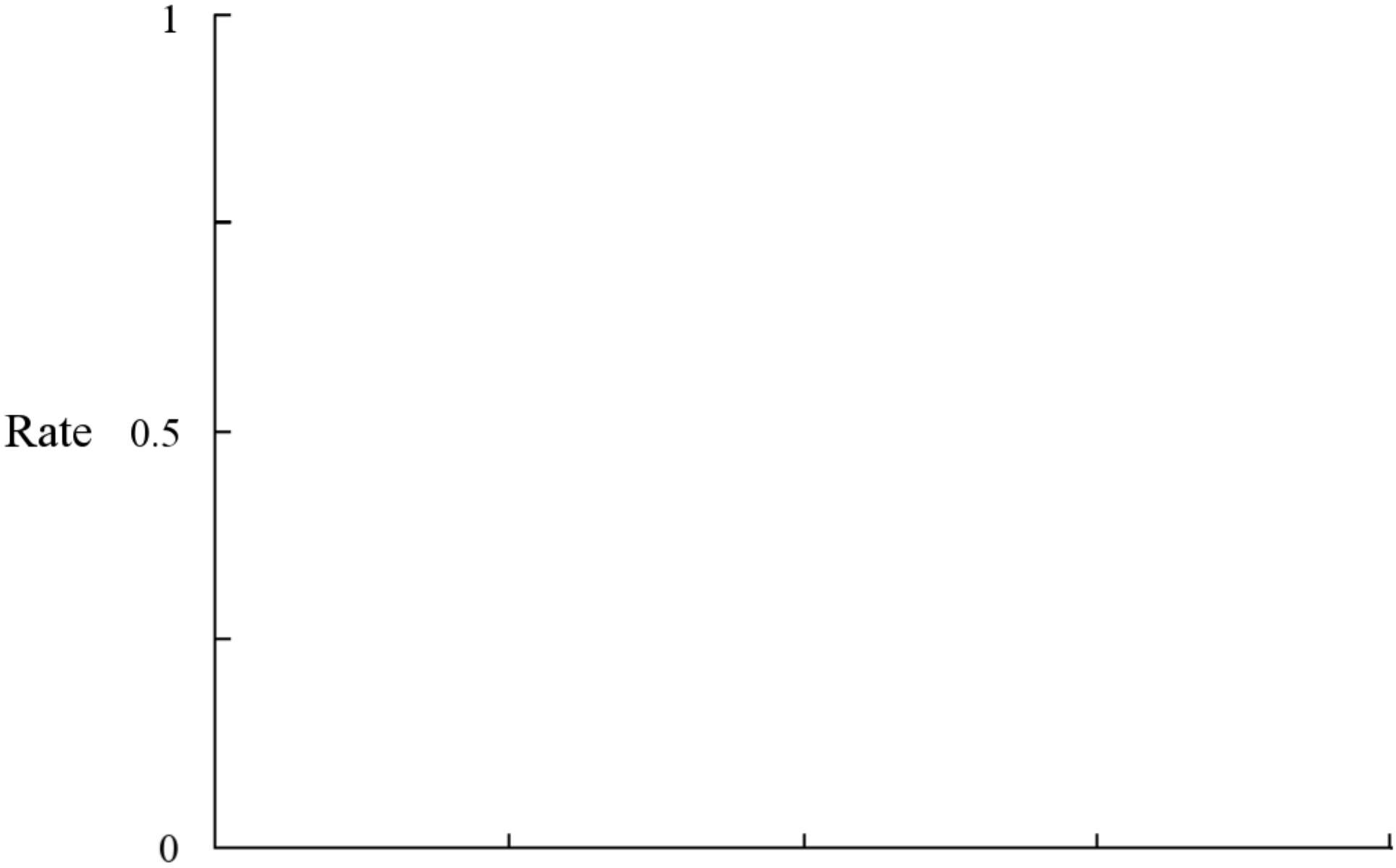
$c \log_2 \binom{n}{k}$ tests,

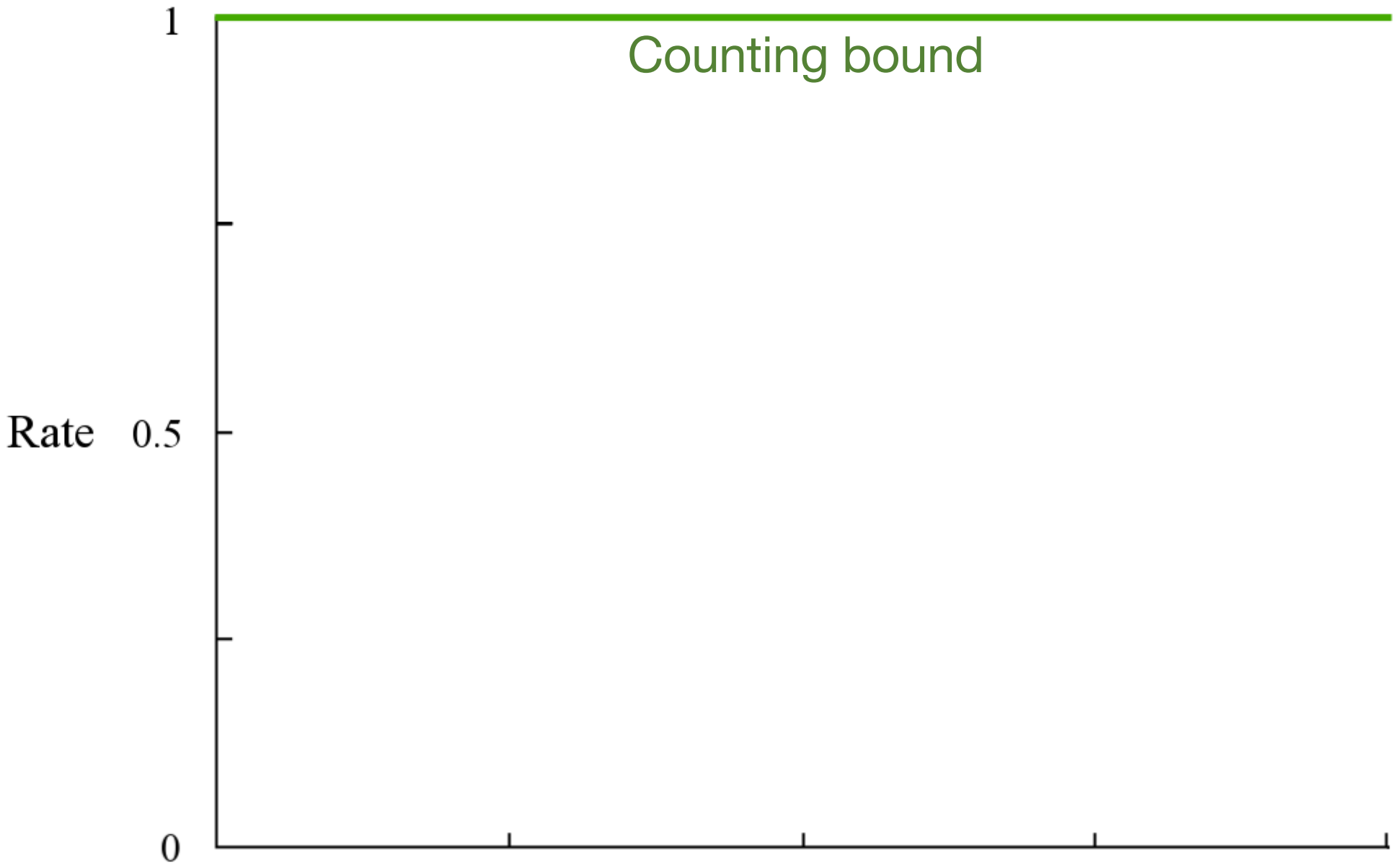
(and have error probability → 0)

we have learned at a **rate** of

$\frac{1}{c}$ bits per test

Higher rate = Fewer tests = Better


1
Rate
0.5
0


1
Counting bound
Rate
0.5
0

Defective set

The set of defective items, $\mathcal{K}$
is uniformly random among
all subsets of $\{1, 2, \ldots, n\}$ of size k

Sparsity

We assume defects are rare:

$$k = o(n)$$

Sparsity

We assume defects are rare

$$k = o(n)$$

Specifically,

$$k = \Theta(n^{\theta})$$

where $0 < \theta < 1$ is the

sparsity parameter

Sparsity

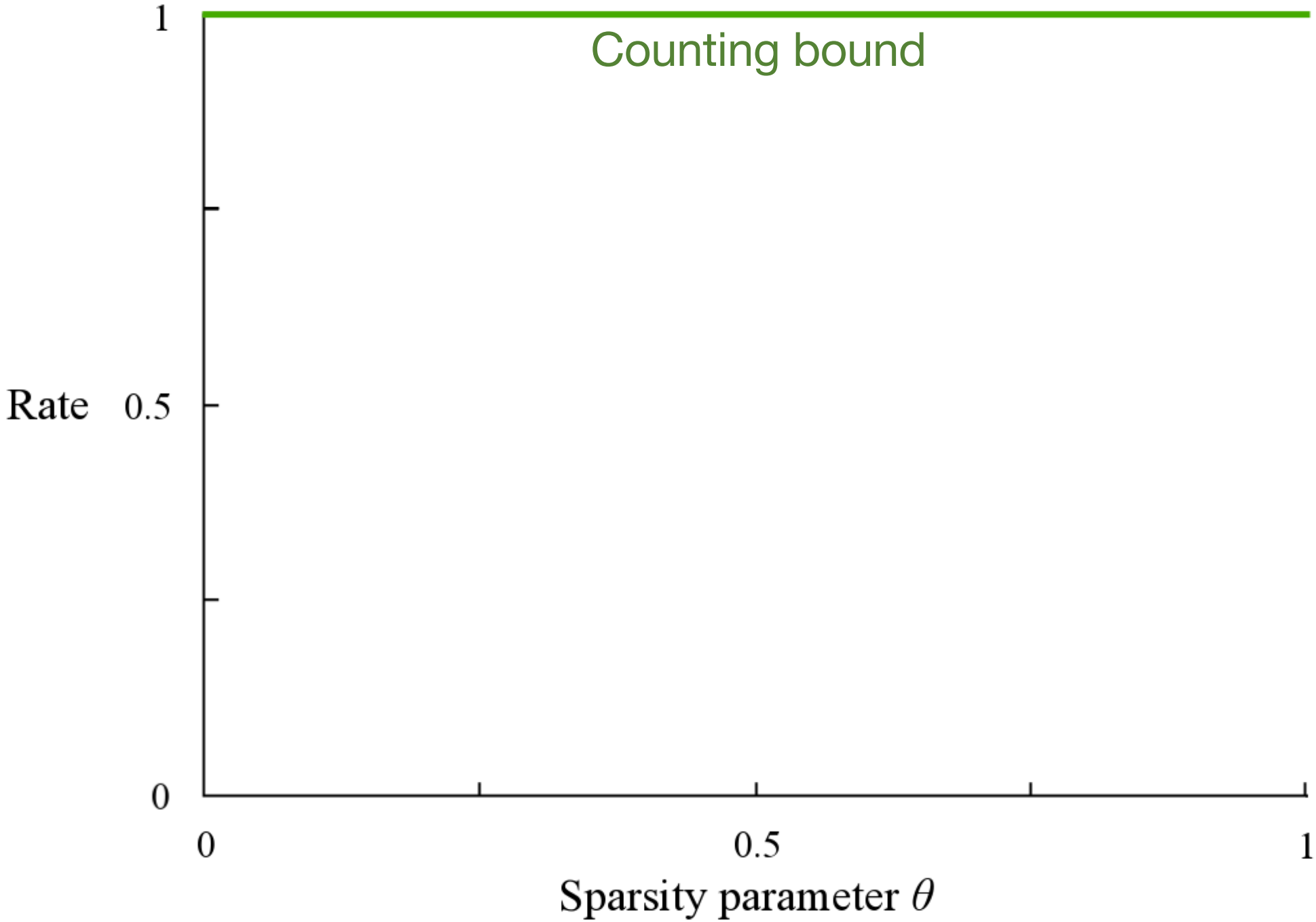
We assume defects are rare:

$k = \Theta(n^{\theta})$ where $0 < \theta < 1$

Useful fact:

In this regime,

$$\log_2 \binom{n}{k} \approx \log_2 \frac{n}{k} = (1 - \theta)\log_2 n$$


Counting bound
1
Rate
0.5
0
0
0.5
1
Sparsity parameter θ

3

Types of group testing

Types of problem

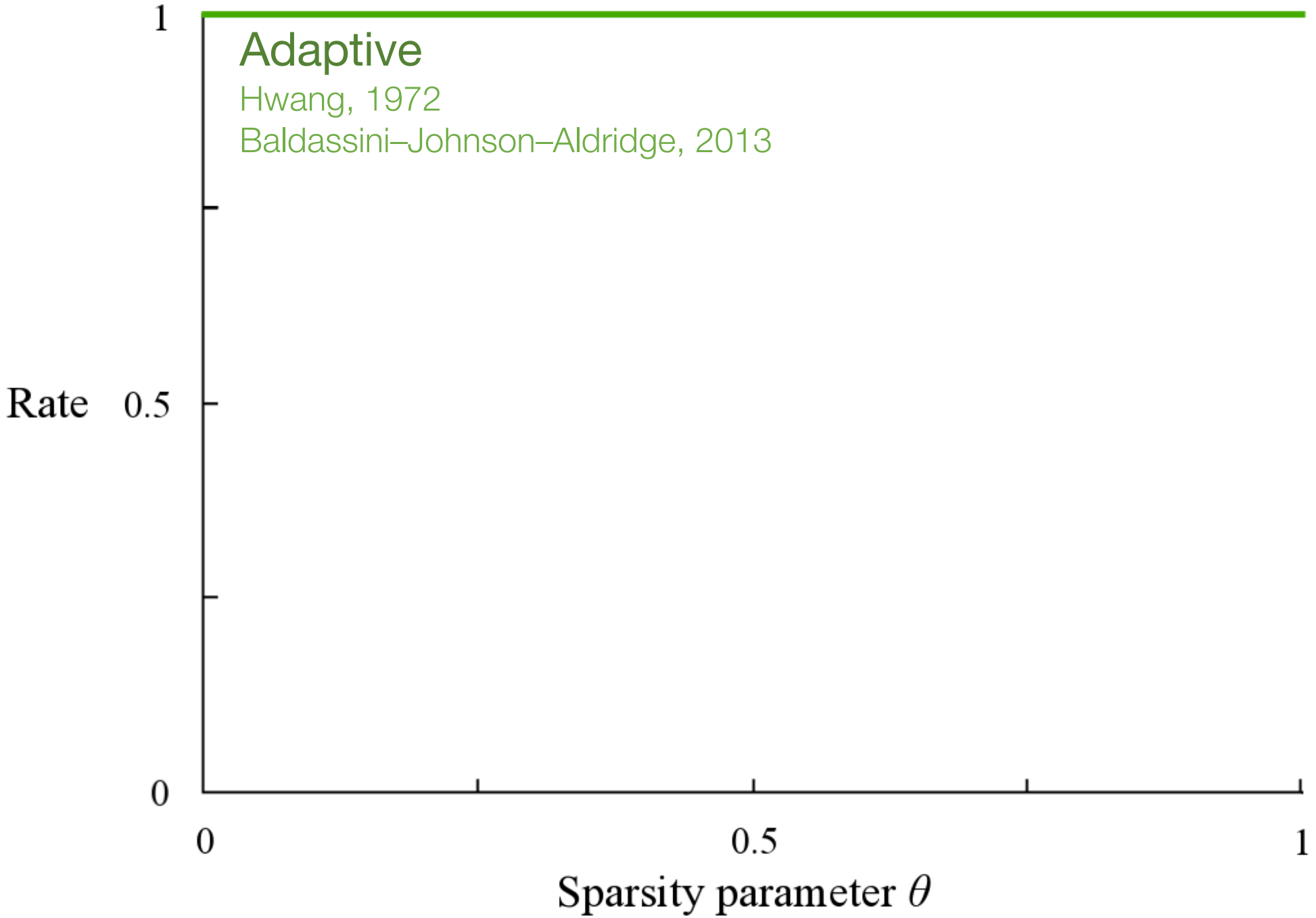
Adaptive
look at previous tests
before designing the next

Nonadaptive
all tests designed
in advance

Types of problem

Adaptive
look at previous tests
before designing the next

Nonadaptive
all tests designed
in advance


Adaptive
Hwang, 1972
Baldassini–Johnson–Aldridge, 2013
Rate
1
0.5
0
0
0.5
1
Sparsity parameter θ

Types of problem

Adaptive
look at previous tests
before designing the next

Nonadaptive
all tests designed
in advance

Zero-error
Error probability 0
(over random defective set)

Small error
Error probability $\longrightarrow 0$
as $n \longrightarrow \infty$

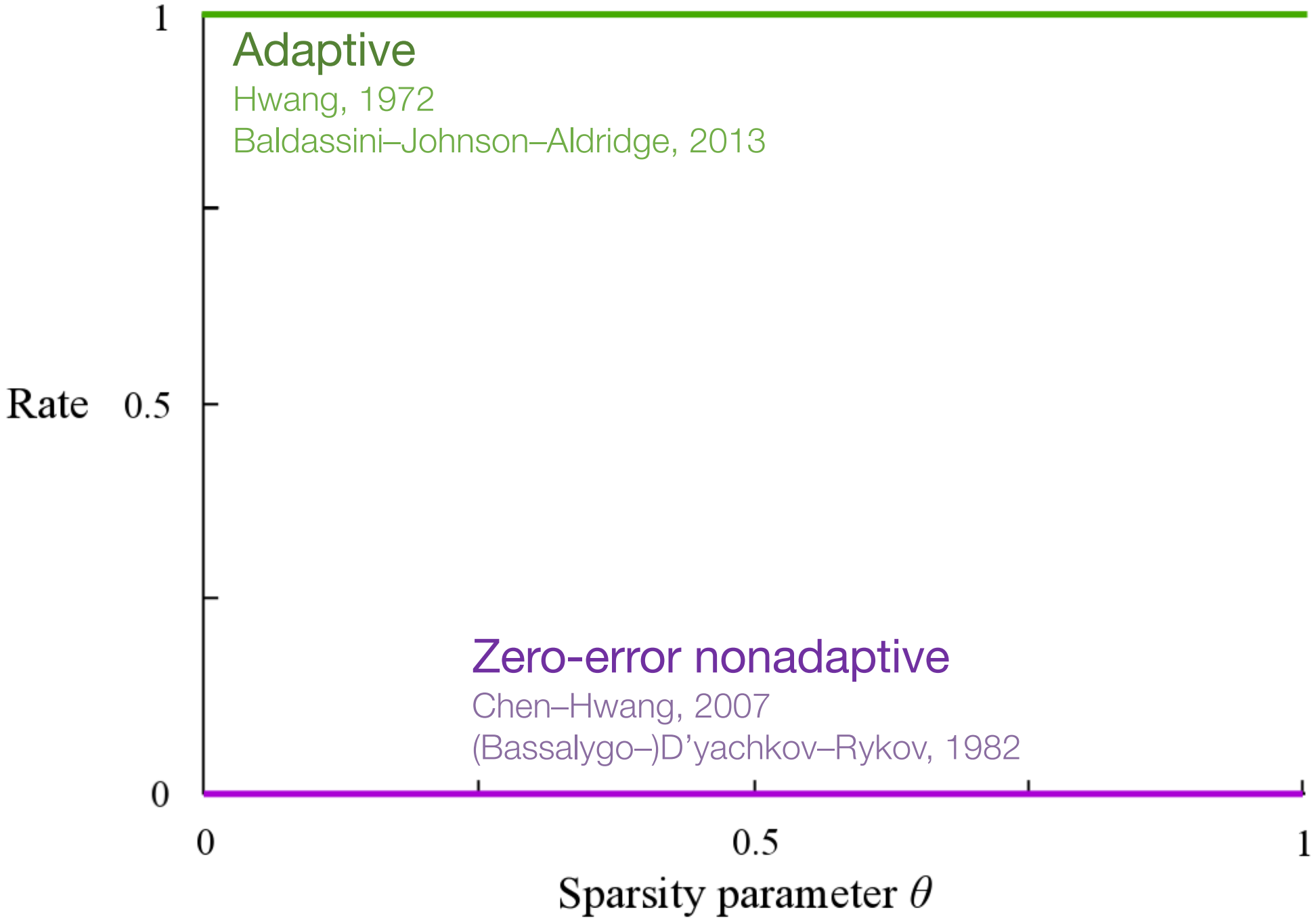
Types of problem

Adaptive
look at previous tests
before designing the next

Zero-error
Error probability 0
(over random defective set)

Nonadaptive
all tests designed
in advance

Small error
Error probability $\longrightarrow 0$
as $n \longrightarrow \infty$


1
Adaptive
Hwang, 1972
Baldassini–Johnson–Aldridge, 2013
Rate
0.5
Zero-error nonadaptive
Chen–Hwang, 2007
(Bassalygo–)D'yachkov–Rykov, 1982
0
0
0.5
1
Sparsity parameter θ

Types of problem

Adaptive
look at previous tests
before designing the next

Nonadaptive
all tests designed
in advance

Zero-error
Error probability 0
(over random defective set)

Small error
Error probability $\longrightarrow 0$
as $n \longrightarrow \infty$

Perfect reconstruction
Identify every defective item
and no others

Partial reconstruction
$|\mathcal{K}' \,\Delta\, \mathcal{K}| < \varepsilon k$

Types of problem

Adaptive
look at previous tests
before designing the next

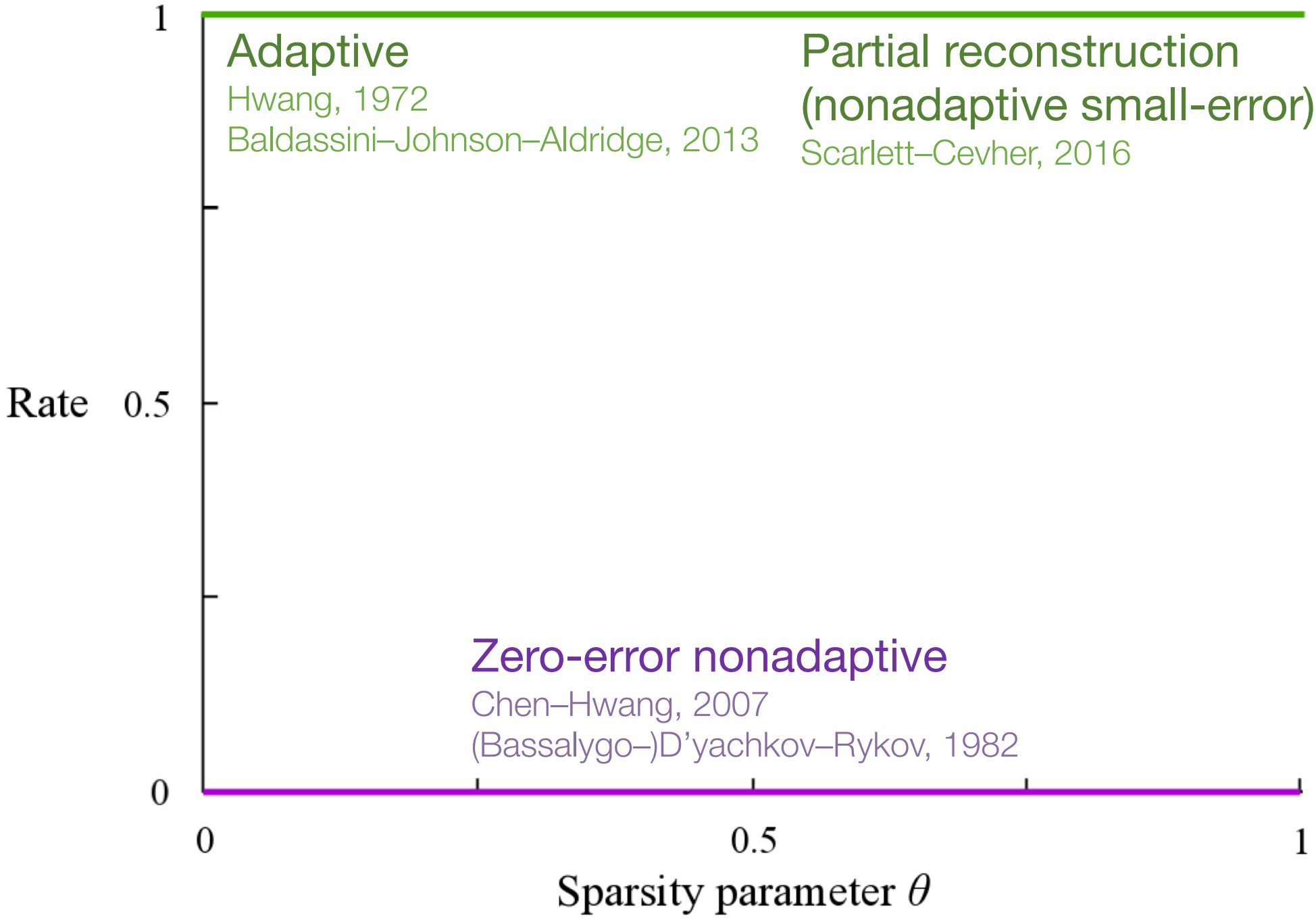
Nonadaptive
all tests designed
in advance

Zero-error
Error probability 0
(over random defective set)

Small error
Error probability $\longrightarrow 0$
as $n \longrightarrow \infty$

Perfect reconstruction
Identify every defective item
and no others

Partial reconstruction
$|\mathcal{K}' \,\Delta\, \mathcal{K}| < \varepsilon k$


Adaptive
Hwang, 1972
Baldassini–Johnson–Aldridge, 2013
Partial reconstruction
(nonadaptive small-error)
Scarlett–Cevher, 2016
Zero-error nonadaptive
Chen–Hwang, 2007
(Bassalygo–)D'yachkov–Rykov, 1982
Rate
1
0.5
0
0
0.5
1
Sparsity parameter θ

Types of problem

Adaptive
look at previous tests
before designing the next

Nonadaptive
all tests designed
in advance

Zero-error
Error probability 0
(over random defective set)

Small error
Error probability $\longrightarrow 0$
as $n \longrightarrow \infty$

Perfect reconstruction
Identify every defective item
and no others

Partial reconstruction
$|\mathcal{K}' \,\Delta\, \mathcal{K}| < \varepsilon k$

4

Bernoulli designs and three detection algorithms

Two subproblems

Design

How should we design
the testing pools?

Detection

Given the testing pools
and the test outcomes,
how should we guess
which items were defective?

Design

Try a random
Bernoulli design

For each item i and test t

item i is in test t with probability p
item i is not in test t with probability $1 - p$

independently for all i and t

Design

Here we use
Bernoulli designs

For each item i and test t
item i is in test t with probability p

Optimal order for all θ
Optimal constants for some θ
Most widely studied designs
Easiest to prove things about

Detection

We'll look at three detection algorithms:

COMP (combinatorial optimal matching pursuit)

DD (definite defectives)

ML (maximum likelihood)

Detection

We'll look at three detection algorithms:

COMP (combinatorial optimal matching pursuit)

Assume items are defective
unless we're sure they're nondefective

DD (definite defectives)

Assume items are nondefective
unless we're sure they're defective

ML (maximum likelihood)

Theoretically optimal but impractical for large problems

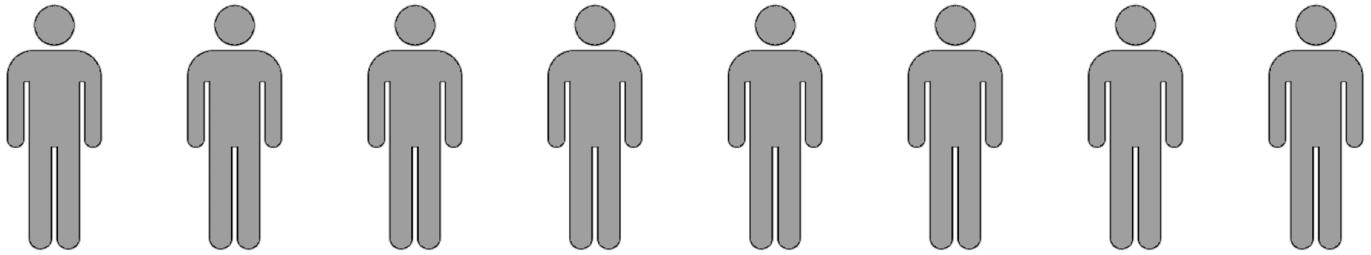
COMP

Any item in a negative test
is **definitely nondefective**

Assume everything else is defective

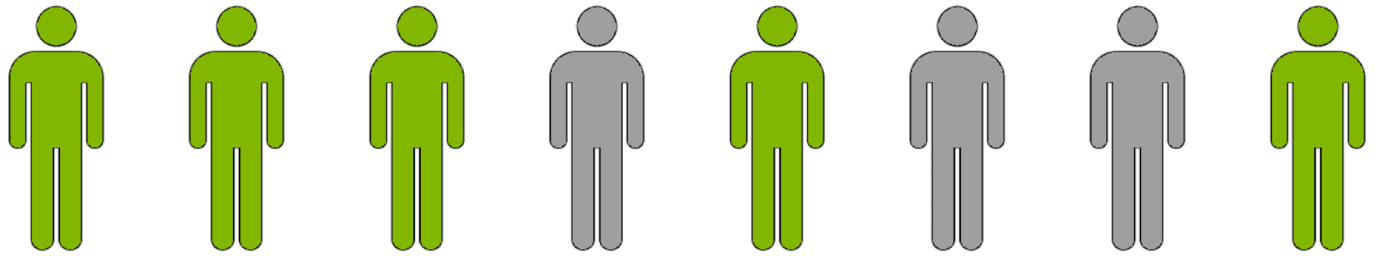
(Kautz–Singleton, 1964
Chan–Che–Jaggi–Saligrama, 2011)

COMP



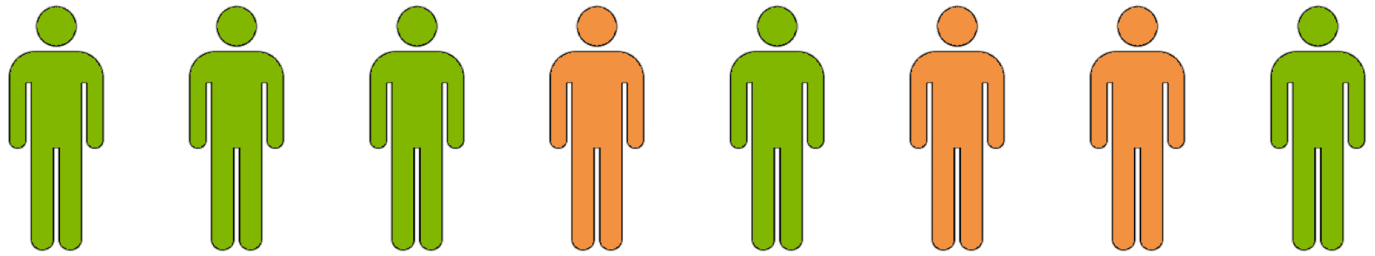
●			●			●	●	pos
	●			●			●	pos
●	●				●			● neg
	●	●			●			● neg
	●			●		●	●	pos

COMP



● ● ● ●	pos
● ● ●	pos
● ● ● ●	neg
● ● ● ●	neg
● ● ● ●	pos

COMP



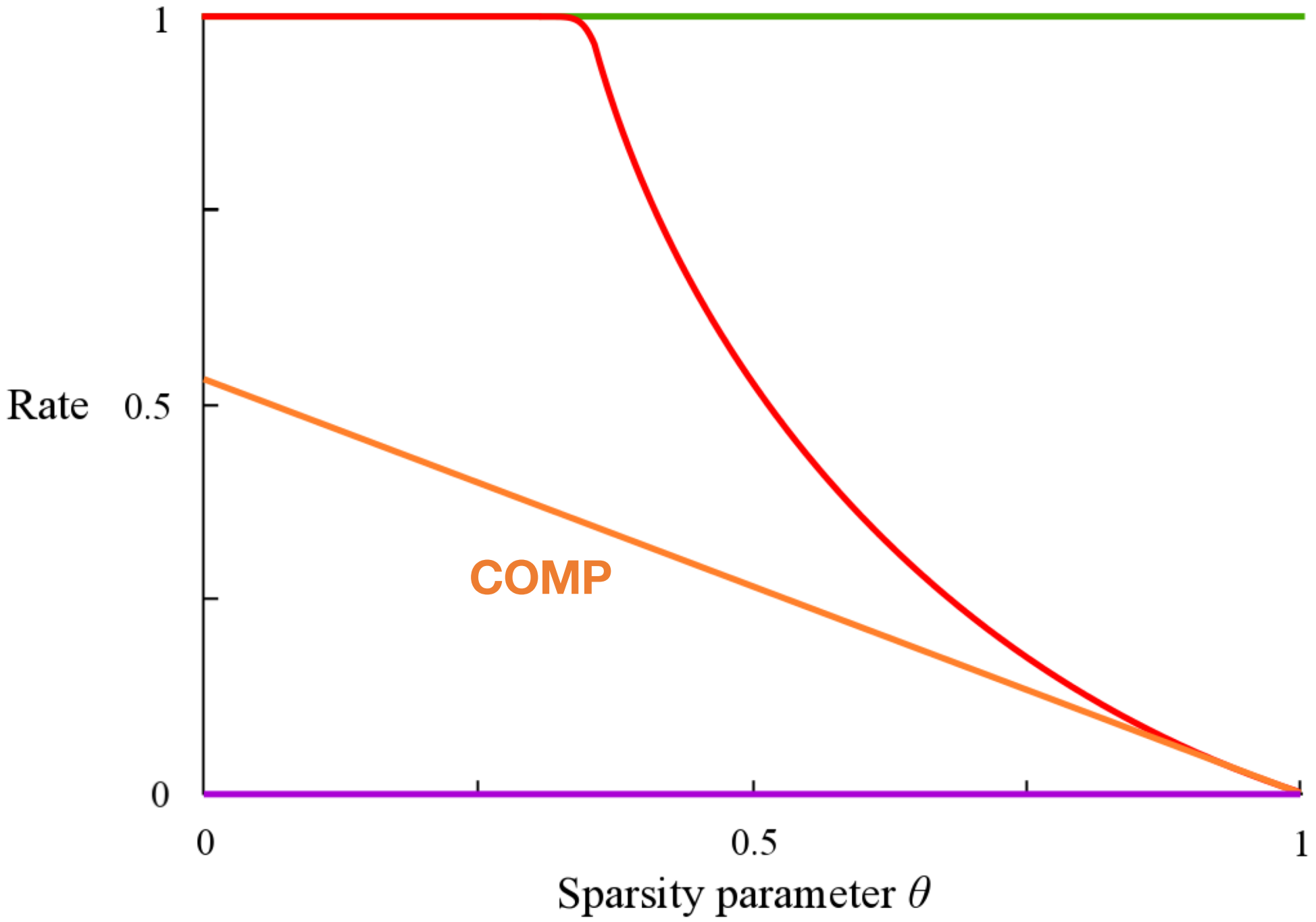
● ● ● ●	pos
● ● ●	pos
● ● ● ●	neg
● ● ● ●	neg
● ● ● ●	pos

COMP

COMP achieves the rate

$$R = \frac{1}{\mathrm{e} \ln 2}(1-\theta) = 0.53(1-\theta)$$

(Chan–Che–Jaggi–Saligrama, 2011)
(Aldridge, 2017)


1
Rate
0.5
0
COMP
0
0.5
1
Sparsity parameter θ

COMP

COMP achieves the rate

$$R = \frac{1}{\mathrm{e} \ln 2}(1-\theta)$$

Proof

The expected number of nondefectives found in a test is

$$(n-k)p(1-p)^k.$$

This is maximised at $p = \frac{1}{k+1} \approx \frac{1}{k}$. So take $p = \frac{1}{k}$.

COMP

Proof

The probability a test is negative is

$$(1-p)^k = \left(1-\frac{1}{k}\right)^k = \mathrm{e}^{-1},$$

so the the number of negative tests is very close to $\mathrm{e}^{-1}T$.

The number of nondefective items in each such test is very close to

$$np = \frac{n}{k}.$$

Overall, we "collect" about $\mathrm{e}^{-1}T\,\frac{n}{k}$ nondefective items.

COMP

Overall, we “collect” about $\mathrm{e}^{-1}T\frac{n}{k}$ nondefective items.

We need to rule out all $n-k \approx n$ nondefective items.

By the coupon collector problem, this requires

$$e^{-1}T\frac{n}{k} = n\ln n$$

and therefore

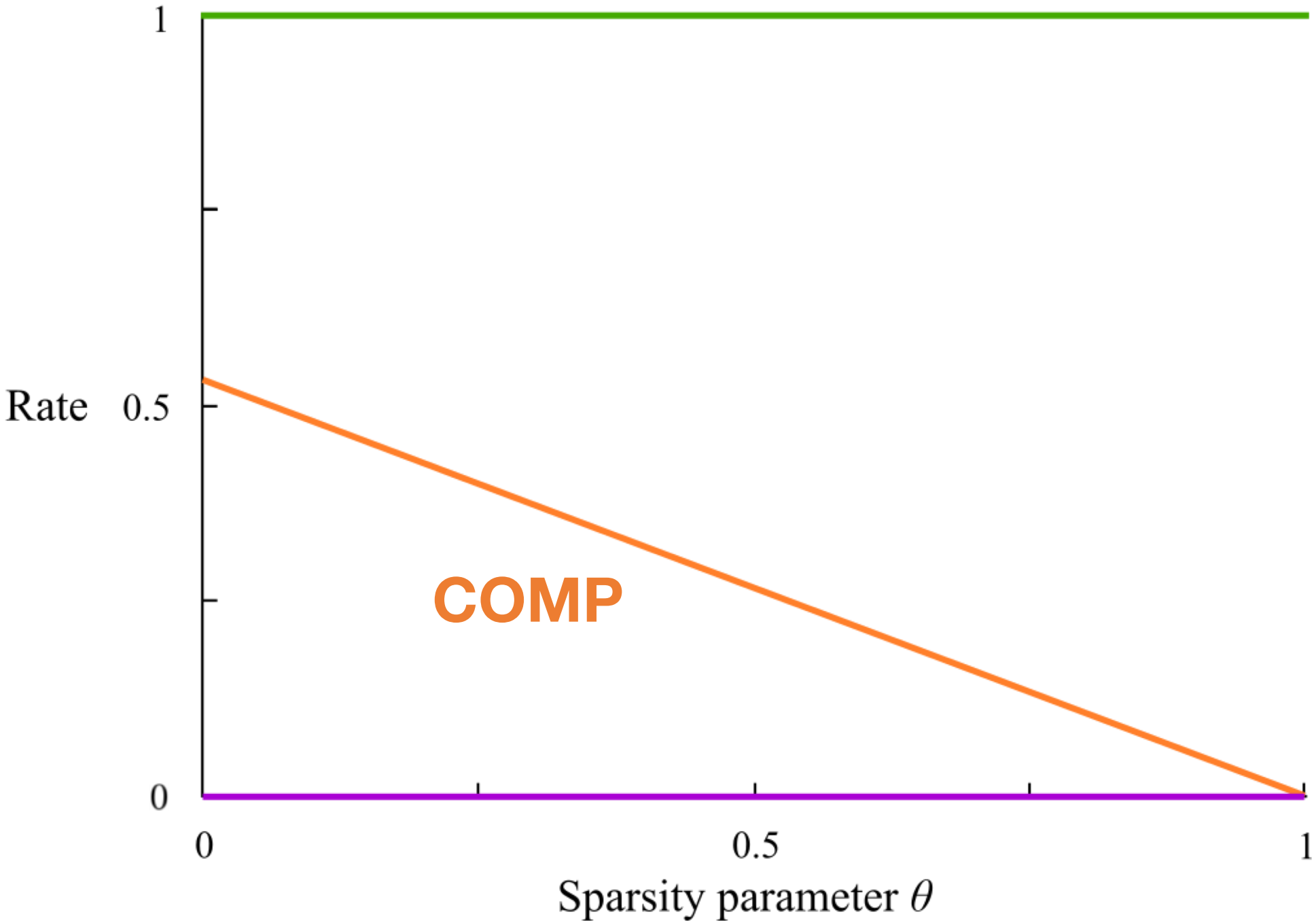
$$T = \mathrm{e}k\ln n = (\mathrm{e}\ln 2)\,k\log_2 n$$

COMP

COMP achieves the rate

$$R = \frac{1}{\mathrm{e} \ln 2}(1 - \theta)$$

$$T = \mathrm{e}k \ln n = (\mathrm{e} \ln 2)\, k \log_2 n$$


1
Rate
0.5
0
COMP
0
0.5
1
Sparsity parameter θ

DD

Any item in a negative test
is definitely nondefective

(Aldridge–Baldassini–Johnson, 2014)

DD

Any item in a negative test
is definitely nondefective

Call everything else “possibly defective”

(Aldridge–Baldassini–Johnson, 2014)

DD

Any item in a negative test
is definitely nondefective

Call everything else “possibly defective”

If a positive test has only one possible defective,
then that item is **“definitely defective”**

(Aldridge–Baldassini–Johnson, 2014)

DD

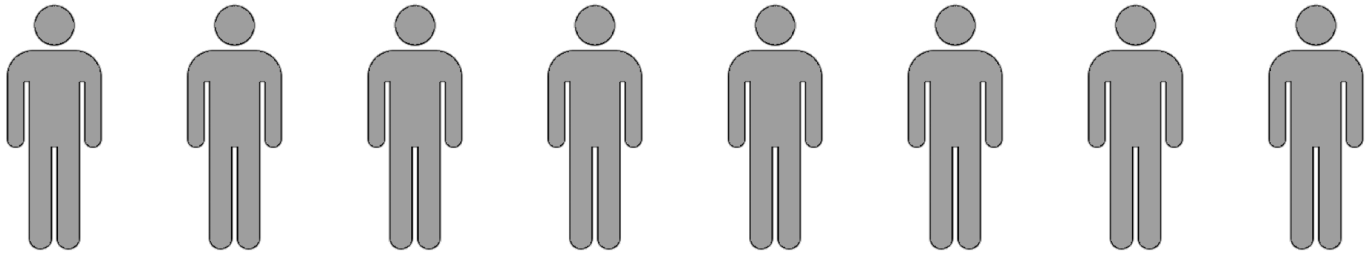
Any item in a negative test
is definitely nondefective

Call everything else “possibly defective”

If a positive test has only one possible defective,
then that item is “definitely defective”

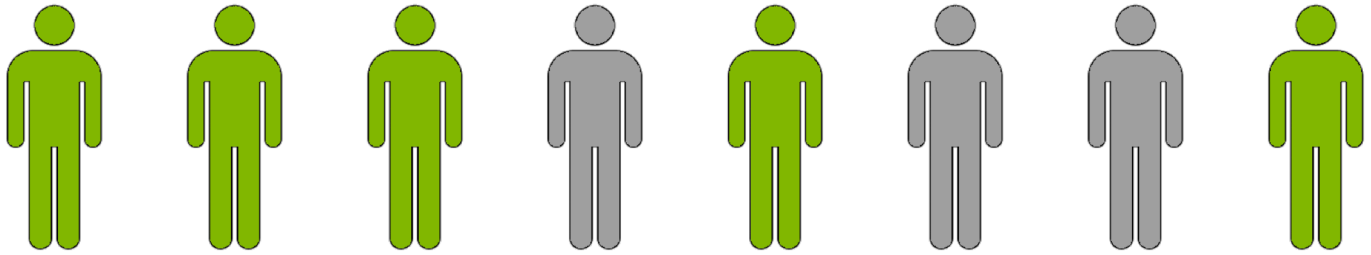
Assume everything else is nondefective

Group testing



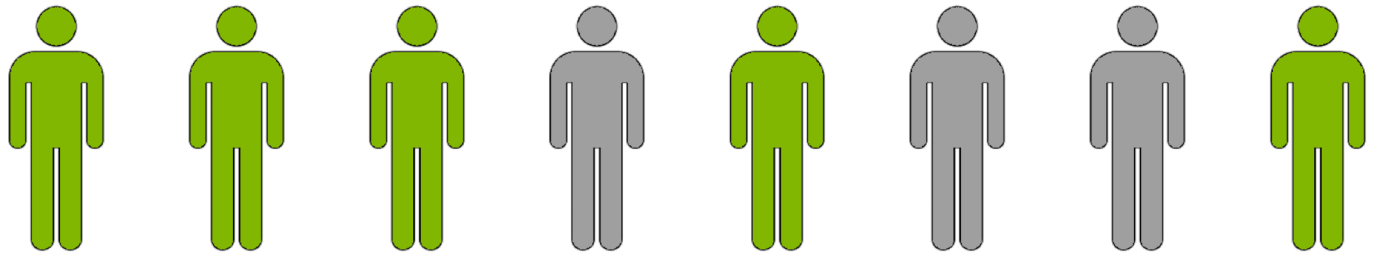
●			●			●	●	pos
	●						●	pos
●	●				●			●	neg
	●		●			●		●	neg
	●			●		●			pos

Group testing



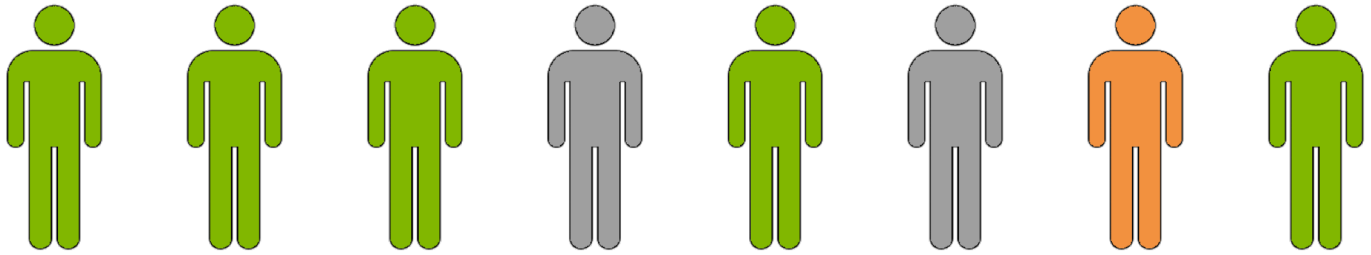
● ● ● ●	pos
● ● ●	pos
● ● ● ●	neg
● ● ● ●	neg
● ● ● ●	pos

Group testing



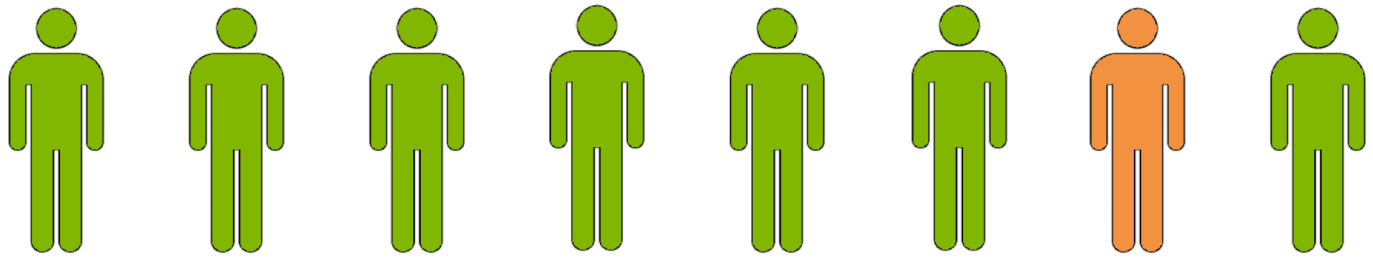
● ● ● ●	pos
● ●	pos
● ● ● ●	neg
● ● ● ●	neg
● ● ●	pos

Group testing



● ● ● ●	pos
● ● ●	pos
● ● ● ●	neg
● ● ● ●	neg
● ● ●	pos

Group testing



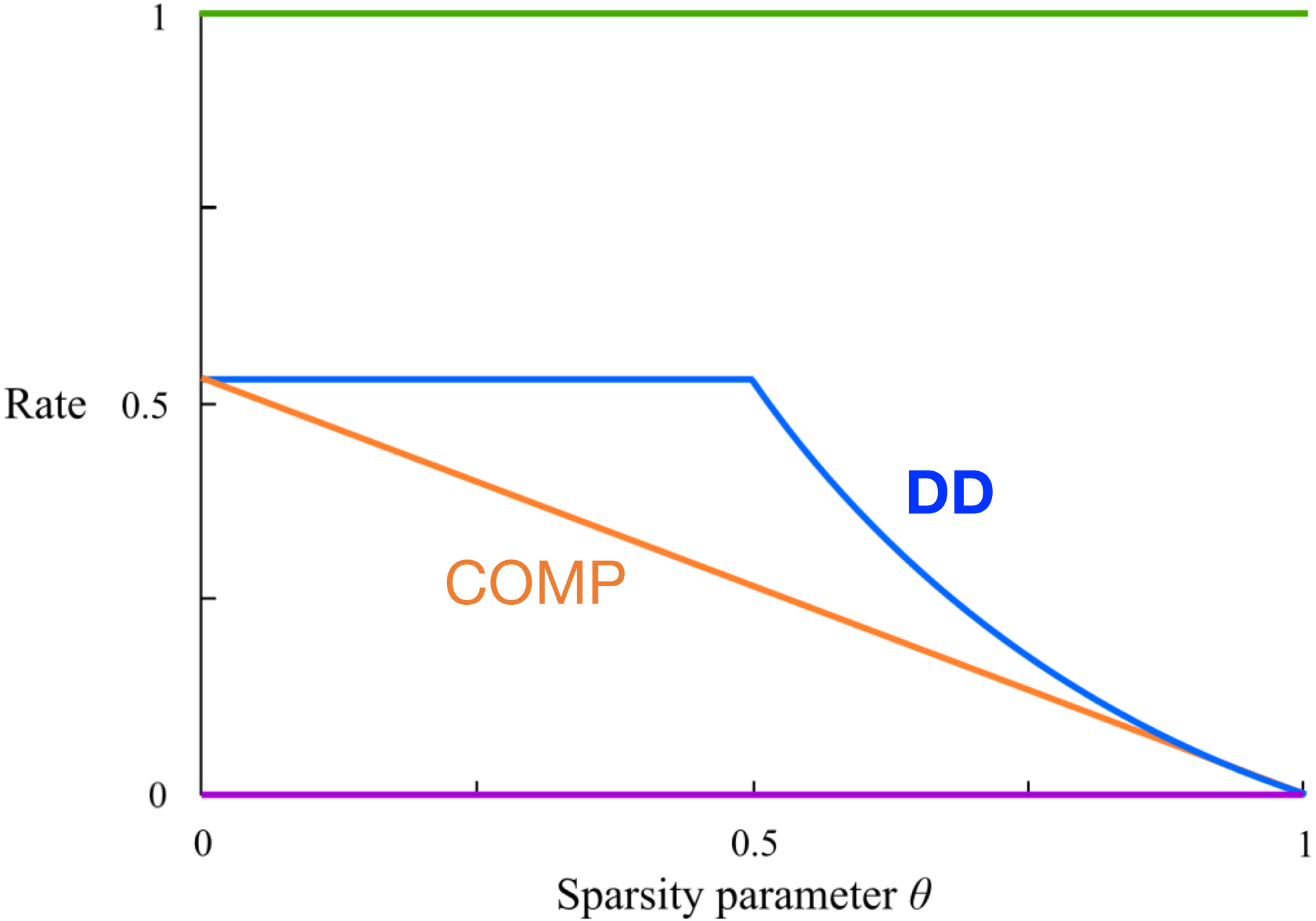
● ● ● ●	pos
● ●	pos
● ● ● ●	neg
● ● ● ●	neg
● ● ●	pos

DD

DD achieves the rate

$$R = \frac{1}{\mathrm{e} \ln 2} \min\left\{1, \frac{1-\theta}{\theta}\right\} = 0.53 \min\left\{1, \frac{1-\theta}{\theta}\right\}$$

(Aldridge–Baldassini–Johnson, 2014
Scarlett–Johnson, 2018)


1
Rate
0.5
0
DD
COMP
0
0.5
1
Sparsity parameter θ

DD

DD achieves the rate

$$R^* \geq \frac{1}{\mathrm{e} \ln 2} \min\left\{1, \frac{1-\theta}{\theta}\right\}$$

Proof:

By conditioning on the number of positive tests,
write down the probability DD succeeds
as a complicated triple summation.

Bound this expression in many pages of unpleasant work.

Maximum likelihood

(Scarlett–Cevher, 2016; Aldridge 2018)

The maximum likelihood detector with a Bernoulli design achieves rate

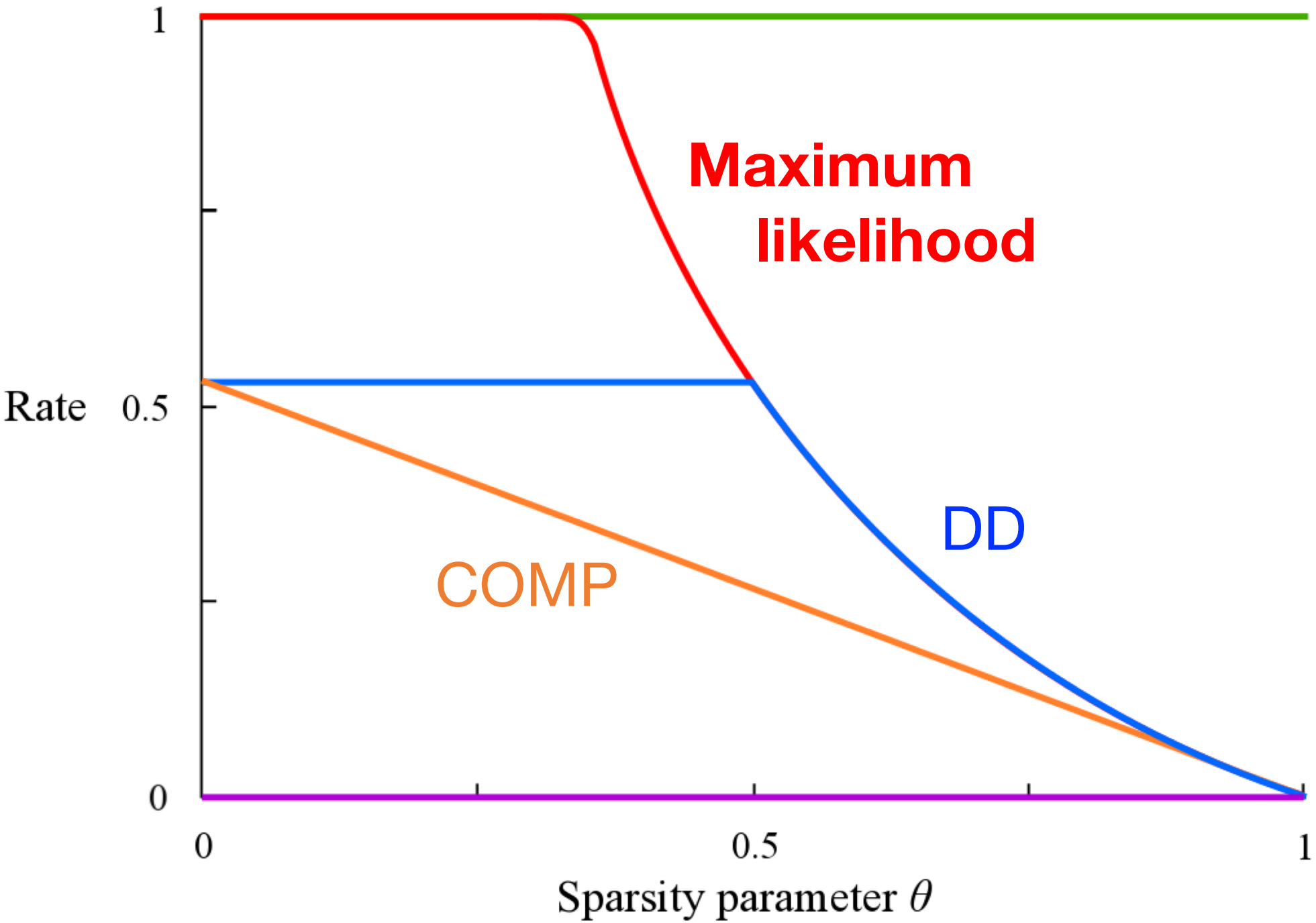
$$R = \max_{\nu>0} \min\left\{h(\mathrm{e}^{-\nu}),\ \frac{\nu}{\mathrm{e}^{\nu}\ln 2}\frac{1-\theta}{\theta}\right\}$$

Maximum likelihood

(Scarlett–Cevher, 2016; Aldridge 2018)

The maximum likelihood detector with a Bernoulli design achieves rate

$$C \leq \begin{cases} 1 & \theta \leq 1/3 \\ \text{it's complicated} & \text{in between} \\ 0.53\,\dfrac{1-\theta}{\theta} & \theta > 0.36 \end{cases}$$


1
Rate
0.5
0
Maximum
likelihood
DD
COMP
0
0.5
1
Sparsity parameter θ

Maximum likelihood

(Scarlett–Cevher, 2016; Aldridge 2018)

The maximum likelihood detector with a Bernoulli design achieves rate

$$C = \max_{\nu>0} \min \left\{ h(\mathrm{e}^{-\nu}) \ , \ \frac{\nu}{\mathrm{e}^{\nu} \ln 2} \frac{1-\theta}{\theta} \right\}$$

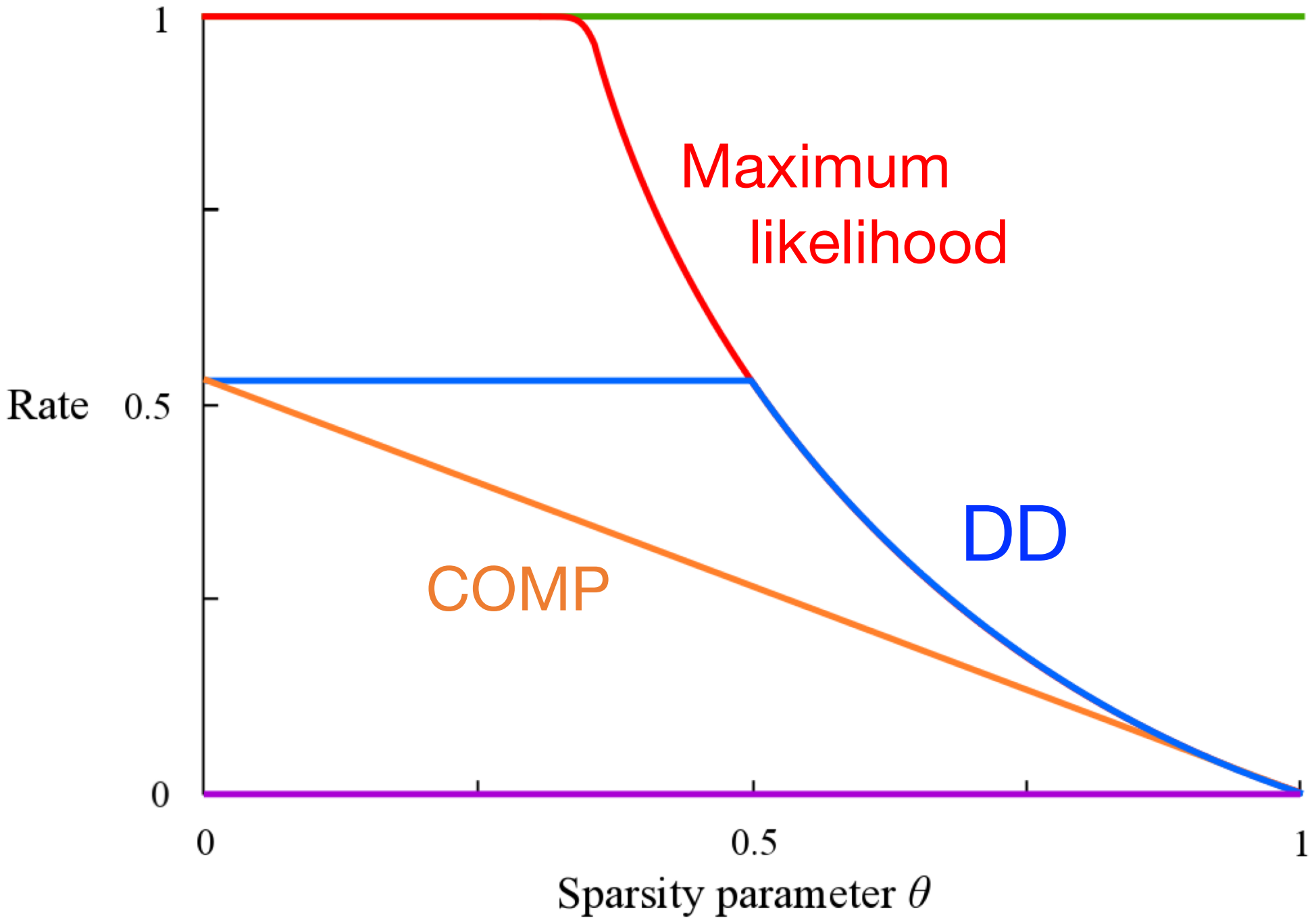
Proof: Information theory methods, like Shannon's channel coding theorem, but with "correlated messages"

Maximum likelihood

The maximum likelihood detector
is **impractical** because:

✗ It requires exact knowledge of k

✗ It requires solving an NP hard problem


1
Rate
0.5
0
Maximum
likelihood
DD
COMP
0
0.5
1
Sparsity parameter θ

5 decoding algorithms

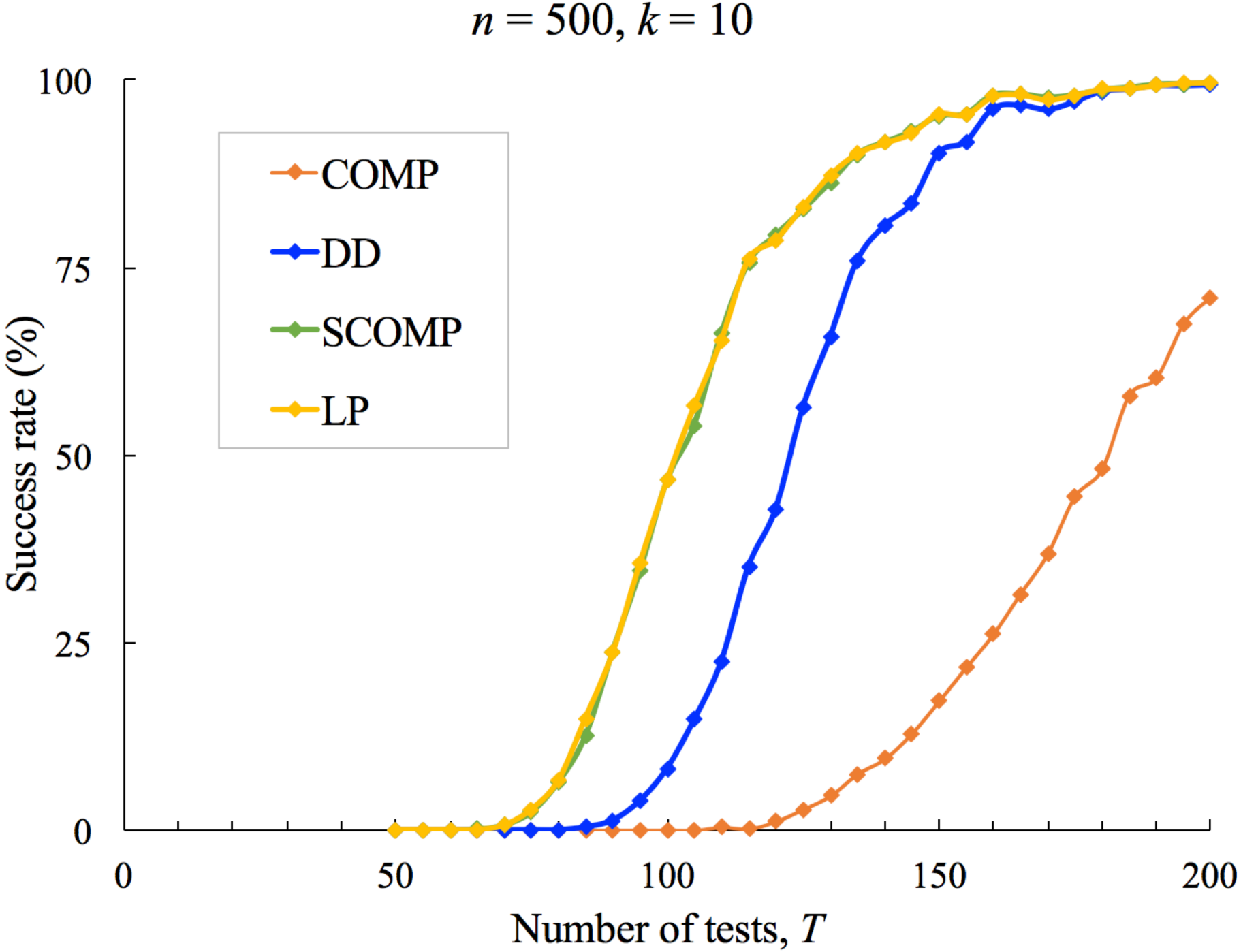
Maximum Likelihood

COMP: Only say an item is nondefective
if you're sure it's nondefective

DD: Only say an item is defective
if you're sure it's defective

SCOMP: Greedily declare items defective
until you get a satisfying set
(Aldridge–Baldassini–Johnson, 2014)

LP: Approximate ML using linear programming
(Malioutov–Malyutov, 2014; Aldridge, 2017)


$n = 500, k = 10$
100
75
50
25
0
Success rate (%)
COMP
DD
SCOMP
LP
0
50
100
150
200
Number of tests, T

5

Beyond Bernoulli designs

Better designs

Can we do better if we pick a design other than the Bernoulli random design?

Better designs

Can we do better if we pick a design
other than the Bernoulli random design?

Idea: Keep the design random
but add some more structure

Better designs

Can we do better if we pick a design
other than the Bernoulli random design?

Idea: Keep the design random
but add some more structure

Constant tests-per-item design:
Each item is placed in exactly L tests,
with the L tests chosen uniformly at random
independently from all other items.

Better designs

Why might the **constant tests-per-item design** be better?

Better designs

Why might the **constant tests-per-item design** be better?

We're introducing anti-correlation into tests,
to try and make sure we get "new" bits of information.

Better designs

Why might the **constant tests-per-item design** be better?

We're introducing anti-correlation into tests,
to try and make sure we get "new" bits of information.

The chance of a nondefective item
being in no negative tests is smaller.

With a **Bernoulli design** and $p = (\ln 2)/k$,
a nondefective item is in X negative tests, where

$$X \sim \text{Bin}\left(\frac{T}{2}, \frac{\ln 2}{k}\right)$$

With a **Bernoulli design** and $p = (\ln 2)/k$,
a nondefective item is in X negative tests, where

$$X \sim \text{Bin}\left(\frac{T}{2}, \frac{\ln 2}{k}\right)$$

With **constant tests-per-item** and $L = (\ln 2)T/k$,
a nondefective item is in Y negative tests, where

$$Y \sim \text{Bin}\left(\ln 2 \frac{T}{k}, \frac{1}{2}\right)$$

With a **Bernoulli design** and $p = (\ln 2)/k$,
a nondefective item is in X negative tests, where

$$X \sim \text{Bin}\left(\frac{T}{2}, \frac{\ln 2}{k}\right)$$

$$\mathbb{E}X = \frac{\ln 2}{2}\frac{T}{k}, \qquad \text{Var}(X) \approx \frac{\ln 2}{2}\frac{T}{k}$$

With **constant tests-per-item** and $L = (\ln 2)T/k$,
a nondefective item is in Y negative tests, where

$$Y \sim \text{Bin}\left(\ln 2 \frac{T}{k}, \frac{1}{2}\right)$$

$$\mathbb{E}Y = \frac{\ln 2}{2}\frac{T}{k}, \qquad \text{Var}(Y) \approx \frac{\ln 2}{4}\frac{T}{k}$$

Better designs

Why might the **constant tests-per-item design** be better?

We're introducing anti-correlation into tests,
to try and make sure we get "new" bits of information.

The chance of a nondefective item
being in no negative tests is smaller.

Better designs

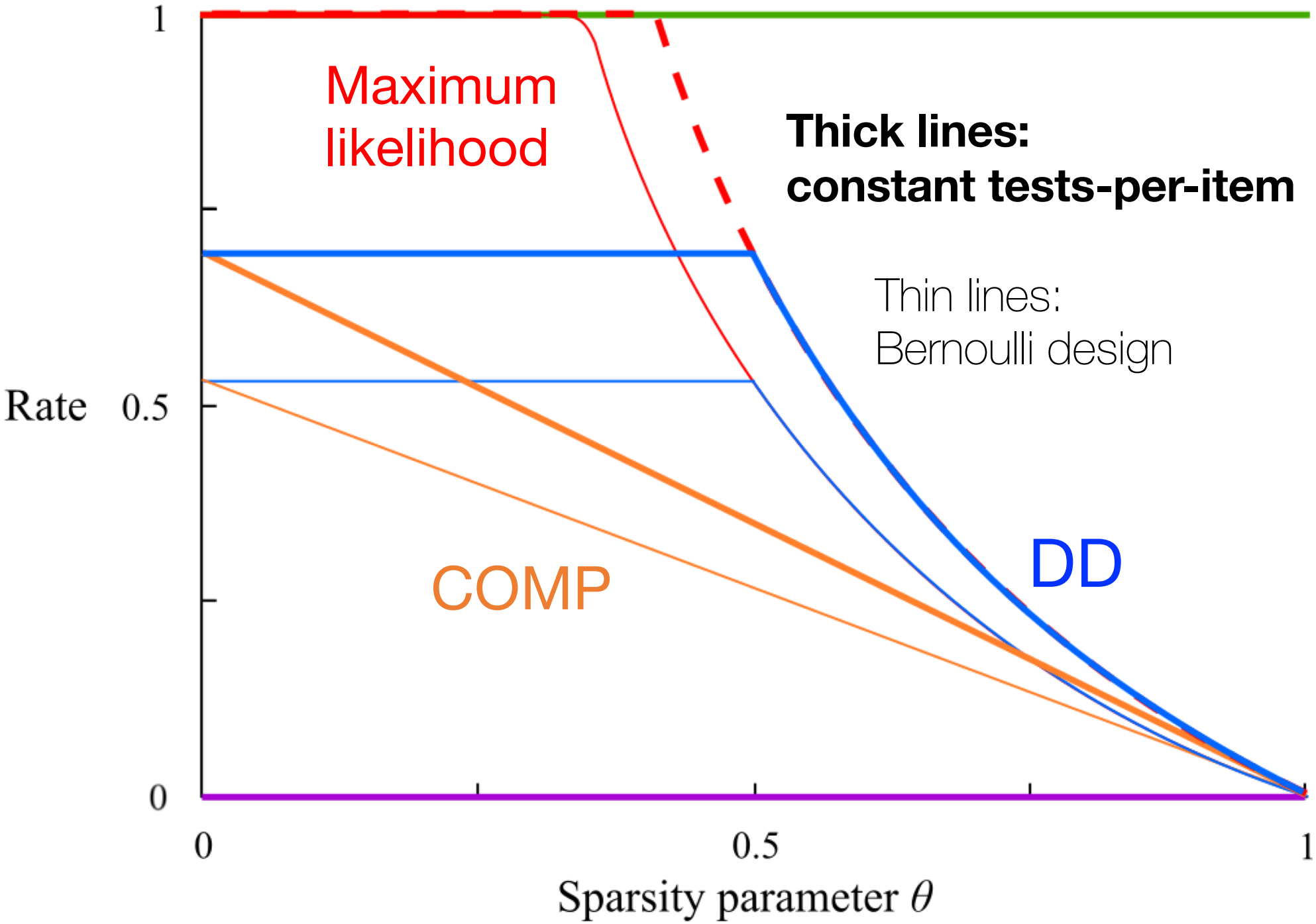
Why might the **constant tests-per-item design** be better?

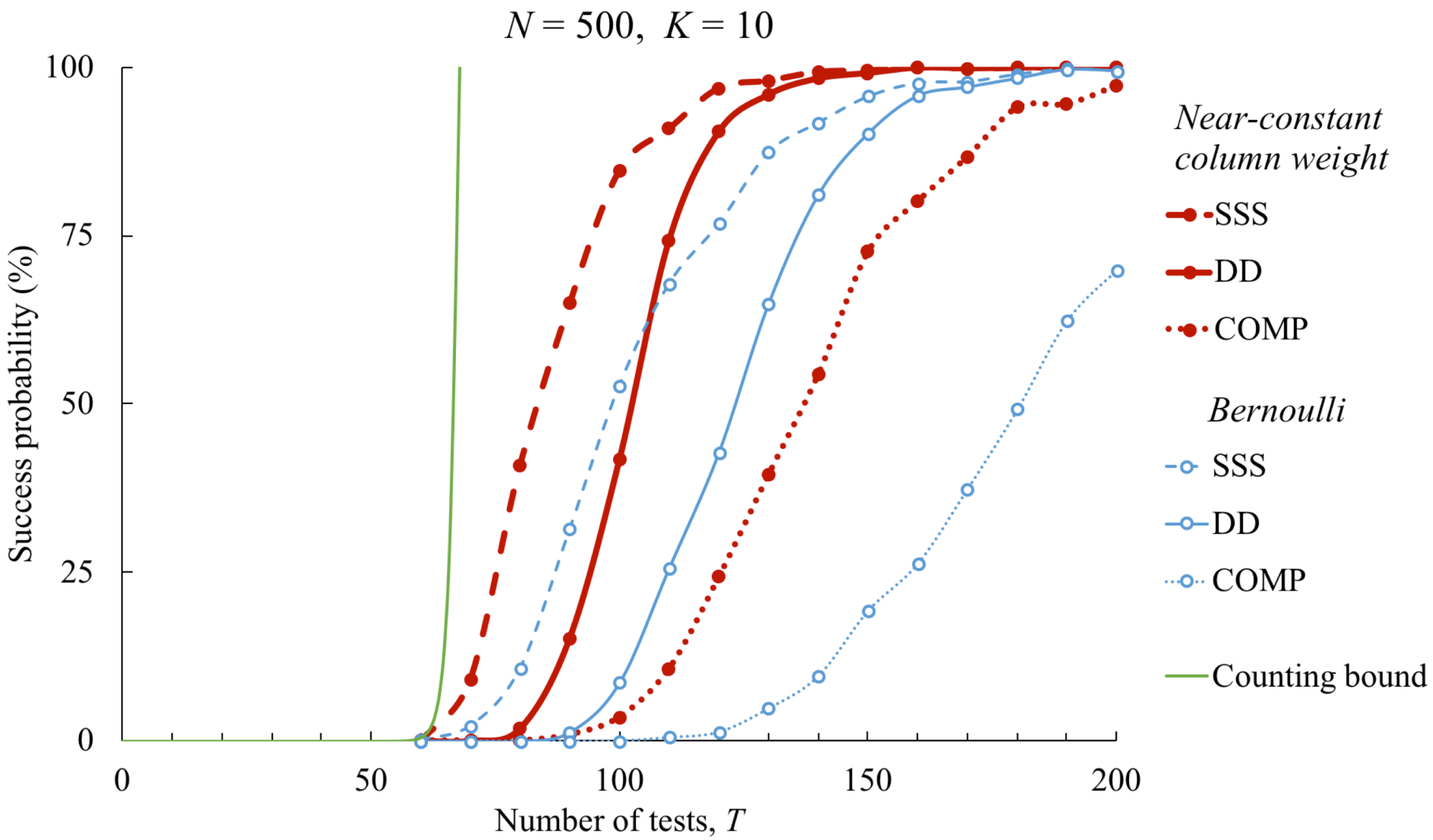
We're introducing anti-correlation into tests,
to try and make sure we get "new" bits of information.

The chance of a nondefective item
being in no negative tests is smaller.

It just is.

	Bernoulli design	Constant tests-per-item
COMP	$0.53(1-\theta)$ (Chan–Che–Jaggi–Saligrama, 2011)	$0.69(1-\theta)$ (Johnson–Aldridge–Scarlett, 2018)
DD	$0.53 \max\left\{1, \frac{1-\theta}{\theta}\right\}$ (Aldridge–Baldassini–Johnson, 2014)	$0.69 \max\left\{1, \frac{1-\theta}{\theta}\right\}$ (Johnson–Aldridge–Scarlett, 2018)
ML	$\max\left\{1, 0.53\,\frac{1-\theta}{\theta}\right\}$ except between 0.33 and 0.36 (Aldridge–Baldassini–Johnson, 2014)	$\max\left\{1, 0.69\,\frac{1-\theta}{\theta}\right\}$ suggested by heuristics (Mézard–Tarzia–Toninelli, 2012)


1
Maximum likelihood
Thick lines:
constant tests-per-item
Thin lines:
Bernoulli design
Rate
0.5
COMP
DD
0
0
0.5
1
Sparsity parameter θ


$N = 500,\ K = 10$
100
75
50
25
0
Success probability (%)
0
50
100
150
200
Number of tests, T
Near-constant column weight
SSS
DD
COMP
Bernoulli
SSS
DD
COMP
Counting bound

Big open question

What is the
maximum possible rate
of nonadaptive group testing?

What is the maximum possible rate of nonadaptive group testing?

I conjecture that the answer is

$$\max\left\{1, \ln 2 \frac{1-\theta}{\theta}\right\} = \max\left\{1, 0.69 \frac{1-\theta}{\theta}\right\}$$

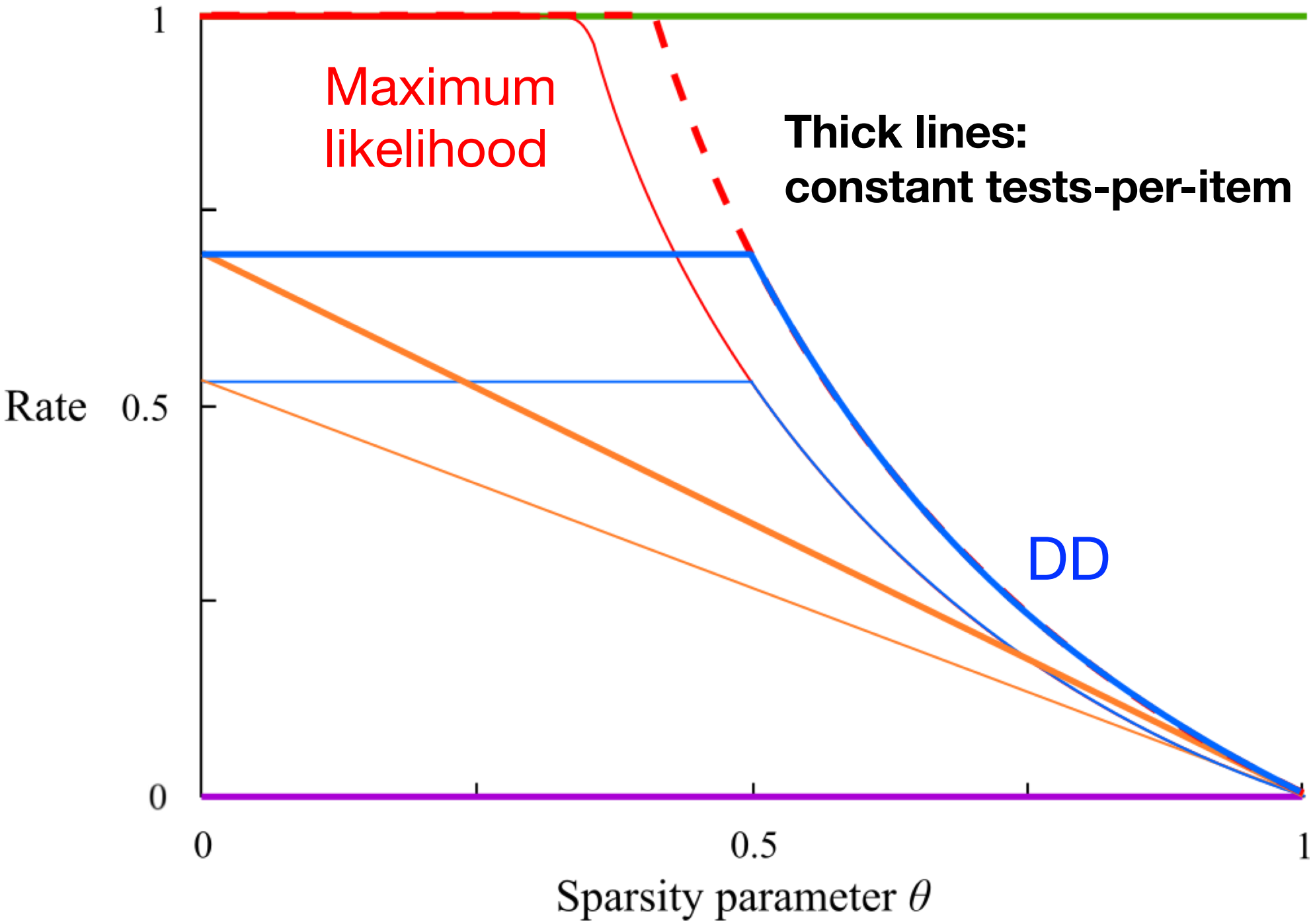
and that it can be achieved with the **constant tests-per-item** design and **maximum likelihood** detection

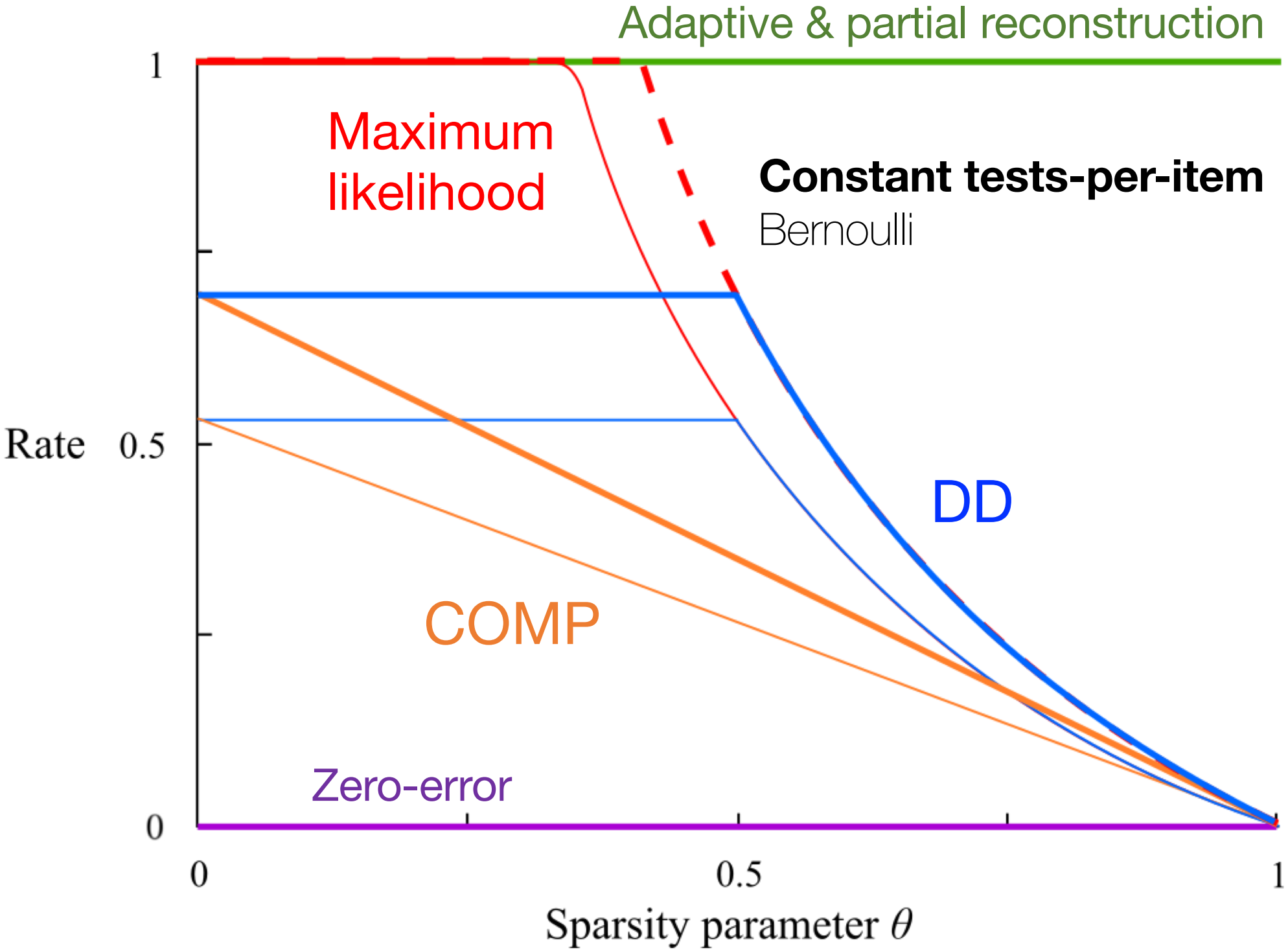
I conjecture that the answer is

$$\max\left\{1, \ln 2 \frac{1-\theta}{\theta}\right\} = \max\left\{1, 0.69 \frac{1-\theta}{\theta}\right\}$$

and that it can be achieved with
the **constant tests-per-item** design
and **maximum likelihood** detection

In other words,
the heuristics from statistical physics are correct
but there is no better design
than constant tests-per-item


1
Maximum likelihood
Thick lines: constant tests-per-item
Rate
0.5
DD
0
0
0.5
1
Sparsity parameter θ


Adaptive & partial reconstruction
1
Maximum likelihood
Constant tests-per-item
Bernoulli
Rate
0.5
DD
COMP
Zero-error
0
0
0.5
1
Sparsity parameter θ