PROOFS OF PROXIMITY FOR DISTRIBUTION TESTING

(Distribution testing – now with proofs!)

Tom Gur (UC Berkeley)

October 14, 2017
FOCS 2017 Workshop: Frontiers in Distribution Testing

Joint work with **Alessandro Chiesa** (UC Berkeley)

PROOFS OF PROXIMITY?

WHAT AND WHY?

What?

What?

Proofs of Proximity are proof systems for property testing [EKR04, BGH+06]

What?

Proofs of Proximity are proof systems for property testing [EKR04, BGH$^+$06]

Many flavors:

What?

Proofs of Proximity are proof systems for property testing [EKR04, BGH+06]

Many flavors: NP

What?

Proofs of Proximity are proof systems for property testing [EKR04, BGH$^+$06]

Many flavors: NP, IP

What?

Proofs of Proximity are proof systems for property testing [EKR04, BGH^{+}06]

Many flavors: NP, IP, PCP

What?

Proofs of Proximity are proof systems for property testing [EKR04, BGH+06]

Many flavors: NP, IP, PCP, MA, and more...

What?

Proofs of Proximity are proof systems for property testing [EKR04, BGH$^+$06]

Many flavors: NP, IP, PCP, MA, and more...

The key difference: approximate decision problems

What?

Proofs of Proximity are proof systems for property testing [EKR04, BGH+06]

Many flavors: NP, IP, PCP, MA, and more...

The key difference: approximate decision problems

Why?

What?

Proofs of Proximity are proof systems for property testing [EKR04, BGH$^+$06]

Many flavors: NP, IP, PCP, MA, and more...

The key difference: approximate decision problems

Why?

Theory: understanding the power and limitations of proofs

What?

Proofs of Proximity are proof systems for property testing [EKR04, BGH+06]

Many flavors: NP, IP, PCP, MA, and more...

The key difference: **approximate decision problems**

Why?

Theory: understanding the power and limitations of proofs

Theory application: while many properties can be tested efficiently

What?

Proofs of Proximity are proof systems for property testing [EKR04, BGH+06]

Many flavors: NP, IP, PCP, MA, and more...

The key difference: approximate decision problems

Why?

Theory: understanding the power and limitations of proofs

Theory application: while many properties can be tested efficiently many other natural properties require a lot of samples

What?

Proofs of Proximity are proof systems for property testing [EKR04, BGH+06]

Many flavors: NP, IP, PCP, MA, and more...

The key difference: approximate decision problems

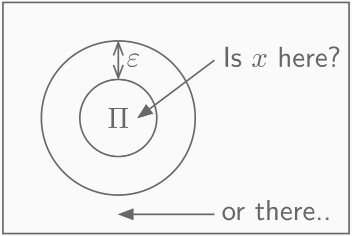
Why?

Theory: understanding the power and limitations of proofs

Theory application: while many properties can be tested efficiently many other natural properties require a lot of samples

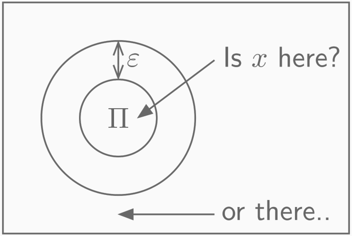
Application: delegation of computation

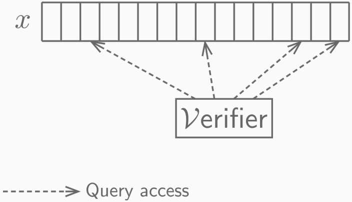
For example, consider interactive proofs of proximity [RVW13]



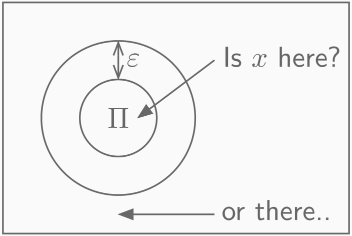
THE STANDARD SETTING: FUNCTIONS

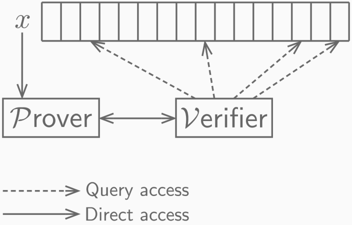
For example, consider interactive proofs of proximity [RVW13]



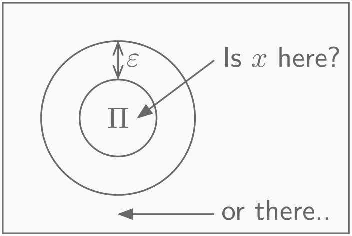


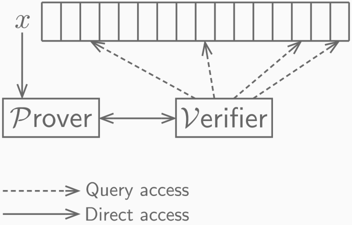
For example, consider interactive proofs of proximity [RVW13]





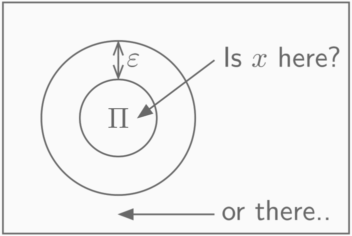
For example, consider interactive proofs of proximity [RVW13]

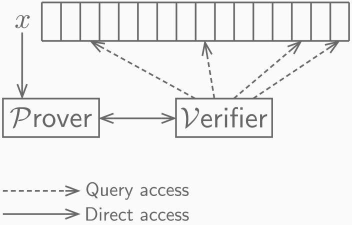




- If $x \in \Pi$, $\exists$ prover strategy P such that $\langle P(x), V^x \rangle(\varepsilon) = 1$

For example, consider interactive proofs of proximity [RVW13]





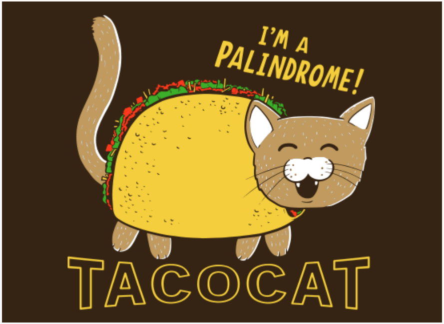
- If $x \in \Pi$, $\exists$ prover strategy P such that $\langle P(x), V^x \rangle(\varepsilon) = 1$
- If x is ε-far from Π, $\forall$ prover strategy $\langle P^*, V^x \rangle(\varepsilon) = 0$ w.h.p.

AN EASY EXAMPLE

How can proofs help testing algorithms?

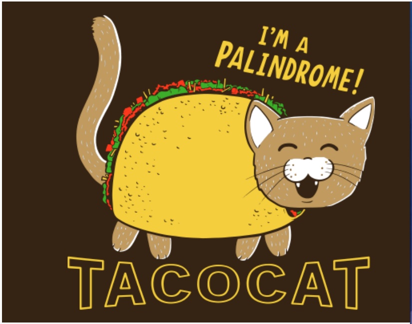
How can proofs help testing algorithms?

Tell us **where** one palindrome ends and the other starts! [FGL14]



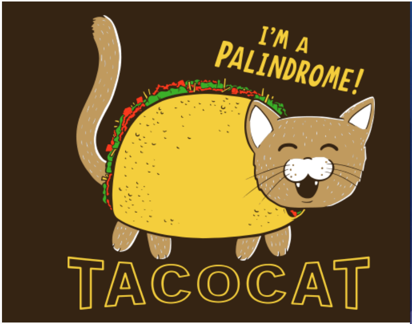
How can proofs help testing algorithms?

Tell us **where** one palindrome ends and the other starts! [FGL14]



How can proofs help testing algorithms?

Tell us **where** one palindrome ends and the other starts! [FGL14]



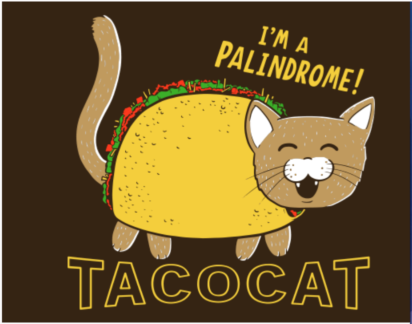
T	A	C	O	C	A	T	U	F	O	T	O	F	U

“Concatenated palindromes” requires $\Omega(\sqrt{n})$ queries [AKNS01]

AN EASY EXAMPLE

How can proofs help testing algorithms?

Tell us **where** one palindrome ends and the other starts! [FGL14]

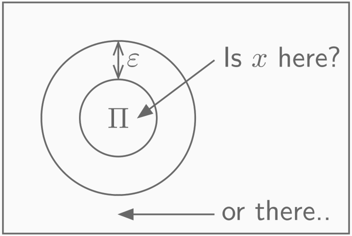


T	A	C	O	C	A	T	U	F	O	T	O	F	U
1	2	3	4	5	6	7	8	9	10	11	12	13	14

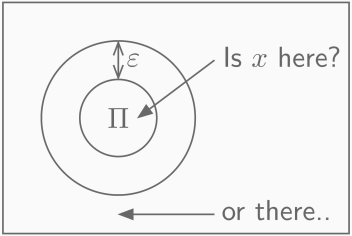
"Concatenated palindromes" requires $\Omega(\sqrt{n})$ queries [AKNS01]

However, a tiny proof of length $\log(n)$ reduces the queries to $O(1)$

NOW FOR DISTRIBUTIONS!

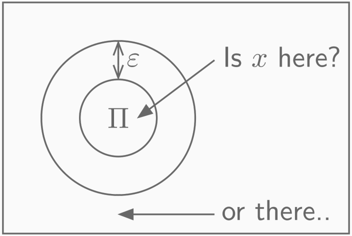


Same problem, different **object**



Same problem, different **object**

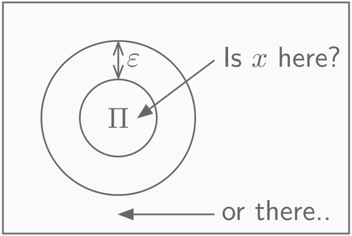
...and **access**



Same problem, different object

...and access

...and distance

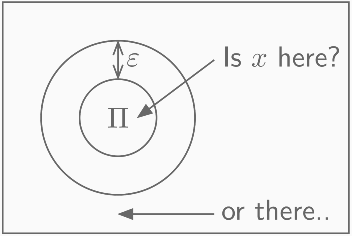


Same problem, different **object**

...and **access**

...and **distance**

Does it really make a difference?



Same problem, different **object**

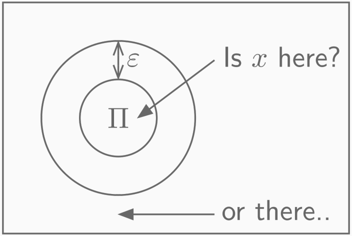
...and **access**

...and **distance**

Does it really make a difference?

The setting:

Known domain (here $[n] = \{1, \ldots, n\}$)



Same problem, different **object**

...and **access**

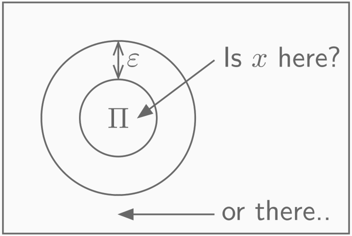
...and **distance**

Does it really make a difference?

The setting:

Known domain (here $[n] = \{1, \ldots, n\}$)

x is now a distribution, let's call it $D \in \Delta([n])$



Same problem, different **object**

...and **access**

...and **distance**

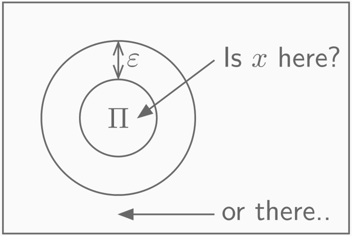
Does it really make a difference?

The setting:

Known domain (here $[n] = \{1, \ldots, n\}$)

x is now a distribution, let's call it $D \in \Delta([n])$

Property $\Pi \subseteq \Delta([n])$, proximity parameter $\varepsilon \in (0, 1]$



Same problem, different **object**

...and **access**

...and **distance**

Does it really make a difference?

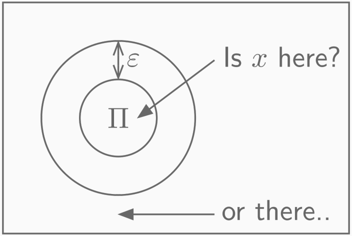
The setting:

Known domain (here $[n] = \{1, \ldots, n\}$)

x is now a distribution, let's call it $D \in \Delta([n])$

Property $\Pi \subseteq \Delta([n])$, proximity parameter $\varepsilon \in (0, 1]$

Sample access to D



Same problem, different **object**

...and **access**

...and **distance**

Does it really make a difference?

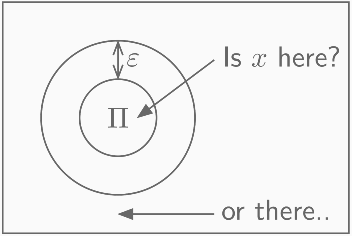
The setting:

Known domain (here $[n] = \{1, \ldots, n\}$)

x is now a distribution, let's call it $D \in \Delta([n])$

Property $\Pi \subseteq \Delta([n])$, proximity parameter $\varepsilon \in (0, 1]$

Sample access to D



Same problem, different **object**

...and **access**

...and **distance**

Does it really make a difference?

The setting:

Known domain (here $[n] = \{1, \ldots, n\}$)

x is now a distribution, let's call it $D \in \Delta([n])$

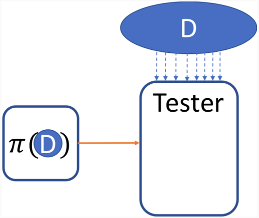
Property $\Pi \subseteq \Delta([n])$, proximity parameter $\varepsilon \in (0, 1]$

Sample access to D

Decide with high probability:

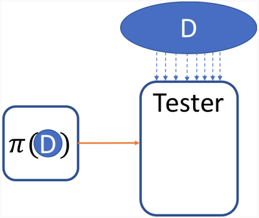
$$\text{Is } D \in \Pi, \text{ or } \delta_{\mathrm{TV}}(D, \Pi) > \varepsilon?$$

NP distribution testers



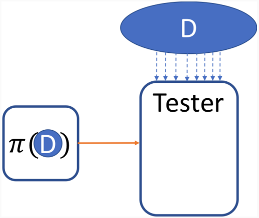
Deterministic algorithm T with

NP distribution testers



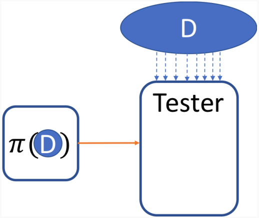
Deterministic algorithm T with sample access to $D \in \Delta([n])$

NP distribution testers



Deterministic algorithm T with
sample access to $D \in \Delta([n])$
explicit access to $\varepsilon > 0$ and a proof π:

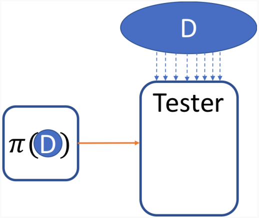
NP distribution testers



Deterministic algorithm T with
sample access to $D \in \Delta([n])$
explicit access to $\varepsilon > 0$ and a proof π:

* For every $D \in \Pi$, there exists proof π s.t. $T^D(\varepsilon, \pi) = 1$

NP distribution testers

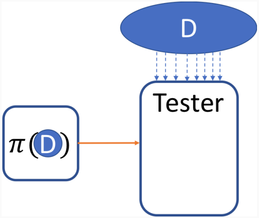


Deterministic algorithm T with
sample access to $D \in \Delta([n])$
explicit access to $\varepsilon > 0$ and a proof π:

* For every $D \in \Pi$, there exists proof π s.t. $T^D(\varepsilon, \pi) = 1$

* For every $\delta_{\mathrm{TV}}(D, \Pi) > \varepsilon$ and any "proof" π, $\Pr[T^D(\varepsilon, \pi) = 0] \geq 2/3$

NP distribution testers

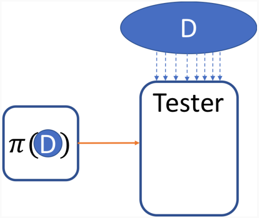


Deterministic algorithm T with
sample access to $D \in \Delta([n])$
explicit access to $\varepsilon > 0$ and a proof π:

* For every $D \in \Pi$, there exists proof π s.t. $T^D(\varepsilon, \pi) = 1$

* For every $\delta_{\mathrm{TV}}(D, \Pi) > \varepsilon$ and any "proof" π, $\Pr[T^D(\varepsilon, \pi) = 0] \geq 2/3$

NP distribution testers



Deterministic algorithm T with
sample access to $D \in \Delta([n])$
explicit access to $\varepsilon > 0$ and a proof π:

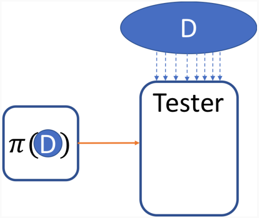
* For every $D \in \Pi$, there exists proof π s.t. $T^D(\varepsilon, \pi) = 1$

* For every $\delta_{\mathrm{TV}}(D, \Pi) > \varepsilon$ and any "proof" π, $\Pr[T^D(\varepsilon, \pi) = 0] \geq 2/3$

MA distribution testers

NP distribution testers that are allowed to toss coins

NP distribution testers



Deterministic algorithm T with
sample access to $D \in \Delta([n])$
explicit access to $\varepsilon > 0$ and a proof π:

* For every $D \in \Pi$, there exists proof π s.t. $T^D(\varepsilon, \pi) = 1$

* For every $\delta_{\mathrm{TV}}(D, \Pi) > \varepsilon$ and any "proof" π, $\Pr[T^D(\varepsilon, \pi) = 0] \geq 2/3$

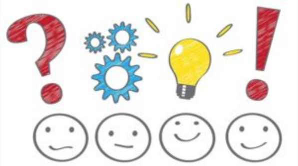
MA distribution testers

NP distribution testers that are allowed to toss coins

IP distribution testers

MA distribution testers that interact with a prover

This is all very nice, but:



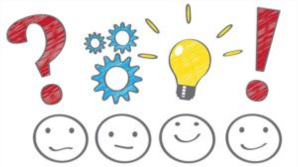
SOME QUESTIONS

This is all very nice, but:

- Are proof-augmented testers stronger than standard testers?

SOME QUESTIONS

This is all very nice, but:

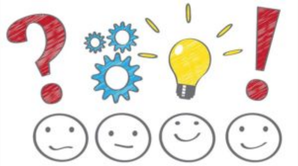


- Are proof-augmented testers stronger than standard testers?
- If so, to what extent?

This is all very nice, but:

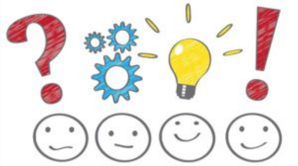
- Are proof-augmented testers stronger than standard testers?
- If so, to what extent?

This is all very nice, but:



- Are proof-augmented testers stronger than standard testers?
- If so, to what extent? Polynomially better?

This is all very nice, but:



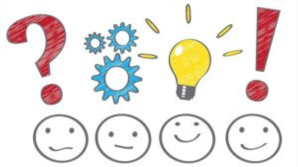
- Are proof-augmented testers stronger than standard testers?
- If so, to what extent? Polynomially better? Exponentially better?

This is all very nice, but:

- Are proof-augmented testers stronger than standard testers?
- If so, to what extent? Polynomially better? Exponentially better? Large classes?

SOME QUESTIONS

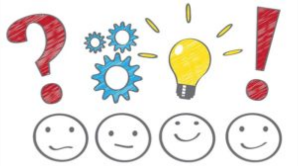
This is all very nice, but:



- Are proof-augmented testers stronger than standard testers?
- If so, to what extent? Polynomially better? Exponentially better? Large classes?
- What are the most important resources?

SOME QUESTIONS

This is all very nice, but:

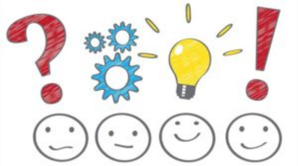


- Are proof-augmented testers stronger than standard testers?
- If so, to what extent? Polynomially better? Exponentially better? Large classes?
- What are the most important resources?

This is all very nice, but:

- Are proof-augmented testers stronger than standard testers?
- If so, to what extent? Polynomially better? Exponentially better? Large classes?
- What are the most important resources? Randomness?

This is all very nice, but:



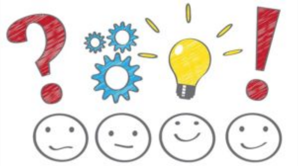
- Are proof-augmented testers stronger than standard testers?
- If so, to what extent? Polynomially better? Exponentially better? Large classes?
- What are the most important resources? Randomness? Interaction?

This is all very nice, but:

- Are proof-augmented testers stronger than standard testers?
- If so, to what extent? Polynomially better? Exponentially better? Large classes?
- What are the most important resources? Randomness? Interaction? Private coins?

SOME QUESTIONS

This is all very nice, but:



- Are proof-augmented testers stronger than standard testers?
- If so, to what extent? Polynomially better? Exponentially better? Large classes?
- What are the most important resources? Randomness? Interaction? Private coins?
- What can and cannot be achieved with each proof system?

FUNCTIONS VS DISTRIBUTIONS

		Testing Distributions this work	**Testing Functions** [RVW13; GR16; FGL14; GR17]
non-interactive proofs	**Proofs of linear length**	**Different!**	reduce sample complexity of *any* property to $O(1)$
	MA proofs of proximity vs. standard testers	**Different!**	exponentially stronger
	Probabilistic (MA) vs. deterministic (NP) verification	**Different!**	NP proofs of proximity are extremely weak
	Hardest property for non-interactive proofs	**Different!**	non-explicit (random property); linear length proof is required to outperform standard testers
interactive proofs	**Private vs. public coin protocols**	**Different!**	almost equivalent
	AM round hierarchy coin protocols	**Different!**	there is a property for which the AM complexity is $\approx \mathsf{n}^{1/\mathsf{r}}$ for r-round protocols

		Testing Distributions this work	**Testing Functions** [RVW13; GR16; FGL14; GR17]
non-interactive proofs	**Proofs of linear length**	**Different!**	reduce sample complexity of *any* property to $O(1)$
	MA proofs of proximity vs. standard testers	**Different!**	exponentially stronger
	Probabilistic (MA) vs. deterministic (NP) verification	**Different!**	NP proofs of proximity are extremely weak
	Hardest property for non-interactive proofs	**Different!**	non-explicit (random property); linear length proof is required to outperform standard testers
interactive proofs	**Private vs. public coin protocols**	**Different!**	almost equivalent
	AM round hierarchy coin protocols	**Different!**	there is a property for which the AM complexity is $\approx \mathsf{n}^{1/\mathsf{r}}$ for r-round protocols

FIRST EXAMPLE

SUPPORT SIZE

Consider the support size problem:

$$\mathrm{SuppSize}_{\leq n/2} = \{D \in \Delta([n]) : |\mathrm{supp}(D)| \leq n/2\}$$

Consider the support size problem:

$$\mathrm{SuppSize}_{\leq n/2} = \{D \in \Delta([n]) : |\mathrm{supp}(D)| \leq n/2\}$$

This is a hard problem

Consider the support size problem:

$$\mathrm{SuppSize}_{\leq n/2} = \{D \in \Delta([n]) : |\mathrm{supp}(D)| \leq n/2\}$$

This is a hard problem (requires $\Omega(n/\log(n))$ samples [Val11])

SUPPORT SIZE

Consider the support size problem:

$$\mathrm{SuppSize}_{\leq n/2} = \{D \in \Delta([n]) : |\mathrm{supp}(D)| \leq n/2\}$$

This is a hard problem (requires $\Omega(n/\log(n))$ samples [Val11])

...unless a prover is giving us support!

Consider the support size problem:

$$\mathrm{SuppSize}_{\leq n/2} = \{D \in \Delta([n]) : |\mathrm{supp}(D)| \leq n/2\}$$

This is a hard problem (requires $\Omega(n/\log(n))$ samples [Val11])

...unless a prover is giving us support!

Or rather, a prover is specifying $\mathrm{supp}(D)$...

Consider the support size problem:

$$\mathrm{SuppSize}_{\leq n/2} = \{D \in \Delta([n]) : |\mathrm{supp}(D)| \leq n/2\}$$

This is a hard problem (requires $\Omega(n/\log(n))$ samples [Val11])

...unless a prover is giving us support!

Or rather, a prover is specifying $\mathrm{supp}(D)$...

Then we only need $O(1/\varepsilon)$ samples to detect whether is D is ε-far from $\mathrm{SuppSize}_{\leq k}$

SUPPORT SIZE

Consider the support size problem:

$$\mathrm{SuppSize}_{\leq n/2} = \{D \in \Delta([n]) : |\mathrm{supp}(D)| \leq n/2\}$$

This is a hard problem (requires $\Omega(n/\log(n))$ samples [Val11])

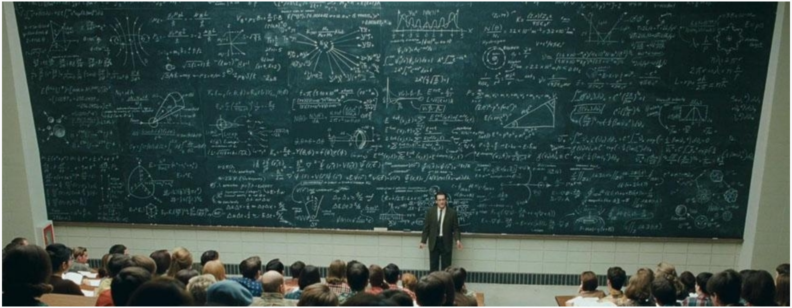
...unless a prover is giving us support!

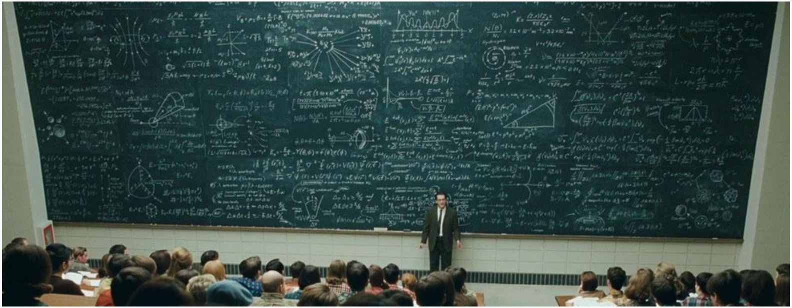
Or rather, a prover is specifying $\mathrm{supp}(D)$...

Then we only need $O(1/\varepsilon)$ samples to detect whether is D is ε-far from $\mathrm{SuppSize}_{\leq k}$

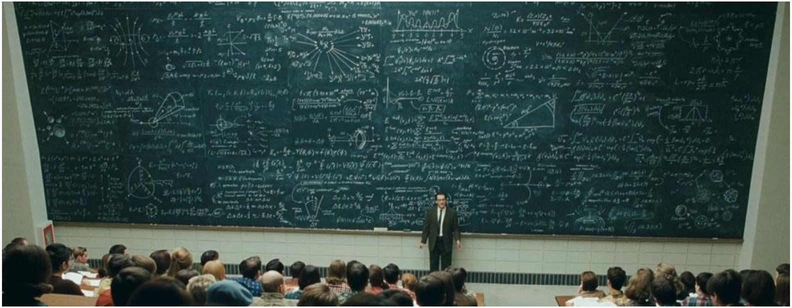
Caveat: this requires a long proof ($O(n \log n)$ bits)

ON LONG PROOFS – FUNCTIONS





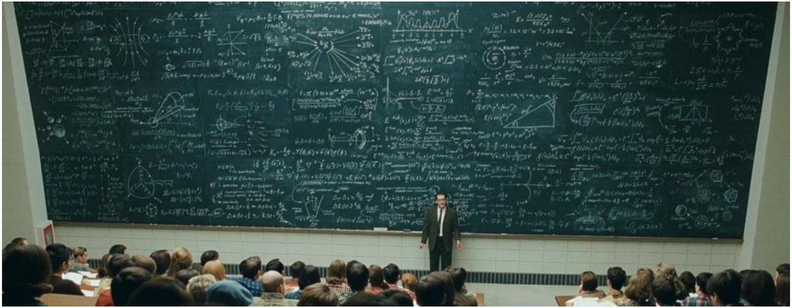
The proof length is a key complexity measure for proofs of proximity



The proof length is a key complexity measure for proofs of proximity

For functions, linear-length proofs completely trivialize the model!

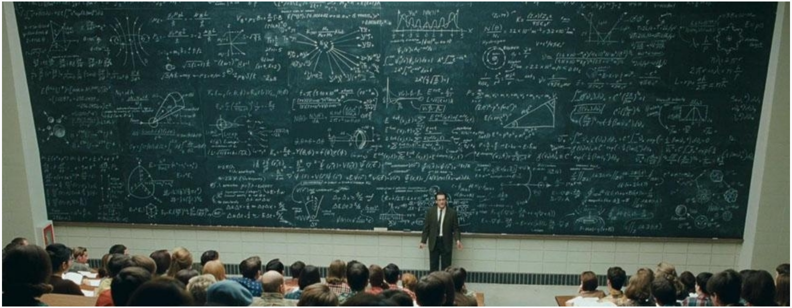
Why?



The proof length is a key complexity measure for proofs of proximity

For functions, linear-length proofs completely trivialize the model!

Why? (How to check that function f has property Π)

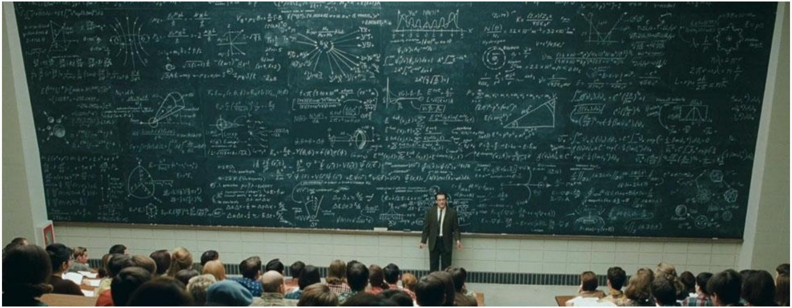


The proof length is a key complexity measure for proofs of proximity

For functions, linear-length proofs completely trivialize the model!

Why? (How to check that function f has property Π)

The tester has explicit access to the proof π

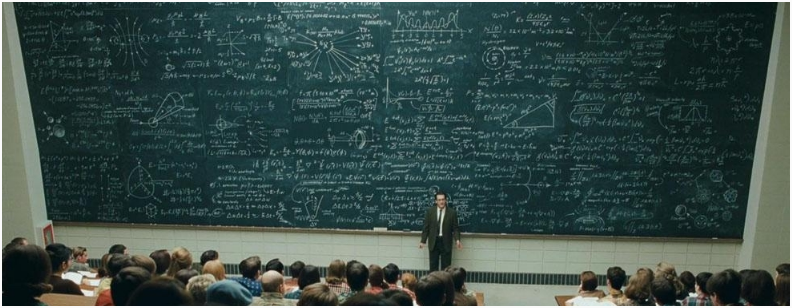


The proof length is a key complexity measure for proofs of proximity

For functions, linear-length proofs completely trivialize the model!

Why? (How to check that function f has property Π)

The tester has explicit access to the proof π
If $\pi = f$ it can directly check whether $\pi \in \Pi$

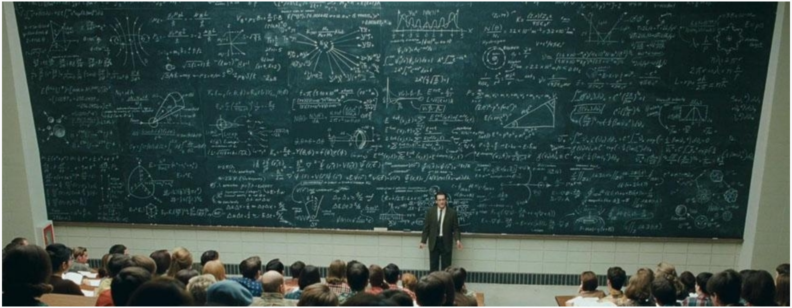


The proof length is a key complexity measure for proofs of proximity

For functions, linear-length proofs completely trivialize the model!

Why? (How to check that function f has property Π)

The tester has explicit access to the proof π
If $\pi = f$ it can directly check whether $\pi \in \Pi$
Hence, it boils down to test that f is identical to π

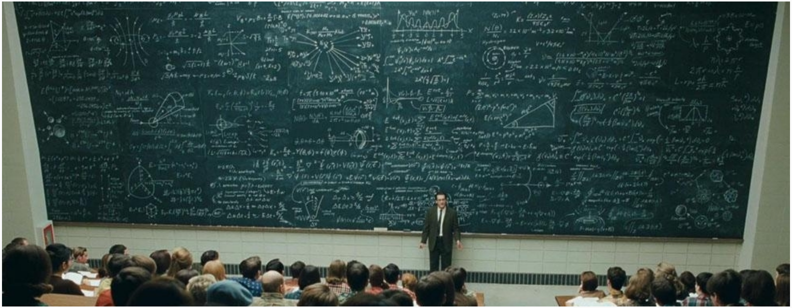


The proof length is a key complexity measure for proofs of proximity

For functions, linear-length proofs completely trivialize the model!

Why? (How to check that function f has property Π)

The tester has explicit access to the proof π
If $\pi = f$ it can directly check whether $\pi \in \Pi$
Hence, it boils down to test that f is identical to π
which can easily be done using $O(1/\varepsilon)$ queries



The proof length is a key complexity measure for proofs of proximity

For functions, linear-length proofs completely trivialize the model!

Why? (How to check that function f has property Π)

The tester has explicit access to the proof π
If $\pi = f$ it can directly check whether $\pi \in \Pi$
Hence, it boils down to test that f is identical to π
which can easily be done using $O(1/\varepsilon)$ queries...for functions

For distribution testing, testing identity is much harder:

For distribution testing, testing identity is much harder: $O(\sqrt{n}/\varepsilon^2)$

ON LONG PROOFS – DISTRIBUTIONS

For distribution testing, testing identity is much harder: $O(\sqrt{n}/\varepsilon^2)$

…or even $O(\|D^{-\max}_{-\varepsilon/16}\|_{2/3})$ [VV17] where $\|\cdot\|_{2/3}$ denotes the $\ell_{2/3}$ quasi-norm, and $D^{-\mathbf{max}}_{-\varepsilon/16}$ is the distribution obtained by removing the maximal element of D as well as removing a maximal set of elements of total mass $\varepsilon/16$

ON LONG PROOFS — DISTRIBUTIONS

For distribution testing, testing identity is much harder: $O(\sqrt{n}/\varepsilon^2)$

...or even $O(\|D^{-\max}_{-\varepsilon/16}\|_{2/3})$ [VV17] where $\|\cdot\|_{2/3}$ denotes the $\ell_{2/3}$ quasi-norm, and $D^{-\mathbf{max}}_{-\varepsilon/16}$ is the distribution obtained by removing the maximal element of D as well as removing a maximal set of elements of total mass $\varepsilon/16$

...or perhaps $O(\kappa_D^{-1}(1-c\varepsilon))$ [BCG17] where $c > 0$ is a constant, and κ_D is the K-functional between ℓ_1 and ℓ_2 with respect to the distribution D

For distribution testing, testing identity is much harder: $O(\sqrt{n}/\varepsilon^2)$

...or even $O(\|D^{-\max}_{-\varepsilon/16}\|_{2/3})$ [VV17] where $\|\cdot\|_{2/3}$ denotes the $\ell_{2/3}$ quasi-norm, and $D^{-\mathbf{max}}_{-\varepsilon/16}$ is the distribution obtained by removing the maximal element of D as well as removing a maximal set of elements of total mass $\varepsilon/16$

...or perhaps $O(\kappa_D^{-1}(1 - c\varepsilon))$ [BCG17] where $c > 0$ is a constant, and κ_D is the K-functional between ℓ_1 and ℓ_2 with respect to the distribution D

But wait, how can the proof fully describe the distribution?

For distribution testing, testing identity is **much harder**: $O(\sqrt{n}/\varepsilon^2)$

…or even $O(\|D^{-\max}_{-\varepsilon/16}\|_{2/3})$ [VV17] where $\|\cdot\|_{2/3}$ denotes the $\ell_{2/3}$ quasi-norm, and $D^{-\max}_{-\varepsilon/16}$ is the distribution obtained by removing the maximal element of D as well as removing a maximal set of elements of total mass $\varepsilon/16$

…or perhaps $O(\kappa_D^{-1}(1-c\varepsilon))$ [BCG17] where $c > 0$ is a constant, and κ_D is the K-functional between ℓ_1 and ℓ_2 with respect to the distribution D

But wait, how can the proof fully describe the distribution?

The description of $D \in \Delta([n])$ may be very large
(even infinite…)

For distribution testing, testing identity is **much harder**: $O(\sqrt{n}/\varepsilon^2)$

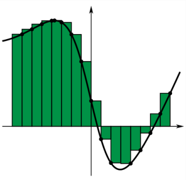
...or even $O(\|D^{-\max}_{-\varepsilon/16}\|_{2/3})$ [VV17] where $\|\cdot\|_{2/3}$ denotes the $\ell_{2/3}$ quasi-norm, and $D^{-\max}_{-\varepsilon/16}$ is the distribution obtained by removing the maximal element of D as well as removing a maximal set of elements of total mass $\varepsilon/16$

...or perhaps $O(\kappa_D^{-1}(1-c\varepsilon))$ [BCG17] where $c > 0$ is a constant, and κ_D is the K-functional between ℓ_1 and ℓ_2 with respect to the distribution D

But wait, how can the proof fully describe the distribution?

The description of $D \in \Delta([n])$ may be very large (even infinite...)

Luckily, it suffices to send a **granular approximation** D_{approx} of D



For distribution testing, testing identity is **much harder**: $O(\sqrt{n}/\varepsilon^2)$

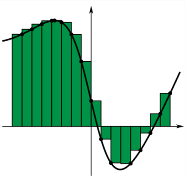
...or even $O(\|D^{-\max}_{-\varepsilon/16}\|_{2/3})$ [VV17] where $\|\cdot\|_{2/3}$ denotes the $\ell_{2/3}$ quasi-norm, and $D^{-\max}_{-\varepsilon/16}$ is the distribution obtained by removing the maximal element of D as well as removing a maximal set of elements of total mass $\varepsilon/16$

...or perhaps $O(\kappa_D^{-1}(1-c\varepsilon))$ [BCG17] where $c > 0$ is a constant, and κ_D is the K-functional between ℓ_1 and ℓ_2 with respect to the distribution D

But wait, how can the proof fully describe the distribution?

The description of $D \in \Delta([n])$ may be very large (even infinite...)

Luckily, it suffices to send a **granular approximation** D_{approx} of D



What if $D \in \Pi$, but D_{approx} is close to, yet not in Π?

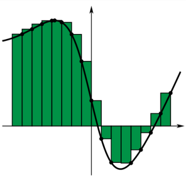
For distribution testing, testing identity is much harder: $O(\sqrt{n}/\varepsilon^2)$

...or even $O(\|D^{-\max}_{-\varepsilon/16}\|_{2/3})$ [VV17] where $\|\cdot\|_{2/3}$ denotes the $\ell_{2/3}$ quasi-norm, and $D^{-\max}_{-\varepsilon/16}$ is the distribution obtained by removing the maximal element of D as well as removing a maximal set of elements of total mass $\varepsilon/16$

...or perhaps $O(\kappa_D^{-1}(1-c\varepsilon))$ [BCG17] where $c > 0$ is a constant, and κ_D is the K-functional between ℓ_1 and ℓ_2 with respect to the distribution D

But wait, how can the proof fully describe the distribution?

The description of $D \in \Delta([n])$ may be very large (even infinite...)



Luckily, it suffices to send a granular approximation D_{approx} of D

What if $D \in \Pi$, but D_{approx} is close to, yet not in Π?

We can use a tolerant tester to make sure it rules the same

FUNCTIONS VS DISTRIBUTIONS

		Testing Distributions this work	**Testing Functions** [RVW13; GR16; FGL14; GR17]
non-interactive proofs	**Proofs of linear length**	reduce sample complexity of *any* property to $O(\sqrt{\mathsf{n}})$	reduce sample complexity of *any* property to $O(1)$
	MA proofs of proximity vs. standard testers	**Different!**	exponentially stronger
	Probabilistic (MA) vs. deterministic (NP) verification	**Different!**	NP proofs of proximity are extremely weak
	Hardest property for non-interactive proofs	**Different!**	non-explicit (random property); linear length proof is required to outperform standard testers
interactive proofs	**Private vs. public coin protocols**	**Different!**	almost equivalent
	AM round hierarchy coin protocols	**Different!**	there is a property for which the AM complexity is $\approx \mathsf{n}^{1/\mathsf{r}}$ for r-round protocols

WHAT ABOUT SHORT PROOFS?

So far we saw that:

- any property can be tested using $O(\sqrt{n}/\varepsilon^2)$-ish samples

WHAT ABOUT SHORT PROOFS?

So far we saw that:

- any property can be tested using $O(\sqrt{n}/\varepsilon^2)$-ish samples
- there exists a (hard) property that is testable via $O(1/\varepsilon)$ samples

WHAT ABOUT SHORT PROOFS?

So far we saw that:

- any property can be tested using $O(\sqrt{n}/\varepsilon^2)$-ish samples
- there exists a (hard) property that is testable via $O(1/\varepsilon)$ samples

WHAT ABOUT SHORT PROOFS?

So far we saw that:

- any property can be tested using $O(\sqrt{n}/\varepsilon^2)$-ish samples
- there exists a (hard) property that is testable via $O(1/\varepsilon)$ samples

Can we get significant savings via short (sublinear) proofs?

So far we saw that:

- any property can be tested using $O(\sqrt{n}/\varepsilon^2)$-ish samples
- there exists a (hard) property that is testable via $O(1/\varepsilon)$ samples

Can we get significant savings via short (sublinear) proofs? Yes!

Theorem

There exists a property that requires $\tilde{\Omega}(\sqrt{n})$ samples to test, yet $\forall\beta$, given a proof of length $\tilde{O}(n^{\beta})$ can be tested using $O(n^{1-\beta})$ samples

So far we saw that:

- any property can be tested using $O(\sqrt{n}/\varepsilon^2)$-ish samples
- there exists a (hard) property that is testable via $O(1/\varepsilon)$ samples

Can we get significant savings via short (sublinear) proofs? Yes!

Theorem

There exists a property that requires $\tilde{\Omega}(\sqrt{n})$ samples to test, yet $\forall\beta$, given a proof of length $\tilde{O}(n^\beta)$ can be tested using $O(n^{1-\beta})$ samples

So far we saw that:

- any property can be tested using $O(\sqrt{n}/\varepsilon^2)$-ish samples
- there exists a (hard) property that is testable via $O(1/\varepsilon)$ samples

Can we get significant savings via short (sublinear) proofs? Yes!

Theorem

There exists a property that requires $\tilde{\Omega}(\sqrt{n})$ samples to test, yet $\forall\beta$, given a proof of length $\tilde{O}(n^{\beta})$ can be tested using $O(n^{1-\beta})$ samples

But can we do better?

So far we saw that:

- any property can be tested using $O(\sqrt{n}/\varepsilon^2)$-ish samples
- there exists a (hard) property that is testable via $O(1/\varepsilon)$ samples

Can we get significant savings via short (sublinear) proofs? Yes!

Theorem

There exists a property that requires $\tilde{\Omega}(\sqrt{n})$ samples to test, yet $\forall\beta$, given a proof of length $\tilde{O}(n^\beta)$ can be tested using $O(n^{1-\beta})$ samples

But can we do better? Not much... (not without interaction)

Lemma

For every Π and MA distribution tester for Π with proof length p and sample complexity s, it holds that $p \cdot s = \Omega(\mathrm{SAMP}(\Pi))$

The idea

Lemma

For every Π and MA distribution tester for Π with proof length p and sample complexity s, it holds that $p \cdot s = \Omega(\mathrm{SAMP}(\Pi))$

The idea

Distribution testers are not only non-adaptive w.r.t. the samples,

Lemma

For every Π and MA distribution tester for Π with proof length p and sample complexity s, it holds that $p \cdot s = \Omega(\mathrm{SAMP}(\Pi))$

The idea

Distribution testers are not only non-adaptive w.r.t. the samples, but also w.r.t. the proof

Lemma

For every Π and MA distribution tester for Π with proof length p and sample complexity s, it holds that $p \cdot s = \Omega(\mathrm{SAMP}(\Pi))$

The idea

Distribution testers are not only non-adaptive w.r.t. the samples, but also w.r.t. the proof

Thus, testers can emulate all possible proofs reusing the samples!

Lemma

For every Π and MA distribution tester for Π with proof length p and sample complexity s, it holds that $p \cdot s = \Omega(\mathrm{SAMP}(\Pi))$

The idea

Distribution testers are not only non-adaptive w.r.t. the samples, but also w.r.t. the proof

Thus, testers can emulate all possible proofs reusing the samples!

Since there are 2^p possible proofs, we need to amplify the soundness to assure no error occurs w.h.p.

Lemma

For every Π and MA distribution tester for Π with proof length p and sample complexity s, it holds that $p \cdot s = \Omega(\mathrm{SAMP}(\Pi))$

The idea

Distribution testers are not only non-adaptive w.r.t. the samples, but also w.r.t. the proof

Thus, testers can emulate all possible proofs reusing the samples!

Since there are 2^p possible proofs, we need to amplify the soundness to assure no error occurs w.h.p.

To this end, we invoke the tester $O(p)$ times, increasing the sample complexity to $O(p \cdot s)$.

Lemma

For every Π and MA distribution tester for Π with proof length p and sample complexity s, it holds that $p \cdot s = \Omega(\mathrm{SAMP}(\Pi))$

What does this mean?

Lemma

For every Π and MA distribution tester for Π with proof length p and sample complexity s, it holds that $p \cdot s = \Omega(\mathrm{SAMP}(\Pi))$

What does this mean?

- Non-interactive proofs can only yield multiplicative tradeoffs

Lemma

For every Π and MA distribution tester for Π with proof length p and sample complexity s, it holds that $p \cdot s = \Omega(\mathrm{SAMP}(\Pi))$

What does this mean?

- Non-interactive proofs can only yield **multiplicative tradeoffs**
- The max between proof and sample complexity can only be **quadratically** better

Lemma

For every Π and MA distribution tester for Π with proof length p and sample complexity s, it holds that $p \cdot s = \Omega(\mathrm{SAMP}(\Pi))$

What does this mean?

- Non-interactive proofs can only yield **multiplicative tradeoffs**
- The max between proof and sample complexity can only be **quadratically** better
- This lemma allows us to “lift” standard lower bounds to MA lower bounds

Lemma

For every Π and MA distribution tester for Π with proof length p and sample complexity s, it holds that $p \cdot s = \Omega(\mathrm{SAMP}(\Pi))$

What does this mean?

- Non-interactive proofs can only yield multiplicative tradeoffs
- The max between proof and sample complexity can only be quadratically better
- This lemma allows us to “lift” standard lower bounds to MA lower bounds
- Dramatically different behavior than in the functional setting (there MA is exponentially stronger than standard testers)

FUNCTIONS VS DISTRIBUTIONS

		Testing Distributions this work	**Testing Functions** [RVW13; GR16; FGL14; GR17]
non-interactive proofs	**Proofs of linear length**	reduce sample complexity of *any* property to $O(\sqrt{n})$	reduce sample complexity of *any* property to $O(1)$
	MA proofs of proximity vs. standard testers	quadratically stronger	exponentially stronger
	Probabilistic (MA) vs. deterministic (NP) verification	**Different!**	NP proofs of proximity are extremely weak
	Hardest property for non-interactive proofs	**Different!**	non-explicit (random property); linear length proof is required to outperform standard testers
interactive proofs	**Private vs. public coin protocols**	**Different!**	almost equivalent
	AM round hierarchy coin protocols	**Different!**	there is a property for which the AM complexity is $\approx n^{1/r}$ for r-round protocols

In all the proof systems we saw, the testers toss coins

In all the proof systems we saw, the testers toss coins

Indeed, in the functional setting, NP proofs of proximity are extremely weak – the focus is on MA

In all the proof systems we saw, the testers toss coins

Indeed, in the functional setting, NP proofs of proximity are extremely weak – the focus is on MA

In stark contrast, in distribution testing, in turns out that NP proofs are nearly equivalent to MA proofs!

In all the proof systems we saw, the testers toss coins

Indeed, in the functional setting, NP proofs of proximity are extremely weak – the focus is on MA

In stark contrast, in distribution testing, in turns out that NP proofs are nearly equivalent to MA proofs!

Theorem

Every MA distribution tester with sample complexity s can be emulated by an NP distribution tester with the same proof length and sample complexity $O(s + \log n)$.

Key idea: The deterministic tester has access to random samples

Key idea: The deterministic tester has access to random samples
It can extract its inner randomness from them!

Key idea: The deterministic tester has access to random samples
It can extract its inner randomness from them!

1. **Draw samples.** Max between the sample complexity and samples needed to extract randomness.

Key idea: The deterministic tester has access to random samples
It can extract its inner randomness from them!

1. **Draw samples.** Max between the sample complexity and samples needed to extract randomness.
2. **Low-entropy test.** If the samples are not "random enough" for efficient extraction – we can test without proofs

Key idea: The deterministic tester has access to random samples
It can extract its inner randomness from them!

1. **Draw samples.** Max between the sample complexity and samples needed to extract randomness.
2. **Low-entropy test.** If the samples are not "random enough" for efficient extraction – we can test without proofs
3. **Deterministic extraction.** Generalize the Von-Neumann extractor [Von51]

Key idea: The deterministic tester has access to random samples
It can extract its inner randomness from them!

1. **Draw samples.** Max between the sample complexity and samples needed to extract randomness.
2. **Low-entropy test.** If the samples are not "random enough" for efficient extraction – we can test without proofs
3. **Deterministic extraction.** Generalize the Von-Neumann extractor [Von51]
4. **Invoke the MA distribution tester.** Where coin tosses are replaced with the extracted randomness

Key idea: The deterministic tester has access to random samples
It can extract its inner randomness from them!

1. **Draw samples.** Max between the sample complexity and samples needed to extract randomness.
2. **Low-entropy test.** If the samples are not "random enough" for efficient extraction – we can test without proofs
3. **Deterministic extraction.** Generalize the Von-Neumann extractor [Von51]
4. **Invoke the MA distribution tester.** Where coin tosses are replaced with the extracted randomness

Key idea: The deterministic tester has access to random samples
It can extract its inner randomness from them!

1. **Draw samples.** Max between the sample complexity and samples needed to extract randomness.
2. **Low-entropy test.** If the samples are not "random enough" for efficient extraction – we can test without proofs
3. **Deterministic extraction.** Generalize the Von-Neumann extractor [Von51]
4. **Invoke the MA distribution tester.** Where coin tosses are replaced with the extracted randomness

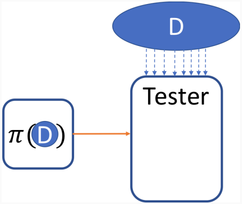
Main technical difficulty: prove a randomness reduction lemma

FUNCTIONS VS DISTRIBUTIONS

		Testing Distributions this work	**Testing Functions** [RVW13; GR16; FGL14; GR17]
non-interactive proofs	**Proofs of linear length**	reduce sample complexity of *any* property to $O(\sqrt{n})$	reduce sample complexity of *any* property to $O(1)$
	MA proofs of proximity vs. standard testers	quadratically stronger	exponentially stronger
	Probabilistic (MA) vs. deterministic (NP) verification	nearly equivalent	NP proofs of proximity are extremely weak
	Hardest property for non-interactive proofs	explicit and natural; no better than standard testers, regardless of proof length	non-explicit (random property); linear length proof is required to outperform standard testers
interactive proofs	**Private vs. public coin protocols**	**Different!**	almost equivalent
	AM round hierarchy coin protocols	**Different!**	there is a property for which the AM complexity is $\approx n^{1/r}$ for r-round protocols

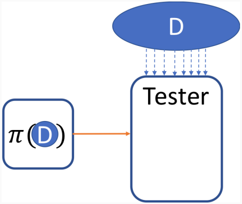
INTERACTION: SKY IS THE LIMIT...

Before (MA/NP)



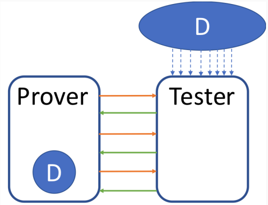
REPLACING PROOF BY PROVER

Before (MA/NP)

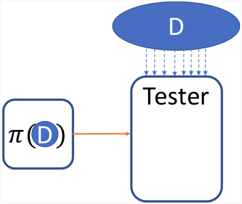


Max of proof and sample complexity can only be quadratically better

Now (IP)

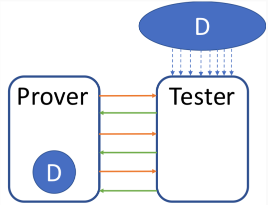


Before (MA/NP)



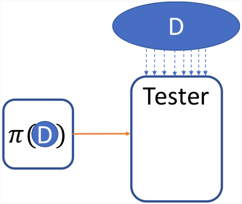
Max of proof and sample complexity can only be quadratically better

Now (IP)



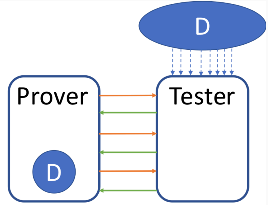
Both communication and sample complexity can be exponentially better

Before (MA/NP)



Max of proof and sample complexity can only be quadratically better

Now (IP)



Both communication and sample complexity can be exponentially better

Using 1 round of interaction!

HOW CAN INTERACTION HELP?

Consider the isolated elements property:

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \ : \ \forall i \in [n] \ i \notin \mathrm{supp}(D) \text{ or } (i+1) \notin \mathrm{supp}(D)\}$$

- We show that Π_{Isolated} requires $\Omega(\sqrt{n})$ samples for standard testers (via reduction from SMP communication complexity [BCG17])

HOW CAN INTERACTION HELP?

Consider the isolated elements property:

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \ : \ \forall i \in [n] \ i \notin \text{supp}(D) \text{ or } (i+1) \notin \text{supp}(D)\}$$

- We show that Π_{Isolated} requires $\Omega(\sqrt{n})$ samples for standard testers (via reduction from SMP communication complexity [BCG17])
- By the lifting lemma, every MA distribution tester has $\text{proof} \cdot \text{sample} = \Omega(\sqrt{n})$

HOW CAN INTERACTION HELP?

Consider the isolated elements property:

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \;:\; \forall i \in [n]\ i \notin \text{supp}(D) \text{ or } (i+1) \notin \text{supp}(D)\}$$

- We show that Π_{Isolated} requires $\Omega(\sqrt{n})$ samples for standard testers (via reduction from SMP communication complexity [BCG17])
- By the lifting lemma, every MA distribution tester has $\text{proof} \cdot \text{sample} = \Omega(\sqrt{n})$
- In fact, by a more involved lifting lemma, this also holds for public-coin IP, regardless of #rounds

Consider the isolated elements property:

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \ : \ \forall i \in [n] \ i \notin \text{supp}(D) \text{ or } (i+1) \notin \text{supp}(D)\}$$

- We show that Π_{Isolated} requires $\Omega(\sqrt{n})$ samples for standard testers (via reduction from SMP communication complexity [BCG17])
- By the lifting lemma, every MA distribution tester has $\text{proof} \cdot \text{sample} = \Omega(\sqrt{n})$
- In fact, by a more involved lifting lemma, this also holds for public-coin IP, regardless of #rounds

HOW CAN INTERACTION HELP?

Consider the isolated elements property:

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \ : \ \forall i \in [n] \ i \notin \text{supp}(D) \text{ or } (i+1) \notin \text{supp}(D)\}$$

- We show that Π_{Isolated} requires $\Omega(\sqrt{n})$ samples for standard testers (via reduction from SMP communication complexity [BCG17])
- By the lifting lemma, every MA distribution tester has $\text{proof} \cdot \text{sample} = \Omega(\sqrt{n})$
- In fact, by a more involved lifting lemma, this also holds for public-coin IP, regardless of #rounds

For (private-coin) IP, we can do exponentially better!

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \ : \ \forall i \in [n] \ i \notin \text{supp}(D) \text{ or } (i+1) \notin \text{supp}(D)\}$$

1. Tester:

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \ : \ \forall i \in [n] \ i \notin \text{supp}(D) \text{ or } (i+1) \notin \text{supp}(D)\}$$

1. Tester:
    1.1 draw samples $S = \{s_1, \ldots, s_{O(1/eps)}\}$ samples from D

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \ : \ \forall i \in [n] \ i \notin \text{supp}(D) \text{ or } (i+1) \notin \text{supp}(D)\}$$

1. Tester:
   1.1 draw samples $S = \{s_1, \ldots, s_{O(1/eps)}\}$ samples from D
   1.2 mask S by shifting each $s \in S$ to $s + 1$ w.p. $1/2$

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \;:\; \forall i \in [n] \; i \notin \text{supp}(D) \text{ or } (i+1) \notin \text{supp}(D)\}$$

1. Tester:
   1.1 draw samples $S = \{s_1, \ldots, s_{O(1/eps)}\}$ samples from D
   1.2 mask S by shifting each $s \in S$ to $s + 1$ w.p. 1/2
   1.3 send the masked samples $M = \mathrm{Mask}(S)$

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \ : \ \forall i \in [n] \ i \notin \text{supp}(D) \text{ or } (i+1) \notin \text{supp}(D)\}$$

1. Tester:
   1.1 draw samples $S = \{s_1, \ldots, s_{O(1/eps)}\}$ samples from D
   1.2 mask S by shifting each $s \in S$ to $s + 1$ w.p. 1/2
   1.3 send the masked samples $M = \mathrm{Mask}(S)$
2. Prover: unshift the samples, and send the purported preimage $\tilde{S} = \mathrm{Mask}^{-1}(M)$

$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \; : \; \forall i \in [n] \; i \notin \text{supp}(D) \text{ or } (i+1) \notin \text{supp}(D)\}$

1. **Tester:**
   1.1 draw samples $S = \{s_1, \ldots, s_{O(1/eps)}\}$ samples from D
   1.2 **mask** S by shifting each $s \in S$ to $s+1$ w.p. $1/2$
   1.3 send the masked samples $M = \mathrm{Mask}(S)$
2. **Prover:** unshift the samples, and send the purported preimage $\tilde{S} = \mathrm{Mask}^{-1}(M)$
3. **Tester:** check that the prover unmasked correctly: $\tilde{S} = S$

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \ : \ \forall i \in [n] \ i \notin \text{supp}(D) \text{ or } (i+1) \notin \text{supp}(D)\}$$

1. **Tester:**
   1.1 draw samples $S = \{s_1, \ldots, s_{O(1/eps)}\}$ samples from D
   1.2 **mask** S by shifting each $s \in S$ to $s + 1$ w.p. 1/2
   1.3 send the masked samples $M = \mathrm{Mask}(S)$
2. **Prover:** unshift the samples, and send the purported preimage $\tilde{S} = \mathrm{Mask}^{-1}(M)$
3. **Tester:** check that the prover unmasked correctly: $\tilde{S} = S$

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \; : \; \forall i \in [n] \; i \notin \text{supp}(D) \text{ or } (i+1) \notin \text{supp}(D)\}$$

1. **Tester:**
   1.1 draw samples $S = \{s_1, \ldots, s_{O(1/eps)}\}$ samples from D
   1.2 **mask** S by shifting each $s \in S$ to $s+1$ w.p. 1/2
   1.3 send the masked samples $M = \mathrm{Mask}(S)$
2. **Prover:** unshift the samples, and send the purported preimage $\tilde{S} = \mathrm{Mask}^{-1}(M)$
3. **Tester:** check that the prover unmasked correctly: $\tilde{S} = S$

Sample complexity: $O(1/\varepsilon)$

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \; : \; \forall i \in [n] \; i \notin \text{supp}(D) \text{ or } (i+1) \notin \text{supp}(D)\}$$

1. **Tester:**
    1.1 draw samples $S = \{s_1, \ldots, s_{O(1/eps)}\}$ samples from D
    1.2 **mask** S by shifting each $s \in S$ to $s+1$ w.p. 1/2
    1.3 send the masked samples $M = \mathrm{Mask}(S)$
2. **Prover:** unshift the samples, and send the purported preimage $\tilde{S} = \mathrm{Mask}^{-1}(M)$
3. **Tester:** check that the prover unmasked correctly: $\tilde{S} = S$

Sample complexity: $O(1/\varepsilon)$ Communication complexity: $O(\log(n)/\varepsilon)$

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \ : \ \forall i \in [n] \ i \notin \text{supp}(D) \text{ or } (i+1) \notin \text{supp}(D)\}$$

Tester:
* draw $S = \{s_1, \ldots, s_{O(1/\text{eps})}\}$ from D
* shift each $s \in S$ to $s + 1$ w.p. 1/2
* send masked samples $M = \mathrm{Mask}(S)$

Prover: send alleged $\tilde{S} = \mathrm{Mask}^{-1}(M)$

Tester: check that $\tilde{S} = S$

Why does this work?

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \;:\; \forall i \in [n]\; i \notin \text{supp}(D) \text{ or } (i+1) \notin \text{supp}(D)\}$$

Tester:
* draw $S = \{s_1, \ldots, s_{O(1/eps)}\}$ from D
* shift each $s \in S$ to $s + 1$ w.p. 1/2
* send masked samples $M = \mathrm{Mask}(S)$

Prover: send alleged $\tilde{S} = \mathrm{Mask}^{-1}(M)$

Tester: check that $\tilde{S} = S$

Why does this work?

If the elements of D are isolated

$$\Pi_{\mathrm{Isolated}} = \{D \in \Delta([n]) \ : \ \forall i \in [n] \ i \notin \mathrm{supp}(D) \text{ or } (i+1) \notin \mathrm{supp}(D)\}$$

Tester:
* draw $S = \{s_1, \ldots, s_{O(1/eps)}\}$ from D
* shift each $s \in S$ to $s + 1$ w.p. 1/2
* send masked samples $M = \mathrm{Mask}(S)$

Prover: send alleged $\tilde{S} = \mathrm{Mask}^{-1}(M)$

Tester: check that $\tilde{S} = S$

Why does this work?

If the elements of D are isolated, the mask is invertible

If D is ε-far from isolated

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \;:\; \forall i \in [n]\ i \notin \mathrm{supp}(D) \text{ or } (i+1) \notin \mathrm{supp}(D)\}$$

Tester:
* draw $S = \{s_1, \ldots, s_{O(1/eps)}\}$ from D
* shift each $s \in S$ to $s+1$ w.p. $1/2$
* send masked samples $M = \mathrm{Mask}(S)$

Prover: send alleged $\tilde{S} = \mathrm{Mask}^{-1}(M)$

Tester: check that $\tilde{S} = S$

Why does this work?

If the elements of D are isolated, the mask is invertible

If D is ε-far from isolated, $\exists$ adjacent supported elements of weight $\Omega(\varepsilon)$

$$\Pi_{\text{Isolated}} = \{D \in \Delta([n]) \;:\; \forall i \in [n]\; i \notin \mathrm{supp}(D) \text{ or } (i+1) \notin \mathrm{supp}(D)\}$$

Tester:
* draw $S = \{s_1, \ldots, s_{O(1/\text{eps})}\}$ from D
* shift each $s \in S$ to $s + 1$ w.p. 1/2
* send masked samples $M = \mathrm{Mask}(S)$

Prover: send alleged $\tilde{S} = \mathrm{Mask}^{-1}(M)$

Tester: check that $\tilde{S} = S$

Why does this work?

If the elements of D are isolated, the mask is invertible

If D is ε-far from isolated, $\exists$ adjacent supported elements of weight $\Omega(\varepsilon)$

Prover has to guess their preimage!

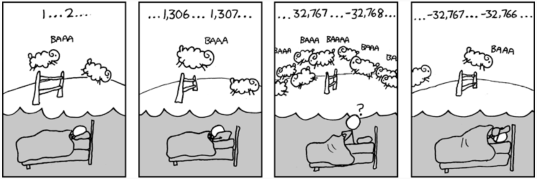
		Testing Distributions this work	**Testing Functions** [RVW13; GR16; FGL14; GR17]
non-interactive proofs	**Proofs of linear length**	reduce sample complexity of *any* property to $O(\sqrt{\mathsf{n}})$	reduce sample complexity of *any* property to $O(1)$
	MA proofs of proximity vs. standard testers	quadratically stronger	exponentially stronger
	Probabilistic (MA) vs. deterministic (NP) verification	nearly equivalent	NP proofs of proximity are extremely weak
	Hardest property for non-interactive proofs	explicit and natural; no better than standard testers, regardless of proof length	non-explicit (random property); linear length proof is required to outperform standard testers
interactive proofs	**Private vs. public coin protocols**	exponential separation	almost equivalent
	AM round hierarchy coin protocols	AM complexity is quadratically related to the sample complexity of standard testers	there is a property for which the AM complexity is $\approx \mathsf{n}^{1/\mathsf{r}}$ for r-round protocols

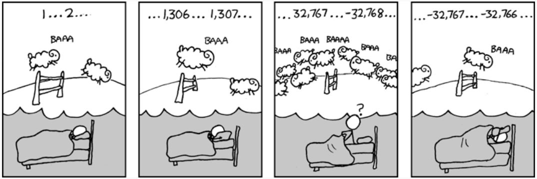
FUNCTIONS VS DISTRIBUTIONS

		Testing Distributions this work	**Testing Functions** [RVW13; GR16; FGL14; GR17]
non-interactive proofs	**Proofs of linear length**	reduce sample complexity of *any* property to $O(\sqrt{\mathsf{n}})$	reduce sample complexity of *any* property to $O(1)$
	MA proofs of proximity vs. standard testers	quadratically stronger	exponentially stronger
	Probabilistic (MA) vs. deterministic (NP) verification	nearly equivalent	NP proofs of proximity are extremely weak
	Hardest property for non-interactive proofs	explicit and natural; no better than standard testers, regardless of proof length	non-explicit (random property); linear length proof is required to outperform standard testers
interactive proofs	**Private vs. public coin protocols**	exponential separation	almost equivalent
	AM round hierarchy coin protocols	AM complexity is quadratically related to the sample complexity of standard testers	there is a property for which the AM complexity is $\approx \mathsf{n}^{1/\mathsf{r}}$ for r-round protocols

OPEN PROBLEMS

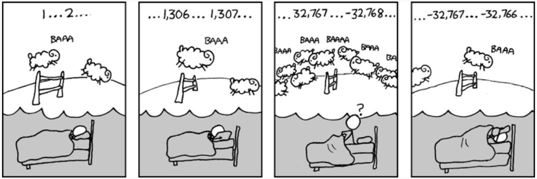
OPEN PROBLEMS





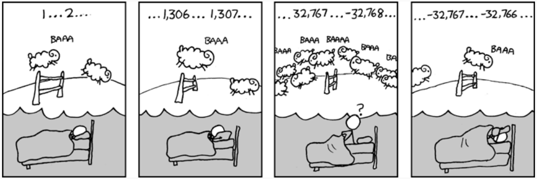
There are many of them…let's focus on one:

OPEN PROBLEMS



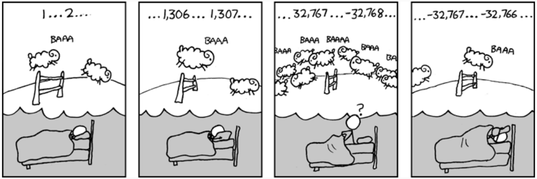
There are many of them...let's focus on one:

1. For MA distribution testers we have $\mathrm{proof} \cdot \mathrm{sample} = \Omega(\mathrm{SAMP}(\Pi))$



There are many of them...let's focus on one:

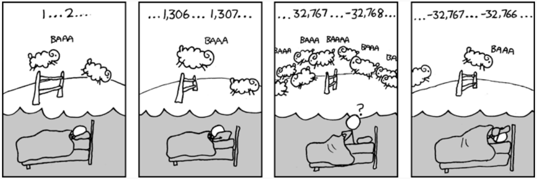
1. For MA distribution testers we have $\text{proof} \cdot \text{sample} = \Omega(\mathrm{SAMP}(\Pi))$
2. For r-round AM distribution testers we have



There are many of them...let's focus on one:

1. For MA distribution testers we have
   $\text{proof} \cdot \text{sample} = \Omega(\text{SAMP}(\Pi))$
2. For r-round AM distribution testers we have

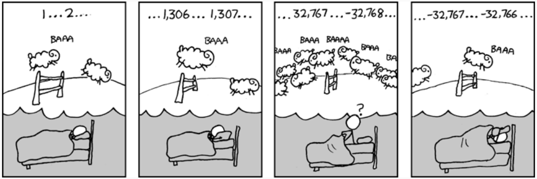
OPEN PROBLEMS



There are many of them...let's focus on one:

1. For MA distribution testers we have $\text{proof} \cdot \text{sample} = \Omega(\text{SAMP}(\Pi))$
2. For r-round AM distribution testers we have ...well $\text{communication} \cdot \text{sample} = \Omega(\text{SAMP}(\Pi))$!

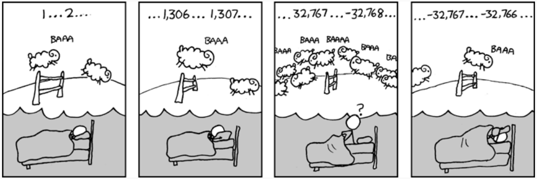
OPEN PROBLEMS



There are many of them...let's focus on one:

1. For MA distribution testers we have
   $\mathrm{proof} \cdot \mathrm{sample} = \Omega(\mathrm{SAMP}(\Pi))$
2. For r-round AM distribution testers we have
   ...well $\mathrm{communication} \cdot \mathrm{sample} = \Omega(\mathrm{SAMP}(\Pi))$!
3. But often we can get AM, but do not know how to get MA...

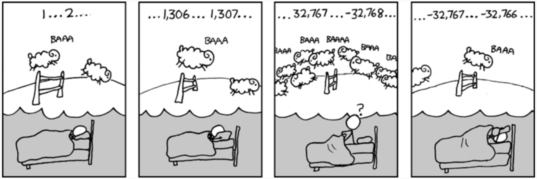
OPEN PROBLEMS



There are many of them...let's focus on one:

1. For MA distribution testers we have $\mathrm{proof} \cdot \mathrm{sample} = \Omega(\mathrm{SAMP}(\Pi))$
2. For r-round AM distribution testers we have ...well $\mathrm{communication} \cdot \mathrm{sample} = \Omega(\mathrm{SAMP}(\Pi))$!
3. But often we can get AM, but do not know how to get MA...

OPEN PROBLEMS



There are many of them...let's focus on one:

1. For MA distribution testers we have $\text{proof} \cdot \text{sample} = \Omega(\text{SAMP}(\Pi))$
2. For r-round AM distribution testers we have ...well $\text{communication} \cdot \text{sample} = \Omega(\text{SAMP}(\Pi))$!
3. But often we can get AM, but do not know how to get MA...

Is AM strictly stronger than MA?

Thank you!

Noga Alon, Michael Krivelevich, Ilan Newman, and Mario Szegedy.
Regular languages are testable with a constant number of queries.
SIAM Journal on Computing, 30(6):1842–1862, 2001.

Eric Blais, Clément L. Canonne, and Tom Gur.
Distribution testing lower bounds via reductions from communication complexity (Alice and Bob don't talk to each other anymore.).
In Proceedings of the 32th Conference on Computational Complexity, CCC 2017, pages 1–42, 2017.

Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil P. Vadhan.
Robust PCPs of proximity, shorter PCPs, and applications to coding.
SIAM Journal on Computing, 36(4):889–974, 2006.

Funda Ergün, Ravi Kumar, and Ronitt Rubinfeld.
Fast approximate probabilistically checkable proofs.
Information and Computation, 189(2):135–159, 2004.

Eldar Fischer, Yonatan Goldhirsh, and Oded Lachish.
Partial tests, universal tests and decomposability.
In Proceedings of the 5th Innovations in Theoretical Computer Science Conference, ITCS 2014, pages 483–500, 2014.

Guy N. Rothblum, Salil P. Vadhan, and Avi Wigderson.
Interactive proofs of proximity: delegating computation in sublinear time.
In Proceedings of the 45th Symposium on Theory of Computing, STOC 2013, pages 793–802, 2013.

Paul Valiant.
Testing symmetric properties of distributions.
SIAM Journal on Computing, 40:1927–1968, 2011.

John Von Neumann.
Various techniques used in connection with random digits.
National Bureau of Standards Applied Math Series, 12:36–38, 1951.

Gregory Valiant and Paul Valiant.
An automatic inequality prover and instance optimal identity testing.
SIAM Journal on Computing, 46(1):429–455, 2017.