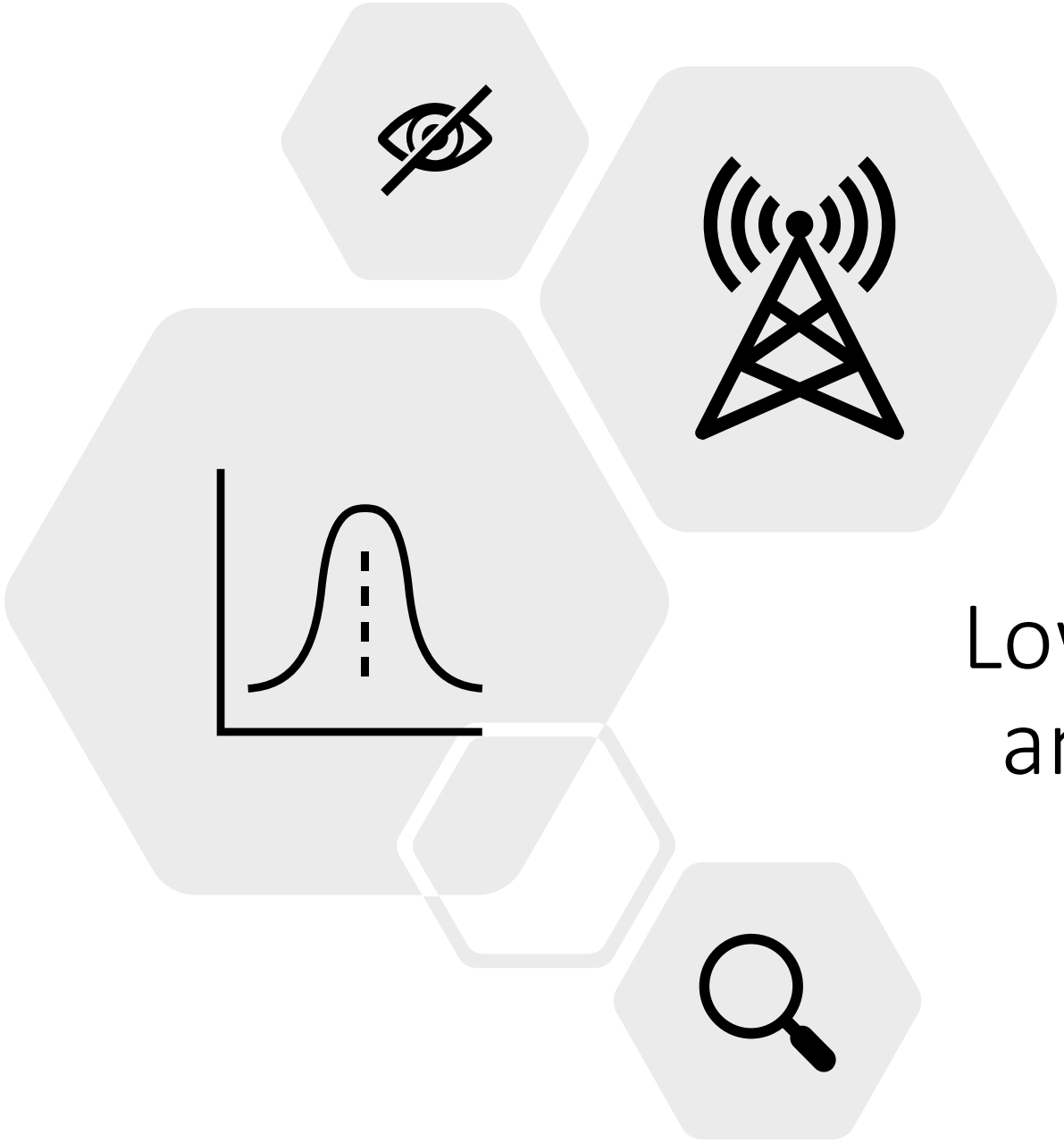


Coogee'22 — Sydney Quantum Information Theory Workshop

Clément Canonne (University of Sydney)

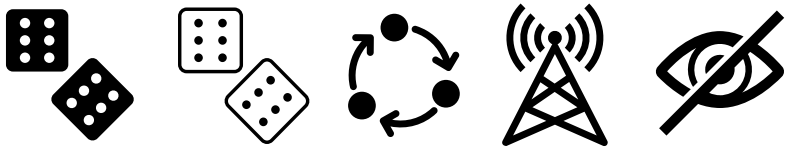
Lower bounds for estimation and testing under restricted measurements

Contents of this talk

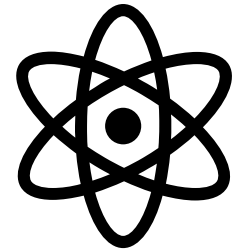
1. What are learning and testing?
2. Baseline: the "centralised" setting
3. Beyond the centralised setting: 3 flavours
4. Restricted measurements?
   - Two guiding examples: communication and privacy

Contents of this talk

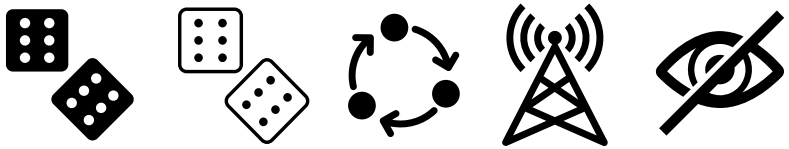
1. What are learning and testing?
2. Baseline: the "centralised" setting
3. Beyond the centralised setting: 3 flavours
4. Restricted measurements?
   - Two guiding examples: communication and privacy



Contents of this talk



1. What are learning and testing?
2. Baseline: the "centralised" setting
3. Beyond the centralised setting: 3 flavours
4. Restricted measurements?
   - Two guiding examples: communication and privacy
   - And, what about quantum?



What are learning and testing?

Standard statistical setting: n iid samples from some unknown probability distribution p

What are learning and testing?

Standard statistical setting: n iid samples from some unknown probability distribution p

Goal: estimate something about p

What are learning and testing?

Standard statistical setting: n iid samples from some unknown probability distribution p

Goal: estimate something about p

→ learn p: output $\hat{p}$ such that

$$\mathbb{E}[\ell(\hat{p}, p)] \leq \varepsilon$$

What are learning and testing?

Standard statistical setting: n iid samples from some unknown probability distribution p

Goal: estimate something about p

→ learn p : output $\hat{p}$ such that

depends on the n samples

loss function

target rate

$$\mathbb{E}_{p}\left[\ell(\hat{p}, p)\right] \leq \varepsilon$$

What are learning and testing?

Examples of loss functions:

- $KL(p \| q) = -\sum_x p(x) \log \frac{q(x)}{p(x)}$
- $\ell_2^2(p,q) = \sum_x (p(x) - q(x))^2$
- $\chi^2(p,q) = \sum_x \frac{(p(x) - q(x))^2}{q(x)}$
- $TV(p,q) = \sup_S (p(S) - q(S)) = \frac{1}{2} \sum_x |p(x) - q(x)|$

What are learning and testing?

Examples of loss functions:

- $KL(p \| q) = -\sum_x p(x) \log \frac{q(x)}{p(x)}$
- $\ell_2^2(p,q) = \sum_x (p(x) - q(x))^2$
- $\chi^2(p,q) = \sum_x \frac{(p(x) - q(x))^2}{q(x)}$
- $TV(p,q) = \sup_S (p(S) - q(S)) = \frac{1}{2} \sum_x |p(x) - q(x)|$

What are ~~learning~~ estimating and testing?

Standard statistical setting: n iid samples from some unknown probability distribution p

Goal: estimate something about p

What are ~~learning~~ estimating and testing?

Standard statistical setting: n iid samples from some unknown probability distribution p

Goal: estimate something about p

↳ learn a parameter / functional θ of p

output $\hat{\theta}$ such that

$$\mathop{\mathbb{E}}_{p}\left[\ell(\hat{\theta}, \theta(p))\right] \leq \varepsilon$$

What are ~~learning~~ estimating and testing?

Standard statistical setting: n iid samples from some unknown probability distribution p

Goal: estimate something about p

↳ learn a parameter / functional θ of p

output $\hat{\theta}$ such that

$$\mathbb{E}_p\left[\ell(\hat{\theta}, \theta(p))\right] \leq \varepsilon$$

for instance, the mean of p

What are learning and testing?

Standard statistical setting: n iid samples from some unknown probability distribution p

Goal: estimate something about p

↳ is p what I thought it was?

What are learning and testing?

Standard statistical setting: n iid samples from some unknown probability distribution p

Goal: estimate something about p

↳ Hypothesis: $\mathcal{H}_0 =$ "$p = q$" (null)

$\mathcal{H}_1 =$ "$TV(p,q) > \varepsilon$" (altern.)

Output $\hat{b} \in \{0,1\}$ st. $\mathbb{P}_q\{\hat{b}=1\} + \sup_{p \in \mathcal{H}_1} \mathbb{P}\{\hat{b}=0\} \leq \frac{1}{10}$

What are learning and testing?

Standard statistical setting: n iid samples from some unknown probability distribution p

Goal: estimate something about p

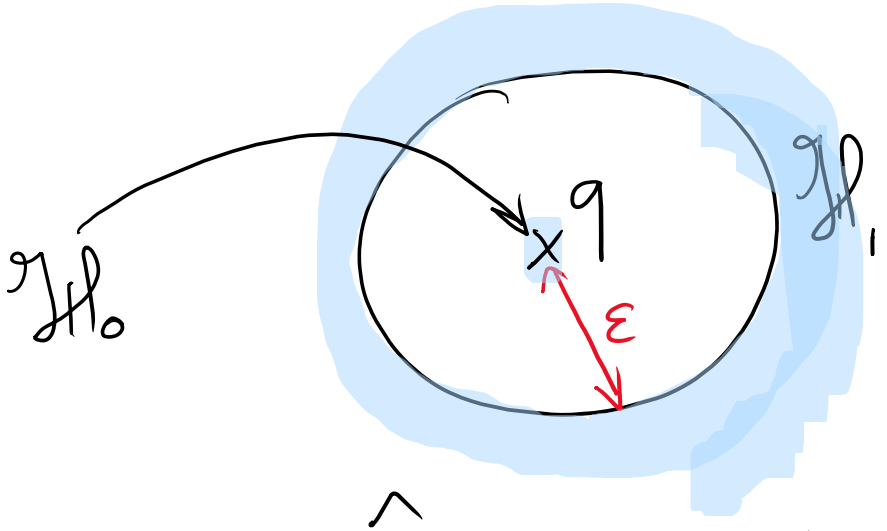
↳ Hypothesis: $\mathcal{H}_0 =$ "$p = q$" (null)

$\mathcal{H}_1 =$ "$TV(p,q) > \varepsilon$" (altern.)

Output $\hat{b} \in \{0,1\}$ st. $\mathbb{P}_q\{\hat{b}=1\} + \sup_{p \in \mathcal{H}_1} \mathbb{P}_p\{\hat{b}=0\} \leq \frac{1}{10}$

What are learning and testing?

Goal: estimate something about p



$\mathcal{H}_0 =$ "$p = q$" (null)

$\mathcal{H}_1 =$ "$TV(p,q) > \varepsilon$" (altern.)

Output $\hat{b} \in \{0,1\}$ s.t. $\mathbb{P}_q\{\hat{b}=1\} + \sup_{p \in \mathcal{H}_1} \mathbb{P}_p\{\hat{b}=0\} \leq \frac{1}{10}$

TYPE I

TYPE II

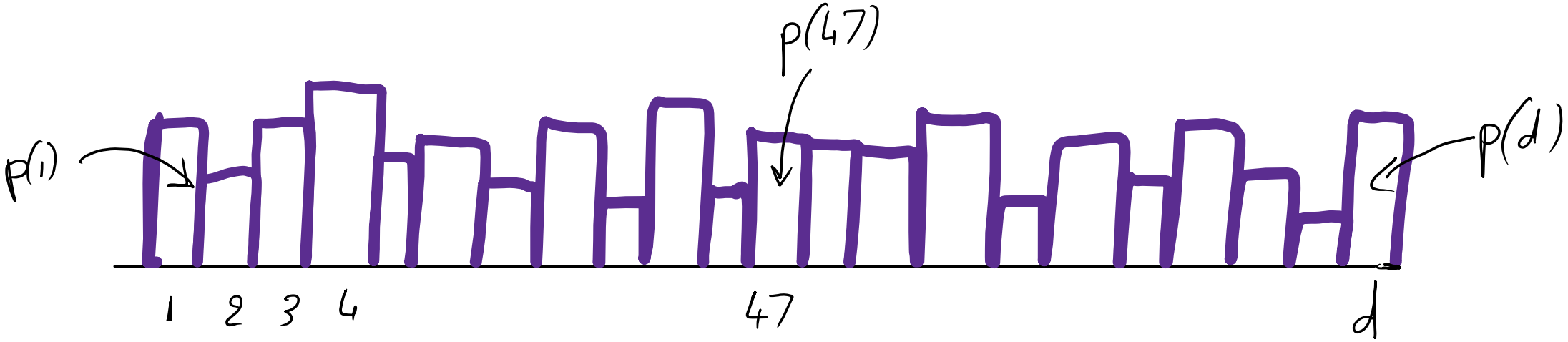
$\approx$ arbitrary

Now, what are we learning and testing?

Now, what are we learning and testing?

① Discrete distributions over d elements: $[d] := \{1, 2, \dots, d\}$

→ Learning under TV loss → Testing if uniform on $[d]$



Now, what are we learning and testing?

② High-dimensional Gaussians (with identity covariance)

↑ dimension d

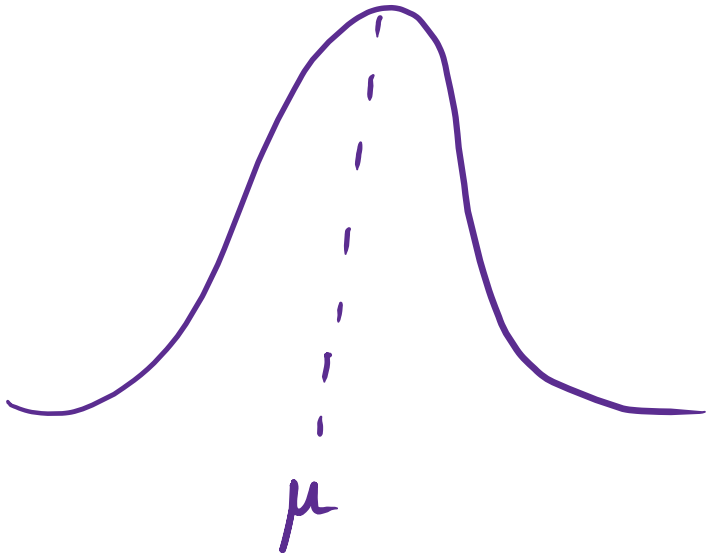
Now, what are we learning and testing?

② High-dimensional Gaussians (with identity covariance)

↑ dimension d

Learning the mean under ℓ_2^2 loss

Testing if the mean is zero (also ℓ_2)



$p = N(\mu, I_d)$

$\mu \in \mathbb{R}^d$

Baseline: the "centralised" setting

$X_1, X_2, \ldots, X_n \sim p$ fully accessible to the algorithm.

How large must n be to solve the learning or testing question?

Baseline: the "centralised" setting

$X_1, X_2, \ldots, X_n \sim p$ fully accessible to the algorithm.

How large must n be to solve the learning or testing question? (as a function of d, ε)

"Minimax sample complexity"

Baseline: the "centralised" setting

Discrete distributions

$d \gg 1$

$\varepsilon \in (0, 1]$

Theorem. Learning an arbitrary p over $[d]$ to TV loss ε has sample complexity ____.

Theorem. Testing if an arbitrary p over $[d]$ is u (uniform over $[d]$) or has $TV(p,u) > \varepsilon$ has sample complexity ____.

Baseline: the "centralised" setting

$d \gg 1$

$\varepsilon \in (0, 1]$

Discrete distributions

Theorem. Learning an arbitrary p over $[d]$ to TV loss ε has sample complexity $\Theta\left(\frac{d}{\varepsilon^2}\right)$.

Theorem. Testing if an arbitrary p over $[d]$ is u or has $TV(p,u) > \varepsilon$ has sample complexity ____ .

Baseline: the "centralised" setting

$d \gg 1$

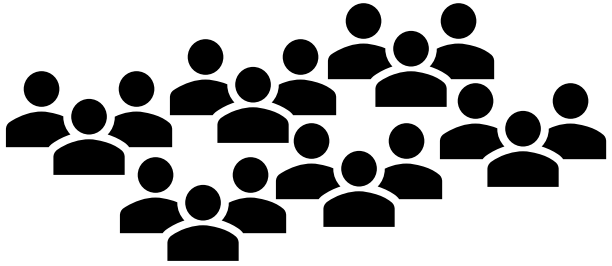
$\varepsilon \in (0, 1]$

Discrete distributions

Theorem. Learning an arbitrary p over $[d]$ to TV loss ε has sample complexity $\Theta\left(\frac{d}{\varepsilon^2}\right)$.

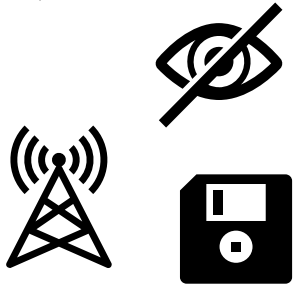
Theorem. Testing if an arbitrary p over $[d]$ is u or has $TV(p,u) > \varepsilon$ has sample complexity $\Theta\left(\frac{\sqrt{d}}{\varepsilon^2}\right)$.

Beyond the centralised setting



Distributed data

Information or computational constraints



Limited types of measurements

Beyond the centralised setting

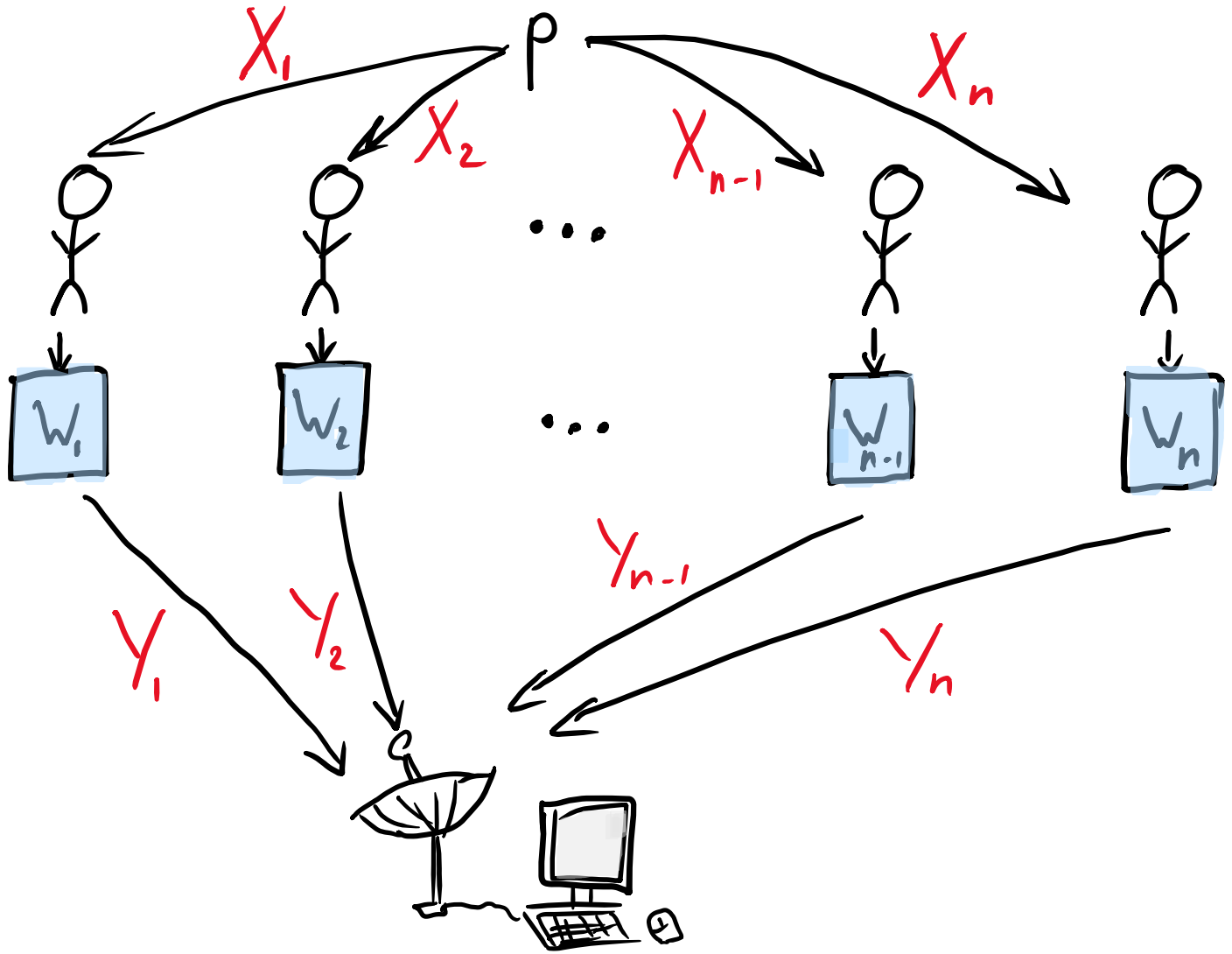
- n users, each holding one sample from (same) p
- One center, which has no sample but needs to solve the learning/testing task
- Each user can only send a "limited" type of message

Beyond the centralised setting

- n users, each holding one sample from (same) p
- One center, which has no sample but needs to solve the learning/testing task
- Each user can only send a "limited" type of message

"local" constraint

Beyond the centralised setting



Channels
$W_1, \ldots, W_n \in \mathcal{W}$

Beyond the centralised setting

Channel: $W: \mathcal{X} \to \mathcal{Y}$ randomised

$\mathcal{X}$: input space (samples)

$\mathcal{Y}$: output space (messages)

Notation: $W(y|x) = \mathbb{P}\{W(x) = y\}$

Beyond the centralised setting

Channel: $W: \mathcal{X} \to \mathcal{Y}$ randomised

$\mathcal{X}$: input space (samples)

$\mathcal{Y}$: output space (messages)

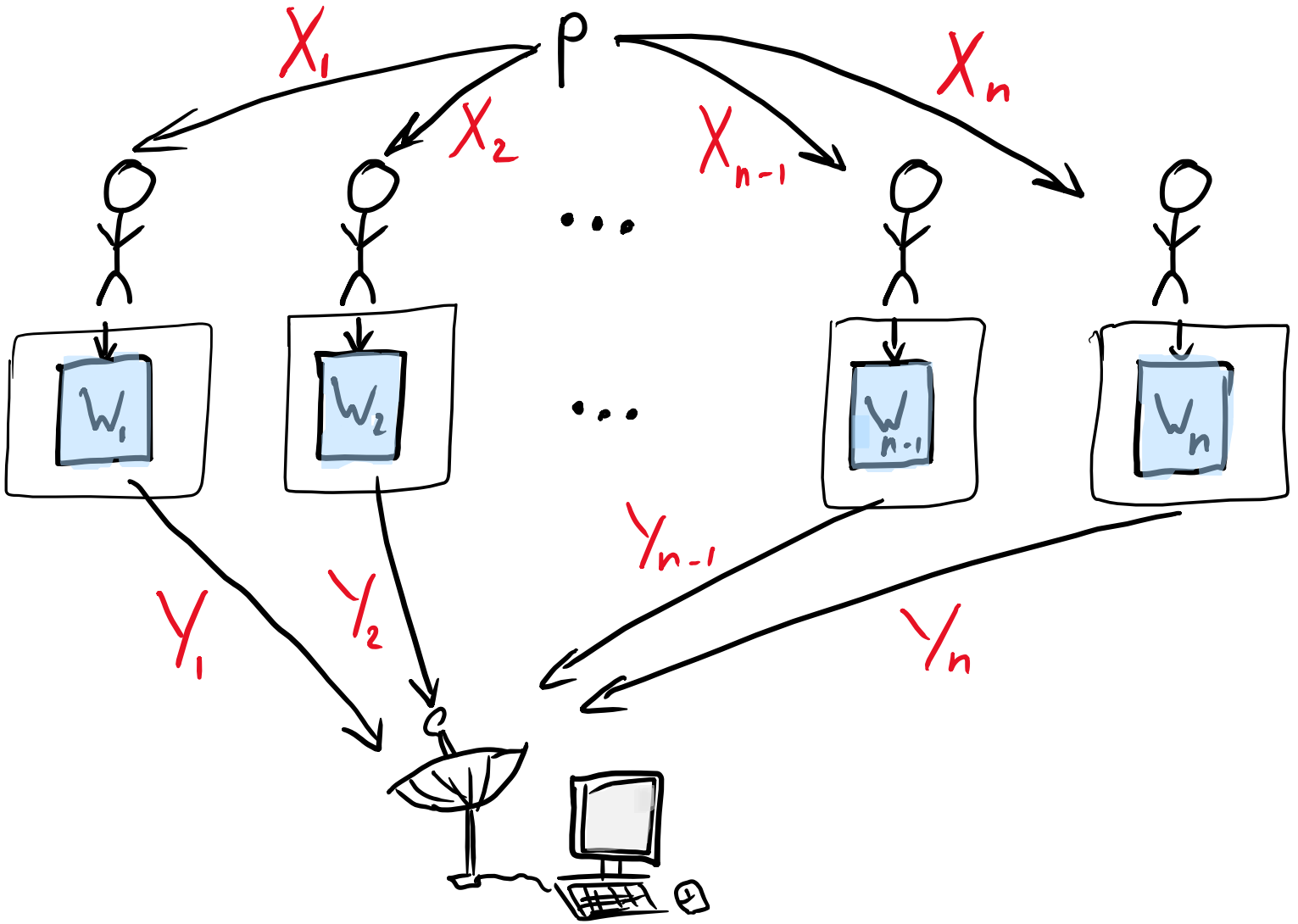
Notation: $W(y|x) = \mathbb{P}\{W(x) = y\}$

$\mathcal{W} \subseteq \{\mathcal{X} \to \mathcal{Y}\}$ set of allowed channels

What happens if W contains the identity mapping?

Beyond the centralised setting

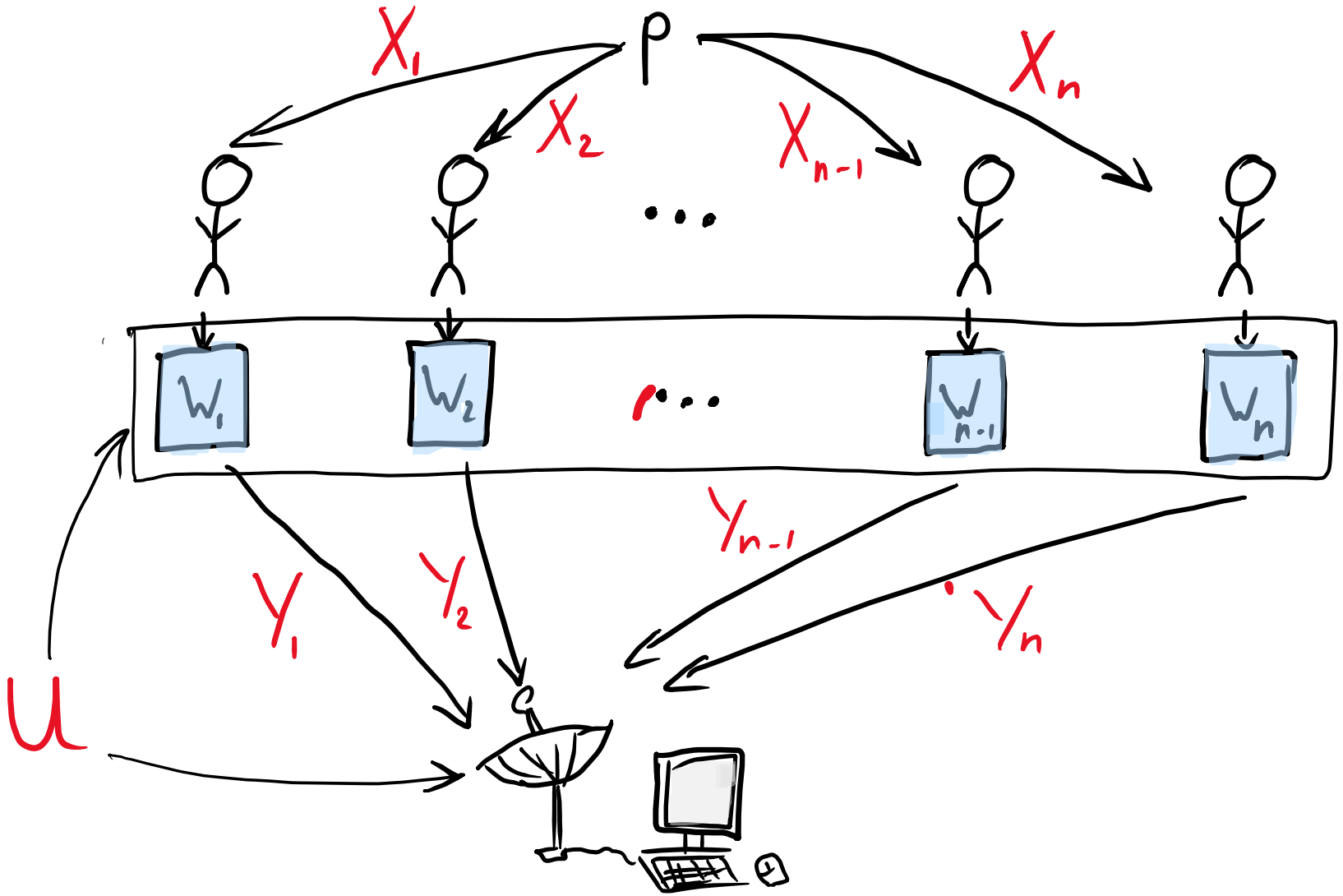
Private-coin



Channels $W_1, \ldots, W_n \in \mathcal{W}$ independently randomised

Beyond the centralised setting

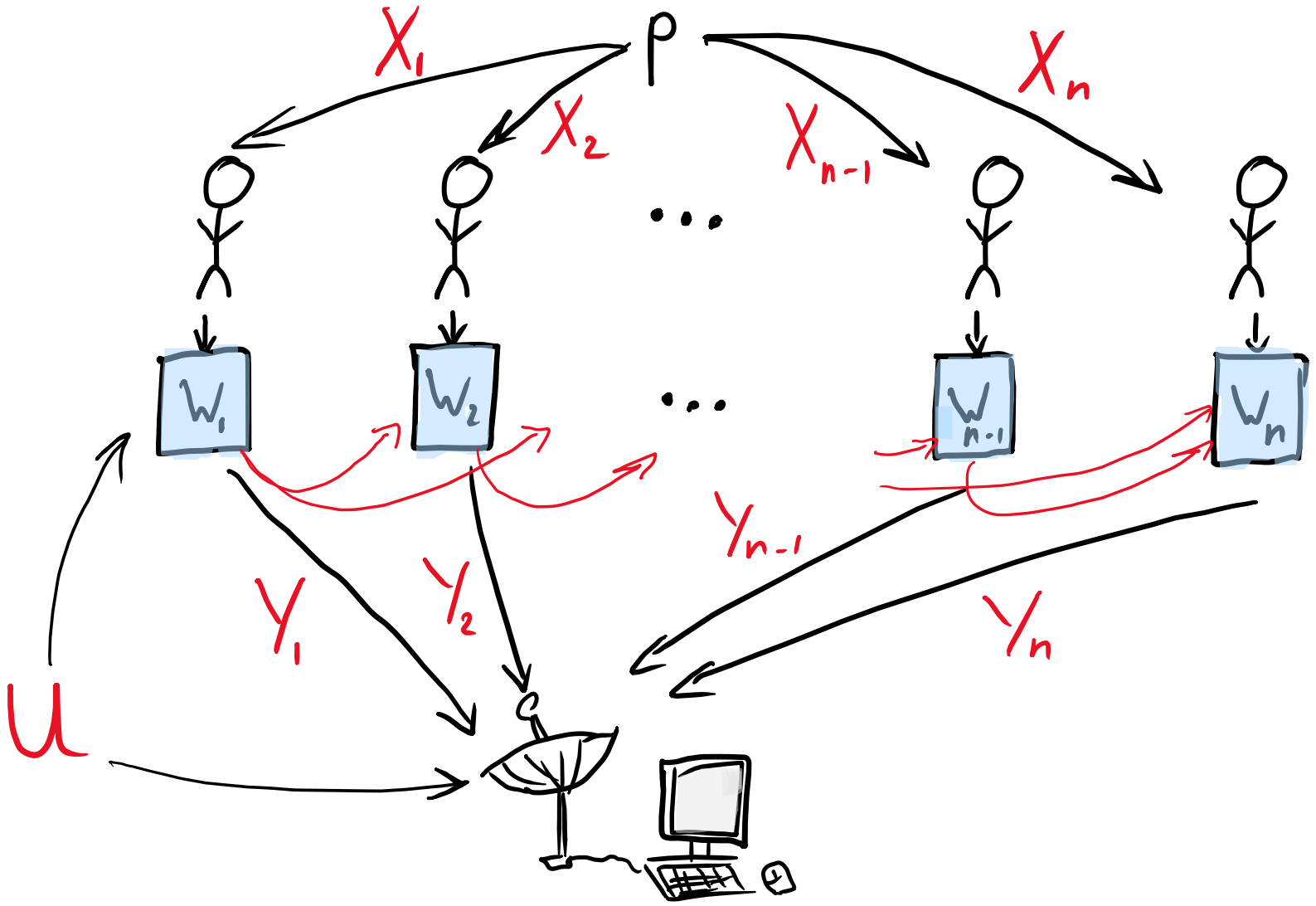
Public-coin



Channels $W_1, \ldots, W_n \in \mathcal{W}$ - jointly randomised

Beyond the centralised setting

Interactive



Channels
$W_1, \dots, W_n \in \mathcal{W}$

$W_t = W^{Y_1, \dots, Y_{t-1}}$

depends on previous messages

(+ public randomness)

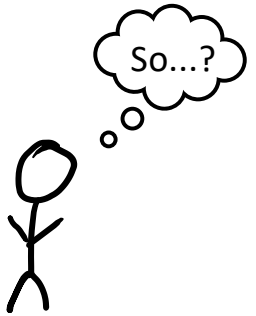
Beyond the centralised setting

Implementation and deployment

Private-coin ≤ Public-coin ≤ Interactive

Sample complexity

Private-coin ≥ Public-coin ≥ Interactive



Two guiding examples of channel families

Communication

Each user can only send ℓ bits

$$\mathcal{W}_\ell = \{ W : \mathcal{X} \to \{0,1\}^\ell \}$$

Local Privacy

Each user requires ε- differential privacy

$$\forall W \in \mathcal{W}_\varepsilon$$

$$\forall y, x, x', \quad W(y|x) \leq e^\varepsilon W(y|x')$$

Two guiding examples of channel families

Communication

Each user can only send ℓ bits

$$\mathcal{W}_\ell = \{ W : \mathcal{X} \to \{0,1\}^\ell \}$$

Local Privacy

Each user requires ε- differential privacy

$$\forall W \in \mathcal{W}_\varepsilon$$

$$\forall y, x, x', \quad W(y|x) \le e^\varepsilon W(y|x') \approx (1+\varepsilon) W(y|x')$$

(think of $\varepsilon \in (0,1]$)

Two guiding examples of channel families

Communication

Each user can only send ℓ bits

$$\mathcal{W}_\ell = \{ W : \mathcal{X} \to \{0,1\}^\ell \}$$

Can't send too much

Local Privacy

Each user requires ε- differential privacy

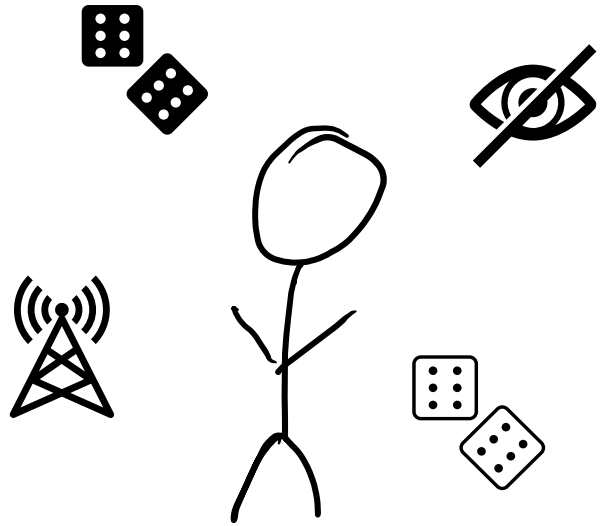
$$\forall W \in \mathcal{W}_\varepsilon$$

$$\forall y, x, x', \quad W(y|x) \leq e^\varepsilon W(y|x')$$

Can't reveal too much

Let's go:

Learning and testing discrete distributions under those information constraints



$d \gg 1$
$\varepsilon \in (0,1]$
$\ell \leq \log_2 d$
$\rho \in (0,1]$

Upper bounds

Theorem. Learning an arbitrary p over $[d]$ to TV loss ε under ℓ-bit communication constraints has sample complexity ———— .

Theorem. Learning an arbitrary p over $[d]$ to TV loss ε under ρ-local privacy (LDP) constraints has sample complexity ———— .

Upper bounds

$d \gg 1$
$\varepsilon \in (0,1]$
$\ell \leq \log_2 d$
$\rho \in (0,1]$

Theorem. Learning an arbitrary p over $[d]$ to TV loss ε under ℓ-bit communication constraints has sample complexity ———— .

Theorem. Learning an arbitrary p over $[d]$ to TV loss ε under ρ-local privacy (LDP) constraints has sample complexity ———— .

$d \gg 1$
$\varepsilon \in (0,1]$
$\ell \leq \log_2 d$
$\rho \in (0,1]$

Upper bounds

Theorem. Learning an arbitrary p over $[d]$ to TV loss ε under ℓ-bit communication constraints has sample complexity $O\left(\frac{d^2}{2^\ell \varepsilon^2}\right)$.

Theorem. Learning an arbitrary p over $[d]$ to TV loss ε under ρ-local privacy (LDP) constraints has sample complexity $O\left(\frac{d^2}{\rho^2 \varepsilon^2}\right)$.

Upper bounds

$d \gg 1$
$\varepsilon \in (0,1]$
$\ell \le \log_2 d$
$\rho \in (0,1]$

Theorem. Learning an arbitrary p over $[d]$ to TV loss ε under ℓ-bit communication constraints has sample complexity $O\left(\frac{d^2}{2^\ell \varepsilon^2}\right)$. Further, this is attained by a private-coin protocol.

Theorem. Learning an arbitrary p over $[d]$ to TV loss ε under ρ-local privacy (LDP) constraints has sample complexity $O\left(\frac{d^2}{\rho^2 \varepsilon^2}\right)$. Further, this is attained by a private-coin protocol.

Upper bounds

Recall: $\frac{d}{\varepsilon^2}$ in the centralised case.

$d \gg 1$
$\varepsilon \in (0,1]$
$\ell \le \log_2 d$
$\rho \in (0,1]$

Theorem. Learning an arbitrary p over $[d]$ to TV loss ε under ℓ-bit communication constraints has sample complexity $O\left(\frac{d^2}{2^\ell \varepsilon^2}\right)$. Further, this is attained by a private-coin protocol.

Theorem. Learning an arbitrary p over $[d]$ to TV loss ε under ρ-local privacy (LDP) constraints has sample complexity $O\left(\frac{d^2}{\rho^2 \varepsilon^2}\right)$. Further, this is attained by a private-coin protocol.

What about testing?

d? $\sqrt{d}$? d^2? $d^{2/3}$?

$d^{3/4}$? $d^{3/2}$?

Upper bounds

$d \gg 1$
$\varepsilon \in (0,1]$
$\ell \leq \log_2 d$
$\rho \in (0,1]$

Theorem. Testing if an arbitrary p over $[d]$ is u or has $TV(p,u) > \varepsilon$ under ℓ-bit communication constraints has sample complexity ———— .

Theorem. Testing if an arbitrary p over $[d]$ is u or has $TV(p,u) > \varepsilon$ under ρ-local privacy (LDP) constraints has sample complexity ———— .

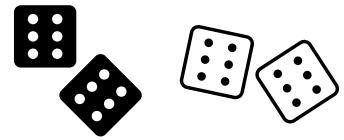
Upper bounds

$d \gg 1$
$\varepsilon \in (0,1]$
$\ell \leq \log_2 d$
$\rho \in (0,1]$

Theorem. Testing if an arbitrary p over $[d]$ is u or has $TV(p,u) > \varepsilon$ under ℓ-bit communication constraints has sample complexity $O\left(\frac{d^{3/2}}{2^{\ell}\varepsilon^2}\right)$ (private-coin)

Theorem. Testing if an arbitrary p over $[d]$ is u or has $TV(p,u) > \varepsilon$ under ρ-local privacy (LDP) constraints has sample complexity $O\left(\frac{d^{3/2}}{\rho^2\varepsilon^2}\right)$ (private-coin)

Upper bounds



$d \gg 1$
$\varepsilon \in (0,1]$
$\ell \leq \log_2 d$
$\rho \in (0,1]$

Theorem. Testing if an arbitrary p over $[d]$ is u or has $TV(p,u) > \varepsilon$ under ℓ-bit communication constraints has sample complexity $O\left(\frac{d^{3/2}}{2^{\ell}\varepsilon^2}\right)$ (private-coin) and $O\left(\frac{d}{2^{\ell/2}\varepsilon^2}\right)$ (public-coin).

Theorem. Testing if an arbitrary p over $[d]$ is u or has $TV(p,u) > \varepsilon$ under ρ-local privacy (LDP) constraints has sample complexity $O\left(\frac{d^{3/2}}{\rho^2\varepsilon^2}\right)$ (private-coin) and $O\left(\frac{d}{\rho^2\varepsilon^2}\right)$ (public-coin).

Upper bounds

Recall: $\frac{\sqrt{d}}{\varepsilon^2}$ in the centralised case.

$d \gg 1$
$\varepsilon \in (0,1]$
$\ell \leq \log_2 d$
$\rho \in (0,1]$

Theorem. Testing if an arbitrary p over $[d]$ is u or has $TV(p,u) > \varepsilon$ under ℓ-bit communication constraints has sample complexity $O\left(\frac{d^{3/2}}{2^\ell \varepsilon^2}\right)$ (private-coin) and $O\left(\frac{d}{2^{\ell/2}\varepsilon^2}\right)$ (public-coin).

Theorem. Testing if an arbitrary p over $[d]$ is u or has $TV(p,u) > \varepsilon$ under ρ-local privacy (LDP) constraints has sample complexity $O\left(\frac{d^{3/2}}{\rho^2 \varepsilon^2}\right)$ (private-coin) and $O\left(\frac{d}{\rho^2 \varepsilon^2}\right)$ (public-coin).

Lower bounds

Can we do better?

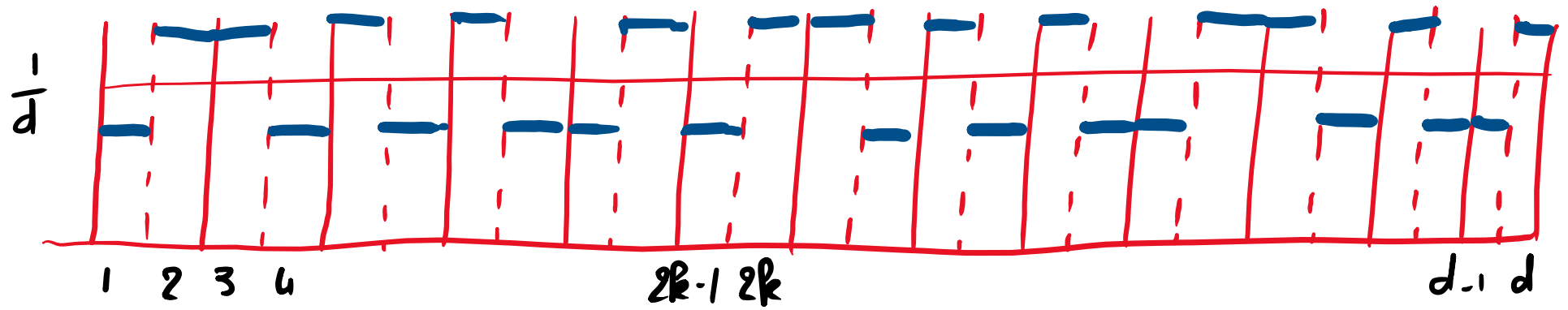
Lower bounds

Can we do better? No.

(But how to prove it?)

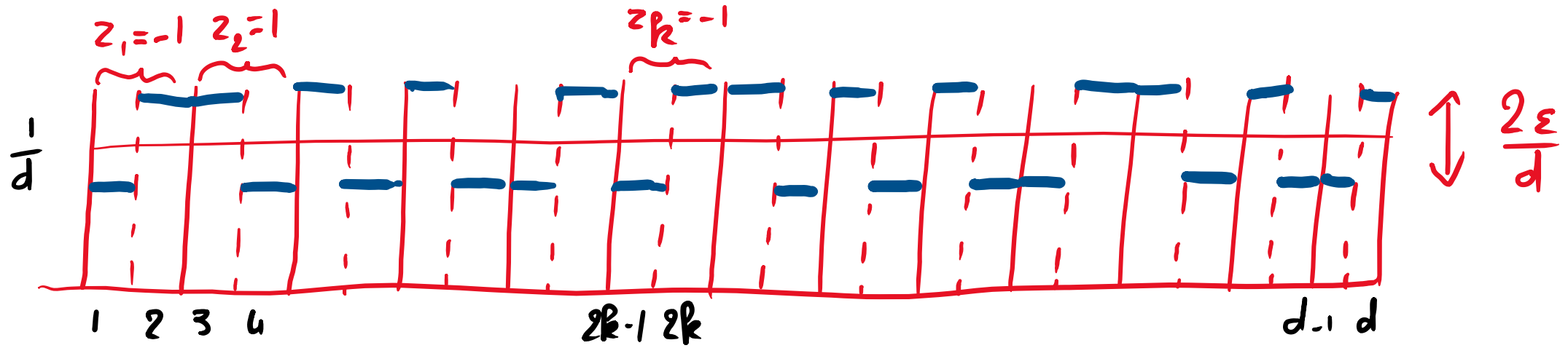
Let's start with a collection of hard instances $\mathcal{P} = \{p_z\}_{z \in \{\pm 1\}^{d/2}}$:

$$p_z = \frac{1}{d}\left(1+\varepsilon z_1, 1-\varepsilon z_1, 1+\varepsilon z_2, 1-\varepsilon z_2, \dots, 1+\varepsilon z_{\frac{d}{2}}, 1-\varepsilon z_{\frac{d}{2}}\right)$$



Let's start with a collection of hard instances $\mathcal{P} = \{p_z\}_{z \in \{\pm 1\}^{d/2}}$:

$$p_z = \frac{1}{d}\Big(\underbrace{1+\varepsilon z_1, 1-\varepsilon z_1}_{\text{sum to } 2}, \underbrace{1+\varepsilon z_2, 1-\varepsilon z_2}_{\text{sum to } 2}, \dots, \underbrace{1+\varepsilon z_{\frac{d}{2}}, 1-\varepsilon z_{\frac{d}{2}}}_{\text{sum to } 2}\Big)$$



Let's start with a collection of hard instances $\mathcal{P} = \{p_z\}_{z \in \{\pm 1\}^{d/2}}$:

$$p_z = \frac{1}{d}\left(1+2\varepsilon z_1, 1-2\varepsilon z_1, 1+2\varepsilon z_2, 1-2\varepsilon z_2, \dots, 1+2\varepsilon z_{\frac{d}{2}}, 1-2\varepsilon z_{\frac{d}{2}}\right)$$

Note that $TV(p_z, u) = \varepsilon$, and $TV(p_z, p_{z'}) = \frac{2\varepsilon}{d} \cdot Ham(z, z')$

Let's start with a collection of hard instances $\mathcal{P} = \{p_z\}_{z \in \{\pm 1\}^{d/2}}$:

$$p_z = \frac{1}{d}\left(1+2\varepsilon z_1, 1-2\varepsilon z_1, 1+2\varepsilon z_2, 1-2\varepsilon z_2, \dots, 1+2\varepsilon z_{\frac{d}{2}}, 1-2\varepsilon z_{\frac{d}{2}}\right)$$

Note that $TV(p_z, u) = \varepsilon$, and $TV(p_z, p_{z'}) = \frac{2\varepsilon}{d} \cdot Ham(z, z')$

useful for testing

useful for learning

Fix $\mathcal{W}$ (constraints). For $W \in \mathcal{W}$, $w : [d] \to \mathcal{Y}$, $X \sim p$ induces a distribution on $\mathcal{Y}$:

$$p^W(y) = \mathop{\mathbb{E}}_{X \sim p}[W(y \mid X)] \qquad \forall y \in \mathcal{Y}$$

Fix $\mathcal{W}$ (constraints). For $W \in \mathcal{W}$, $W : [d] \to \mathcal{Y}$, $X \sim p$ induces a distribution on $\mathcal{Y}$:

$$p^W(y) = \mathop{\mathbb{E}}_{X \sim p} [W(y \mid X)] \qquad \forall y \in \mathcal{Y}$$

Fix any (interactive) protocol w/ n users under constraints $\mathcal{W}$, with message space $\mathcal{Y}$.

Inputs $X_1, \dots, X_n \sim p$ (iid) $\longrightarrow$ induced distribution on $\mathcal{Y}^n$

(not a product distribution)

Fix $\mathcal{W}$ (constraints). For $W \in \mathcal{W}$, $W : [d] \to \mathcal{Y}$, $X \sim p$ induces a distribution on $\mathcal{Y}$:

$$p^W(y) = \mathop{\mathbb{E}}_{X \sim p} [W(y \mid X)] \qquad \forall y \in \mathcal{Y}$$

Fix any (interactive) protocol w/ n users under constraints $\mathcal{W}$, with message space $\mathcal{Y}$.

Inputs $X_1, \dots, X_n \sim p$ (iid) $\longrightarrow$ induced distribution on $\mathcal{Y}^n$

$p^{\mathcal{Y}^n}$ ← depends on p, and the protocol (and thus $\mathcal{W}$)

We will take a uniform prior on Z: $Z_1, -, Z_{d/2}$ iid. ± 1.

Our goal:

① Lower bound $\sum_{i=1}^{d/2} I(Z_i; Y^n)$ for both learning and testing

We will take a uniform prior on Z: $Z_1, \dots, Z_{d/2}$ iid. ± 1.

Our goal:

① Lower bound $\sum_{i=1}^{d/2} I(Z_i; Y^n)$ for both learning and testing

note: not $I(Z; Y^n)$!

"Assouad-type bound"

Le Cam's method

We will take a uniform prior on Z: $Z_1, \dots, Z_{d/2}$ iid. ± 1.

Our goal:

① Lower bound $\sum_{i=1}^{d/2} I(Z_i; Y^n)$ for both learning and testing

"Assouad-type bound"

Le Cam's method

② Upper bound $\sum_{i=1}^{d/2} I(Z_i; Y^n)$ for both learning and testing

We will take a uniform prior on Z: $Z_1, \dots, Z_{d/2}$ iid. ± 1.

Our goal:

① Lower bound $\sum_{i=1}^{d/2} I(Z_i; Y^n)$ for both learning and testing

"Assouad-type bound"

Le Cam's method

② Upper bound $\sum_{i=1}^{d/2} I(Z_i; Y^n)$ for both learning and testing

as a function of n, ε, d, W

We will take a uniform prior on Z: $Z_1, \dots, Z_{d/2}$ iid. ± 1.

Our goal:

① Lower bound $\sum_{i=1}^{d/2} I(Z_i; Y^n)$ for both learning and testing

"Assouad-type bound" ↙ ↘ Le Cam's method

② Upper bound $\sum_{i=1}^{d/2} I(Z_i; Y^n)$ for both learning and testing *

as a function of n, ε, d, W

③ Put things together to get a LB on n.

to each $W \in \mathcal{W}$ corresponds a psd matrix $H(W)$

Define, for $W \in \mathcal{W}$, the matrix $H(W)$ by

$$H(W)_{ij} = \sum_y \frac{(W(y|2i-1) - W(y|2i))(W(y|2j-1) - W(y|2j))}{\sum_x W(y|x)} \qquad i,j \in [d/2]$$

$\sup_{W \in \mathcal{W}} \|H(W)\|_{op}$

let us call this $\|H(\mathcal{W})\|_{op}$

$\sup_{W \in \mathcal{W}} \mathrm{Tr}[H(W)]$

$\|H(\mathcal{W})\|_*$

For interactive protocols under constraint $\mathcal{W}$

$$\text{Learning: } n = \Omega\left(\frac{d^2}{\varepsilon^2 \|H(\mathcal{W})\|_*}\right)$$

$$\text{Testing: } n = \Omega\left(\frac{d}{\varepsilon^2 \sqrt{\|H(\mathcal{W})\|_* \|H(\mathcal{W})\|_{op}}}\right)$$

where $\|H(\mathcal{W})\| := \sup_{W \in \mathcal{W}} \|H(W)\|$

What about the $\Omega(k^{3/2})$ private-coin lower bound?

Are interactive and public-coin the same?

Testing

Long story short: get

$$n = \Omega\left(\frac{d^{3/2}}{\varepsilon^2 \|H(w)\|_*}\right)$$

for private-coin; and

$$n = \Omega\left(\frac{d}{\varepsilon^2 \|H(w)\|_F}\right)$$

for public-coin.

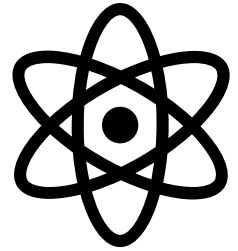
Hölder: $\|H(w)\|_F^2 \leq \|H(w)\|_{op} \|H(w)\|_*$

ℓ_2^2 $\quad$ ℓ_∞ $\quad$ ℓ_1

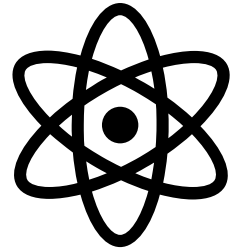
More details, discussion, full proofs:

- *Inference under Information Constraints I: Lower Bounds from Chi-Square Contraction.* Jayadev Acharya, **C.**, and Himanshu Tyagi (IEEE Trans. Inf. Theory, 2020). arXiv:1812.11476
- *Interactive Inference under Information Constraints.* Jayadev Acharya, **C.**, Yuhan Liu, Ziteng Sun, and Himanshu Tyagi (IEEE Trans. Inf. Theory, 2021). arXiv:2007.10976
- *Unified lower bounds for interactive high-dimensional estimation under information constraints.* Jayadev Acharya, **C.**, Ziteng Sun, and Himanshu Tyagi (2021). arXiv:2010.06562 [Generalisation]

But I promised some quantum

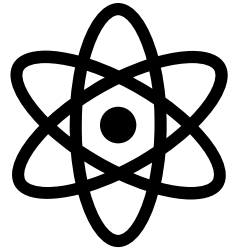


But I promised some quantum



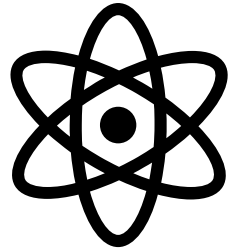
- We've been considering testing (and learning) of classical discrete probability distributions over d elements, with the distance metric being total variation, so ℓ_1.
- One can also consider tomography and mixedness testing:
  - Given n copies of a d-dim mixed state ρ, learn ρ to **trace distance** ε
  - Given n copies of a d-dim mixed state ρ, test if ρ is maximally mixed (vs. ε-far from it in **trace distance**)
  - More generally, one can also consider quantum certification (wrt known σ)

Natural generalisation of the previous problems



Tomography	Mixedness testing/certification
$\Theta(d^2/\varepsilon^2)$	$\Theta(d/\varepsilon^2)$
[O'Donnell–Wright] arXiv:1508.01907	[O'Donnell–Wright] arXiv:1501.05028 [Badescu–O'Donnell–Wright] arXiv:1708.06002

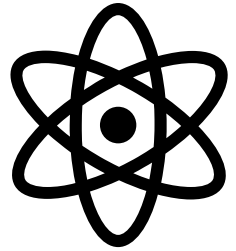
But that requires full entanglement



What if we want to do the same testing tasks with **simpler measurements**?

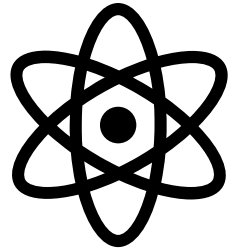
- Non-entangled
- Even non-adaptive!

Note: fits nicely into the "restricted measurement" view



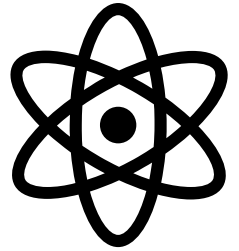
Mixedness testing/certification (non-adaptive)	Mixedness testing/certification (adaptive)
$\Theta(d^{3/2}/\varepsilon^2)$	$\Omega(d^{4/3}/\varepsilon^2)$
[Bubeck–Chen–Li] arXiv:2004.07869	

Discussion



- What about the tight bound for adaptive non-entangled testing?
- Is there a finer-grained separation between public- and private-coin?
- What about similar separations for tomography?
- Can the techniques and framework developed for the classical case be applied to the quantum generalisation?

Discussion



- What about the tight bound for adaptive non-entangled testing?
- Is there a finer-grained separation between public- and private-coin?
- What about similar separations for tomography?
- Can the techniques and framework developed for the classical case be applied to the quantum generalisation?

Thank you