Katsushika Hokusai (葛飾 北斎)
Enoshima in Sagami Province (相州江の島)
from Thirty-six views of Mount Fuji

RUTGERS

An information theorist visits differential privacy

Anand D. Sarwate, Rutgers University

20 May 2025

INFORMED AI Seminar
University of Bristol

Some thanks and credits

Thanks for helpful discussions with
Shahab Asoodeh (McMaster)
Flavio Calmon (Harvard)
Oliver Kosut (Arizona State)
Lalitha Sankar (Arizona State)
Mario Diaz (UNAM) - in memoriam

Image credits:

- Wikimedia Commons
- ARC Ukiyo-e dataset
- **OpenMoji.org**

Today's talk

I will tell you stuff you know already (possibly? probably?)

Today's talk

I will tell you stuff you know already (possibly? probably?)

Differential privacy is a way of quantifying how different two distributions are.
Information theorists also think about this kind of thing.

Today's talk

I will tell you stuff you know already (possibly? probably?)

Differential privacy is a way of quantifying how different two distributions are.
Information theorists also think about this kind of thing.

But: the kinds of questions/settings can have a different flavor. Today:

Today's talk

I will tell you stuff you know already (possibly? probably?)

Differential privacy is a way of quantifying how different two distributions are. Information theorists also think about this kind of thing.

But: the kinds of questions/settings can have a different flavor. Today:

hypothesis testing   f-Divergences   contraction coefficients

Today's talk

I will tell you stuff you know already (possibly? probably?)

Differential privacy is a way of quantifying how different two distributions are. Information theorists also think about this kind of thing.

But: the kinds of questions/settings can have a different flavor. Today:

hypothesis testing **f-Divergences** **contraction coefficients**

Goals:

Today's talk

I will tell you stuff you know already (possibly? probably?)

Differential privacy is a way of quantifying how different two distributions are. Information theorists also think about this kind of thing.

But: the kinds of questions/settings can have a different flavor. Today:

hypothesis testing **f-Divergences** **contraction coefficients**

Goals:

- Describe some of these three connections for those less familiar

Today's talk

I will tell you stuff you know already (possibly? probably?)

Differential privacy is a way of quantifying how different two distributions are. Information theorists also think about this kind of thing.

But: the kinds of questions/settings can have a different flavor. Today:

hypothesis testing **f-Divergences** **contraction coefficients**

Goals:

- Describe some of these three connections for those less familiar
- Suggest some questions for discussion later?

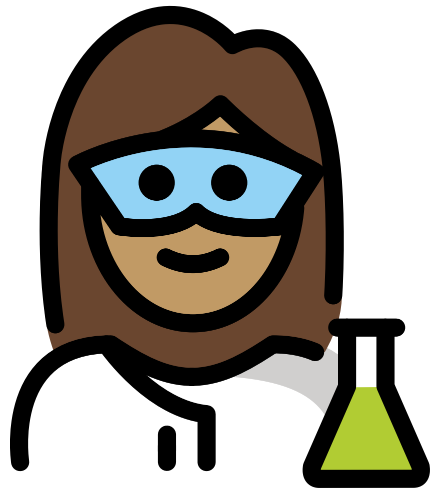
The binary hypothesis test

Let’s start simple

The binary hypothesis test

Let's start simple

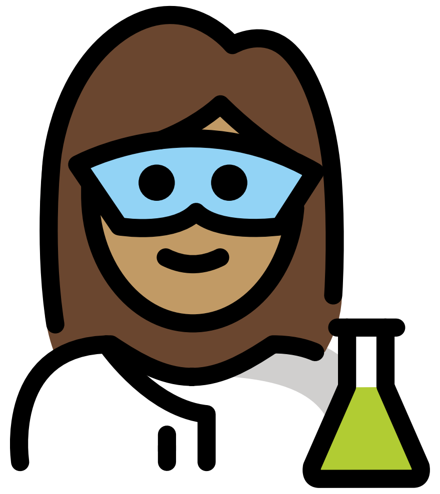
Sasha



The binary hypothesis test

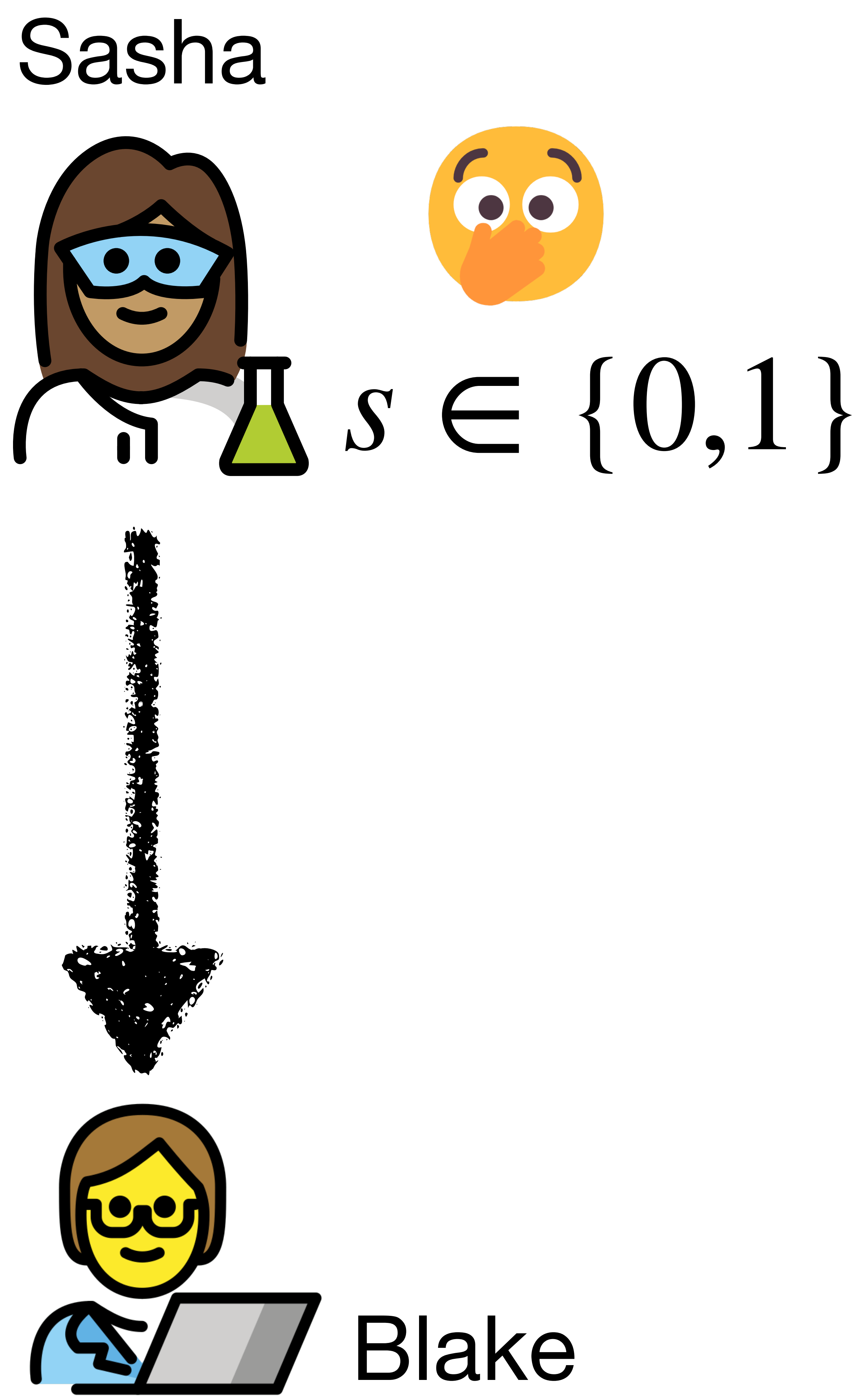
Let's start simple

Sasha

 $s \in \{0,1\}$

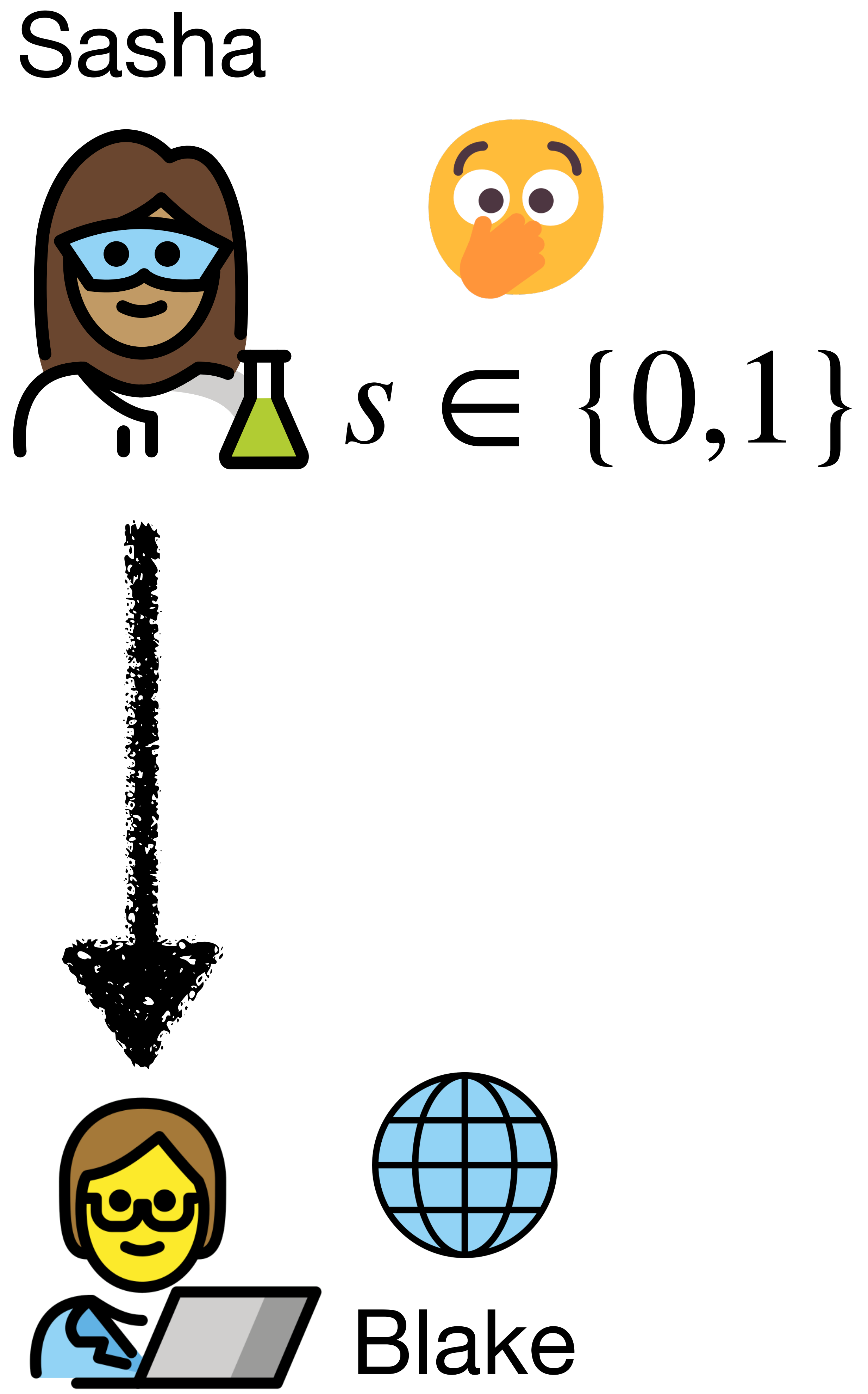
The binary hypothesis test

Let's start simple



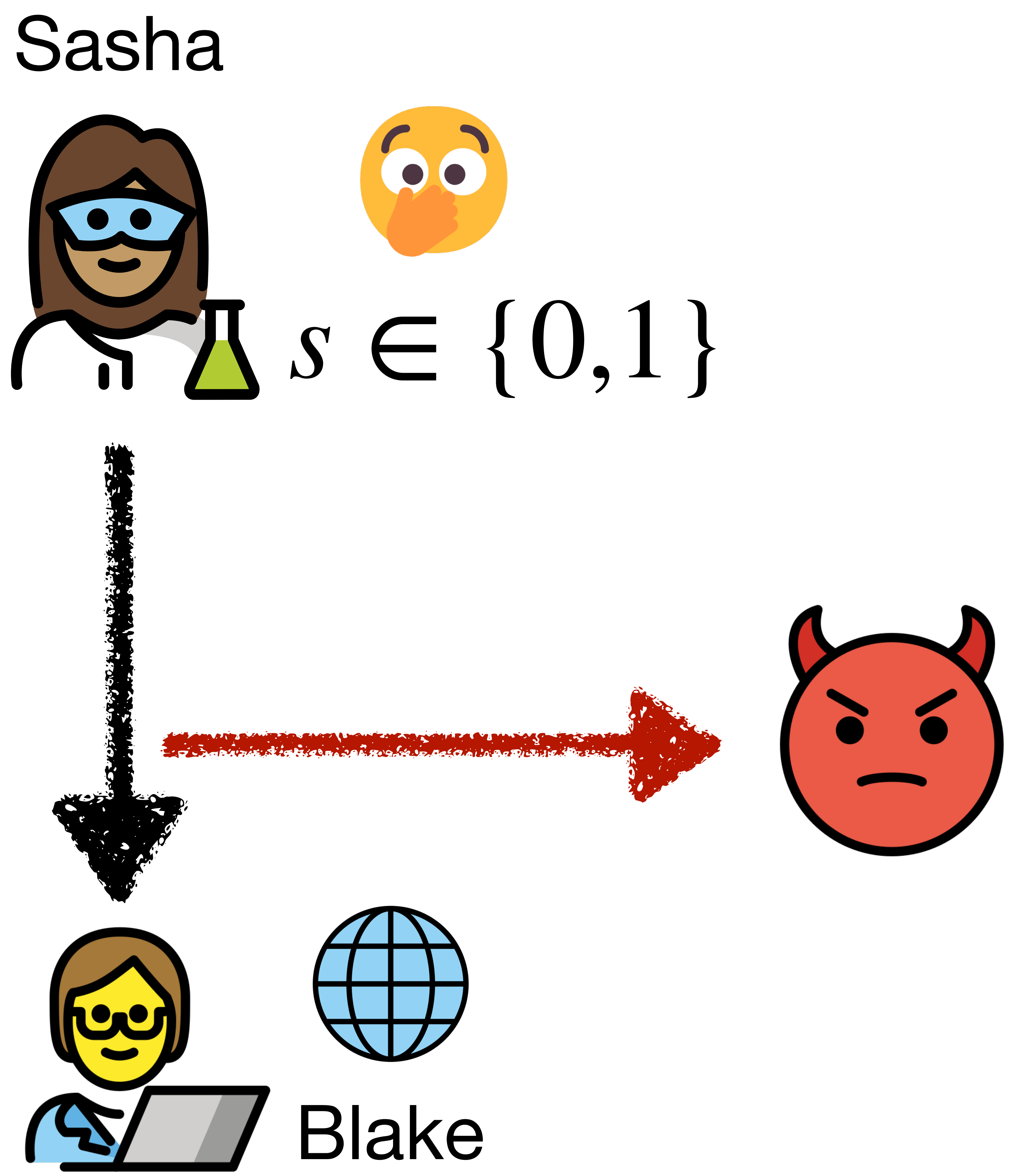
The binary hypothesis test

Let's start simple



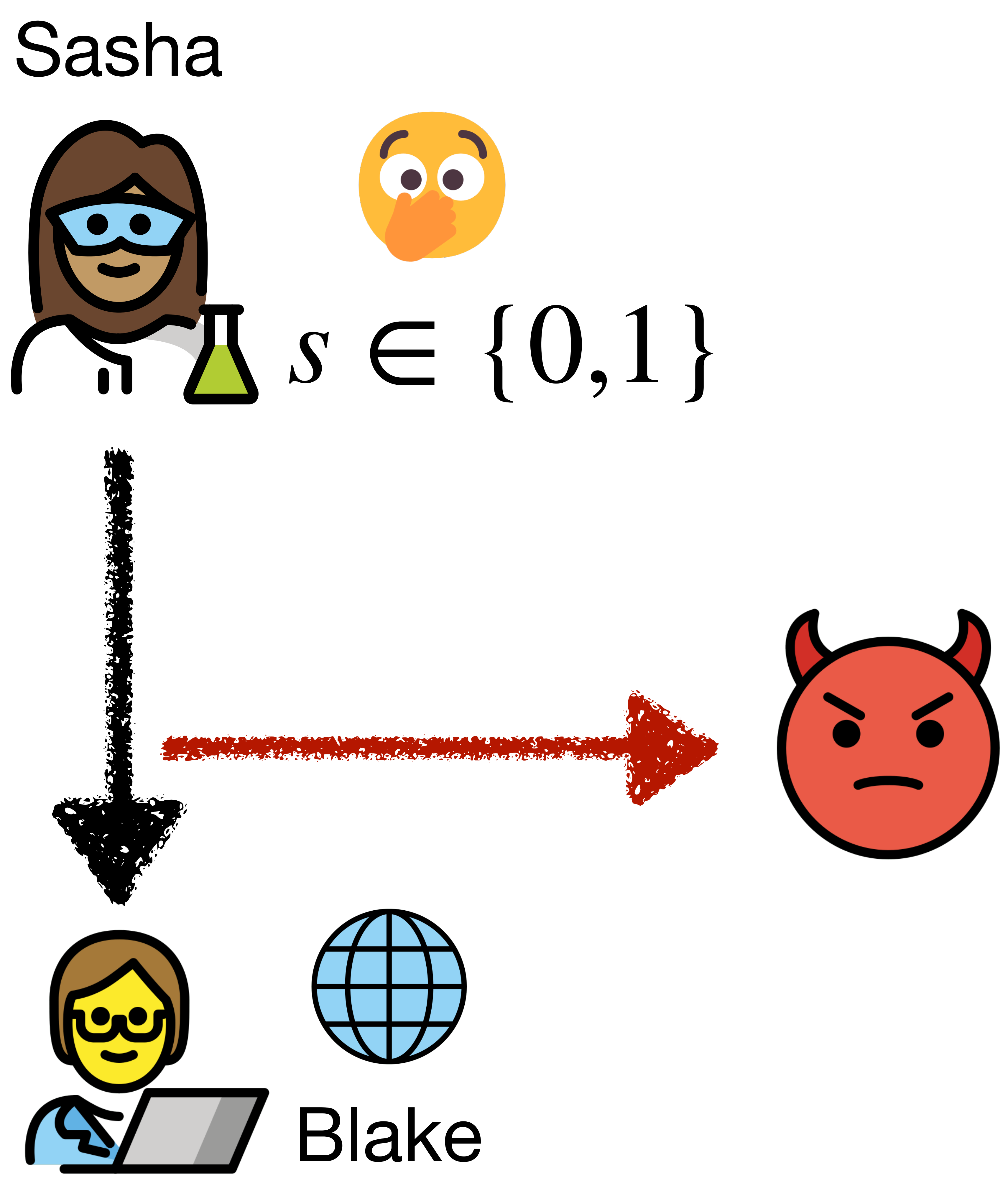
The binary hypothesis test

Let's start simple



The binary hypothesis test

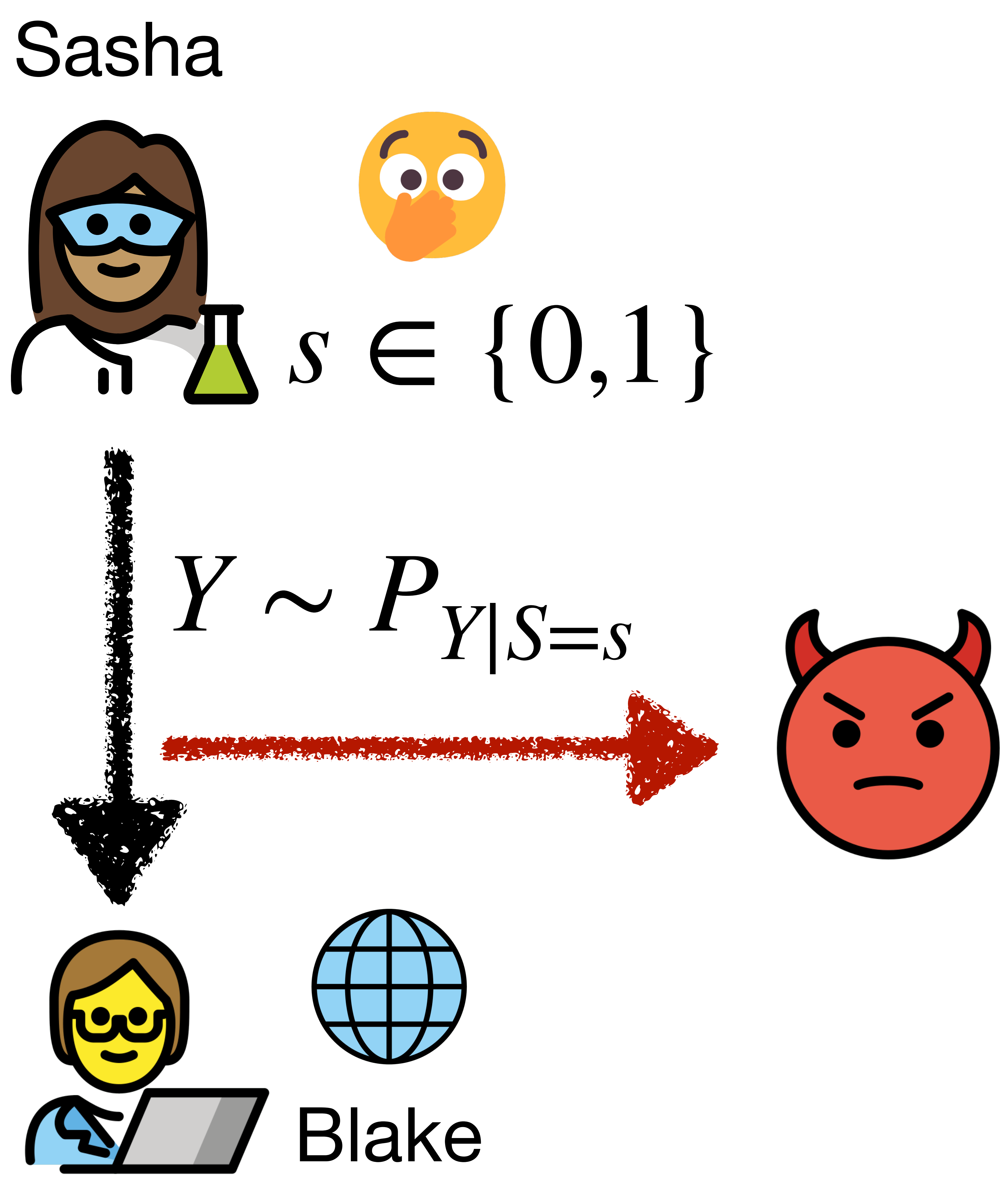
Let’s start simple



Want to hide one bit $s \in \{0,1\}$ but have to reveal a random variable Y whose distribution depends on s.

The binary hypothesis test

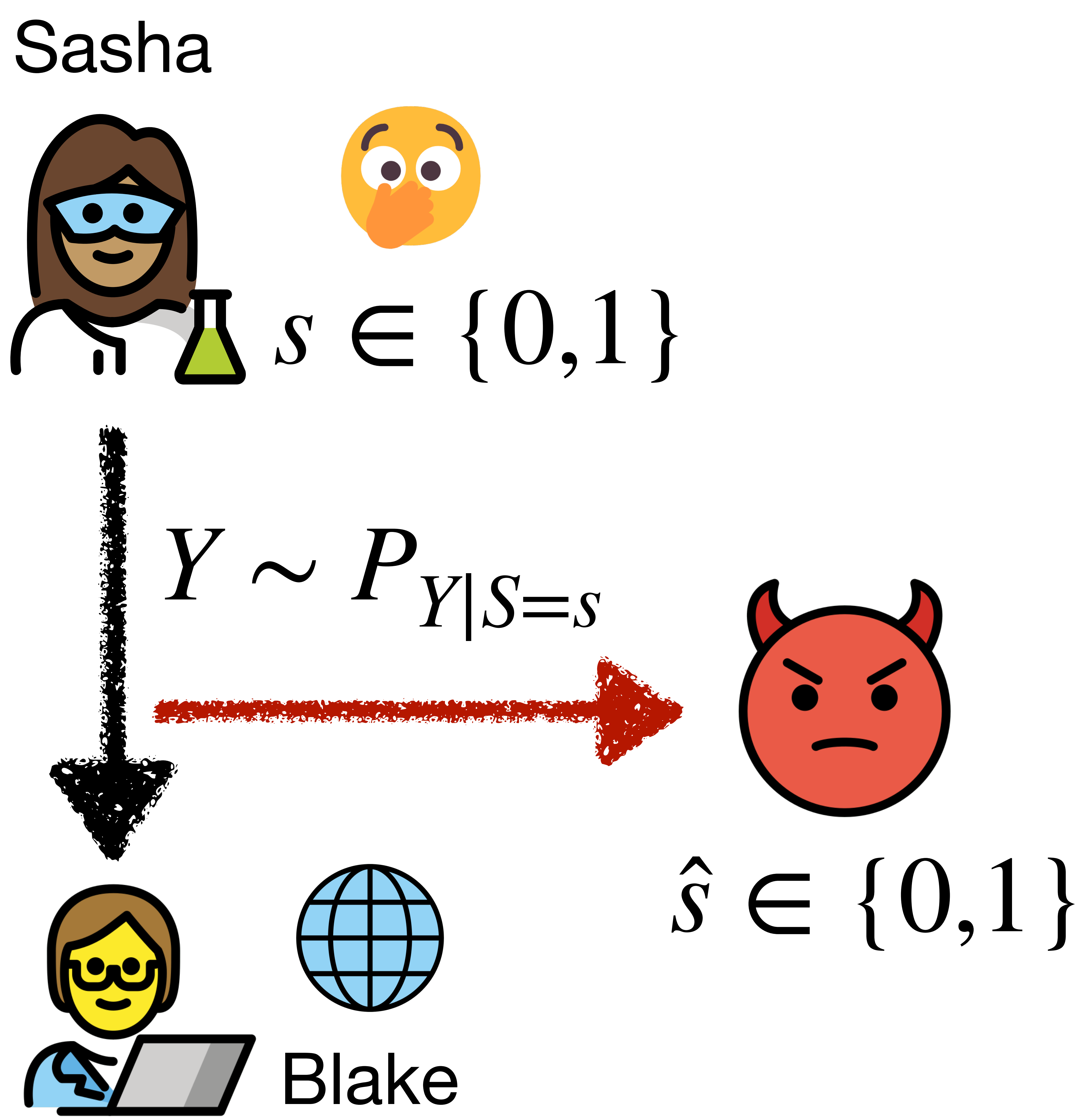
Let's start simple



Want to hide one bit $s \in \{0,1\}$ but have to reveal a random variable Y whose distribution depends on s.

The binary hypothesis test

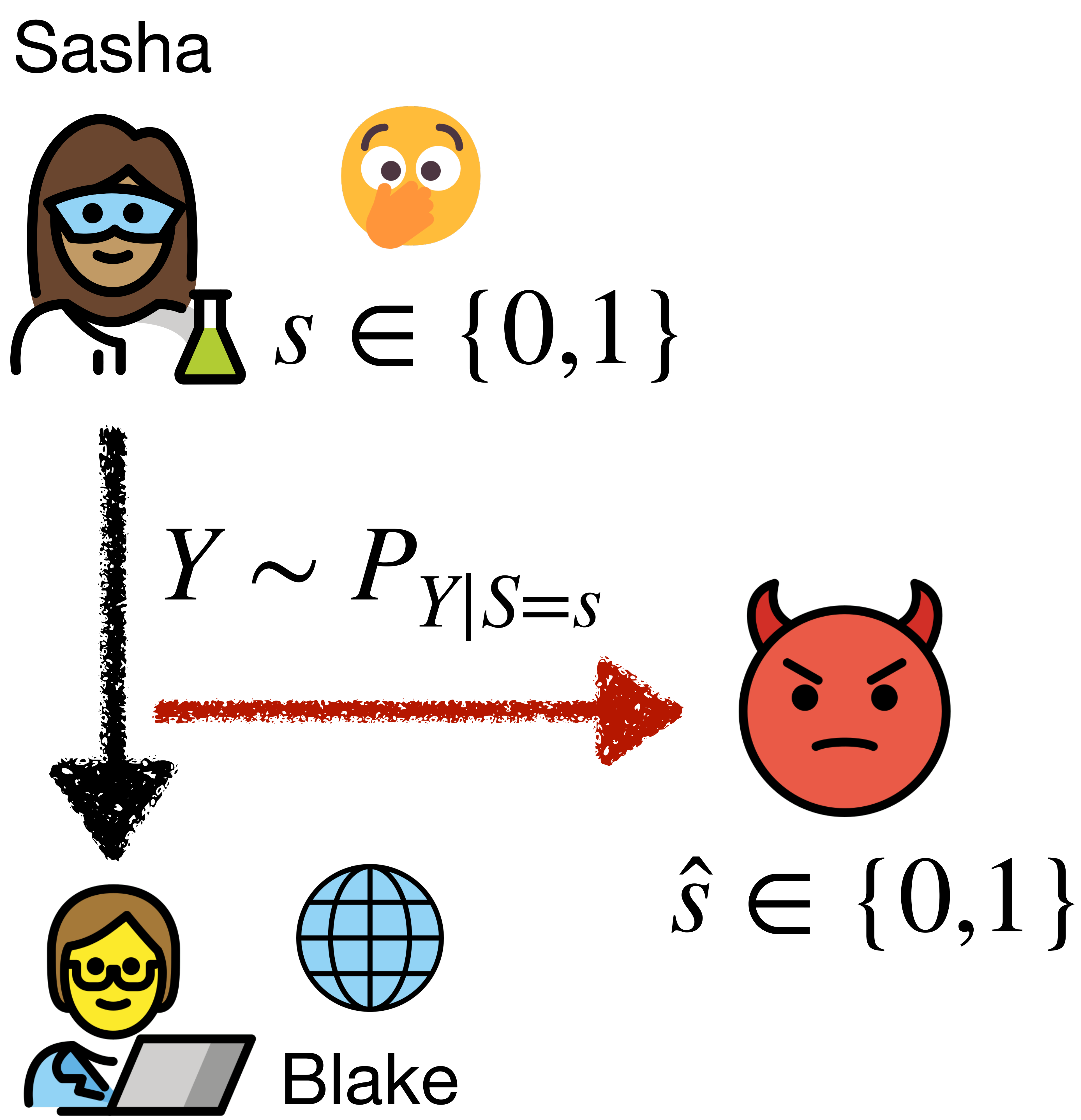
Let's start simple



Want to hide one bit $s \in \{0,1\}$ but have to reveal a random variable Y whose distribution depends on s.

The binary hypothesis test

Let's start simple



Want to hide one bit $s \in \{0,1\}$ but have to reveal a random variable Y whose distribution depends on s.

The **privacy question** is a **hypothesis testing question**:

$$\mathscr{H}_0 : Y \sim P_{Y|S=0}$$

$$\mathscr{H}_1 : Y \sim P_{Y|S=1}$$

The Lake of Hakone in Sagami Province

相州箱根湖水

Sōshū Hakone Kosui

Vista 1

hypothesis testing

Neyman-Pearson tells us the optimal test

Adversarial inference is a generalized LRT

Neyman-Pearson tells us the optimal test

Adversarial inference is a generalized LRT

The optimal test for the adversary is a
likelihood ratio test:

Neyman-Pearson tells us the optimal test

Adversarial inference is a generalized LRT

The optimal test for the adversary is a **likelihood ratio test**:

$$\hat{s}(y) = \begin{cases} 1 & \log \frac{P_{Y|S=1}(y)}{P_{Y|S=0}(y)} \geq \tau \\ 0 & \log \frac{P_{Y|S=1}(y)}{P_{Y|S=0}(y)} < \tau \end{cases}$$

Neyman-Pearson tells us the optimal test

Adversarial inference is a generalized LRT

The optimal test for the adversary is a **likelihood ratio test**:

$$\hat{s}(y) = \begin{cases} 1 & \log \dfrac{P_{Y|S=1}(y)}{P_{Y|S=0}(y)} \geq \tau \\ 0 & \log \dfrac{P_{Y|S=1}(y)}{P_{Y|S=0}(y)} < \tau \end{cases}$$

Neyman-Pearson tells us the optimal test

Adversarial inference is a generalized LRT

The optimal test for the adversary is a **likelihood ratio test**:

$$\hat{s}(y) = \begin{cases} 1 & \log \dfrac{P_{Y|S=1}(y)}{P_{Y|S=0}(y)} \geq \tau \\ 0 & \log \dfrac{P_{Y|S=1}(y)}{P_{Y|S=0}(y)} < \tau \end{cases}$$

Neyman-Pearson tells us the optimal test

Adversarial inference is a generalized LRT

The optimal test for the adversary is a **likelihood ratio test**:

$$\hat{s}(y) = \begin{cases} 1 & \log \dfrac{P_{Y|S=1}(y)}{P_{Y|S=0}(y)} \geq \tau \\ 0 & \log \dfrac{P_{Y|S=1}(y)}{P_{Y|S=0}(y)} < \tau \end{cases}$$

Example

$\mathcal{H}_0: Y = 0 + Z \sim \mathcal{N}(0, \sigma^2)$

$\mathcal{H}_1: Y = 1 + Z \sim \mathcal{N}(1, \sigma^2)$

Example: additive Gaussian noise

Everyone's favorite example: Gaussians!

Example: additive Gaussian noise

Everyone's favorite example: Gaussians!

If the revealed information Z is **Gaussian**:

Example: additive Gaussian noise

Everyone's favorite example: Gaussians!

If the revealed information Z is **Gaussian**:

$$\mathscr{H}_0 : Z \sim \mathcal{N}(0, \sigma^2)$$

Example: additive Gaussian noise

Everyone's favorite example: Gaussians!

If the revealed information Z is **Gaussian**:

$$\mathcal{H}_0: Z \sim \mathcal{N}(0, \sigma^2)$$

$$\mathcal{H}_1: Z \sim \mathcal{N}(1, \sigma^2)$$

Example: additive Gaussian noise

Everyone's favorite example: Gaussians!

If the revealed information Z is **Gaussian**:

$$\mathscr{H}_0 : Z \sim \mathcal{N}(0, \sigma^2)$$

$$\mathscr{H}_1 : Z \sim \mathcal{N}(1, \sigma^2)$$

We can write the error probabilities in terms of Q functions:

Example: additive Gaussian noise

Everyone's favorite example: Gaussians!

If the revealed information Z is **Gaussian**:

$$\mathcal{H}_0 : Z \sim \mathcal{N}(0, \sigma^2)$$

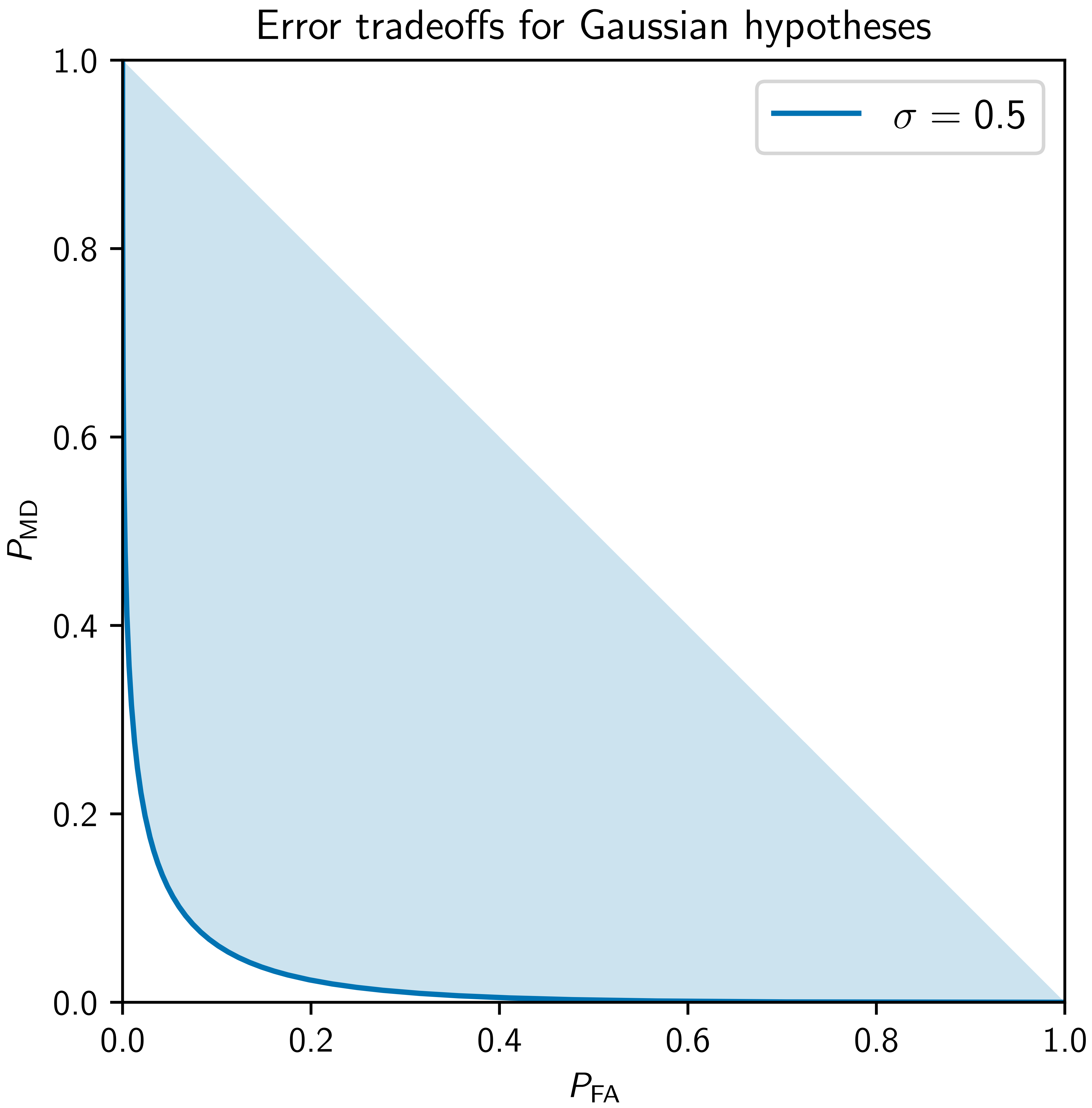
$$\mathcal{H}_1 : Z \sim \mathcal{N}(1, \sigma^2)$$

We can write the error probabilities in terms of Q functions:

$$P_{\mathrm{FA}} = Q\left(\frac{t}{\sigma}\right), P_{\mathrm{MD}} = Q\left(\frac{1-t}{\sigma}\right).$$

Example: additive Gaussian noise

Everyone's favorite example: Gaussians!



If the revealed information Z is **Gaussian**:

$$\mathscr{H}_0 : Z \sim \mathcal{N}(0, \sigma^2)$$

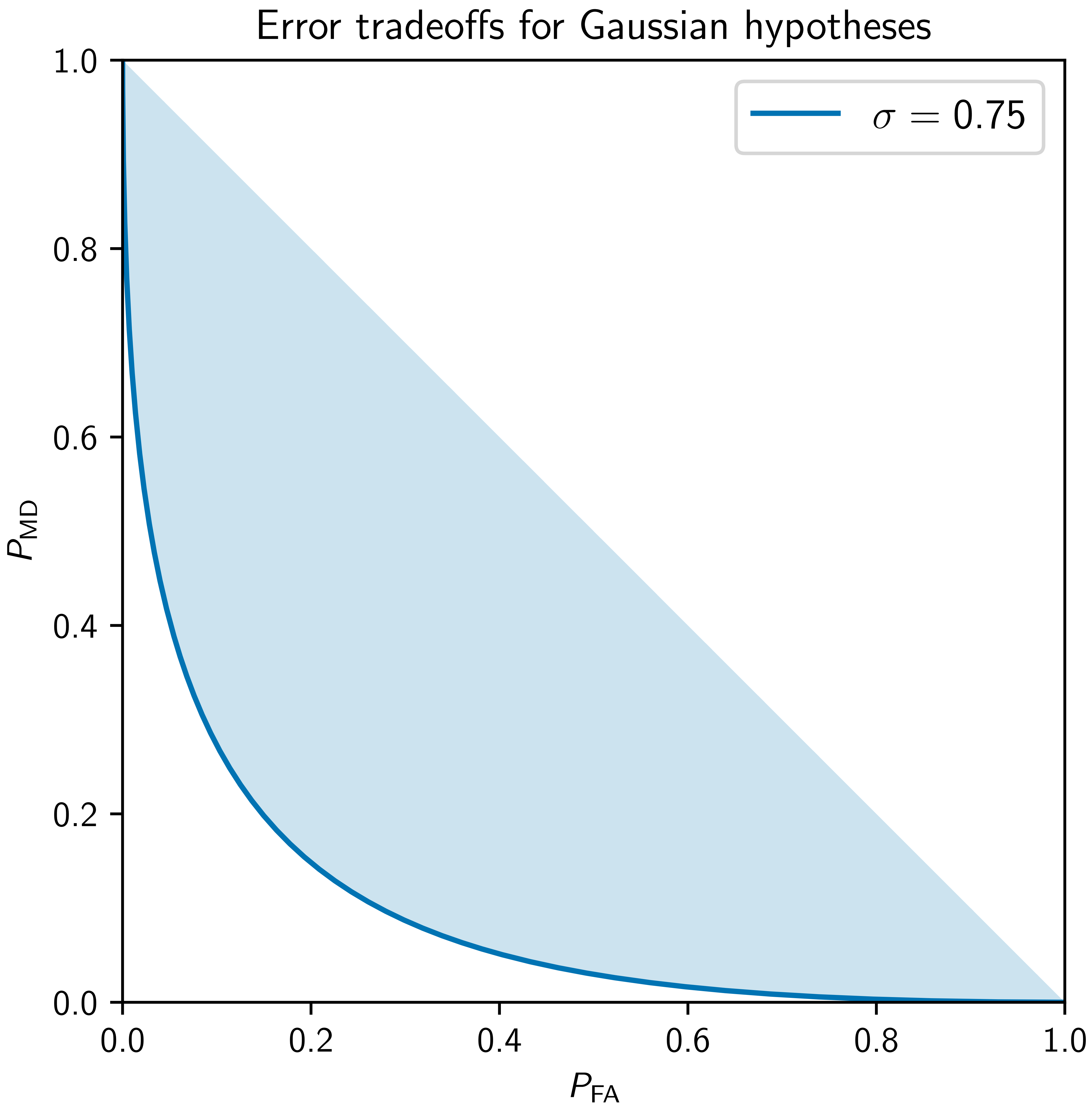
$$\mathscr{H}_1 : Z \sim \mathcal{N}(1, \sigma^2)$$

We can write the error probabilities in terms of Q functions:

$$P_{\mathrm{FA}} = Q\left(\frac{t}{\sigma}\right), P_{\mathrm{MD}} = Q\left(\frac{1-t}{\sigma}\right).$$

Example: additive Gaussian noise

Everyone's favorite example: Gaussians!



If the revealed information Z is **Gaussian**:

$$\mathcal{H}_0 : Z \sim \mathcal{N}(0, \sigma^2)$$

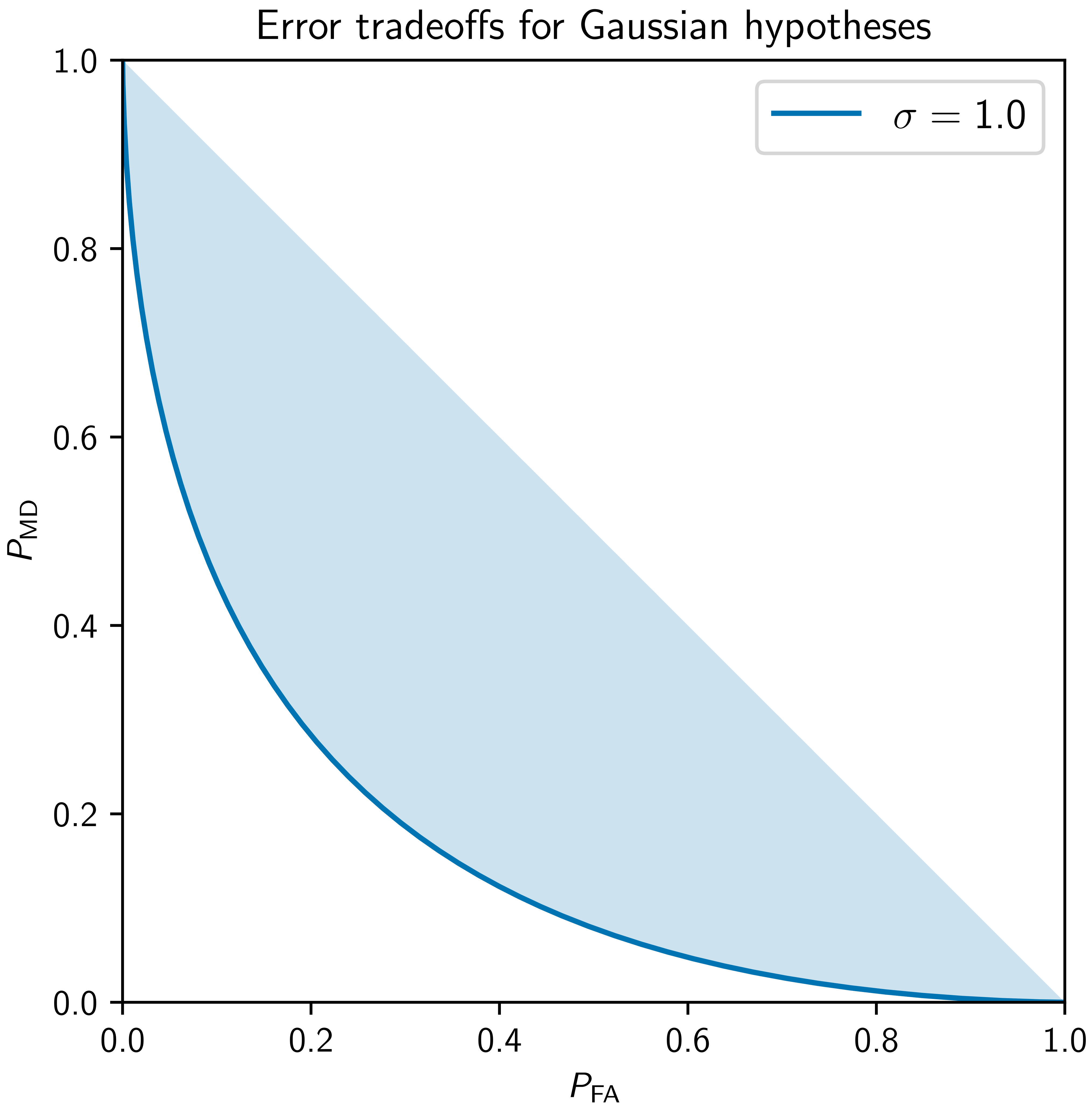
$$\mathcal{H}_1 : Z \sim \mathcal{N}(1, \sigma^2)$$

We can write the error probabilities in terms of Q functions:

$$P_{\mathrm{FA}} = Q\left(\frac{t}{\sigma}\right), P_{\mathrm{MD}} = Q\left(\frac{1-t}{\sigma}\right).$$

Example: additive Gaussian noise

Everyone's favorite example: Gaussians!



If the revealed information Z is **Gaussian**:

$$\mathcal{H}_0 : Z \sim \mathcal{N}(0, \sigma^2)$$

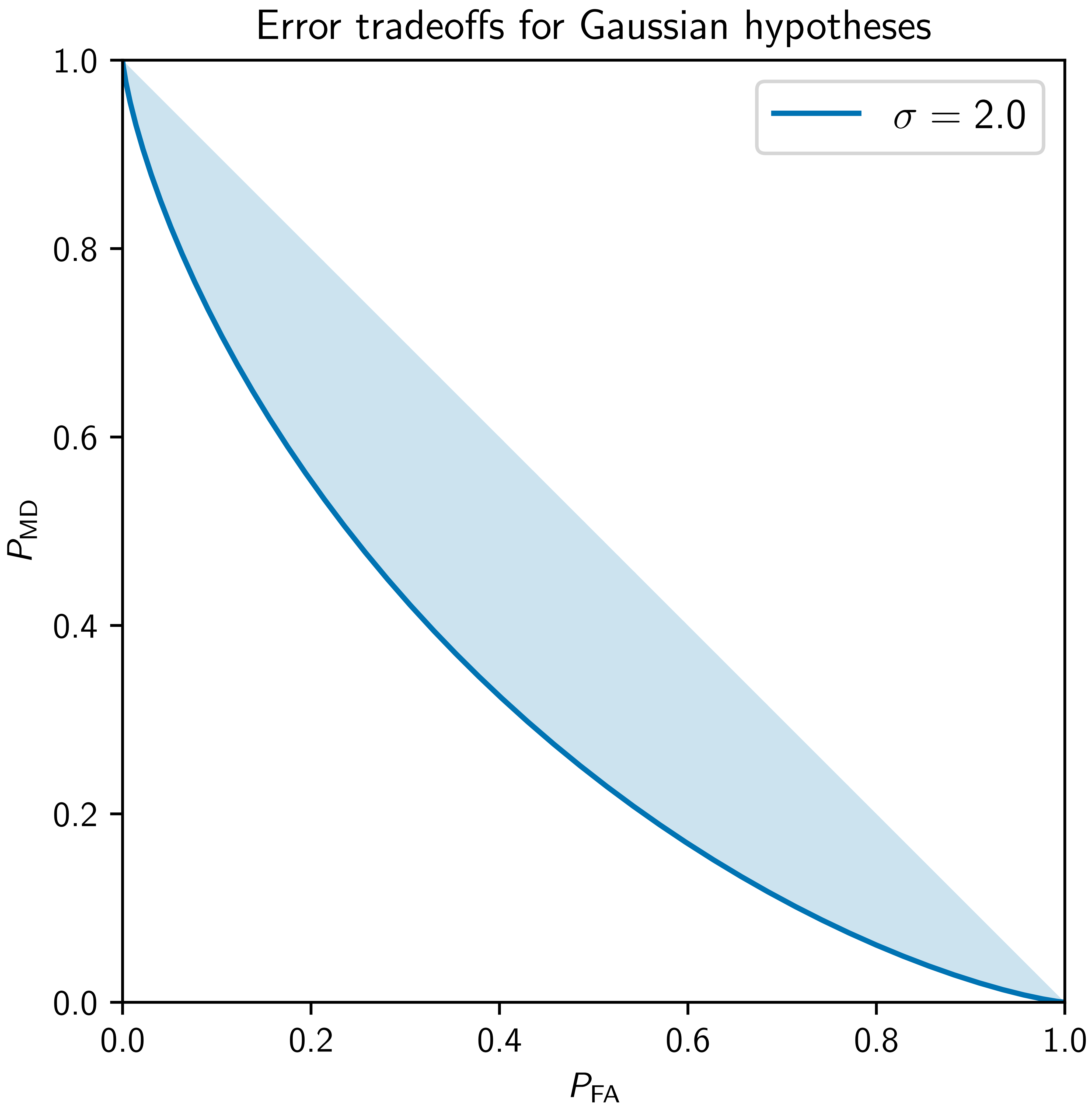
$$\mathcal{H}_1 : Z \sim \mathcal{N}(1, \sigma^2)$$

We can write the error probabilities in terms of Q functions:

$$P_{FA} = Q\left(\frac{t}{\sigma}\right), P_{MD} = Q\left(\frac{1-t}{\sigma}\right).$$

Example: additive Gaussian noise

Everyone's favorite example: Gaussians!



If the revealed information Z is **Gaussian**:

$$\mathscr{H}_0 : Z \sim \mathcal{N}(0, \sigma^2)$$

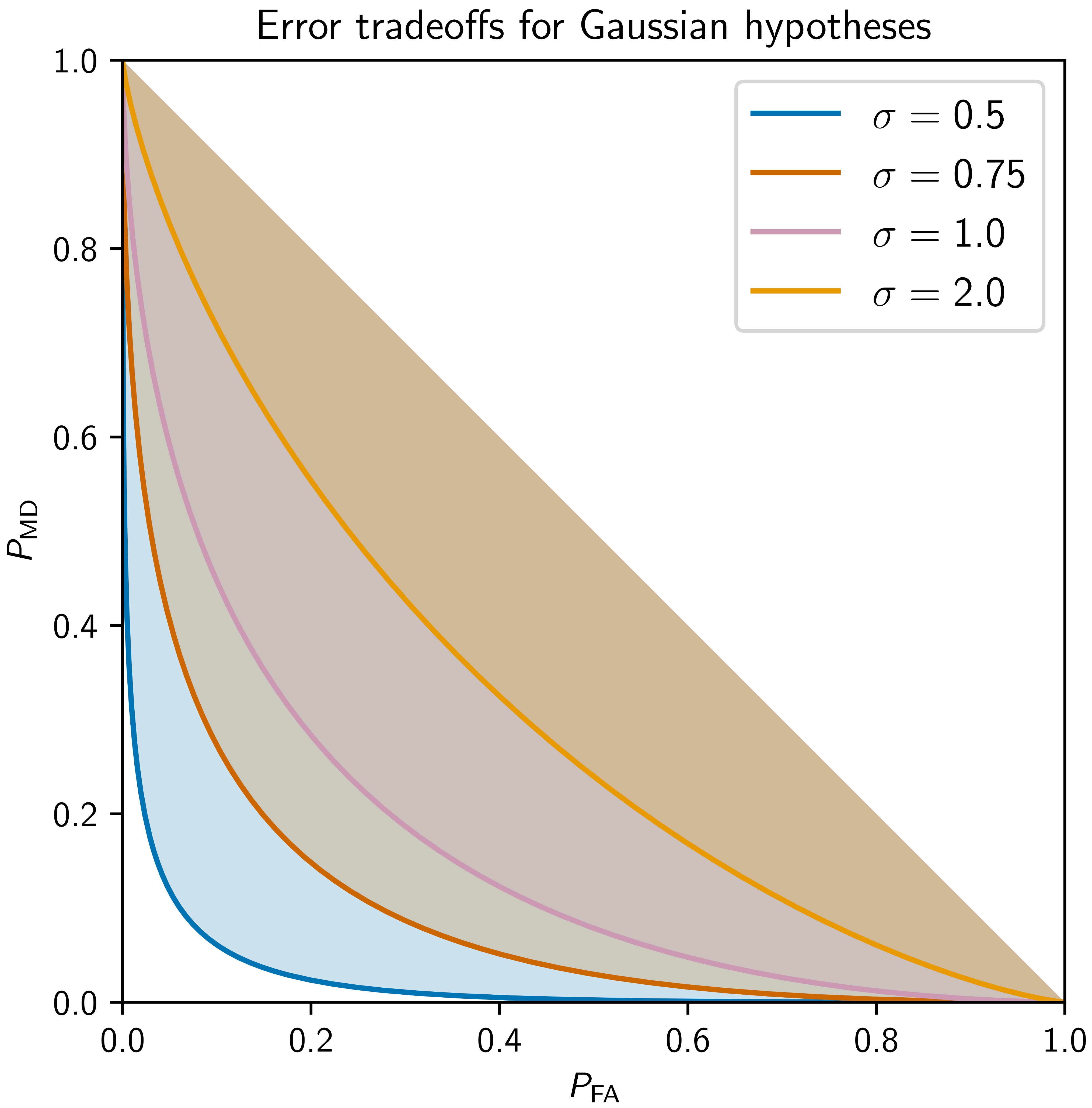
$$\mathscr{H}_1 : Z \sim \mathcal{N}(1, \sigma^2)$$

We can write the error probabilities in terms of Q functions:

$$P_{\mathrm{FA}} = Q\left(\frac{t}{\sigma}\right), P_{\mathrm{MD}} = Q\left(\frac{1-t}{\sigma}\right).$$

Example: additive Gaussian noise

Everyone's favorite example: Gaussians!



If the revealed information Z is **Gaussian**:

$$\mathcal{H}_0 : Z \sim \mathcal{N}(0, \sigma^2)$$

$$\mathcal{H}_1 : Z \sim \mathcal{N}(1, \sigma^2)$$

We can write the error probabilities in terms of Q functions:

$$P_{\mathrm{FA}} = Q\left(\frac{t}{\sigma}\right), P_{\mathrm{MD}} = Q\left(\frac{1-t}{\sigma}\right).$$

Example: additive Laplace noise

We can do more than just Gaussians!

Example: additive Laplace noise

We can do more than just Gaussians!

If the revealed information Z is **Lapace**:

Example: additive Laplace noise

We can do more than just Gaussians!

If the revealed information Z is **Lapace**:

$$\mathscr{H}_0 : X \sim \text{Laplace}(\lambda)$$

Example: additive Laplace noise

We can do more than just Gaussians!

If the revealed information Z is **Lapace**:

$$\mathcal{H}_0 : X \sim \text{Laplace}(\lambda)$$

$$\mathcal{H}_1 : X \sim 1 + \text{Laplace}(\lambda)$$

Example: additive Laplace noise

We can do more than just Gaussians!

If the revealed information Z is **Lapace**:

$$\mathscr{H}_0 : X \sim \text{Laplace}(\lambda)$$

$$\mathscr{H}_1 : X \sim 1 + \text{Laplace}(\lambda)$$

Where $\text{Laplace}(\lambda)$ has density

Example: additive Laplace noise

We can do more than just Gaussians!

If the revealed information Z is **Lapace**:

$$\mathscr{H}_0 : X \sim \text{Laplace}(\lambda)$$

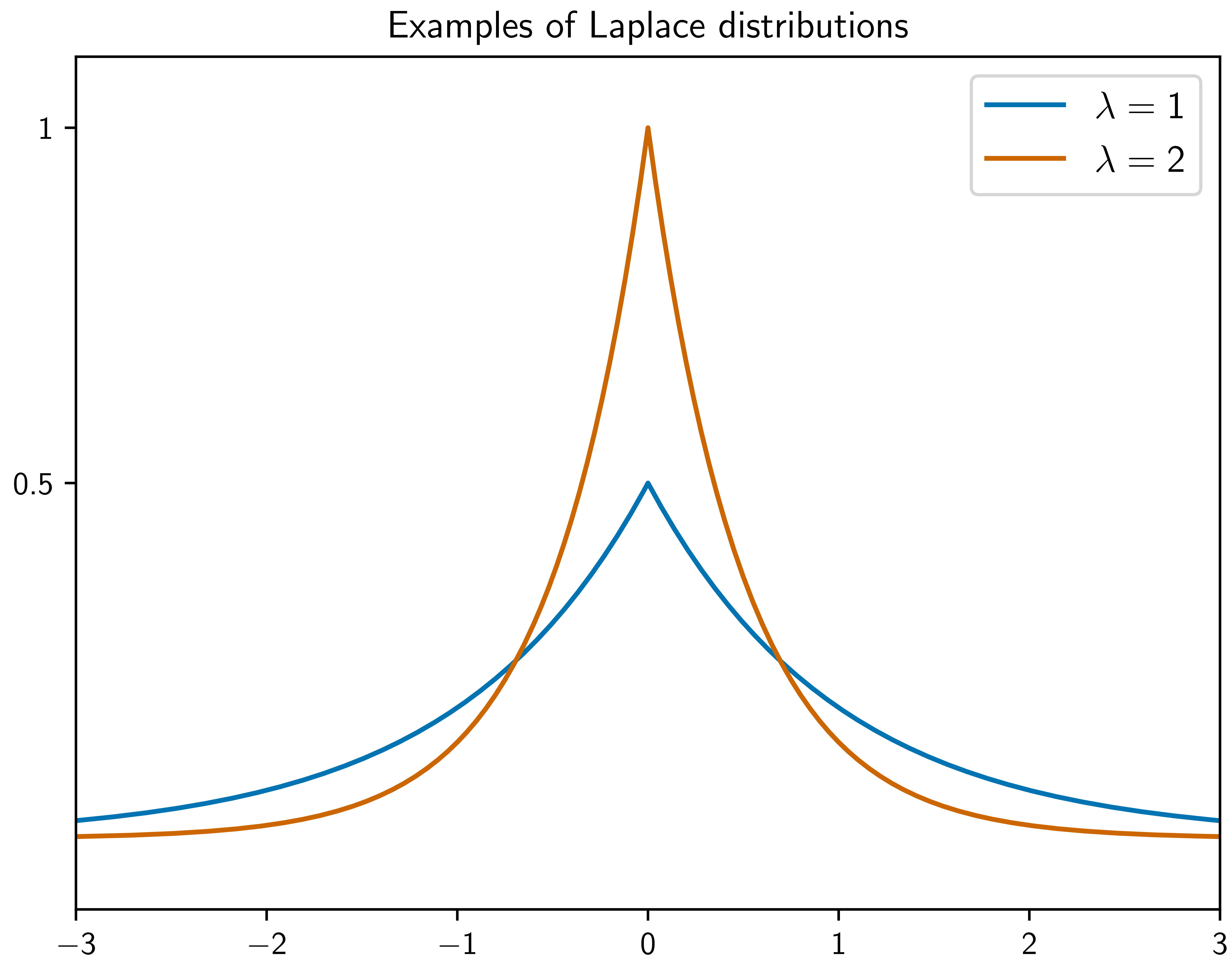
$$\mathscr{H}_1 : X \sim 1 + \text{Laplace}(\lambda)$$

Where $\text{Laplace}(\lambda)$ has density

$$p(z) = \frac{\lambda}{2} \exp(-\lambda |z|).$$

Example: additive Laplace noise

We can do more than just Gaussians!



If the revealed information Z is **Lapace**:

$$\mathscr{H}_0 : X \sim \text{Laplace}(\lambda)$$

$$\mathscr{H}_1 : X \sim 1 + \text{Laplace}(\lambda)$$

Where $\text{Laplace}(\lambda)$ has density

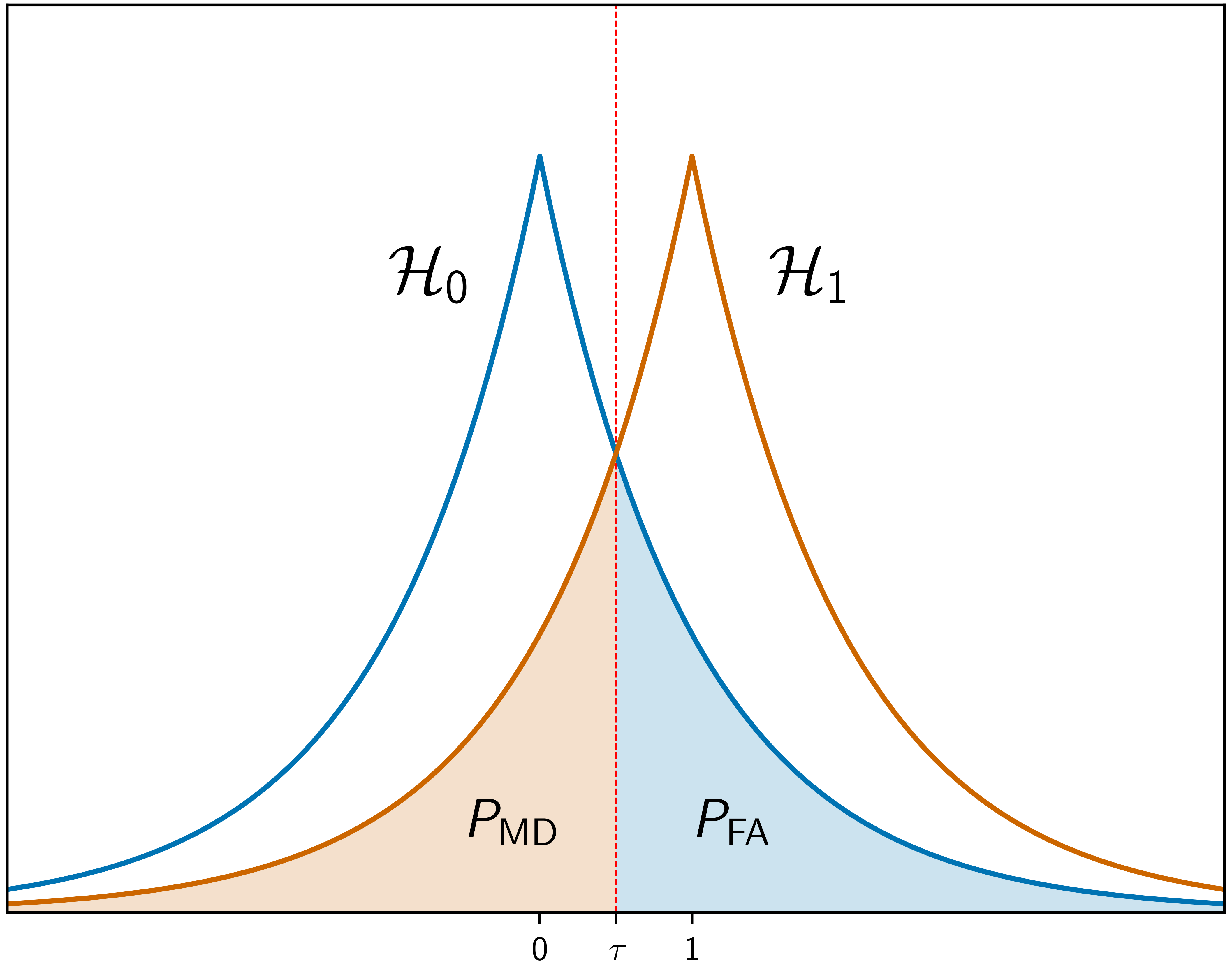
$$p(z) = \frac{\lambda}{2} \exp(-\lambda |z|).$$

Error tradeoffs for Laplace noise

Lighter tails give a different shape

Error tradeoffs for Laplace noise

Lighter tails give a different shape



The error probabilities for the test are:

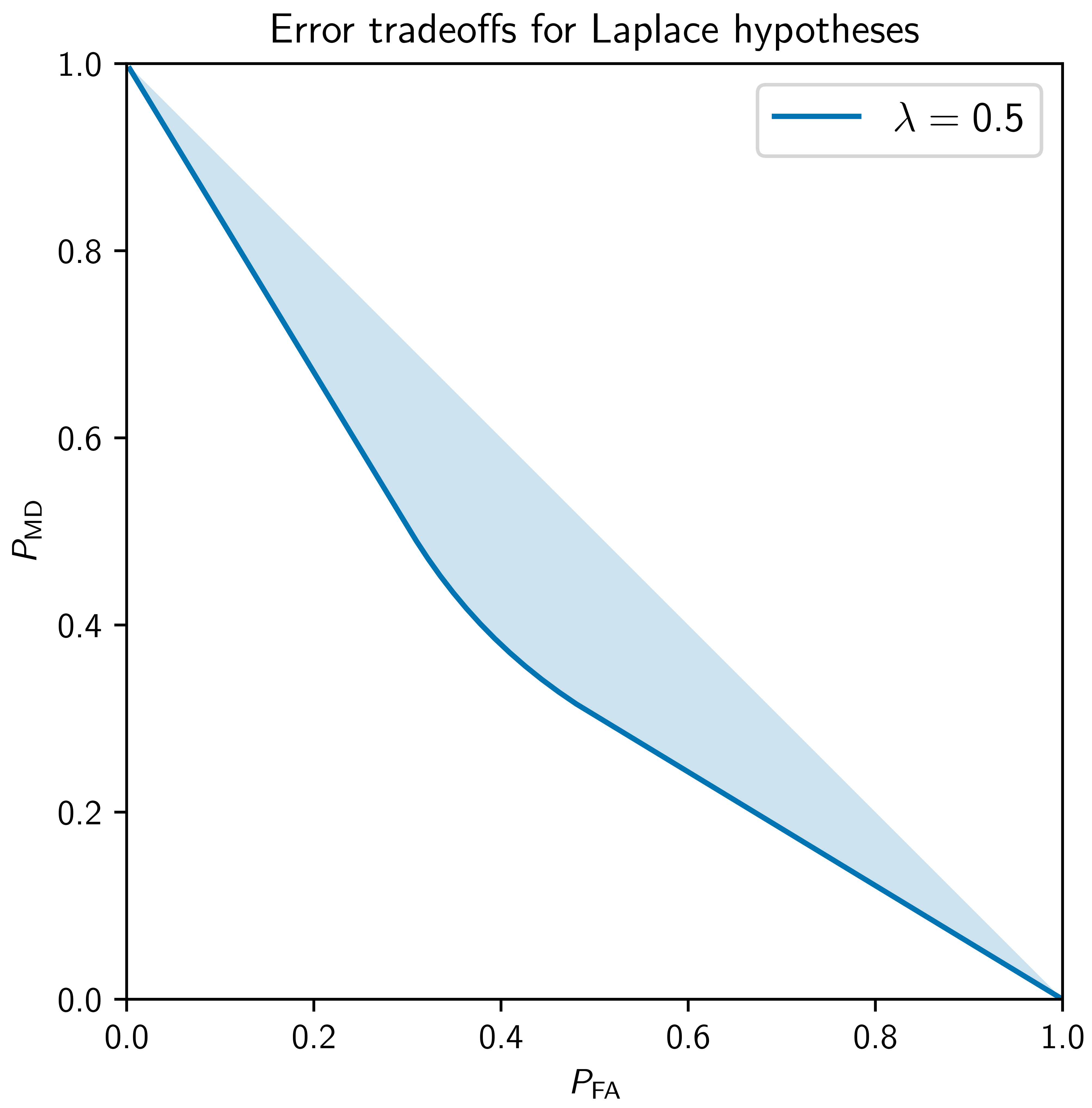
$$P_{\mathrm{FA}} = \int_t^{\infty} \frac{\lambda}{2} \exp(-|t|\lambda) dt$$

$$P_{\mathrm{MD}} = \int_{-\infty}^{t} \frac{\lambda}{2} \exp(-|A-t|\lambda) dt$$

The tradeoff is similar to the Gaussian but the slope at the corners is different.

Error tradeoffs for Laplace noise

Lighter tails give a different shape



The error probabilities for the test are:

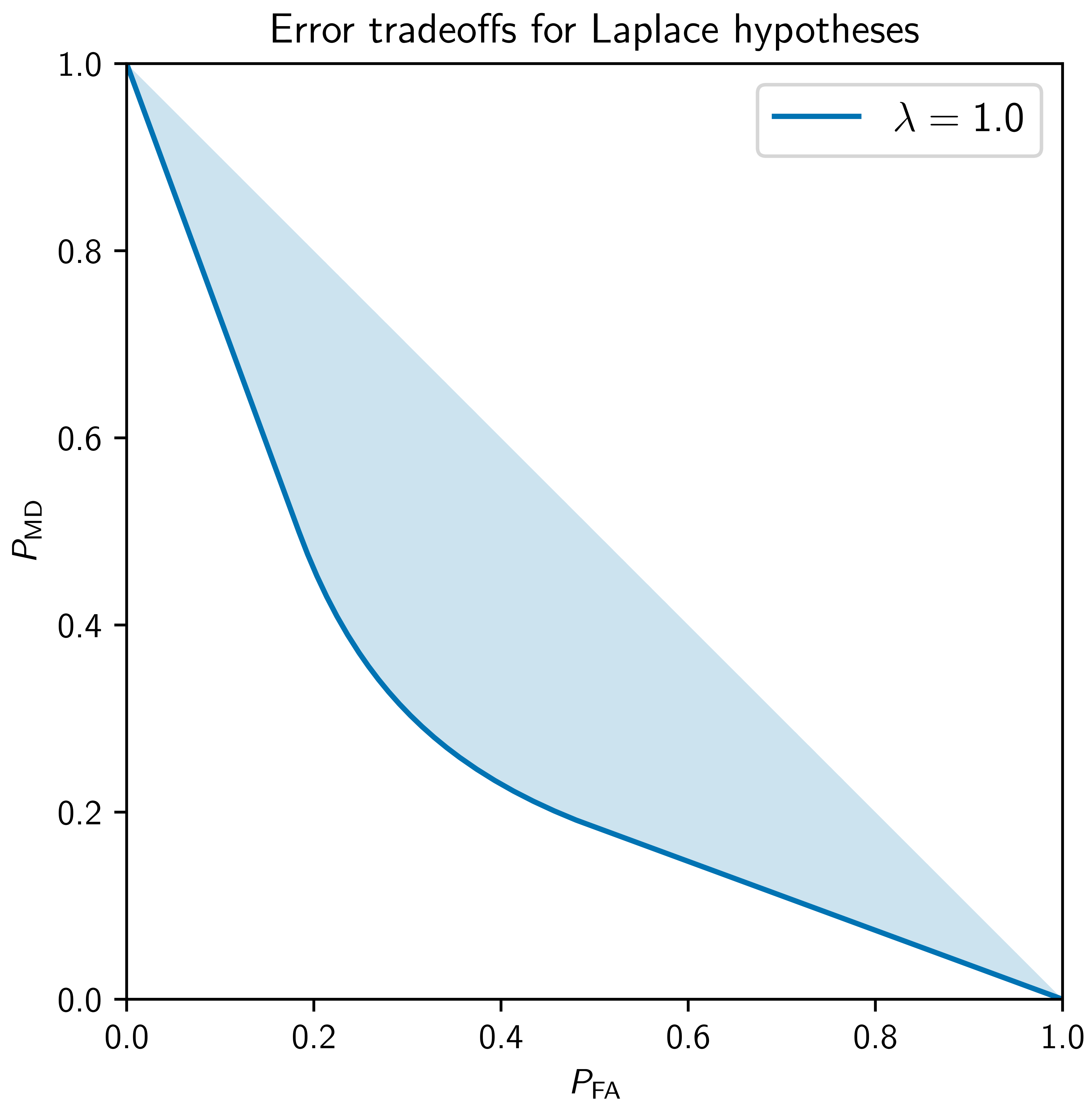
$$P_{\text{FA}} = \int_t^{\infty} \frac{\lambda}{2} \exp(-|t|\lambda)dt$$

$$P_{\text{MD}} = \int_{-\infty}^{t} \frac{\lambda}{2} \exp(-|A-t|\lambda)dt$$

The tradeoff is similar to the Gaussian but the slope at the corners is different.

Error tradeoffs for Laplace noise

Lighter tails give a different shape



The error probabilities for the test are:

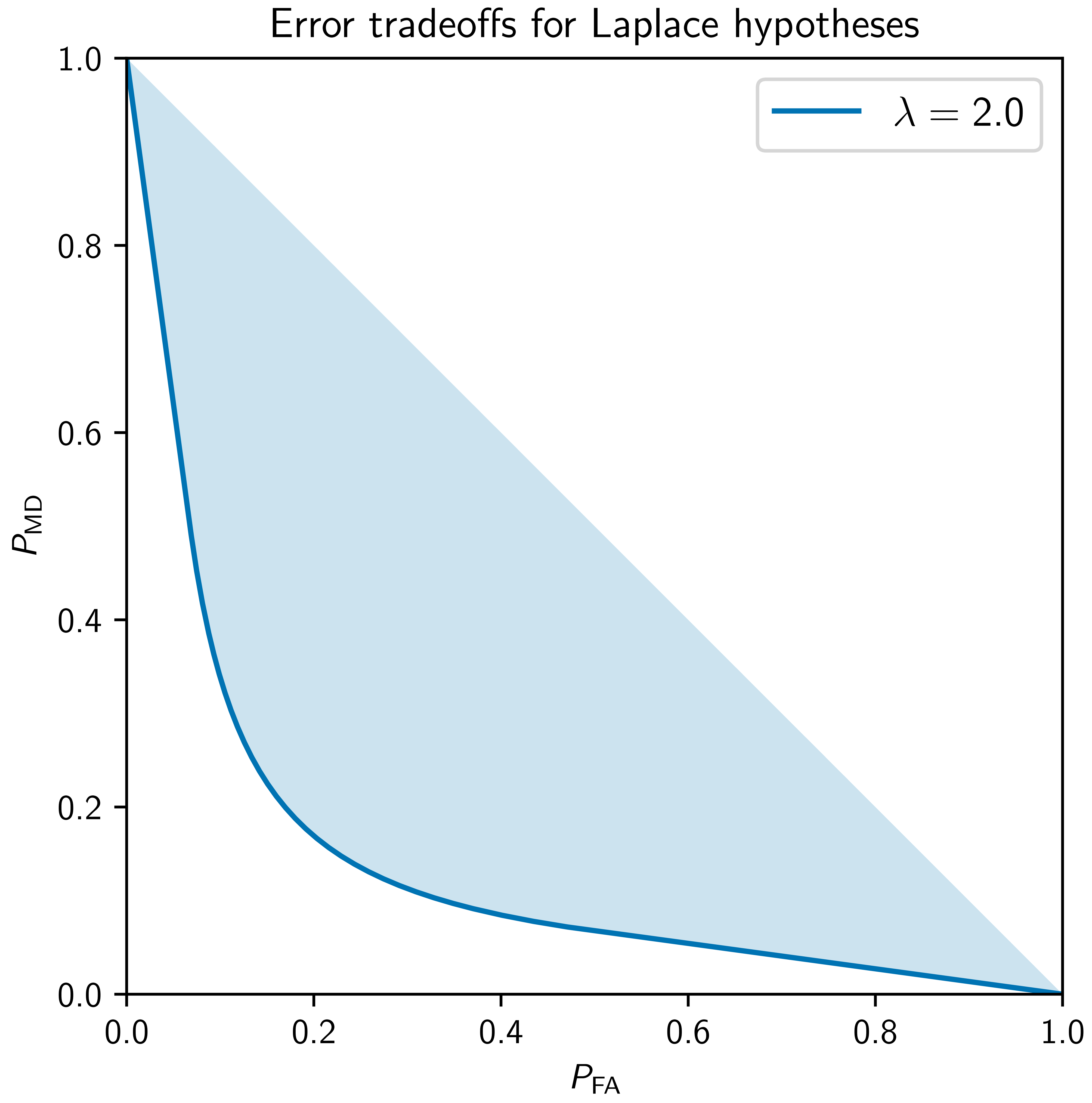
$$P_{\mathrm{FA}} = \int_t^{\infty} \frac{\lambda}{2} \exp(-|t|\lambda) dt$$

$$P_{\mathrm{MD}} = \int_{-\infty}^{t} \frac{\lambda}{2} \exp(-|A-t|\lambda) dt$$

The tradeoff is similar to the Gaussian but the slope at the corners is different.

Error tradeoffs for Laplace noise

Lighter tails give a different shape



The error probabilities for the test are:

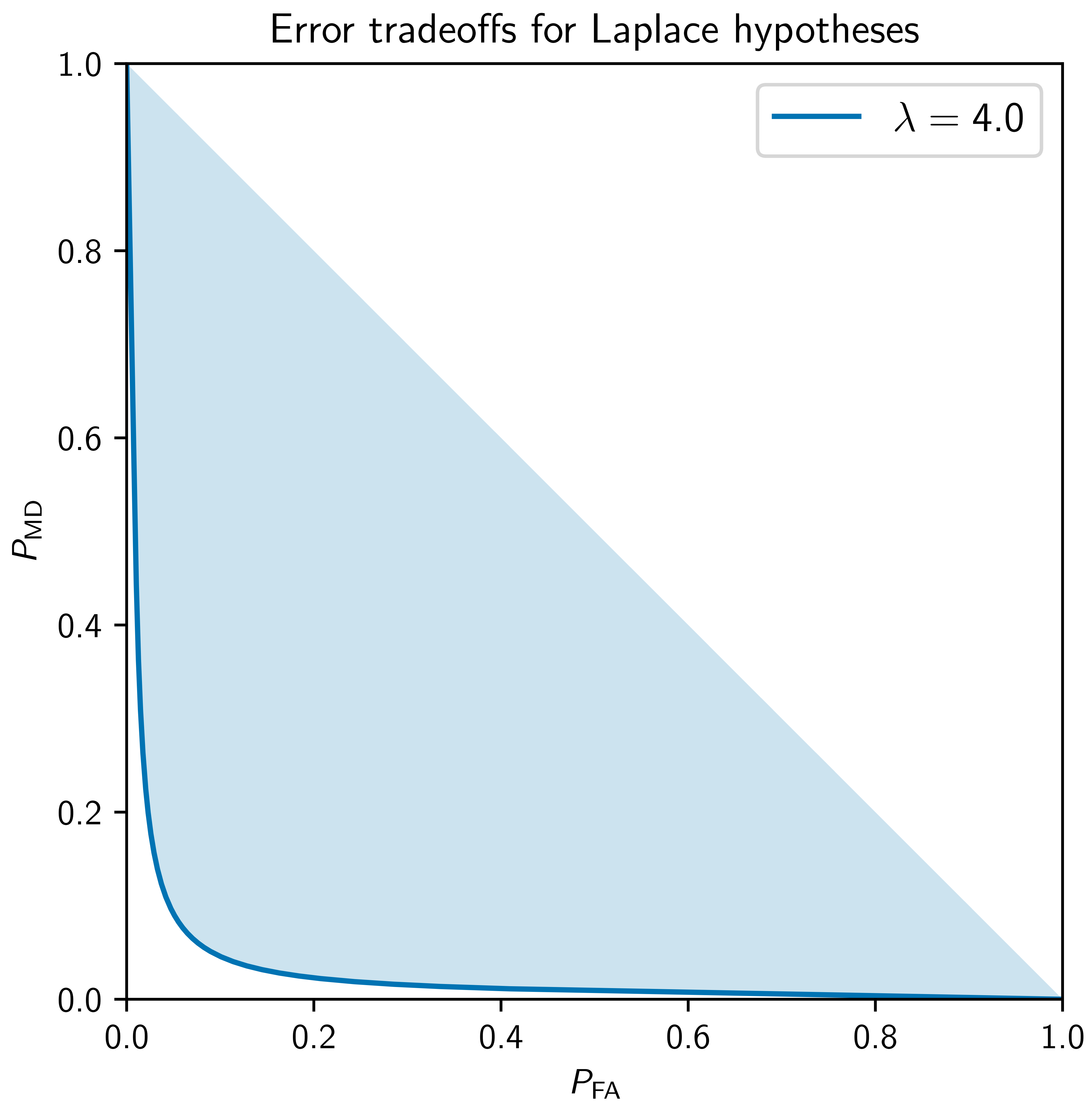
$$P_{\mathrm{FA}} = \int_t^{\infty} \frac{\lambda}{2} \exp(-|t|\lambda)dt$$

$$P_{\mathrm{MD}} = \int_{-\infty}^{t} \frac{\lambda}{2} \exp(-|A-t|\lambda)dt$$

The tradeoff is similar to the Gaussian but the slope at the corners is different.

Error tradeoffs for Laplace noise

Lighter tails give a different shape



The error probabilities for the test are:

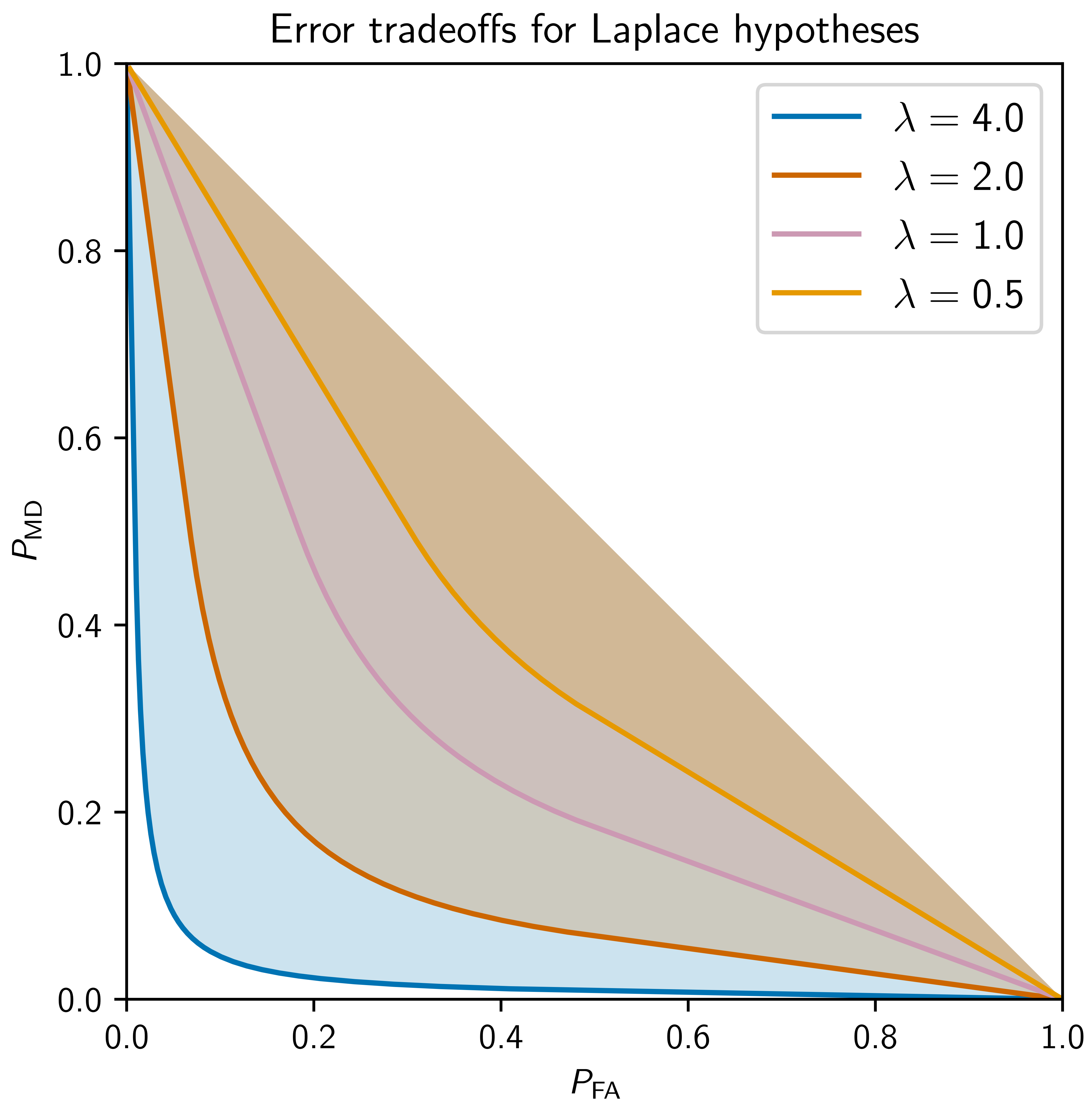
$$P_{\mathrm{FA}} = \int_t^{\infty} \frac{\lambda}{2} \exp(-|t|\lambda)dt$$

$$P_{\mathrm{MD}} = \int_{-\infty}^{t} \frac{\lambda}{2} \exp(-|A-t|\lambda)dt$$

The tradeoff is similar to the Gaussian but the slope at the corners is different.

Error tradeoffs for Laplace noise

Lighter tails give a different shape



The error probabilities for the test are:

$$P_{\mathrm{FA}} = \int_t^{\infty} \frac{\lambda}{2} \exp(-|t|\lambda) dt$$

$$P_{\mathrm{MD}} = \int_{-\infty}^{t} \frac{\lambda}{2} \exp(-|A-t|\lambda) dt$$

The tradeoff is similar to the Gaussian but the slope at the corners is different.

Tradeoffs between P_{FA} and P_{MD}

We get more privacy when the hypothesis test is "hard"

Tradeoffs between P_{FA} and P_{MD}

We get more privacy when the hypothesis test is "hard"

- A **privacy guarantee** is made by the tradeoff between probabilities of

Tradeoffs between P_{FA} and P_{MD}

We get more privacy when the hypothesis test is "hard"

- A **privacy guarantee** is made by the tradeoff between probabilities of
  - **false alarm** (Type I error) and

Tradeoffs between P_{FA} and P_{MD}

We get more privacy when the hypothesis test is “hard”

- A **privacy guarantee** is made by the tradeoff between probabilities of
  - **false alarm** (Type I error) and
  - **missed detection** (Type 2 error)

Tradeoffs between P_{FA} and P_{MD}

We get more privacy when the hypothesis test is “hard”

- A **privacy guarantee** is made by the tradeoff between probabilities of
  - **false alarm** (Type I error) and
  - **missed detection** (Type 2 error)
- If the likelihood ratio is small, the test will have a higher error.

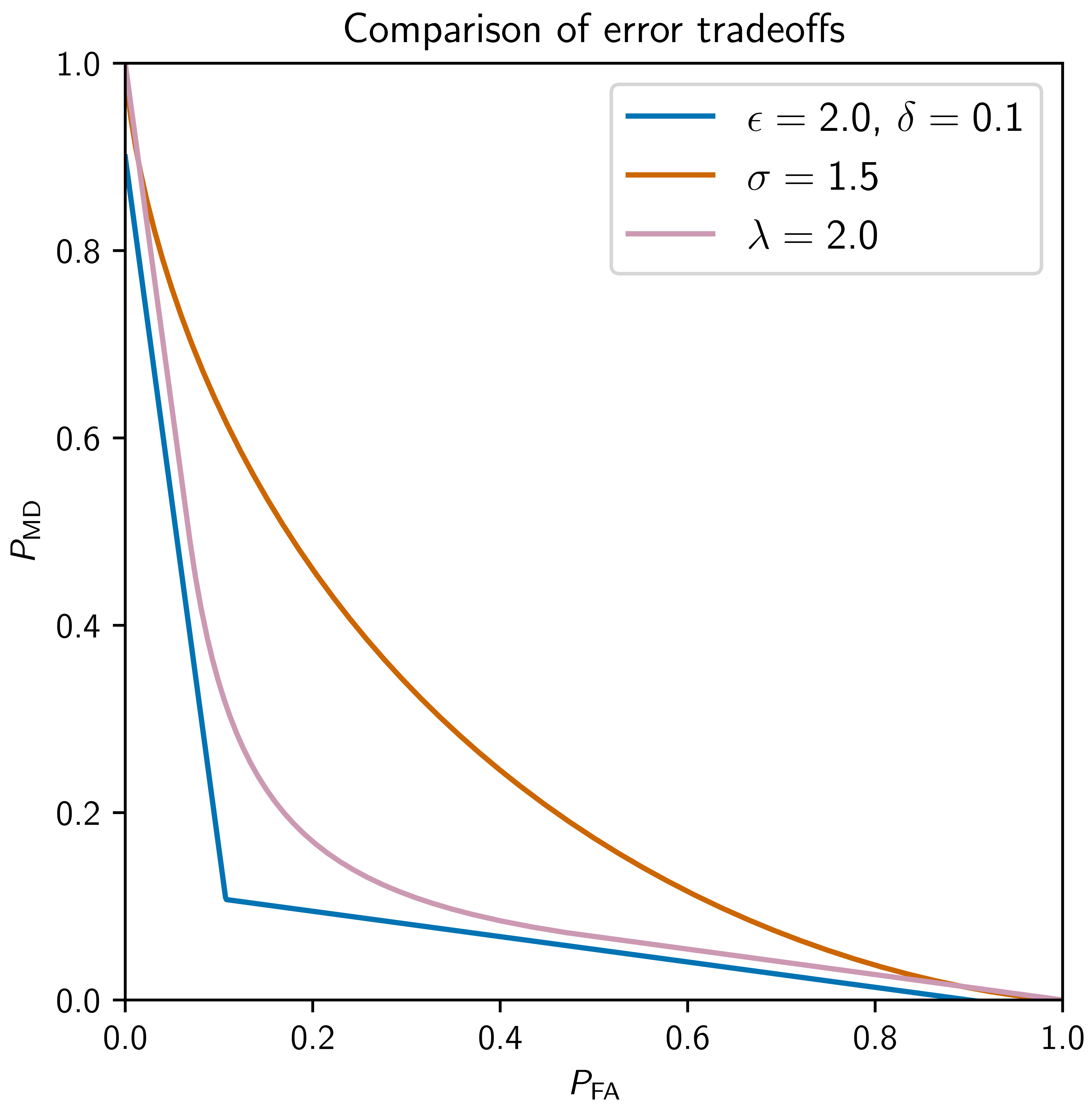
Tradeoffs between P_{FA} and P_{MD}

We get more privacy when the hypothesis test is “hard”

- A **privacy guarantee** is made by the tradeoff between probabilities of
  - **false alarm** (Type I error) and
  - **missed detection** (Type 2 error)
- If the likelihood ratio is small, the test will have a higher error.
- We can use a version of the ROC curve to visualize the kinds of guarantees.

Tradeoffs between P_{FA} and P_{MD}

We get more privacy when the hypothesis test is "hard"



- A **privacy guarantee** is made by the tradeoff between probabilities of
  - **false alarm** (Type I error) and
  - **missed detection** (Type 2 error)
- If the likelihood ratio is small, the test will have a higher error.
- We can use a version of the ROC curve to visualize the kinds of guarantees.

Tradeoffs between P_{FA} and P_{MD}

Commonly used noise distributions do not meet the lower bound

Wasserman, Zhou (2010) Kairouz, Oh, Vishwanath (2015)

Tradeoffs between P_{FA} and P_{MD}

Commonly used noise distributions do not meet the lower bound

A **privacy guarantee** is made by the tradeoff between probabilities of **false alarm** (Type I error/false positive) and **missed detection** (Type 2 error/false negative).

Wasserman, Zhou (2010) Kairouz, Oh, Vishwanath (2015)

Tradeoffs between P_{FA} and P_{MD}

Commonly used noise distributions do not meet the lower bound

A **privacy guarantee** is made by the tradeoff between probabilities of **false alarm** (Type I error/false positive) and **missed detection** (Type 2 error/false negative).

The DP definition prescribes a **piecewise linear lower bound** on the error tradeoff:

Wasserman, Zhou (2010) Kairouz, Oh, Vishwanath (2015)

Tradeoffs between P_{FA} and P_{MD}

Commonly used noise distributions do not meet the lower bound

A **privacy guarantee** is made by the tradeoff between probabilities of **false alarm** (Type I error/false positive) and **missed detection** (Type 2 error/false negative).

The DP definition prescribes a **piecewise linear lower bound** on the error tradeoff:

$$P_{\text{FA}} + e^{\epsilon} P_{\text{FA}} \geq 1 - \delta$$

Wasserman, Zhou (2010) Kairouz, Oh, Vishwanath (2015)

Tradeoffs between P_{FA} and P_{MD}

Commonly used noise distributions do not meet the lower bound

A **privacy guarantee** is made by the tradeoff between probabilities of **false alarm** (Type I error/false positive) and **missed detection** (Type 2 error/false negative).

The DP definition prescribes a **piecewise linear lower bound** on the error tradeoff:

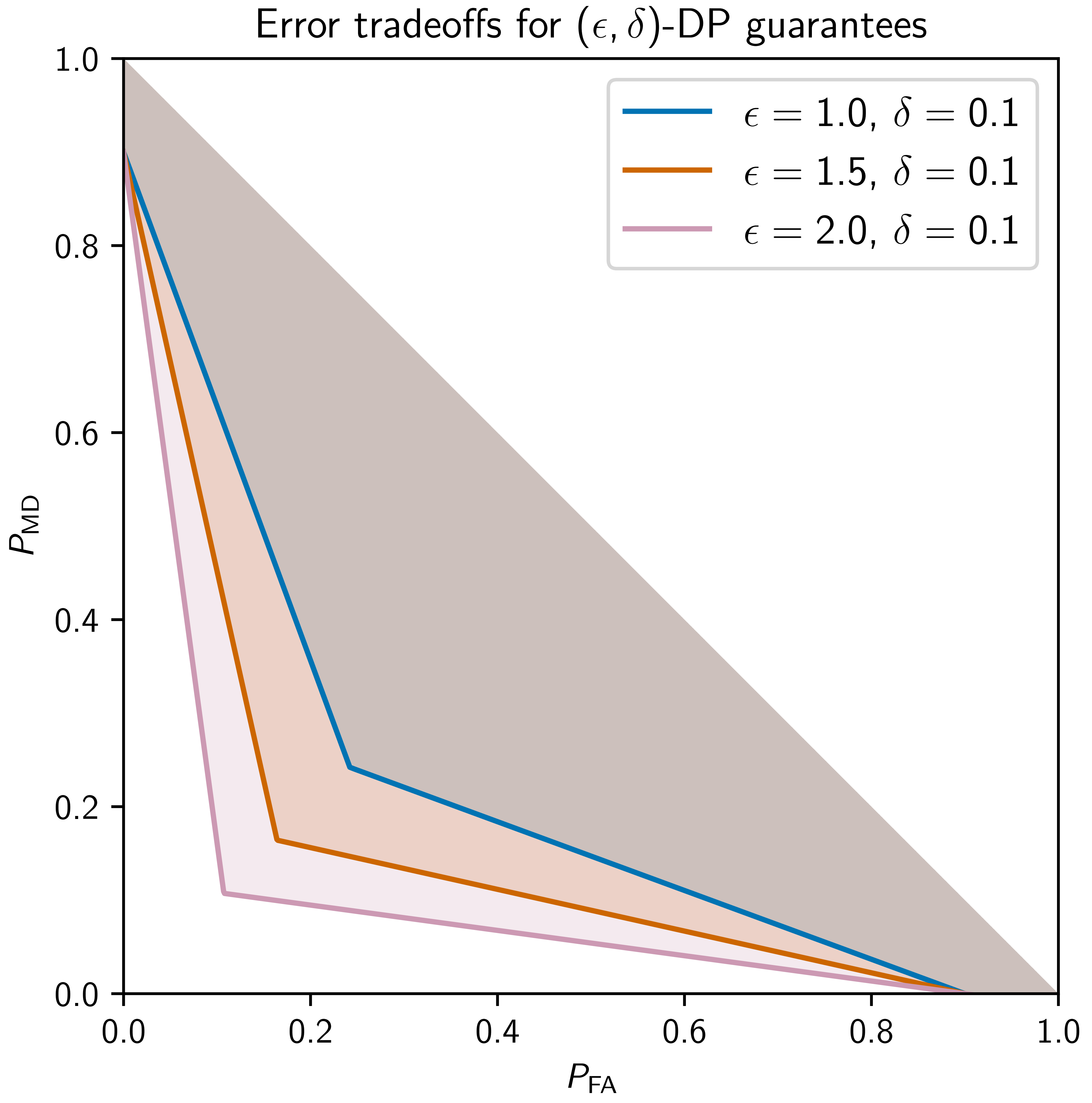
$$P_{\mathrm{FA}} + e^{\epsilon} P_{\mathrm{FA}} \geq 1 - \delta$$

$$e^{\epsilon} P_{\mathrm{FA}} + P_{\mathrm{MD}} \geq 1 - \delta$$

Wasserman, Zhou (2010) Kairouz, Oh, Vishwanath (2015)

Tradeoffs between P_{FA} and P_{MD}

Commonly used noise distributions do not meet the lower bound



A **privacy guarantee** is made by the tradeoff between probabilities of **false alarm** (Type I error/false positive) and **missed detection** (Type 2 error/false negative).

The DP definition prescribes a **piecewise linear lower bound** on the error tradeoff:

$$P_{\mathrm{FA}} + e^{\epsilon} P_{\mathrm{FA}} \geq 1 - \delta$$

$$e^{\epsilon} P_{\mathrm{FA}} + P_{\mathrm{MD}} \geq 1 - \delta$$

Wasserman, Zhou (2010) Kairouz, Oh, Vishwanath (2015)

Privacy versus testing

We get to design the test!

Privacy versus testing

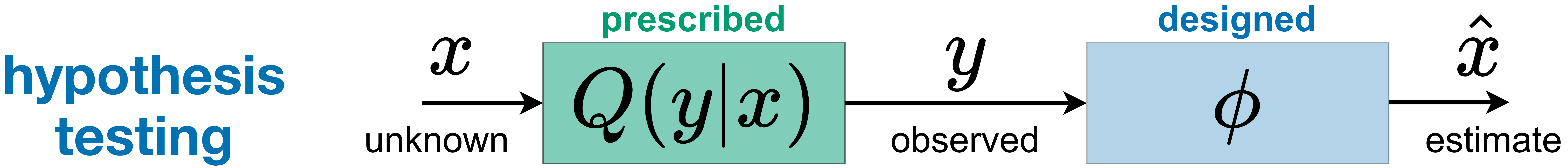
We get to design the test!

The key difference between **hypothesis testing** (as we usually encounter it) and **(differential) privacy** is that we get to design the **likelihoods** but not the **test**!

Privacy versus testing

We get to design the test!

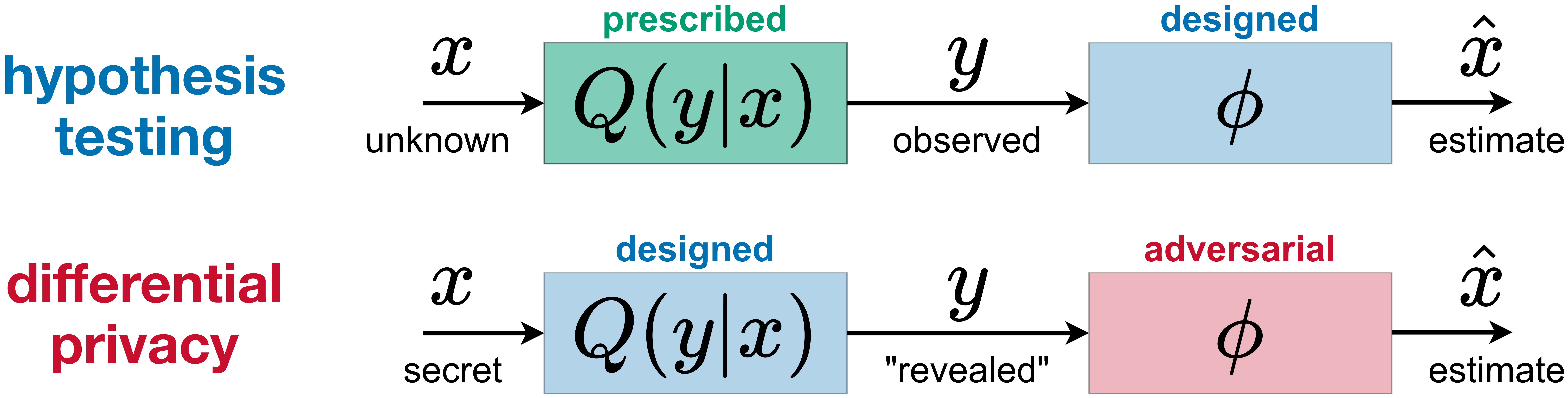
The key difference between **hypothesis testing** (as we usually encounter it) and **(differential) privacy** is that we get to design the likelihoods but not the **test**!



Privacy versus testing

We get to design the test!

The key difference between **hypothesis testing** (as we usually encounter it) and **(differential) privacy** is that we get to design the **likelihoods** but not the **test**!



Comparing hypothesis tests

Asking if one tradeoff curve is above another

Comparing hypothesis tests

Asking if one tradeoff curve is above another

We can sometimes compare two hypothesis testing problems by **comparing the optimal tradeoff curves**.

Comparing hypothesis tests

Asking if one tradeoff curve is above another

We can sometimes compare two hypothesis testing problems by **comparing the optimal tradeoff curves**.

$$\mathcal{H}_0 : Y \sim \mu \qquad \mathcal{H}_1 : Y \sim \nu$$

Comparing hypothesis tests

Asking if one tradeoff curve is above another

We can sometimes compare two hypothesis testing problems by **comparing the optimal tradeoff curves**.

$$\mathcal{H}_0 : Y \sim \mu \qquad \mathcal{H}_1 : Y \sim \nu$$

$$\mathcal{H}'_0 : Y \sim \mu' \qquad \mathcal{H}'_1 : Y \sim \nu'$$

Comparing hypothesis tests

Asking if one tradeoff curve is above another

We can sometimes compare two hypothesis testing problems by **comparing the optimal tradeoff curves**.

$$\mathscr{H}_0 : Y \sim \mu \qquad \mathscr{H}_1 : Y \sim \nu$$

$$\mathscr{H}'_0 : Y \sim \mu' \qquad \mathscr{H}'_1 : Y \sim \nu'$$

If the curve for the test (μ', ν') is above the curve for (μ, ν) then the test (μ', ν') is "harder" = more private:

Comparing hypothesis tests

Asking if one tradeoff curve is above another

We can sometimes compare two hypothesis testing problems by **comparing the optimal tradeoff curves**.

$$\mathscr{H}_0 : Y \sim \mu \qquad \mathscr{H}_1 : Y \sim \nu$$

$$\mathscr{H}'_0 : Y \sim \mu' \qquad \mathscr{H}'_1 : Y \sim \nu'$$

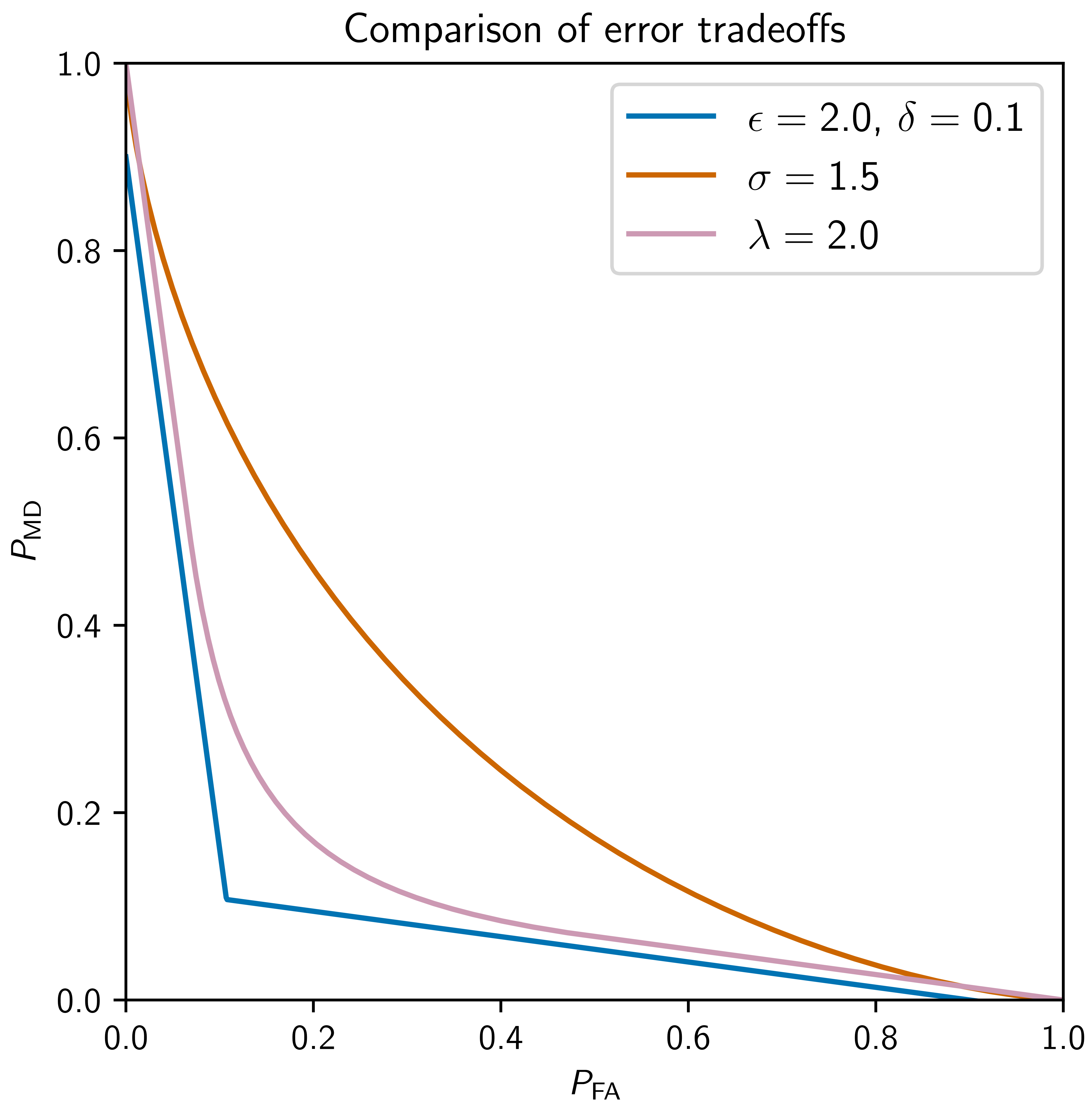
If the curve for the test (μ', ν') is above the curve for (μ, ν) then the test (μ', ν') is "harder" = more private:

$$T(\mu, \nu) \leq T(\mu', \nu')$$

Dong, Roth, Su (2022), Blackwell (1950/51/53), Raginsky (2011)

Comparing hypothesis tests

Asking if one tradeoff curve is above another



We can sometimes compare two hypothesis testing problems by **comparing the optimal tradeoff curves**.

$$\mathscr{H}_0 : Y \sim \mu \qquad \mathscr{H}_1 : Y \sim \nu$$

$$\mathscr{H}'_0 : Y \sim \mu' \qquad \mathscr{H}'_1 : Y \sim \nu'$$

If the curve for the test (μ', ν') is above the curve for (μ, ν) then the test (μ', ν') is "harder" = more private:

$$T(\mu, \nu) \leq T(\mu', \nu')$$

Dong, Roth, Su (2022), Blackwell (1950/51/53), Raginsky (2011)

Sunset Across Ryōgoku Bridge from Ommayagashi

御厩川岸より両国橋夕陽見

Ommayagashi yori Ryōgoku-bashi yūhi-mi

Vista 2

differential privacy the normal way

The “standard” approach to explaining DP

Neighboring databases of individual records

The “standard” approach to explaining DP

Neighboring databases of individual records

In the textbook approach to describing DP we have several ingredients:

The “standard” approach to explaining DP

Neighboring databases of individual records

In the textbook approach to describing DP we have several ingredients:

1. **Data space:** $\mathscr{X}$, often modeled as records from n individuals.

The “standard” approach to explaining DP

Neighboring databases of individual records

In the textbook approach to describing DP we have several ingredients:

1. **Data space:** $\mathscr{X}$, often modeled as records from n individuals.

2. **Neighborhood relationship $\sim$:** for $\mathbf{x}, \mathbf{x}' \in \mathscr{X}$ we write $\mathbf{x} \sim \mathbf{x}'$ if they are “neighbors”.

The "standard" approach to explaining DP

Neighboring databases of individual records

In the textbook approach to describing DP we have several ingredients:

1. **Data space:** $\mathcal{X}$, often modeled as records from n individuals.

2. **Neighborhood relationship $\sim$:** for $\mathbf{x}, \mathbf{x}' \in \mathcal{X}$ we write $\mathbf{x} \sim \mathbf{x}'$ if they are "neighbors".
   - Example: each person has 1 bit so $\mathcal{X} = \{0,1\}^n$ and $\mathbf{x} \sim \mathbf{x}'$ if they differ in one position.

The "standard" approach to explaining DP

Neighboring databases of individual records

In the textbook approach to describing DP we have several ingredients:

1. **Data space:** $\mathscr{X}$, often modeled as records from n individuals.
2. **Neighborhood relationship $\sim$:** for $\mathbf{x}, \mathbf{x}' \in \mathscr{X}$ we write $\mathbf{x} \sim \mathbf{x}'$ if they are "neighbors".
   - Example: each person has 1 bit so $\mathscr{X} = \{0,1\}^n$ and $\mathbf{x} \sim \mathbf{x}'$ if they differ in one position.
3. **Output space:** $\mathscr{Y}$, depends on the functionality/what we want to release.

The "standard" approach to explaining DP

Neighboring databases of individual records

In the textbook approach to describing DP we have several ingredients:

1. **Data space:** $\mathscr{X}$, often modeled as records from n individuals.

2. **Neighborhood relationship $\sim$:** for $\mathbf{x}, \mathbf{x}' \in \mathscr{X}$ we write $\mathbf{x} \sim \mathbf{x}'$ if they are "neighbors".
   - Example: each person has 1 bit so $\mathscr{X} = \{0,1\}^n$ and $\mathbf{x} \sim \mathbf{x}'$ if they differ in one position.

3. **Output space:** $\mathscr{Y}$, depends on the functionality/what we want to release.
   - Example: If we want the average of data $\mathscr{X} = [0,1]^n$, we have $\mathscr{Y} = [0,1]$.

The “standard” approach to explaining DP

Neighboring databases of individual records

In the textbook approach to describing DP we have several ingredients:

1. **Data space:** $\mathscr{X}$, often modeled as records from n individuals.
2. **Neighborhood relationship $\sim$:** for $\mathbf{x}, \mathbf{x}' \in \mathscr{X}$ we write $\mathbf{x} \sim \mathbf{x}'$ if they are “neighbors”.
   - Example: each person has 1 bit so $\mathscr{X} = \{0,1\}^n$ and $\mathbf{x} \sim \mathbf{x}'$ if they differ in one position.
3. **Output space:** $\mathscr{Y}$, depends on the functionality/what we want to release.
   - Example: If we want the average of data $\mathscr{X} = [0,1]^n$, we have $\mathscr{Y} = [0,1]$.
   - Example: If we want to train a classifier using data $\mathscr{X} = \{\mathbb{R}^d \times \{0,1\}\}^n$, $\mathscr{Y} = \mathbb{R}^d$.

The “standard” approach to explaining DP

Neighboring databases of individual records

In the textbook approach to describing DP we have several ingredients:

1. **Data space:** $\mathscr{X}$, often modeled as records from n individuals.

2. **Neighborhood relationship $\sim$:** for $\mathbf{x}, \mathbf{x}' \in \mathscr{X}$ we write $\mathbf{x} \sim \mathbf{x}'$ if they are “neighbors”.
   - Example: each person has 1 bit so $\mathscr{X} = \{0,1\}^n$ and $\mathbf{x} \sim \mathbf{x}'$ if they differ in one position.

3. **Output space:** $\mathscr{Y}$, depends on the functionality/what we want to release.
   - Example: If we want the average of data $\mathscr{X} = [0,1]^n$, we have $\mathscr{Y} = [0,1]$.
   - Example: If we want to train a classifier using data $\mathscr{X} = \{\mathbb{R}^d \times \{0,1\}\}^n$, $\mathscr{Y} = \mathbb{R}^d$.

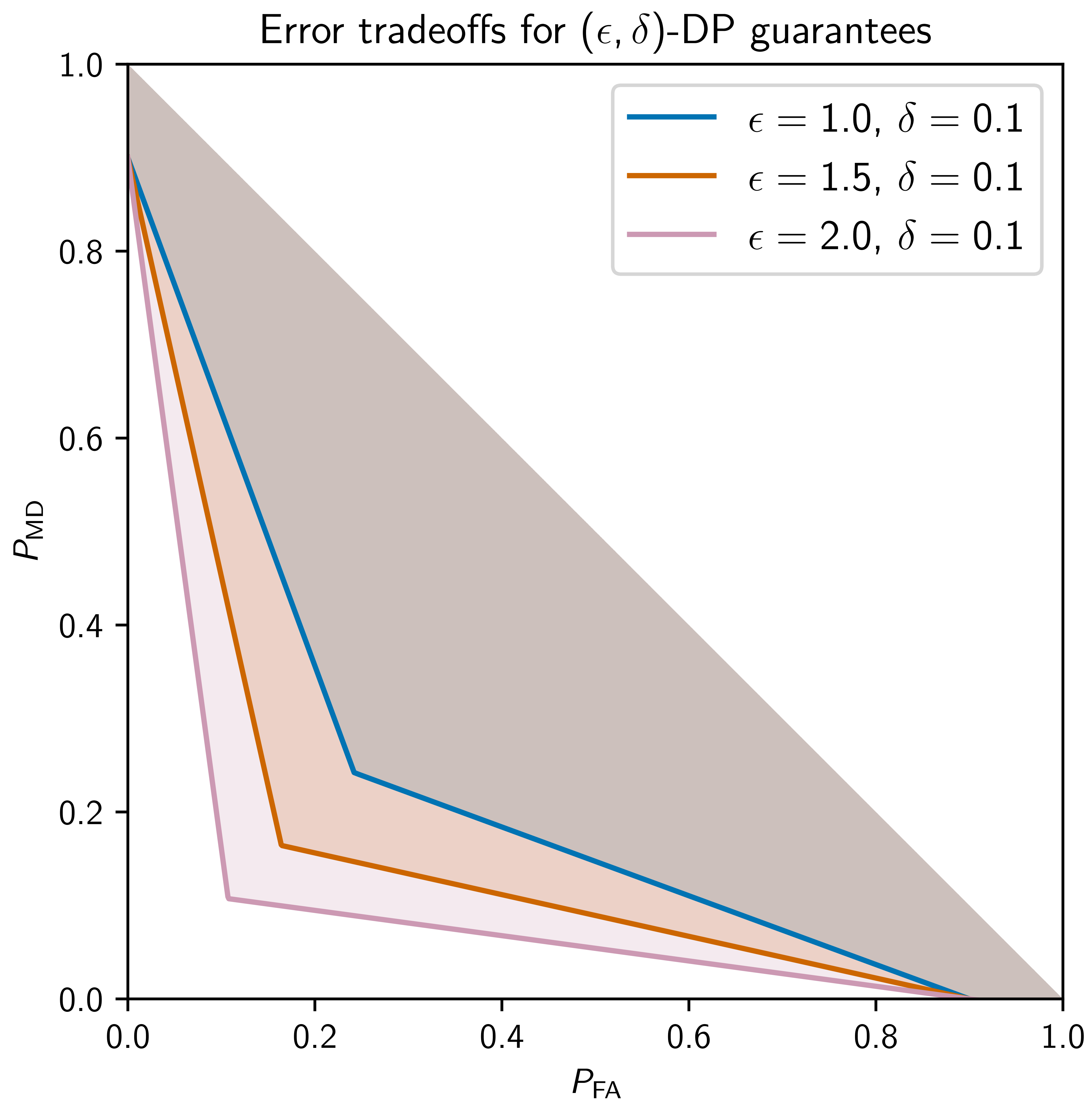
4. **Algorithm:** a randomized map/conditional distribution/channel $Q: \mathscr{X} \to \mathscr{Y}$.

DP makes many hypothesis tests hard

Protecting many single bits simultaneously

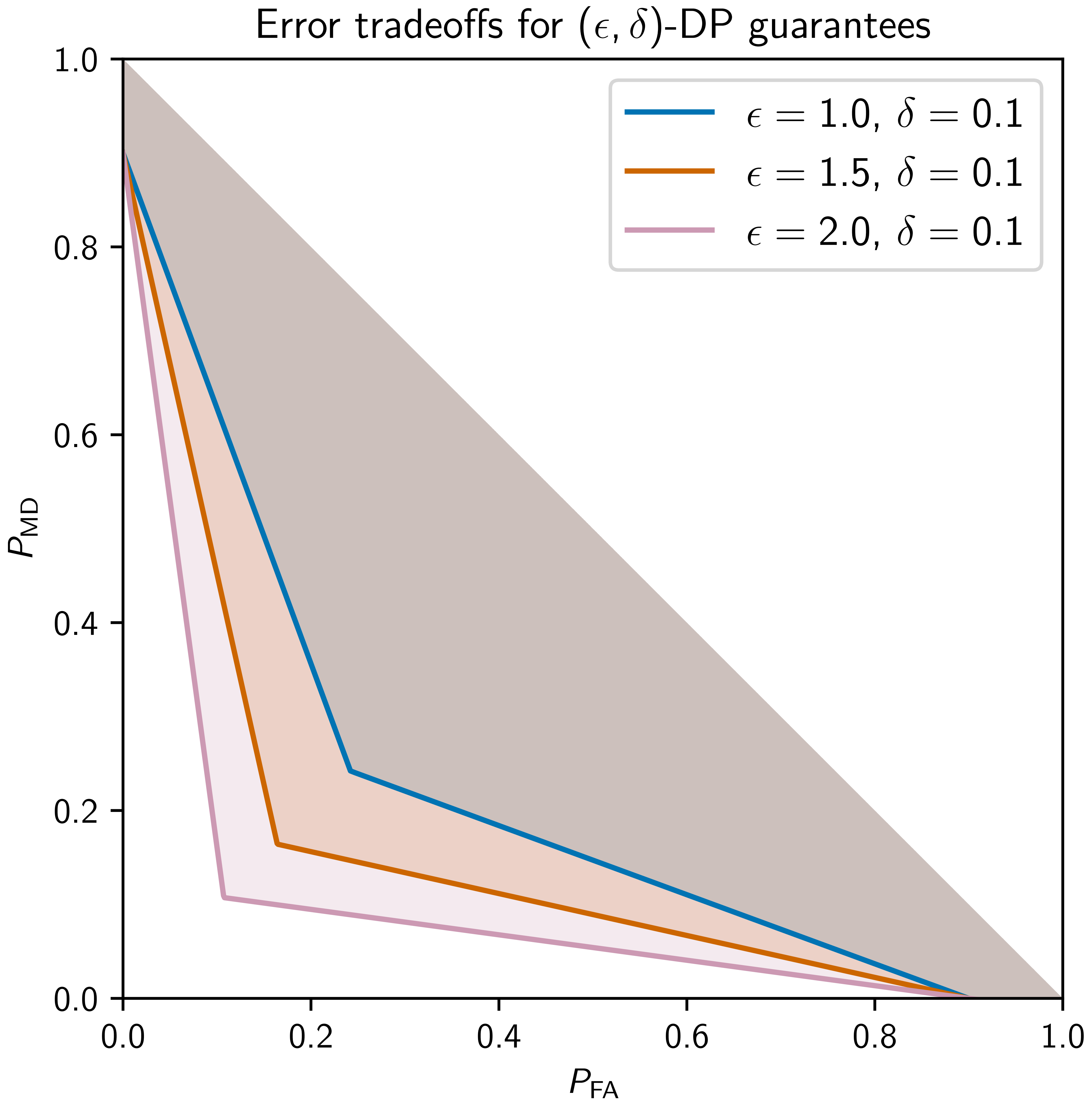
DP makes many hypothesis tests hard

Protecting many single bits simultaneously



DP makes many hypothesis tests hard

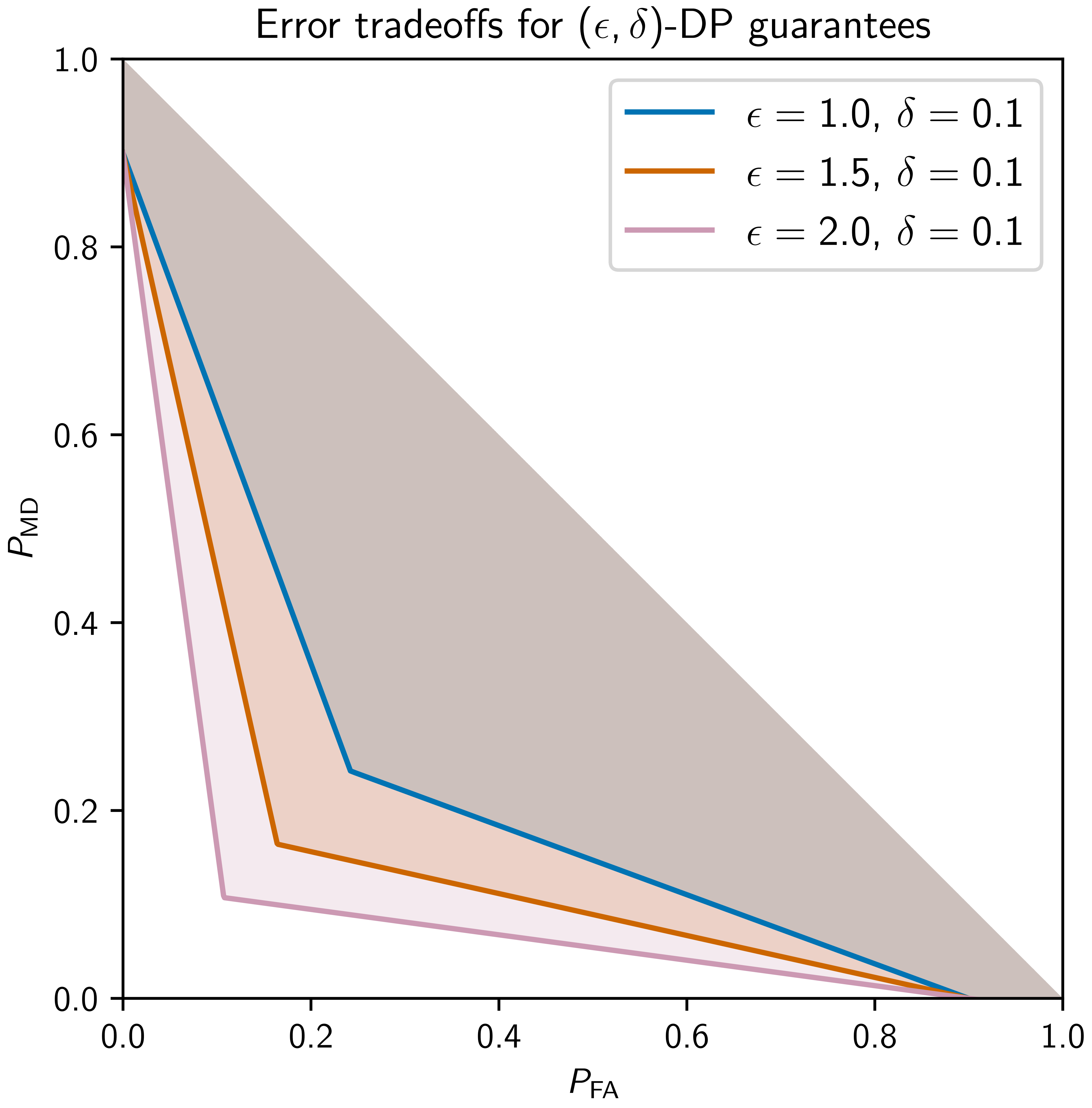
Protecting many single bits simultaneously



Compared to our single private bit b, in DP we want many hypothesis tests to hard for the adversary. For every $\mathbf{x} \sim \mathbf{x}'$ the test

DP makes many hypothesis tests hard

Protecting many single bits simultaneously

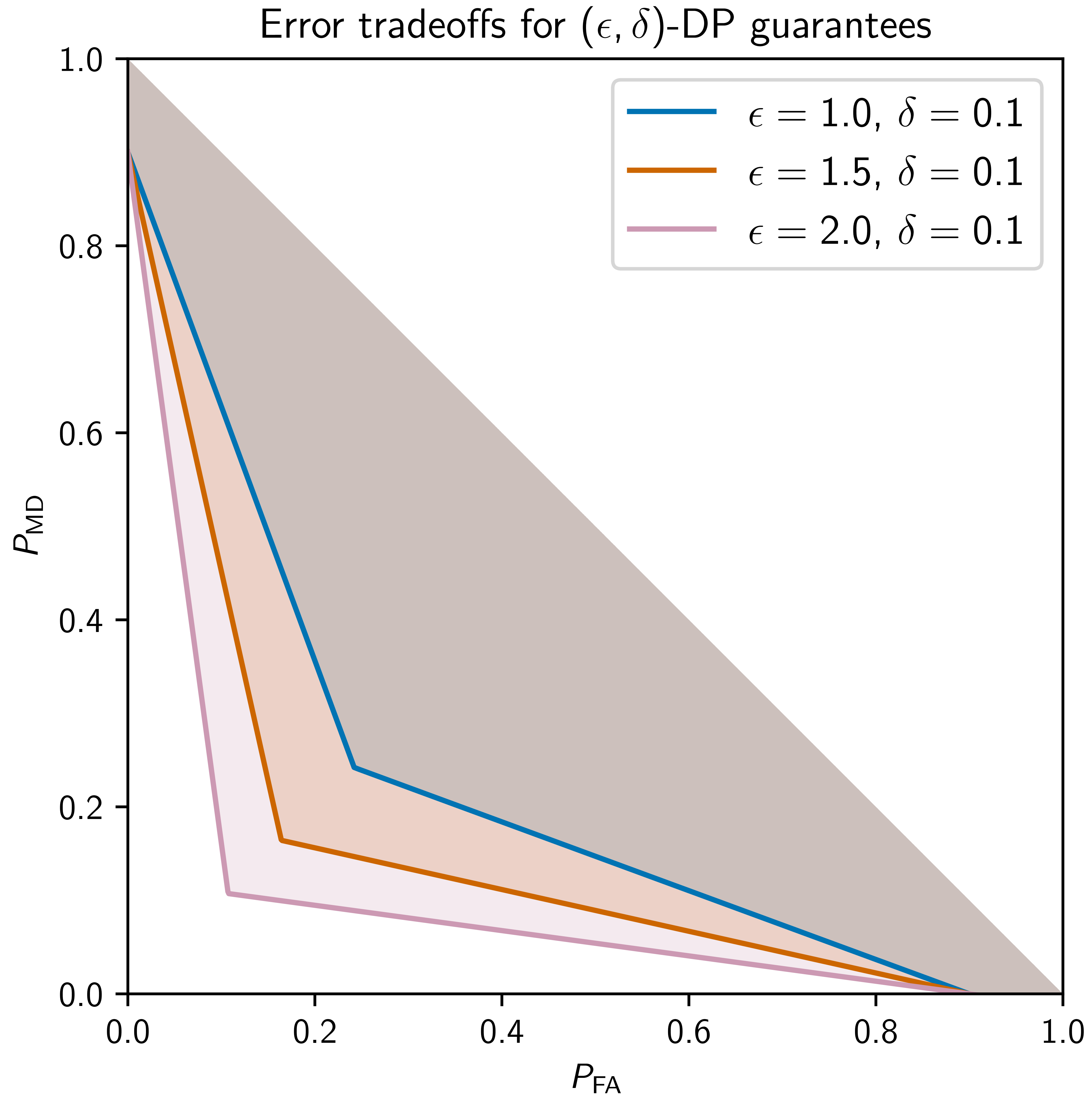


Compared to our single private bit b, in DP we want many hypothesis tests to hard for the adversary. For every $\mathbf{x} \sim \mathbf{x}'$ the test

$$\mathscr{H}_0 \colon \mathbf{y} \sim Q(\,\cdot\,|\,\mathbf{x})$$

DP makes many hypothesis tests hard

Protecting many single bits simultaneously



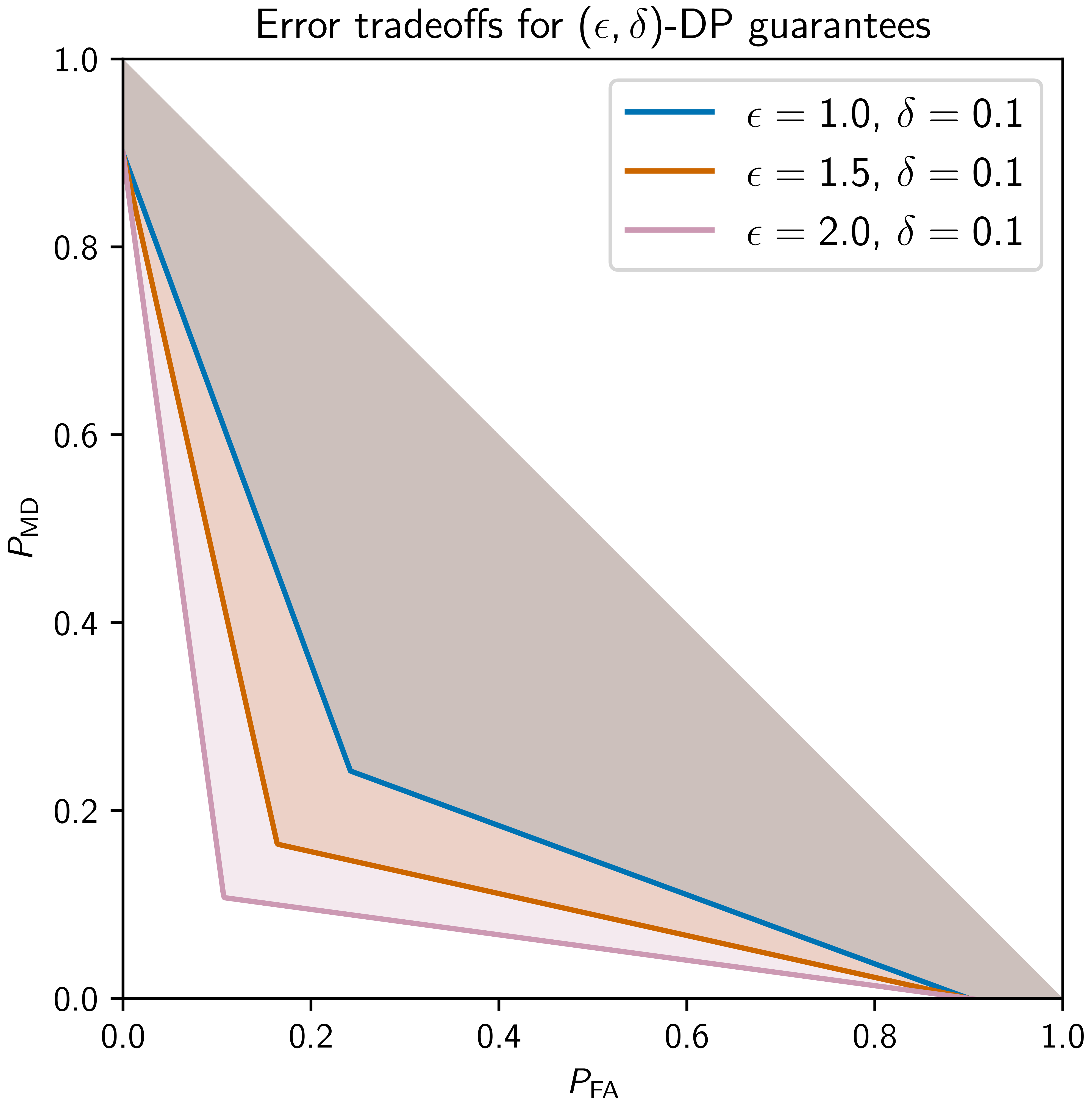
Compared to our single private bit b, in DP we want many hypothesis tests to hard for the adversary. For every $\mathbf{x} \sim \mathbf{x}'$ the test

$$\mathscr{H}_0 : \mathbf{y} \sim Q(\,\cdot\,|\,\mathbf{x})$$

$$\mathscr{H}_1 : \mathbf{y} \sim Q(\,\cdot\,|\,\mathbf{x}')$$

DP makes many hypothesis tests hard

Protecting many single bits simultaneously



Compared to our single private bit b, in DP we want many hypothesis tests to hard for the adversary. For every $\mathbf{x} \sim \mathbf{x}'$ the test

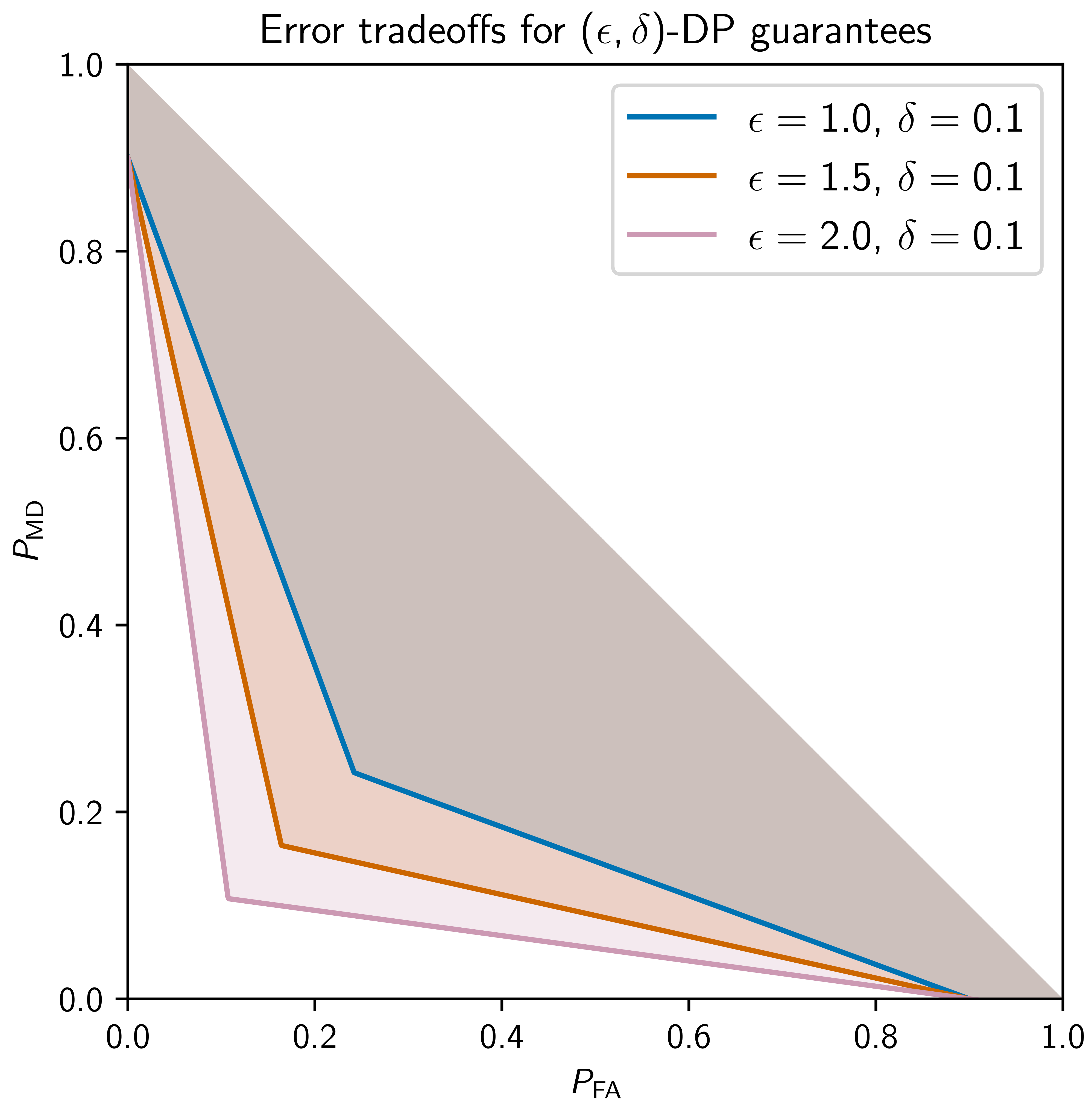
$$\mathscr{H}_0 : \mathbf{y} \sim Q(\cdot \,|\, \mathbf{x})$$

$$\mathscr{H}_1 : \mathbf{y} \sim Q(\cdot \,|\, \mathbf{x}')$$

should have a large probability of error.

DP makes many hypothesis tests hard

Protecting many single bits simultaneously



Compared to our single private bit b, in DP we want many hypothesis tests to hard for the adversary. For every $\mathbf{x} \sim \mathbf{x}'$ the test

$$\mathscr{H}_0 : \mathbf{y} \sim Q(\cdot \,|\, \mathbf{x})$$

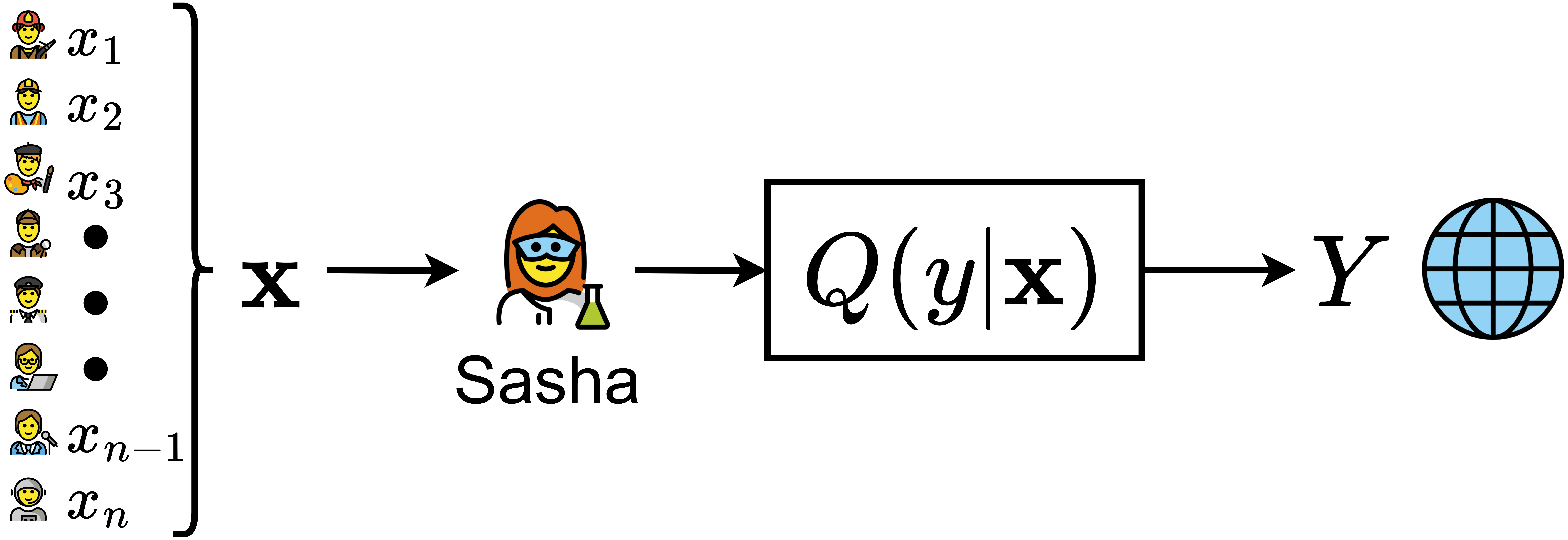
$$\mathscr{H}_1 : \mathbf{y} \sim Q(\cdot \,|\, \mathbf{x}')$$

should have a large probability of error.

When can we do this? When neighboring data sets make similar output distributions.

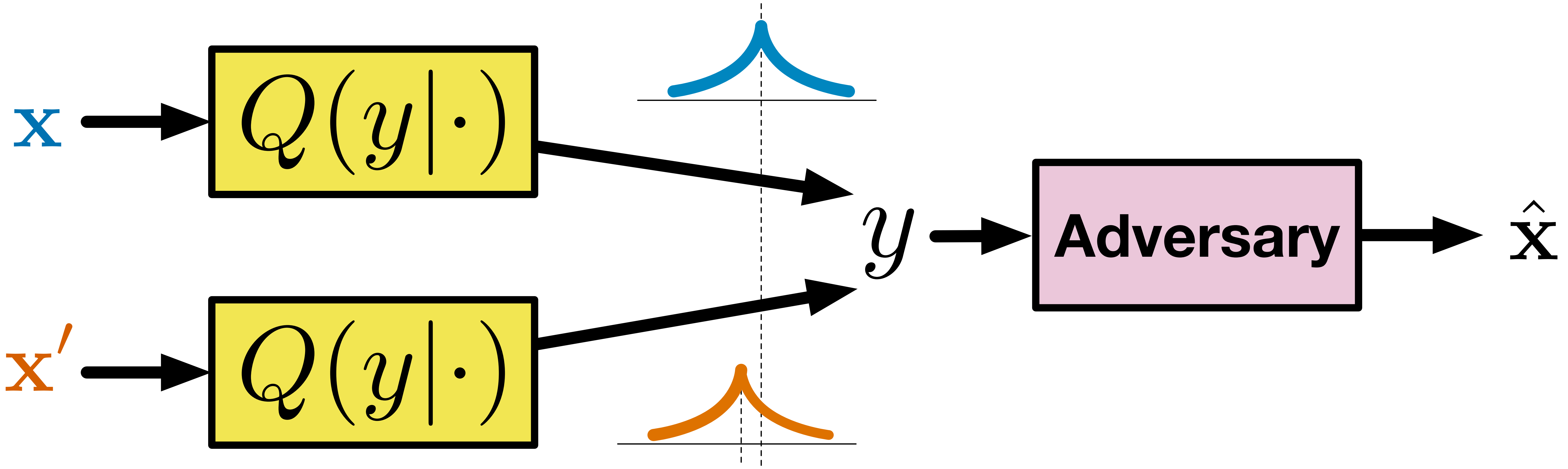
In a snapshot

Replacing a single bit with a database



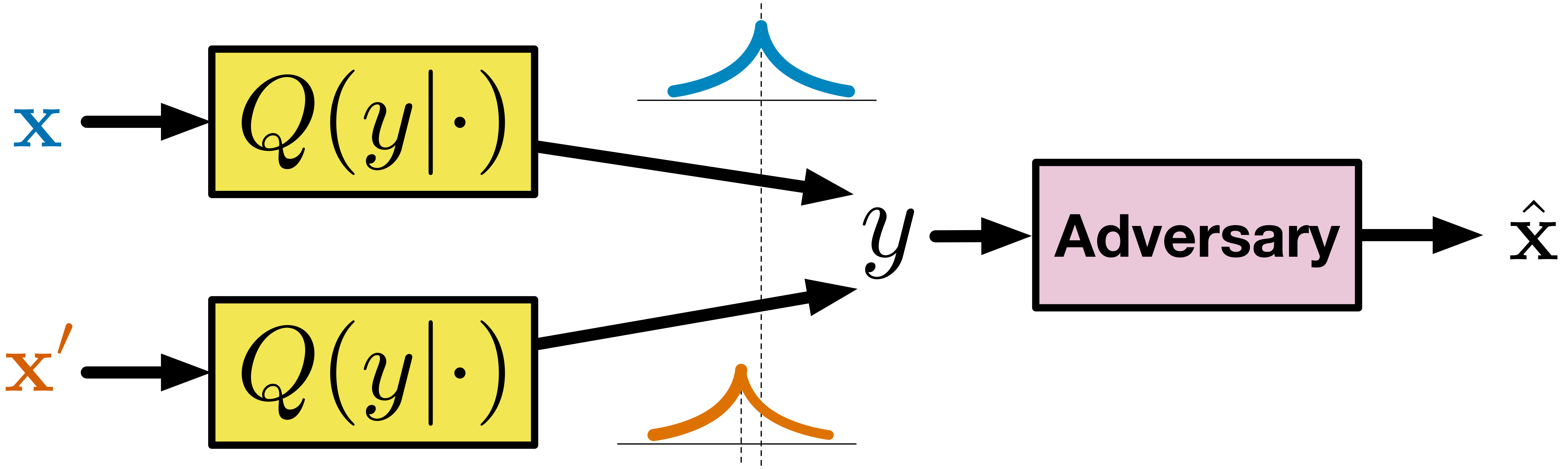
The hypothesis testing in DP

DP is a property of the channel



The hypothesis testing in DP

DP is a property of the channel



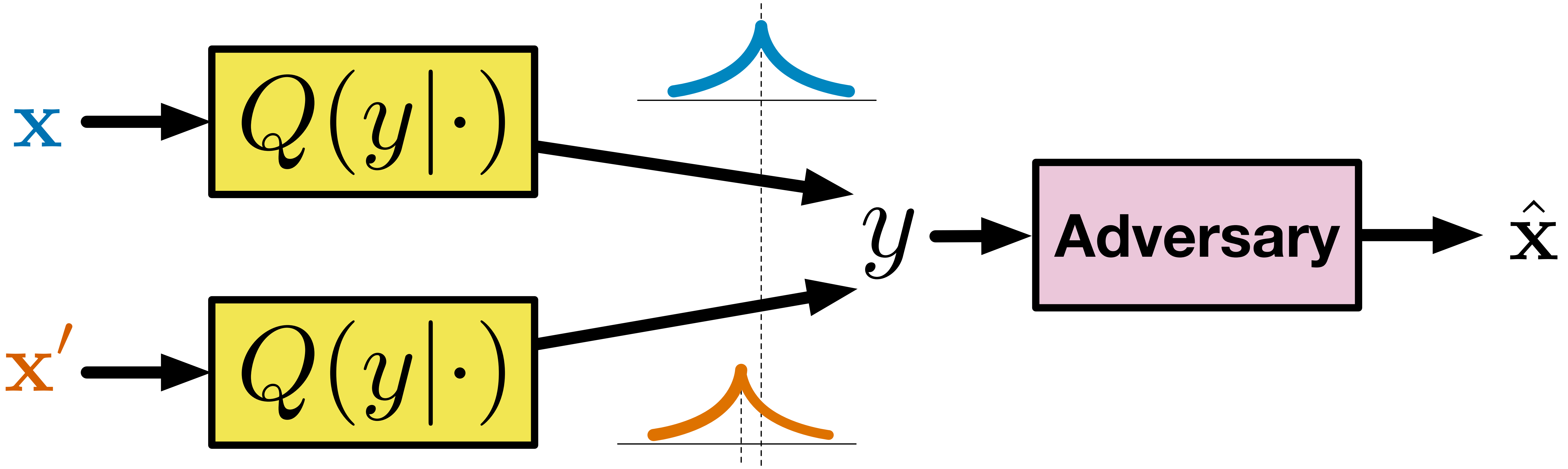
A channel/"mechanism"/algorithm Q is (ϵ, δ)**-differentially private** if

$$Q(\mathcal{T} \mid \mathbf{x}) \leq e^{\epsilon} Q(\mathcal{T} \mid \mathbf{x}') + \delta$$

For all measurable subsets $\mathcal{T} \subseteq \mathcal{Y}$ and all $\mathbf{x} \sim \mathbf{x}'$.

The hypothesis testing in DP

DP is a property of the channel



A channel/"mechanism"/algorithm Q is **(ϵ, δ)-differentially private** if

$$Q(\mathscr{T} \,|\, \mathbf{x}) \leq e^{\epsilon} Q(\mathscr{T} \,|\, \mathbf{x}') + \delta$$

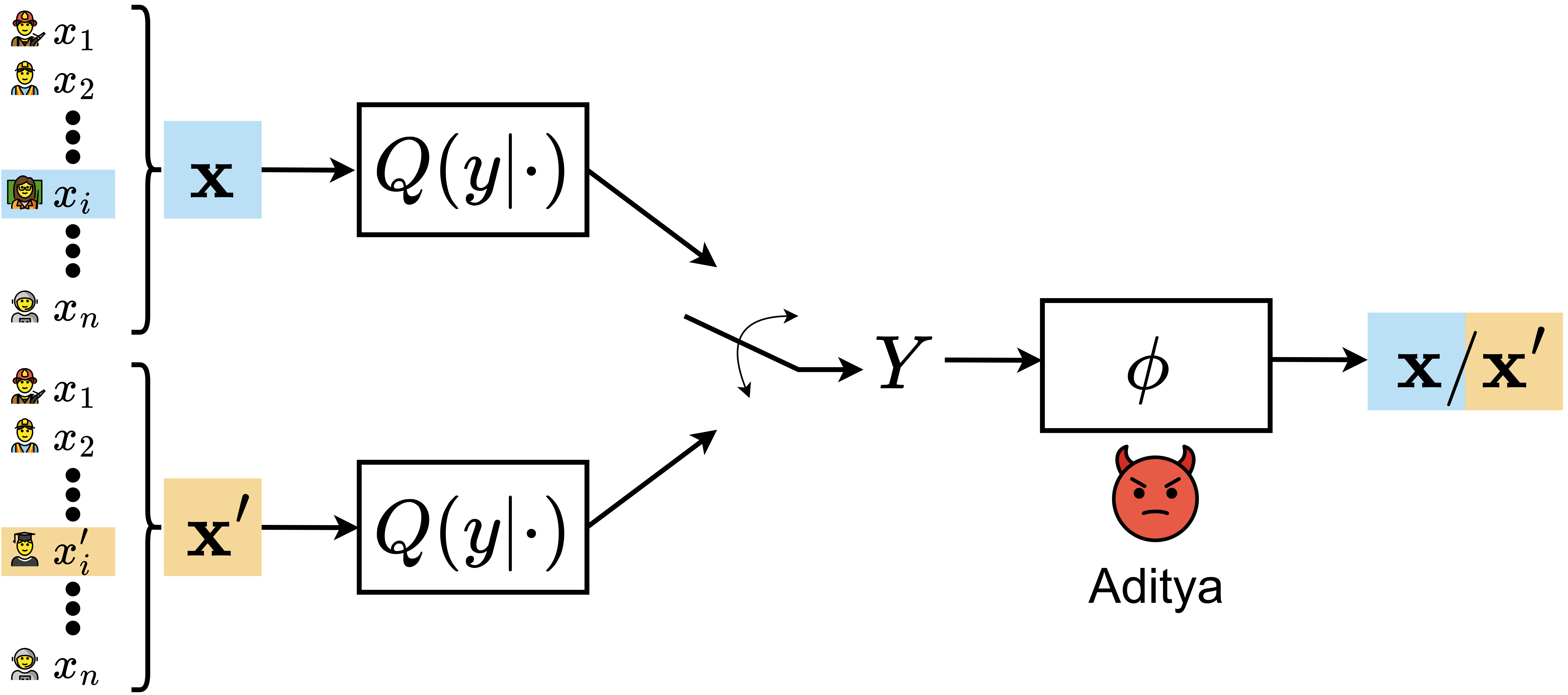
For all measurable subsets $\mathscr{T} \subseteq \mathscr{Y}$ and all $\mathbf{x} \sim \mathbf{x}'$.

[Dwork-Kenthapadi-McSherry-Mironov-Naor 2006]

[Wasserman-Zhou 2010]

Neighboring datasets in a picture

The adversary's hypothesis test



Some notes on the definition

DP's underlying assumptions are slightly different

Some notes on the definition

DP's underlying assumptions are slightly different

- **Differential privacy is a pretty stringent requirement:** The probability of any event is similar under $\mathbf{x}$ and any other neighboring $\mathbf{x}'$.

Some notes on the definition

DP's underlying assumptions are slightly different

- **Differential privacy is a pretty stringent requirement:** The probability of any event is similar under $\mathbf{x}$ and any other neighboring $\mathbf{x}'$.
- **Guarantee is on conditional probabilities given the data:** same risk holds regardless of side information (e.g. linkage attacks).

Some notes on the definition

DP's underlying assumptions are slightly different

- **Differential privacy is a pretty stringent requirement:** The probability of any event is similar under $\mathbf{x}$ and any other neighboring $\mathbf{x}'$.
- **Guarantee is on conditional probabilities given the data:** same risk holds regardless of side information (e.g. linkage attacks).
- **There is no statistical assumption on the data:** $\mathbf{x}$ is not drawn from some (prior) distribution.

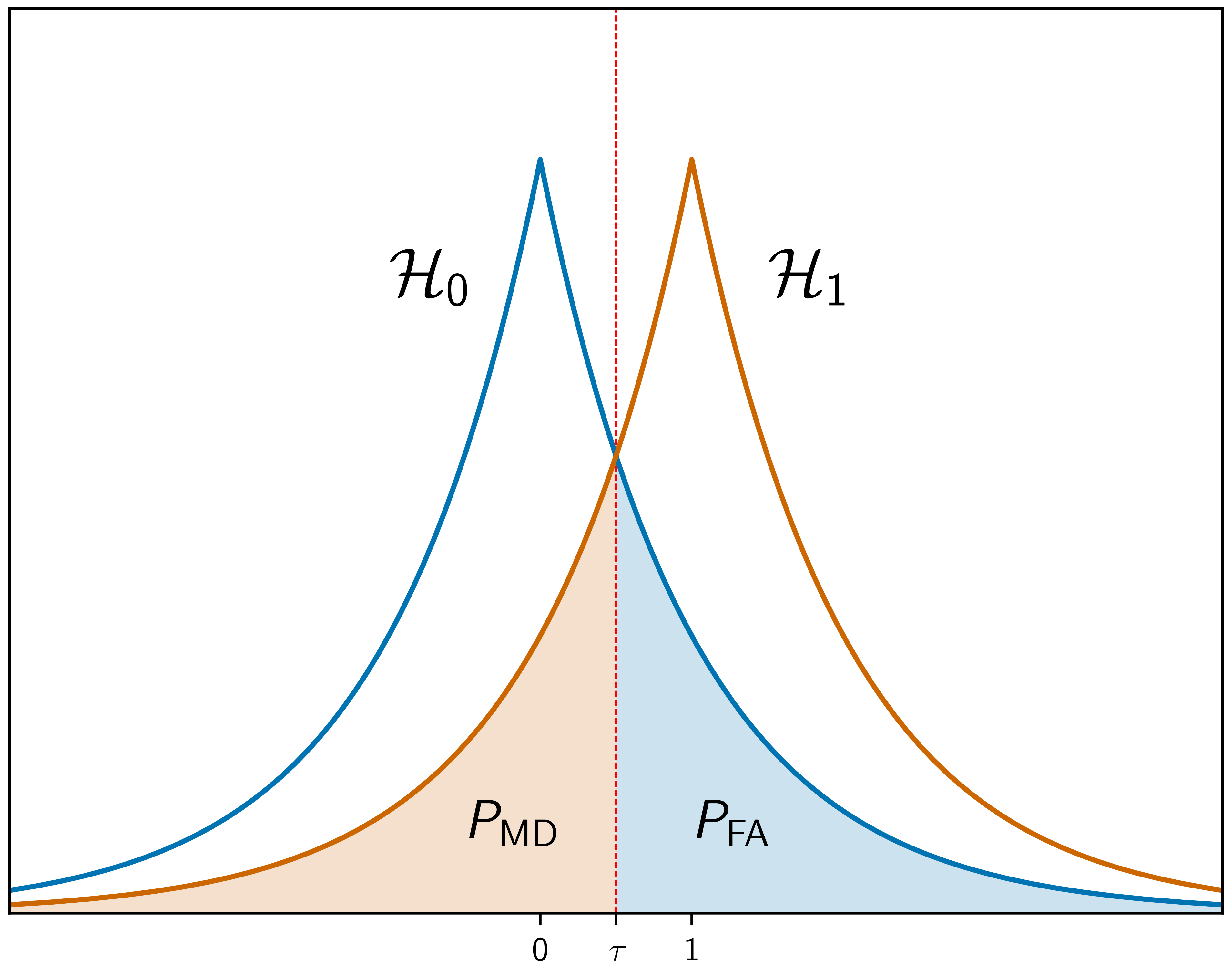
Some notes on the definition

DP's underlying assumptions are slightly different

- **Differential privacy is a pretty stringent requirement:** The probability of any event is similar under $\mathbf{x}$ and any other neighboring $\mathbf{x}'$.
- **Guarantee is on conditional probabilities given the data:** same risk holds regardless of side information (e.g. linkage attacks).
- **There is no statistical assumption on the data:** $\mathbf{x}$ is not drawn from some (prior) distribution.
- **The data itself is considered identifying:** no notion of some parts being personally identifiable information (PII) and others not.

Sensitivity of scalar functions

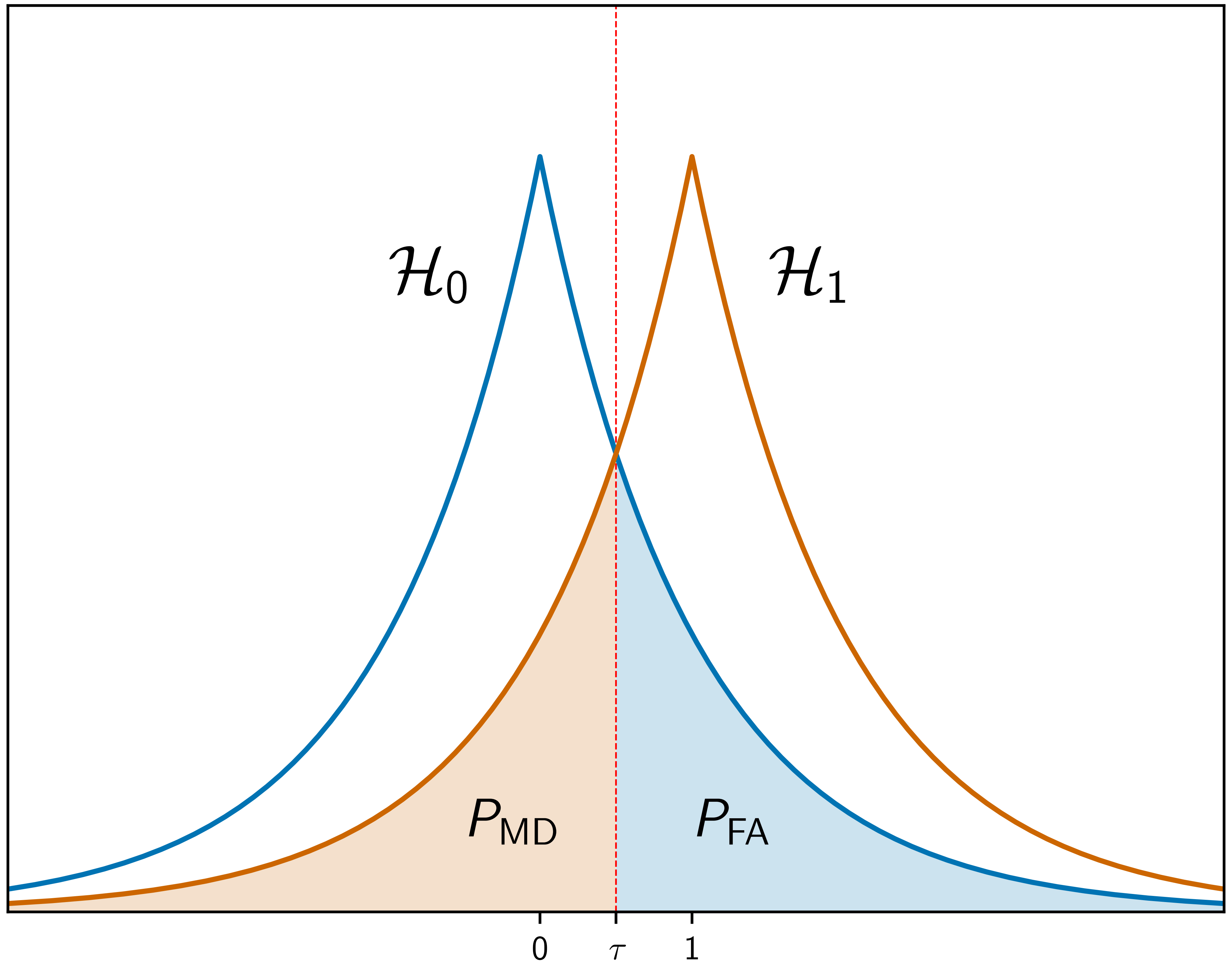
Understanding the distance between hypotheses



Sensitivity of scalar functions

Understanding the distance between hypotheses

In DP, we usually want to approximate some function of the data.

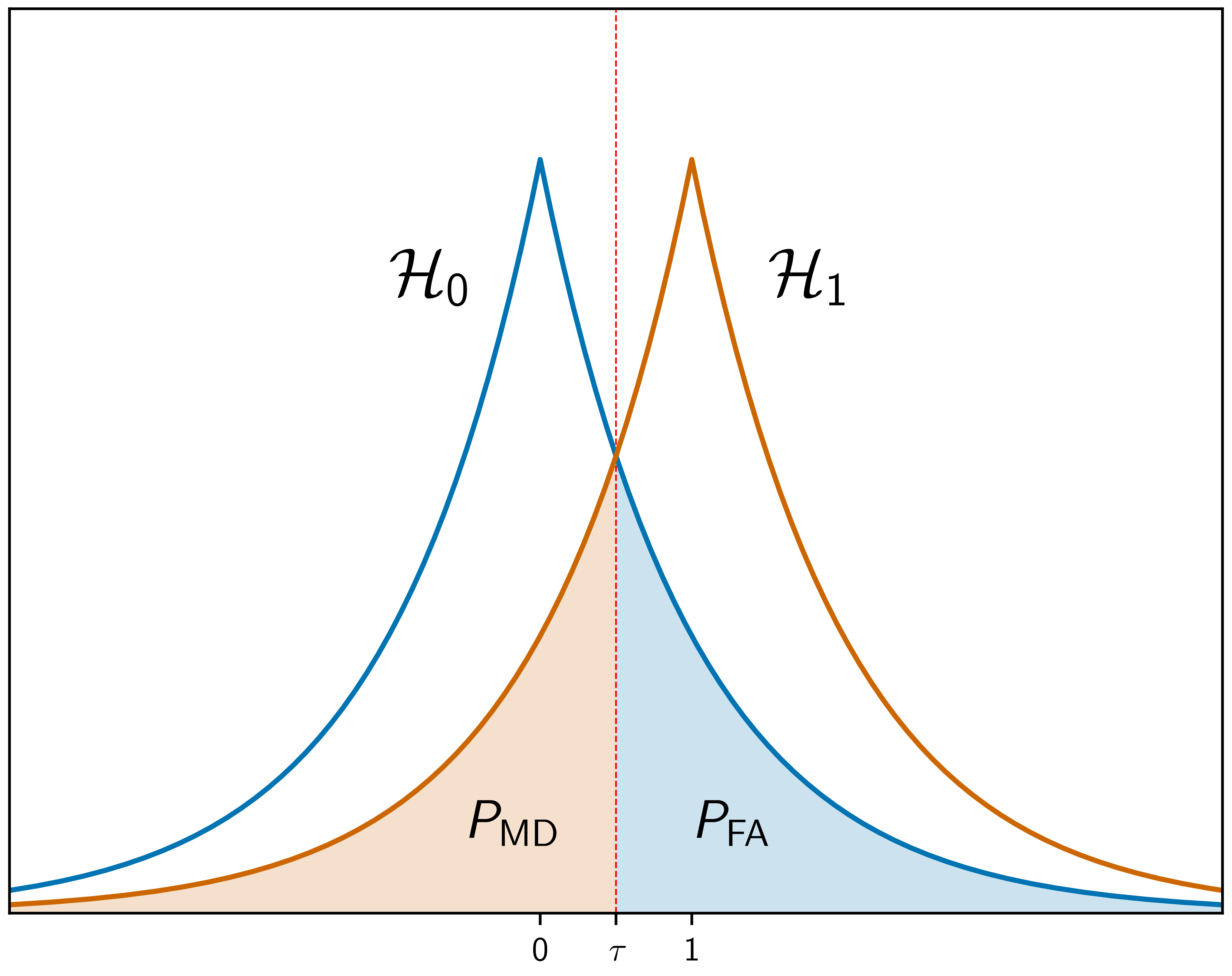


Sensitivity of scalar functions

Understanding the distance between hypotheses

In DP, we usually want to approximate some function of the data.

Suppose we want $f : \mathscr{X} \to \mathbb{R}$. We want the test to be hard for any pair $(\mathbf{x}, \mathbf{x}')$ which are "neighbors" $(\mathbf{x} \sim \mathbf{x}')$.



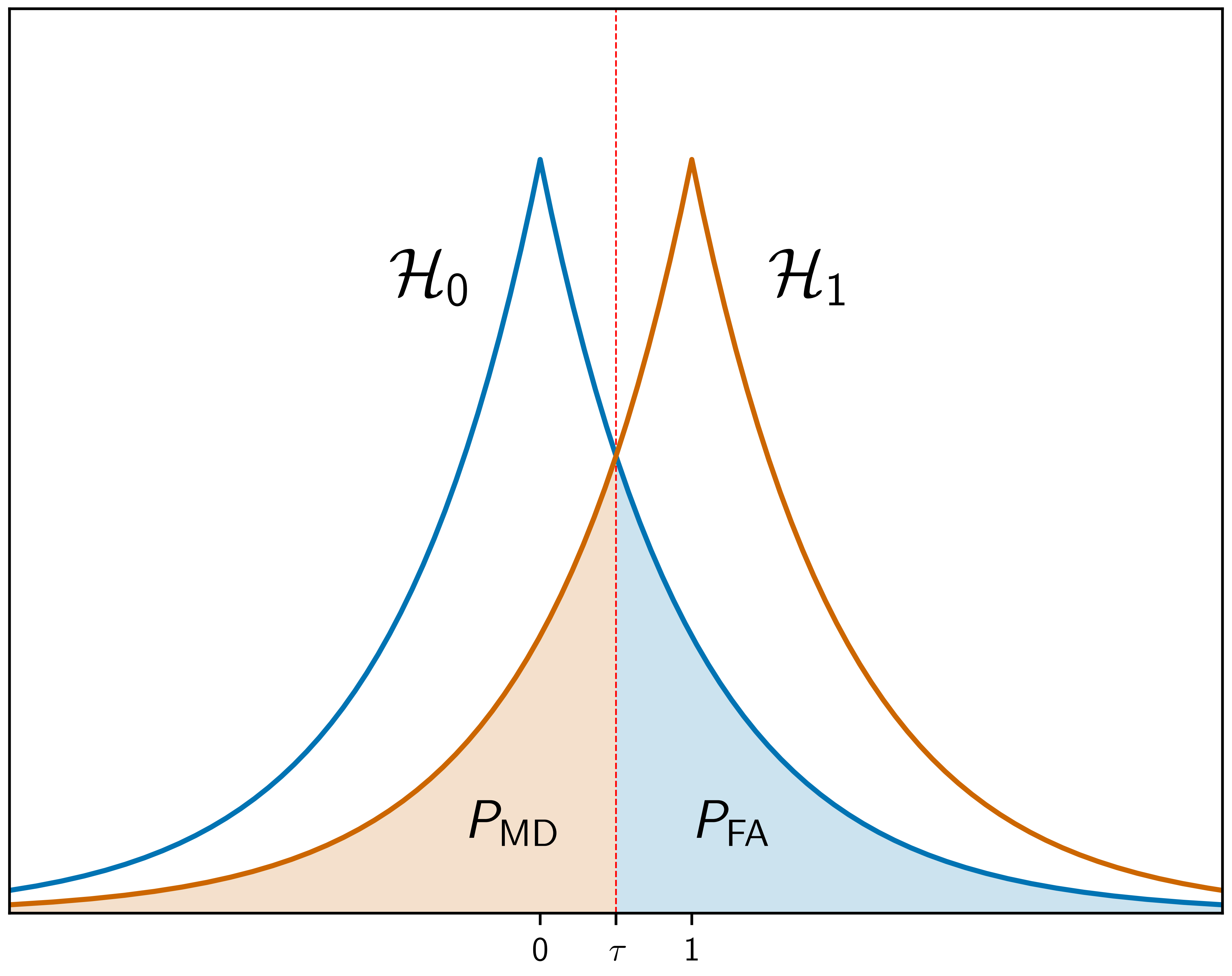
Sensitivity of scalar functions

Understanding the distance between hypotheses

In DP, we usually want to approximate some function of the data.

Suppose we want $f : \mathscr{X} \to \mathbb{R}$. We want the test to be hard for any pair $(\mathbf{x}, \mathbf{x}')$ which are "neighbors" $(\mathbf{x} \sim \mathbf{x}')$.

If $f(\,\cdot\,)$ is small for all neighbors, this should be easier.



Sensitivity of scalar functions

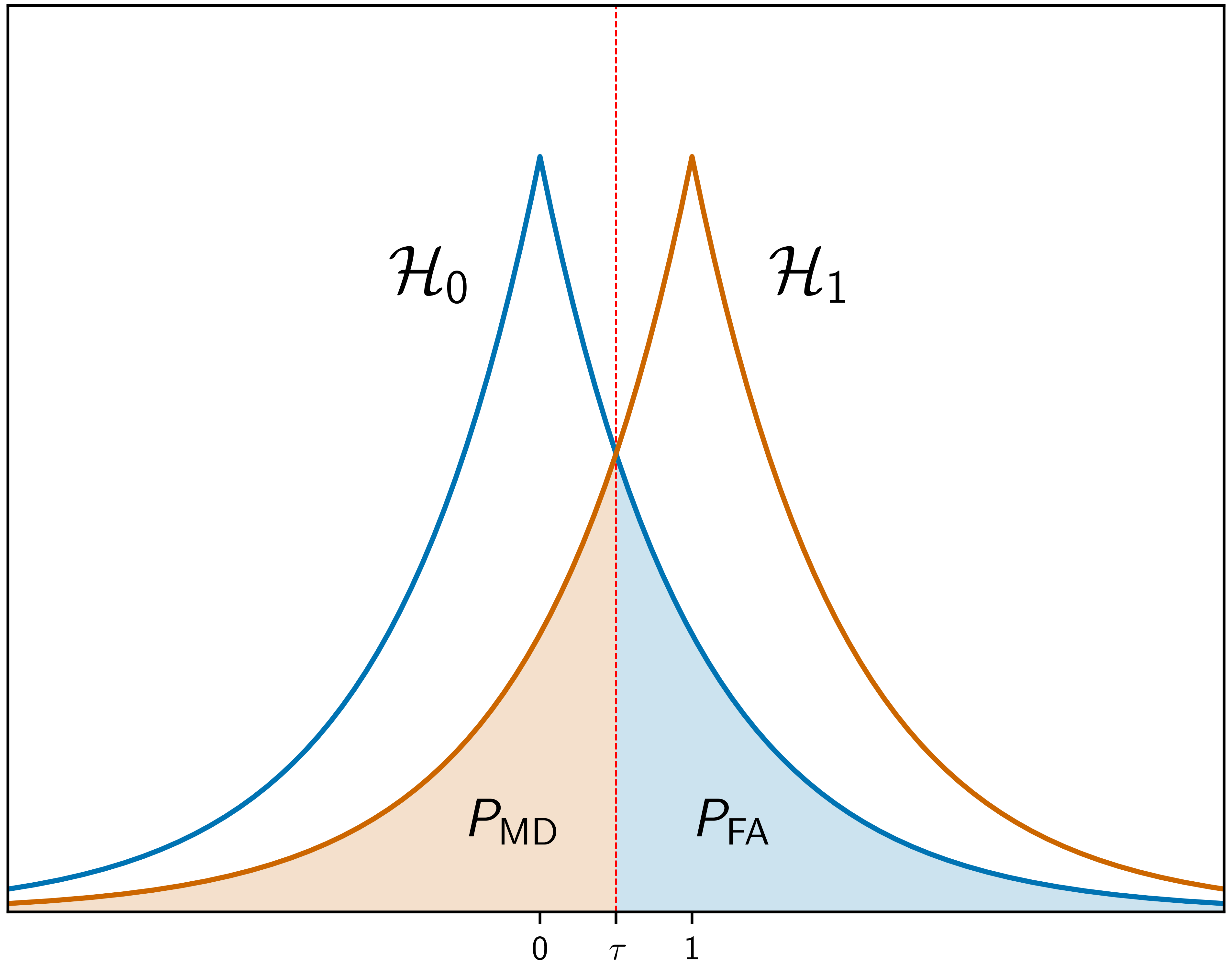
Understanding the distance between hypotheses

In DP, we usually want to approximate some function of the data.

Suppose we want $f : \mathcal{X} \to \mathbb{R}$. We want the test to be hard for any pair $(\mathbf{x}, \mathbf{x}')$ which are "neighbors" $(\mathbf{x} \sim \mathbf{x}')$.

If $f(\cdot)$ is small for all neighbors, this should be easier.

Example: $f(x) = \frac{1}{n}\sum_{i=1}^{n} x_i$ can change by at most $\frac{1}{n}$ for $x_i \in [0,1]$.



Sensitivity of scalar functions

Understanding the distance between hypotheses

Sensitivity of scalar functions

Understanding the distance between hypotheses

The **global sensitivity** of $f(\,\cdot\,)$ is

Sensitivity of scalar functions

Understanding the distance between hypotheses

The **global sensitivity** of $f(\,\cdot\,)$ is

$$\Delta(f) = \max_{\mathbf{x} \sim \mathbf{x}'} \left| f(\mathbf{x}) - f(\mathbf{x}') \right| .$$

Sensitivity of scalar functions

Understanding the distance between hypotheses

The **global sensitivity** of $f(\,\cdot\,)$ is

$$\Delta(f) = \max_{\mathbf{x} \sim \mathbf{x}'} \left| f(\mathbf{x}) - f(\mathbf{x}') \right|.$$

If we use additive noise (like in the Laplace and Gaussian case) we have

Sensitivity of scalar functions

Understanding the distance between hypotheses

The **global sensitivity** of $f(\,\cdot\,)$ is

$$\Delta(f) = \max_{\mathbf{x} \sim \mathbf{x}'} \left| f(\mathbf{x}) - f(\mathbf{x}') \right|.$$

If we use additive noise (like in the Laplace and Gaussian case) we have

$$\mathcal{H}_0 : Z \sim p(z - f(\mathbf{x})) \qquad \text{vs.} \qquad \mathcal{H}_1 : Z \sim p(z - f(\mathbf{x}'))$$

Sensitivity of scalar functions

Understanding the distance between hypotheses

The **global sensitivity** of $f(\,\cdot\,)$ is

$$\Delta(f) = \max_{\mathbf{x} \sim \mathbf{x}'} \left| f(\mathbf{x}) - f(\mathbf{x}') \right| .$$

If we use additive noise (like in the Laplace and Gaussian case) we have

$$\mathcal{H}_0 : Z \sim p(z - f(\mathbf{x})) \qquad \text{vs.} \qquad \mathcal{H}_1 : Z \sim p(z - f(\mathbf{x}'))$$

We can make a guarantee for all “neighbors” if following test is hard:

Sensitivity of scalar functions

Understanding the distance between hypotheses

The **global sensitivity** of $f(\cdot)$ is

$$\Delta(f) = \max_{\mathbf{x} \sim \mathbf{x}'} \left| f(\mathbf{x}) - f(\mathbf{x}') \right| .$$

If we use additive noise (like in the Laplace and Gaussian case) we have

$$\mathscr{H}_0 : Z \sim p(z - f(\mathbf{x})) \qquad \text{vs.} \qquad \mathscr{H}_1 : Z \sim p(z - f(\mathbf{x}'))$$

We can make a guarantee for all "neighbors" if following test is hard:

$$\mathscr{H}_0 : Z \sim p(z) \qquad \text{vs.} \qquad \mathscr{H}_1 : Z \sim p(z - \Delta(f)).$$

Example: the sample mean

Computing the MSE as a function of privacy risk

Example: the sample mean

Computing the MSE as a function of privacy risk

Suppose we have data in $\mathscr{X} = [A, B]^n$ and want to estimate the mean:

Example: the sample mean

Computing the MSE as a function of privacy risk

Suppose we have data in $\mathcal{X} = [A, B]^n$ and want to estimate the mean:

$$\hat{\mu}(\mathbf{x}^n) = \frac{1}{n} \sum_{j=1}^{n} \mathbf{x}_j + Z$$

Example: the sample mean

Computing the MSE as a function of privacy risk

Suppose we have data in $\mathscr{X} = [A, B]^n$ and want to estimate the mean:

$$\hat{\mu}(\mathbf{x}^n) = \frac{1}{n}\sum_{j=1}^{n} \mathbf{x}_j + Z$$

- Sensitivity of $\hat{\mu}(\mathbf{x}^n)$ is $(B - A)/n$.

Example: the sample mean

Computing the MSE as a function of privacy risk

Suppose we have data in $\mathcal{X} = [A, B]^n$ and want to estimate the mean:

$$\hat{\mu}(\mathbf{x}^n) = \frac{1}{n} \sum_{j=1}^{n} \mathbf{x}_j + Z$$

- Sensitivity of $\hat{\mu}(\mathbf{x}^n)$ is $(B - A)/n$.
- $Z \sim$ Laplace$(n\epsilon/(B - A))$ will guarantee $(\epsilon, 0)$-DP.

Example: the sample mean

Computing the MSE as a function of privacy risk

Suppose we have data in $\mathscr{X} = [A, B]^n$ and want to estimate the mean:

$$\hat{\mu}(\mathbf{x}^n) = \frac{1}{n} \sum_{j=1}^{n} \mathbf{x}_j + Z$$

- Sensitivity of $\hat{\mu}(\mathbf{x}^n)$ is $(B - A)/n$.
- $Z \sim$ Laplace$(n\epsilon/(B - A))$ will guarantee $(\epsilon, 0)$-DP.
- MSE of $\hat{\mu}(\mathbf{x}^n)$ is $2/\lambda^2 = 2\frac{(B-A)^2}{n^2\epsilon^2}$.

Example: the sample mean

Computing the MSE as a function of privacy risk

Suppose we have data in $\mathscr{X} = [A, B]^n$ and want to estimate the mean:

$$\hat{\mu}(\mathbf{x}^n) = \frac{1}{n}\sum_{j=1}^{n} \mathbf{x}_j + Z$$

- Sensitivity of $\hat{\mu}(\mathbf{x}^n)$ is $(B - A)/n$.
- $Z \sim$ Laplace$(n\epsilon/(B - A))$ will guarantee $(\epsilon,0)$-DP.
- MSE of $\hat{\mu}(\mathbf{x}^n)$ is $2/\lambda^2 = 2\frac{(B-A)^2}{n^2\epsilon^2}$.

Example: the sample mean

Computing the MSE as a function of privacy risk

Suppose we have data in $\mathscr{X} = [A, B]^n$ and want to estimate the mean:

$$\hat{\mu}(\mathbf{x}^n) = \frac{1}{n} \sum_{j=1}^{n} \mathbf{x}_j + Z$$

- Sensitivity of $\hat{\mu}(\mathbf{x}^n)$ is $(B - A)/n$.
- $Z \sim$ Laplace$(n\epsilon/(B - A))$ will guarantee $(\epsilon, 0)$-DP.
- MSE of $\hat{\mu}(\mathbf{x}^n)$ is $2/\lambda^2 = 2\dfrac{(B - A)^2}{n^2\epsilon^2}$.

The privacy-utility tradeoff

How much do we lose when we guarantee privacy?

The privacy-utility tradeoff

How much do we lose when we guarantee privacy?

Adding $\text{Laplace}(\lambda)$ noise guarantees privacy, but at what cost? The MSE is:

The privacy-utility tradeoff

How much do we lose when we guarantee privacy?

Adding $\text{Laplace}(\lambda)$ noise guarantees privacy, but at what cost? The MSE is:

$$2/\lambda^2 = 2\frac{(B-A)^2}{n^2\epsilon^2}$$

The privacy-utility tradeoff

How much do we lose when we guarantee privacy?

Adding $\text{Laplace}(\lambda)$ noise guarantees privacy, but at what cost? The MSE is:

$$2/\lambda^2 = 2\frac{(B-A)^2}{n^2\epsilon^2}$$

So we can see that less privacy risk (smaller ϵ) induces more MSE.

The privacy-utility tradeoff

How much do we lose when we guarantee privacy?

Adding $\text{Laplace}(\lambda)$ noise guarantees privacy, but at what cost? The MSE is:

$$2/\lambda^2 = 2\frac{(B-A)^2}{n^2\epsilon^2}$$

So we can see that less privacy risk (smaller ϵ) induces more MSE.

We can try to optimize the privacy mechanism if we know the utility function (like squared error).

The privacy-utility tradeoff

How much do we lose when we guarantee privacy?

Adding $\text{Laplace}(\lambda)$ noise guarantees privacy, but at what cost? The MSE is:

$$2/\lambda^2 = 2\frac{(B-A)^2}{n^2\epsilon^2}$$

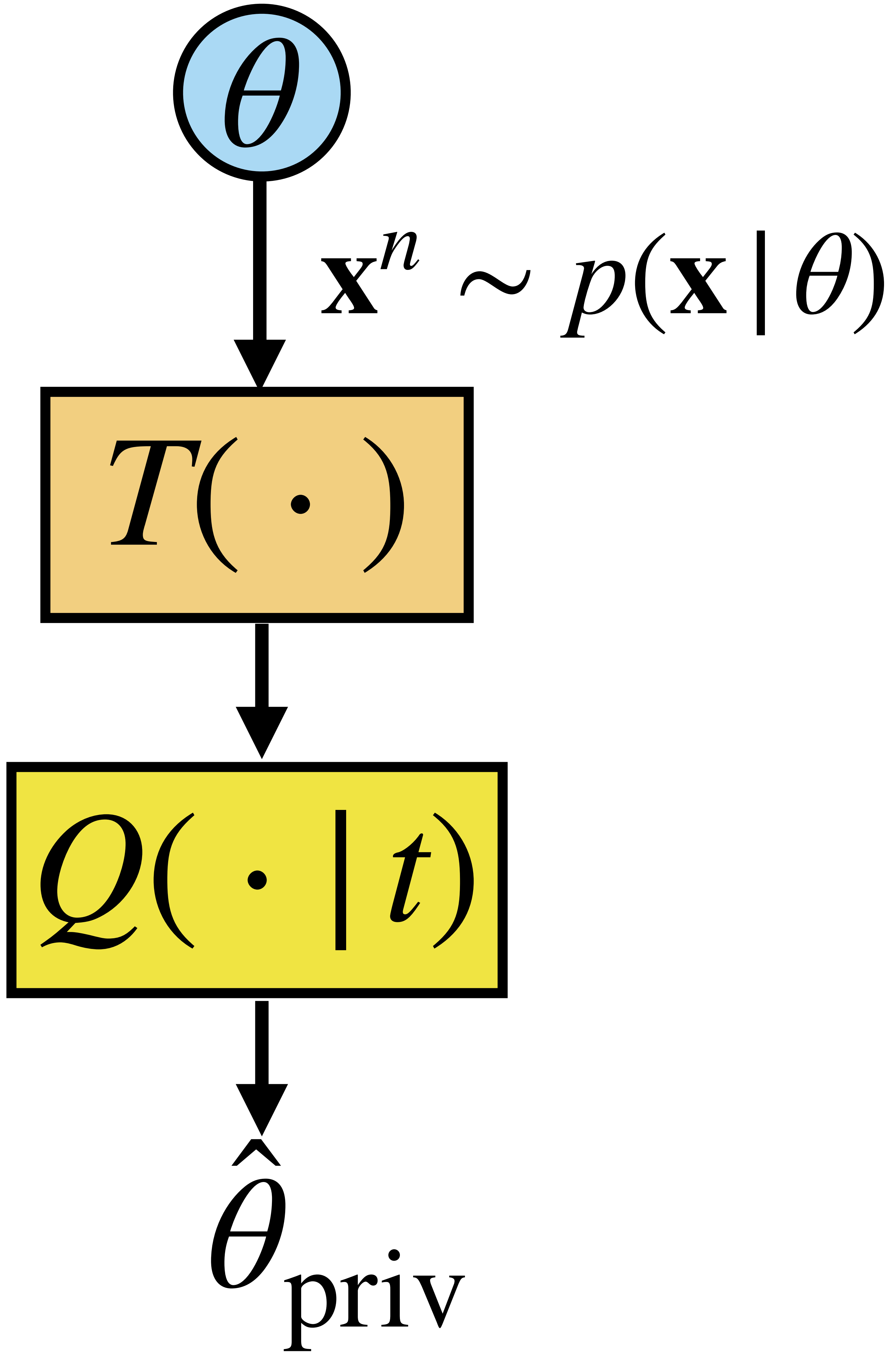
So we can see that less privacy risk (smaller ϵ) induces more MSE.

We can try to optimize the privacy mechanism if we know the utility function (like squared error).

This is what people call the **privacy-utility tradeoff**.

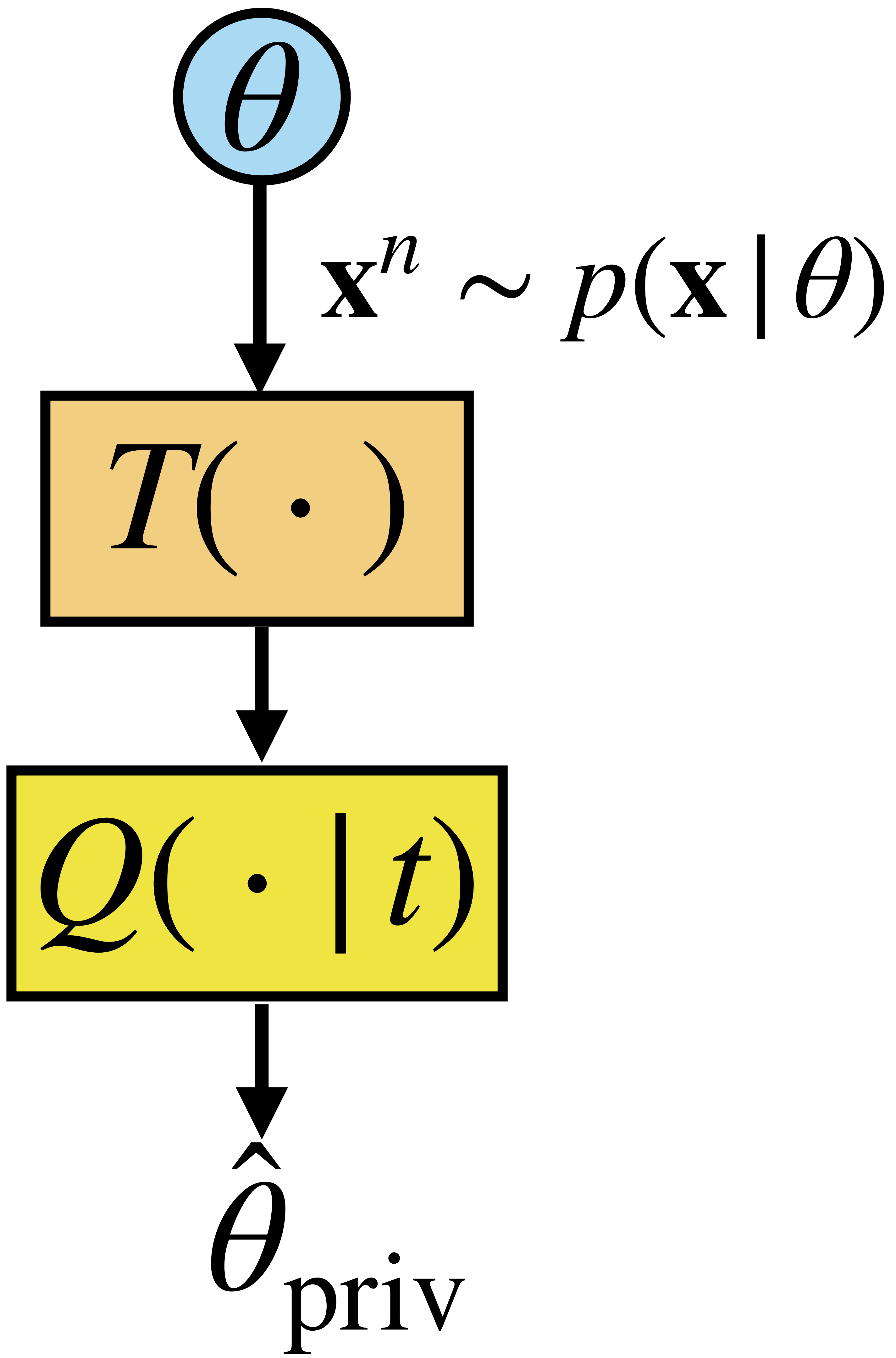
Point estimation with differential privacy

Adding noise to sufficient statistics



Point estimation with differential privacy

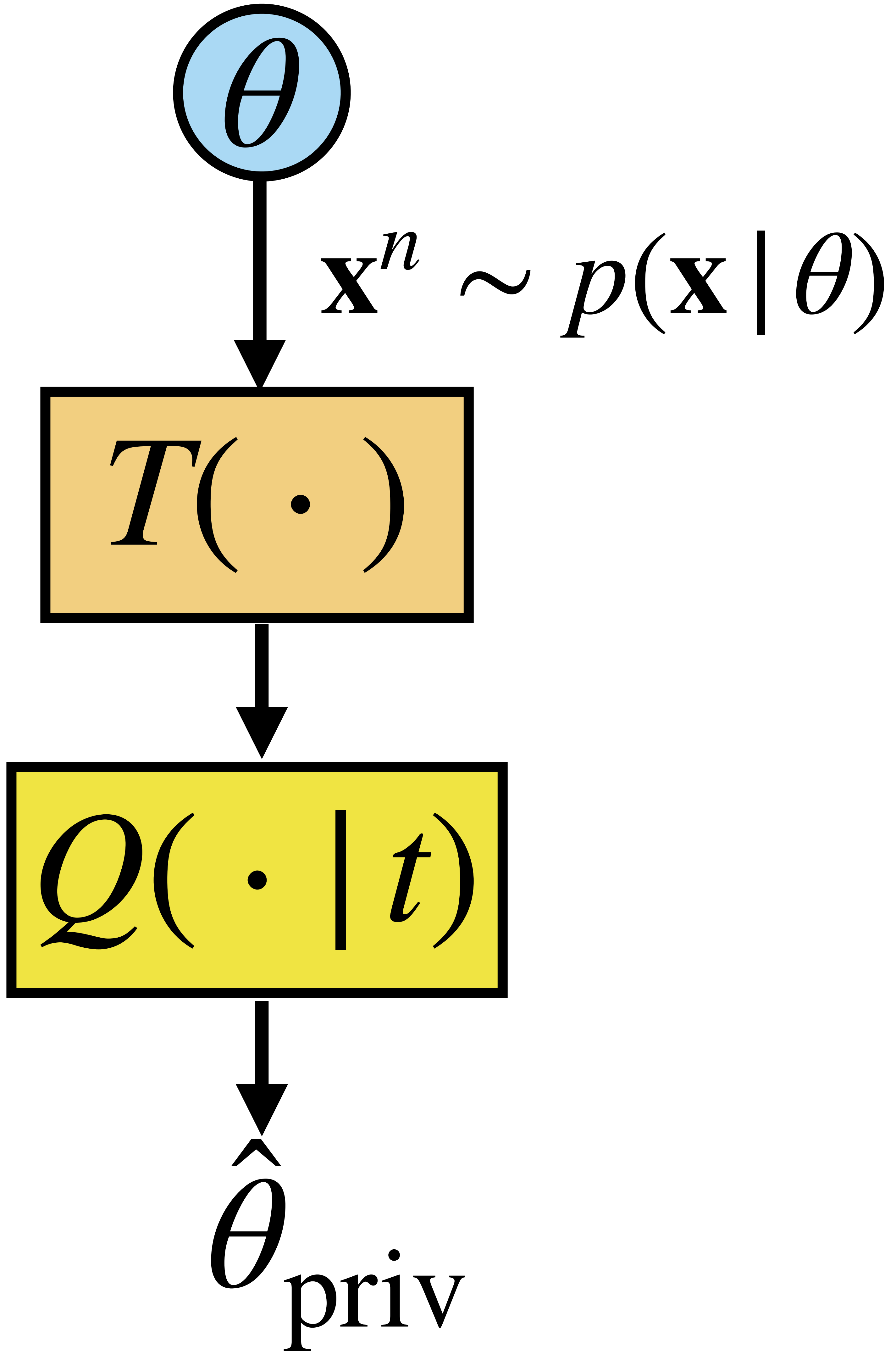
Adding noise to sufficient statistics



A typical DP approach to statistical estimation (Smith 2009):

Point estimation with differential privacy

Adding noise to sufficient statistics

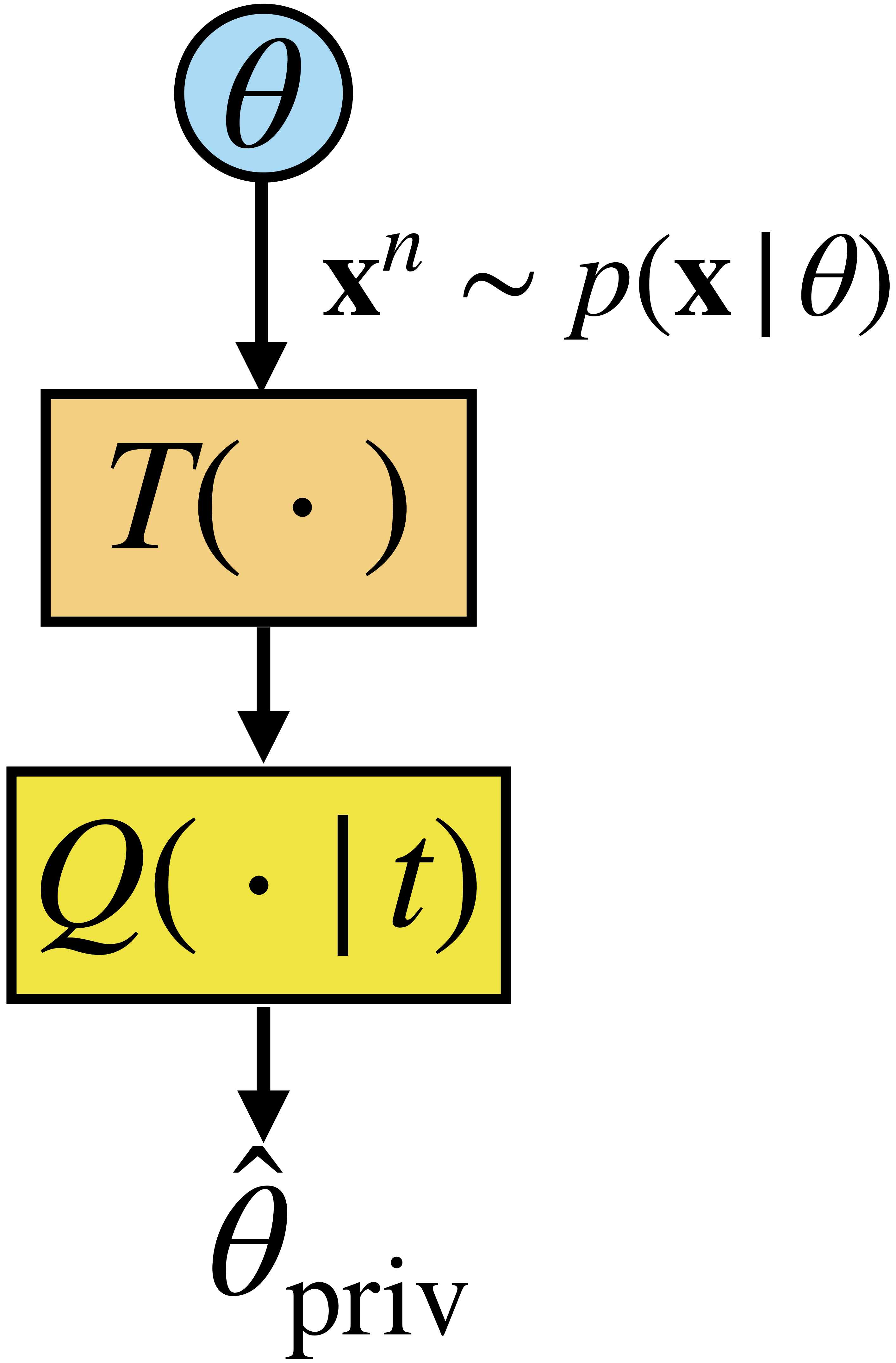


A typical DP approach to statistical estimation (Smith 2009):

- Model data as drawn i.i.d. $\sim p(\mathbf{x} \,|\, \theta)$.

Point estimation with differential privacy

Adding noise to sufficient statistics

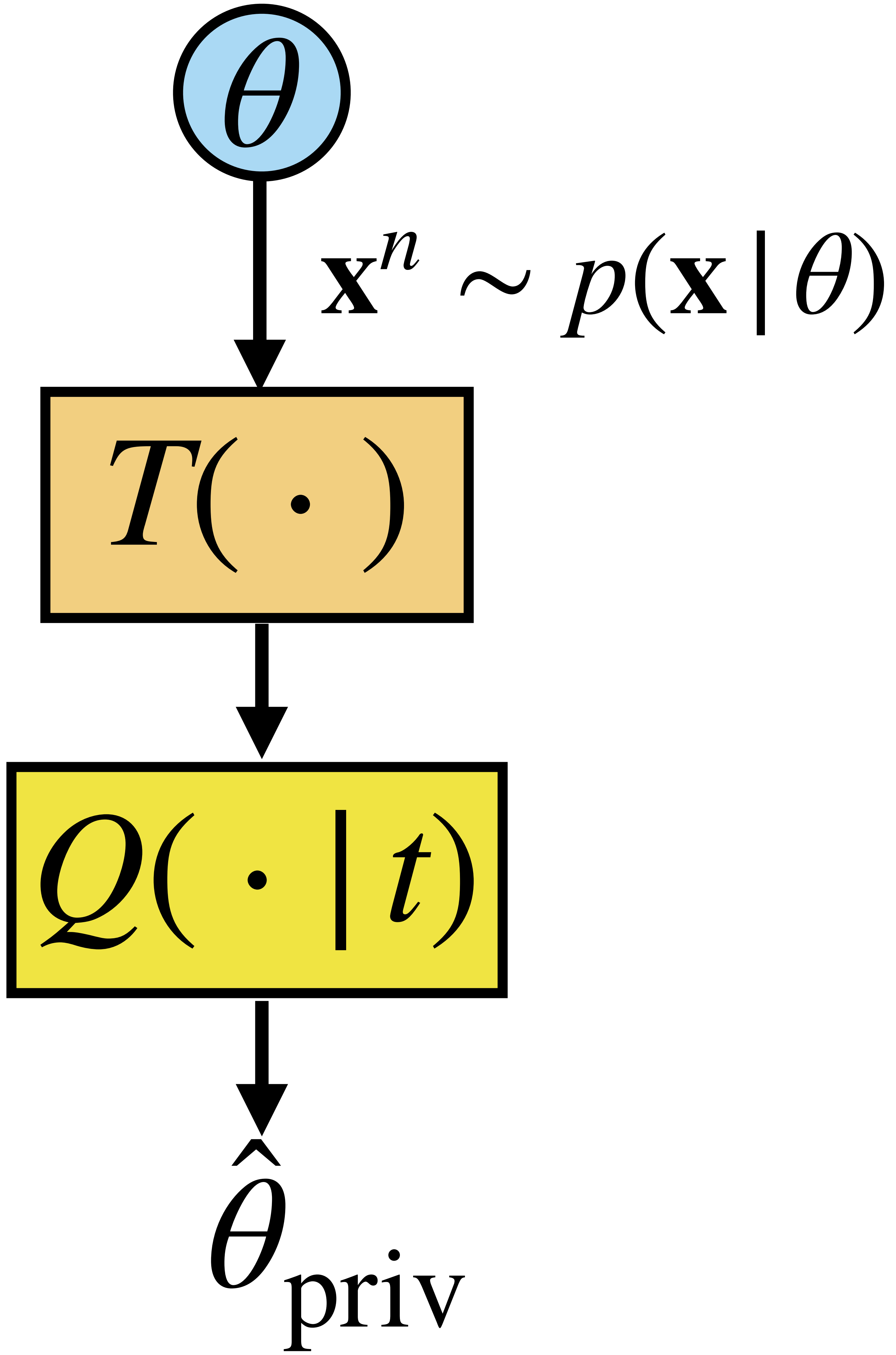


A typical DP approach to statistical estimation (Smith 2009):

- Model data as drawn i.i.d. $\sim p(\mathbf{x} \mid \theta)$.
- Compute a sufficient statistic $T(\mathbf{x}^n)$ for θ.

Point estimation with differential privacy

Adding noise to sufficient statistics

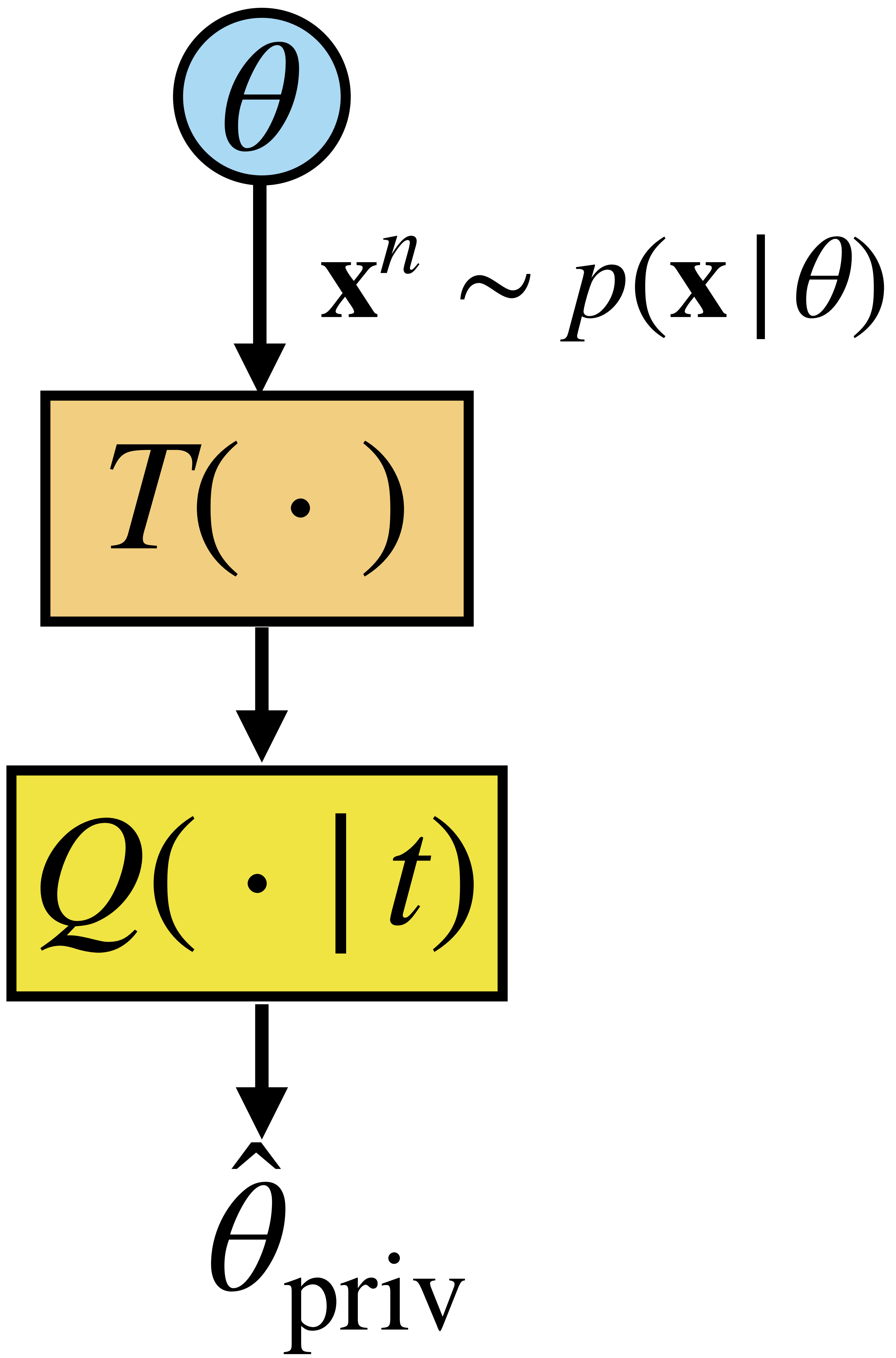


A typical DP approach to statistical estimation (Smith 2009):

- Model data as drawn i.i.d. $\sim p(\mathbf{x} \,|\, \theta)$.
- Compute a sufficient statistic $T(\mathbf{x}^n)$ for θ.
- Add noise to $T(\mathbf{x}^n)$ to guarantee DP.

Point estimation with differential privacy

Adding noise to sufficient statistics

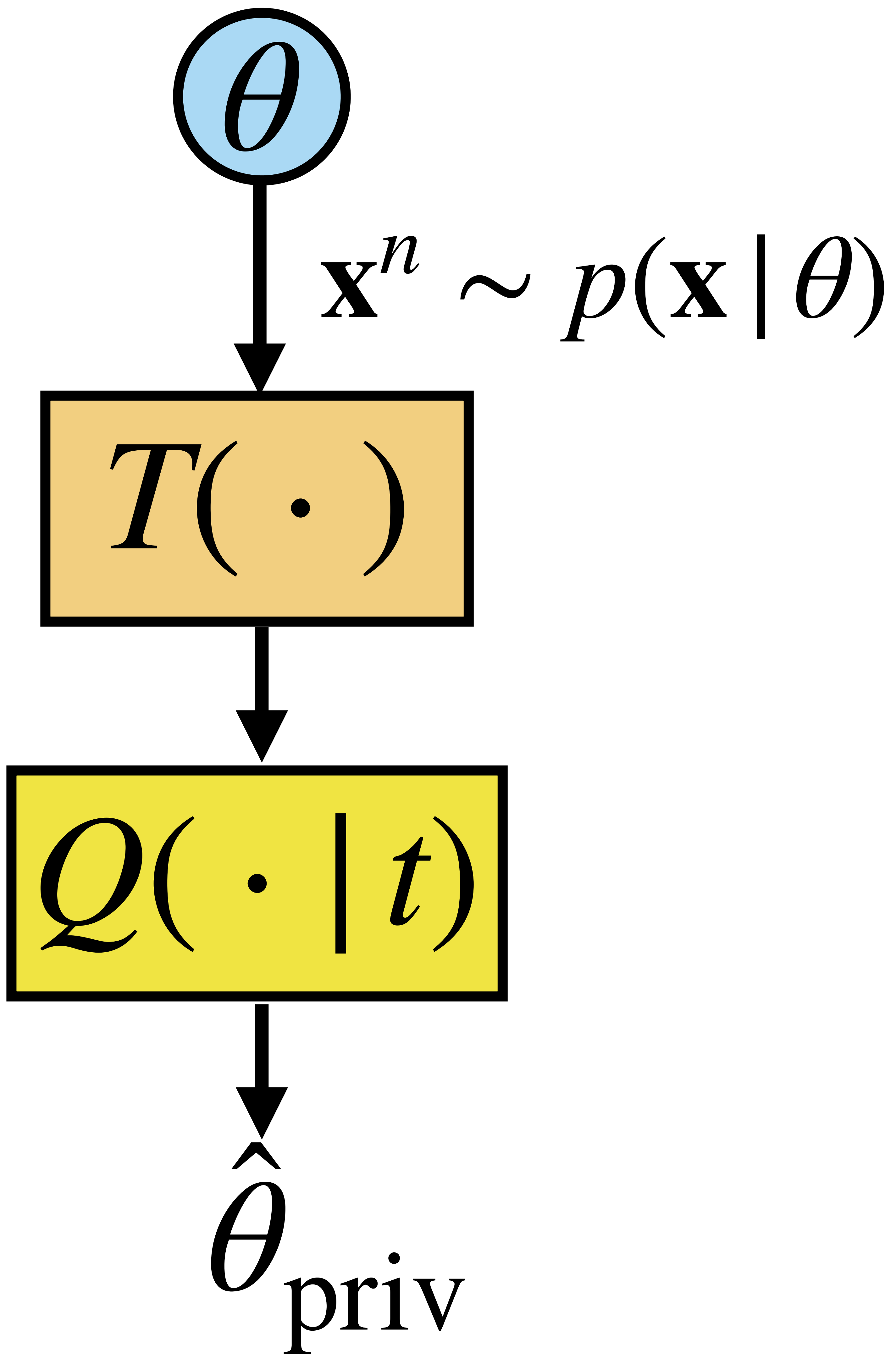


A typical DP approach to statistical estimation (Smith 2009):

- Model data as drawn i.i.d. $\sim p(\mathbf{x} \,|\, \theta)$.
- Compute a sufficient statistic $T(\mathbf{x}^n)$ for θ.
- Add noise to $T(\mathbf{x}^n)$ to guarantee DP.
- Compute a “plug-in” estimate from noisy $T(\mathbf{x}^n)$.

Point estimation with differential privacy

Adding noise to sufficient statistics



A typical DP approach to statistical estimation (Smith 2009):

- Model data as drawn i.i.d. $\sim p(\mathbf{x} \,|\, \theta)$.
- Compute a sufficient statistic $T(\mathbf{x}^n)$ for θ.
- Add noise to $T(\mathbf{x}^n)$ to guarantee DP.
- Compute a “plug-in” estimate from noisy $T(\mathbf{x}^n)$.

We just need the sensitivity of $T(\,\cdot\,)$.

What kind of noise should we use?

So many different choices: a non-comprehensive list

What kind of noise should we use?

So many different choices: a non-comprehensive list

<u>Variations on geometric noise</u>

Ghosh, Roughgarden,
Sundarajan (2009/2012)

Gupte, Sundararajan (2010)

Balcer, Vadhan (2017)

What kind of noise should we use?

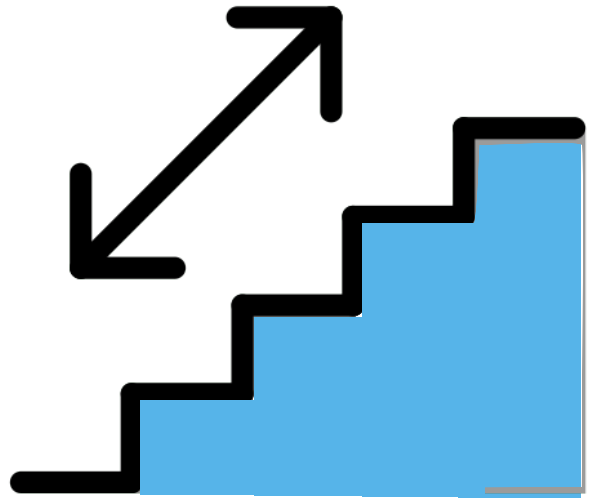
So many different choices: a non-comprehensive list

<u>Variations on geometric noise</u>

Ghosh, Roughgarden, Sundarajan (2009/2012)

Gupte, Sundararajan (2010)

Balcer, Vadhan (2017)



<u>Variations on staircases</u>

Geng, Viswanath (2015, 2016

Kairouz, Oh, Viswanath (2014)

Geng, Kairouz, Oh, Viswanath (2015)

Kalantari, Sankar, Sarwate (2018)

What kind of noise should we use?

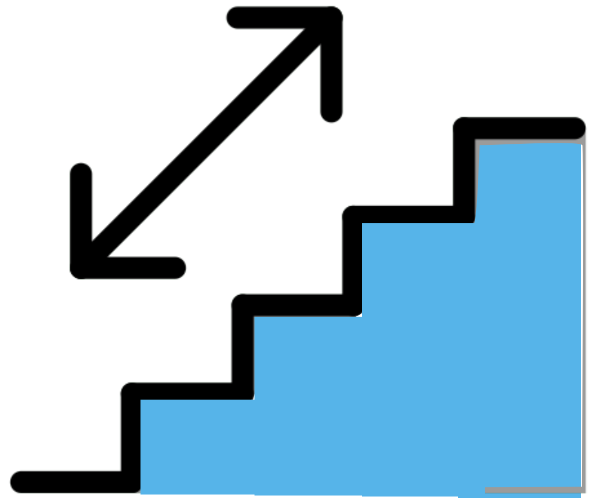
So many different choices: a non-comprehensive list

<u>Variations on geometric noise</u>

Ghosh, Roughgarden, Sundarajan (2009/2012)

Gupte, Sundararajan (2010)

Balcer, Vadhan (2017)



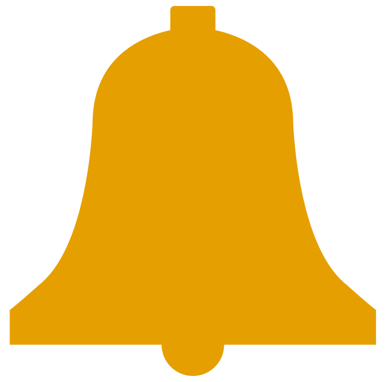
<u>Variations on staircases</u>

Geng, Viswanath (2015, 2016

Kairouz, Oh, Viswanath (2014)

Geng, Kairouz, Oh, Viswanath (2015)

Kalantari, Sankar, Sarwate (2018)



<u>Variations on Gaussians</u>

Cannone, Kamath, Steinke (2020)

Sadeghi, Korki (2022)

Muthukrishnan, Kalyani (2023)

Rinberg, Shumailov, Cummings, Papernot (2024)

What kind of noise should we use?

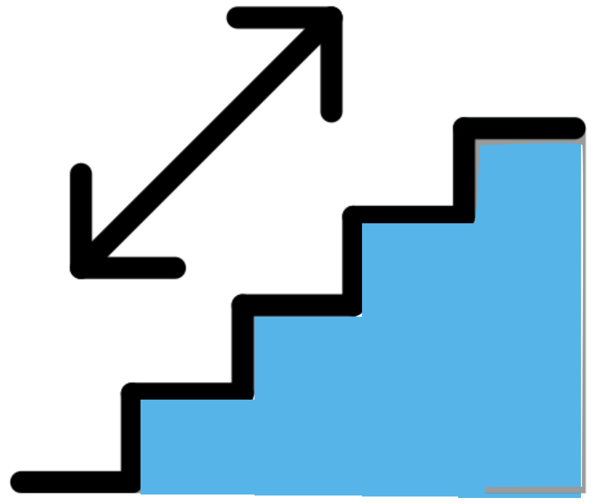
So many different choices: a non-comprehensive list

Variations on geometric noise

Ghosh, Roughgarden, Sundarajan (2009/2012)

Gupte, Sundararajan (2010)

Balcer, Vadhan (2017)



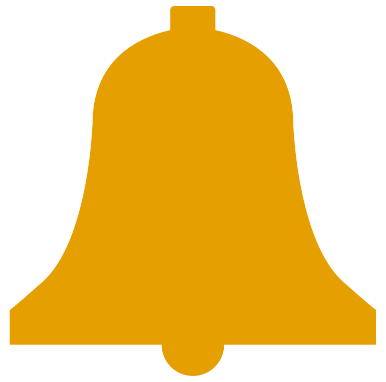
Variations on staircases

Geng, Viswanath (2015, 2016

Kairouz, Oh, Viswanath (2014)

Geng, Kairouz, Oh, Viswanath (2015)

Kalantari, Sankar, Sarwate (2018)



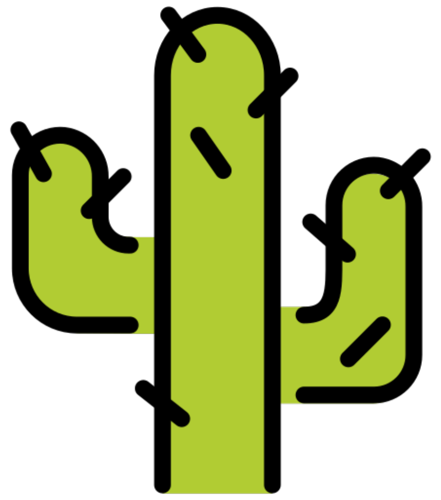
Variations on Gaussians

Cannone, Kamath, Steinke (2020)

Sadeghi, Korki (2022)

Muthukrishnan, Kalyani (2023)

Rinberg, Shumailov, Cummings, Papernot (2024)



“Other"

Geng, Ding, Guo, Kumar (2019/2020)

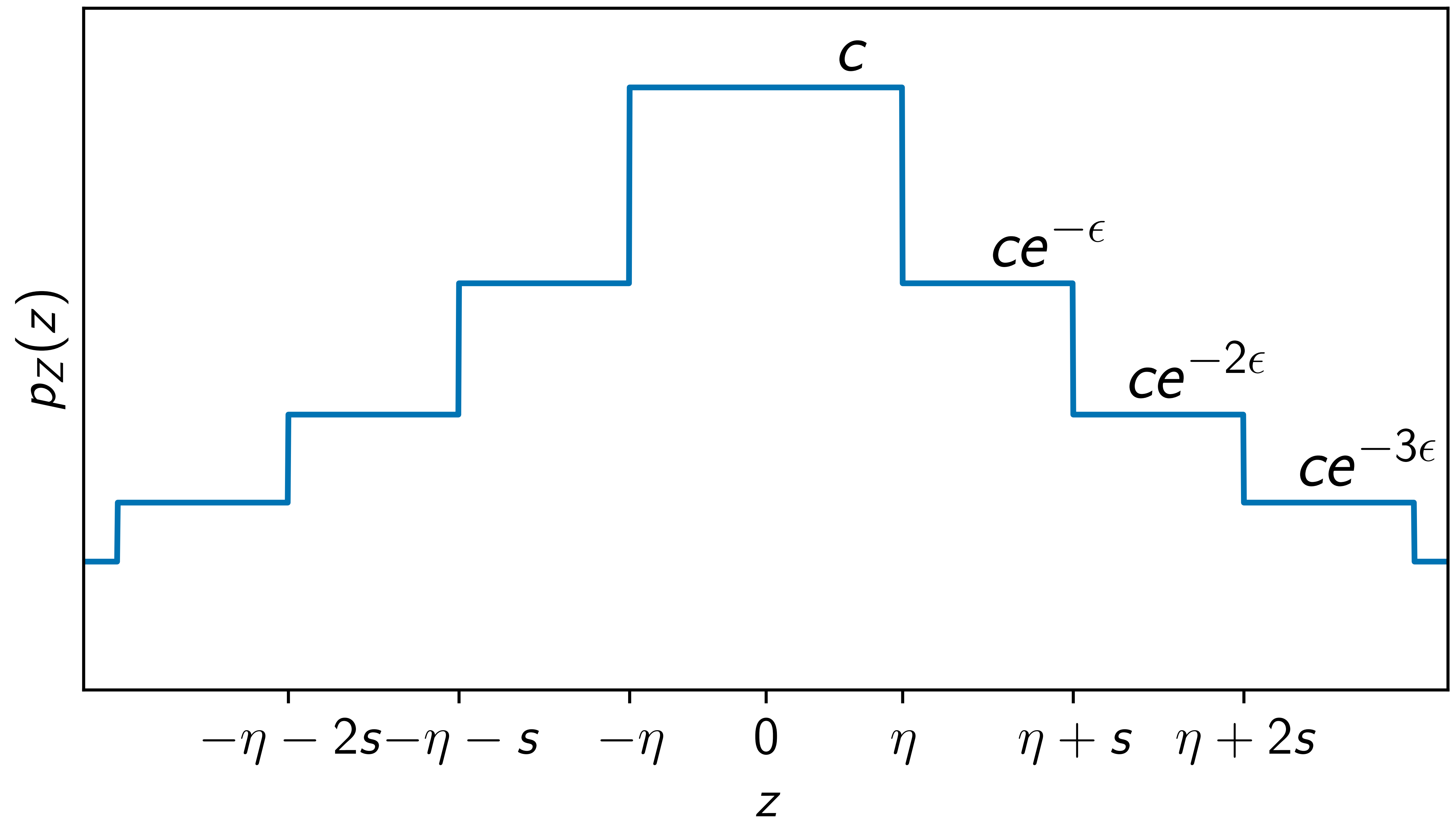
Dong, Su, Zhang (2021)

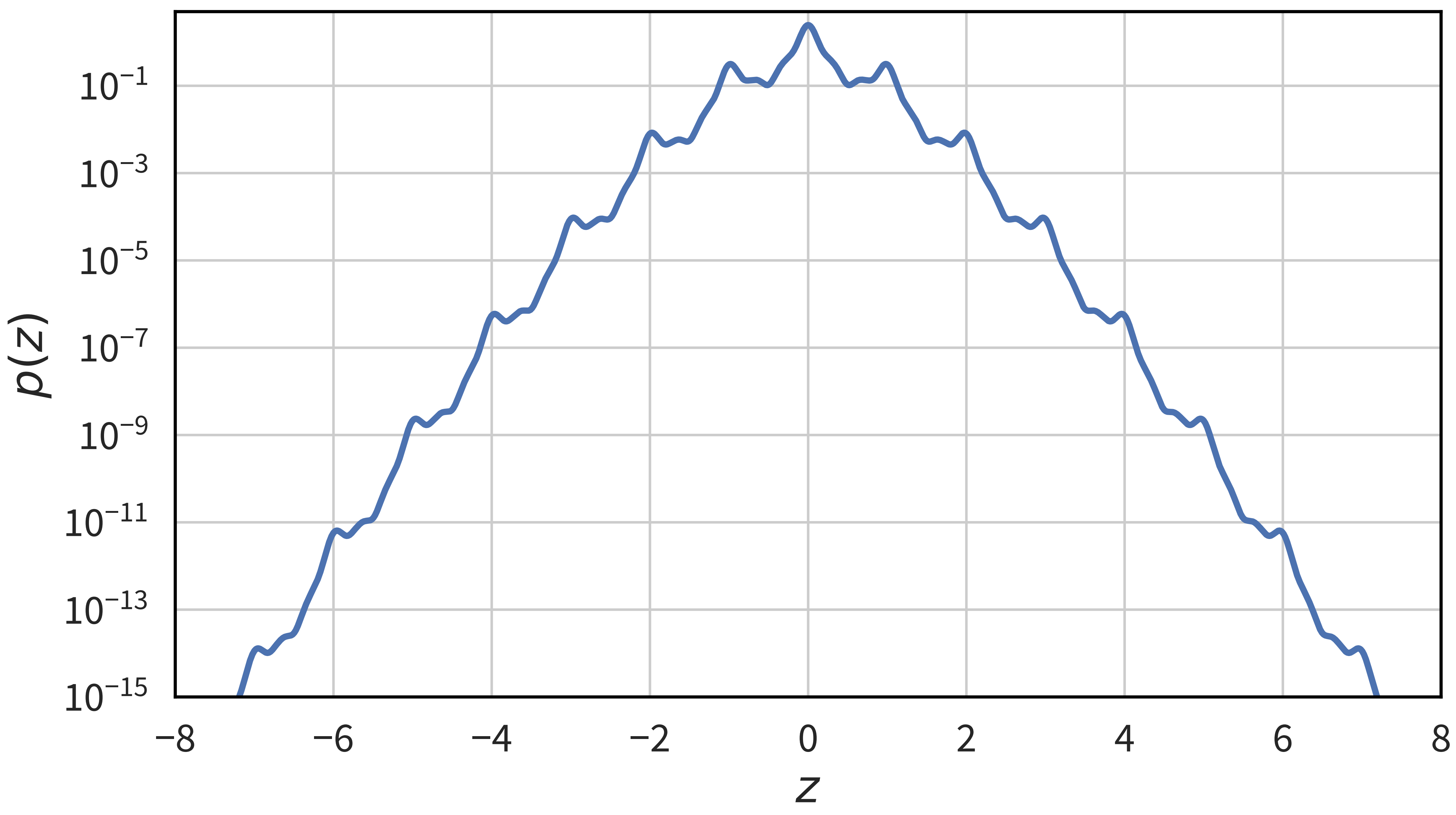
Alghamdi, Asoodeh, Calmon, Kosut, Sankar, Wei (2022)

Alghamdi, Asoodeh, Calmon, Gomez, Kosut, Sankar (2023)

“Optimal” noise distributions

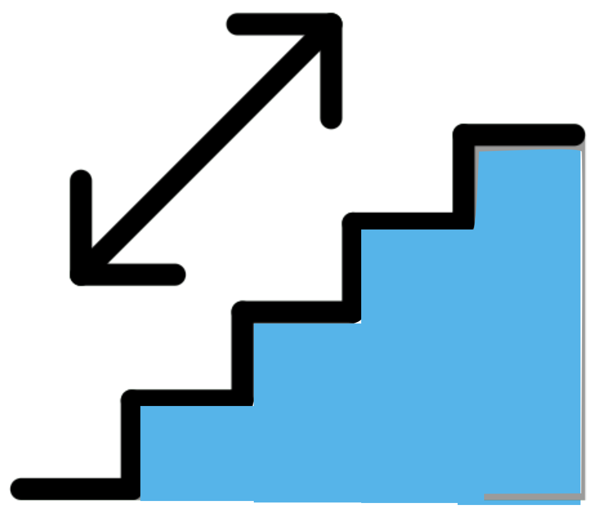
Beyond Gaussian and Laplace

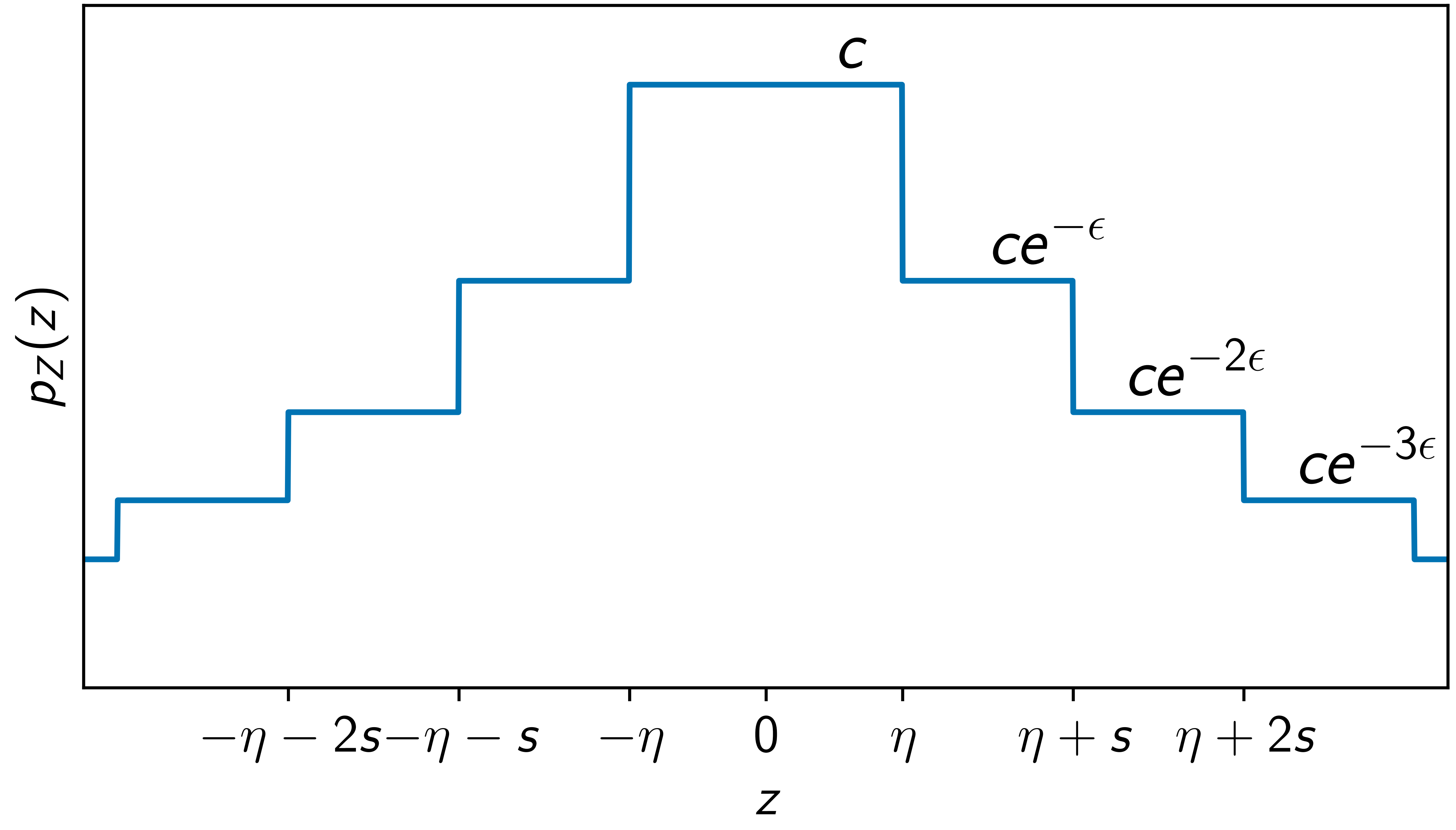


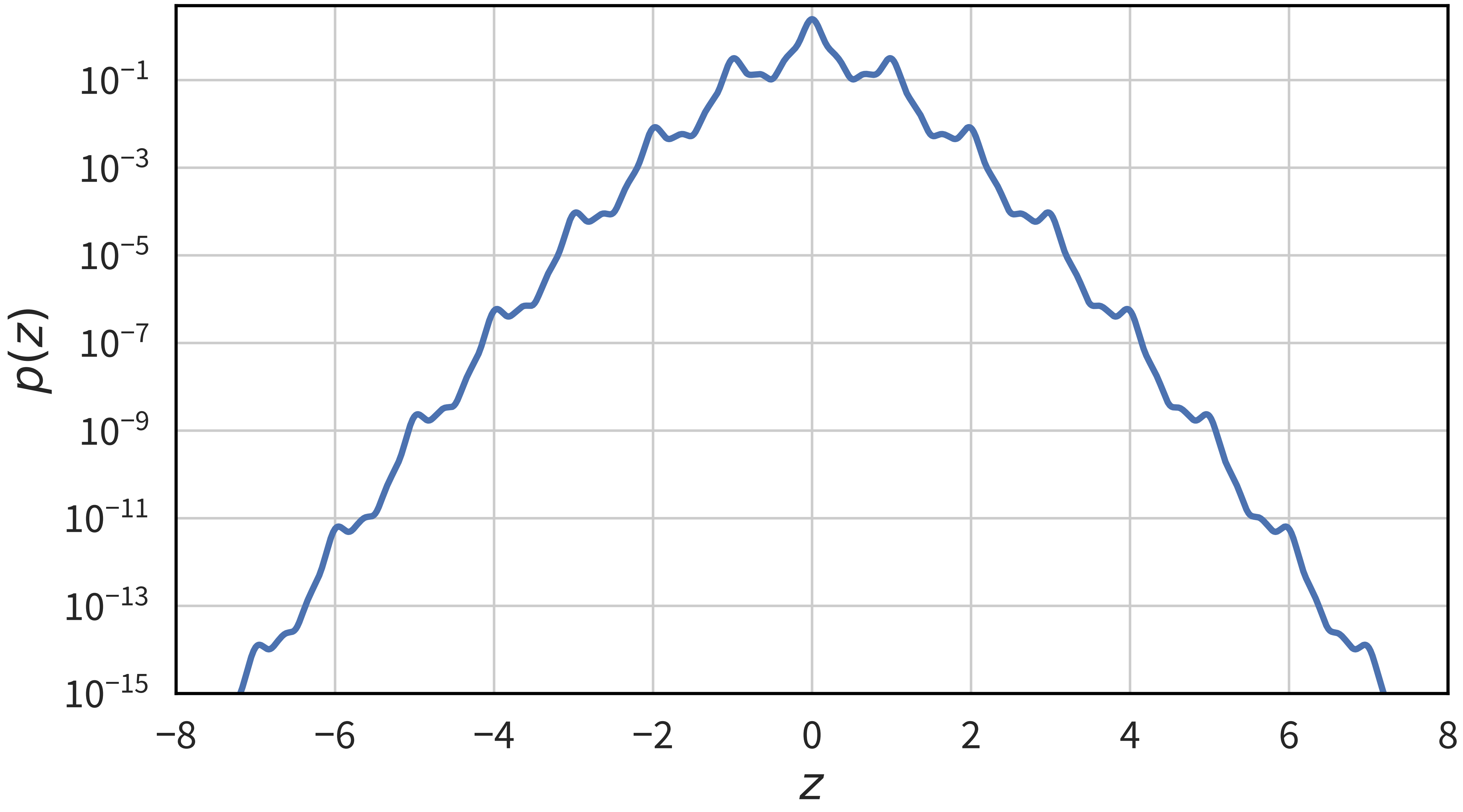


“Optimal” noise distributions

Beyond Gaussian and Laplace

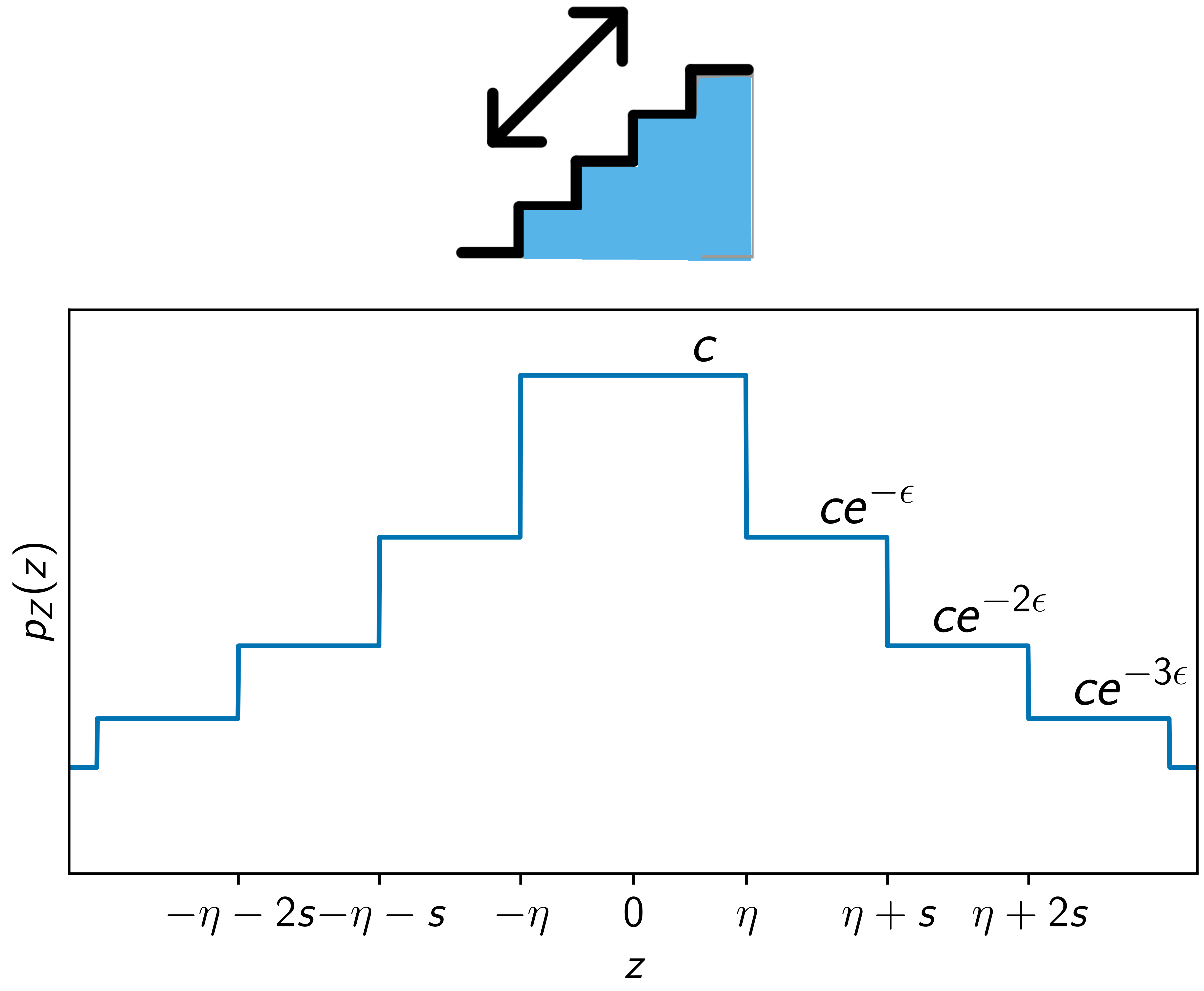


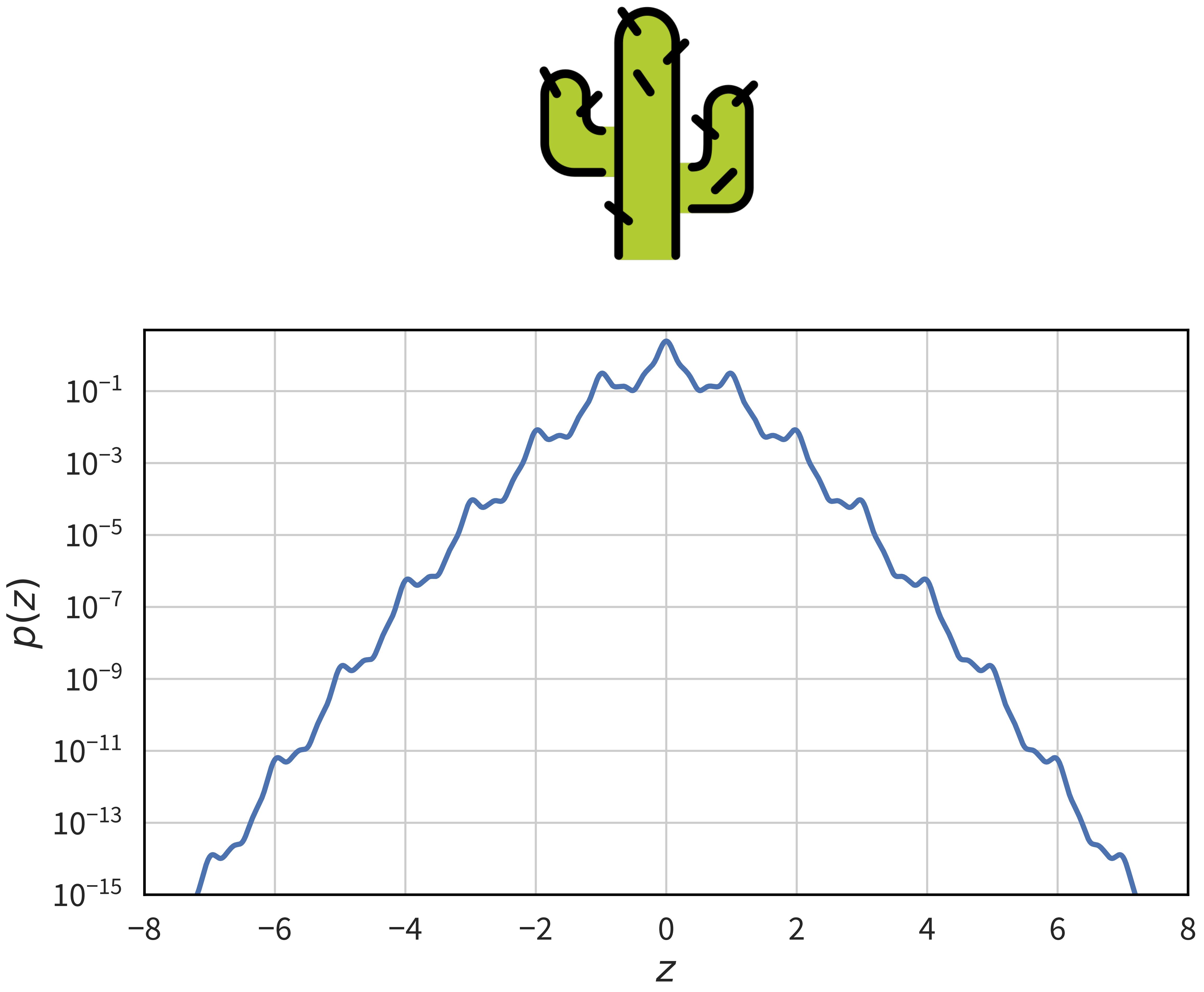




“Optimal” noise distributions

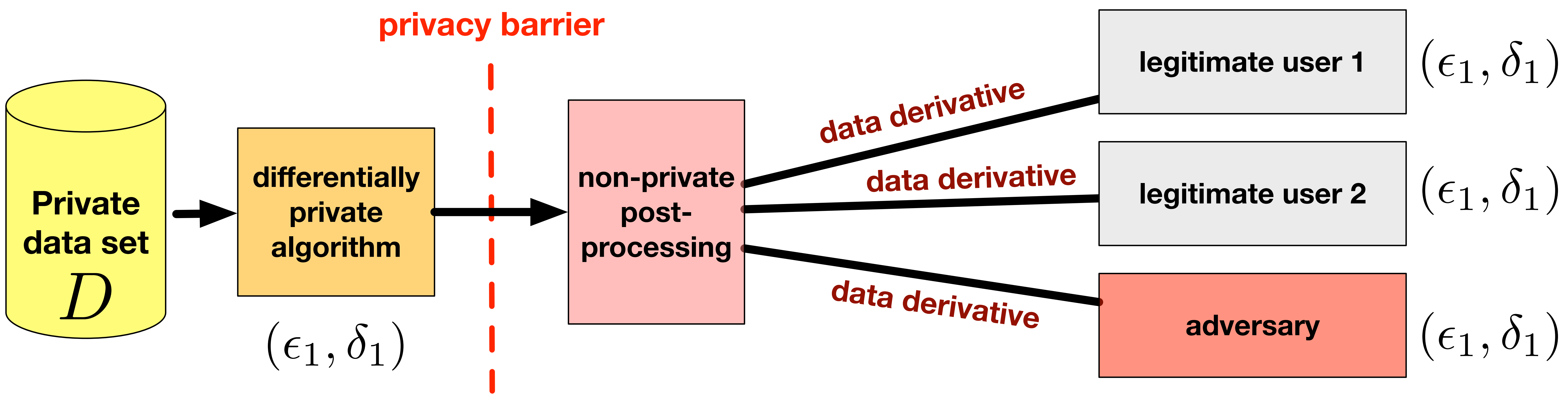
Beyond Gaussian and Laplace





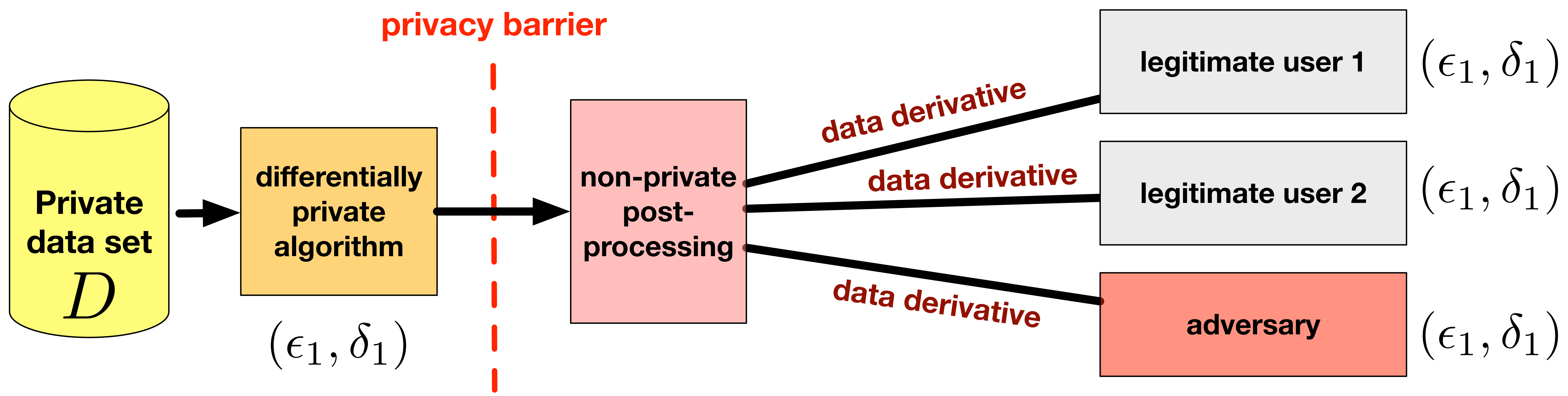
Post-processing invariance and composition

Nice properties of differential privacy



Post-processing invariance and composition

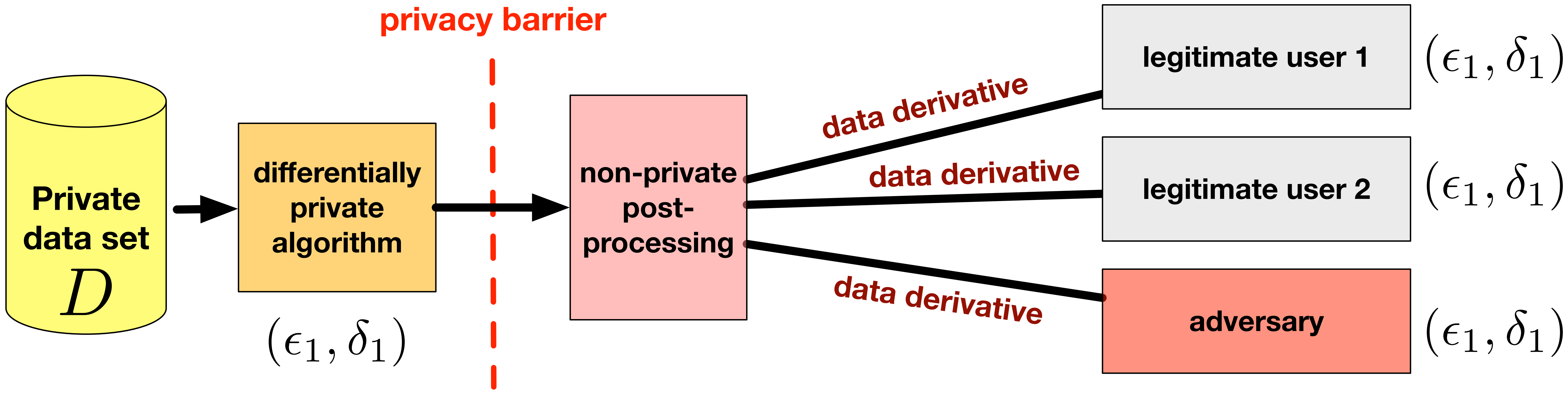
Nice properties of differential privacy



- **Side-information resilience:** measures the additional risk regardless of what is known already.

Post-processing invariance and composition

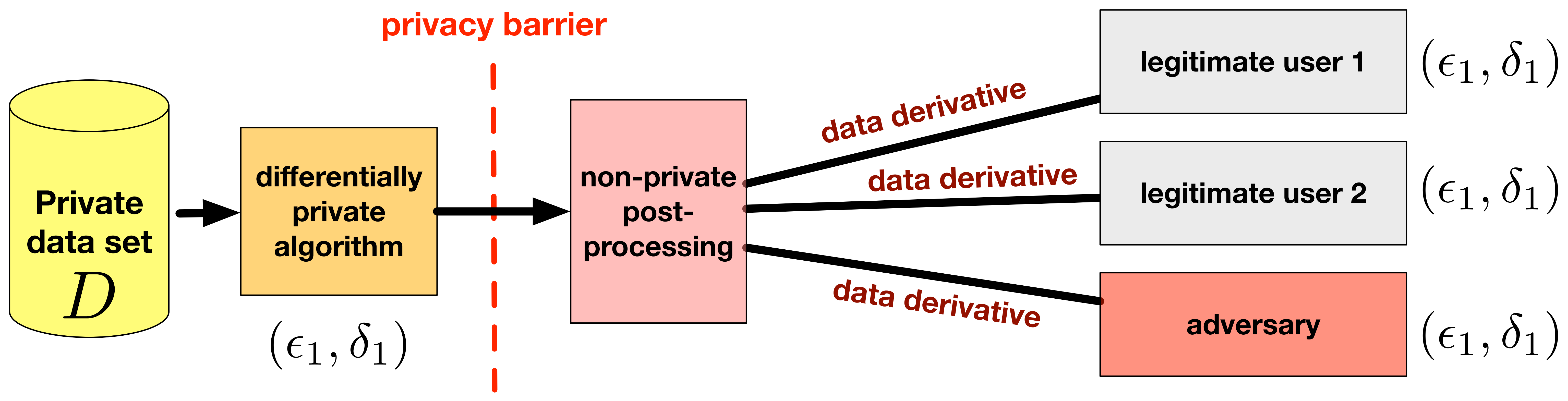
Nice properties of differential privacy



- **Side-information resilience:** measures the additional risk regardless of what is known already.
- **Post-processing invariance:** once we publish something the risk cannot increase from additional computations.

Post-processing invariance and composition

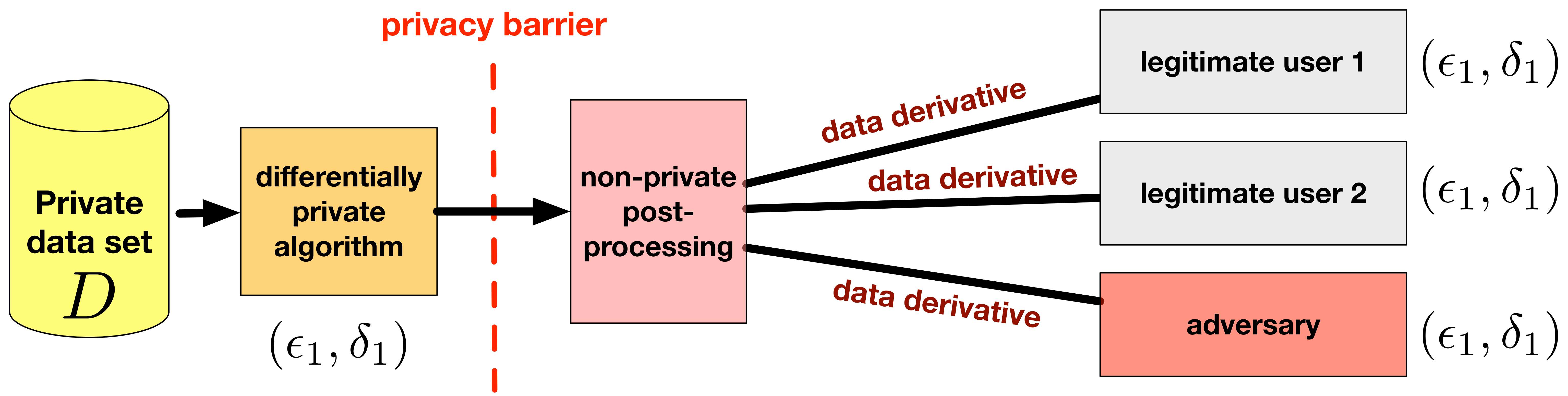
Nice properties of differential privacy



- **Side-information resilience:** measures the additional risk regardless of what is known already.
- **Post-processing invariance:** once we publish something the risk cannot increase from additional computations.
- **Composition:** quantifies how privacy loss “adds up” over multiple releases.

Post-processing invariance and composition

Nice properties of differential privacy



- **Side-information resilience:** measures the additional risk regardless of what is known already.
- **Post-processing invariance:** once we publish something the risk cannot increase from additional computations.
- **Composition:** quantifies how privacy loss “adds up” over multiple releases.

Umezawa in Sagami Province

相州梅沢庄

Soshū Umezawanoshō

Vista 3

f-divergences/composition

Privacy loss random variables

Characterizing the distribution of the LLR

Sommer, Meisner, Mohammadi (2020), Zhu, Dong, Wang (2022)

Privacy loss random variables

Characterizing the distribution of the LLR

The fundamental quantity of interest is called the **privacy loss random variable** (PLRV). Switching notation a little bit, we can define the variable

Sommer, Meisner, Mohammadi (2020), Zhu, Dong, Wang (2022)

Privacy loss random variables

Characterizing the distribution of the LLR

The fundamental quantity of interest is called the **privacy loss random variable** (PLRV). Switching notation a little bit, we can define the variable

$$L_{x,x'} = \log \frac{dP_{Y|X=x}}{dP_{Y|X=x'}}(Y)$$

Sommer, Meisner, Mohammadi (2020), Zhu, Dong, Wang (2022)

Privacy loss random variables

Characterizing the distribution of the LLR

The fundamental quantity of interest is called the **privacy loss random variable** (PLRV). Switching notation a little bit, we can define the variable

$$L_{x,x'} = \log \frac{dP_{Y|X=x}}{dP_{Y|X=x'}}(Y)$$

$$\mathbb{E}[L] = D_{\mathrm{KL}}(P_{Y|x} \| P_{Y|x'})$$

Sommer, Meisner, Mohammadi (2020), Zhu, Dong, Wang (2022)

Privacy loss random variables

Characterizing the distribution of the LLR

The fundamental quantity of interest is called the **privacy loss random variable** (PLRV). Switching notation a little bit, we can define the variable

$$L_{x,x'} = \log \frac{dP_{Y|X=x}}{dP_{Y|X=x'}}(Y)$$

$$\mathbb{E}[L] = D_{\mathrm{KL}}(P_{Y|x} \| P_{Y|x'})$$

The distribution of the PLRV is sometimes called the **privacy loss distribution** (PLD).

Sommer, Meisner, Mohammadi (2020), Zhu, Dong, Wang (2022)

Privacy loss random variables

Characterizing the distribution of the LLR

The fundamental quantity of interest is called the **privacy loss random variable** (PLRV). Switching notation a little bit, we can define the variable

$$L_{x,x'} = \log \frac{dP_{Y|X=x}}{dP_{Y|X=x'}}(Y)$$

$$\mathbb{E}[L] = D_{\mathrm{KL}}(P_{Y|x} \| P_{Y|x'})$$

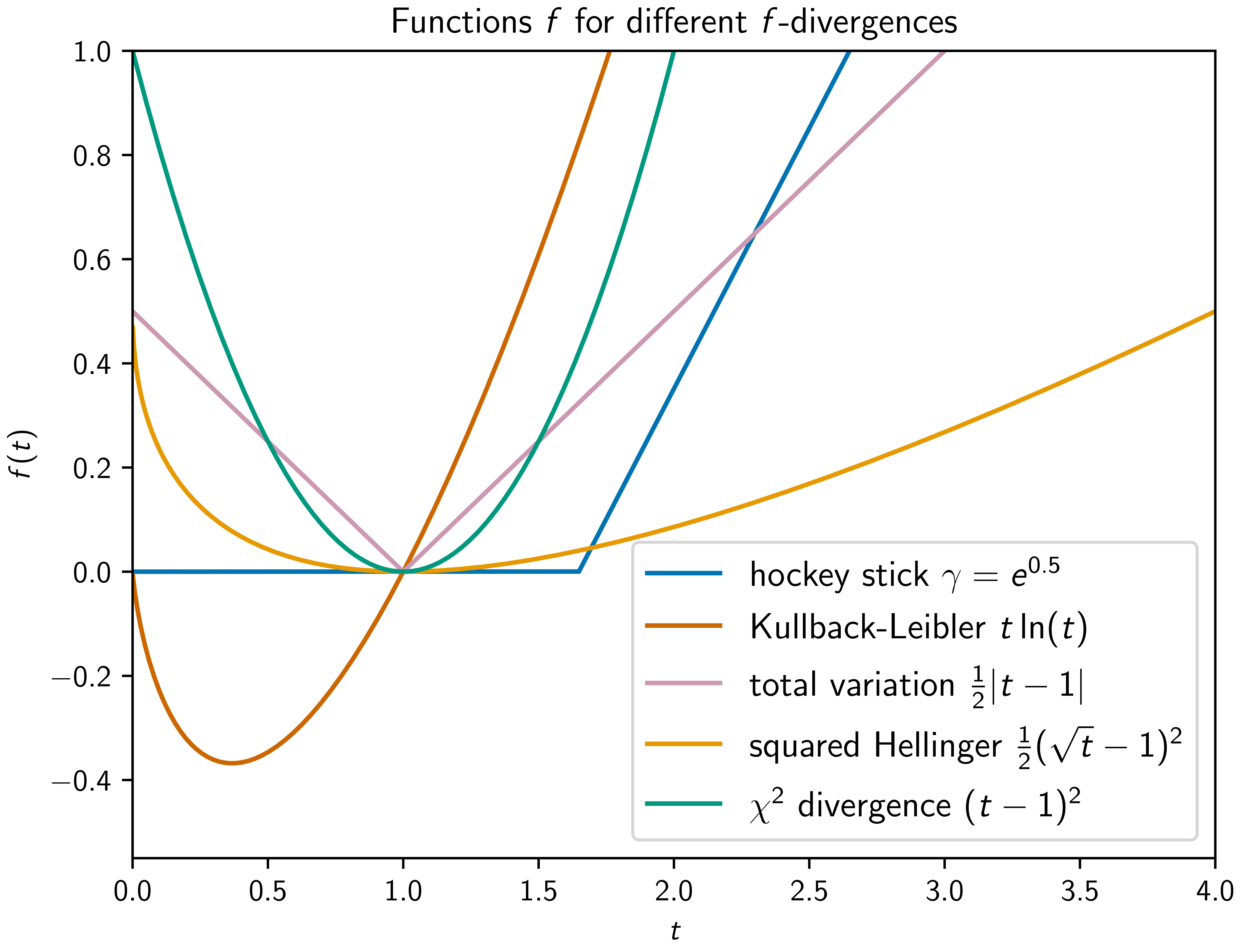
The distribution of the PLRV is sometimes called the **privacy loss distribution** (PLD).

A challenge: this is defined for a single pair of inputs (x, x'). We would like to only deal with the "worst case" pair of inputs.

Sommer, Meisner, Mohammadi (2020), Zhu, Dong, Wang (2022)

Generalized divergences and the 🏒 divergence

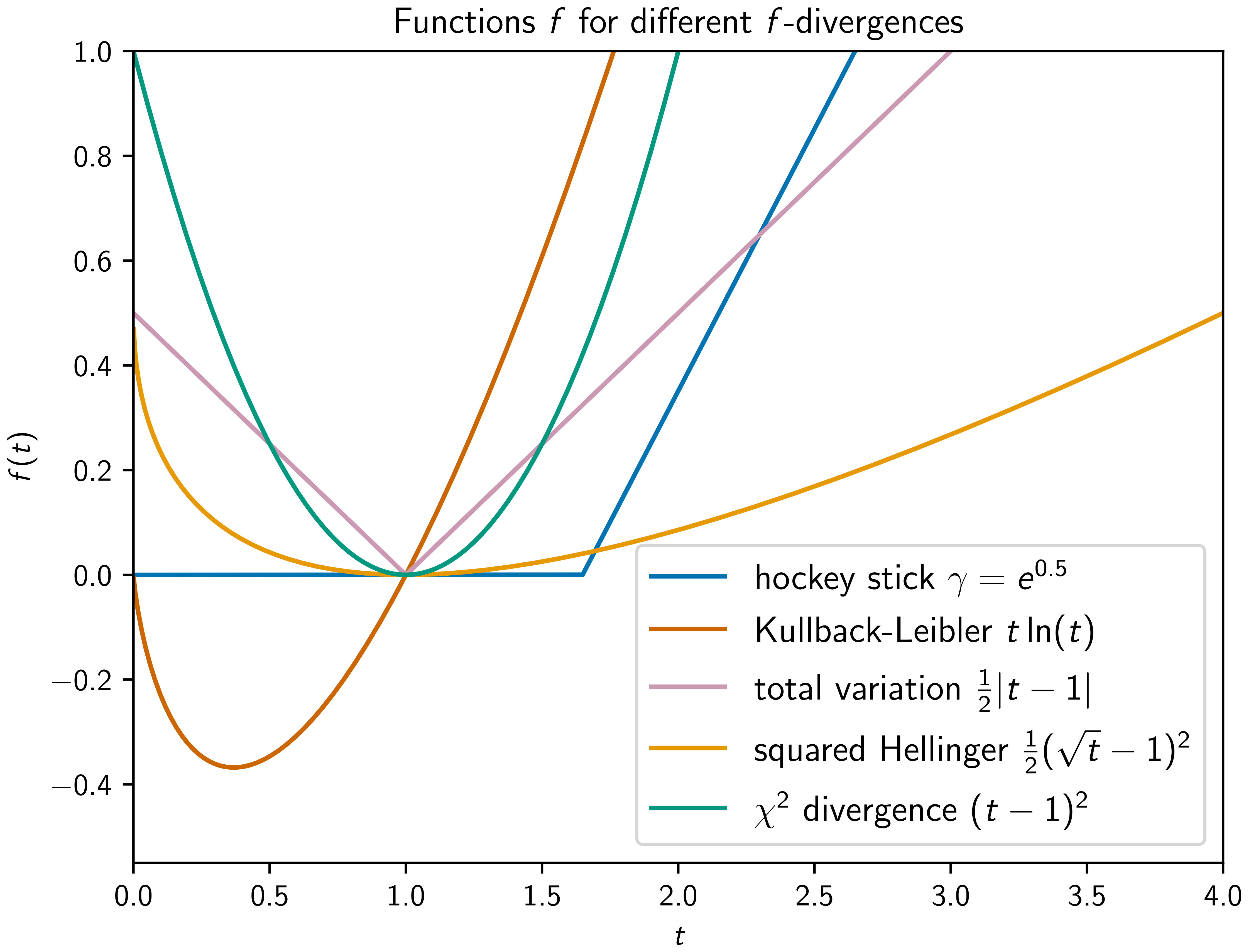
How different are these two distributions?



Rényi (1961), Cziszár (1963), Morimoto (1963), Ali, Silvey (1966), Csiszár (1967),
Polyanskiy, Poor, Verdu (2010), Balle, Barthe, Gaboardi, Geumlek (2019)

Generalized divergences and the 🏒 divergence

How different are these two distributions?

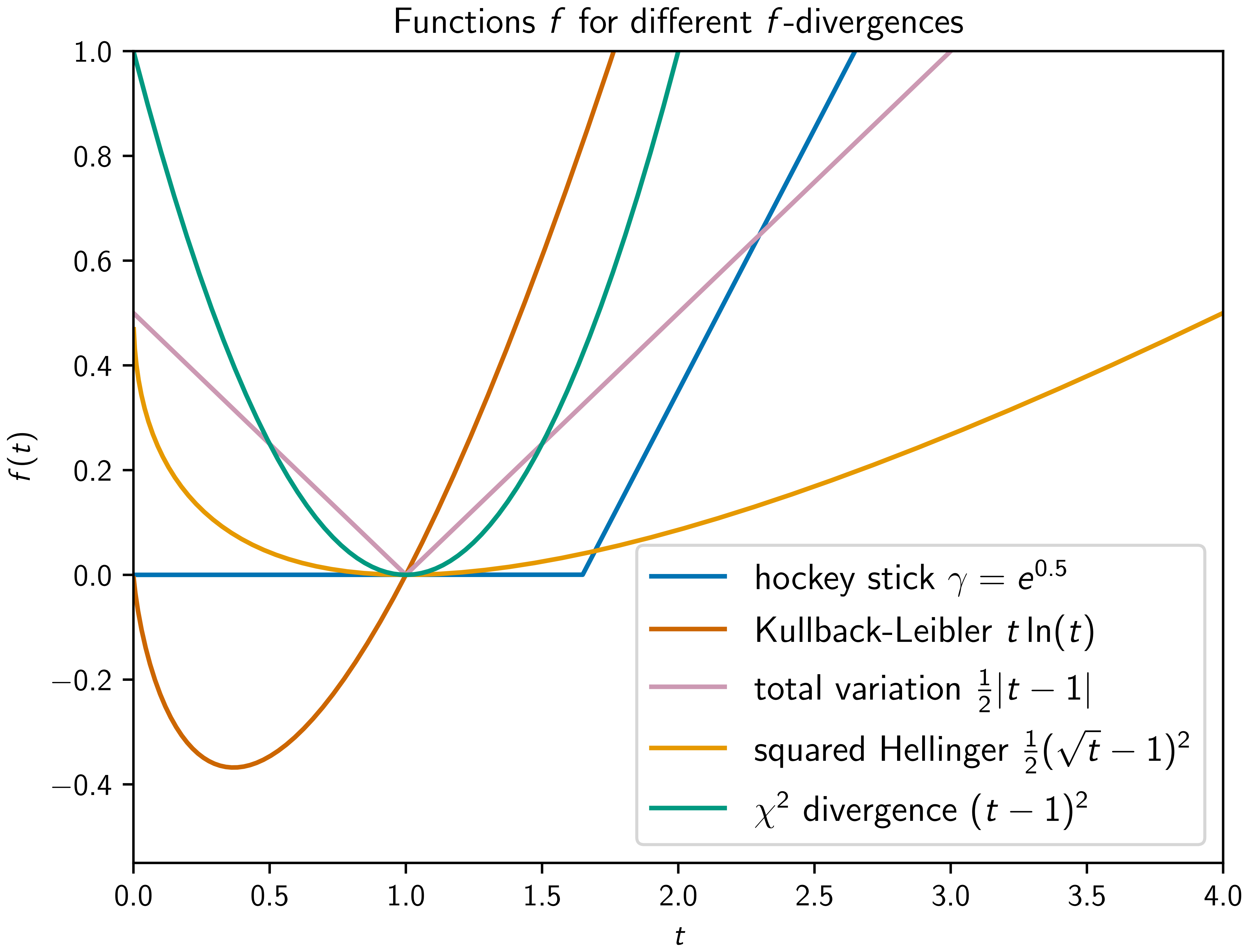


More generally, look at f-divergences: for any convex $f(\,\cdot\,)$,

Rényi (1961), Cziszár (1963), Morimoto (1963), Ali, Silvey (1966), Csiszár (1967), Polyanskiy, Poor, Verdu (2010), Balle, Barthe, Gaboardi, Geumlek (2019)

Generalized divergences and the 🏒 divergence

How different are these two distributions?



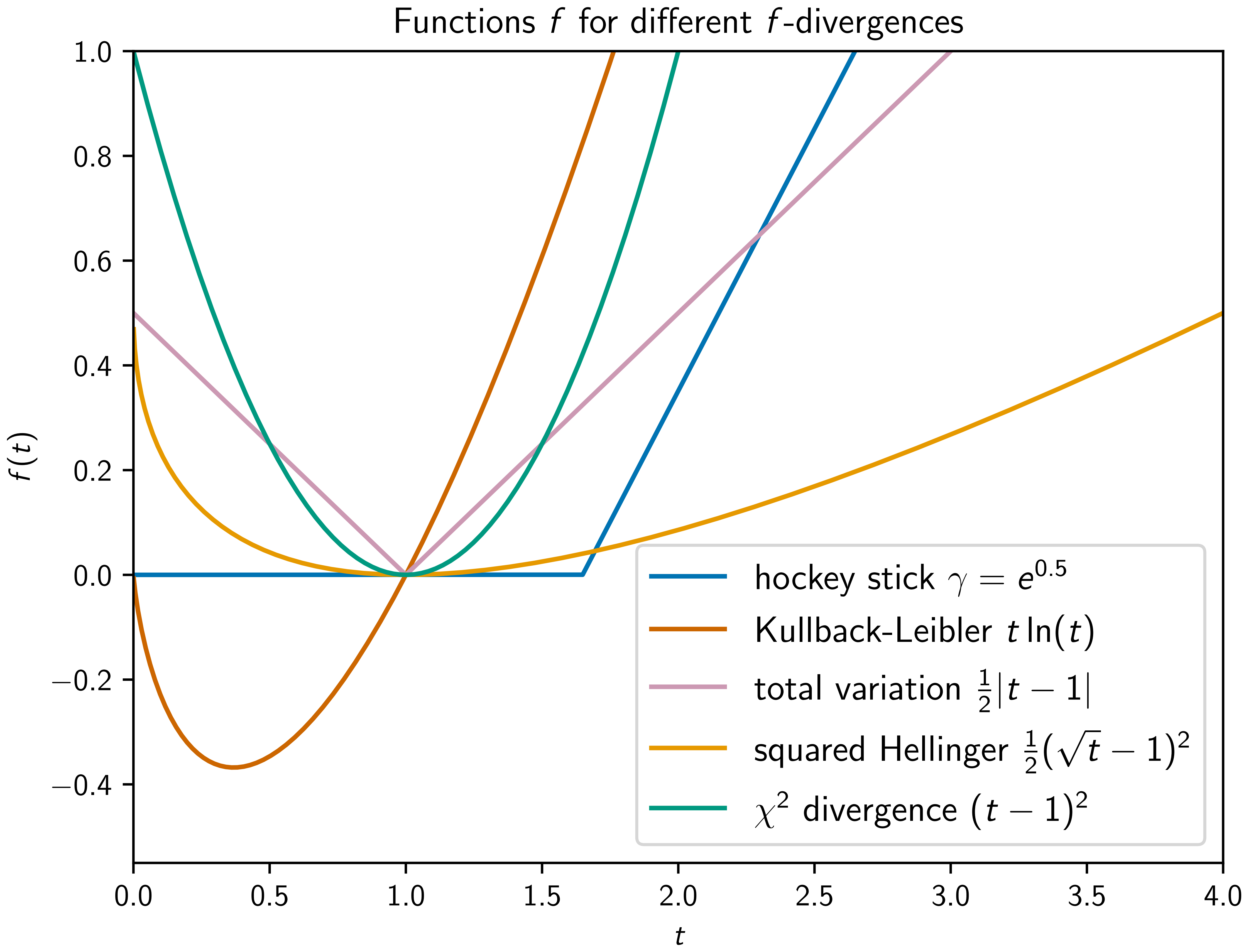
More generally, look at f-divergences: for any convex $f(\cdot)$,

$$D_f(\mu \| \nu) = \mathbb{E}_\nu \left[f\left(\frac{d\mu}{d\nu} \right) \right].$$

Rényi (1961), Cziszár (1963), Morimoto (1963), Ali, Silvey (1966), Csiszár (1967), Polyanskiy, Poor, Verdu (2010), Balle, Barthe, Gaboardi, Geumlek (2019)

Generalized divergences and the 🏒 divergence

How different are these two distributions?



More generally, look at f-divergences: for any convex $f(\,\cdot\,)$,

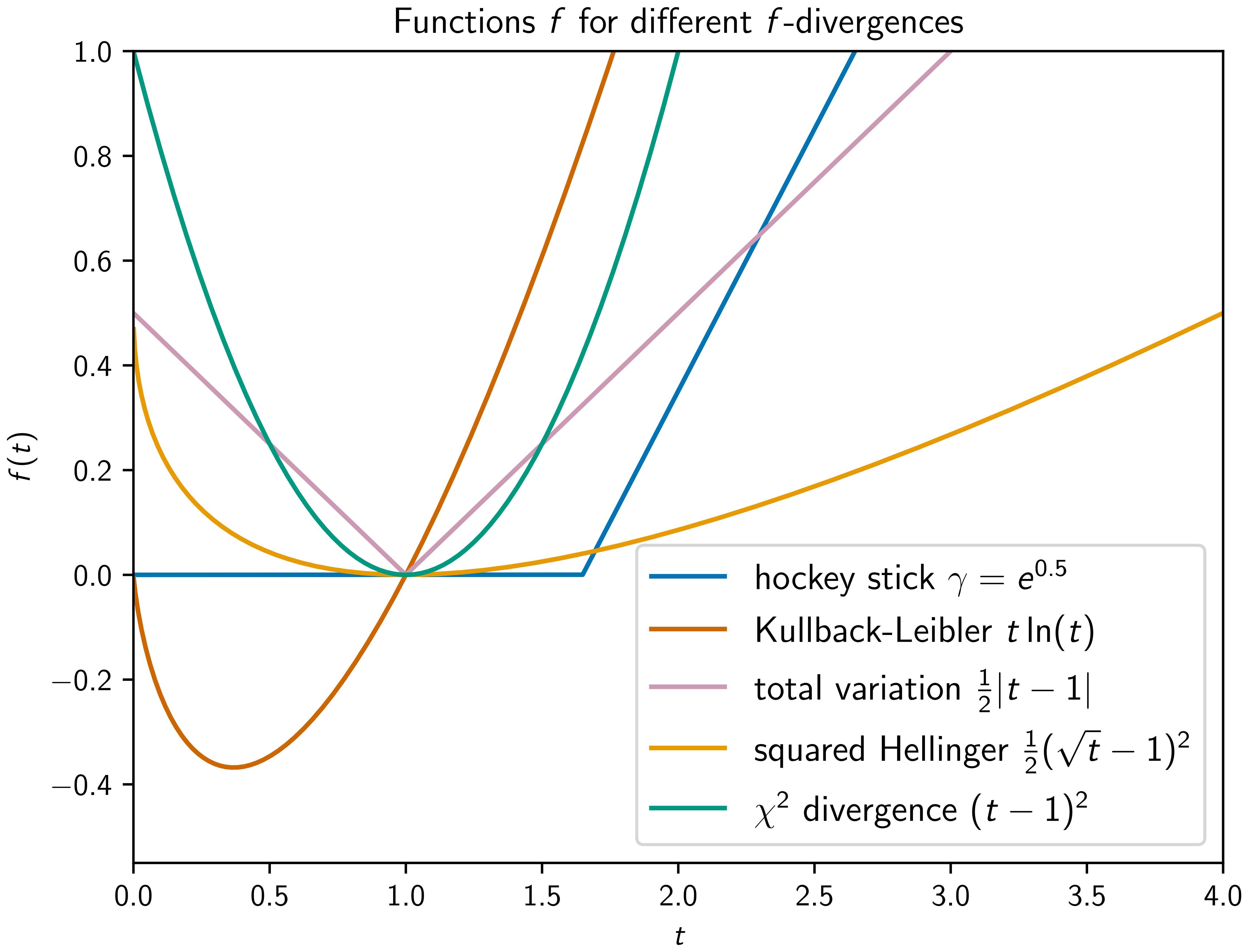
$$D_f(\mu\|\nu) = \mathbb{E}_\nu\left[f\left(\frac{d\mu}{d\nu}\right)\right].$$

If we choose $f(t) = (t-\gamma)^+$ we get the "hockey stick", or E_γ divergence:

Rényi (1961), Cziszár (1963), Morimoto (1963), Ali, Silvey (1966), Csiszár (1967), Polyanskiy, Poor, Verdu (2010), Balle, Barthe, Gaboardi, Geumlek (2019)

Generalized divergences and the 🏒 divergence

How different are these two distributions?



More generally, look at f-divergences: for any convex $f(\,\cdot\,)$,

$$D_f(\mu\|\nu) = \mathbb{E}_\nu\left[f\left(\frac{d\mu}{d\nu}\right)\right].$$

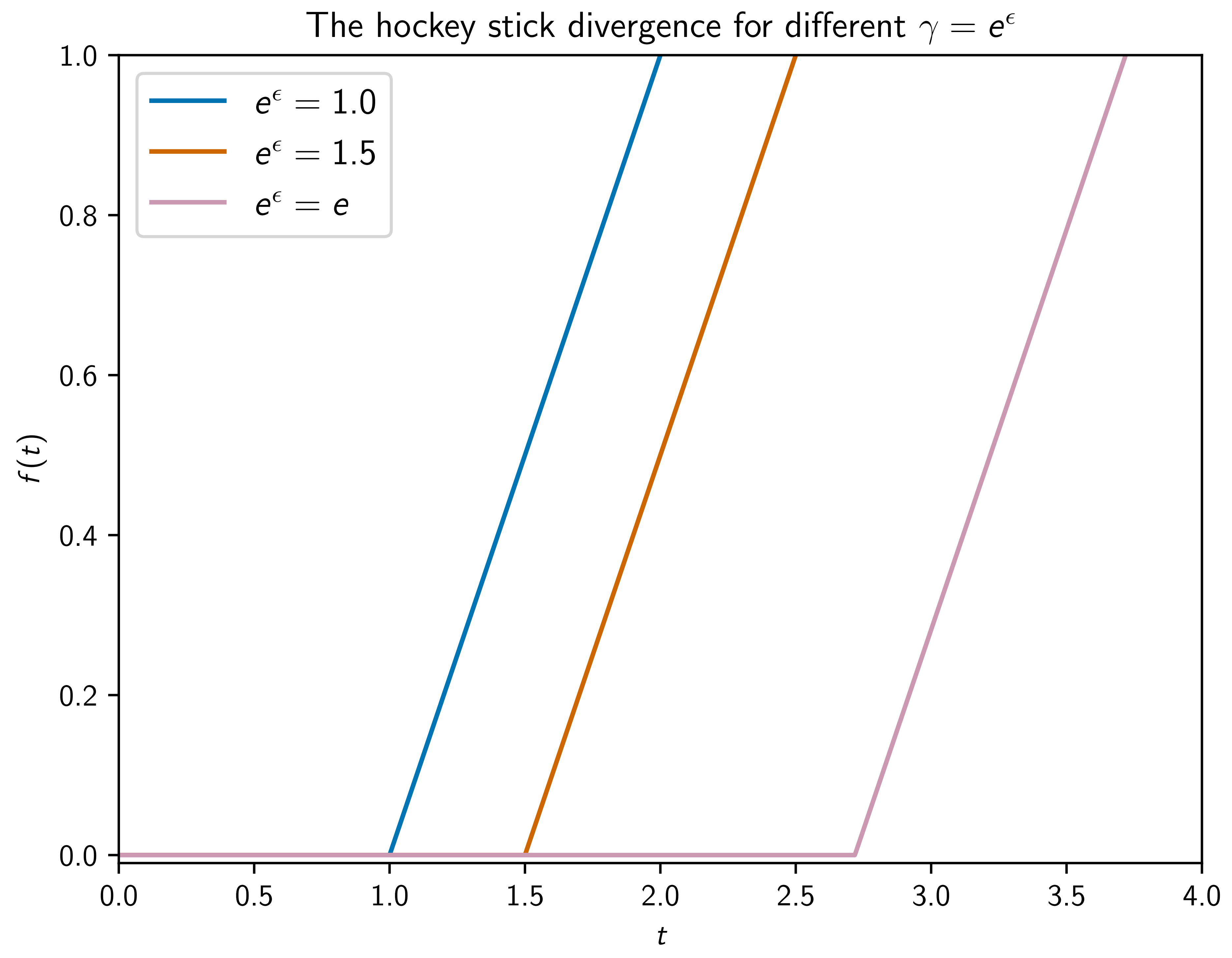
If we choose $f(t) = (t-\gamma)^+$ we get the "hockey stick", or E_γ divergence:

$$\mathrm{E}_\gamma(\mu\|\nu) = \int_\Omega \left(\frac{d\mu}{d\nu} - \gamma\right)^+ d\nu = \sup_A \left[\mu(A) - \gamma\nu(A)\right].$$

Rényi (1961), Cziszár (1963), Morimoto (1963), Ali, Silvey (1966), Csiszár (1967), Polyanskiy, Poor, Verdu (2010), Balle, Barthe, Gaboardi, Geumlek (2019)

The PLRV and divergences

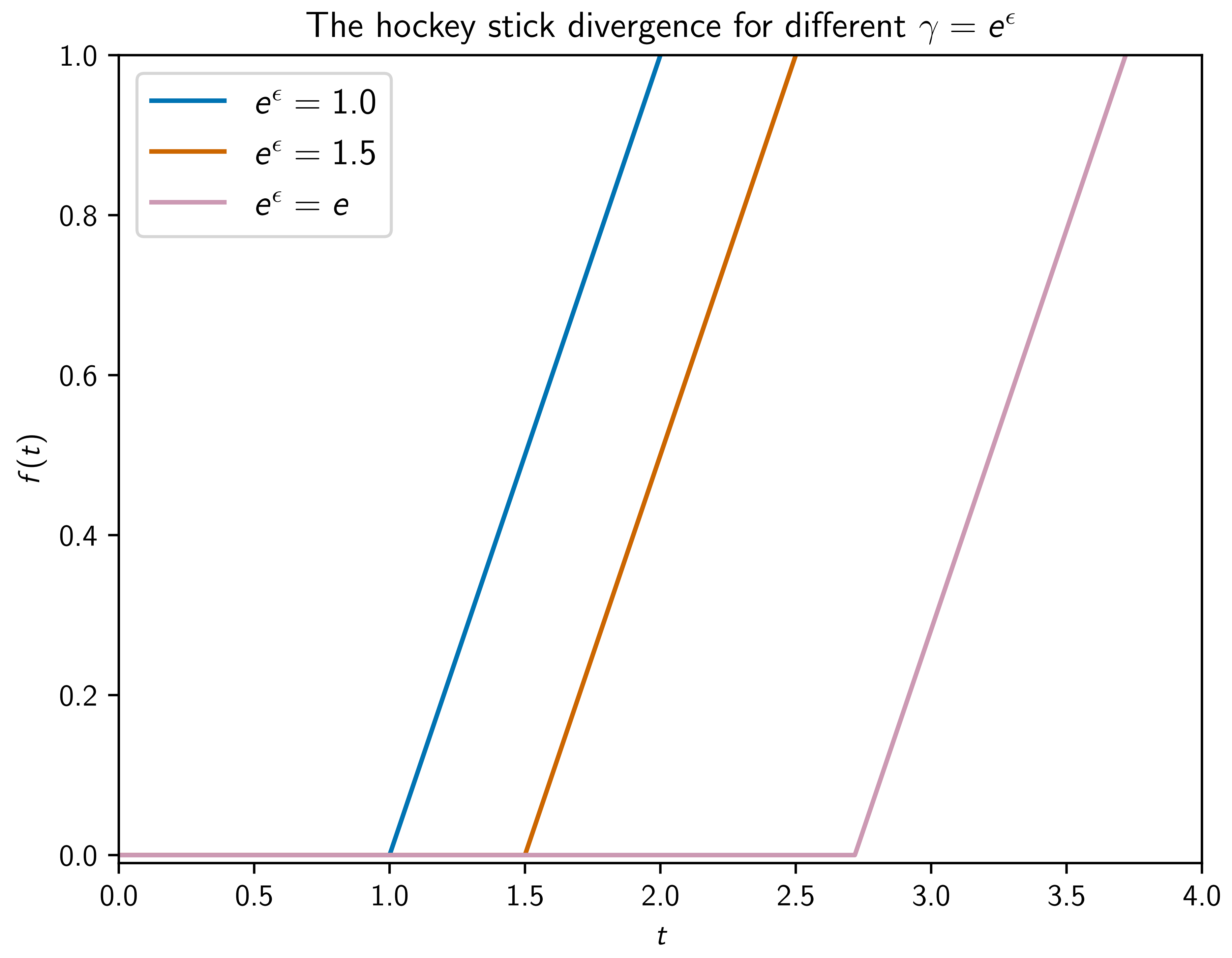
Interpreting DP as an f-divergence



Zhu, Dong, Wang (2022)

The PLRV and divergences

Interpreting DP as an f-divergence

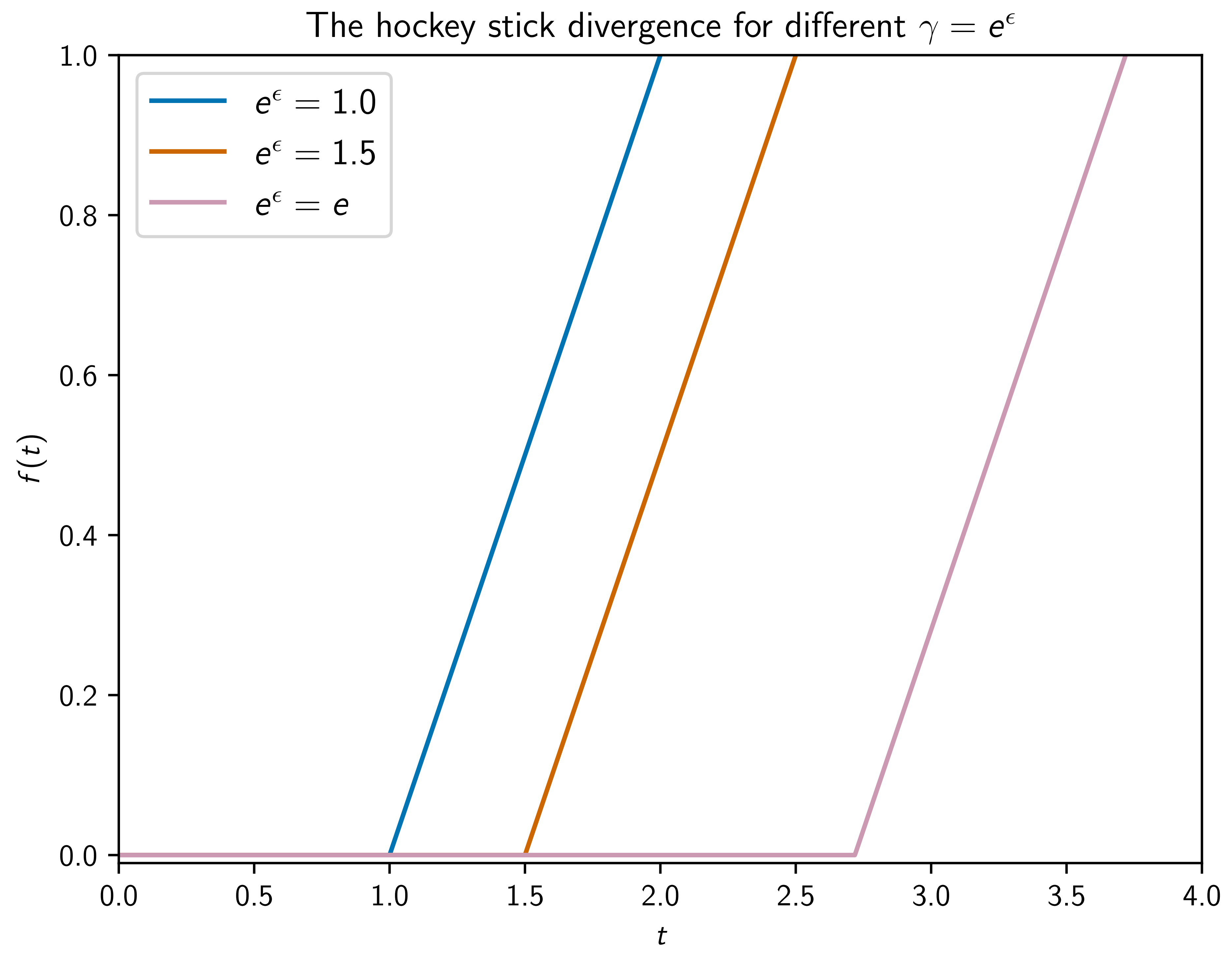


For an (ϵ, δ)-DP mechanism $P_{Y|X}$ we can take $\gamma = e^{\epsilon}$ to get:

Zhu, Dong, Wang (2022)

The PLRV and divergences

Interpreting DP as an f-divergence



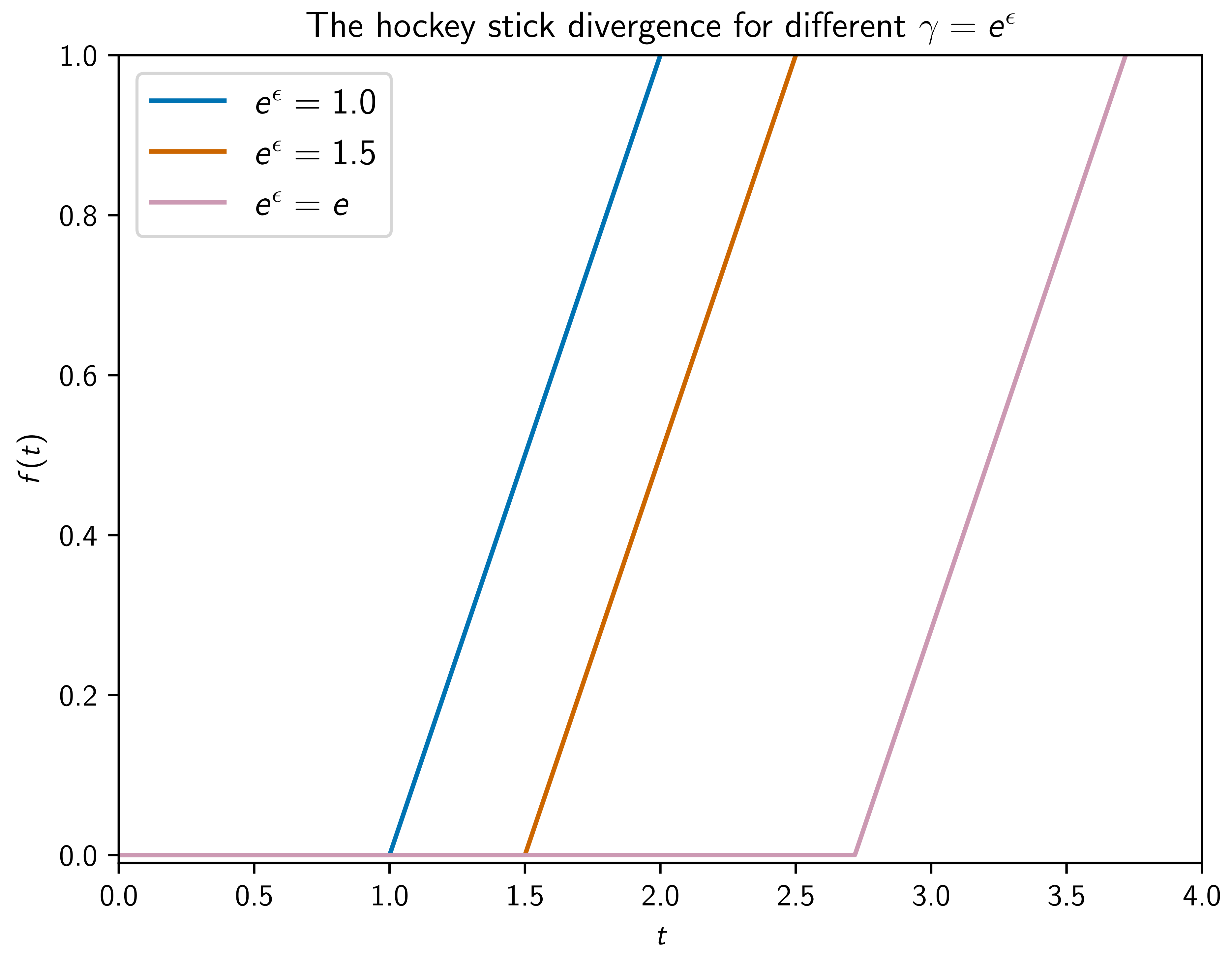
For an (ϵ, δ)-DP mechanism $P_{Y|X}$ we can take $\gamma = e^{\epsilon}$ to get:

$$\delta_{P_{Y|X}}(\varepsilon) = \sup_{x \sim x'} \mathsf{E}_{e^{\varepsilon}}\left(P_{Y|X=x} \middle\| P_{Y|X=x'}\right) = \sup_{x \sim x'} \mathbb{E}\left[\left(1 - e^{\varepsilon - L}\right)^{+}\right].$$

Zhu, Dong, Wang (2022)

The PLRV and divergences

Interpreting DP as an f-divergence



For an (ϵ, δ)-DP mechanism $P_{Y|X}$ we can take $\gamma = e^\epsilon$ to get:

$$\delta_{P_{Y|X}}(\varepsilon) = \sup_{x \sim x'} \mathsf{E}_{e^\varepsilon}\left(P_{Y|X=x} \middle\| P_{Y|X=x'}\right) = \sup_{x \sim x'} \mathbb{E}\left[\left(1 - e^{\varepsilon - L}\right)^+\right].$$

Where L is the PLRV corresponding to (μ, ν).

Zhu, Dong, Wang (2022)

Dominating pairs of distributions

Interpreting DP as an f-divergence

Zhu, Dong, Wang (2022)

Dominating pairs of distributions

Interpreting DP as an f-divergence

We would like to find a **tight dominating pair** of distributions (μ, ν) corresponding to a pair of inputs x and x':

Zhu, Dong, Wang (2022)

Dominating pairs of distributions

Interpreting DP as an f-divergence

We would like to find a **tight dominating pair** of distributions (μ, ν) corresponding to a pair of inputs x and x':

$$\sup_{x \sim x'} \mathsf{E}_\gamma(P_{Y|X=x} \| P_{Y|X=x'}) = \mathsf{E}_\gamma(\mu \| \nu).$$

Zhu, Dong, Wang (2022)

Dominating pairs of distributions

Interpreting DP as an f-divergence

We would like to find a **tight dominating pair** of distributions (μ, ν) corresponding to a pair of inputs x and x':

$$\sup_{x \sim x'} \mathsf{E}_\gamma(P_{Y|X=x} \| P_{Y|X=x'}) = \mathsf{E}_\gamma(\mu \| \nu).$$

However, this does not mean (μ, ν) corresponds to a **worst-case pair** of neighboring inputs. In fact, a worst-case pair may not exist!

Zhu, Dong, Wang (2022)

Dominating pairs of distributions

Interpreting DP as an f-divergence

We would like to find a **tight dominating pair** of distributions (μ, ν) corresponding to a pair of inputs x and x':

$$\sup_{x \sim x'} \mathsf{E}_\gamma(P_{Y|X=x} \| P_{Y|X=x'}) = \mathsf{E}_\gamma(\mu \| \nu).$$

However, this does not mean (μ, ν) corresponds to a **worst-case pair** of neighboring inputs. In fact, a worst-case pair may not exist!

- It's sufficient to look at μ and ν to be to univariate distributions on $[0,1)$.

Zhu, Dong, Wang (2022)

Dominating pairs of distributions

Interpreting DP as an f-divergence

We would like to find a **tight dominating pair** of distributions (μ, ν) corresponding to a pair of inputs x and x':

$$\sup_{x \sim x'} \mathsf{E}_\gamma(P_{Y|X=x} \| P_{Y|X=x'}) = \mathsf{E}_\gamma(\mu \| \nu).$$

However, this does not mean (μ, ν) corresponds to a **worst-case pair** of neighboring inputs. In fact, a worst-case pair may not exist!

- It's sufficient to look at μ and ν to be to univariate distributions on $[0,1)$.
- We can use these dominating pairs to bound the loss for compositions.

Zhu, Dong, Wang (2022)

Composition and divergences

Adding things up

With thanks to Flavio Calmon, Oliver Kosut, and Shahab Asoodeh!

Composition and divergences

Adding things up

If we have multiple releases with PLRVs $L_1, L_2, \ldots, L_k$:

With thanks to Flavio Calmon, Oliver Kosut, and Shahab Asoodeh!

Composition and divergences

Adding things up

If we have multiple releases with PLRVs $L_1, L_2, \ldots, L_k$:

$$L = L_1 + L_2 + \cdots L_k.$$

With thanks to Flavio Calmon, Oliver Kosut, and Shahab Asoodeh!

Composition and divergences

Adding things up

If we have multiple releases with PLRVs $L_1, L_2, \ldots, L_k$:

$$L = L_1 + L_2 + \cdots L_k.$$

Building from measure concentration (Dwork, Rothblum, Vadhan (2010)) or exact composition (Kairouz, Oh, Viswanath (2015), Murtagh, Vadhan (2016)) there are (at least) three main branches of work on composition:

With thanks to Flavio Calmon, Oliver Kosut, and Shahab Asoodeh!

Composition and divergences

Adding things up

If we have multiple releases with PLRVs $L_1, L_2, \ldots, L_k$:

$$L = L_1 + L_2 + \cdots L_k.$$

Building from measure concentration (Dwork, Rothblum, Vadhan (2010)) or exact composition (Kairouz, Oh, Viswanath (2015), Murtagh, Vadhan (2016)) there are (at least) three main branches of work on composition:

- Large deviations

With thanks to Flavio Calmon, Oliver Kosut, and Shahab Asoodeh!

Composition and divergences

Adding things up

If we have multiple releases with PLRVs $L_1, L_2, \ldots, L_k$:

$$L = L_1 + L_2 + \cdots L_k.$$

Building from measure concentration (Dwork, Rothblum, Vadhan (2010)) or exact composition (Kairouz, Oh, Viswanath (2015), Murtagh, Vadhan (2016)) there are (at least) three main branches of work on composition:

- Large deviations
- Central limit theorem

With thanks to Flavio Calmon, Oliver Kosut, and Shahab Asoodeh!

Composition and divergences

Adding things up

If we have multiple releases with PLRVs $L_1, L_2, \ldots, L_k$:

$$L = L_1 + L_2 + \cdots L_k.$$

Building from measure concentration (Dwork, Rothblum, Vadhan (2010)) or exact composition (Kairouz, Oh, Viswanath (2015), Murtagh, Vadhan (2016)) there are (at least) three main branches of work on composition:

- Large deviations
- Central limit theorem
- Numerical approaches

With thanks to Flavio Calmon, Oliver Kosut, and Shahab Asoodeh!

A taxonomy of composition analyses

(on non-interactive settings, also non-exhaustive)

A taxonomy of composition analyses

(on non-interactive settings, also non-exhaustive)

<u>Large deviations</u>

Bun and Steinke (2016)

Abadi, Chu, Goodfellow, McMahan,
Mironov, Talwar, Zhang (2016)

Mironov (2017)

Wang, Balle, Kasiviswanathan (2018)

A taxonomy of composition analyses

(on non-interactive settings, also non-exhaustive)

<u>Large deviations</u>

Bun and Steinke (2016)

Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, Zhang (2016)

Mironov (2017)

Wang, Balle, Kasiviswanathan (2018)

<u>CLT and characteristic functions</u>

Dong, Roth, Su (2019)

Sommer, Meiser, Mohammadi (2020)

Cannone, Kamath, Steinke (2020)

Zhu, Dong, Wang (2022)

A taxonomy of composition analyses
(on non-interactive settings, also non-exhaustive)

<u>Large deviations</u>

Bun and Steinke (2016)

Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, Zhang (2016)

Mironov (2017)

Wang, Balle, Kasiviswanathan (2018)

<u>CLT and characteristic functions</u>

Dong, Roth, Su (2019)

Sommer, Meiser, Mohammadi (2020)

Cannone, Kamath, Steinke (2020)

Zhu, Dong, Wang (2022)

<u>Numerical methods</u>

Koskela, Jälkö, Prediger, Honkela (2021/2021)

Gopi, Lee, Wutschitz (2021)

Ghazi, Kamath, Kumar, Manurangsi (2022)

Doroshenko, Ghazi, Kamath, Kumar, Manurangsi (2022)

A taxonomy of composition analyses

(on non-interactive settings, also non-exhaustive)

Large deviations

Bun and Steinke (2016)

Abadi, Chu, Goodfellow, McMahan, Mironov, Talwar, Zhang (2016)

Mironov (2017)

Wang, Balle, Kasiviswanathan (2018)

CLT and characteristic functions

Dong, Roth, Su (2019)

Sommer, Meiser, Mohammadi (2020)

Cannone, Kamath, Steinke (2020)

Zhu, Dong, Wang (2022)

Numerical methods

Koskela, Jälkö, Prediger, Honkela (2021/2021)

Gopi, Lee, Wutschitz (2021)

Ghazi, Kamath, Kumar, Manurangsi (2022)

Doroshenko, Ghazi, Kamath, Kumar, Manurangsi (2022)

Warning about subsampling!

Lebeda, Regehr, Kamath, Steinke (2024)

Chua, Ghazi, Kamath, Kumar, Manurangsi, Sinha, Zhang (2024)

Tilting a distribution

Maintaining exactness for composition

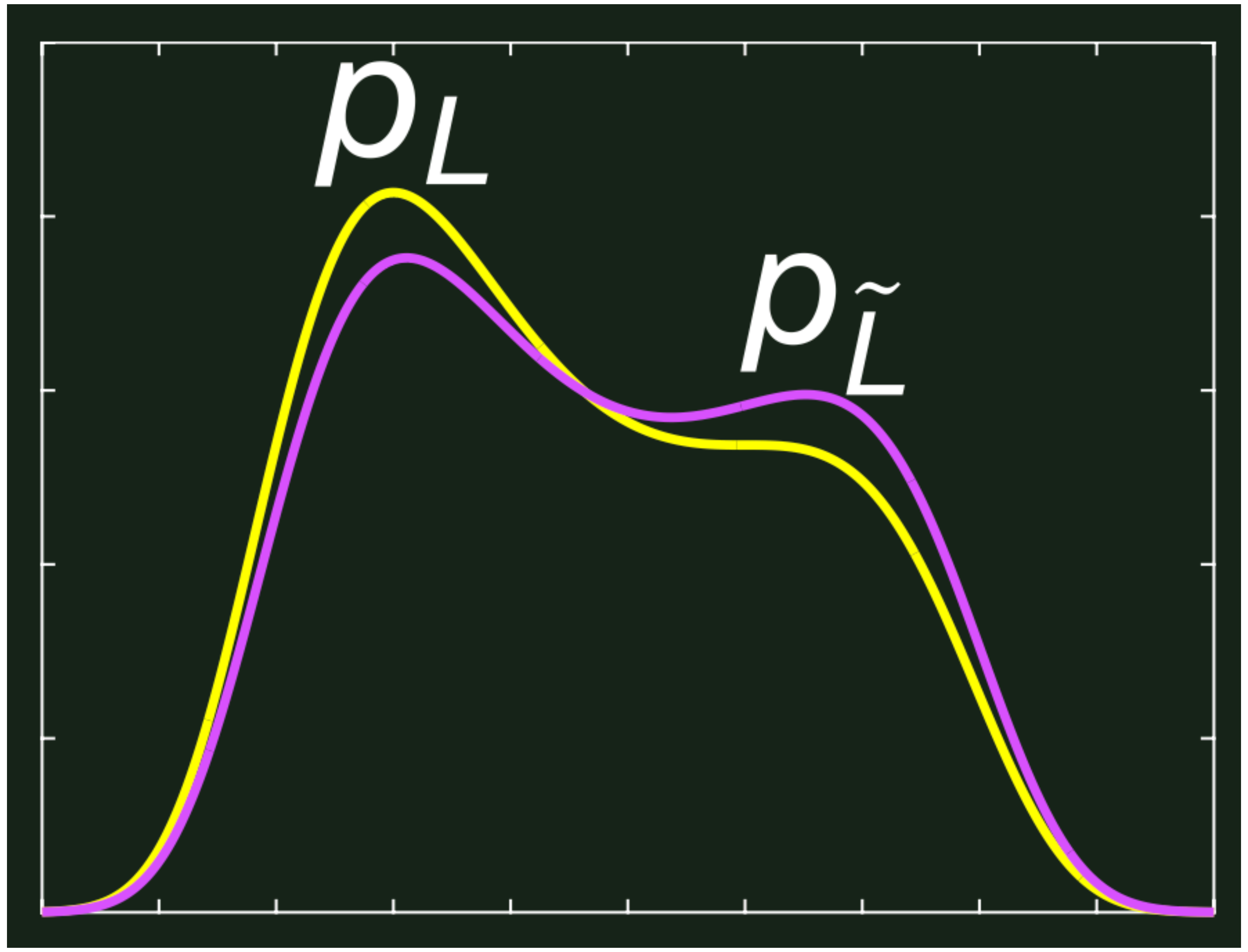


Figure: Oliver Kosut

Alghamdi, Gomez, Asoodeh, Calmon, Kosut, Sankar (2023)

Tilting a distribution

Maintaining exactness for composition

Look at the cumulant generating function:

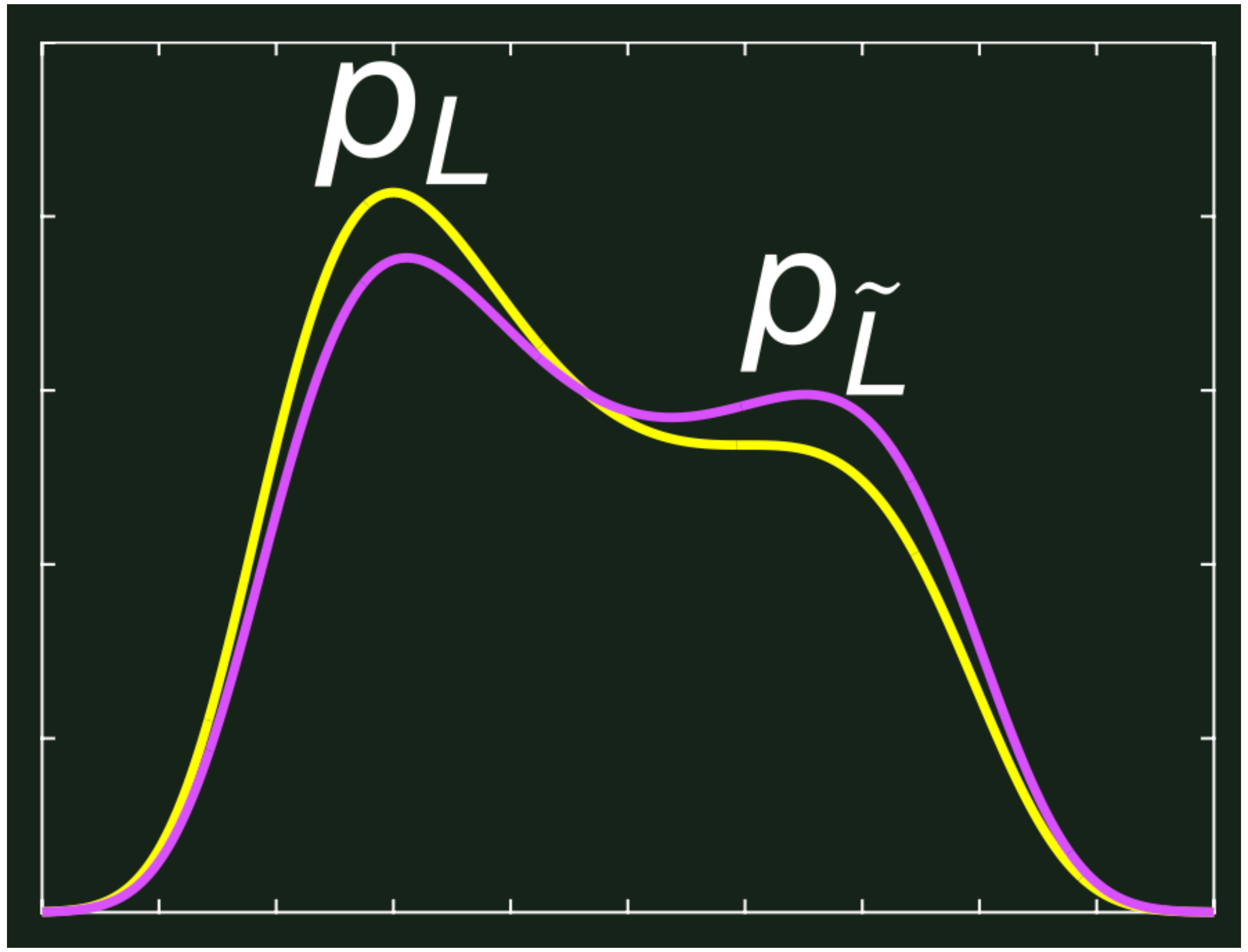


Figure: Oliver Kosut

Alghamdi, Gomez, Asoodeh, Calmon, Kosut, Sankar (2023)

Tilting a distribution

Maintaining exactness for composition

Look at the cumulant generating function:

$$K_L(t) = \log \mathbb{E}\left[e^{tL}\right]$$

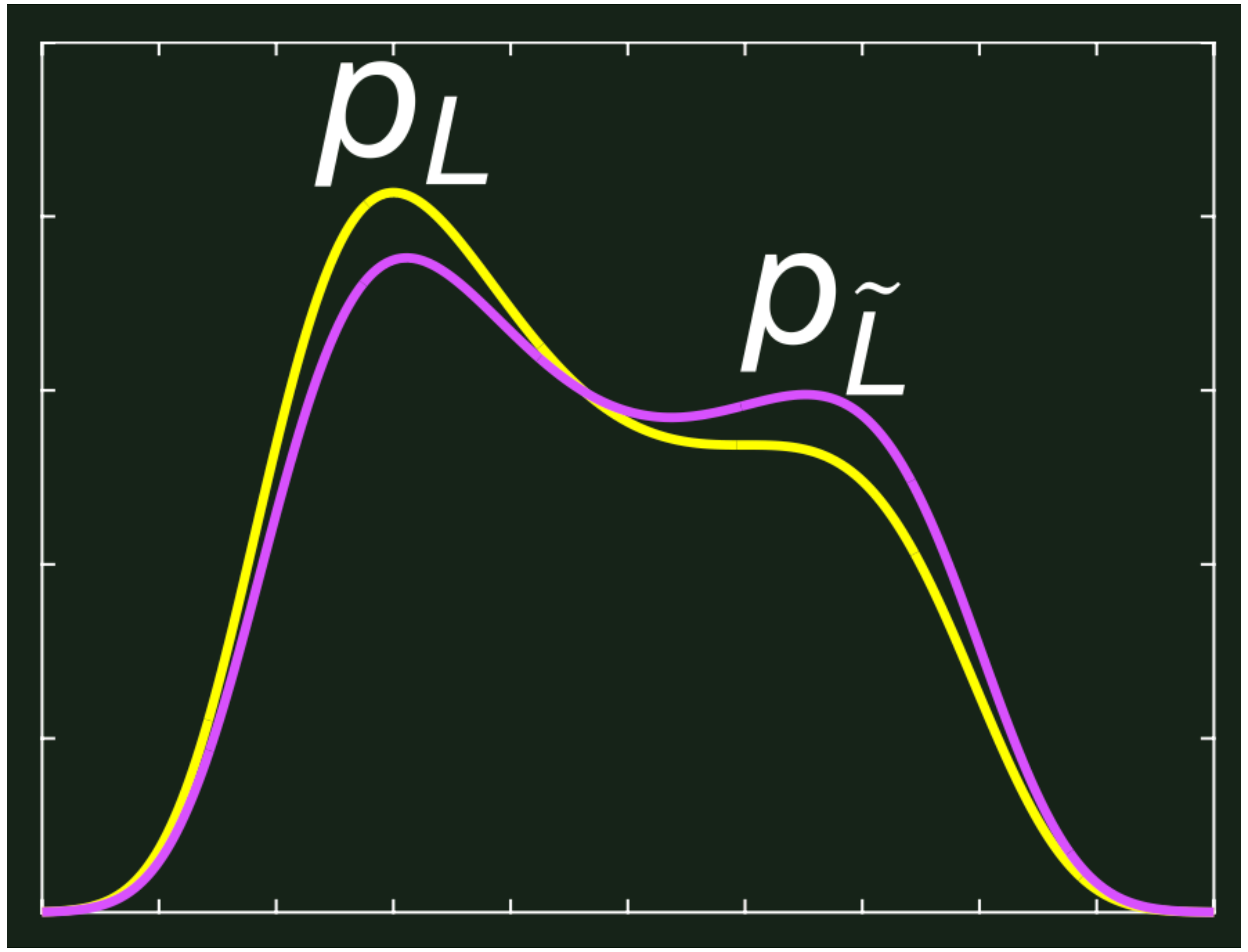


Figure: Oliver Kosut

Alghamdi, Gomez, Asoodeh, Calmon, Kosut, Sankar (2023)

Tilting a distribution

Maintaining exactness for composition

Look at the cumulant generating function:

$$K_L(t) = \log \mathbb{E}\left[e^{tL}\right]$$

and the **"tilted" random variable** (for continuous L)

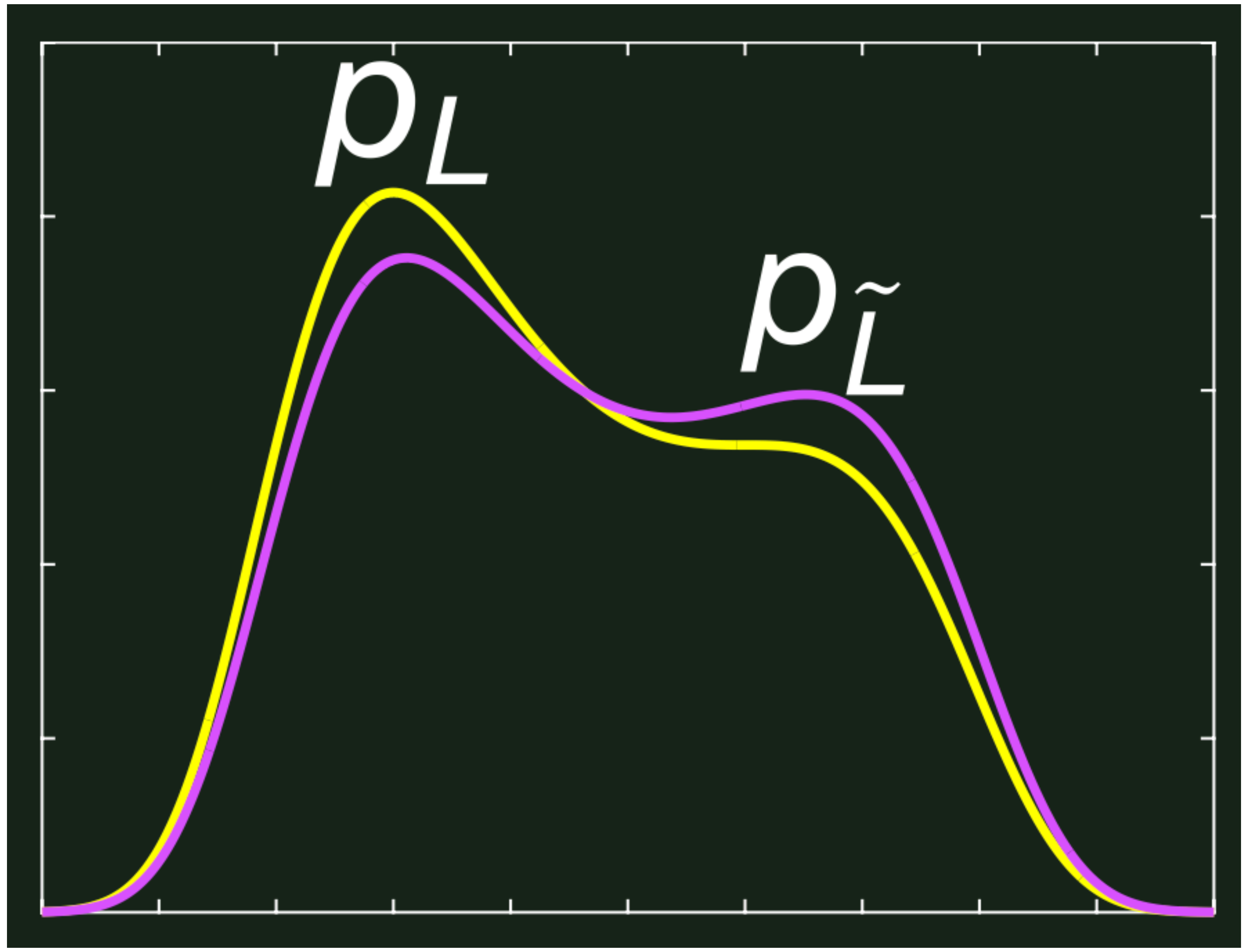


Figure: Oliver Kosut

Alghamdi, Gomez, Asoodeh, Calmon, Kosut, Sankar (2023)

Tilting a distribution

Maintaining exactness for composition

Look at the cumulant generating function:

$$K_L(t) = \log \mathbb{E}\left[e^{tL}\right]$$

and the **"tilted" random variable** (for continuous L)

$$\widetilde{L}_t \sim p_{\widetilde{L}}(y) = e^{-K_L(t)+ty} p_L(y)$$

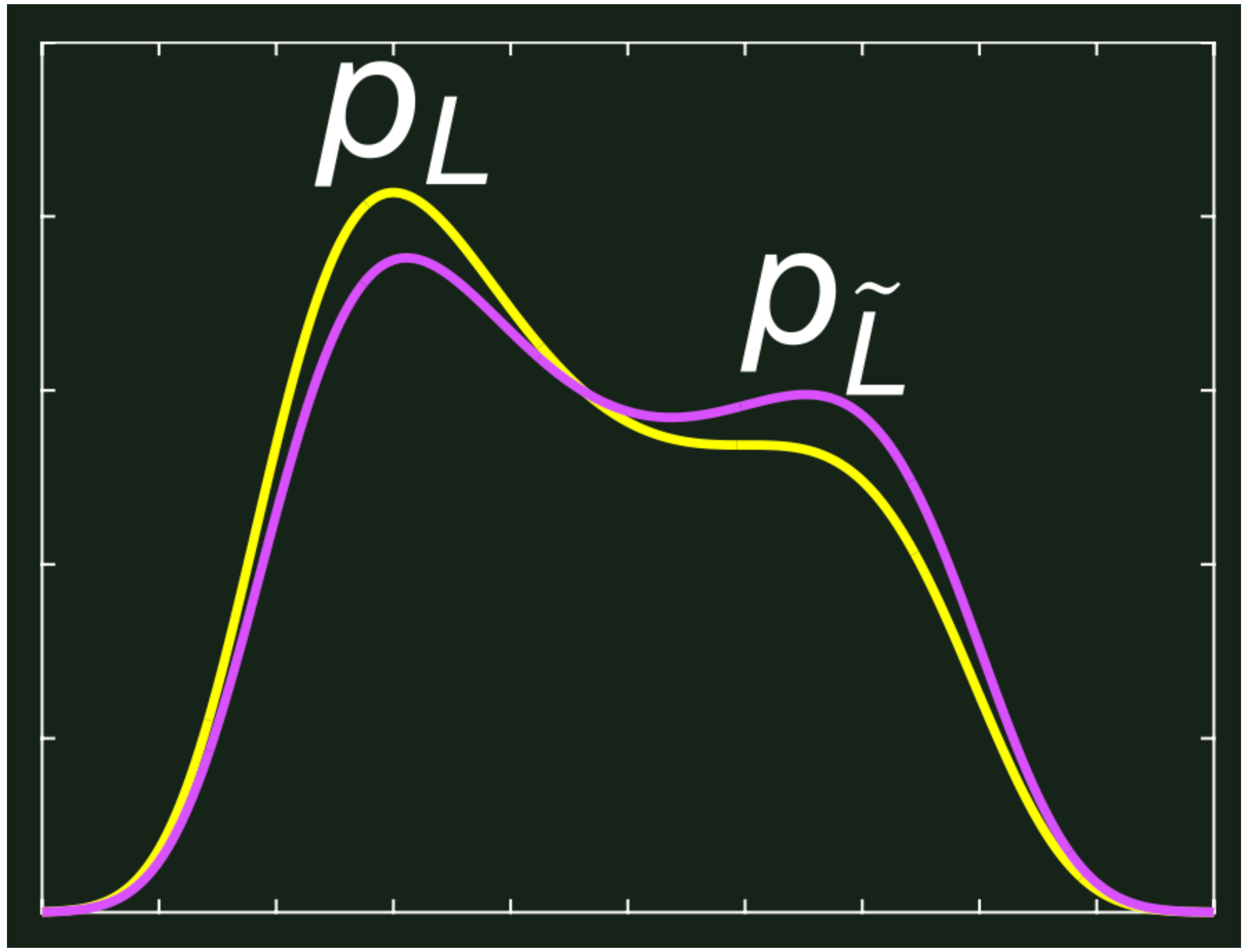


Figure: Oliver Kosut

Alghamdi, Gomez, Asoodeh, Calmon, Kosut, Sankar (2023)

Tilting a distribution

Maintaining exactness for composition

Look at the cumulant generating function:

$$K_L(t) = \log \mathbb{E}\left[e^{tL}\right]$$

and the **"tilted" random variable** (for continuous L)

$$\widetilde{L}_t \sim p_{\widetilde{L}}(y) = e^{-K_L(t)+ty} p_L(y)$$

Then under some mild assumptions

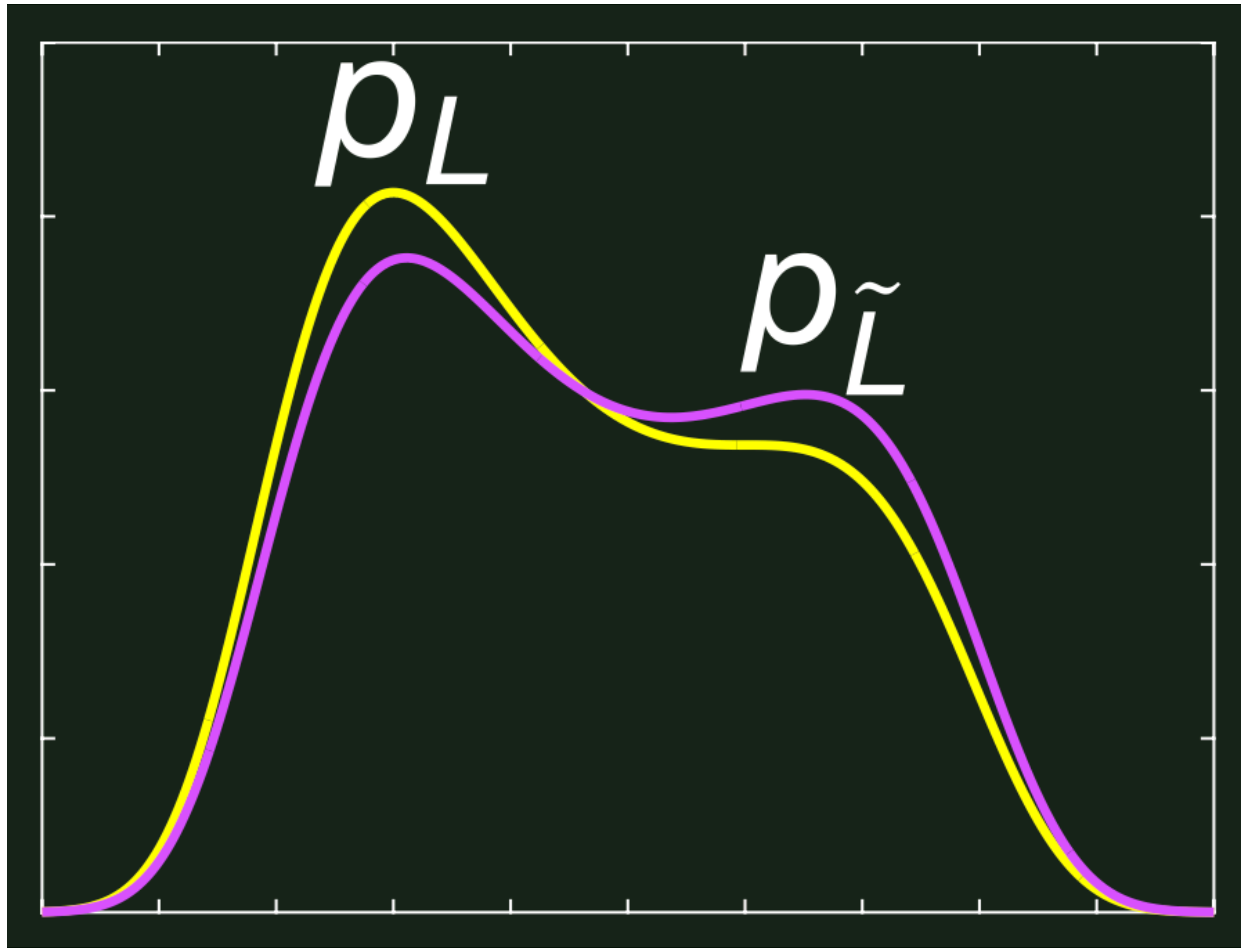


Figure: Oliver Kosut

Alghamdi, Gomez, Asoodeh, Calmon, Kosut, Sankar (2023)

Tilting a distribution

Maintaining exactness for composition

Look at the cumulant generating function:

$$K_L(t) = \log \mathbb{E}\left[e^{tL}\right]$$

and the **"tilted" random variable** (for continuous L)

$$\widetilde{L}_t \sim p_{\widetilde{L}}(y) = e^{-K_L(t)+ty} p_L(y)$$

Then under some mild assumptions

$$\delta(\varepsilon) = \frac{1}{2\pi i} \int_{t-\infty i}^{t+\infty i} e^{K_L(z)-\varepsilon z-\log z-\log(1+z)} dz.$$

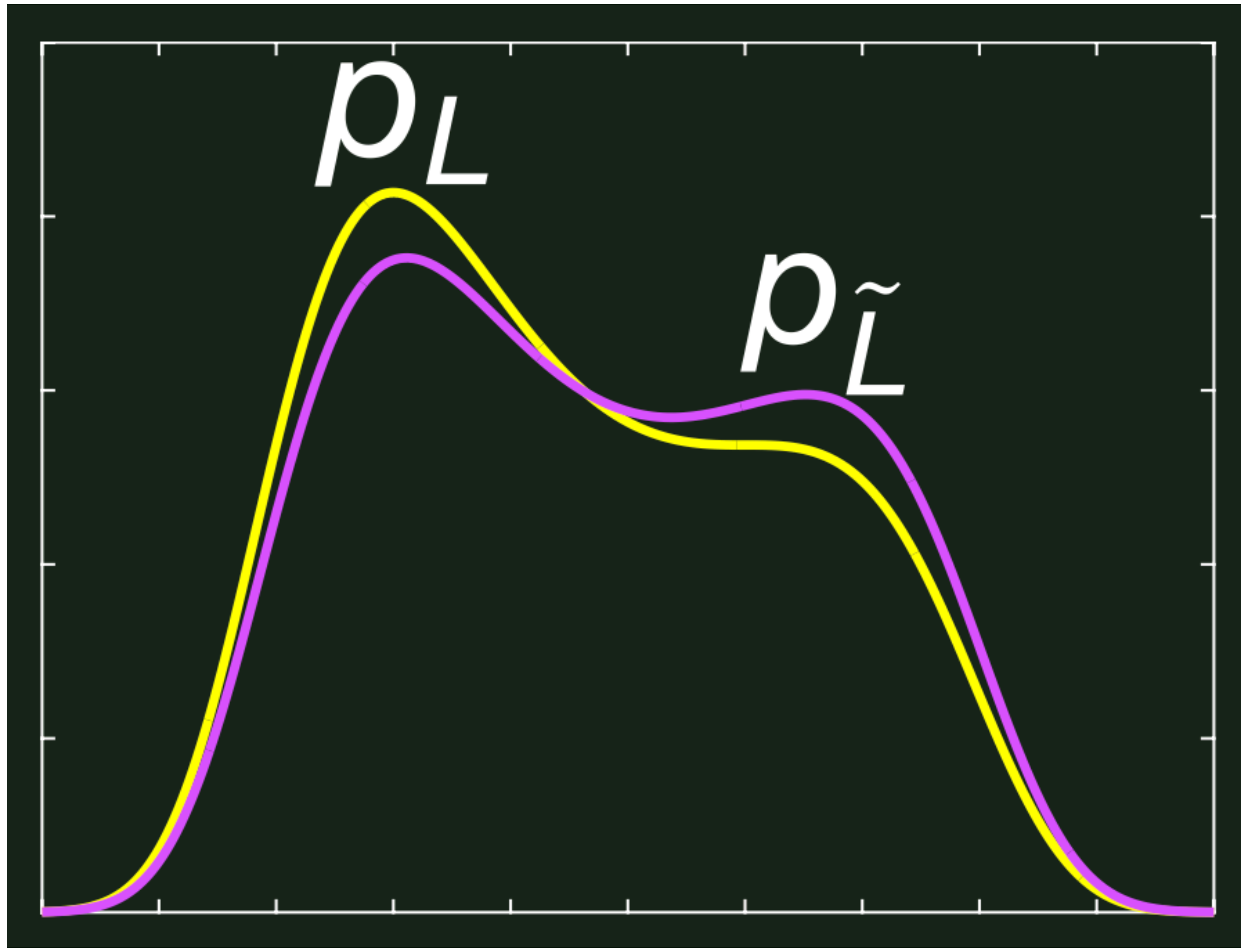


Figure: Oliver Kosut

Alghamdi, Gomez, Asoodeh, Calmon, Kosut, Sankar (2023)

Tilting a distribution

Maintaining exactness for composition

Look at the cumulant generating function:

$$K_L(t) = \log \mathbb{E}\left[e^{tL}\right]$$

and the **“tilted” random variable** (for continuous L)

$$\widetilde{L}_t \sim p_{\widetilde{L}}(y) = e^{-K_L(t)+ty} p_L(y)$$

Then under some mild assumptions

$$\delta(\varepsilon) = \frac{1}{2\pi i} \int_{t-\infty i}^{t+\infty i} e^{K_L(z)-\varepsilon z-\log z-\log(1+z)} dz.$$

Can use this to derive a **“saddle-point” accountant** in terms of the exponent.

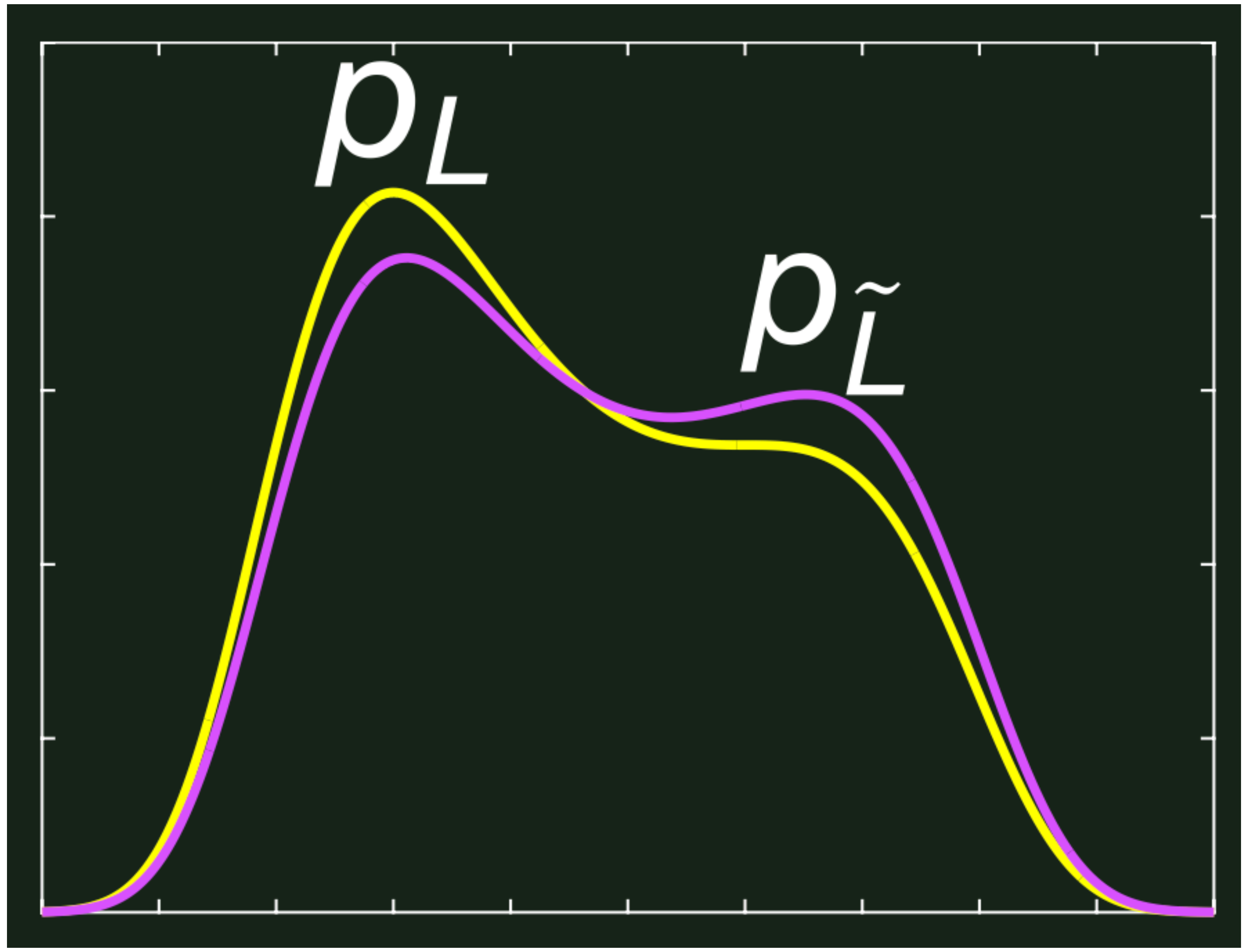


Figure: Oliver Kosut

Alghamdi, Gomez, Asoodeh, Calmon, Kosut, Sankar (2023)

Tilting in other contexts

(Beyond *Don Quixote*)

A tiling (not tilting) by
M.C. Escher, not F. Esscher

Tilting in other contexts

(Beyond *Don Quixote*)

A tiling (not tilting) by
M.C. Escher, not F. Esscher

Tilting is an old idea (originally in Esscher (1932)) and used in risk analysis, rejection sampling/importance sampling, and elsewhere. Connections to:

Tilting in other contexts

(Beyond *Don Quixote*)

A tiling (not tilting) by
M.C. Escher, not F. Esscher

Tilting is an old idea (originally in Esscher (1932)) and used in risk analysis, rejection sampling/importance sampling, and elsewhere. Connections to:

- Exponential families

Tilting in other contexts

(Beyond *Don Quixote*)

A tiling (not tilting) by
M.C. Escher, not F. Esscher

Tilting is an old idea (originally in Esscher (1932)) and used in risk analysis, rejection sampling/importance sampling, and elsewhere. Connections to:

- Exponential families
- Edgeworth expansion

Tilting in other contexts

(Beyond *Don Quixote*)

A tiling (not tilting) by
M.C. Escher, not F. Esscher

Tilting is an old idea (originally in Esscher (1932)) and used in risk analysis, rejection sampling/importance sampling, and elsewhere. Connections to:

- Exponential families
- Edgeworth expansion
- Temperature in Boltzman/Gibbs distributions

Tilting in other contexts

(Beyond *Don Quixote*)

A tiling (not tilting) by
M.C. Escher, not F. Esscher

Tilting is an old idea (originally in Esscher (1932)) and used in risk analysis, rejection sampling/importance sampling, and elsewhere. Connections to:

- Exponential families
- Edgeworth expansion
- Temperature in Boltzman/Gibbs distributions

Perhaps of interest to folks here? Botev (2017) uses it to exact iid simulation from the truncated multivariate normal distribution.

Shichiri Beach in Sagami Province

相州七里浜

Soshū Shichiri-ga-hama

Vista 4

contraction coefficients/iteration

Maximum likelihood and ERM

Optimization and privacy

Maximum likelihood and ERM

Optimization and privacy

Most of “modern” machine learning involves optimization problems, including maximum likelihood estimation and empirical risk minimization:

Maximum likelihood and ERM

Optimization and privacy

Most of "modern" machine learning involves optimization problems, including maximum likelihood estimation and empirical risk minimization:

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i).$$

Maximum likelihood and ERM

Optimization and privacy

Most of “modern” machine learning involves optimization problems, including maximum likelihood estimation and empirical risk minimization:

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i).$$

We can use DP to approximate this in a number of ways:

Maximum likelihood and ERM

Optimization and privacy

Most of “modern” machine learning involves optimization problems, including maximum likelihood estimation and empirical risk minimization:

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i).$$

We can use DP to approximate this in a number of ways:

- **“Output perturbation”**: compute the minimizer and add noise.

Maximum likelihood and ERM

Optimization and privacy

Most of “modern” machine learning involves optimization problems, including maximum likelihood estimation and empirical risk minimization:

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i).$$

We can use DP to approximate this in a number of ways:

- **“Output perturbation”**: compute the minimizer and add noise.
- **“Objective perturbation”**: Add a random term to the objective function and minimize it.

Maximum likelihood and ERM

Optimization and privacy

Most of "modern" machine learning involves optimization problems, including maximum likelihood estimation and empirical risk minimization:

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i).$$

We can use DP to approximate this in a number of ways:

- **"Output perturbation"**: compute the minimizer and add noise.
- **"Objective perturbation"**: Add a random term to the objective function and minimize it.
- **"Functional mechanism"**: Add noise to an approximation of the loss function $\ell(\cdot)$.

Maximum likelihood and ERM

Optimization and privacy

Most of “modern” machine learning involves optimization problems, including maximum likelihood estimation and empirical risk minimization:

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i).$$

We can use DP to approximate this in a number of ways:

- **“Output perturbation”**: compute the minimizer and add noise.
- **“Objective perturbation”**: Add a random term to the objective function and minimize it.
- **“Functional mechanism”**: Add noise to an approximation of the loss function $\ell(\cdot)$.

Maximum likelihood and ERM

Optimization and privacy

Most of “modern” machine learning involves optimization problems, including maximum likelihood estimation and empirical risk minimization:

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i).$$

We can use DP to approximate this in a number of ways:

- **“Output perturbation”**: compute the minimizer and add noise.
- **“Objective perturbation”**: Add a random term to the objective function and minimize it.
- **“Functional mechanism”**: Add noise to an approximation of the loss function $\ell(\cdot)$.

[Chaudhuri, Monteleoni, Sarwate 2011]

Maximum likelihood and ERM

Optimization and privacy

Most of "modern" machine learning involves optimization problems, including maximum likelihood estimation and empirical risk minimization:

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}, \mathbf{x}_i).$$

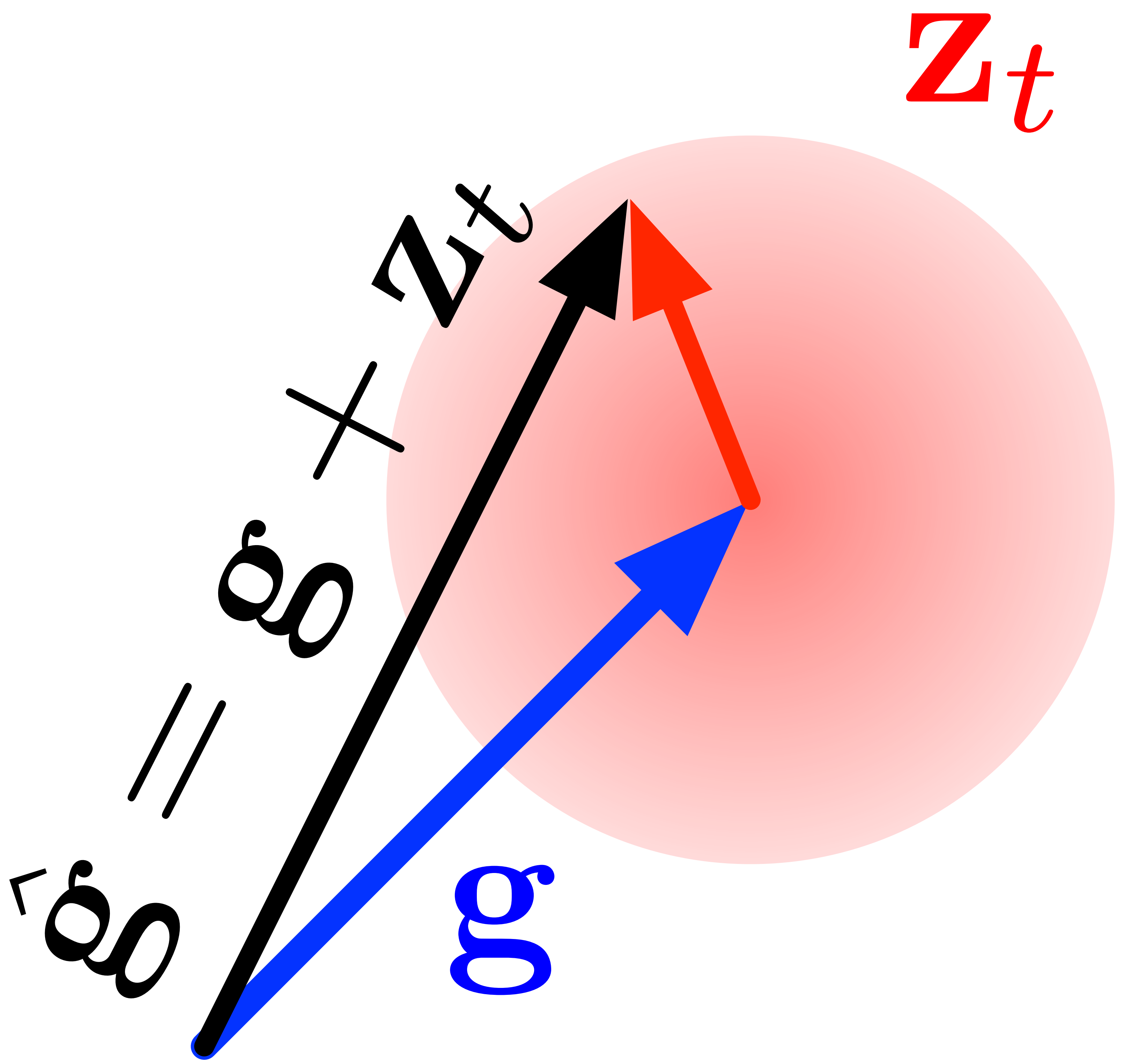
We can use DP to approximate this in a number of ways:

- **"Output perturbation"**: compute the minimizer and add noise.
- **"Objective perturbation"**: Add a random term to the objective function and minimize it.
- **"Functional mechanism"**: Add noise to an approximation of the loss function $\ell(\cdot)$.

[Chaudhuri, Monteleoni, Sarwate 2011]
[Zhang, Zhang, Xiao, Yang, Winslett 2012]

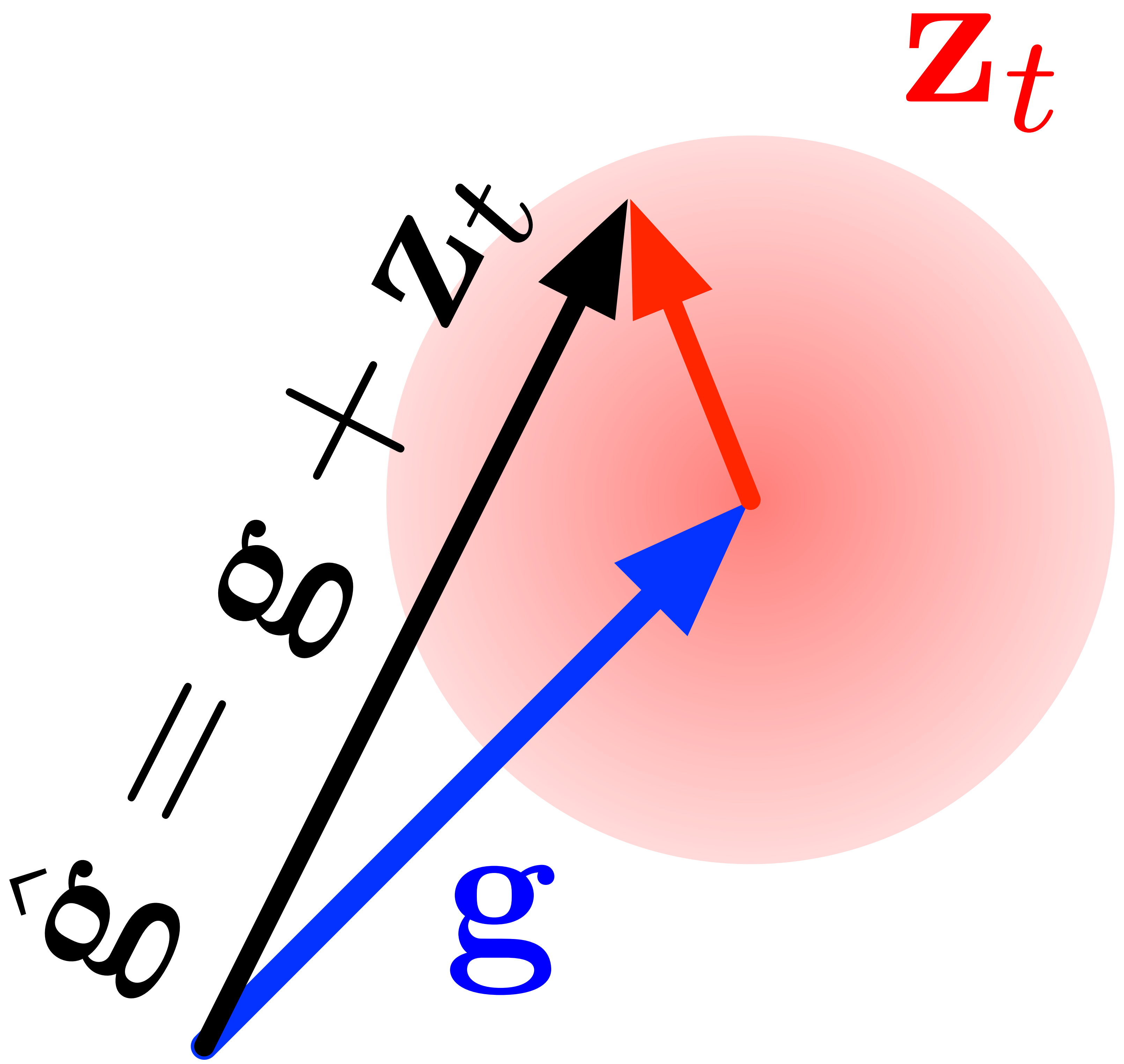
Deep Learning and DP

Privacy for neural networks



Deep Learning and DP

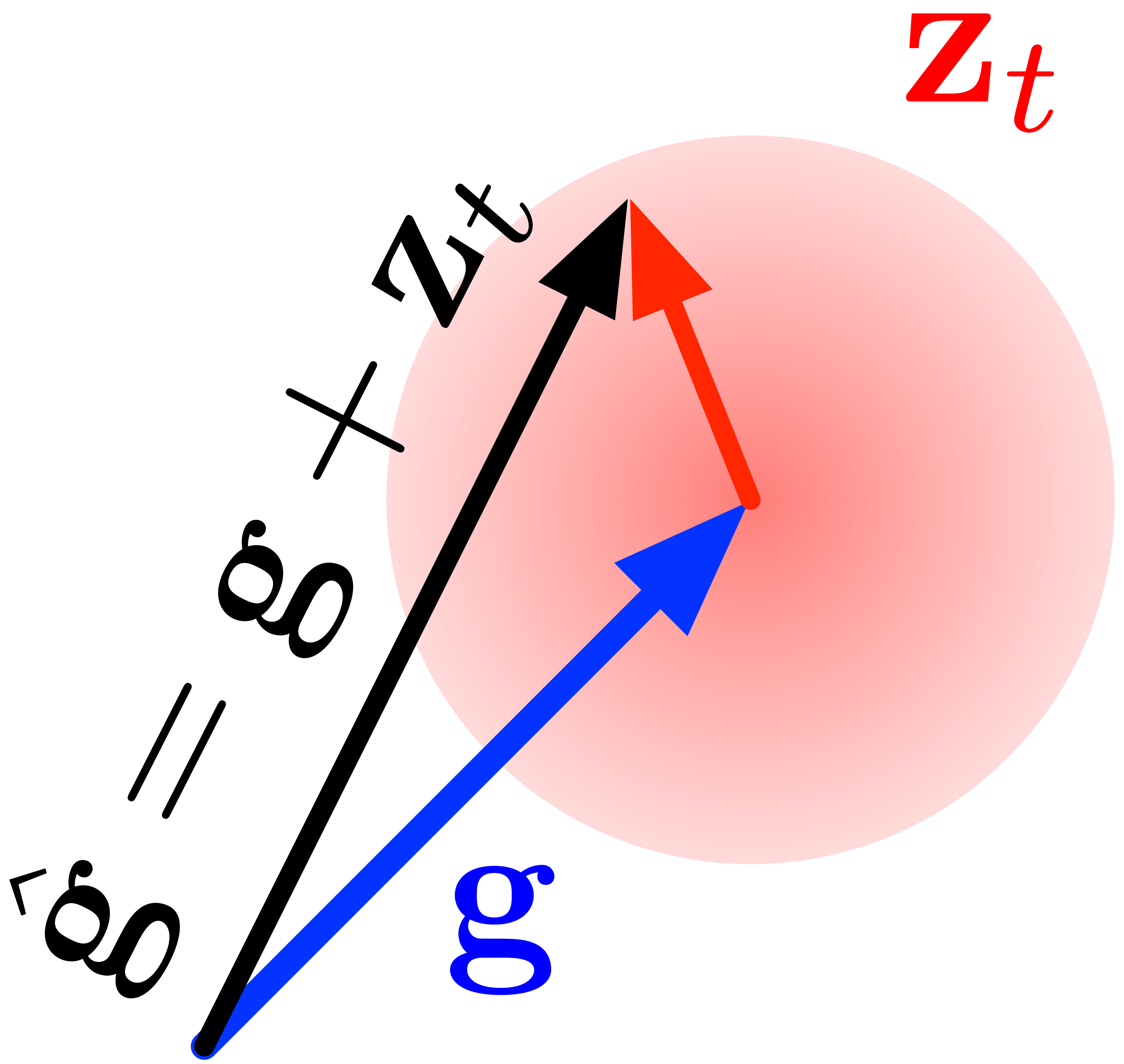
Privacy for neural networks



Deep neural networks (DNNs) also use optimization algorithms in training. To make these private we can add noise to the gradients in stochastic gradient descent (SGD):

Deep Learning and DP

Privacy for neural networks

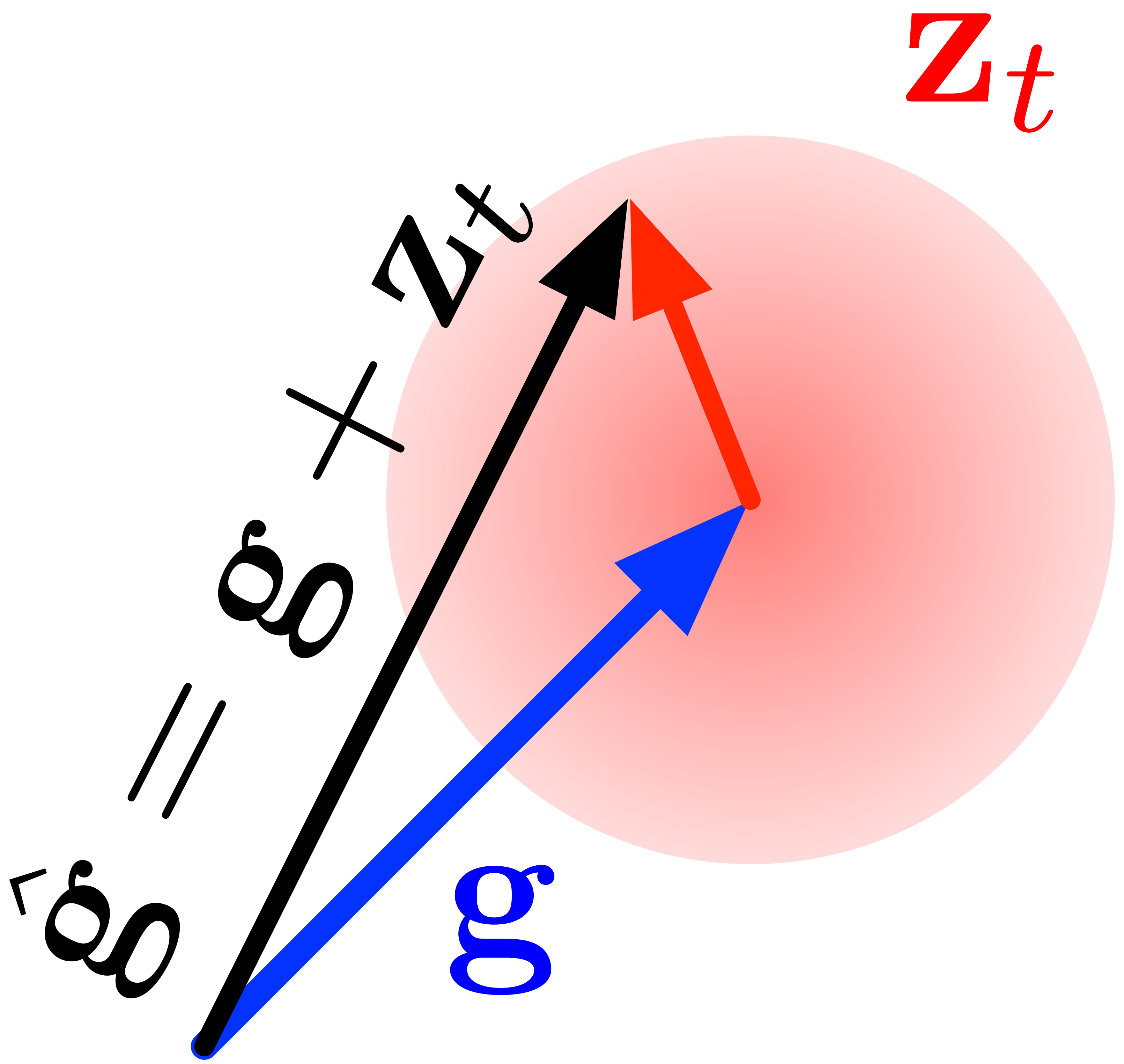


Deep neural networks (DNNs) also use optimization algorithms in training. To make these private we can add noise to the gradients in stochastic gradient descent (SGD):

- Adding noise to gradients provides differential privacy.

Deep Learning and DP

Privacy for neural networks

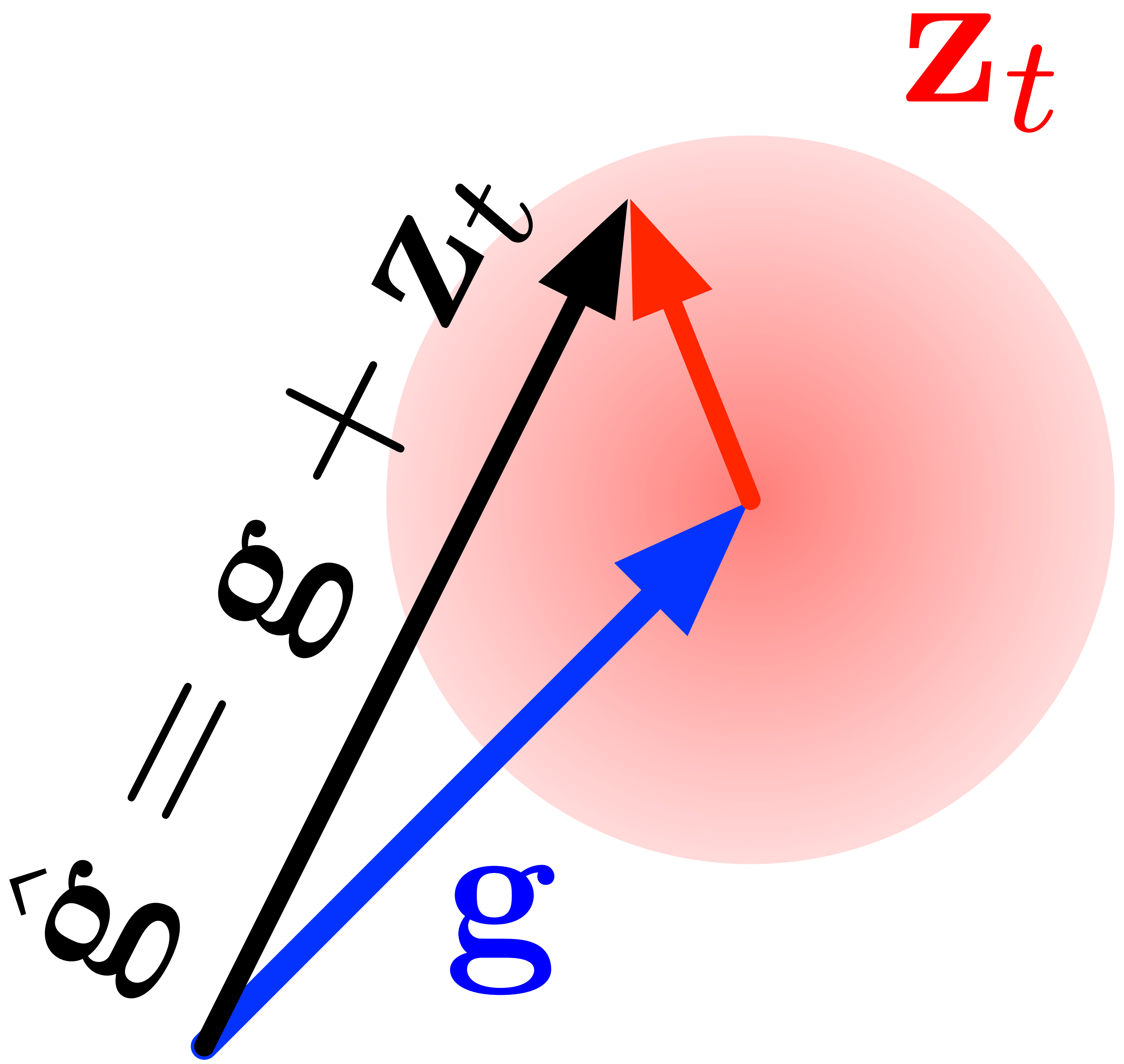


Deep neural networks (DNNs) also use optimization algorithms in training. To make these private we can add noise to the gradients in stochastic gradient descent (SGD):

- Adding noise to gradients provides differential privacy.
- For high-dimensional problems, Gaussian noise is very effective.

Deep Learning and DP

Privacy for neural networks

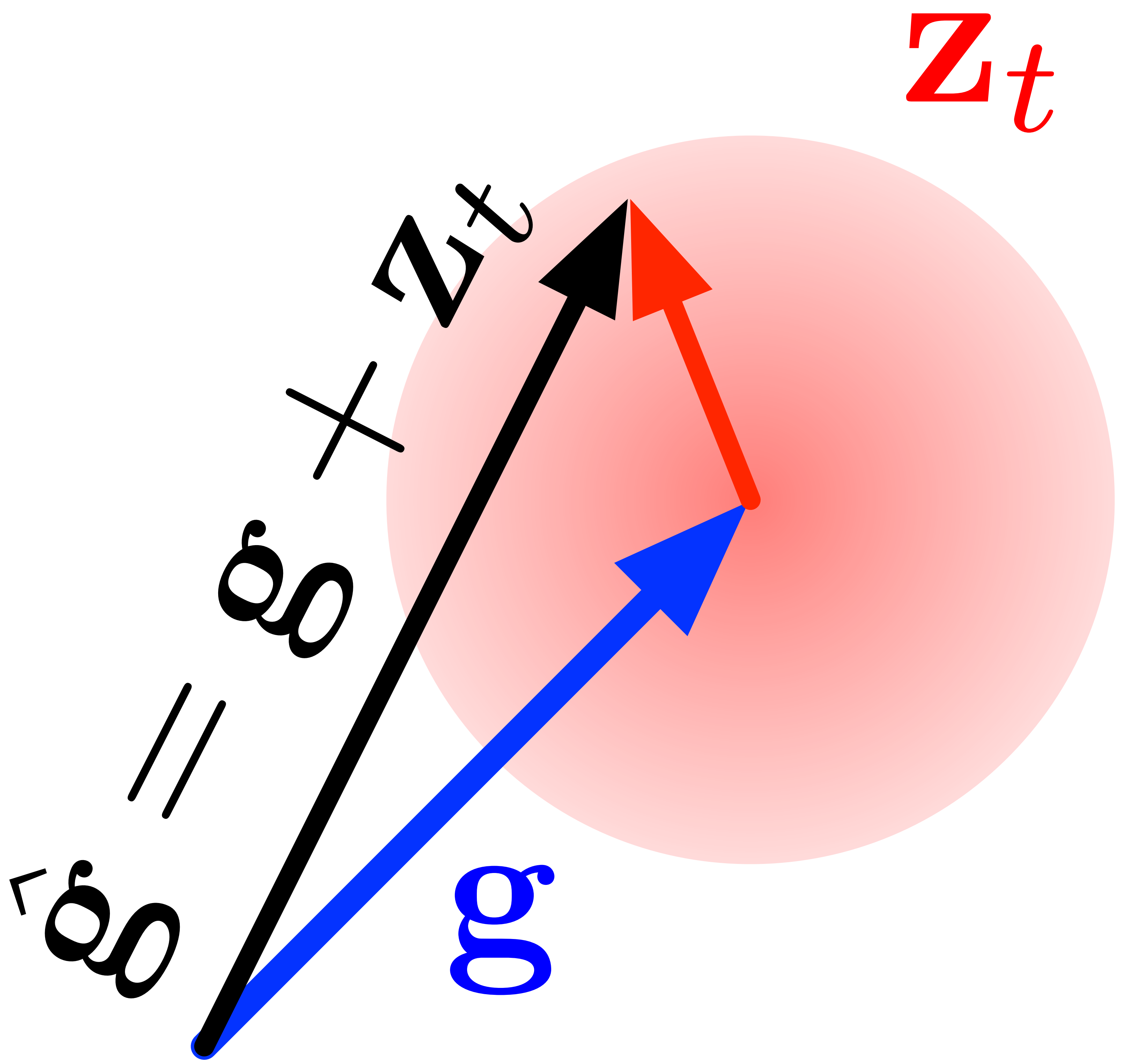


Deep neural networks (DNNs) also use optimization algorithms in training. To make these private we can add noise to the gradients in stochastic gradient descent (SGD):

- Adding noise to gradients provides differential privacy.
- For high-dimensional problems, Gaussian noise is very effective.
- Need to use **privacy accounting**.

Deep Learning and DP

Privacy for neural networks

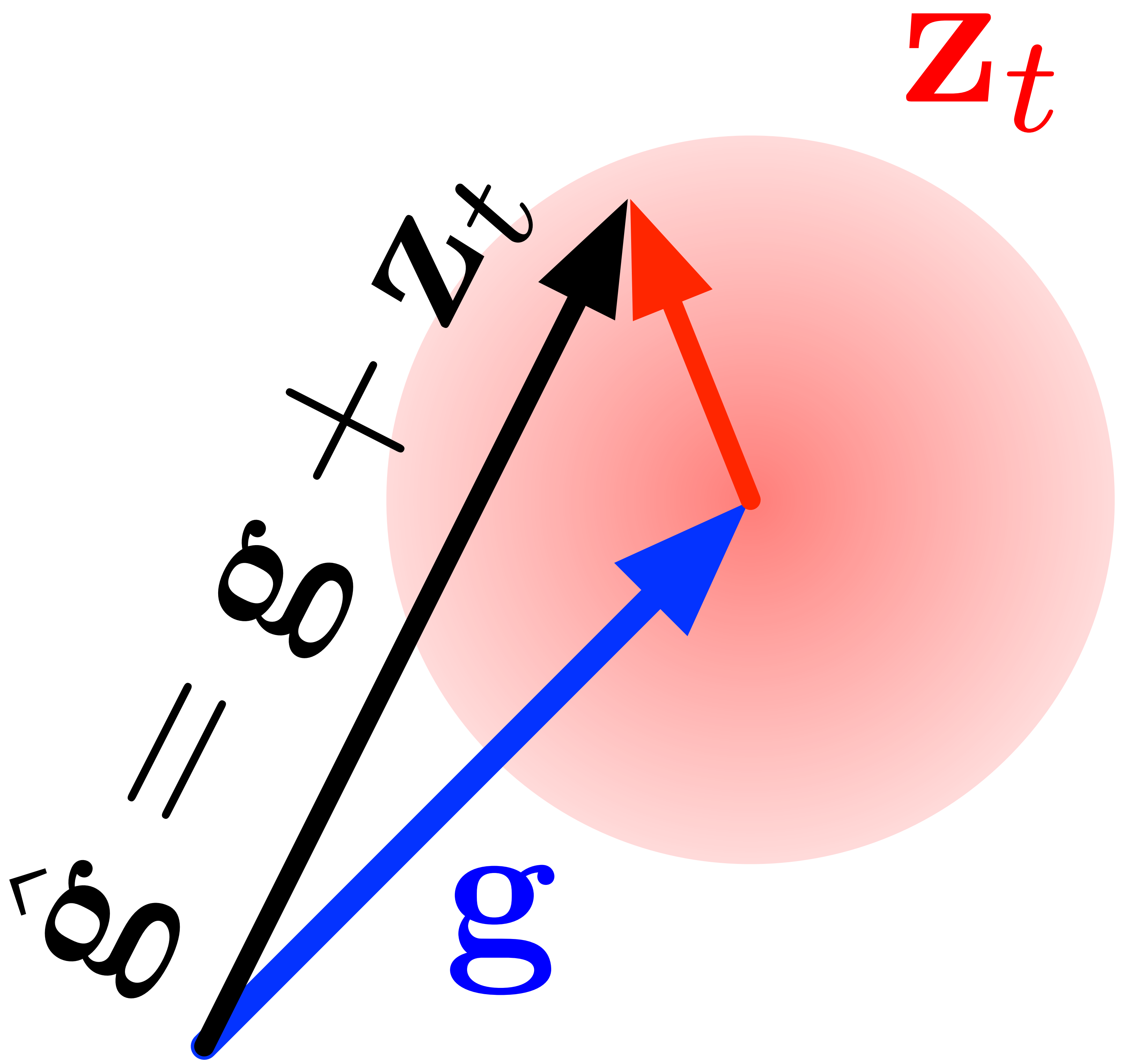


Deep neural networks (DNNs) also use optimization algorithms in training. To make these private we can add noise to the gradients in stochastic gradient descent (SGD):

- Adding noise to gradients provides differential privacy.
- For high-dimensional problems, Gaussian noise is very effective.
- Need to use **privacy accounting**.

Deep Learning and DP

Privacy for neural networks



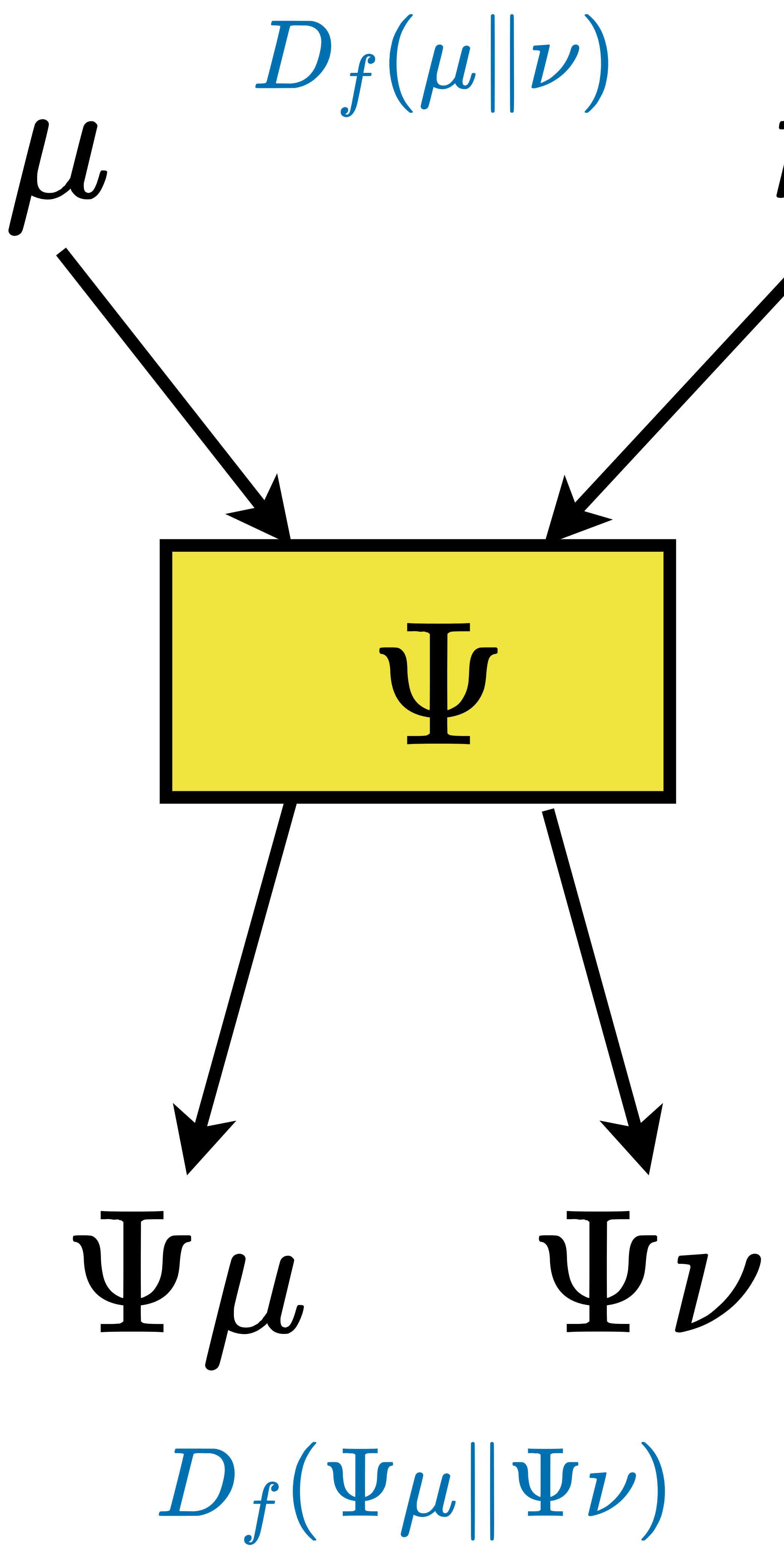
Deep neural networks (DNNs) also use optimization algorithms in training. To make these private we can add noise to the gradients in stochastic gradient descent (SGD):

- Adding noise to gradients provides differential privacy.
- For high-dimensional problems, Gaussian noise is very effective.
- Need to use **privacy accounting**.

[Song et.al. 2013, Duchi et.al. 2014, Abadi et.al. 2016, Mironov 2017]

Strong data processing inequalities

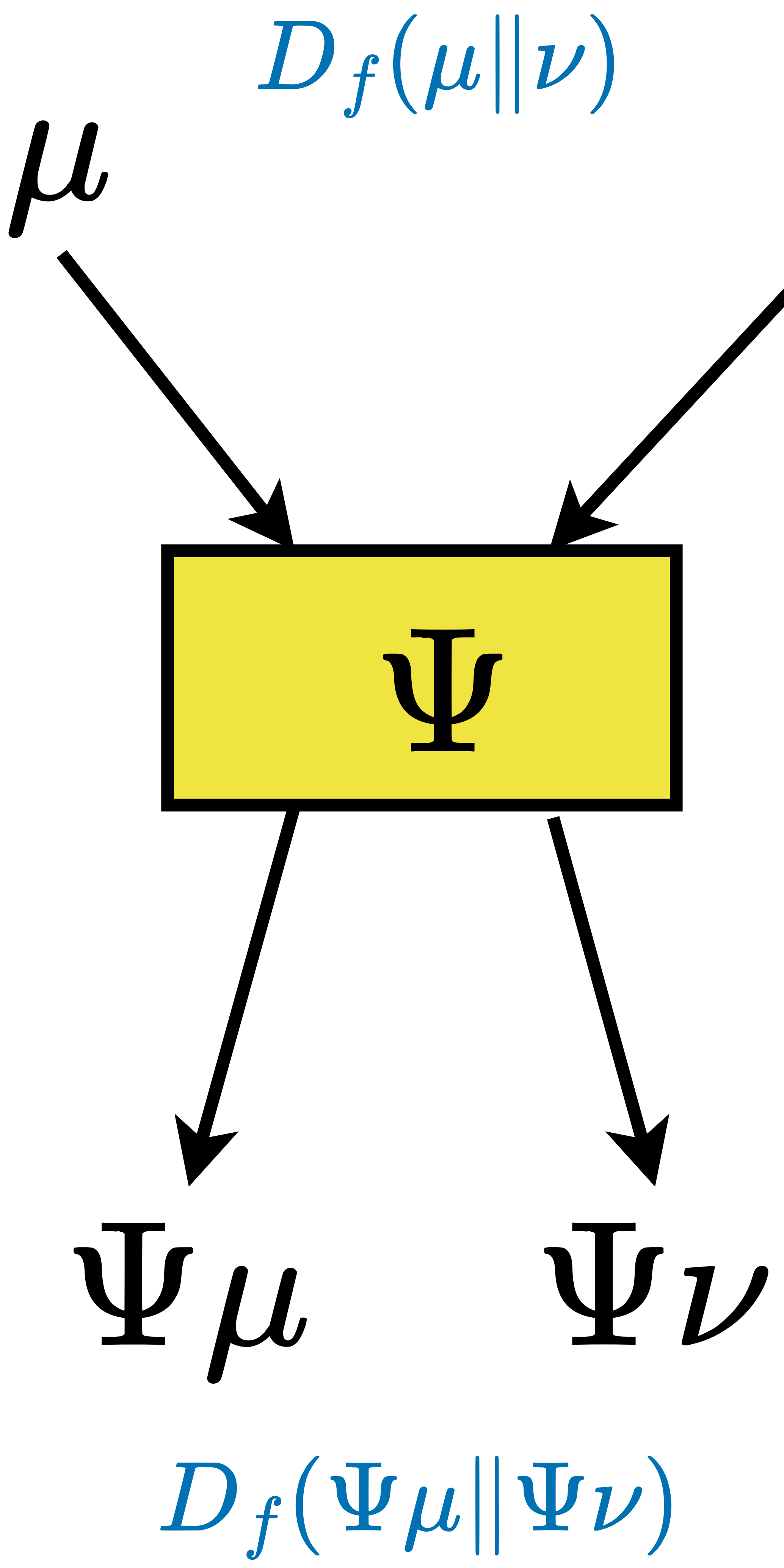
Quantifying the privacy gain from post-processing



Dobrushin (1956), Ahlswede, Gács (1976)

Strong data processing inequalities

Quantifying the privacy gain from post-processing

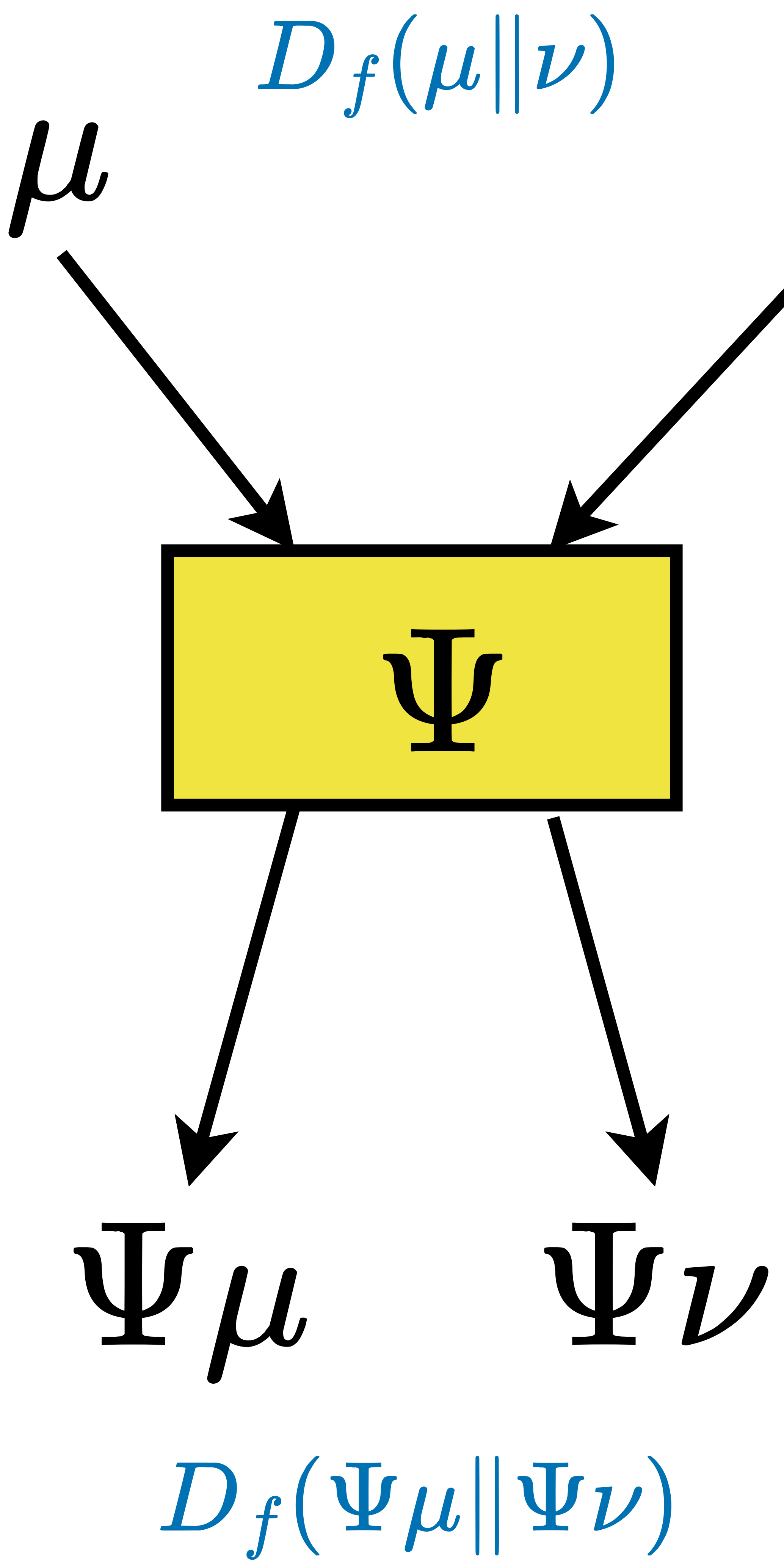


The (Dobrushin) **contraction coefficient** of a channel Ψ for a divergence D_f is

Dobrushin (1956), Ahlswede, Gács (1976)

Strong data processing inequalities

Quantifying the privacy gain from post-processing



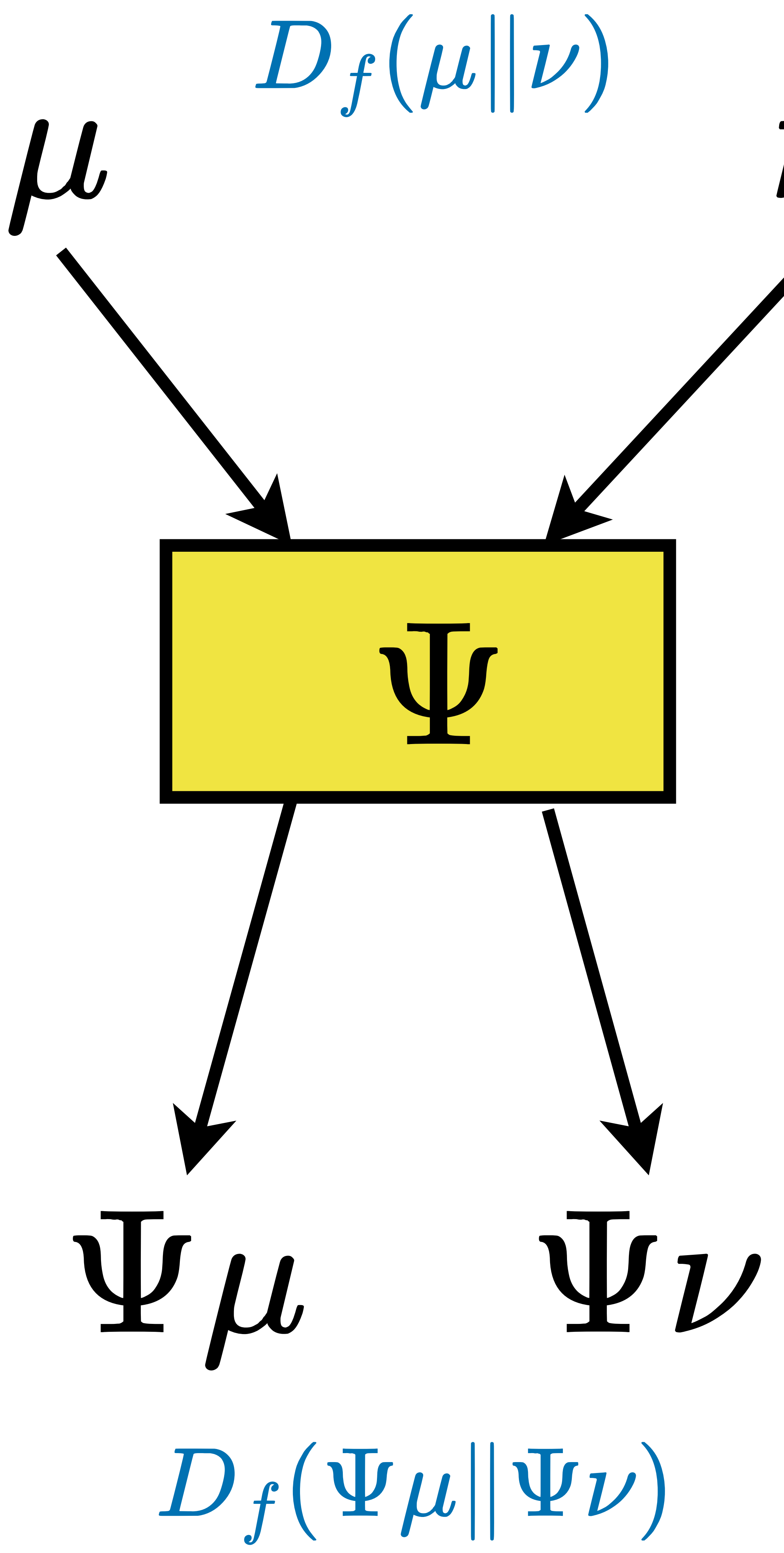
The (Dobrushin) **contraction coefficient** of a channel Ψ for a divergence D_f is

$$\eta_f(\Psi) = \sup_{\mu,\nu : D_f(\mu\|\nu)\neq 0} \frac{D_f(\Psi\mu\|\Psi\nu)}{D_f(\mu\|\nu)}.$$

Dobrushin (1956), Ahlswede, Gács (1976)

Strong data processing inequalities

Quantifying the privacy gain from post-processing



The (Dobrushin) **contraction coefficient** of a channel Ψ for a divergence D_f is

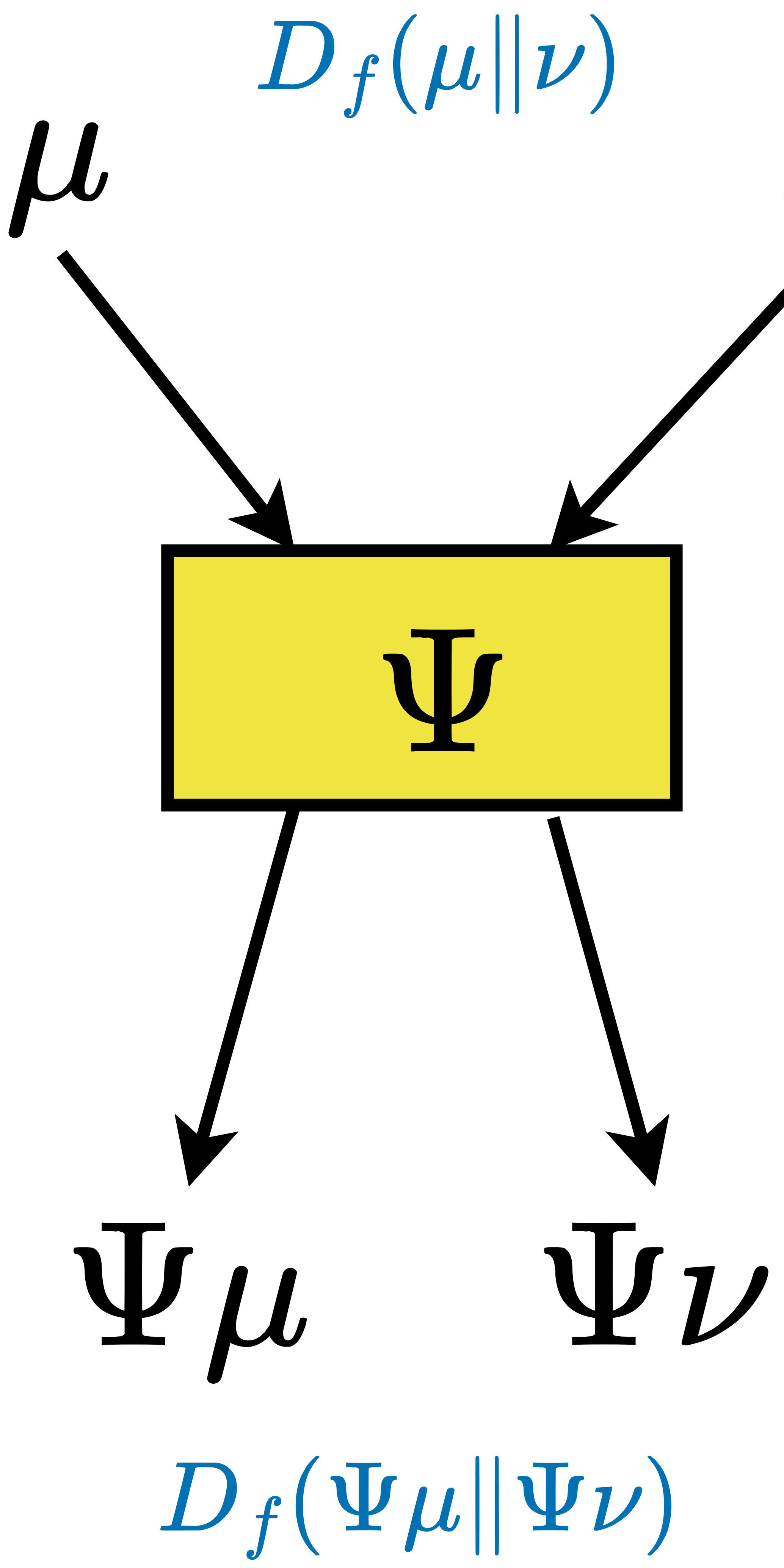
$$\eta_f(\Psi) = \sup_{\mu,\nu : D_f(\mu\|\nu)\neq 0} \frac{D_f(\Psi\mu\|\Psi\nu)}{D_f(\mu\|\nu)}.$$

This quantifies the guaranteed gap (if it exists) in the **data processing inequality (DPI)**:

Dobrushin (1956), Ahlswede, Gács (1976)

Strong data processing inequalities

Quantifying the privacy gain from post-processing



The (Dobrushin) **contraction coefficient** of a channel Ψ for a divergence D_f is

$$\eta_f(\Psi) = \sup_{\mu,\nu : D_f(\mu\|\nu)\neq 0} \frac{D_f(\Psi\mu\|\Psi\nu)}{D_f(\mu\|\nu)}.$$

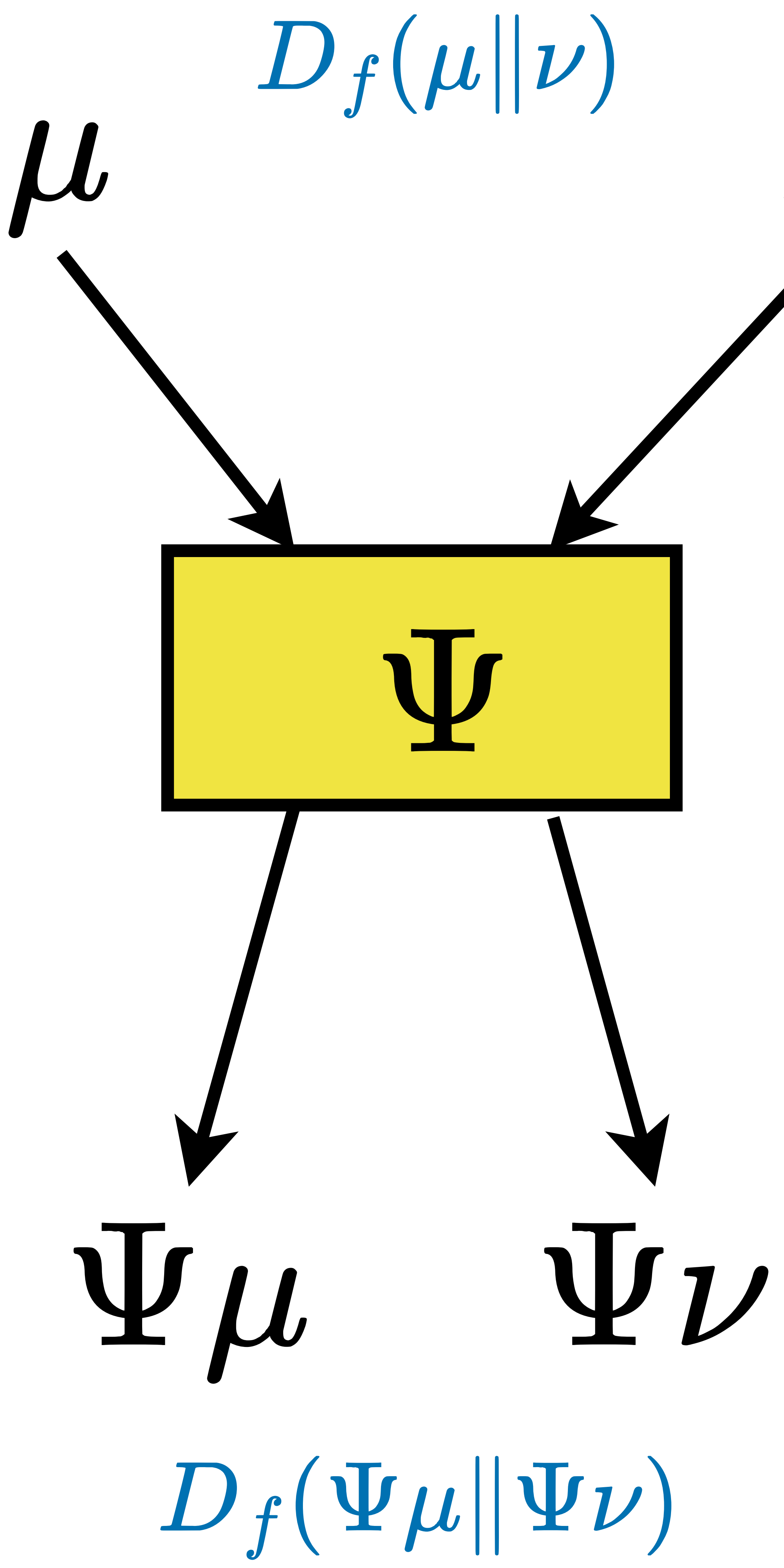
This quantifies the guaranteed gap (if it exists) in the **data processing inequality (DPI)**:

$$D_f(\Psi\mu\|\Psi\nu) \leq D_f(\mu\|\nu).$$

Dobrushin (1956), Ahlswede, Gács (1976)

Strong data processing inequalities

Quantifying the privacy gain from post-processing



The (Dobrushin) **contraction coefficient** of a channel Ψ for a divergence D_f is

$$\eta_f(\Psi) = \sup_{\mu,\nu : D_f(\mu\|\nu)\neq 0} \frac{D_f(\Psi\mu\|\Psi\nu)}{D_f(\mu\|\nu)}.$$

This quantifies the guaranteed gap (if it exists) in the **data processing inequality (DPI)**:

$$D_f(\Psi\mu\|\Psi\nu) \leq D_f(\mu\|\nu).$$

If $\eta_f(\Psi) > 0$ this is a **strong data processing inquality (SDPI)**.

Dobrushin (1956), Ahlswede, Gács (1976)

Contraction for the 🏒 divergence

Applications to DP-SGD and LDP

Dobrushin (1956), Asoodeh, Diaz, Calmon (2020), Balle, Barthe, Gaboardi, Hsu, Sato (2020)

Contraction for the 🏒 divergence

Applications to DP-SGD and LDP

The **contraction coefficient** for the E_γ divergence admits a 2-point characterization:

Dobrushin (1956), Asoodeh, Diaz, Calmon (2020), Balle, Barthe, Gaboardi, Hsu, Sato (2020)

Contraction for the 🏒 divergence

Applications to DP-SGD and LDP

The **contraction coefficient** for the E_γ divergence admits a 2-point characterization:

$$\eta_f(\Psi) = \sup_{w,w'} \mathsf{E}_\gamma(\Psi(w) \| \Psi(w')).$$

Dobrushin (1956), Asoodeh, Diaz, Calmon (2020), Balle, Barthe, Gaboardi, Hsu, Sato (2020)

Contraction for the 🏒 divergence

Applications to DP-SGD and LDP

The **contraction coefficient** for the E_γ divergence admits a 2-point characterization:

$$\eta_f(\Psi) = \sup_{w,w'} \mathsf{E}_\gamma(\Psi(w) \| \Psi(w')).$$

This is very similar to Dobrushin's characterization for total variation:

Dobrushin (1956), Asoodeh, Diaz, Calmon (2020), Balle, Barthe, Gaboardi, Hsu, Sato (2020)

Contraction for the 🏒 divergence

Applications to DP-SGD and LDP

The **contraction coefficient** for the E_γ divergence admits a 2-point characterization:

$$\eta_f(\Psi) = \sup_{w,w'} \mathsf{E}_\gamma(\Psi(w) \| \Psi(w')).$$

This is very similar to Dobrushin's characterization for total variation:

$$\eta_{\mathsf{TV}}(\Psi) = \sup_{w,w'} \mathsf{TV}(\Psi(w), \Psi(w')).$$

Dobrushin (1956), Asoodeh, Diaz, Calmon (2020), Balle, Barthe, Gaboardi, Hsu, Sato (2020)

Application to noisy SGD

Iterations just tack on more Markov kernels

Asoodeh, Diaz, Calmon (2020/2023), Asoodeh, Diaz (2024)

Application to noisy SGD

Iterations just tack on more Markov kernels

In noisy SGD with iterates $\{W_t\}$ compute clipped gradient updates $g_t(W_{t-1})$:

Asoodeh, Diaz, Calmon (2020/2023), Asoodeh, Diaz (2024)

Application to noisy SGD

Iterations just tack on more Markov kernels

In noisy SGD with iterates $\{W_t\}$ compute clipped gradient updates $g_t(W_{t-1})$:

$$W_t \leftarrow \Pi_{\mathscr{W}}(g_t(W_{t-1}) + \sigma Z_t)$$

Asoodeh, Diaz, Calmon (2020/2023), Asoodeh, Diaz (2024)

Application to noisy SGD

Iterations just tack on more Markov kernels

In noisy SGD with iterates $\{W_t\}$ compute clipped gradient updates $g_t(W_{t-1})$:

$$W_t \leftarrow \Pi_{\mathscr{W}}(g_t(W_{t-1}) + \sigma Z_t)$$

At each iteration, take μ, ν to be distributions of W_{t-1} and $\Psi_t \mu$, $\Psi_t \nu$ to be distributions on W_t.. We have two chains:

Asoodeh, Diaz, Calmon (2020/2023), Asoodeh, Diaz (2024)

Application to noisy SGD

Iterations just tack on more Markov kernels

In noisy SGD with iterates $\{W_t\}$ compute clipped gradient updates $g_t(W_{t-1})$:

$$W_t \leftarrow \Pi_{\mathscr{W}}(g_t(W_{t-1}) + \sigma Z_t)$$

At each iteration, take μ, ν to be distributions of W_{t-1} and $\Psi_t \mu$, $\Psi_t \nu$ to be distributions on W_t.. We have two chains:

$$\Psi_T \Psi_{T-1} \cdots \Psi_1 \mu_0 \qquad \Psi_T \Psi_{T-1} \cdots \Psi_1 \nu_0$$

Asoodeh, Diaz, Calmon (2020/2023), Asoodeh, Diaz (2024)

Application to noisy SGD

Iterations just tack on more Markov kernels

In noisy SGD with iterates $\{W_t\}$ compute clipped gradient updates $g_t(W_{t-1})$:

$$W_t \leftarrow \Pi_{\mathscr{W}}(g_t(W_{t-1}) + \sigma Z_t)$$

At each iteration, take μ, ν to be distributions of W_{t-1} and $\Psi_t\mu, \Psi_t\nu$ to be distributions on W_t.. We have two chains:

$$\Psi_T\Psi_{T-1}\cdots\Psi_1\mu_0 \qquad \Psi_T\Psi_{T-1}\cdots\Psi_1\nu_0$$

Can analyze the privacy for the last iterate by understanding contraction for the E_γ divergence. Even better: can extend to some non convex problems by merging SDPIs with coupling arguments.

Asoodeh, Diaz, Calmon (2020/2023), Asoodeh, Diaz (2024)

Privacy amplification by iteration

An abbreviated timeline

Privacy amplification by iteration

An abbreviated timeline

If you hide the iterates, the privacy leakage converges (instead of increasing with the number of iterations.

Privacy amplification by iteration

An abbreviated timeline

If you hide the iterates, the privacy leakage converges (instead of increasing with the number of iterations.

- Feldman, Mironov, Talwar, Thakurta (2018) - convex, smooth

Privacy amplification by iteration

An abbreviated timeline

If you hide the iterates, the privacy leakage converges (instead of increasing with the number of iterations.

- Feldman, Mironov, Talwar, Thakurta (2018) - convex, smooth
- Balle, Barthe, Gaboardi, Geumlek (2019) - strongly convex

Privacy amplification by iteration

An abbreviated timeline

If you hide the iterates, the privacy leakage converges (instead of increasing with the number of iterations.

- Feldman, Mironov, Talwar, Thakurta (2018) - convex, smooth
- Balle, Barthe, Gaboardi, Geumlek (2019) - strongly convex
- Chourasia, Ye, Shokri (2021/2022) Ryffel, Bach, Pointcheval (2022) - strongly convex, smooth for minibatch noisy SGD

Privacy amplification by iteration

An abbreviated timeline

If you hide the iterates, the privacy leakage converges (instead of increasing with the number of iterations.

- Feldman, Mironov, Talwar, Thakurta (2018) - convex, smooth
- Balle, Barthe, Gaboardi, Geumlek (2019) - strongly convex
- Chourasia, Ye, Shokri (2021/2022) Ryffel, Bach, Pointcheval (2022) - strongly convex, smooth for minibatch noisy SGD
- Altschuler, Talwar (2022)/Altschuler, Bok, Talwar (2024) - projected, convex, bounded

Privacy amplification by iteration

An abbreviated timeline

If you hide the iterates, the privacy leakage converges (instead of increasing with the number of iterations.

- Feldman, Mironov, Talwar, Thakurta (2018) - convex, smooth
- Balle, Barthe, Gaboardi, Geumlek (2019) - strongly convex
- Chourasia, Ye, Shokri (2021/2022) Ryffel, Bach, Pointcheval (2022) - strongly convex, smooth for minibatch noisy SGD
- Altschuler, Talwar (2022)/Altschuler, Bok, Talwar (2024) - projected, convex, bounded
- Asoodeh, Diaz (2024) - use data processing inequalities to remove convexity and smoothness assumptions for projected DP-SGD and regularized DP-SGD.

Contraction and Bayesian estimation

Focusing on the local model

Asoodeh, Diaz, Calmon (2020/2023)

Contraction and Bayesian estimation

Focusing on the local model

Suppose we have X_1^n i.i.d. $\sim P_{X|\theta}$ with prior $\theta \sim P_\Theta$ and privatized version Z_1^n with $Z_i = \Psi_{\varepsilon,\delta}(X_i)$ (local DP). Then the **Bayes risk**

Asoodeh, Diaz, Calmon (2020/2023)

Contraction and Bayesian estimation

Focusing on the local model

Suppose we have X_1^n i.i.d. $\sim P_{X|\theta}$ with prior $\theta \sim P_\Theta$ and privatized version Z_1^n with $Z_i = \Psi_{\varepsilon,\delta}(X_i)$ (local DP). Then the **Bayes risk**

$$R(\Theta, \varepsilon, \delta) = \inf_{\Psi_{\epsilon,\delta}} \inf_{\hat{\theta}} \mathbb{E}[\ell(\theta, \hat{\theta}(Y_1^n)]$$

Asoodeh, Diaz, Calmon (2020/2023)

Contraction and Bayesian estimation

Focusing on the local model

Suppose we have X_1^n i.i.d. $\sim P_{X|\theta}$ with prior $\theta \sim P_\Theta$ and privatized version Z_1^n with $Z_i = \Psi_{\varepsilon,\delta}(X_i)$ (local DP). Then the **Bayes risk**

$$R(\Theta, \varepsilon, \delta) = \inf_{\Psi_{\epsilon,\delta}} \inf_{\hat{\theta}} \mathbb{E}[\ell(\theta, \hat{\theta}(Y_1^n))]$$

can be lower bounded in terms of an E_γ**-mutual information**. In the language of "quantitative information flow":

Asoodeh, Diaz, Calmon (2020/2023)

Contraction and Bayesian estimation

Focusing on the local model

Suppose we have X_1^n i.i.d. $\sim P_{X|\theta}$ with prior $\theta \sim P_\Theta$ and privatized version Z_1^n with $Z_i = \Psi_{\varepsilon,\delta}(X_i)$ (local DP). Then the **Bayes risk**

$$R(\Theta, \varepsilon, \delta) = \inf_{\Psi_{\epsilon,\delta}} \inf_{\hat{\theta}} \mathbb{E}[\ell(\theta, \hat{\theta}(Y_1^n)]$$

can be lower bounded in terms of an E_γ**-mutual information**. In the language of "quantitative information flow":

> θ is a secret, the loss ℓ is a negative gain, and we look for the maximally leaky channel subject to an (ε, δ) constraint…

Asoodeh, Diaz, Calmon (2020/2023)

Morning After a Snowfall at Koishikawa

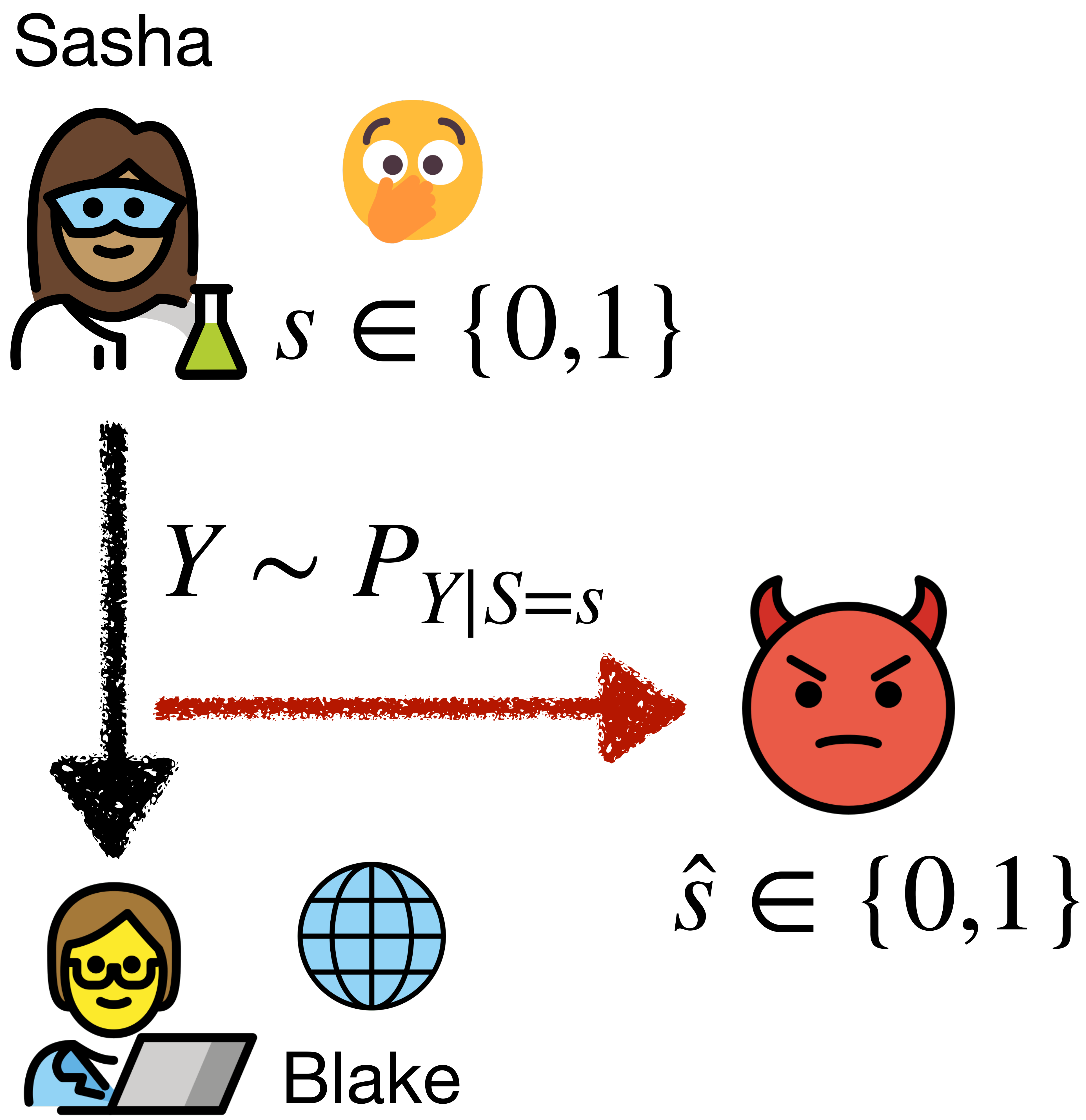
礫川雪の旦

Koishikawa yuki no ashita

other destinations

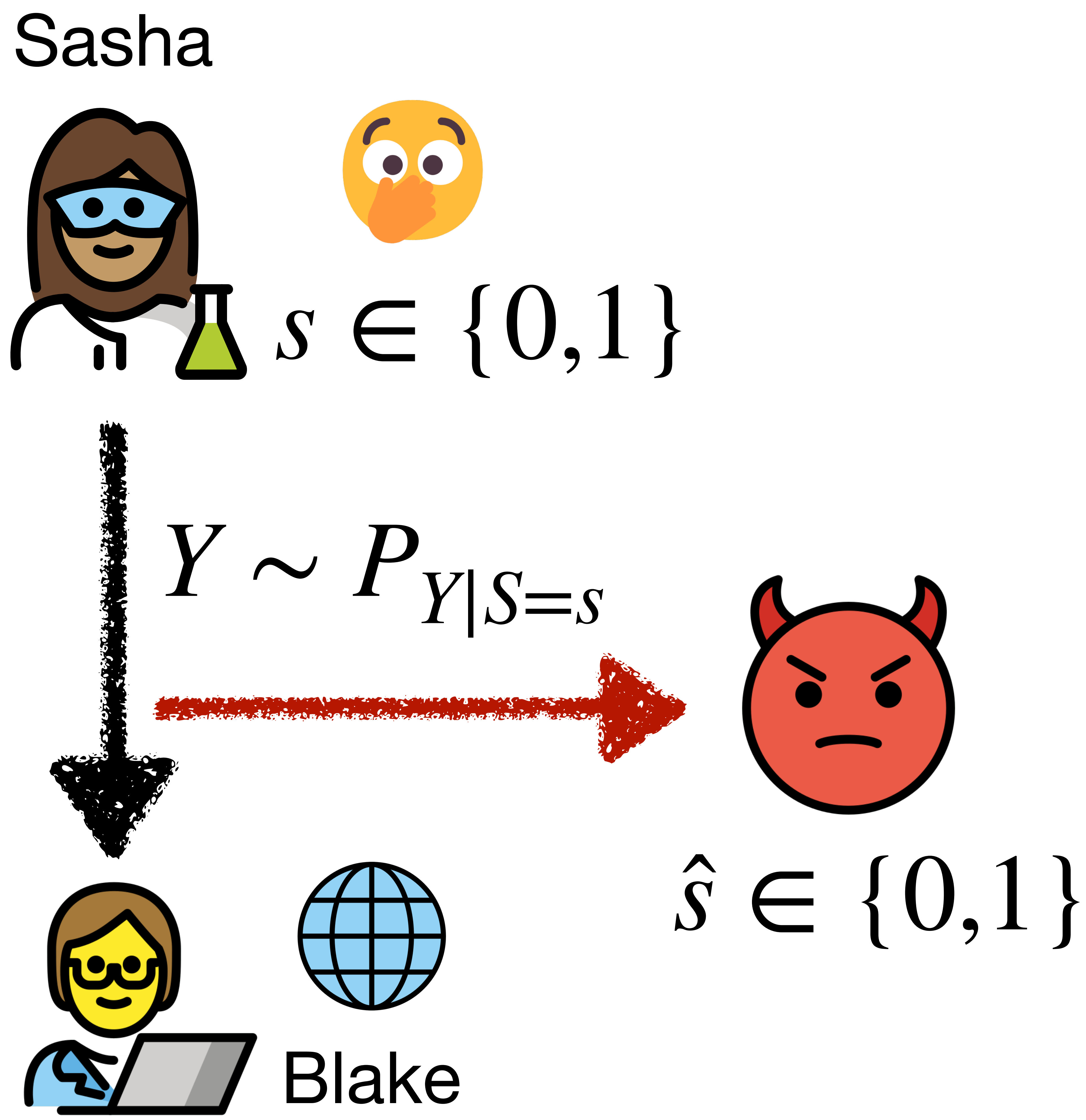
What we’ve seen so far

Let’s start simple



What we've seen so far

Let's start simple



We started out with a simple story: protecting a single bit.

- Differential privacy both is and is not just as simple as hypothesis testing.
- Taking an information-theoretic view opens the door to better analyses.
- The gap between algorithms and analysis is shrinking.
- The gap between algorithms and applications is still large.

The gap between theory and practice

It's wider than you might think

The gap between theory and practice

It's wider than you might think

There are lots of issues with implementing differential privacy in practice:

- Approximate versus exact sampling (and side channels)
- Approximate versus exact optimization
- "Privacy amplification" and it's implementation
- Numerical precision and floating points
- Managing privacy budgets


安野平兵衛
市川市紅

Several interesting challenges left for:

Several interesting challenges left for:

Several interesting challenges left for:

maths

Several interesting challenges left for:

maths

computational stats

Several interesting challenges left for:

maths

computational stats

engineering

Several interesting challenges left for:

maths

computational stats

engineering

human-computer interaction

Several interesting challenges left for:

maths
computational stats
engineering
human-computer interaction
technology policy

The Great Wave off Kanagawa

神奈川沖浪裏

Kanagawa oki nami-ura

Thank you!