Rm Palaniappan, *Alien Planet-X-9*
Viscosity, pencil colour and ink on handmade paper

Are modern ML models like scientific instruments?

Anand D. Sarwate (Rutgers University)

28 July 2025

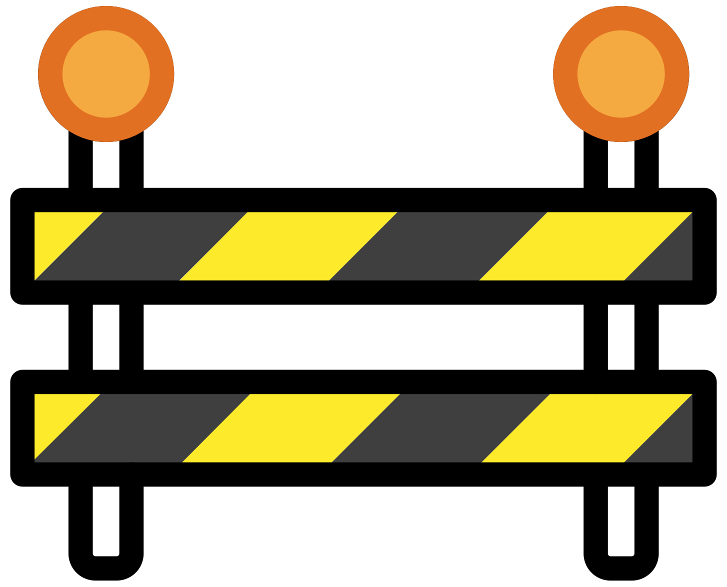
IEEE ITSOC Distinguished Lecture
2025 Guangzhou, Hong Kong and Taipei Joint Workshop
on Artificial Intelligence, Communications and Information Theory (AICIT 2025)
Sun Yat-sen University (中山大学), Guangzhou, China

Some pre-apologies

I am still trying to figure out how to talk about this work

Some pre-apologies

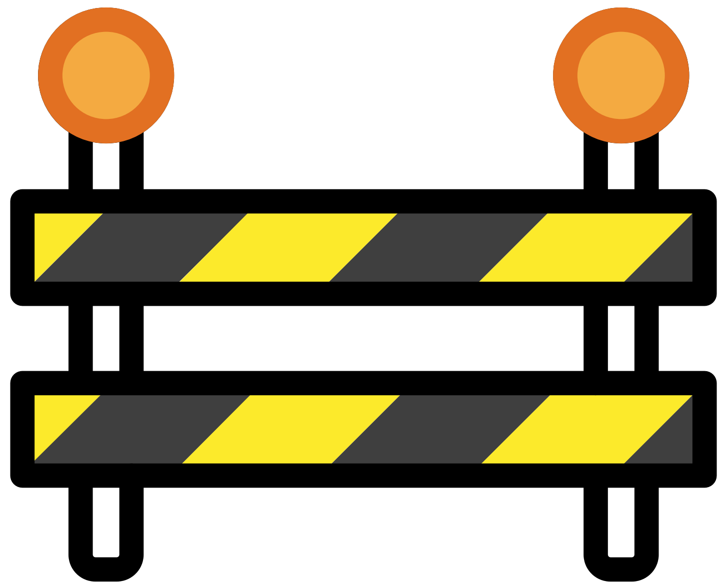
I am still trying to figure out how to talk about this work



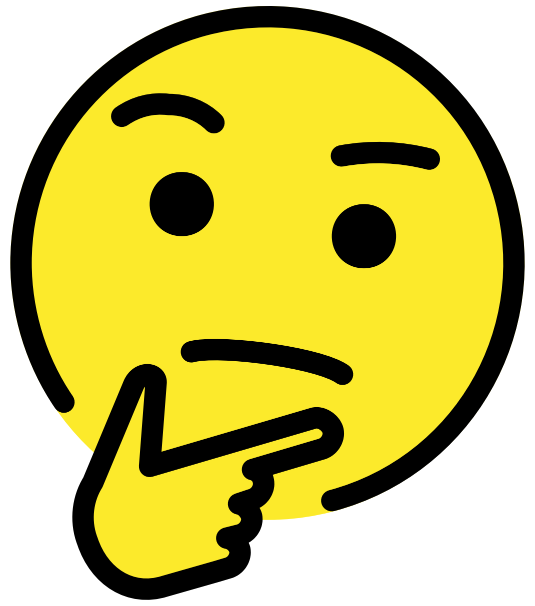
This is (mostly) based on pretty empirical work: not sure if it counts as information ***theory***.

Some pre-apologies

I am still trying to figure out how to talk about this work



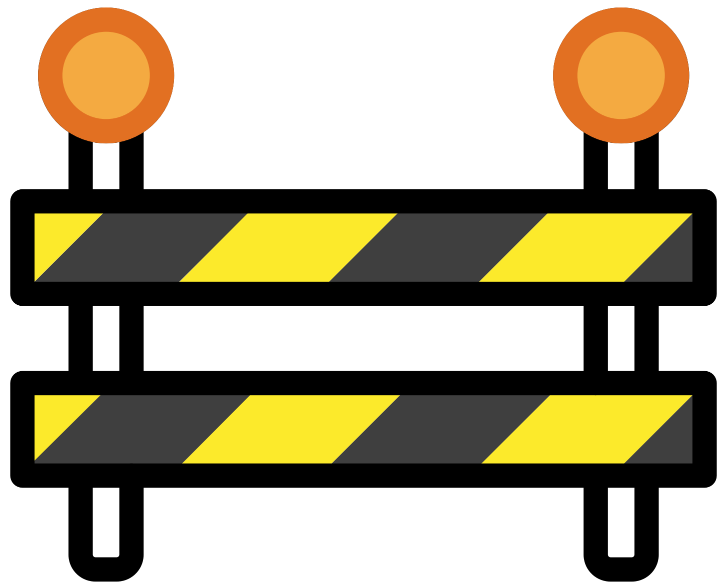
This is (mostly) based on pretty empirical work: not sure if it counts as information ***theory***.



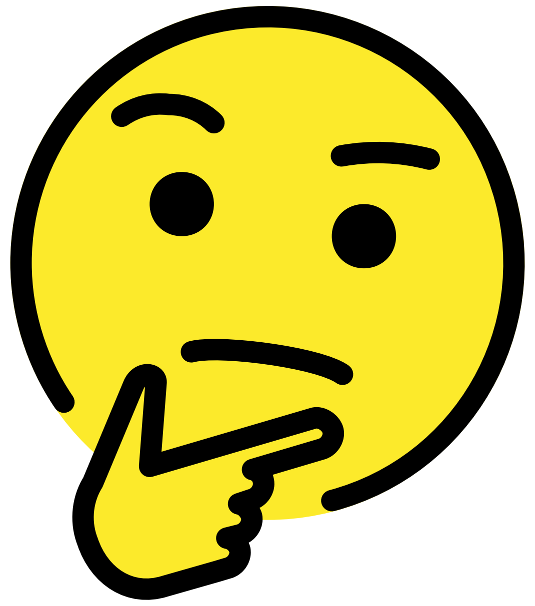
I think there are lots of **interesting questions for theory** which this brings up!

Some pre-apologies

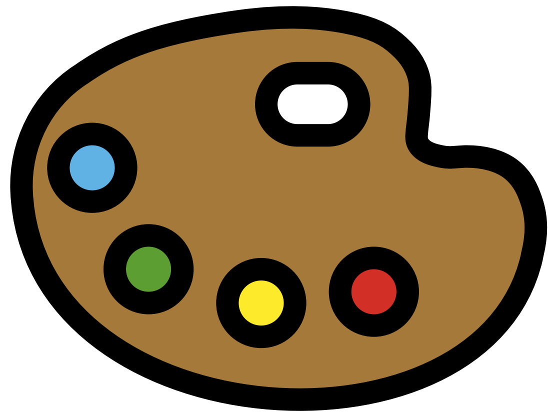
I am still trying to figure out how to talk about this work



This is (mostly) based on pretty empirical work: not sure if it counts as information ***theory***.



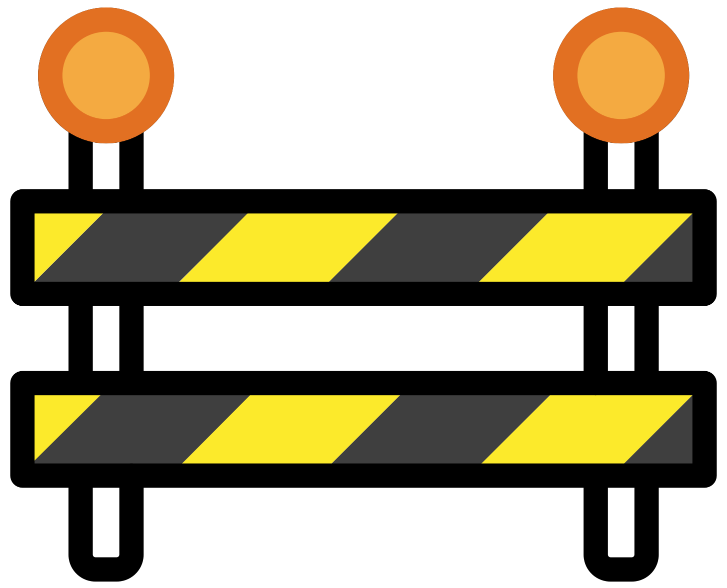
I think there are lots of **interesting questions for theory** which this brings up!



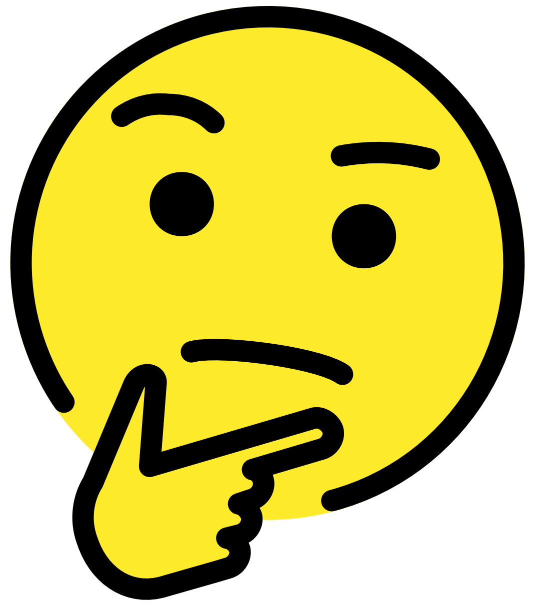
There are a lot of metaphors and analogies (some science-fictional) which are **not always precise**.

Some pre-apologies

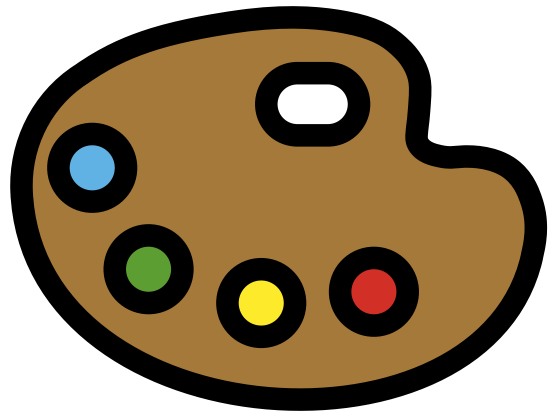
I am still trying to figure out how to talk about this work



This is (mostly) based on pretty empirical work: not sure if it counts as information ***theory***.



I think there are lots of **interesting questions for theory** which this brings up!



There are a lot of metaphors and analogies (some science-fictional) which are **not always precise**.

Real life is a bit messy…

Thanks to my collaborators/coauthors!

Most of this is their work, obviously

Sinjini Banerjee (Rutgers)	Sutenay Choudhury (PNNL)	Xin Li (Rutgers)
Reilly Cannon (PNNL)	Ioana Dumitriu (UC San Diego)	Tim Marrinan (PNNL)
Tony Chiang (ARPA-H)	Andrew Engel (Ohio State)	Max Vargas (PNNL)
Sutenay Choudhury (PNNL)		Zhichao Wang (UC Berkeley)

Papers:

[JSTSP] **Banerjee et al.** https://doi.org/10.1109/JSTSP.2025.3583140
[NeurIPS 2023] **Wang et al.** https://openreview.net/forum?id=gpqBGyKeKH
[ICLR 2024] **Engel et al.** https://openreview.net/forum?id=yKksu38BpM
[ArXiV] **Vargas et al.** https://arxiv.org/abs/2408.10437

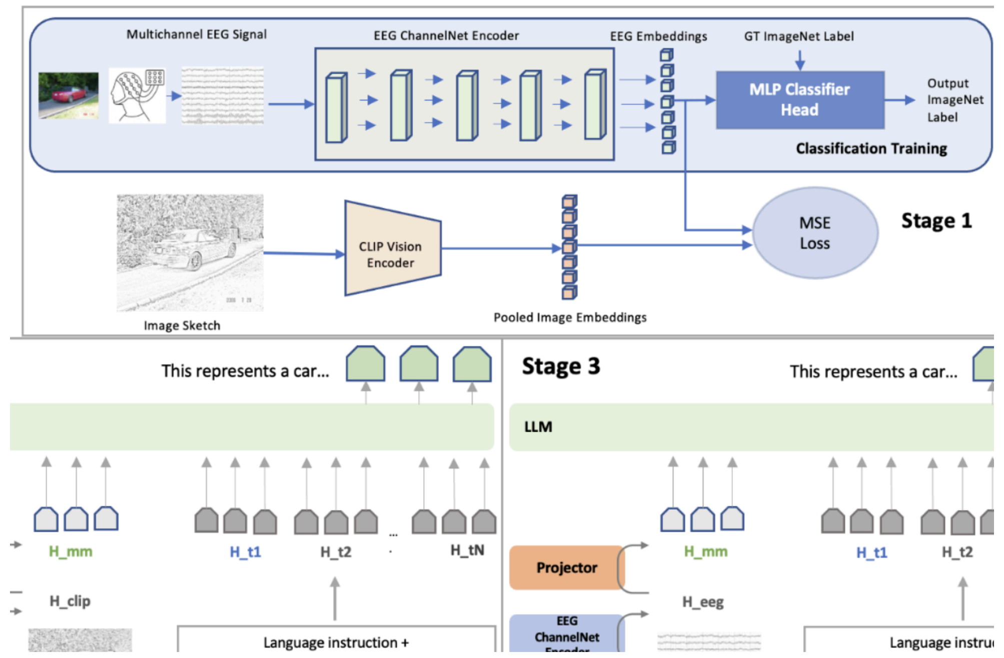
What does the title mean?

What do ML models have to do with science?

Source: Wikipedia

Source: IBM

Source: Mishra et al.



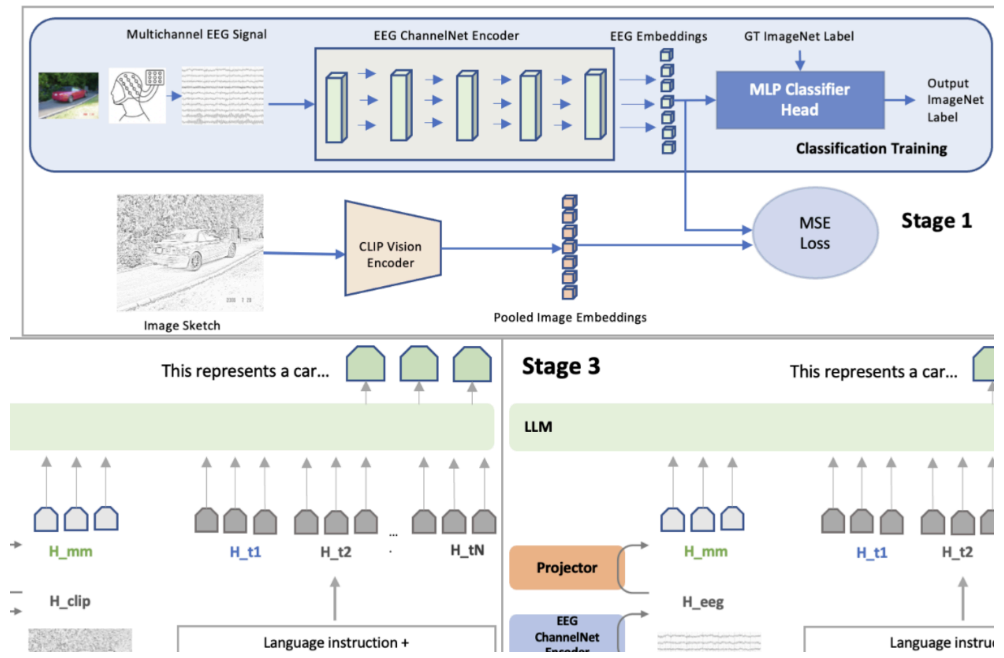
What does the title mean?

What do ML models have to do with science?

Source: Wikipedia

Source: IBM

Source: Mishra et al.



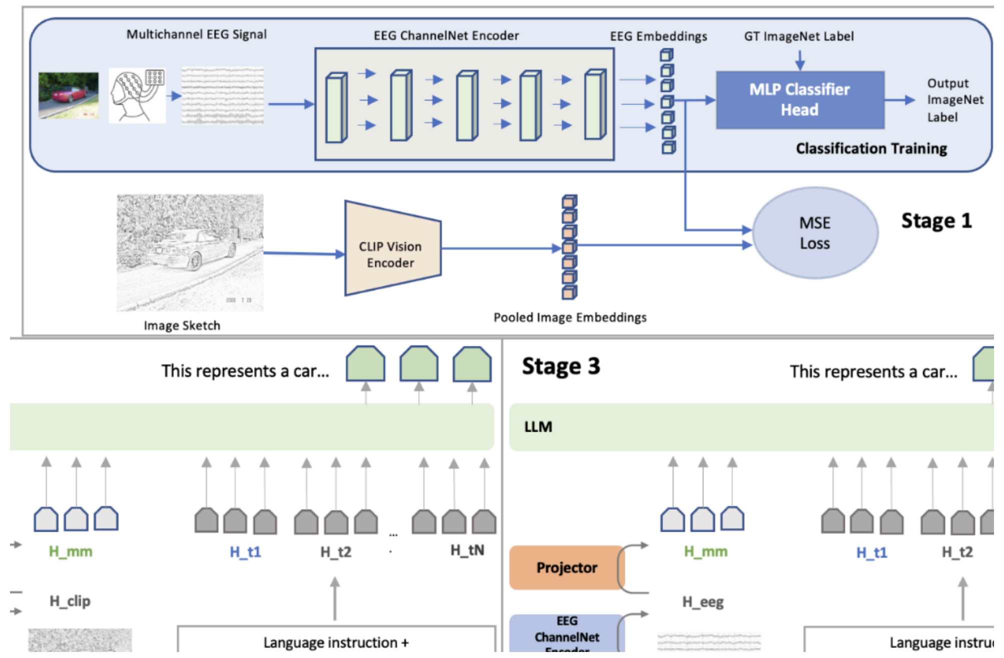
We often hear a lot about "**AI for Science**" but that can mean a lot of different things! Some examples:

What does the title mean?

What do ML models have to do with science?

Source: Wikipedia

Source: IBM



Source: Mishra et al.

We often hear a lot about "**AI for Science**" but that can mean a lot of different things! Some examples:

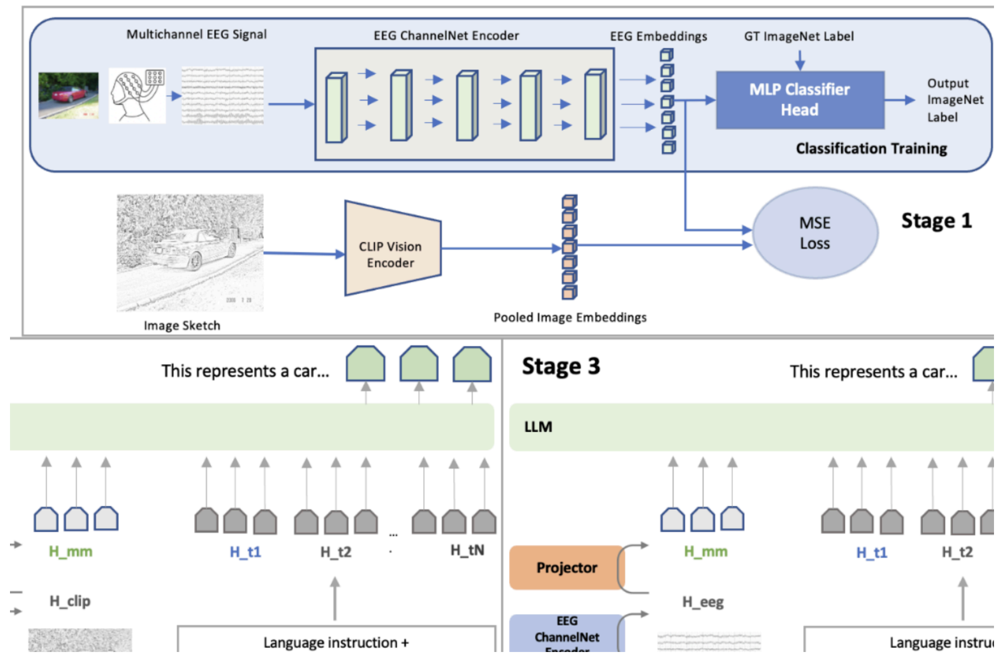
- Using computer vision to do automated analysis of medical images.

What does the title mean?

What do ML models have to do with science?

Source: Wikipedia

Source: IBM



Source: Mishra et al.

We often hear a lot about "**AI for Science**" but that can mean a lot of different things! Some examples:

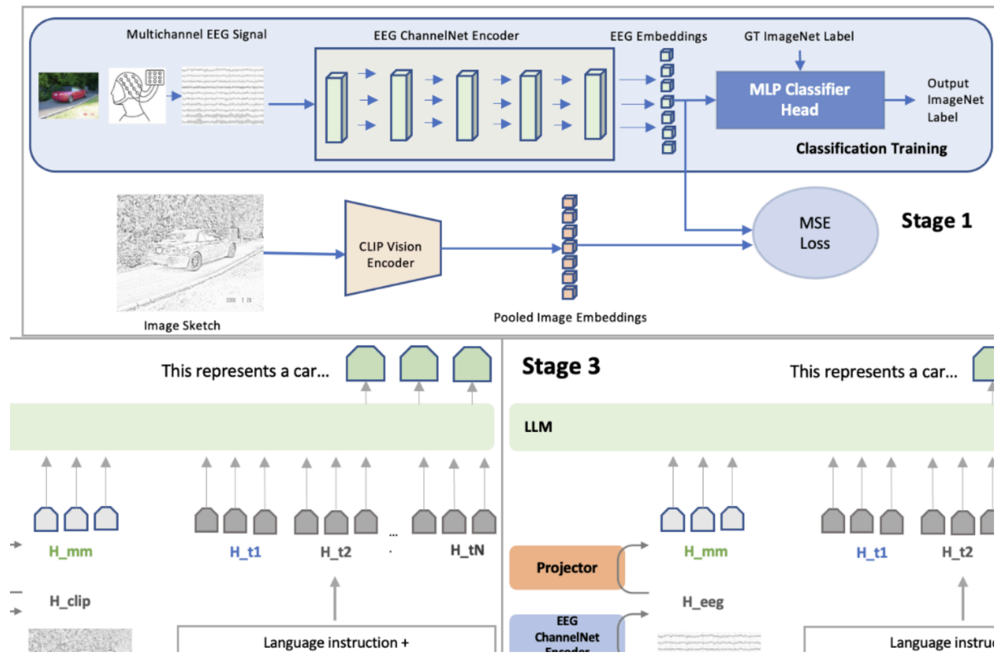
- Using computer vision to do automated analysis of medical images.
- Use generative AI to build a "digital twin" of energy/utility networks for simulation/design

What does the title mean?

What do ML models have to do with science?


Source: Wikipedia


Source: IBM



Source: Mishra et al.

We often hear a lot about "**AI for Science**" but that can mean a lot of different things! Some examples:

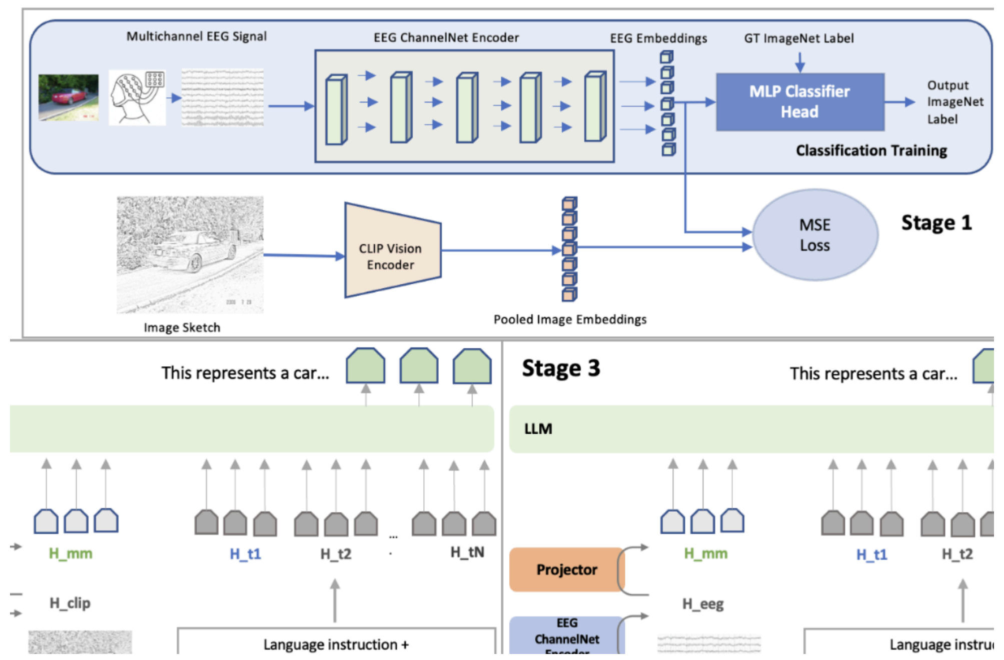
- Using computer vision to do automated analysis of medical images.
- Use generative AI to build a "digital twin" of energy/utility networks for simulation/design
- Use LLM architectures to decode brain activity for assistive technology.

What does the title mean?

What do ML models have to do with science?

Source: Wikipedia

Source: IBM



Source: Mishra et al.

We often hear a lot about "**AI for Science**" but that can mean a lot of different things! Some examples:

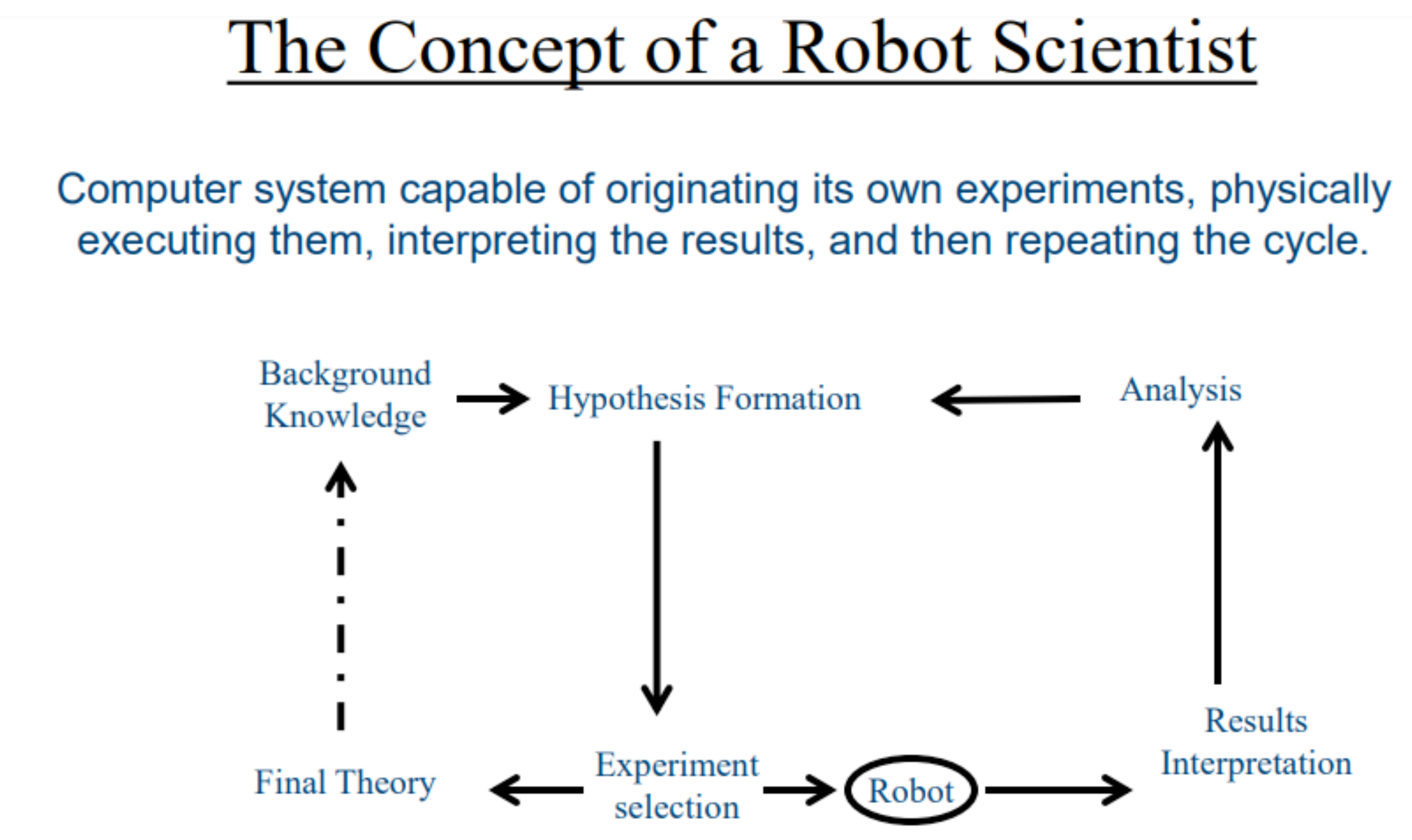
- Using computer vision to do automated analysis of medical images.
- Use generative AI to build a "digital twin" of energy/utility networks for simulation/design
- Use LLM architectures to decode brain activity for assistive technology.
- Many more…

Is “AI for science” the new “Bandwagon”?

Some gap between hype and reality

Is “AI for science” the new “Bandwagon”?

Some gap between hype and reality



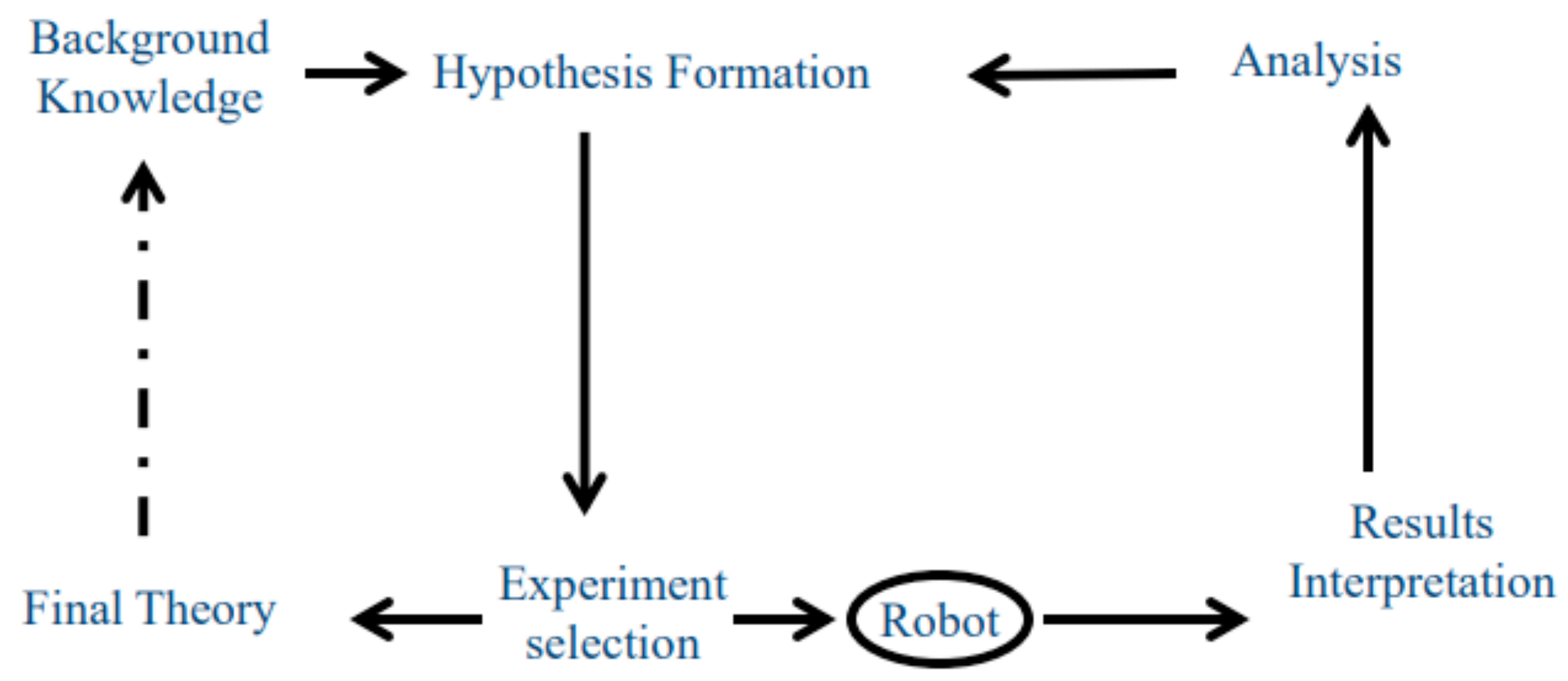
https://futuretech.mit.edu/news/ai-and-the-future-of-scientific-discovery

Is "AI for science" the new "Bandwagon"?

Some gap between hype and reality

The Concept of a Robot Scientist

Computer system capable of originating its own experiments, physically executing them, interpreting the results, and then repeating the cycle.



https://futuretech.mit.edu/news/ai-and-the-future-of-scientific-discovery

NY Times
14 May 2025

Your A.I. Radiologist Will Not Be With You Soon

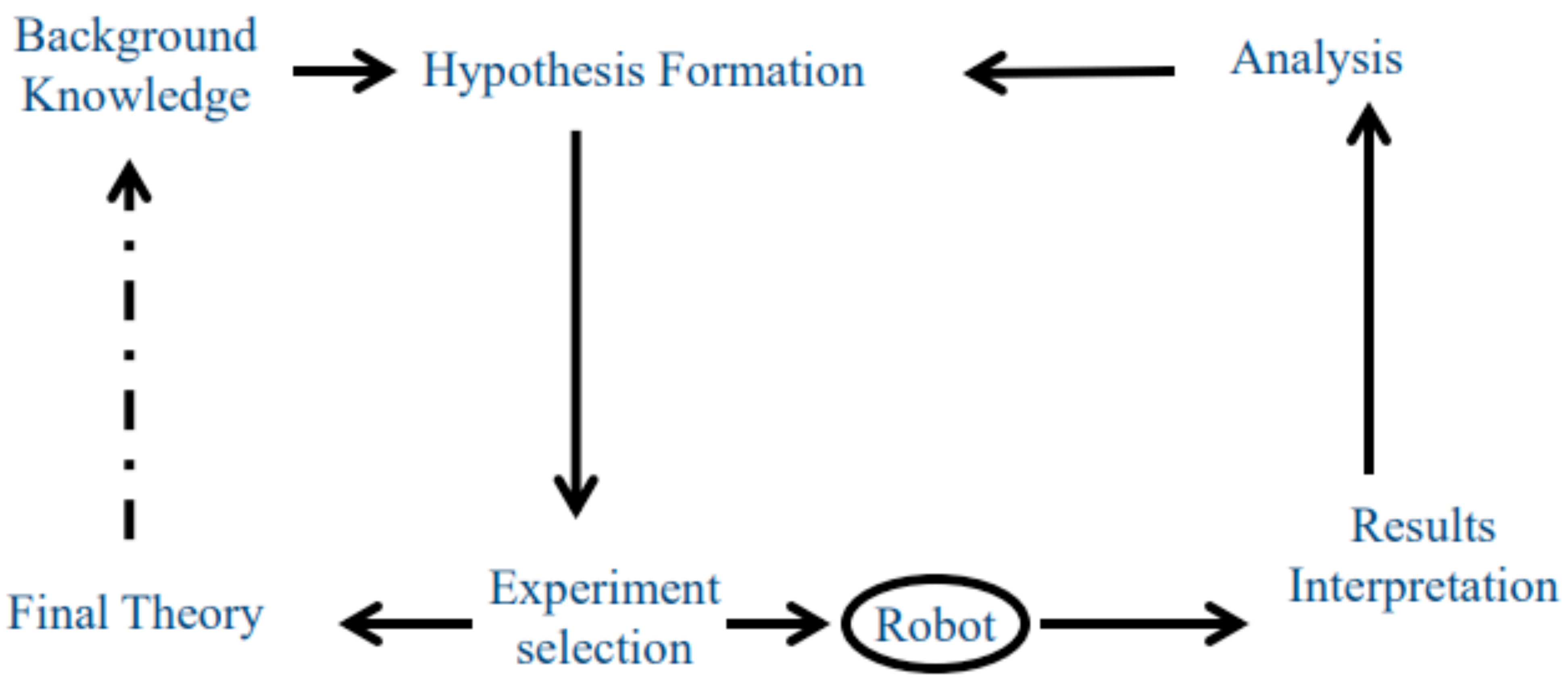
Experts predicted that artificial intelligence would steal radiology jobs. But at the Mayo Clinic, the technology has been more friend than foe.

Is "AI for science" the new "Bandwagon"?

Some gap between hype and reality

The Concept of a Robot Scientist

Computer system capable of originating its own experiments, physically executing them, interpreting the results, and then repeating the cycle.



https://futuretech.mit.edu/news/ai-and-the-future-of-scientific-discovery

NY Times
14 May 2025

Your A.I. Radiologist Will Not Be With You Soon

Experts predicted that artificial intelligence would steal radiology jobs. But at the Mayo Clinic, the technology has been more friend than foe.

IRE TRANSACTIONS—INFORMATION THEORY

The Bandwagon

CLAUDE E. SHANNON

Shannon, 1956

What about information/signal processing?

Some perspective from more solid ground

What about information/signal processing?

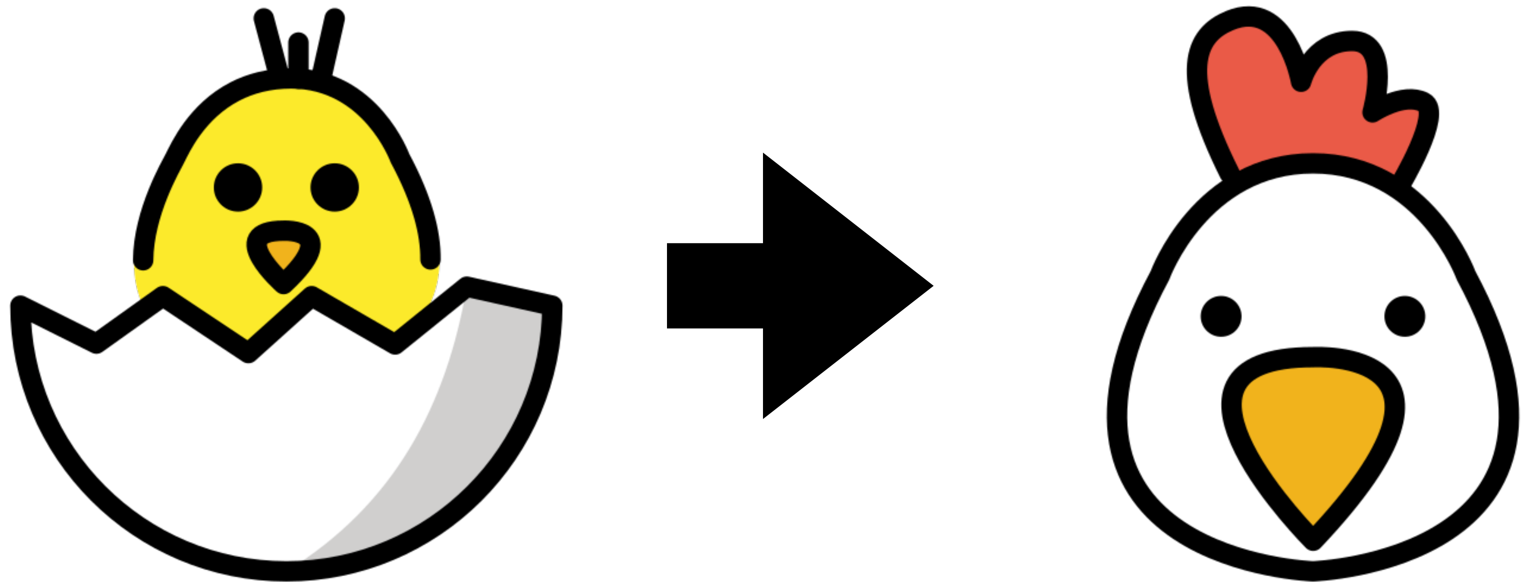
Some perspective from more solid ground

At the end of the day “artificial neural nets” are just a bunch of computational signal processing primitives chained together and jointly optimized with stochastic gradient methods.

- Ben Recht (on argmin.net)

What about information/signal processing?

Some perspective from more solid ground



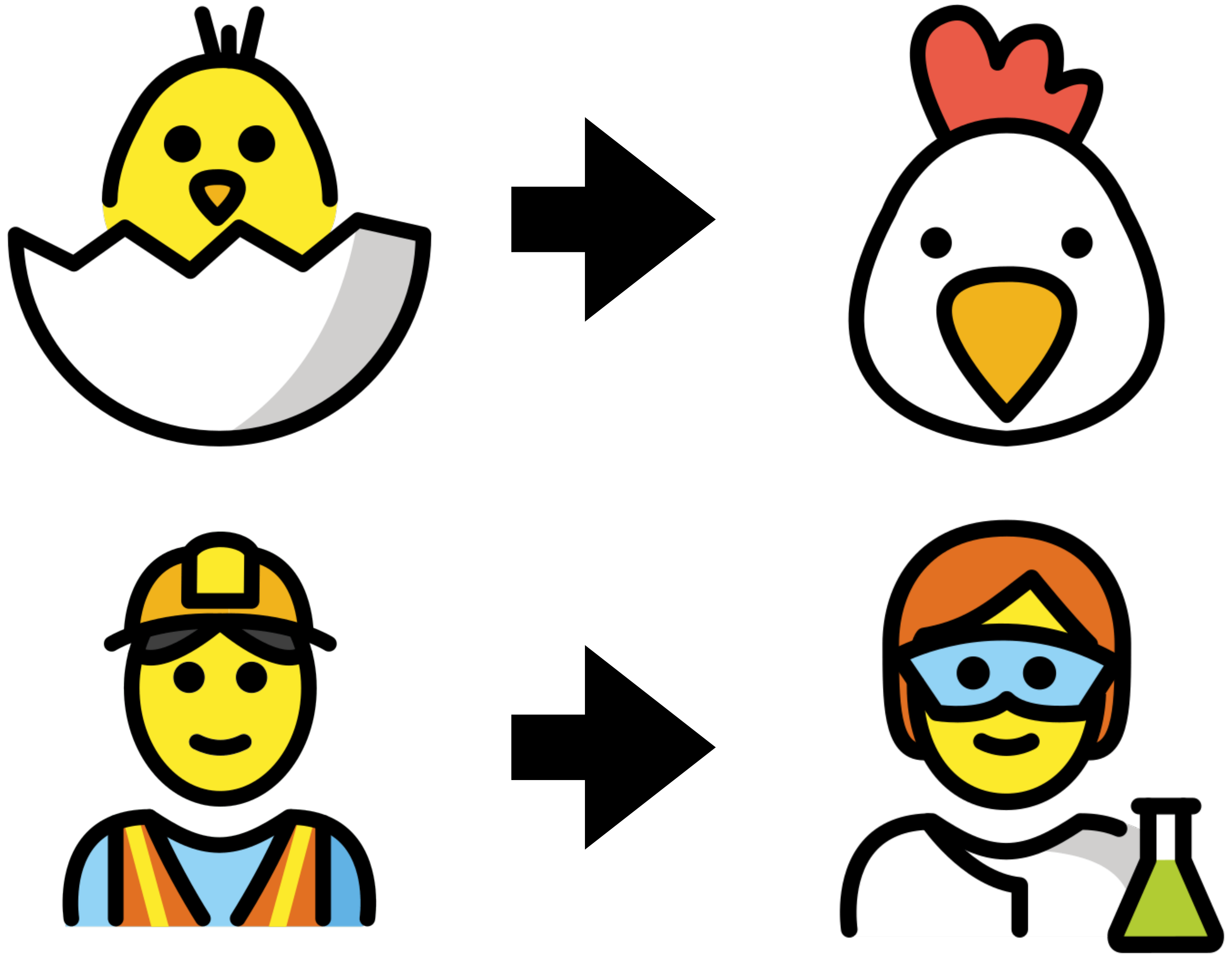
At the end of the day "artificial neural nets" are just a bunch of computational signal processing primitives chained together and jointly optimized with stochastic gradient methods.

- Ben Recht (on argmin.net)

ML/AI frameworks are evolving very quickly.

What about information/signal processing?

Some perspective from more solid ground



At the end of the day "artificial neural nets" are just a bunch of computational signal processing primitives chained together and jointly optimized with stochastic gradient methods.

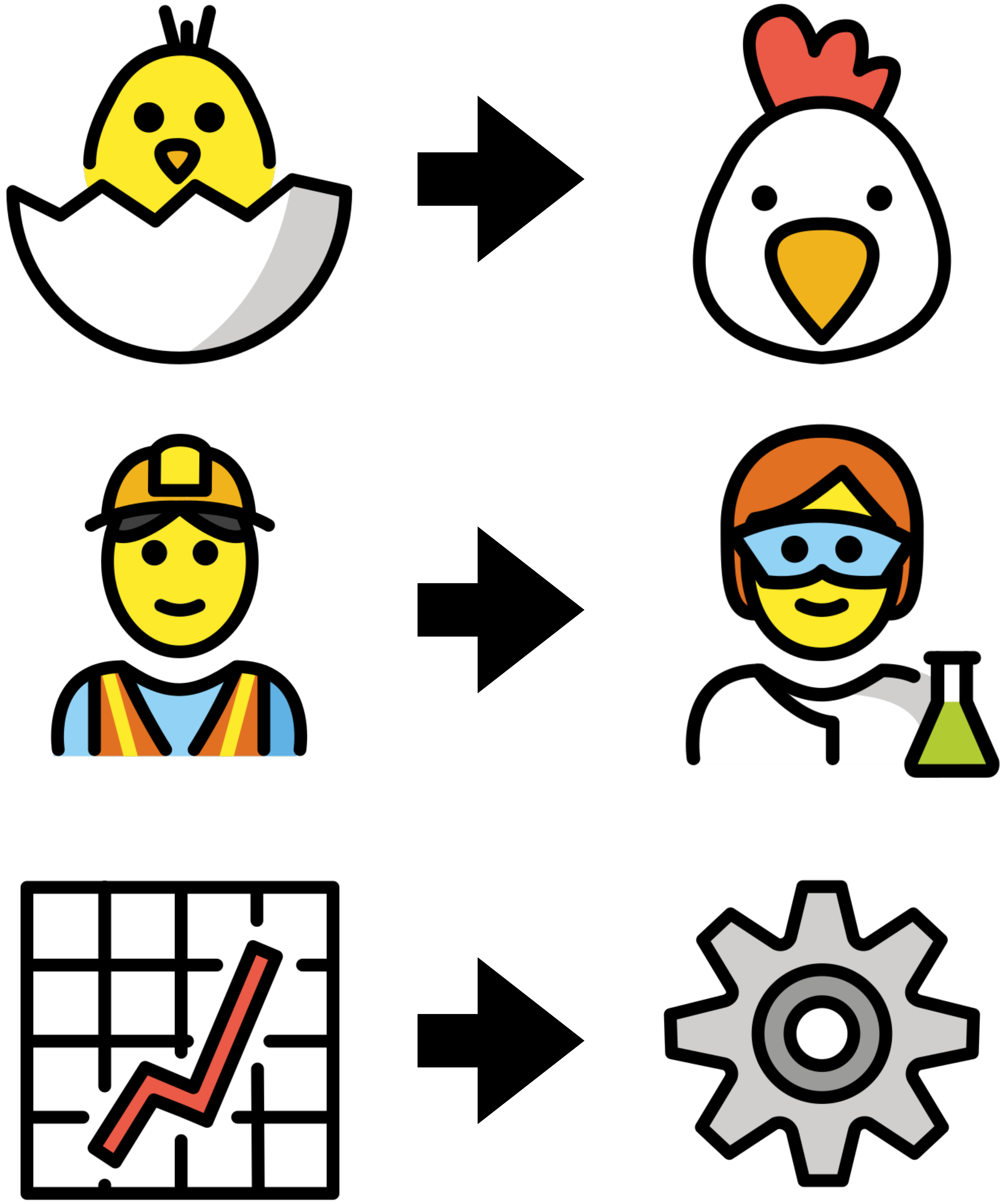
- Ben Recht (on argmin.net)

ML/AI frameworks are evolving very quickly.

➜ Theory often lags behind practice.

What about information/signal processing?

Some perspective from more solid ground



At the end of the day "artificial neural nets" are just a bunch of computational signal processing primitives chained together and jointly optimized with stochastic gradient methods.

- Ben Recht (on argmin.net)

ML/AI frameworks are evolving very quickly.

➜ Theory often lags behind practice.

➜ IT, SP, control, etc. are still relevant!

A traditional division of labor

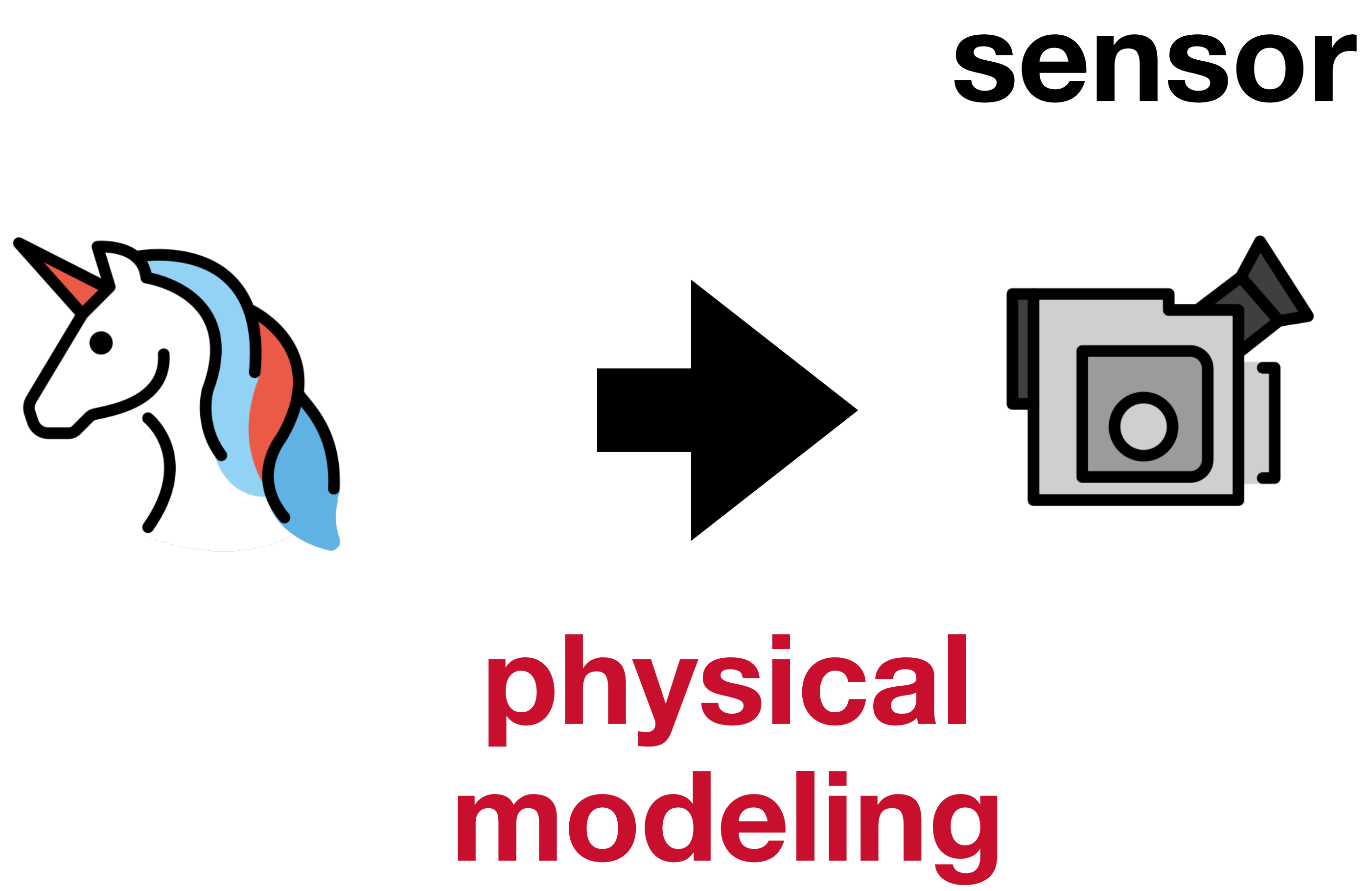
The EE/CS divide in some sense

A traditional division of labor

The EE/CS divide in some sense

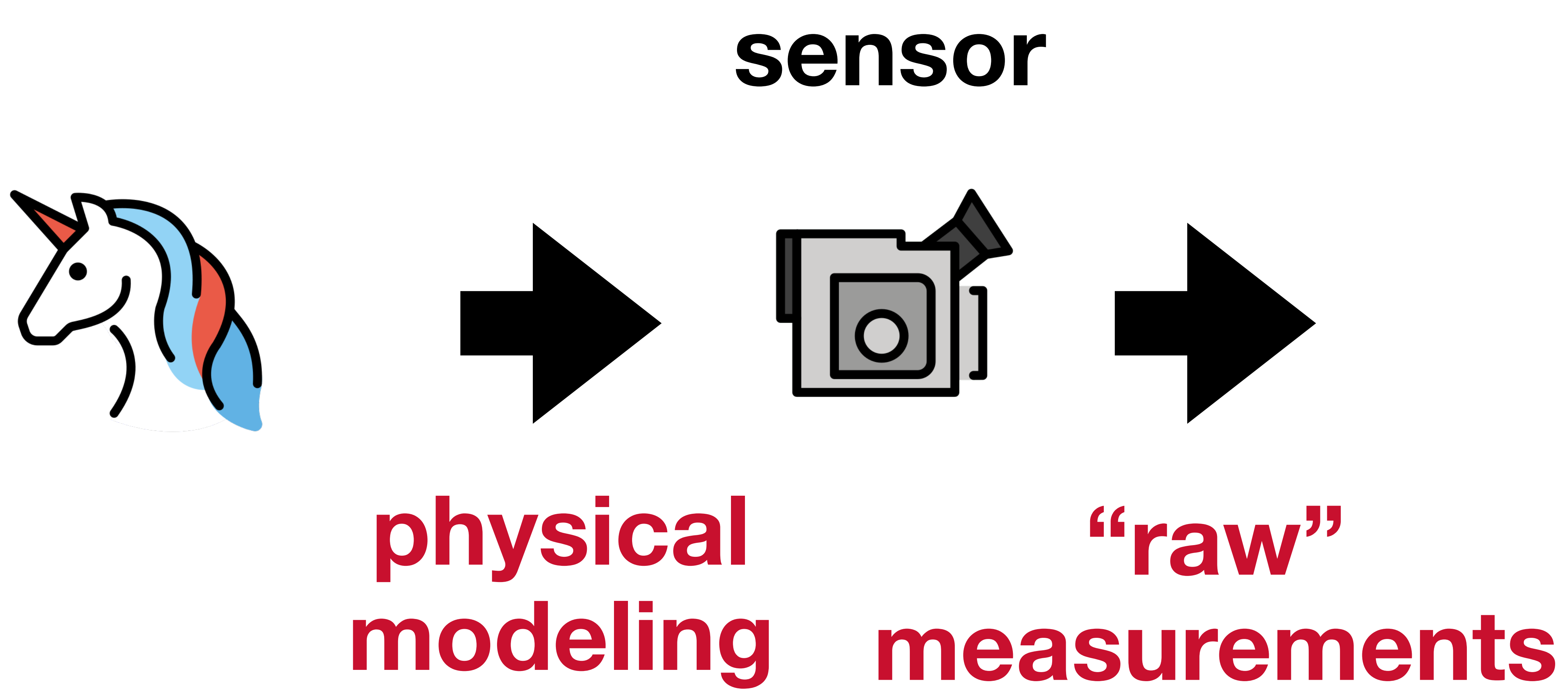
A traditional division of labor

The EE/CS divide in some sense



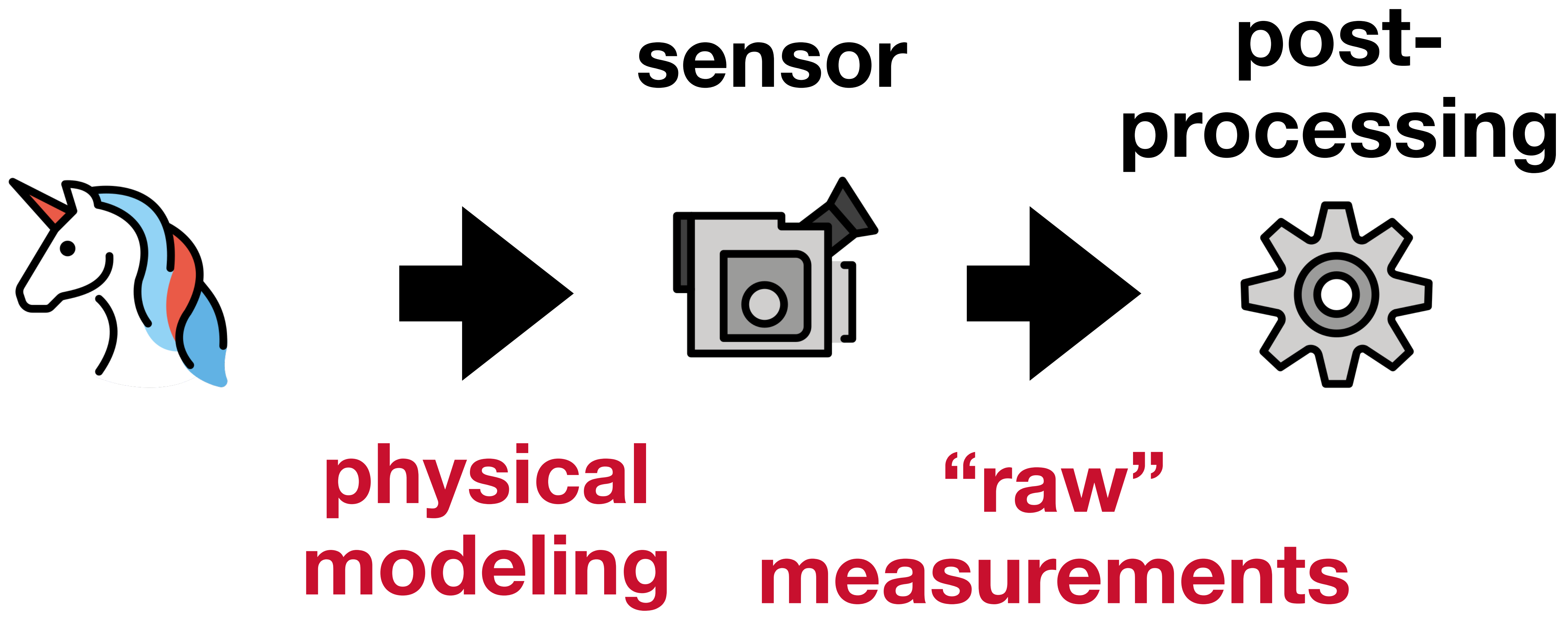
A traditional division of labor

The EE/CS divide in some sense



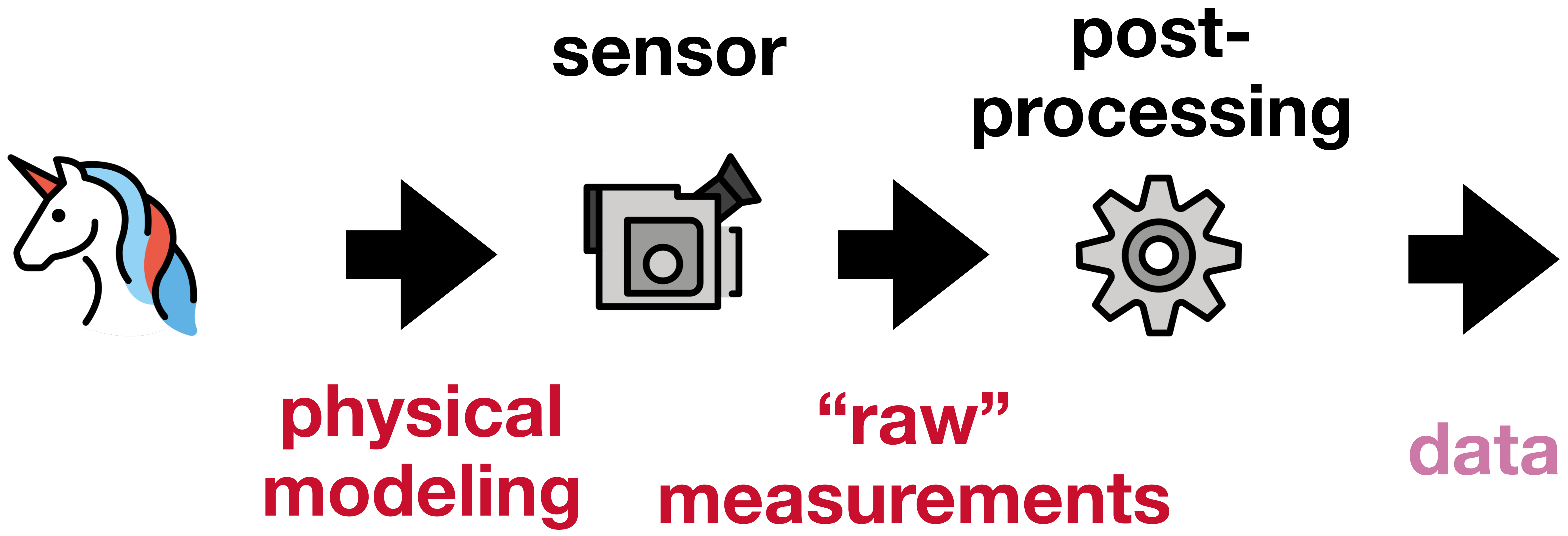
A traditional division of labor

The EE/CS divide in some sense



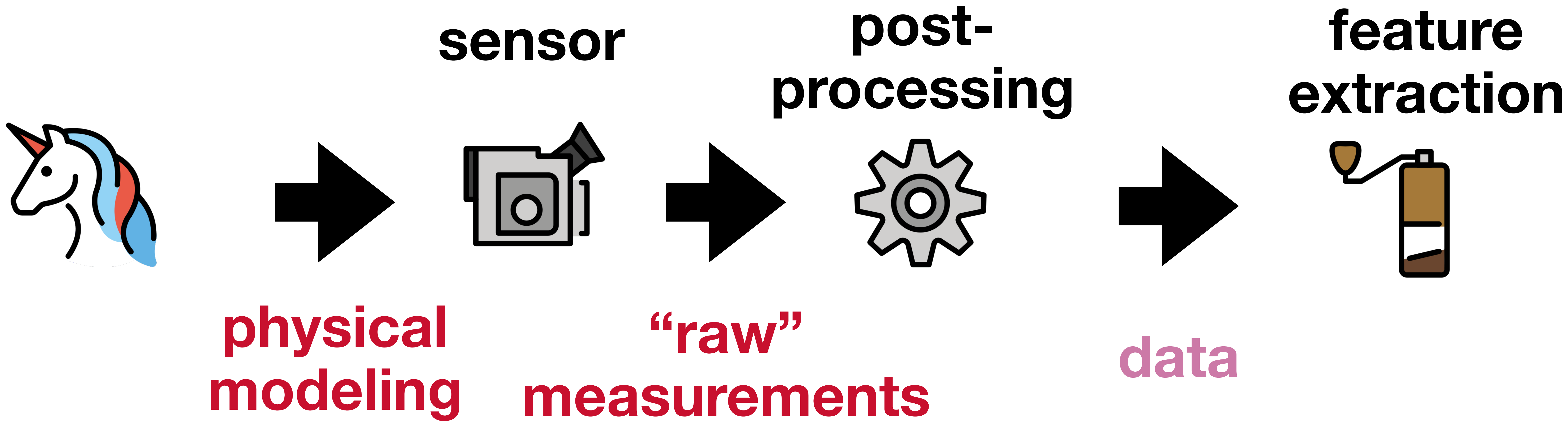
A traditional division of labor

The EE/CS divide in some sense



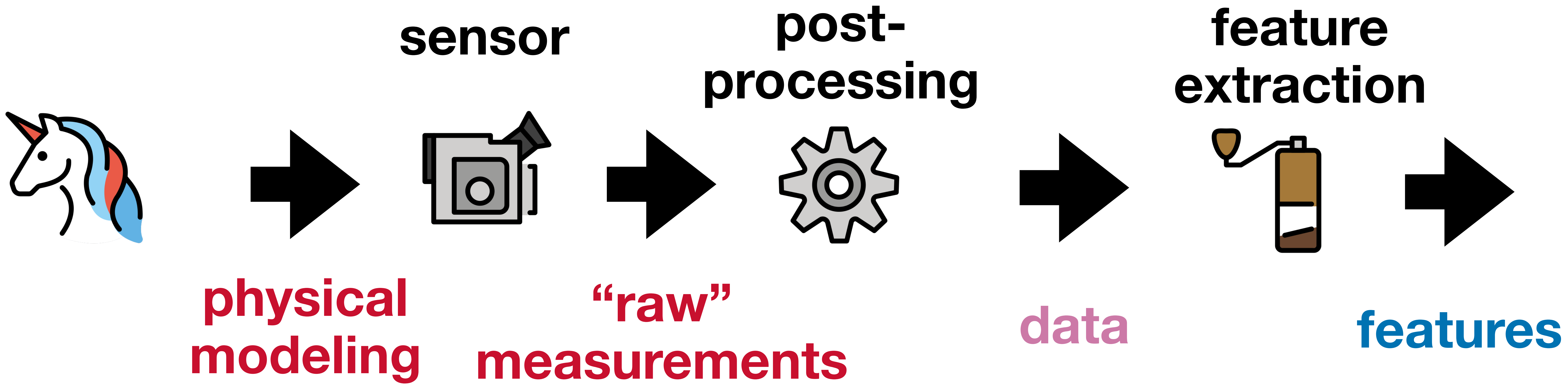
A traditional division of labor

The EE/CS divide in some sense



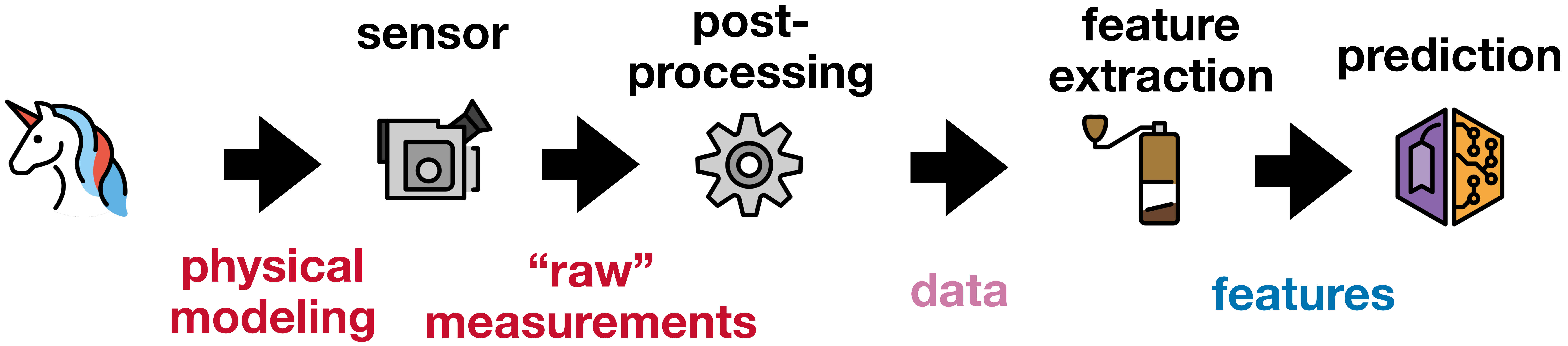
A traditional division of labor

The EE/CS divide in some sense



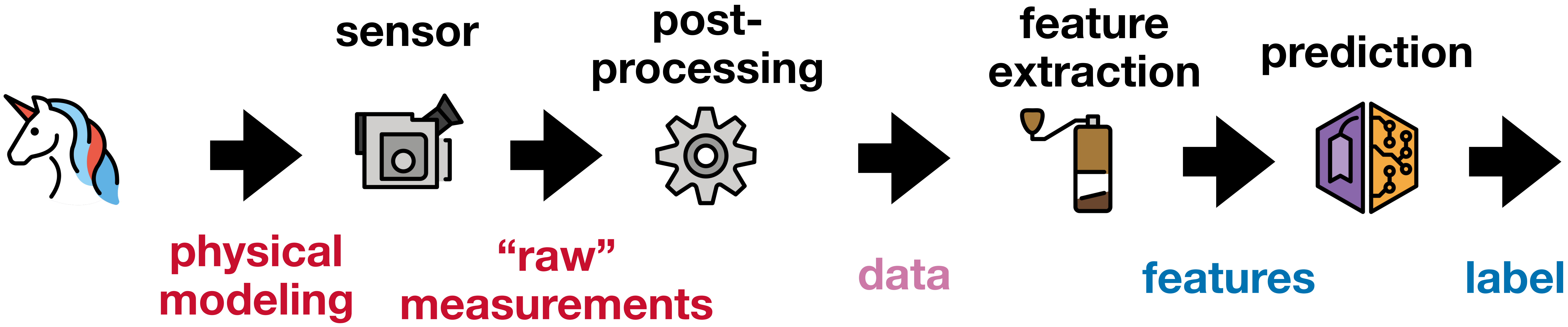
A traditional division of labor

The EE/CS divide in some sense



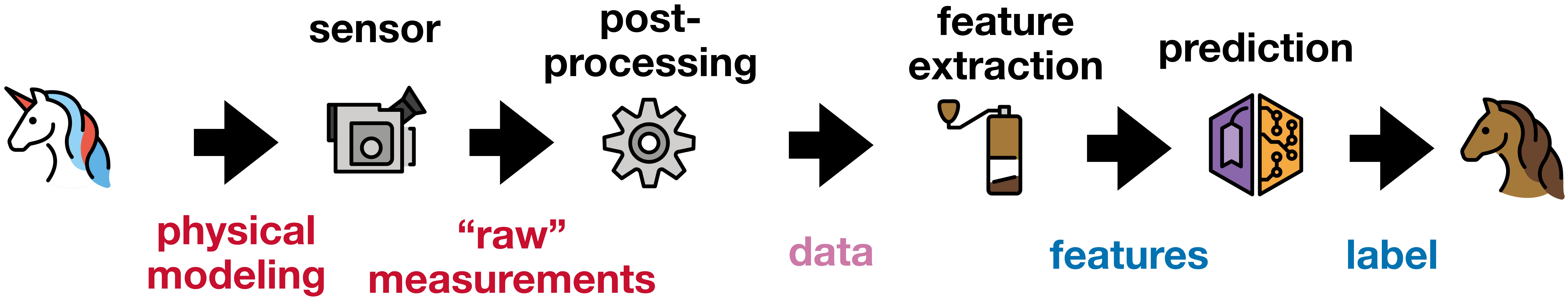
A traditional division of labor

The EE/CS divide in some sense



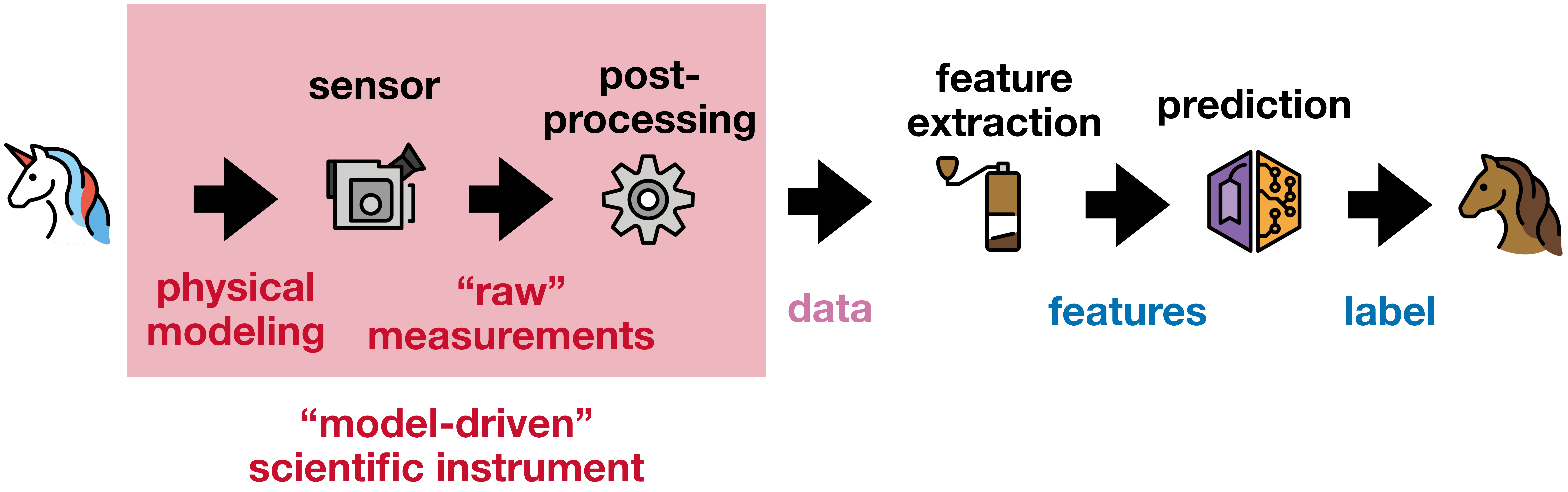
A traditional division of labor

The EE/CS divide in some sense



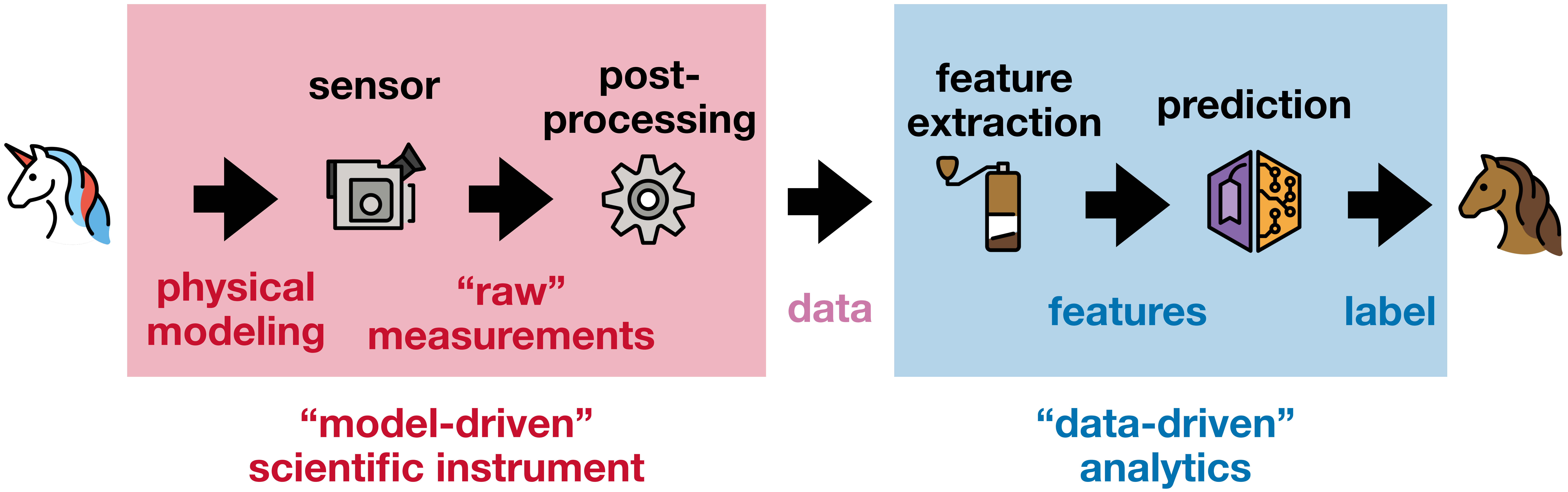
A traditional division of labor

The EE/CS divide in some sense



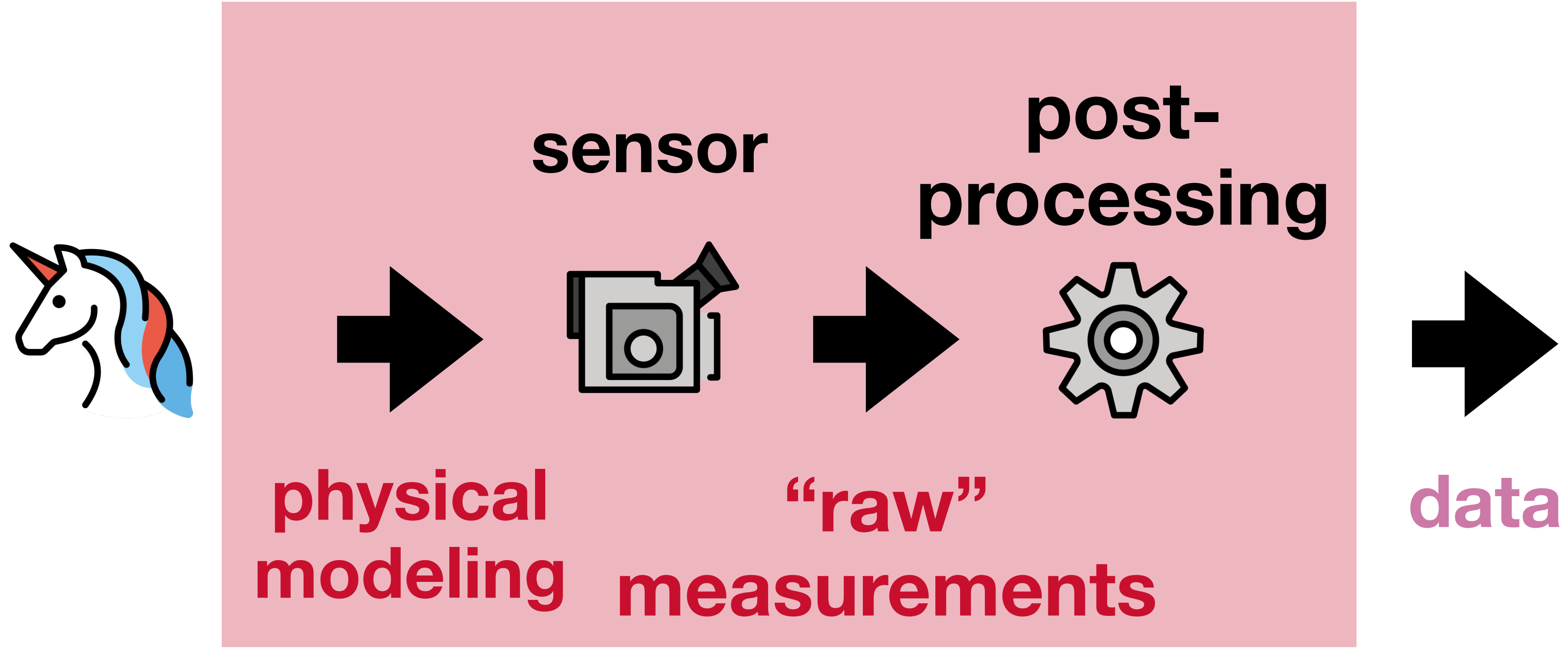
A traditional division of labor

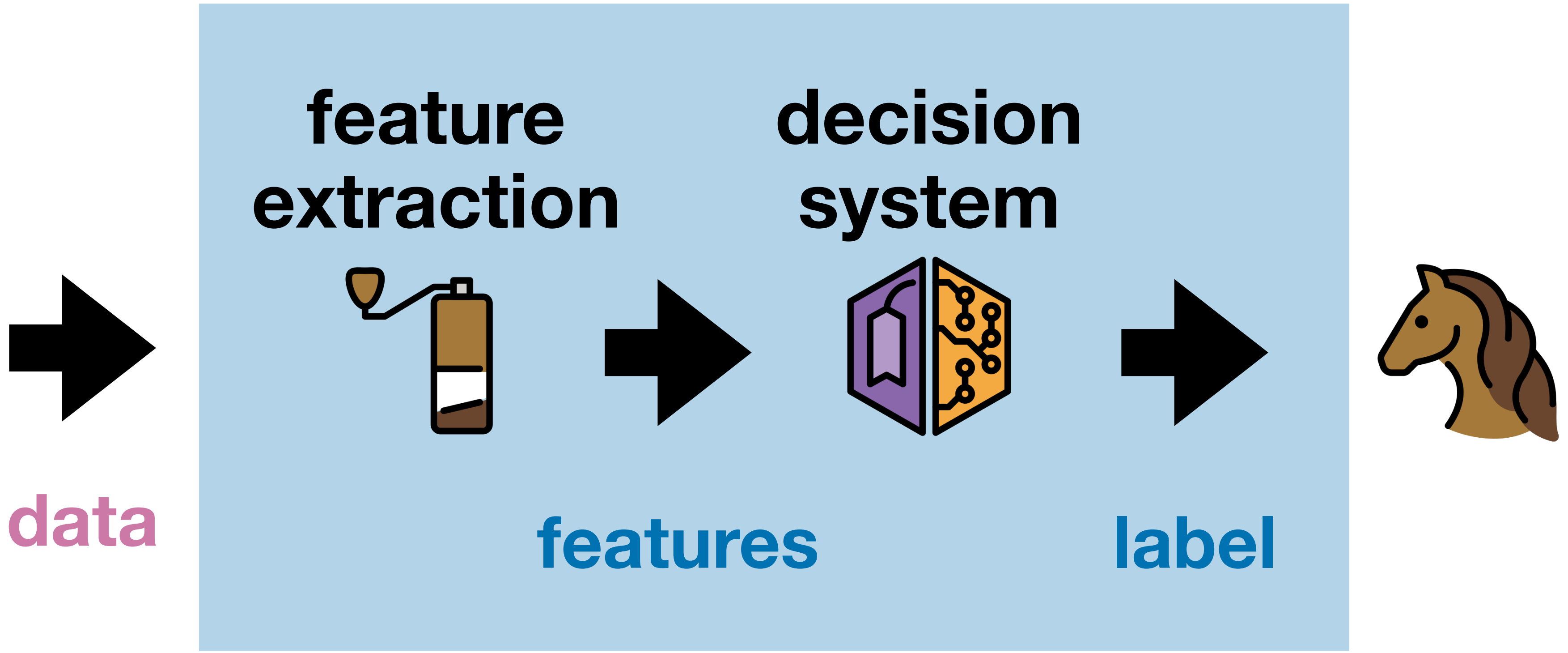
The EE/CS divide in some sense



What is “AI as instrumentation”?

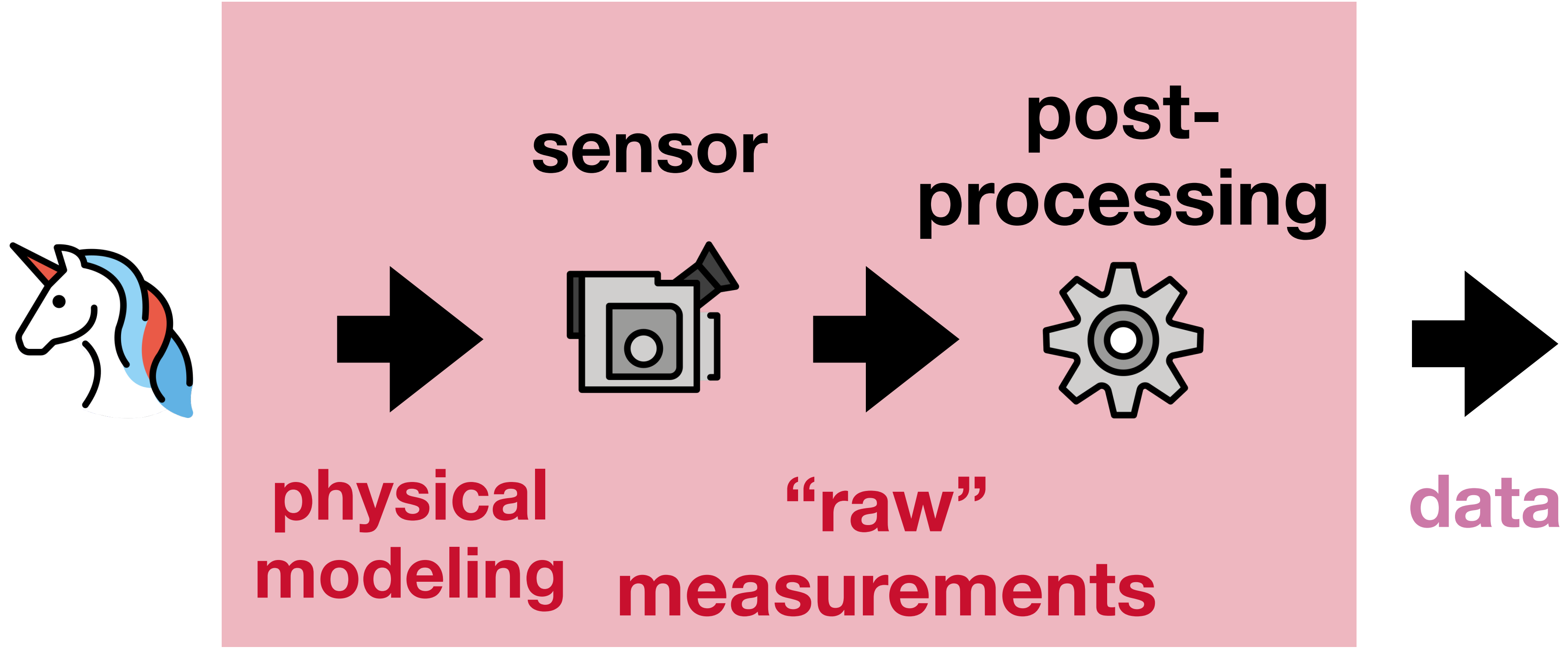
Putting neural networks into measurement devices



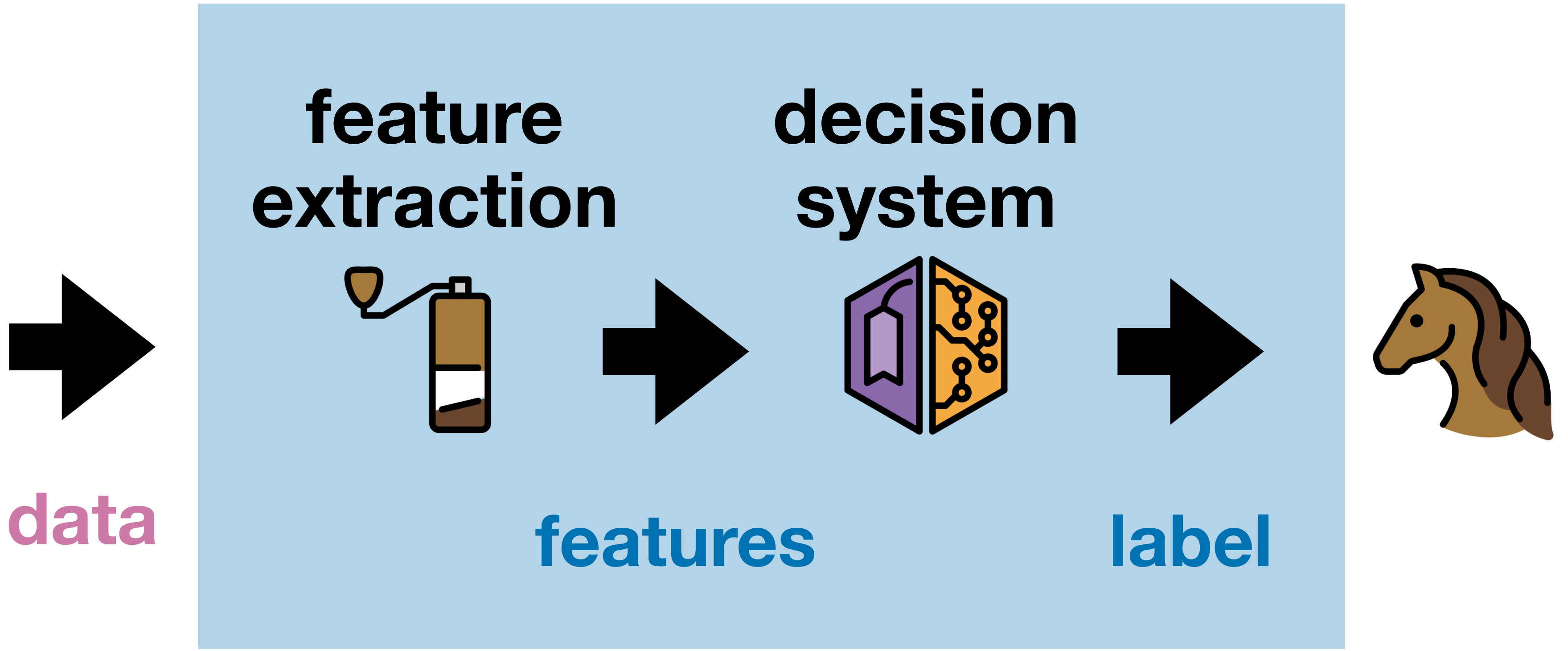


What is "AI as instrumentation"?

Putting neural networks into measurement devices

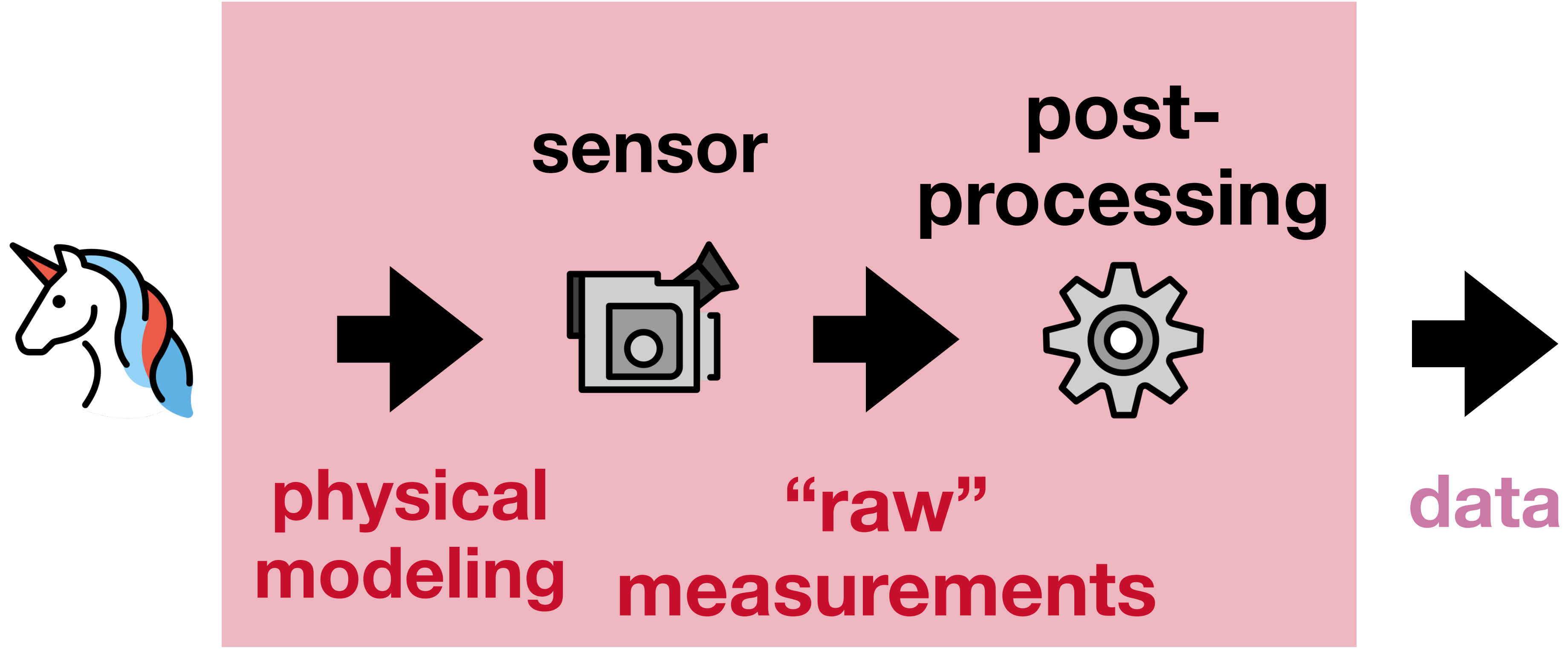


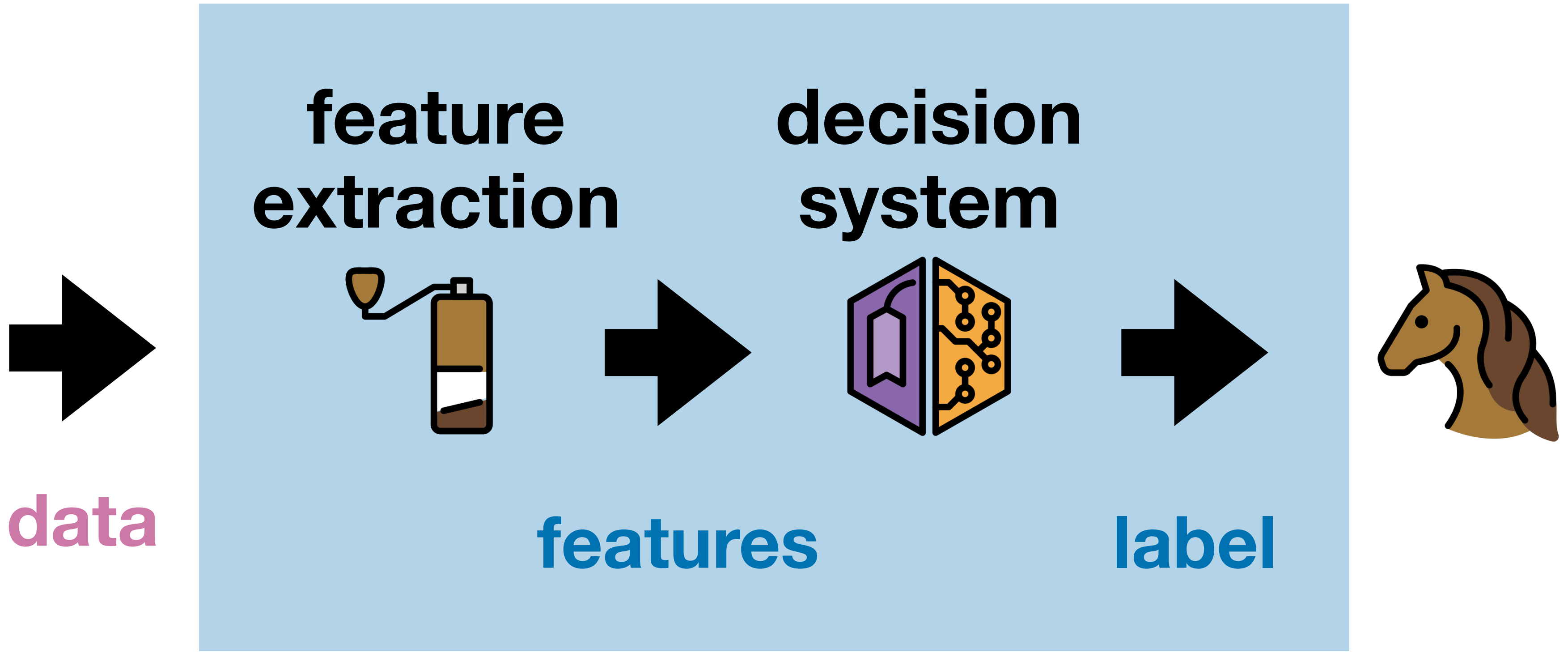
The data (images, time series, etc.) produced by a **scientific instrument** (camera/microscope/scanner) can be described in terms of the **science** (physics, chemistry, biology).



What is “AI as instrumentation”?

Putting neural networks into measurement devices



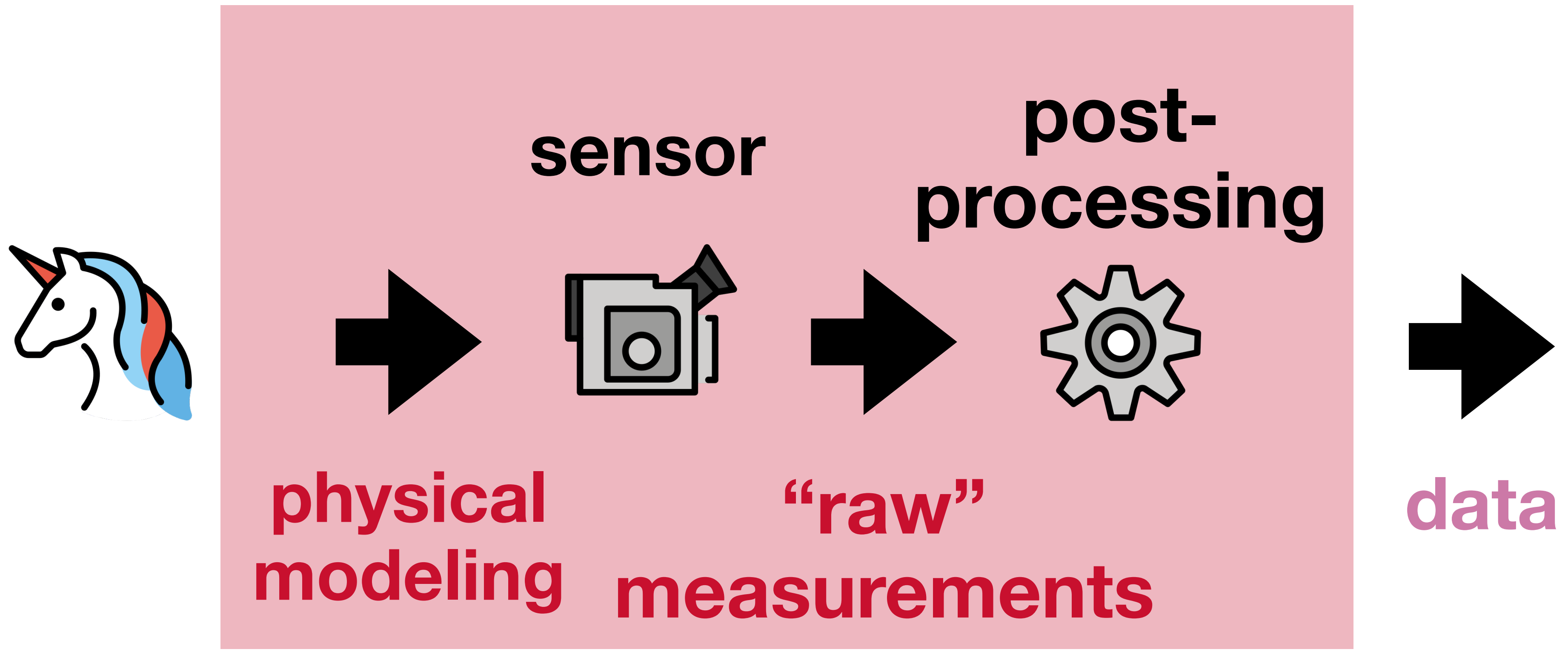


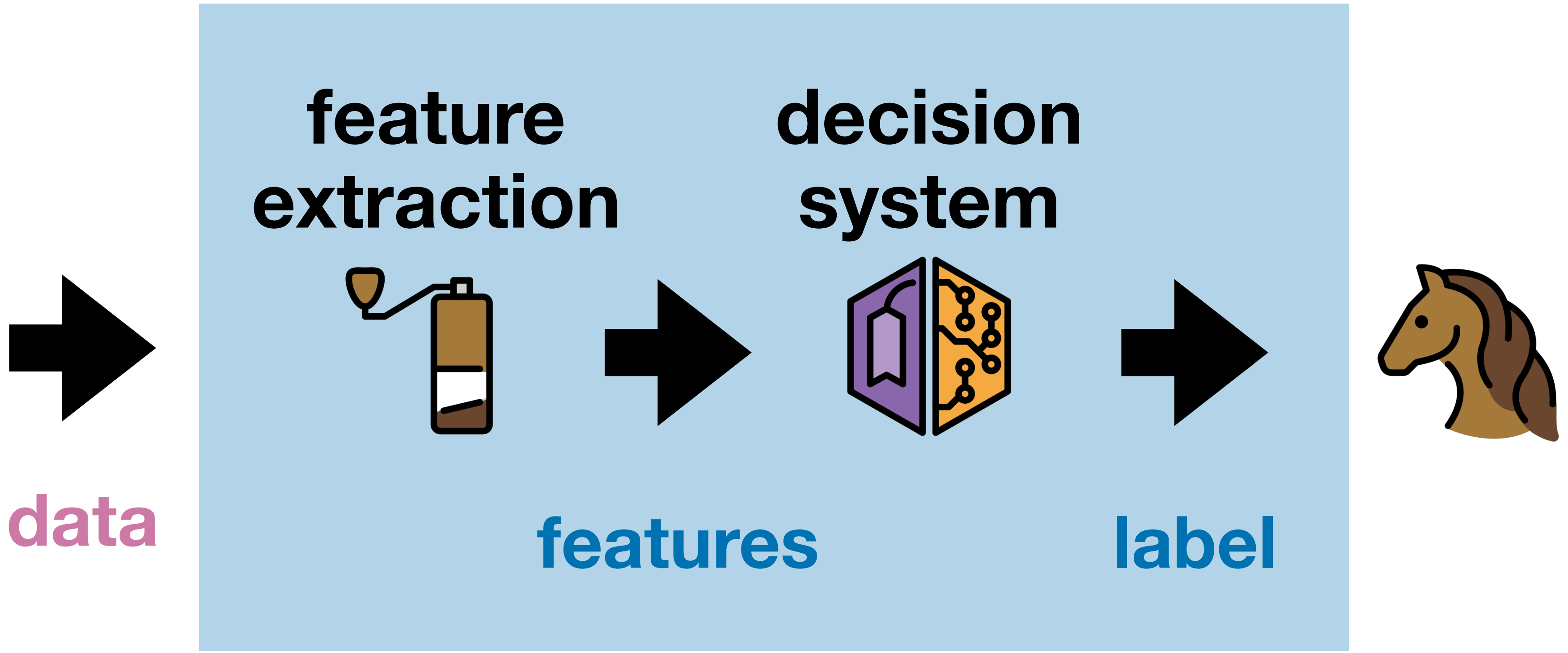
The data (images, time series, etc.) produced by a **scientific instrument** (camera/microscope/scanner) can be described in terms of the **science** (physics, chemistry, biology).

We use the data in **analytics pipelines** for more complex tasks. This relies on assumptions:

What is “AI as instrumentation”?

Putting neural networks into measurement devices





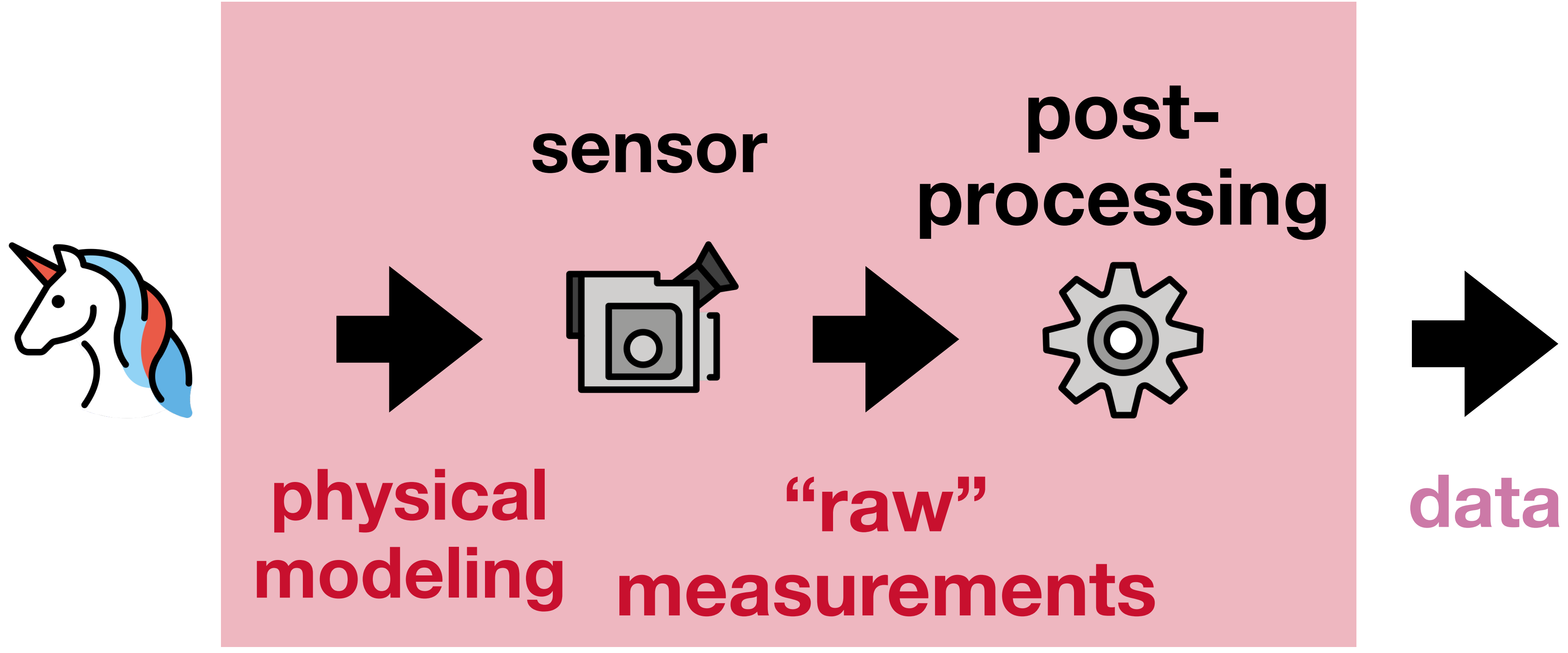
The data (images, time series, etc.) produced by a **scientific instrument** (camera/microscope/scanner) can be described in terms of the **science** (physics, chemistry, biology).

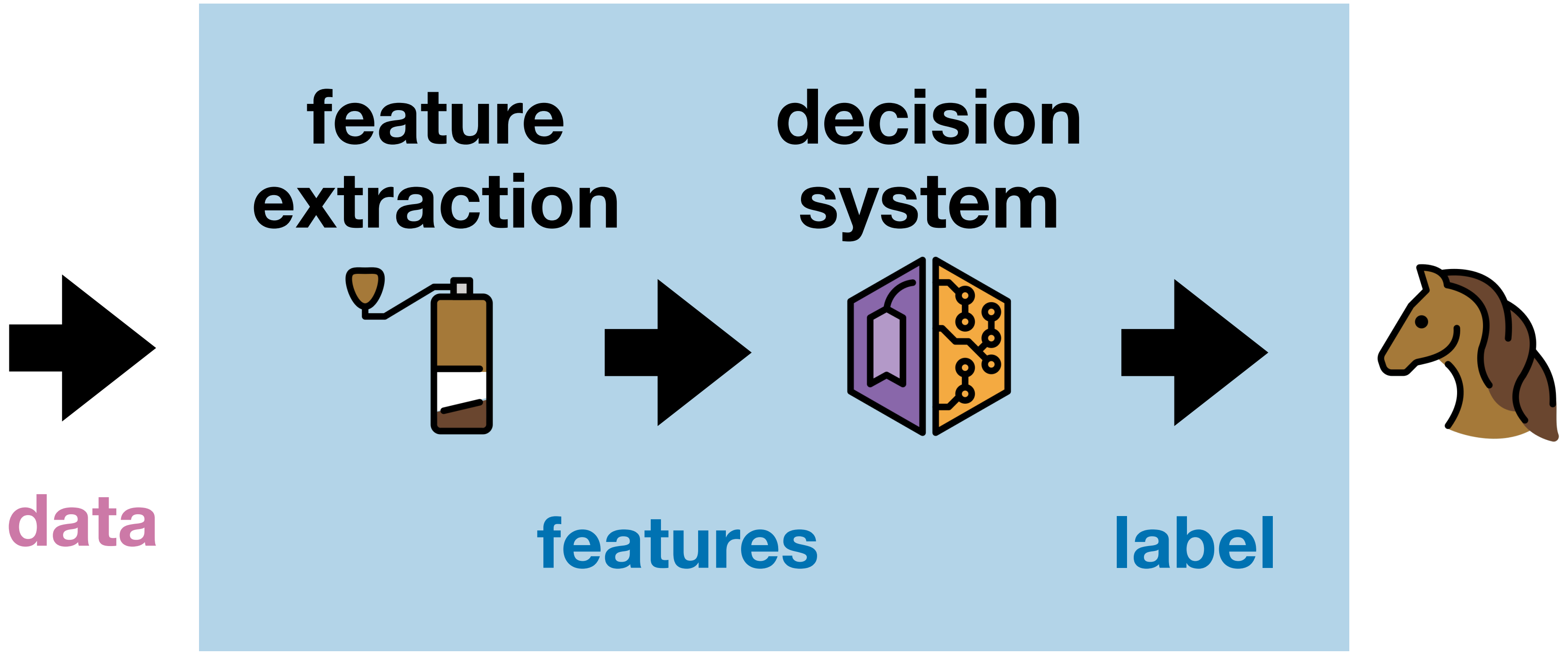
We use the data in **analytics pipelines** for more complex tasks. This relies on assumptions:

- Data from the **same camera** is “consistent”.

What is “AI as instrumentation”?

Putting neural networks into measurement devices





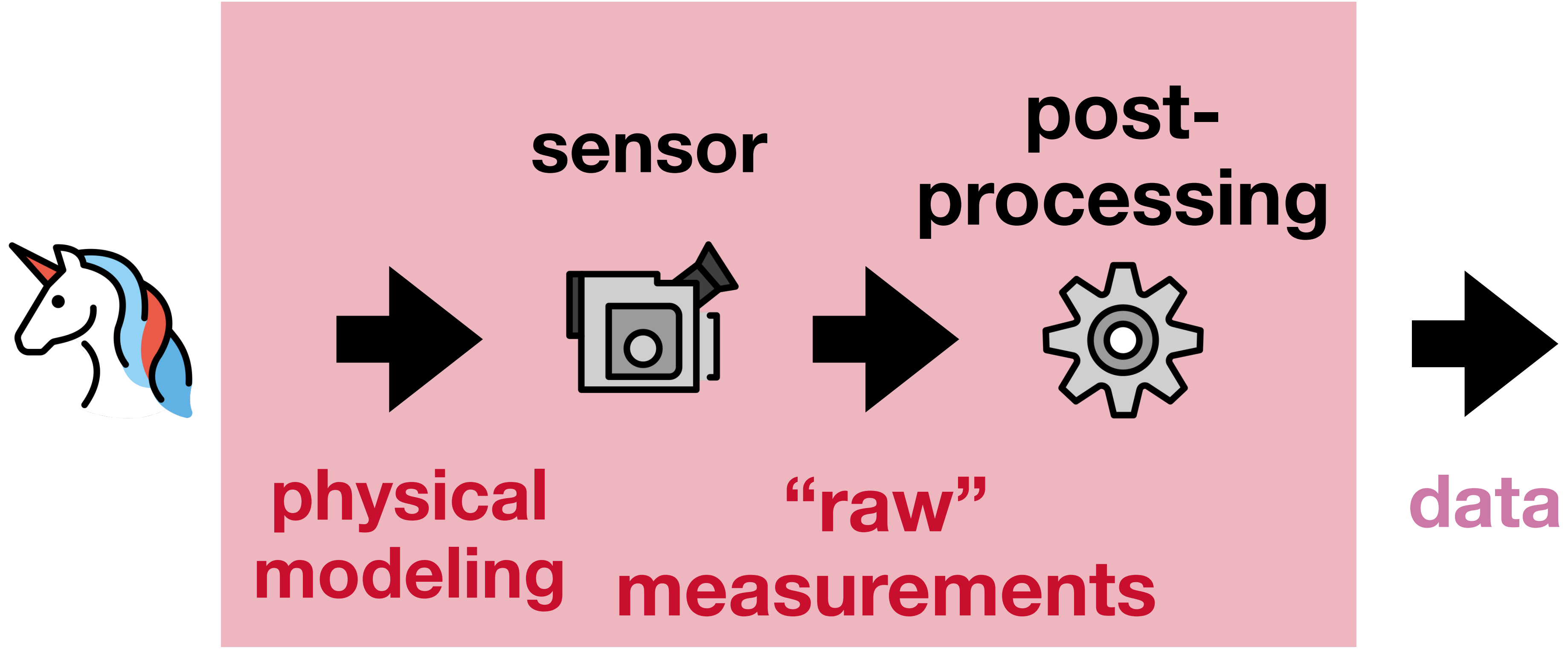
The data (images, time series, etc.) produced by a **scientific instrument** (camera/microscope/scanner) can be described in terms of the **science** (physics, chemistry, biology).

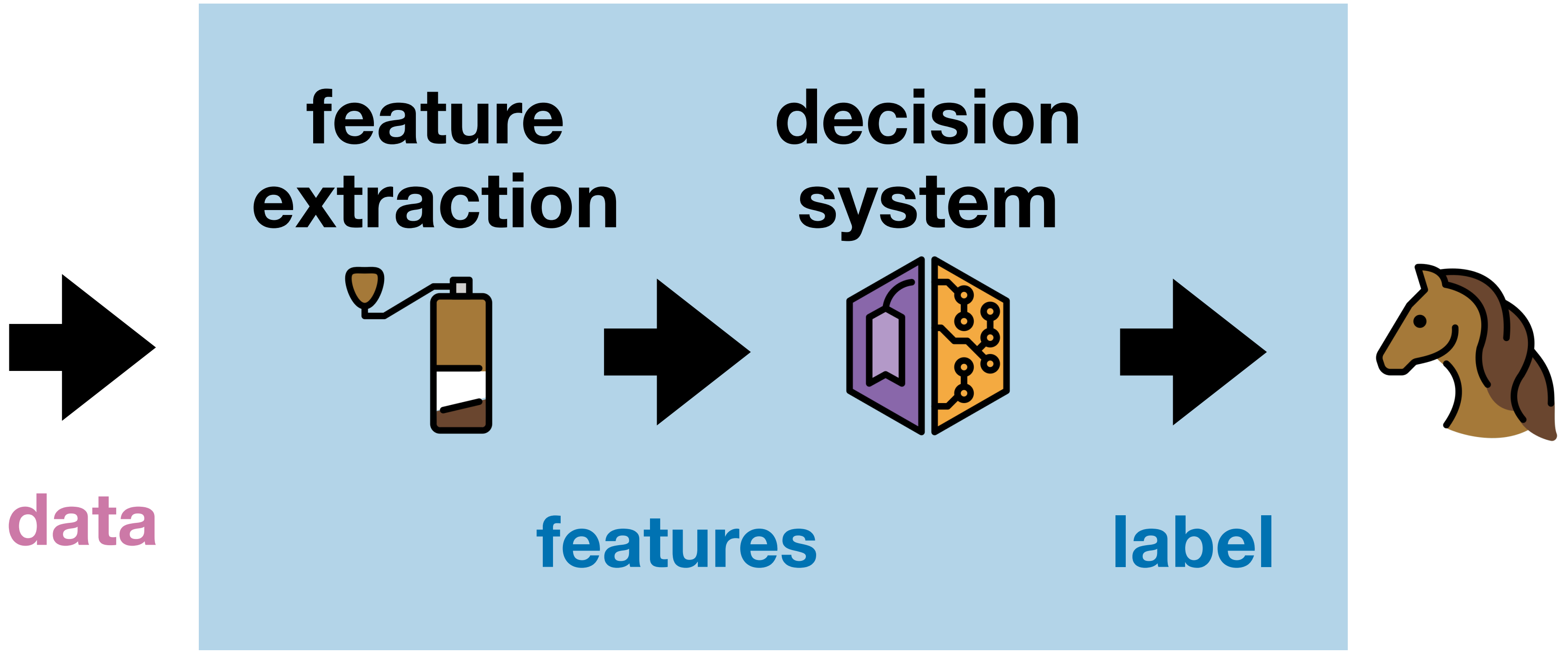
We use the data in **analytics pipelines** for more complex tasks. This relies on assumptions:

- Data from the **same camera** is “consistent”.
- Data from **different cameras** are “consistent”.

What is “AI as instrumentation”?

Putting neural networks into measurement devices





The data (images, time series, etc.) produced by a **scientific instrument** (camera/microscope/ scanner) can be described in terms of the **science** (physics, chemistry, biology).

We use the data in **analytics pipelines** for more complex tasks. This relies on assumptions:

- Data from the **same camera** is “consistent”.
- Data from **different cameras** are “consistent”.

If we put AI “into the camera” will these be true?

It already is not true in actual practice

Assumptions are wrong, but maybe correctable?

Images: pngrepo.com, Tung et al. (2017), OpenMoji.org

It already is not true in actual practice

Assumptions are wrong, but maybe correctable?

Data are almost never consistent in the ways we assume.

Images: pngrepo.com, Tung et al. (2017), OpenMoji.org

It already is not true in actual practice

Assumptions are wrong, but maybe correctable?

Data are almost never consistent in the ways we assume.

- Calibration issues

Images: pngrepo.com, Tung et al. (2017), OpenMoji.org

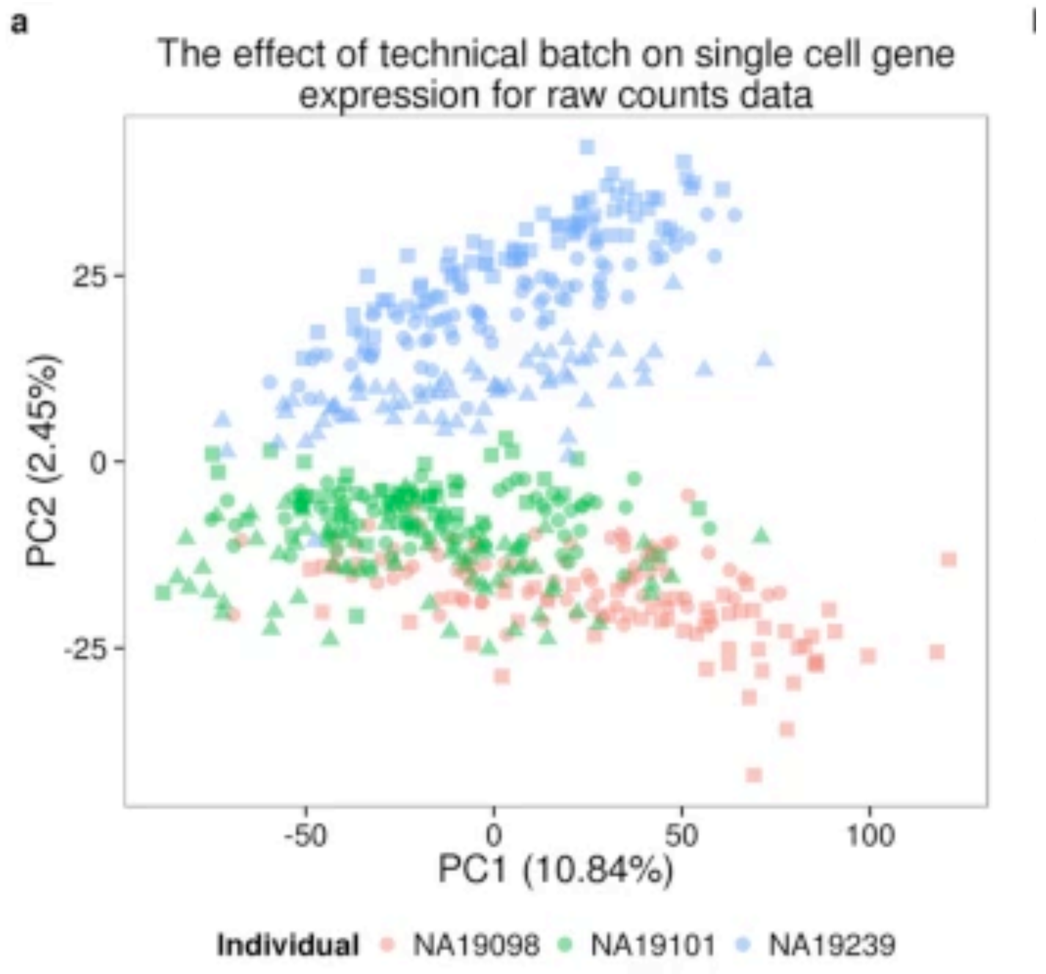
It already is not true in actual practice

Assumptions are wrong, but maybe correctable?

Data are almost never consistent in the ways we assume.

- Calibration issues

- “Batch effects” (c.f. DNA/RNA sequencing)



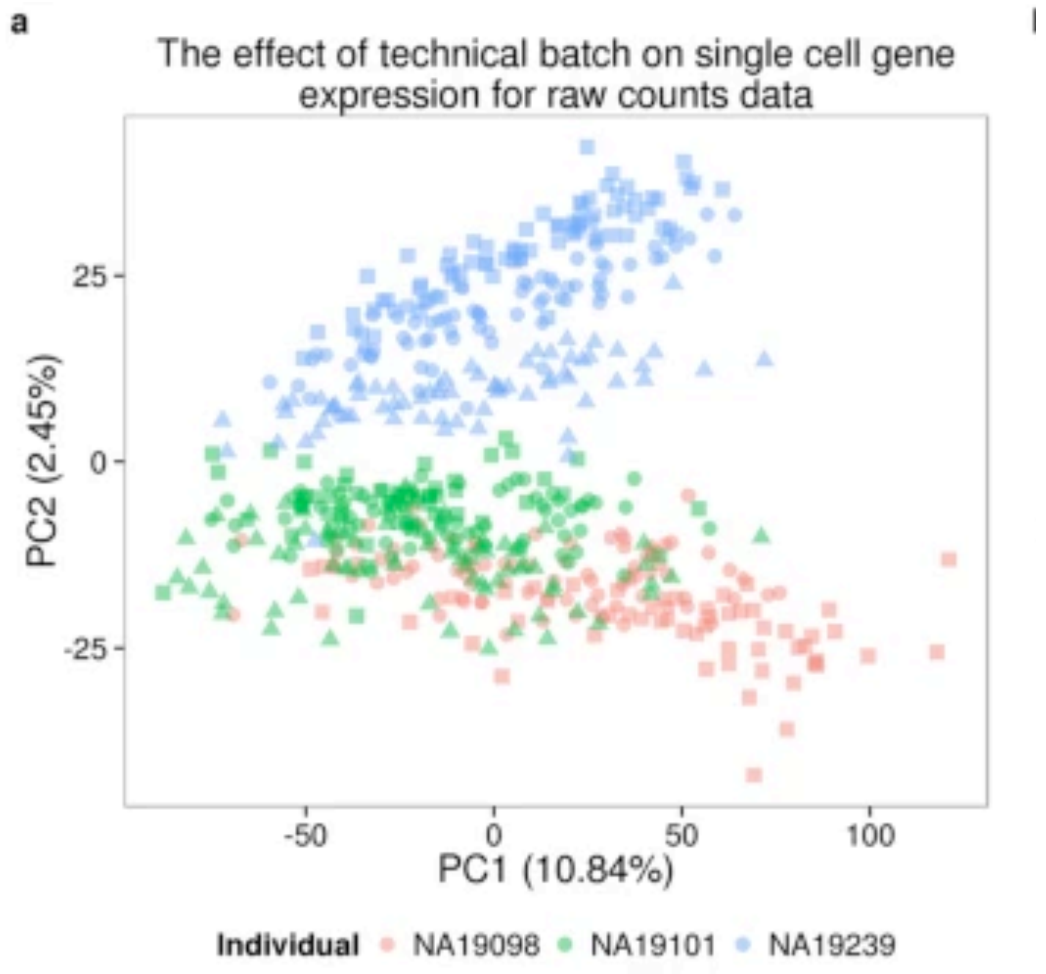
Images: pngrepo.com, Tung et al. (2017), OpenMoji.org

It already is not true in actual practice

Assumptions are wrong, but maybe correctable?

Data are almost never consistent in the ways we assume.

- Calibration issues
- “Batch effects” (c.f. DNA/RNA sequencing)
- Information forensics



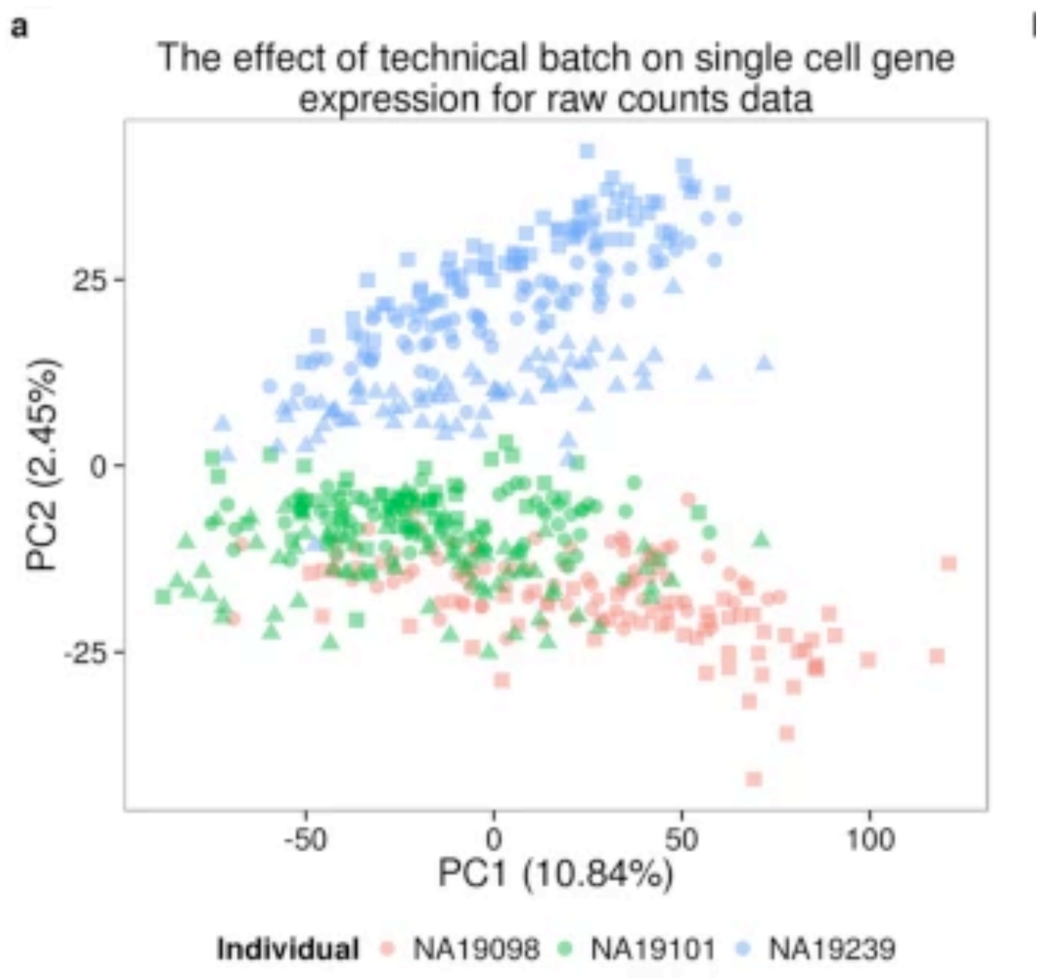
Images: pngrepo.com, Tung et al. (2017), OpenMoji.org

It already is not true in actual practice

Assumptions are wrong, but maybe correctable?

Data are almost never consistent in the ways we assume.

- Calibration issues
- “Batch effects” (c.f. DNA/RNA sequencing)
- Information forensics
- Sampling bias



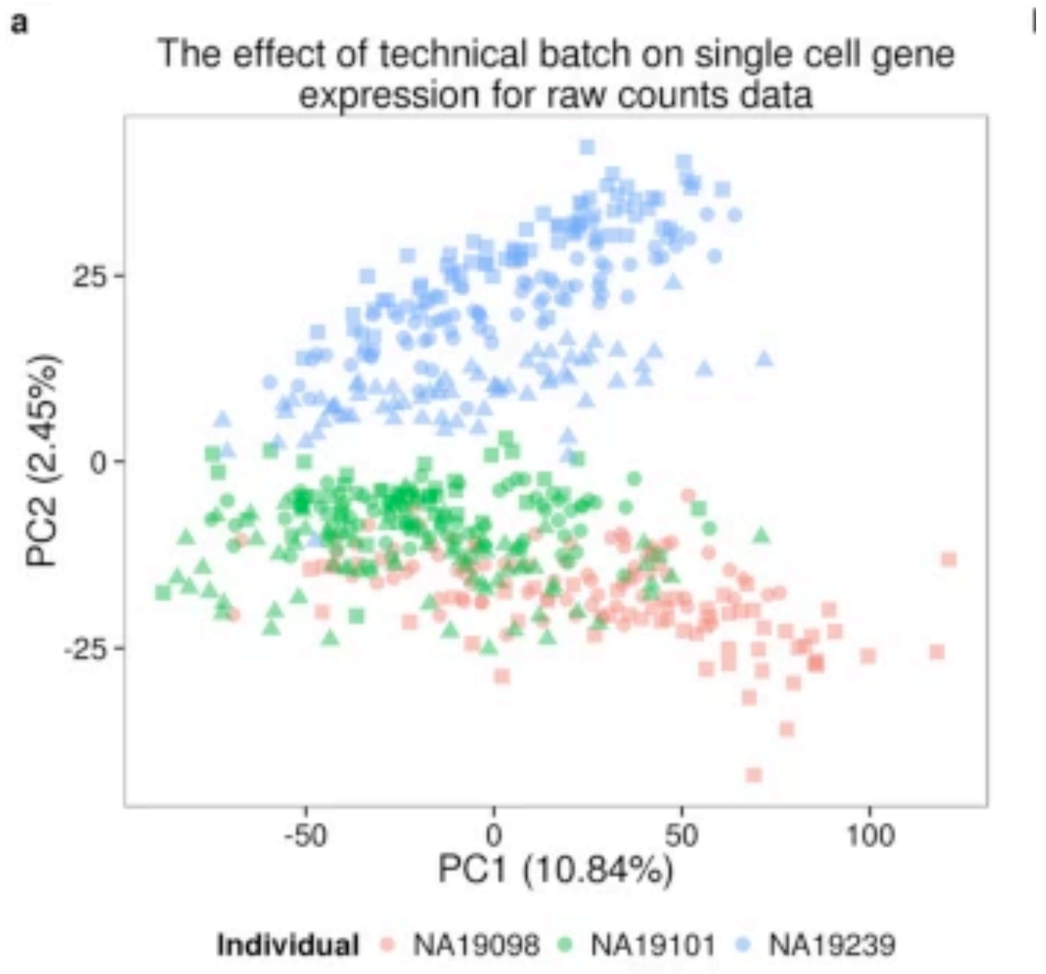
Images: pngrepo.com, Tung et al. (2017), OpenMoji.org

It already is not true in actual practice

Assumptions are wrong, but maybe correctable?

Data are almost never consistent in the ways we assume.

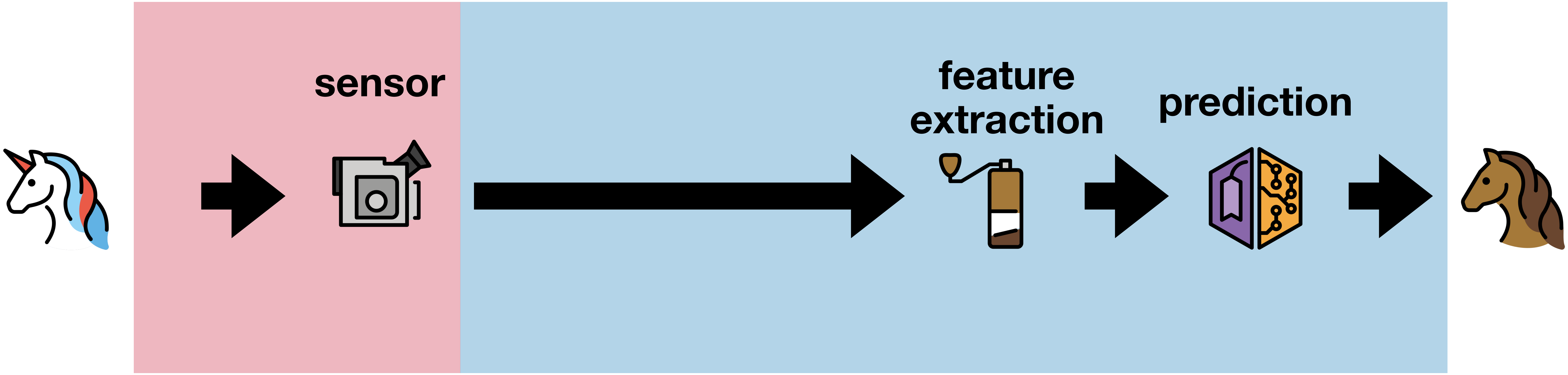
- Calibration issues
- “Batch effects” (c.f. DNA/RNA sequencing)
- Information forensics
- Sampling bias
- Etc…



Images: pngrepo.com, Tung et al. (2017), OpenMoji.org

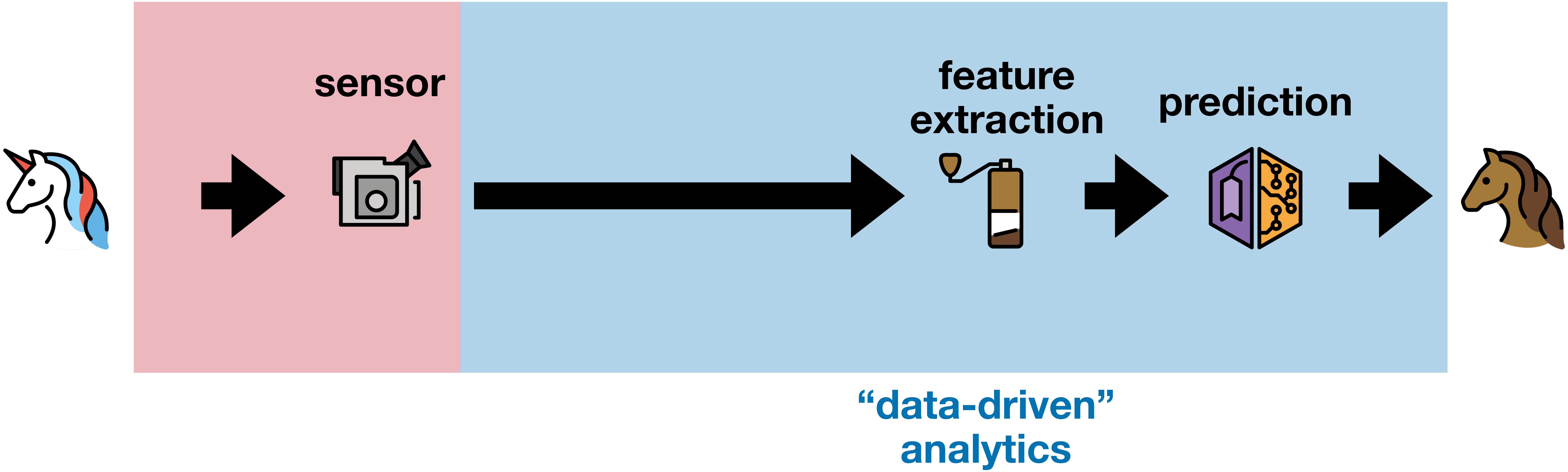
Pushing the kitchen sink backwards

Sensors, instrumentation, and decision support



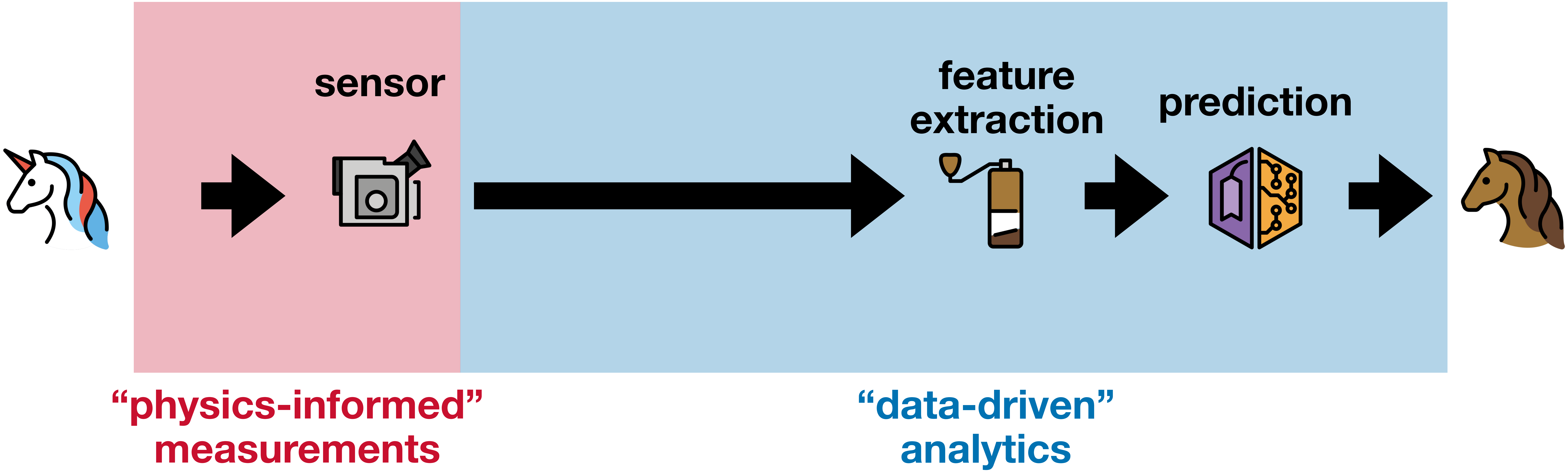
Pushing the kitchen sink backwards

Sensors, instrumentation, and decision support



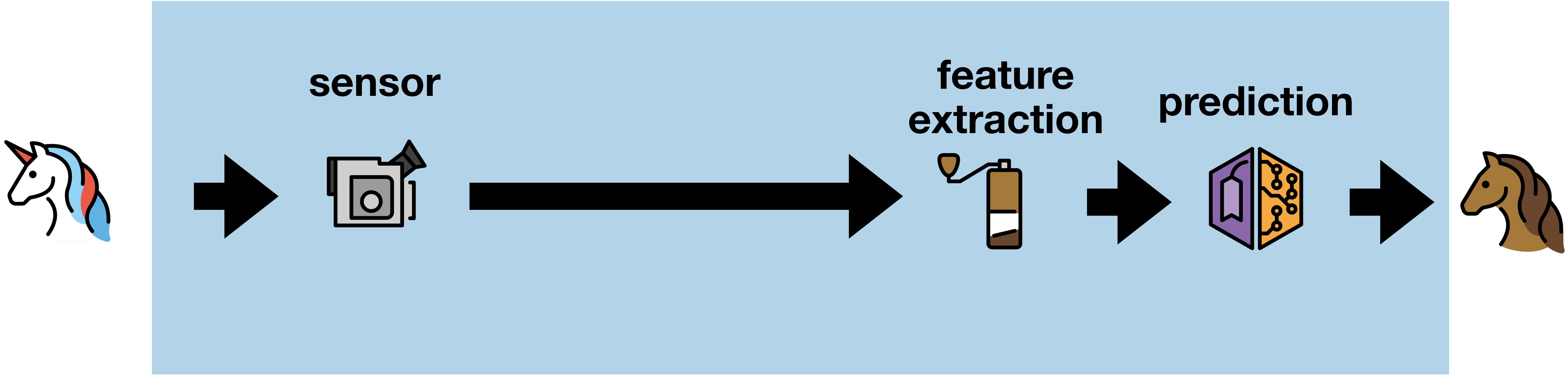
Pushing the kitchen sink backwards

Sensors, instrumentation, and decision support



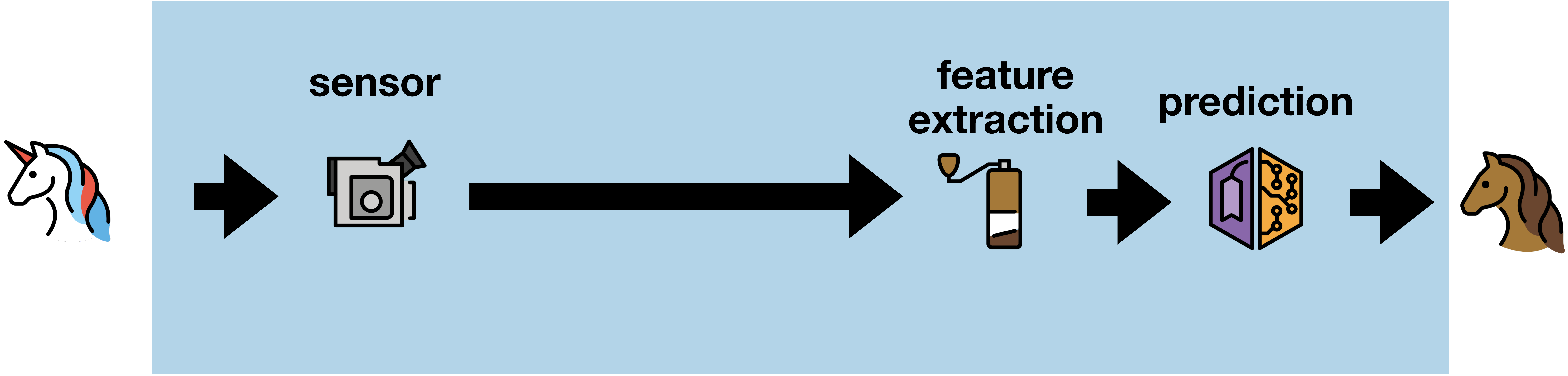
Do we even need to understand physics?

(Asking for an undergrad friend)



Do we even need to understand physics?

(Asking for an undergrad friend)



“data-driven”
scientific instrumentation

What about our assumptions?

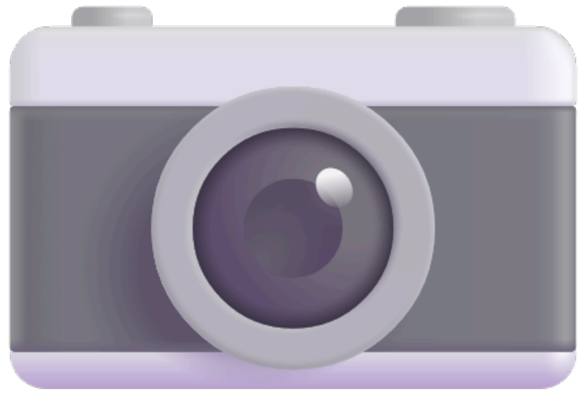
What would it mean of them to hold (if they do)?

iOS 8.3

iOS 18.4

HarmonyOS 4.0

Samsung UI 7.0



MS 3D Fluent

SerenityOS

What about our assumptions?

What would it mean of them to hold (if they do)?

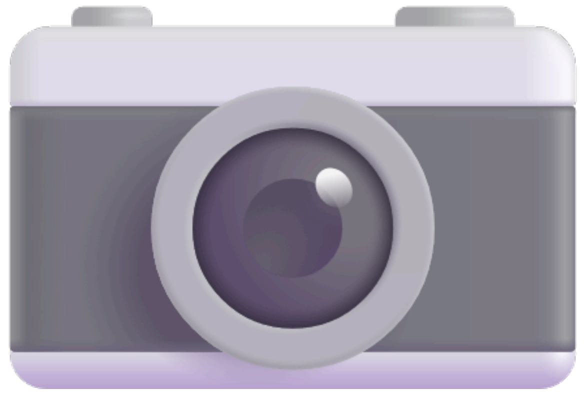
A futuristic thought experiment: **every camera has a AI model** that produces the actual image or a decision based on the image.

iOS 8.3

iOS 18.4

HarmonyOS 4.0

Samsung UI 7.0



MS 3D Fluent

SerenityOS

What about our assumptions?

What would it mean of them to hold (if they do)?

A futuristic thought experiment: **every camera has a AI model** that produces the actual image or a decision based on the image.

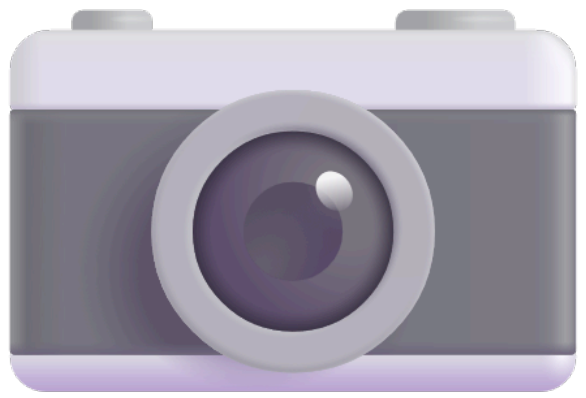
- If we build the camera twice, **will it be the same**?

iOS 8.3

iOS 18.4

HarmonyOS 4.0

Samsung UI 7.0



MS 3D Fluent

SerenityOS

What about our assumptions?

What would it mean of them to hold (if they do)?

A futuristic thought experiment: **every camera has a AI model** that produces the actual image or a decision based on the image.

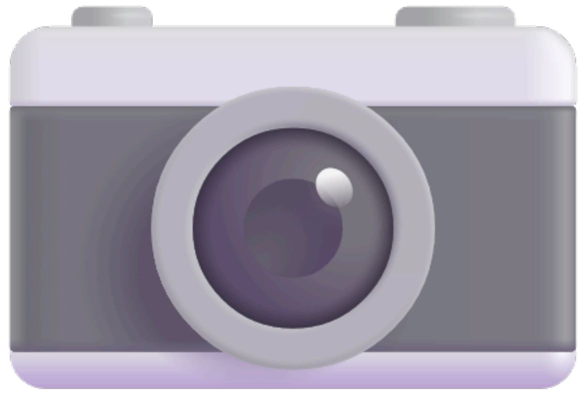
- If we build the camera twice, **will it be the same**?
- If we use two different cameras **will they give similar results**?


iOS 8.3


iOS 18.4


HarmonyOS 4.0


Samsung UI 7.0


MS 3D Fluent


SerenityOS

What about our assumptions?

What would it mean of them to hold (if they do)?

A futuristic thought experiment: **every camera has a AI model** that produces the actual image or a decision based on the image.

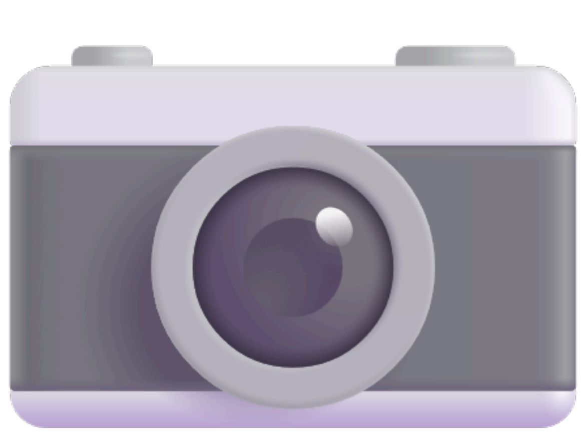
- If we build the camera twice, **will it be the same**?
- If we use two different cameras **will they give similar results**?
- **How do we compare** two models?

iOS 8.3

iOS 18.4

HarmonyOS 4.0

Samsung UI 7.0



MS 3D Fluent

SerenityOS

What about our assumptions?

What would it mean of them to hold (if they do)?

A futuristic thought experiment: **every camera has a AI model** that produces the actual image or a decision based on the image.

- If we build the camera twice, **will it be the same**?
- If we use two different cameras **will they give similar results**?
- **How do we compare** two models?

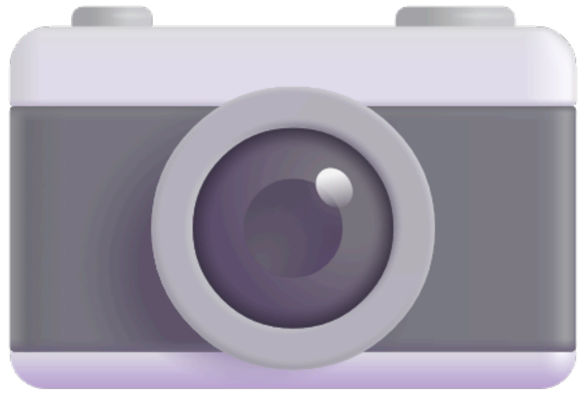
These questions are not new! We can use “classical” tools to try and understand them.

iOS 8.3

iOS 18.4

HarmonyOS 4.0

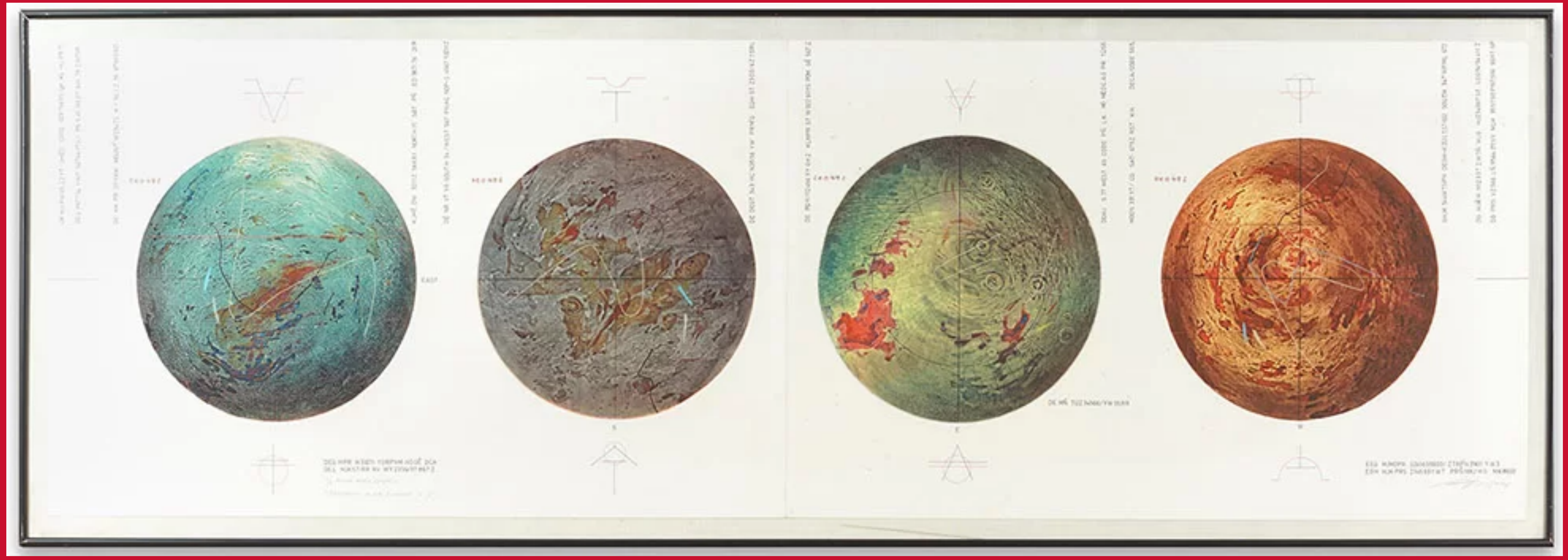
Samsung UI 7.0



MS 3D Fluent

SerenityOS

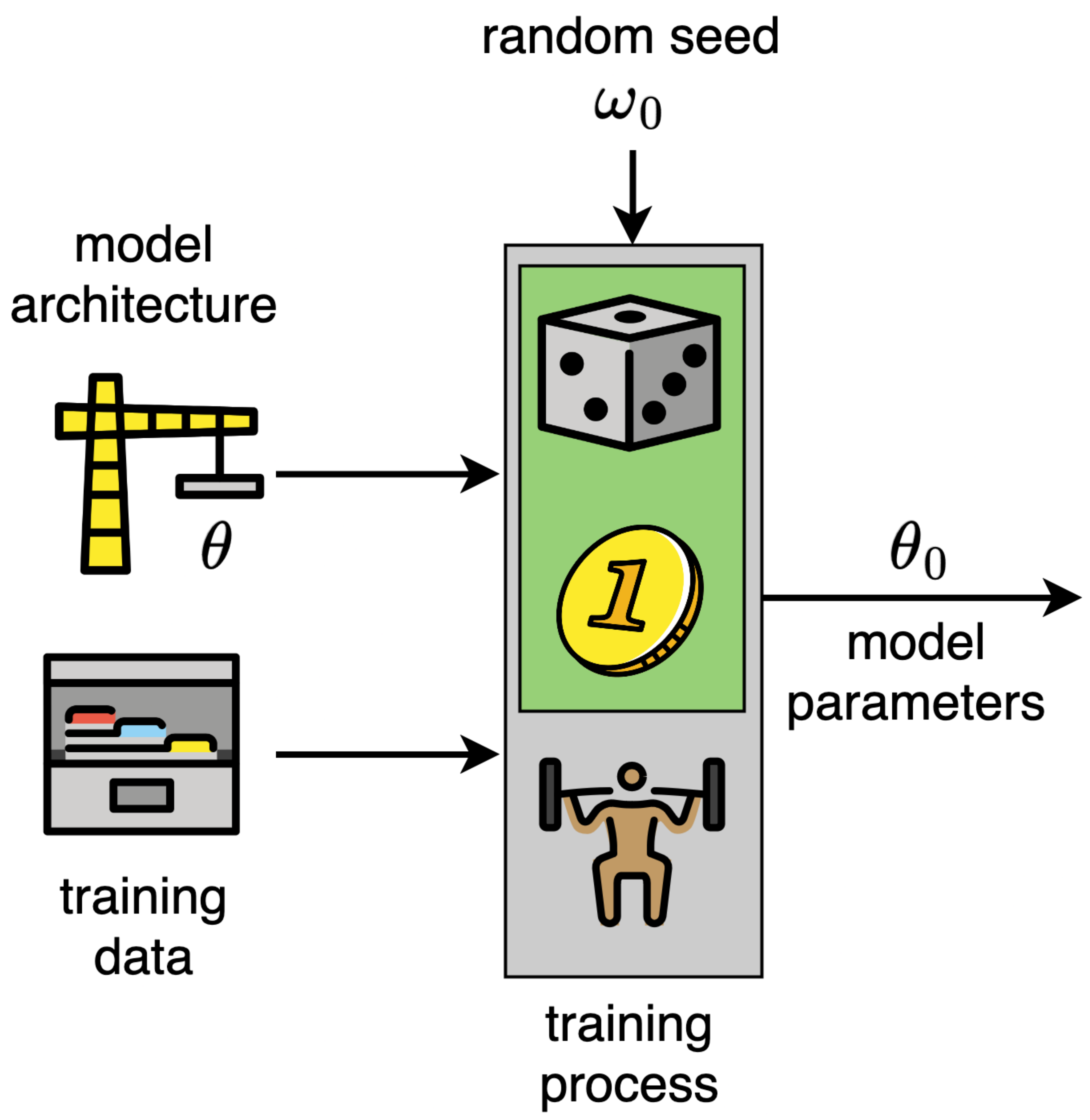
Some preliminaries



Rm Palaniappan, *Alien Planet-A*
Viscosity, pencil colour and ink on handmade paper

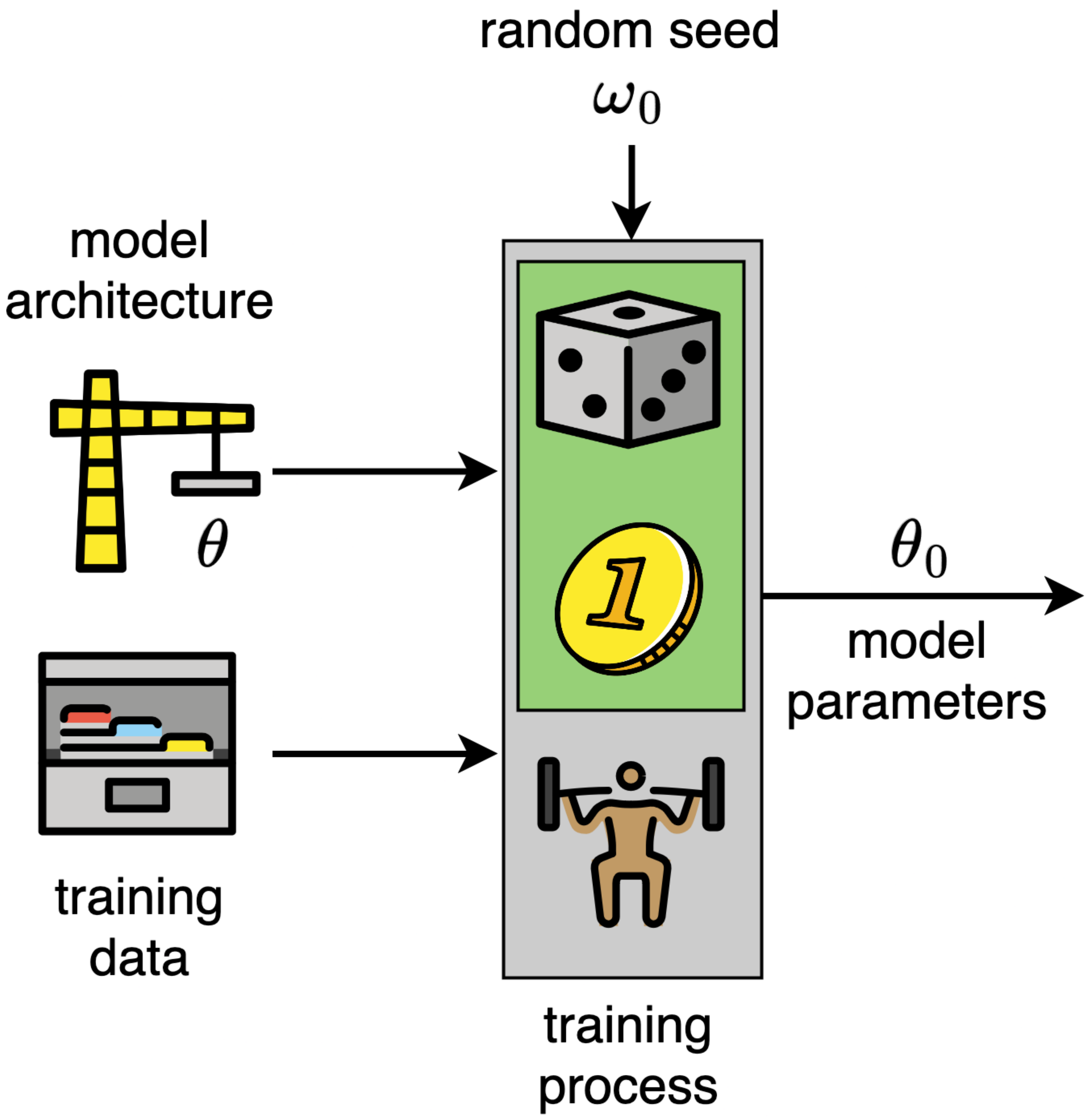
The standard statistical setup for modern ML

Machine learning as function-fitting



The standard statistical setup for modern ML

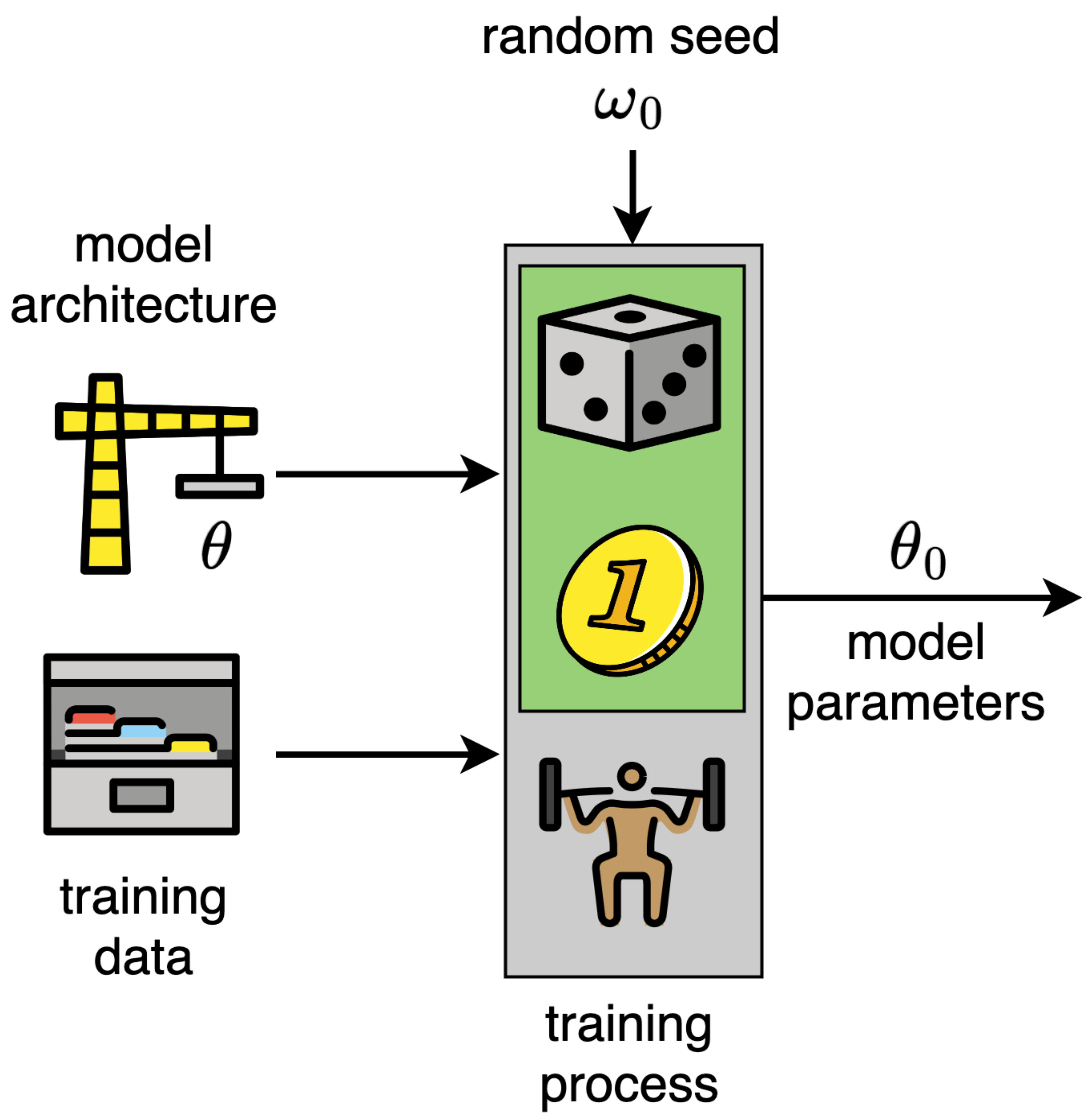
Machine learning as function-fitting



The traditional setup for estimating parameters in a statistical model (or training a neural network:

The standard statistical setup for modern ML

Machine learning as function-fitting

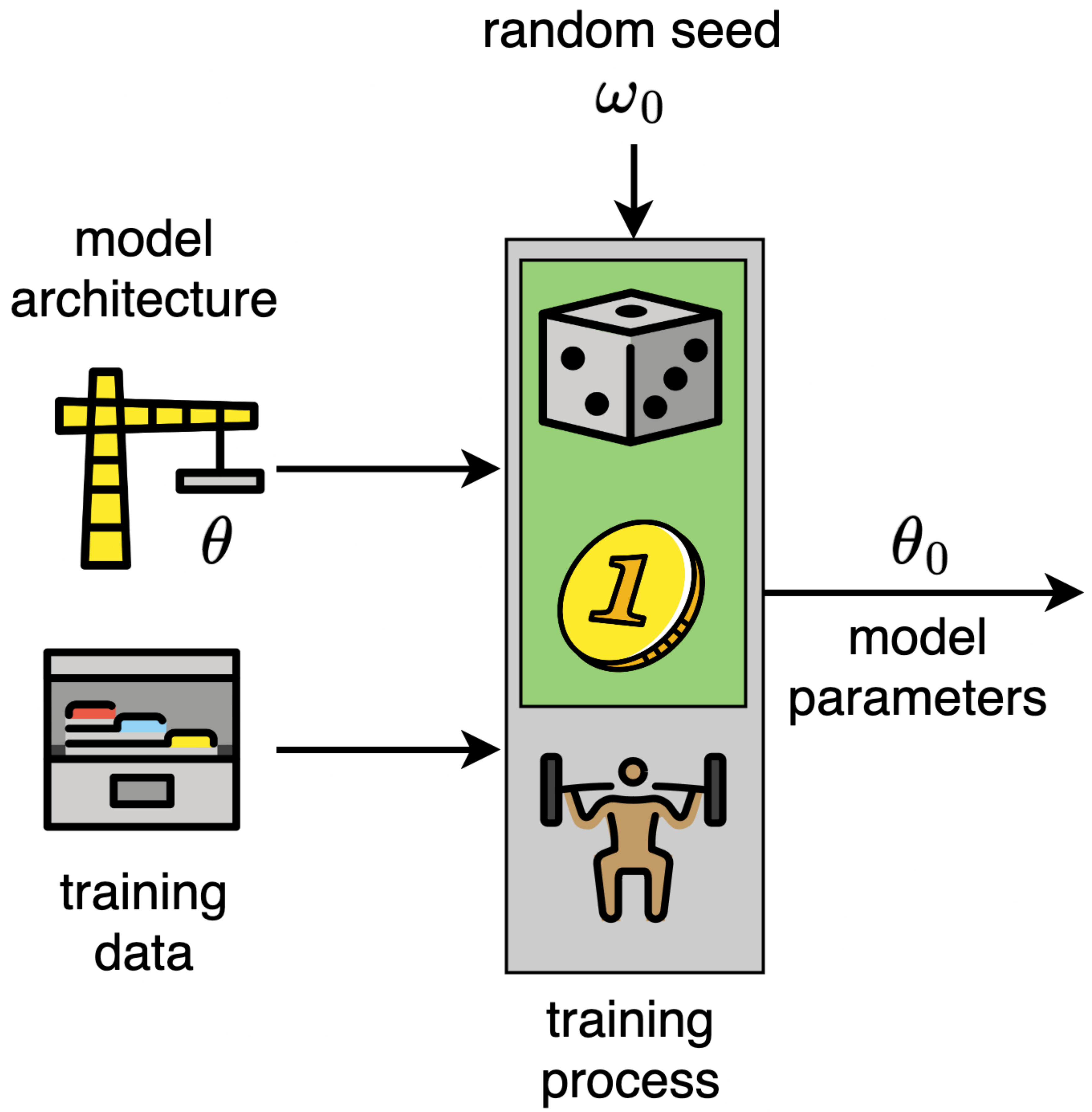


The traditional setup for estimating parameters in a statistical model (or training a neural network:

- Parameterized set of functions/models $\{f(x \mid \theta) : \theta \in \Theta\}$.

The standard statistical setup for modern ML

Machine learning as function-fitting

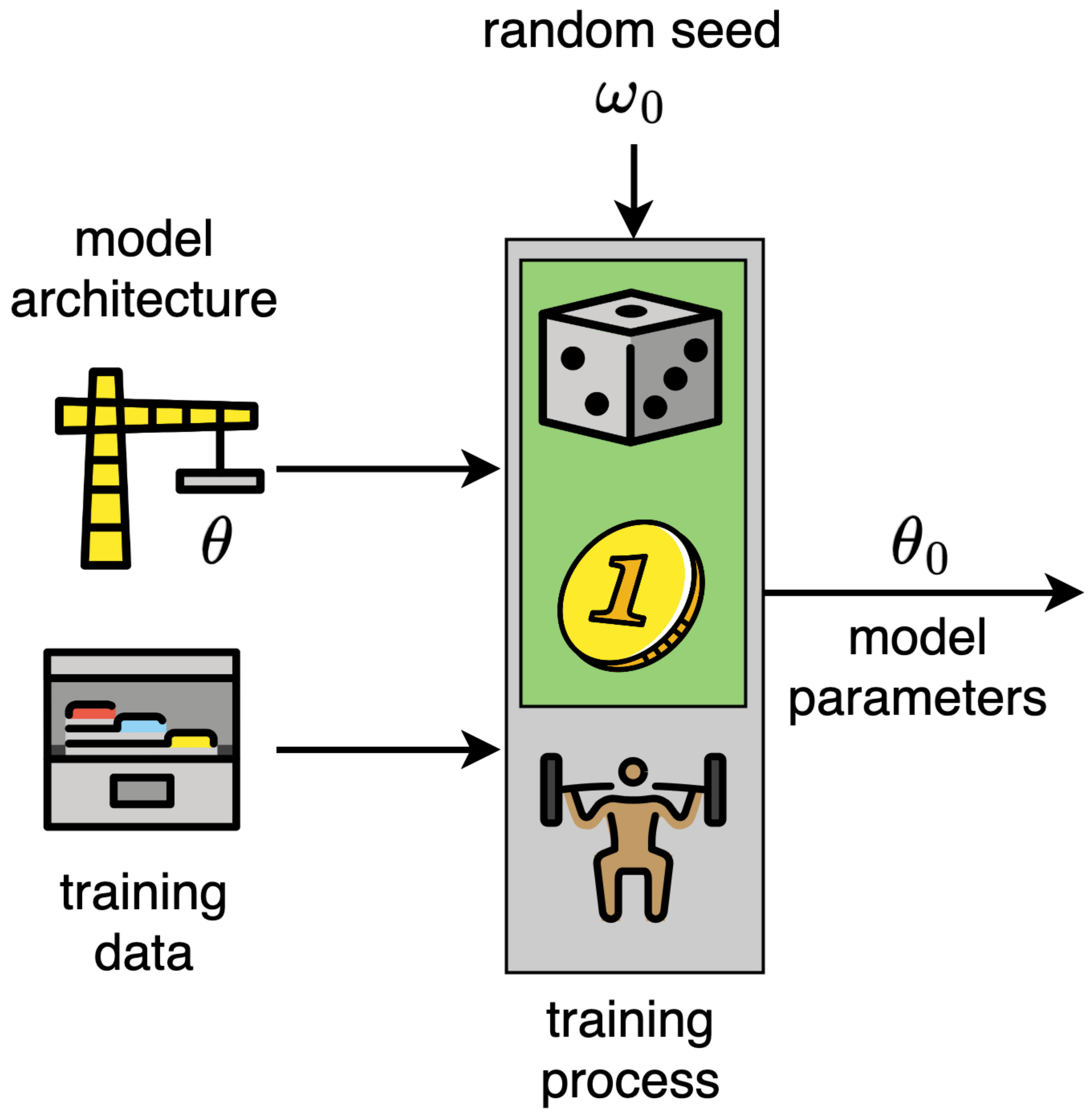


The traditional setup for estimating parameters in a statistical model (or training a neural network:

- Parameterized set of functions/models $\{f(x \mid \theta) : \theta \in \Theta\}$.
- Training data used to estimate the parameters by minimizing some objective function.

The standard statistical setup for modern ML

Machine learning as function-fitting



The traditional setup for estimating parameters in a statistical model (or training a neural network:

- Parameterized set of functions/models $\{f(x \,|\, \theta) : \theta \in \Theta\}$.
- Training data used to estimate the parameters by minimizing some objective function.
- Stochastic optimization algorithm that does the actual minimization.

How should we characterize a model?

Drawing samples from the function space

How should we characterize a model?

Drawing samples from the function space

For a **fixed training set**, **architecture**, and **training algorithm**, we can think of an ML/AI model as a sample from a function space:

How should we characterize a model?

Drawing samples from the function space

For a **fixed training set**, **architecture**, and **training algorithm**, we can think of an ML/AI model as a sample from a function space:

$$\mathscr{F} = \{f : f \text{ representable by the NN}\}$$

How should we characterize a model?

Drawing samples from the function space

For a **fixed training set**, **architecture**, and **training algorithm**, we can think of an ML/AI model as a sample from a function space:

$$\mathscr{F} = \{f : f \text{ representable by the NN}\}$$

Examples:

How should we characterize a model?

Drawing samples from the function space

For a **fixed training set**, **architecture**, and **training algorithm**, we can think of an ML/AI model as a sample from a function space:

$$\mathscr{F} = \{f : f \text{ representable by the NN}\}$$

Examples:

- In classification, each $f : \mathscr{X} \to [L]$ labels input data points.

How should we characterize a model?

Drawing samples from the function space

For a **fixed training set**, **architecture**, and **training algorithm**, we can think of an ML/AI model as a sample from a function space:

$$\mathscr{F} = \{f : f \text{ representable by the NN}\}$$

Examples:

- In classification, each $f : \mathscr{X} \to [L]$ labels input data points.
- In representation learning, each $f : \mathscr{X} \to \mathscr{R}$ maps inputs to representations/embeddings.

Some natural questions

Comparing models is not clear

Some natural questions

Comparing models is not clear

If we have two different models we might have

Some natural questions

Comparing models is not clear

If we have two different models we might have

$$
\begin{aligned}
\mathscr{F} &= \{f : f \text{ representable by NN A}\} \\
\mathscr{G} &= \{g : g \text{ representable by NN B}\}
\end{aligned}
$$

Some natural questions

Comparing models is not clear

If we have two different models we might have

$$\mathscr{F} = \{f : f \text{ representable by NN A}\}$$
$$\mathscr{G} = \{g : g \text{ representable by NN B}\}$$

Can we meaningfully compare these models?

Some natural questions

Comparing models is not clear

If we have two different models we might have

$$\mathscr{F} = \{f : f \text{ representable by NN A}\}$$
$$\mathscr{G} = \{g : g \text{ representable by NN B}\}$$

Can we meaningfully compare these models?

- If $\mathscr{F} = \mathscr{G}$ we can use their outputs to do a comparison.

Some natural questions

Comparing models is not clear

If we have two different models we might have

$$\mathscr{F} = \{f : f \text{ representable by NN A}\}$$
$$\mathscr{G} = \{g : g \text{ representable by NN B}\}$$

Can we meaningfully compare these models?

- If $\mathscr{F} = \mathscr{G}$ we can use their outputs to do a comparison.
- If $\mathscr{F} \neq \mathscr{G}$ we need some way to do a comparison.

Variability in the training process

Is training reliable?

HarmonyOS 4.0

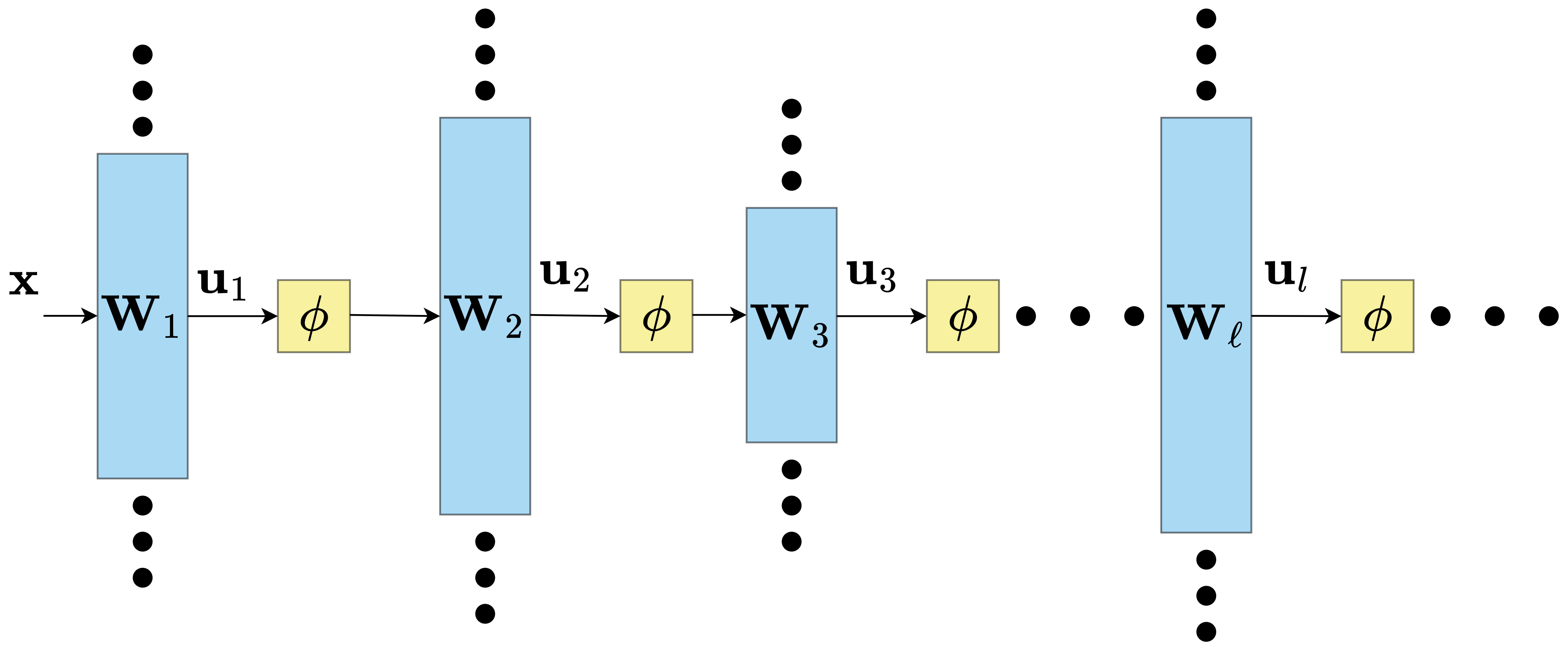
Samsung UI 7.0

If we have two different architectures $\mathscr{F}$ and $\mathscr{G}$ with different output spaces, how can we measure their similarity?

- Focus on **performance**: two models with the same error are "effectively the same".
- Focus on **features**: come up with a mapping from one model to the other to show they are the same.
- Focus on **approximations**: use proxies for each model which are more comparable.

Approximating the NN with a kernel machine

Not practical, but perhaps informative?

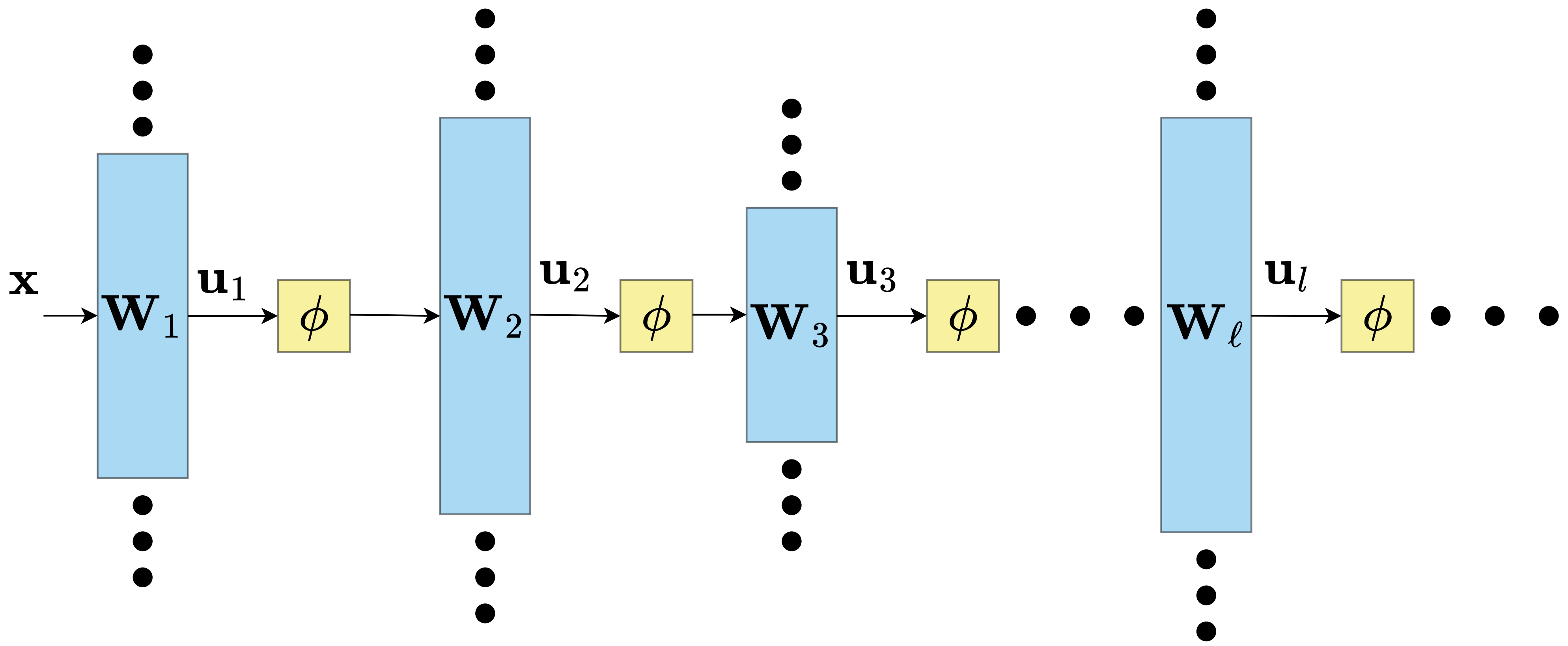


$$\approx$$

$$\text{kGLM}$$

Approximating the NN with a kernel machine

Not practical, but perhaps informative?



$$\approx$$

kGLM

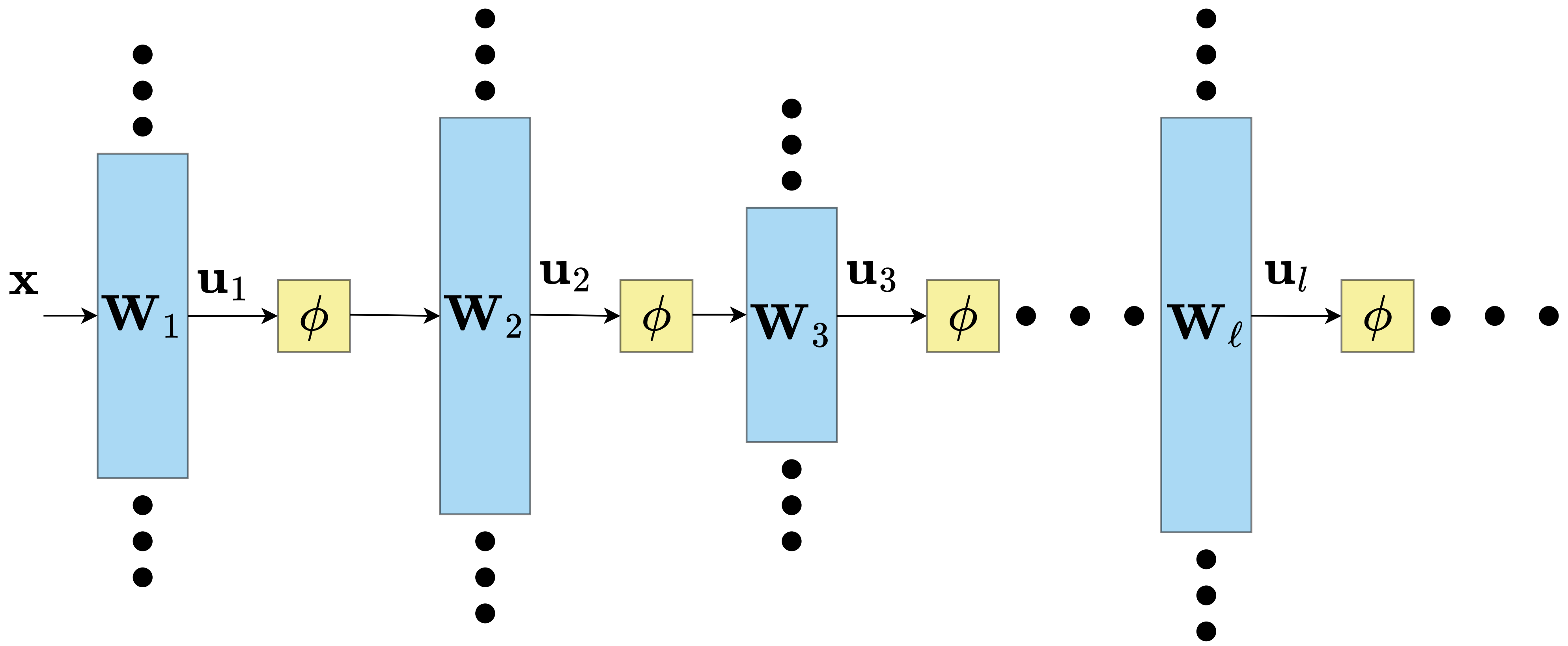
Suppose we compute some kernel function $\mathbf{K}$ associated to the model and fit a **surrogate model** $(\mathbf{V}, \mathbf{b})$:

$$\mathbf{y}_i = \mathbf{V}\mathbf{K}(\mathbf{x}_i, \mathbf{X}) + \mathbf{b}$$

where $\mathbf{y}_i, \mathbf{b} \in \mathbb{R}^C$ and $\mathbf{V} \in \mathbb{R}^{C \times N}$. Fitting is done with the same training data (double dipping).

Approximating the NN with a kernel machine

Not practical, but perhaps informative?



$$\approx$$

$$\text{kGLM}$$

Suppose we compute some kernel function $\mathbf{K}$ associated to the model and fit a **surrogate model** $(\mathbf{V}, \mathbf{b})$:

$$\mathbf{y}_i = \mathbf{V}\mathbf{K}(\mathbf{x}_i, \mathbf{X}) + \mathbf{b}$$

where $\mathbf{y}_i, \mathbf{b} \in \mathbb{R}^C$ and $\mathbf{V} \in \mathbb{R}^{C \times N}$. Fitting is done with the same training data (double dipping).

One example: the **neural tangent kernel**.

Neural Networks as Kernel Machines

Approximating an NN with a “simpler” model

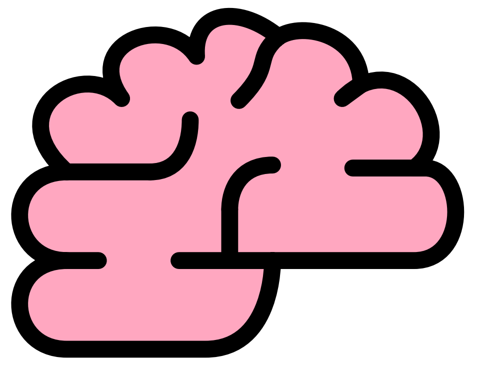
Neural Networks as Kernel Machines

Approximating an NN with a “simpler” model

NTK $\neq$ 🧠 

Neural Networks as Kernel Machines

Approximating an NN with a “simpler” model

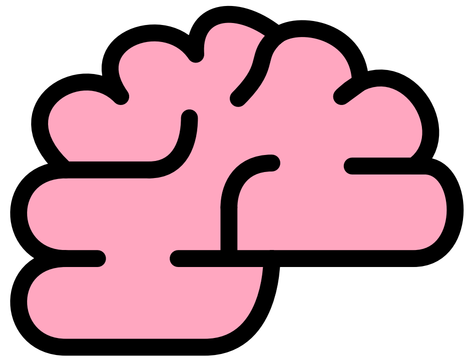
NTK ≠  

Jacot et al. (2018) showed that **infinitely wide** NNs are equivalent to a kernel machine with with the “**neural tangent kernel**” (NTK):

$$K(\mathbf{x}, \mathbf{x}') = \left\langle \nabla_\theta f(\mathbf{x}; \theta), \nabla_\theta f(\mathbf{x}'; \theta) \right\rangle$$

Neural Networks as Kernel Machines

Approximating an NN with a "simpler" model

NTK $\neq$  

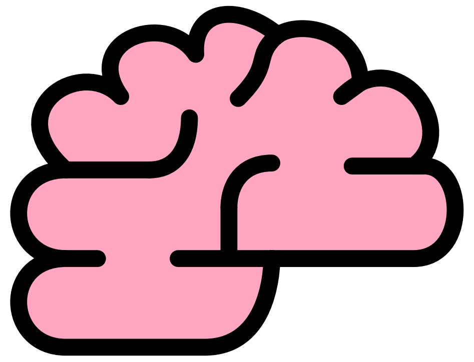
Jacot et al. (2018) showed that **infinitely wide** NNs are equivalent to a kernel machine with with the "**neural tangent kernel**" (NTK):

$$K(\mathbf{x}, \mathbf{x}') = \left\langle \nabla_\theta f(\mathbf{x}; \theta), \nabla_\theta f(\mathbf{x}'; \theta) \right\rangle$$

Measures the **(cosine) similarity between tangent hyperplanes** for $\mathbf{x}$ and $\mathbf{x}'$ at θ.

Neural Networks as Kernel Machines

Approximating an NN with a “simpler” model

NTK $\neq$  

Jacot et al. (2018) showed that **infinitely wide** NNs are equivalent to a kernel machine with with the “**neural tangent kernel**” (NTK):

$$K(\mathbf{x}, \mathbf{x}') = \left\langle \nabla_\theta f(\mathbf{x}; \theta), \nabla_\theta f(\mathbf{x}'; \theta) \right\rangle$$

Measures the **(cosine) similarity between tangent hyperplanes** for $\mathbf{x}$ and $\mathbf{x}'$ at θ.

Finite width networks don’t really behave like infinite width networks… (Chizat et al., 2018; Yang & Hu, 2021; Wang et al., 2022).

Challenge: NTK is asymptotic (infinite width)

Writing an empirical version of the NTK

We would like to handle multi-class problems and large data sets. In the setting **the empirical NTK becomes huge**. For classes i and j define:

$$\mathbf{K}^{\mathrm{NTK}}_{(c,c')}(\mathbf{x}_i, \mathbf{x}_j) = \left\langle \nabla_\theta f^c(\mathbf{x}_i;\theta), \nabla_\theta f^{c'}(\mathbf{x}_j;\theta) \right\rangle$$

Then the NTK has a block structure, where each diagonal block has the "regular" NTK for each class and the off-diagonal blocks are cross terms.

$$\begin{bmatrix} \mathbf{K}^{\mathrm{NTK}}_{1,1} & \mathbf{K}^{\mathrm{NTK}}_{1,2} & \cdots & \mathbf{K}^{\mathrm{NTK}}_{1,C} \\ \mathbf{K}^{\mathrm{NTK}}_{2,1} & \mathbf{K}^{\mathrm{NTK}}_{2,2} & \cdots & \mathbf{K}^{\mathrm{NTK}}_{2,C} \\ \vdots & & \ddots & \vdots \\ \mathbf{K}^{\mathrm{NTK}}_{C,1} & \mathbf{K}^{\mathrm{NTK}}_{C,2} & & \mathbf{K}^{\mathrm{NTK}}_{C,C} \end{bmatrix}$$

Trace NTK: a proxy for the eNTK

Much lower computational overhead needed

We look at a simplification of the NTK:

$$\mathbf{K}^{\text{trNTK}}(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{c=1}^{C} \left\langle \nabla_\theta f^c(\mathbf{x}_i; \theta), \nabla_\theta f^c(\mathbf{x}_j; \theta) \right\rangle}{\left(\sum_{c=1}^{C} \| f^c(\mathbf{x}_i; \theta) \|^2 \right)^{1/2} \left(\sum_{c=1}^{C} \| f^c(\mathbf{x}_j; \theta) \|^2 \right)^{1/2}}$$

This is **different from other surrogate kernels:** the pseudo NTK (pNTK) (Mohamadi & Sutherland, 2022), things based on the CK (Fan & Wang, 2020; Yeh et al., 2018), the un-normalized trNTK, and the embedding kernel (Akyürek et al., 2023).

Fast to compute, also with random projections (Novak et al., 2022, Park et al., 2023))

Some takeaways from the setup

Formalization shows how under-specified model comparison is

$$\begin{Bmatrix} \mathbf{K}^{\mathrm{NTK}}_{1,1} & \mathbf{K}^{\mathrm{NTK}}_{1,2} & \cdots & \mathbf{K}^{\mathrm{NTK}}_{1,C} \\ \mathbf{K}^{\mathrm{NTK}}_{2,1} & \mathbf{K}^{\mathrm{NTK}}_{2,2} & \cdots & \mathbf{K}^{\mathrm{NTK}}_{2,C} \\ \vdots & & \ddots & \vdots \\ \mathbf{K}^{\mathrm{NTK}}_{C,1} & \mathbf{K}^{\mathrm{NTK}}_{C,2} & & \mathbf{K}^{\mathrm{NTK}}_{C,C} \end{Bmatrix}$$

Some takeaways from the setup

Formalization shows how under-specified model comparison is

Understanding a model by its NTK sounds OK but can we really compare two models by their NTKs? Maybe!

$$\begin{bmatrix} \mathbf{K}^{\mathrm{NTK}}_{1,1} & \mathbf{K}^{\mathrm{NTK}}_{1,2} & \cdots & \mathbf{K}^{\mathrm{NTK}}_{1,C} \\ \mathbf{K}^{\mathrm{NTK}}_{2,1} & \mathbf{K}^{\mathrm{NTK}}_{2,2} & \cdots & \mathbf{K}^{\mathrm{NTK}}_{2,C} \\ \vdots & & \ddots & \vdots \\ \mathbf{K}^{\mathrm{NTK}}_{C,1} & \mathbf{K}^{\mathrm{NTK}}_{C,2} & & \mathbf{K}^{\mathrm{NTK}}_{C,C} \end{bmatrix}$$

Some takeaways from the setup

Formalization shows how under-specified model comparison is

Understanding a model by its NTK sounds OK but can we really compare two models by their NTKs? Maybe!

- Computing even the trNTK is expensive, especially for large models.

$$\begin{Bmatrix} \mathbf{K}^{\mathrm{NTK}}_{1,1} & \mathbf{K}^{\mathrm{NTK}}_{1,2} & \cdots & \mathbf{K}^{\mathrm{NTK}}_{1,C} \\ \mathbf{K}^{\mathrm{NTK}}_{2,1} & \mathbf{K}^{\mathrm{NTK}}_{2,2} & \cdots & \mathbf{K}^{\mathrm{NTK}}_{2,C} \\ \vdots & & \ddots & \vdots \\ \mathbf{K}^{\mathrm{NTK}}_{C,1} & \mathbf{K}^{\mathrm{NTK}}_{C,2} & & \mathbf{K}^{\mathrm{NTK}}_{C,C} \end{Bmatrix}$$

Some takeaways from the setup

Formalization shows how under-specified model comparison is

Understanding a model by its NTK sounds OK but can we really compare two models by their NTKs? Maybe!

- Computing even the trNTK is expensive, especially for large models.
- Much easier if you have access to the training corpora.

$$\begin{Bmatrix} \mathbf{K}^{\mathrm{NTK}}_{1,1} & \mathbf{K}^{\mathrm{NTK}}_{1,2} & \cdots & \mathbf{K}^{\mathrm{NTK}}_{1,C} \\ \mathbf{K}^{\mathrm{NTK}}_{2,1} & \mathbf{K}^{\mathrm{NTK}}_{2,2} & \cdots & \mathbf{K}^{\mathrm{NTK}}_{2,C} \\ \vdots & & \ddots & \vdots \\ \mathbf{K}^{\mathrm{NTK}}_{C,1} & \mathbf{K}^{\mathrm{NTK}}_{C,2} & & \mathbf{K}^{\mathrm{NTK}}_{C,C} \end{Bmatrix}$$

Some takeaways from the setup

Formalization shows how under-specified model comparison is

Understanding a model by its NTK sounds OK but can we really compare two models by their NTKs? Maybe!

- Computing even the trNTK is expensive, especially for large models.
- Much easier if you have access to the training corpora.
- Challenging because of invariants.

$$\begin{Bmatrix} \mathbf{K}^{\mathrm{NTK}}_{1,1} & \mathbf{K}^{\mathrm{NTK}}_{1,2} & \cdots & \mathbf{K}^{\mathrm{NTK}}_{1,C} \\ \mathbf{K}^{\mathrm{NTK}}_{2,1} & \mathbf{K}^{\mathrm{NTK}}_{2,2} & \cdots & \mathbf{K}^{\mathrm{NTK}}_{2,C} \\ \vdots & & \ddots & \vdots \\ \mathbf{K}^{\mathrm{NTK}}_{C,1} & \mathbf{K}^{\mathrm{NTK}}_{C,2} & & \mathbf{K}^{\mathrm{NTK}}_{C,C} \end{Bmatrix}$$

Embedding spaces and model comparisons

Rm Palaniappan, *Alien Planet-B*
Viscosity, pencil colour and ink on handmade paper

A question of interoperability

Challenges in collaborating with AI instruments

HarmonyOS 4.0

Samsung UI 7.0

A question of interoperability

Challenges in collaborating with AI instruments

HarmonyOS 4.0

Samsung UI 7.0

Suppose we have two manufacturers of these **AI scientific instruments** based on generative AI.

A question of interoperability

Challenges in collaborating with AI instruments

HarmonyOS 4.0

Samsung UI 7.0

Suppose we have two manufacturers of these **AI scientific instruments** based on generative AI.

You wan to collaborate with a lab which has a **different model than you do**.

A question of interoperability

Challenges in collaborating with AI instruments

HarmonyOS 4.0

Samsung UI 7.0

Suppose we have two manufacturers of these **AI scientific instruments** based on generative AI.

You wan to collaborate with a lab which has a **different model than you do**.

Are these models producing outputs that **“look the same?”**

A question of interoperability

Challenges in collaborating with AI instruments

HarmonyOS 4.0

Samsung UI 7.0

Suppose we have two manufacturers of these **AI scientific instruments** based on generative AI.

You wan to collaborate with a lab which has a **different model than you do**.

Are these models producing outputs that **“look the same?”**

Challenges:

A question of interoperability

Challenges in collaborating with AI instruments

HarmonyOS 4.0

Samsung UI 7.0

Suppose we have two manufacturers of these **AI scientific instruments** based on generative AI.

You wan to collaborate with a lab which has a **different model than you do**.

Are these models producing outputs that **“look the same?”**

Challenges:

- To the human eye, they are functionally similar.

A question of interoperability

Challenges in collaborating with AI instruments

HarmonyOS 4.0

Samsung UI 7.0

Suppose we have two manufacturers of these **AI scientific instruments** based on generative AI.

You wan to collaborate with a lab which has a **different model than you do**.

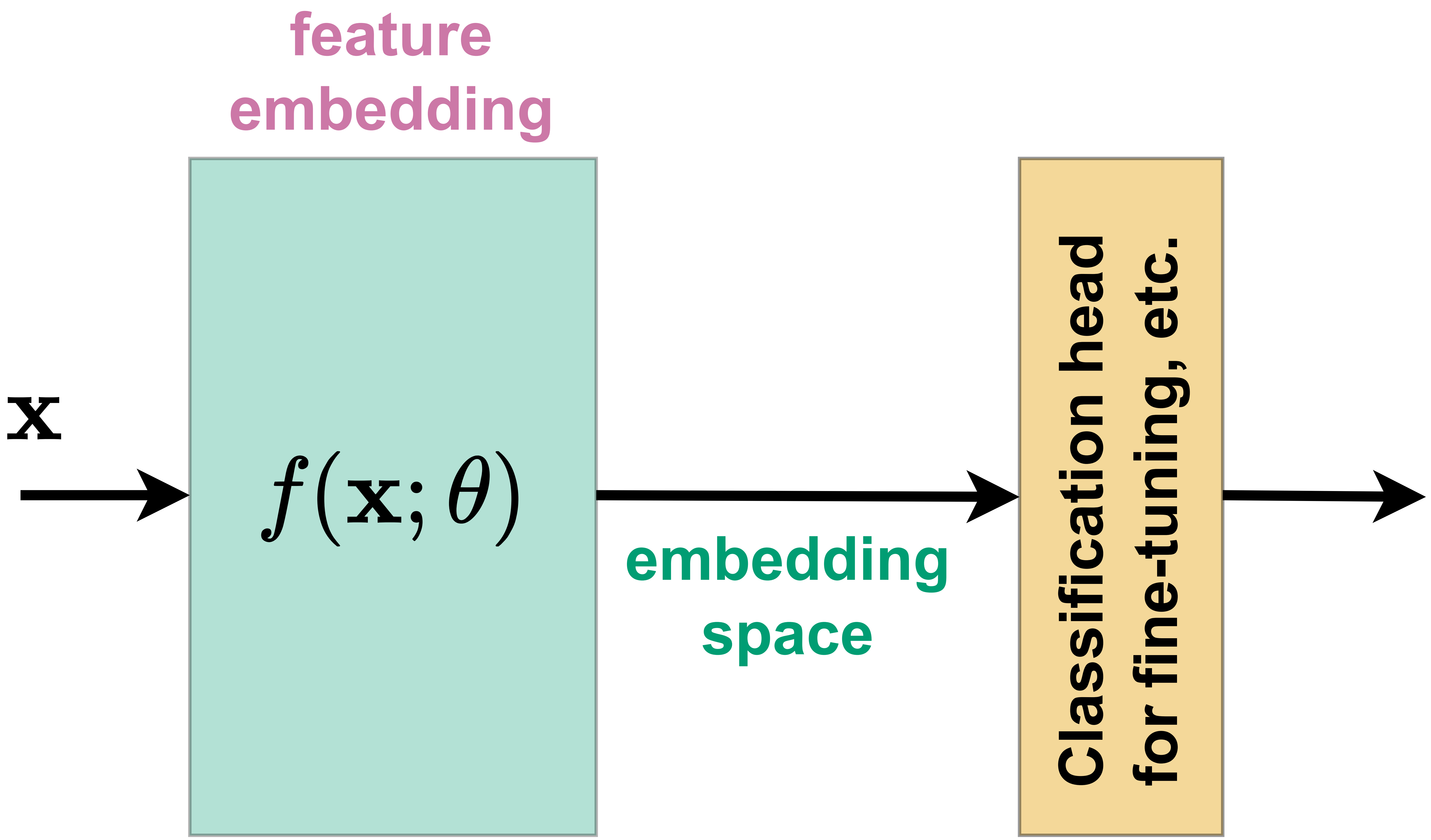
Are these models producing outputs that **“look the same?”**

Challenges:

- To the human eye, they are functionally similar.
- Can we quantitatively see if they are different?

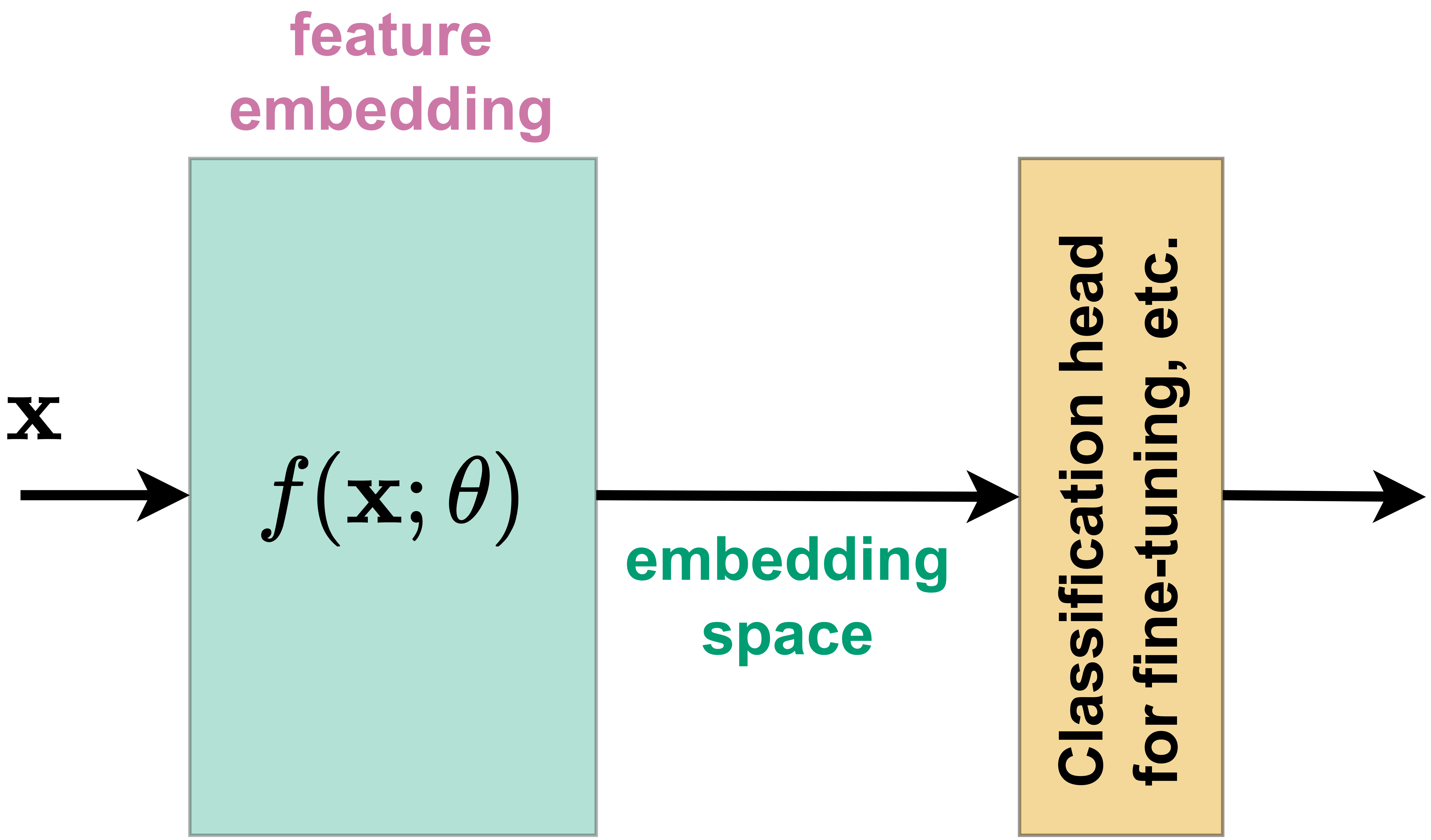
Embedding spaces of large models

Splitting a model into a feature extraction and decision



Embedding spaces of large models

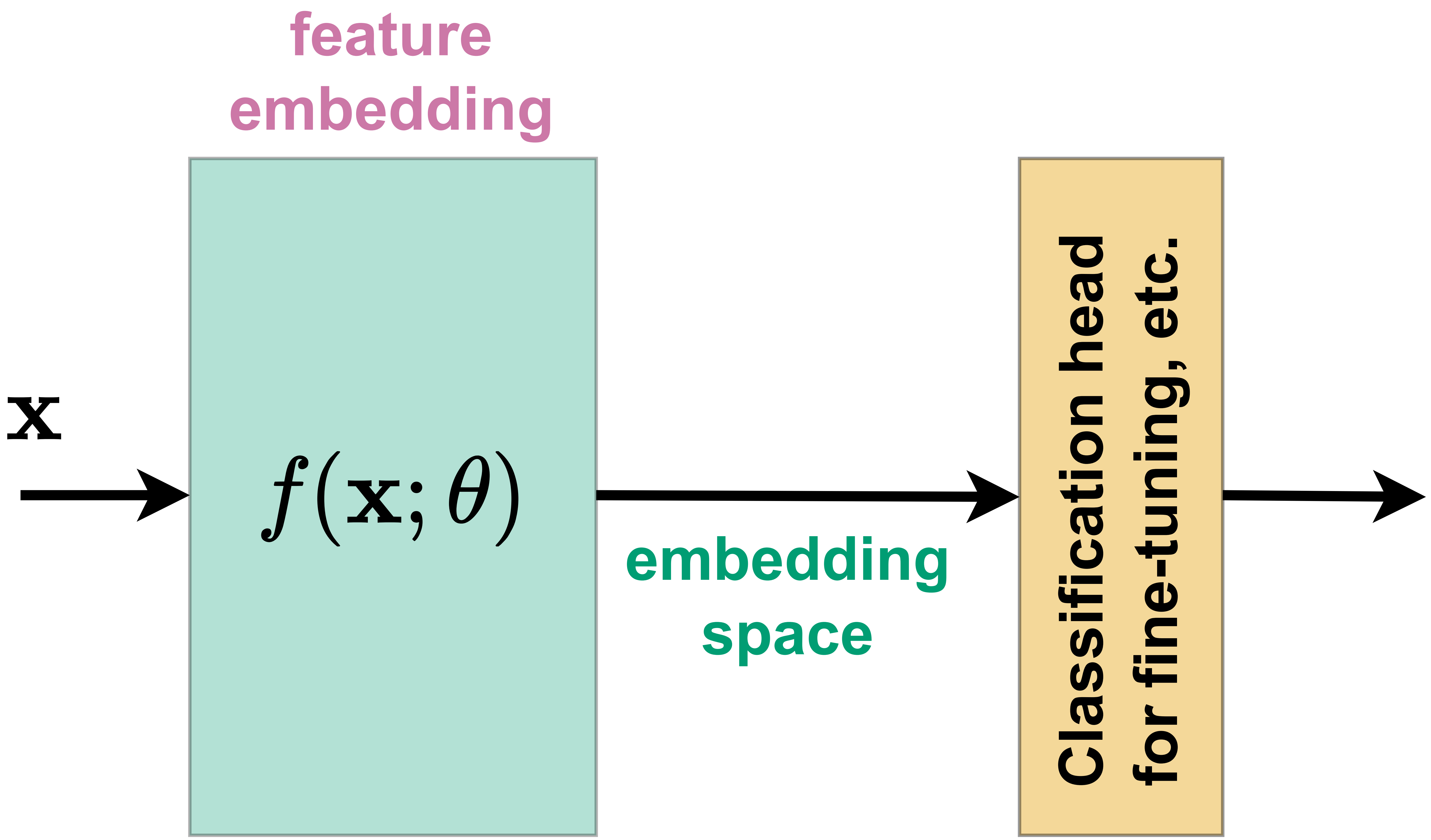
Splitting a model into a feature extraction and decision



We can think of many models as having "feature embedding" stage followed by "downstream tasks."

Embedding spaces of large models

Splitting a model into a feature extraction and decision

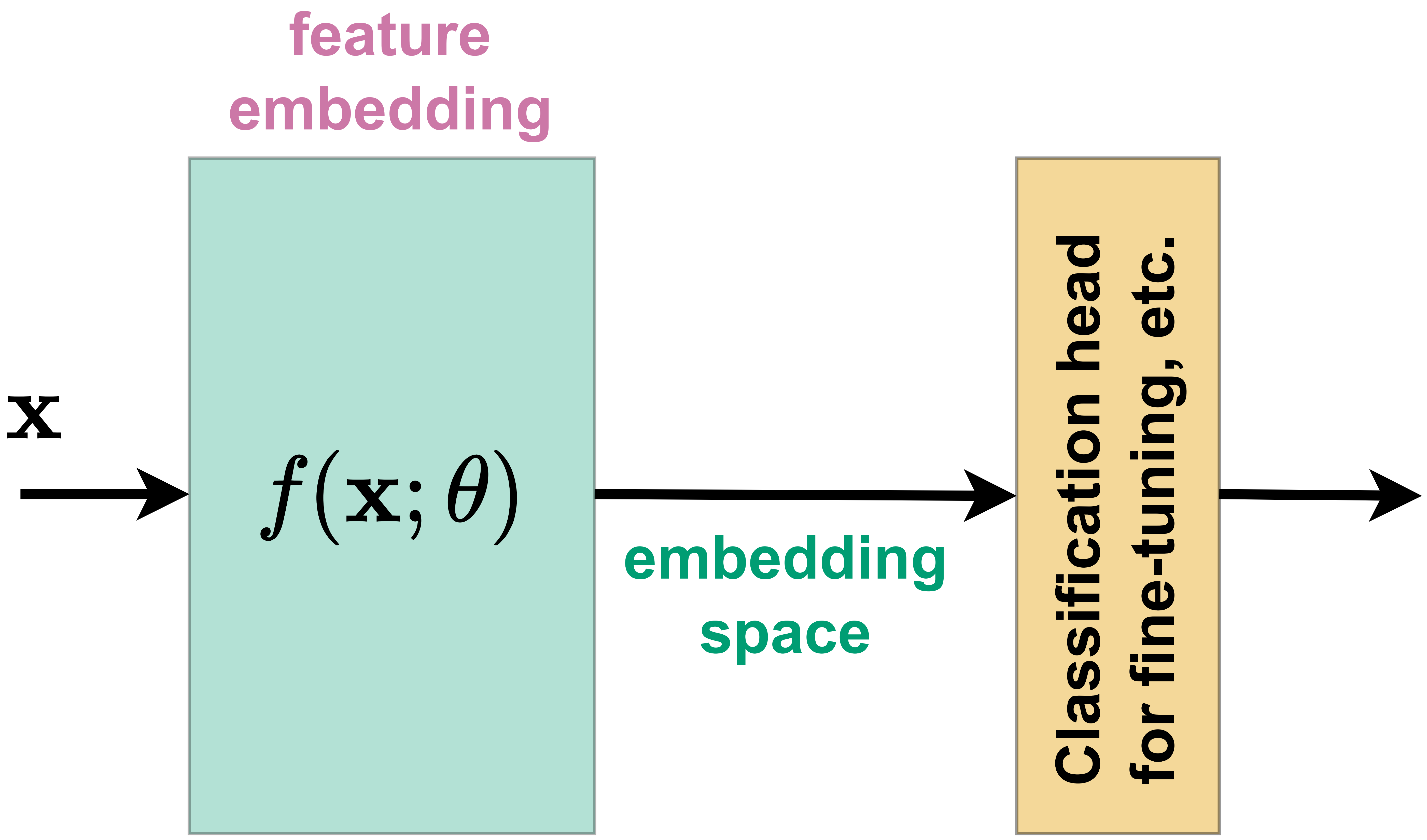


We can think of many models as having “feature embedding” stage followed by “downstream tasks.”

Fine-tuning works because these embeddings carry a lot of information.

Embedding spaces of large models

Splitting a model into a feature extraction and decision



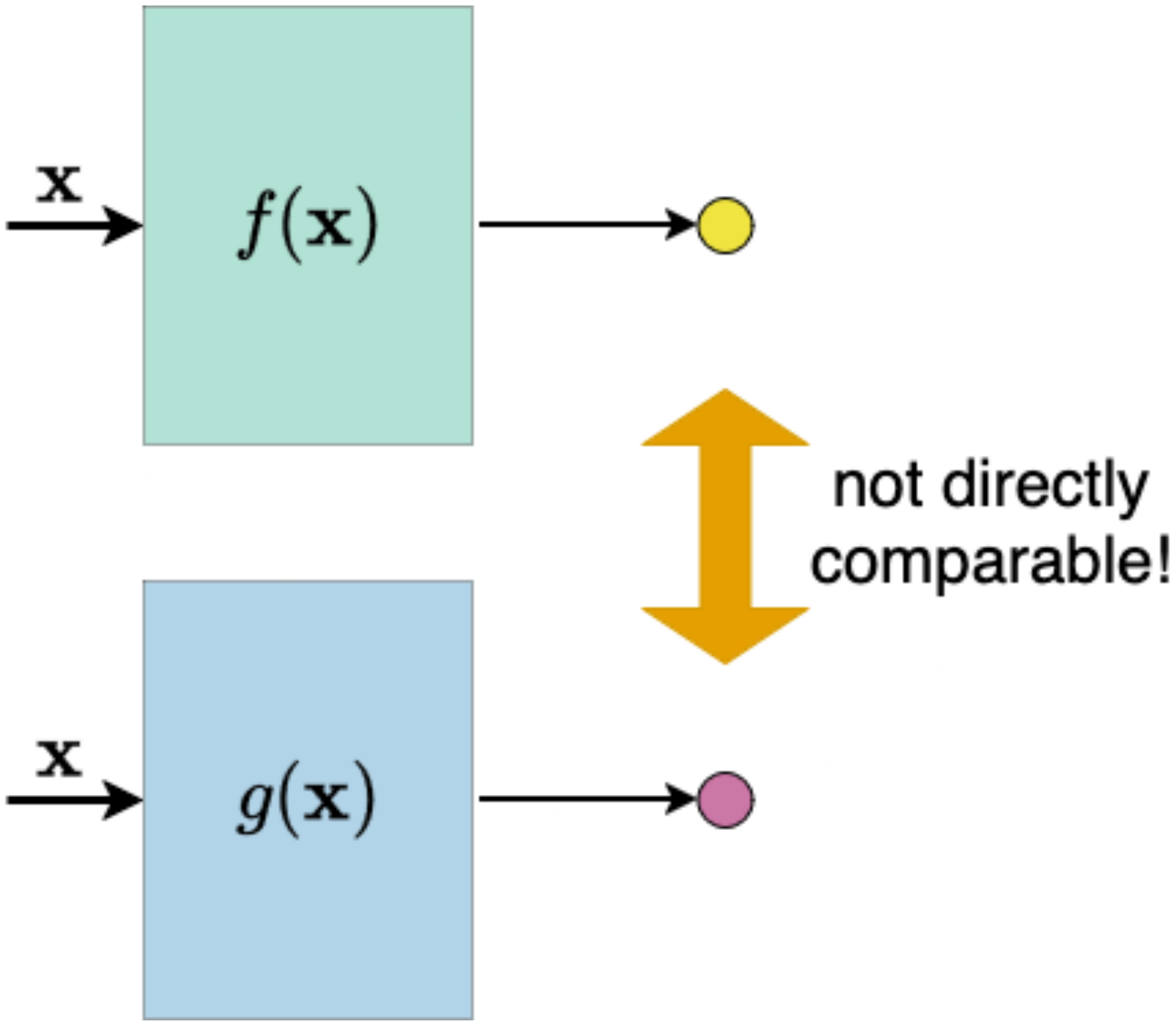
We can think of many models as having "feature embedding" stage followed by "downstream tasks."

Fine-tuning works because these embeddings carry a lot of information.

Idea: can we compare the embedding spaces of models to tell the difference between them?

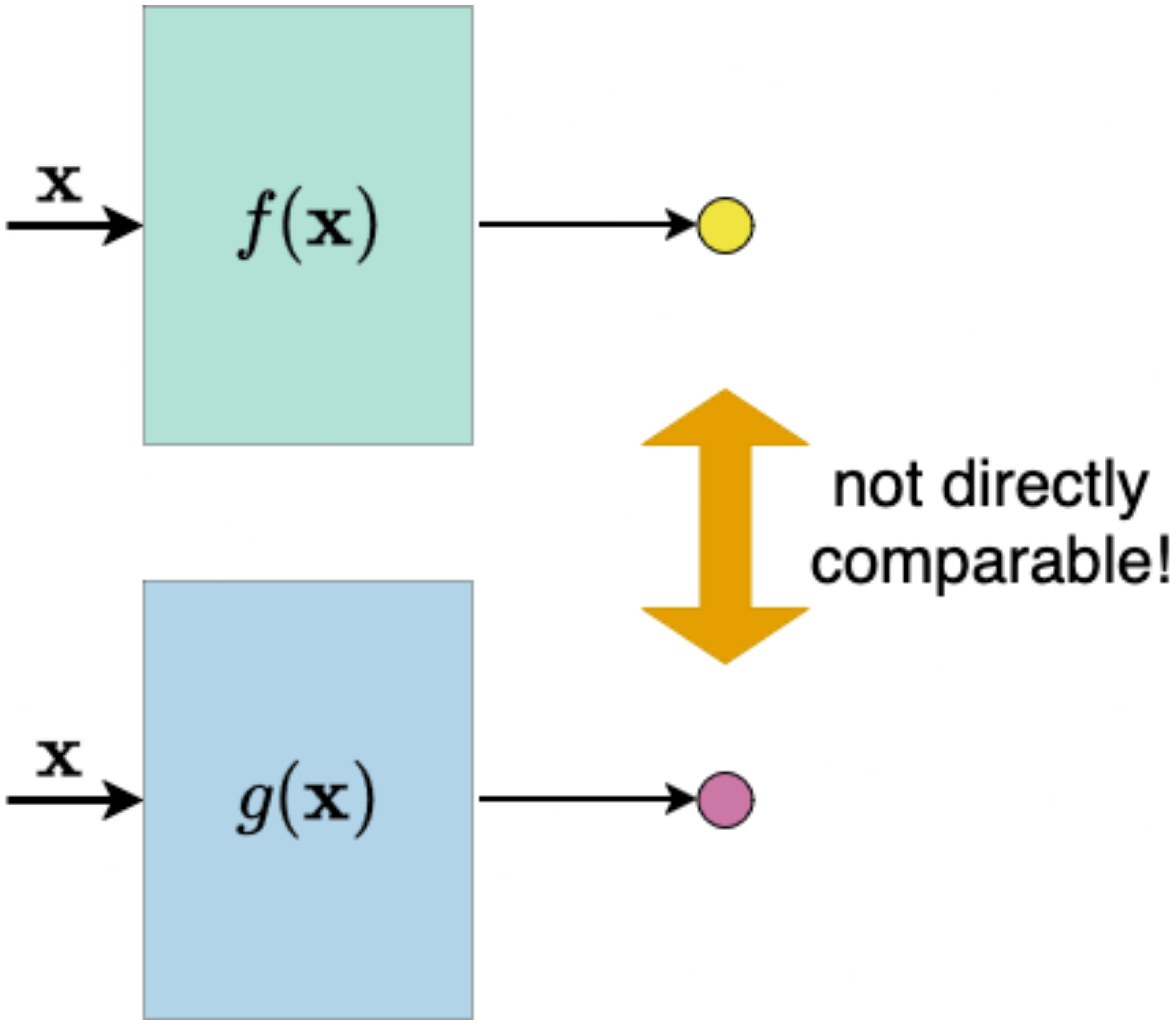
Comparing embedding spaces directly?

Generally this is a non-starter



Comparing embedding spaces directly?

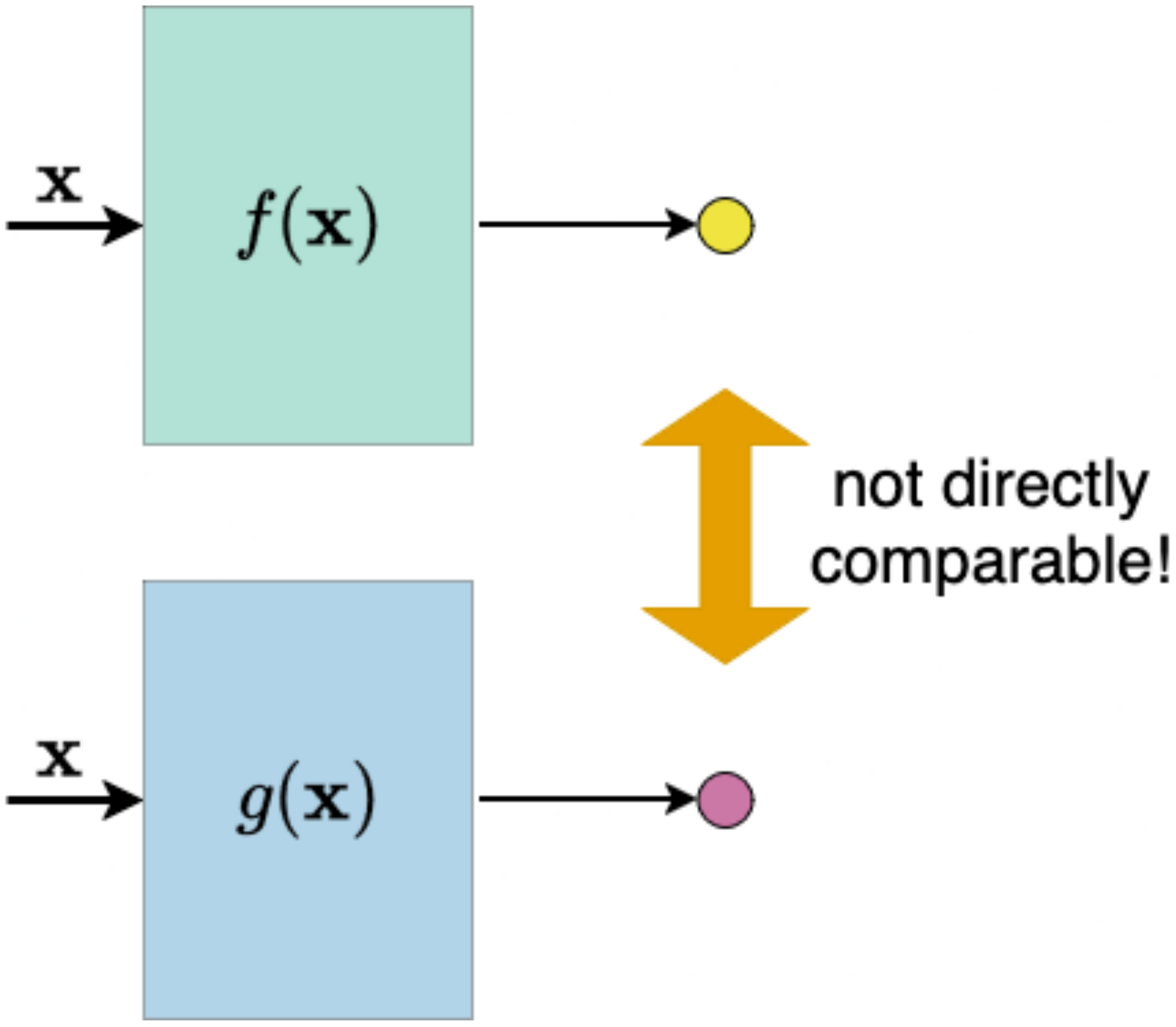
Generally this is a non-starter



Given two models with different architectures, we cannot compare the embedding spaces directly.

Comparing embedding spaces directly?

Generally this is a non-starter

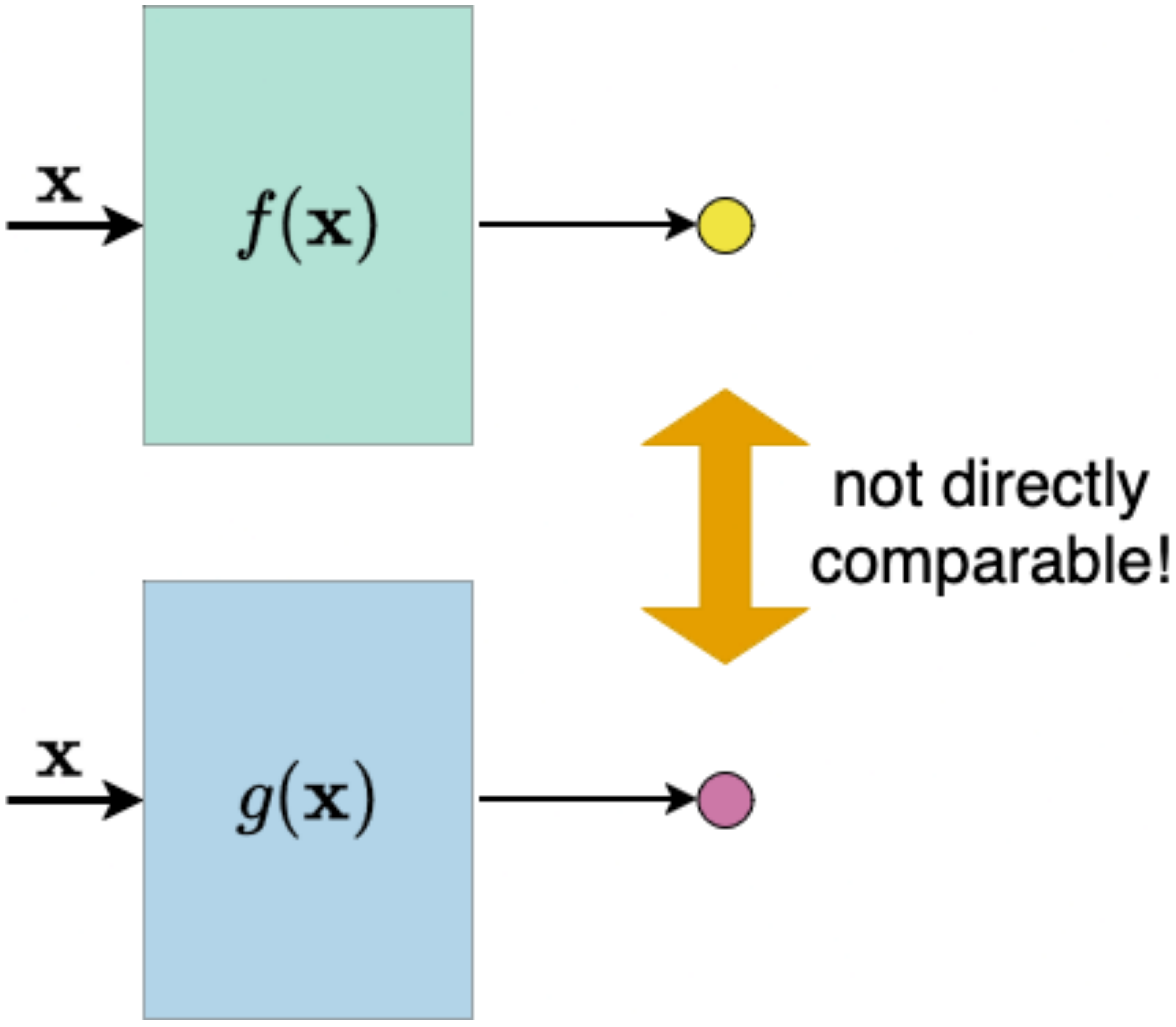


Given two models with different architectures, we cannot compare the embedding spaces directly.

- Different **dimensions**.

Comparing embedding spaces directly?

Generally this is a non-starter

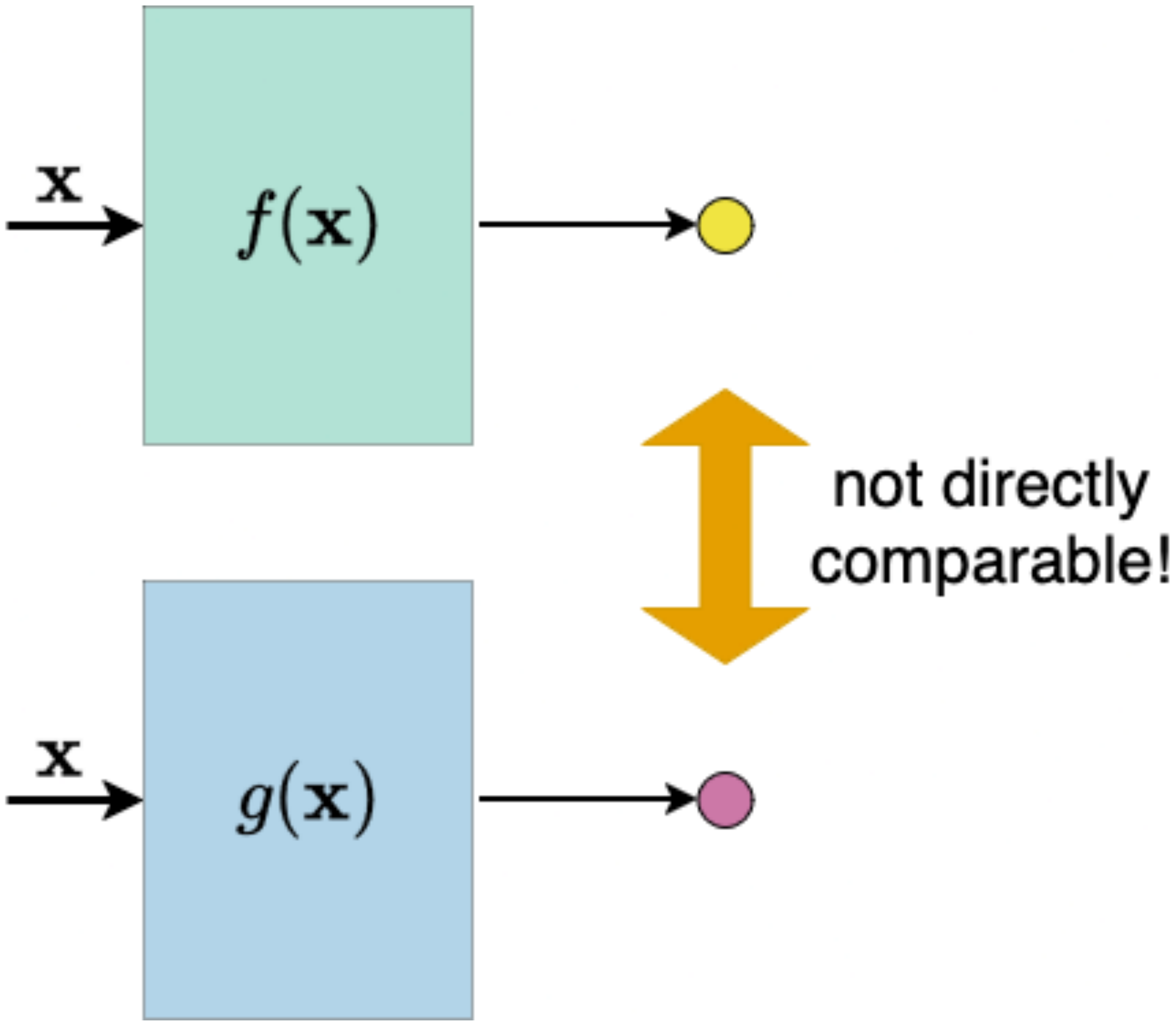


Given two models with different architectures, we cannot compare the embedding spaces directly.

- Different **dimensions**.
- Different **compression strategies**

Comparing embedding spaces directly?

Generally this is a non-starter

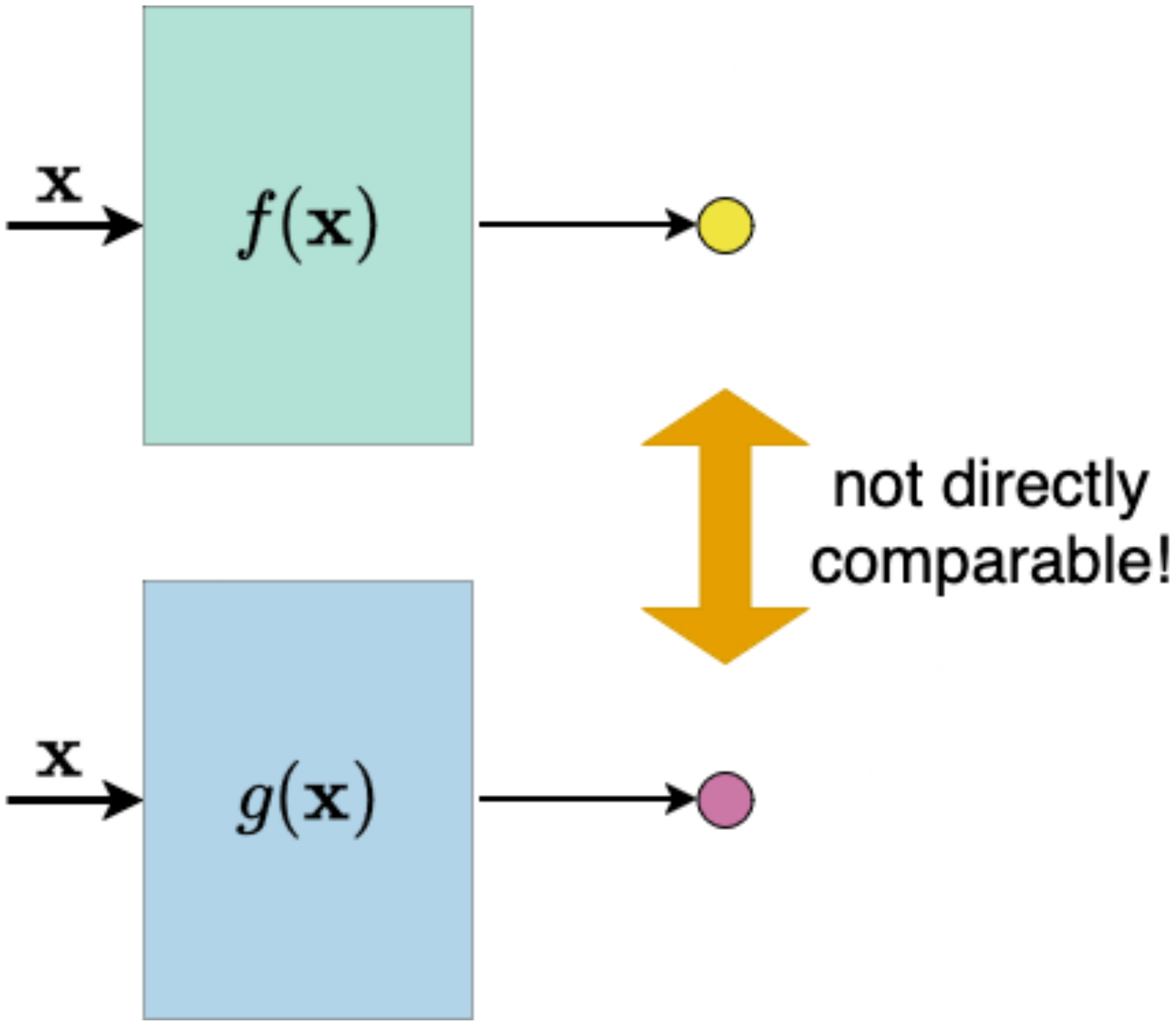


Given two models with different architectures, we cannot compare the embedding spaces directly.

- Different **dimensions**.
- Different **compression strategies**
- Different **“semantics”**

Comparing embedding spaces directly?

Generally this is a non-starter



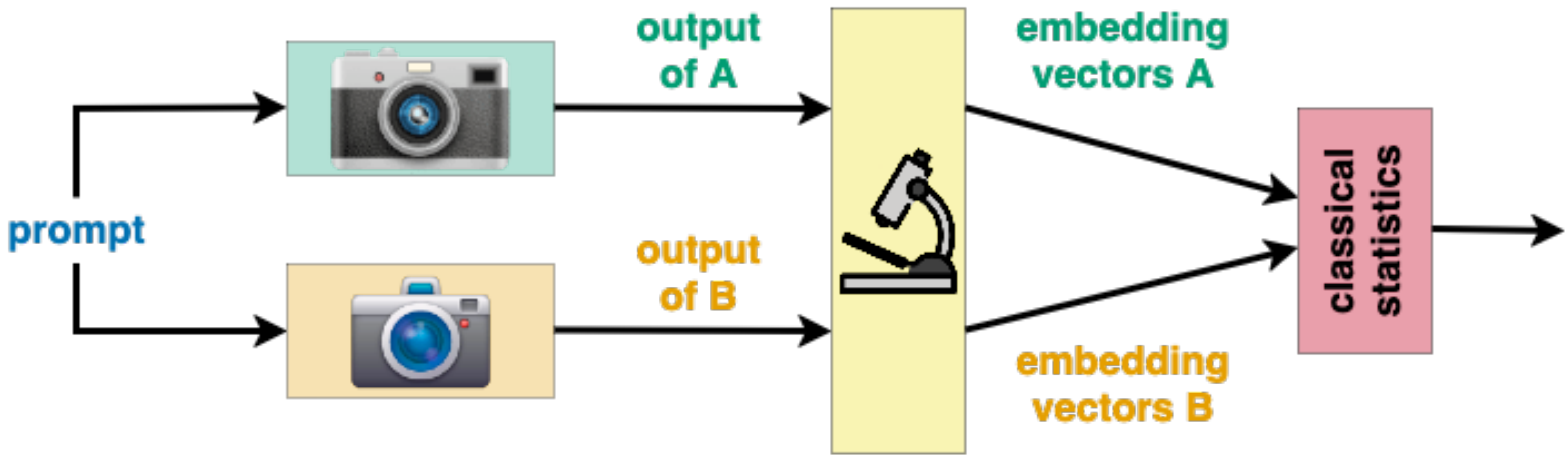
Given two models with different architectures, we cannot compare the embedding spaces directly.

- Different **dimensions**.
- Different **compression strategies**
- Different **"semantics"**

Unlike with classification, we need to compare the outputs of the generative models.

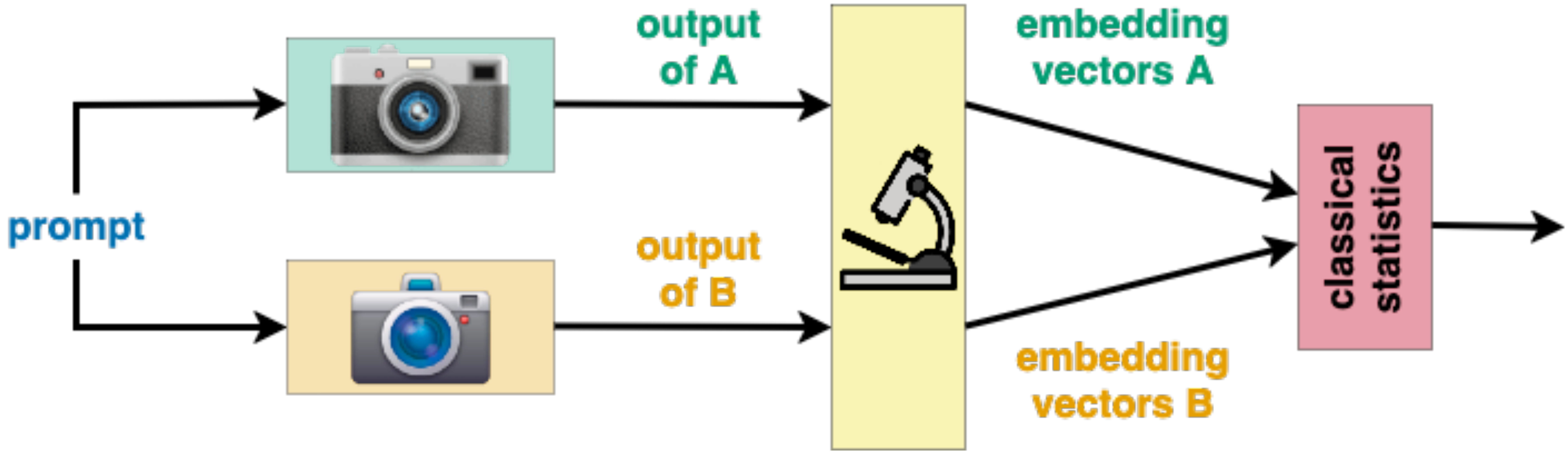
Using a pre-trained model to distinguish

Use the embedding space to compare outputs of models



Using a pre-trained model to distinguish

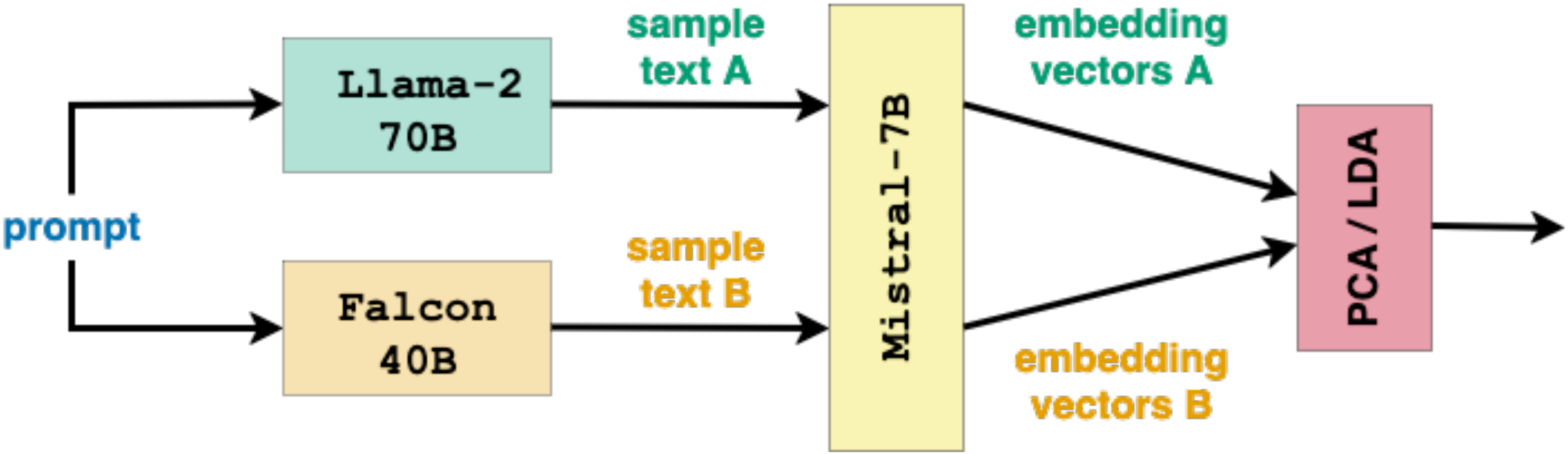
Use the embedding space to compare outputs of models



New idea: use the embedding space of a **third AI model** as a “microscope” to compare the outputs of two AI models.s

A specific example for GenAI

Compare the outputs using a 3rd model for embedding



Using a large model as an instrument

It takes one to know one

Using a large model as an instrument

It takes one to know one

Idea: Use a large model to embed the outputs of the models we want to compare.

Using a large model as an instrument

It takes one to know one

Idea: Use a large model to embed the outputs of the models we want to compare.

- `Mistral-7B`: LLM, transformer-based, 32 layers, 13b parameters per token and 32 token vocabulary. Embeddings from the final hidden layer of dimension 4,096.

Using a large model as an instrument

It takes one to know one

Idea: Use a large model to embed the outputs of the models we want to compare.

- **`Mistral-7B`**: LLM, transformer-based, 32 layers, 13b parameters per token and 32 token vocabulary. Embeddings from the final hidden layer of dimension 4,096.
- **`Multilingual-e5-large`**: extracts sentence embeddings from text in different languages to 1024-dimensional embedding vectors. 60M parameters, context window of 512 tokens and long text is truncated to fit within this window.

Using a large model as an instrument

It takes one to know one

Idea: Use a large model to embed the outputs of the models we want to compare.

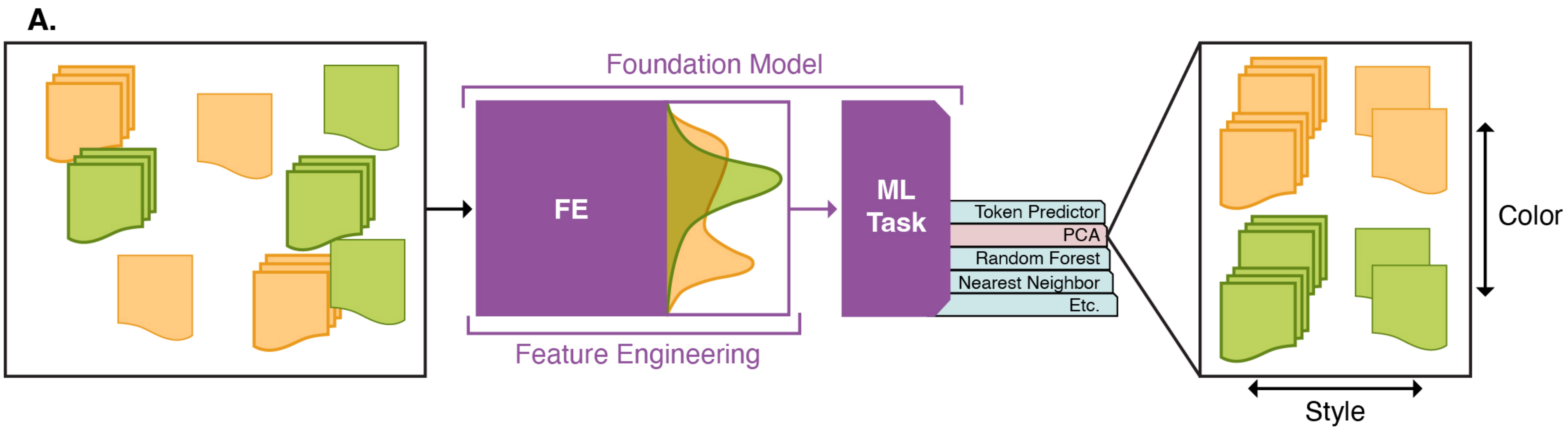
- **`Mistral-7B`**: LLM, transformer-based, 32 layers, 13b parameters per token and 32 token vocabulary. Embeddings from the final hidden layer of dimension 4,096.

- **`Multilingual-e5-large`**: extracts sentence embeddings from text in different languages to 1024-dimensional embedding vectors. 60M parameters, context window of 512 tokens and long text is truncated to fit within this window.

- **`Data Filtering Network`**: a CLIP model trained on 5B images that were filtered from an uncurated dataset of image-text pairs. It has 1B parameters and can be used to encode both text and images.

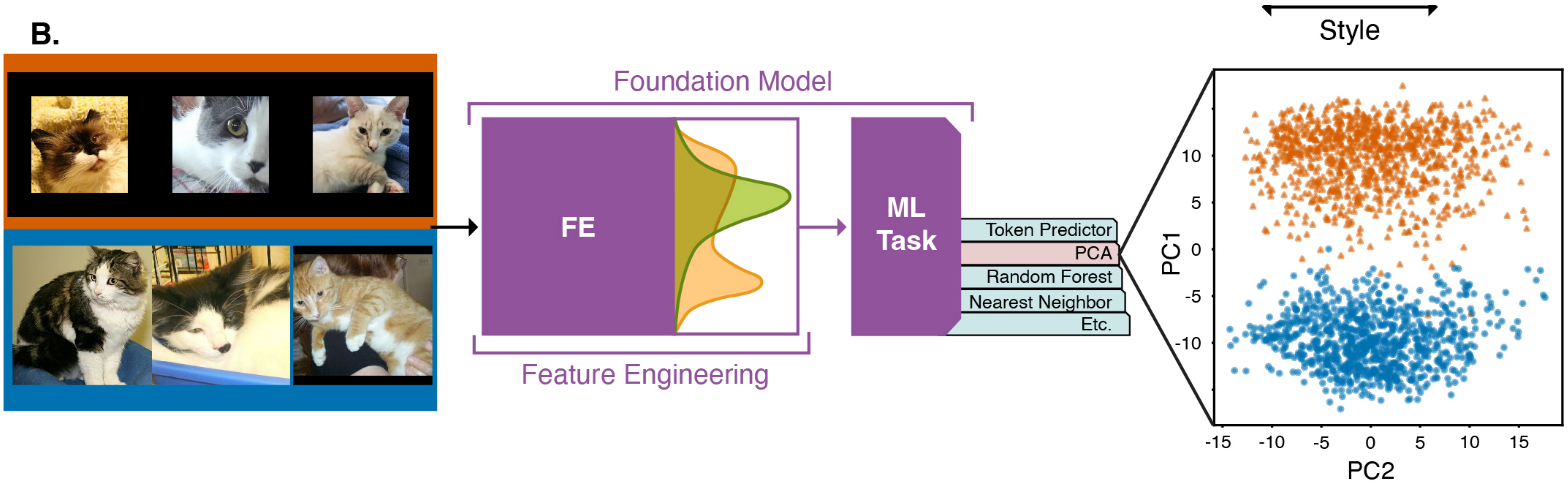
The generic approach in different contexts

The structure is similar, but the models are different



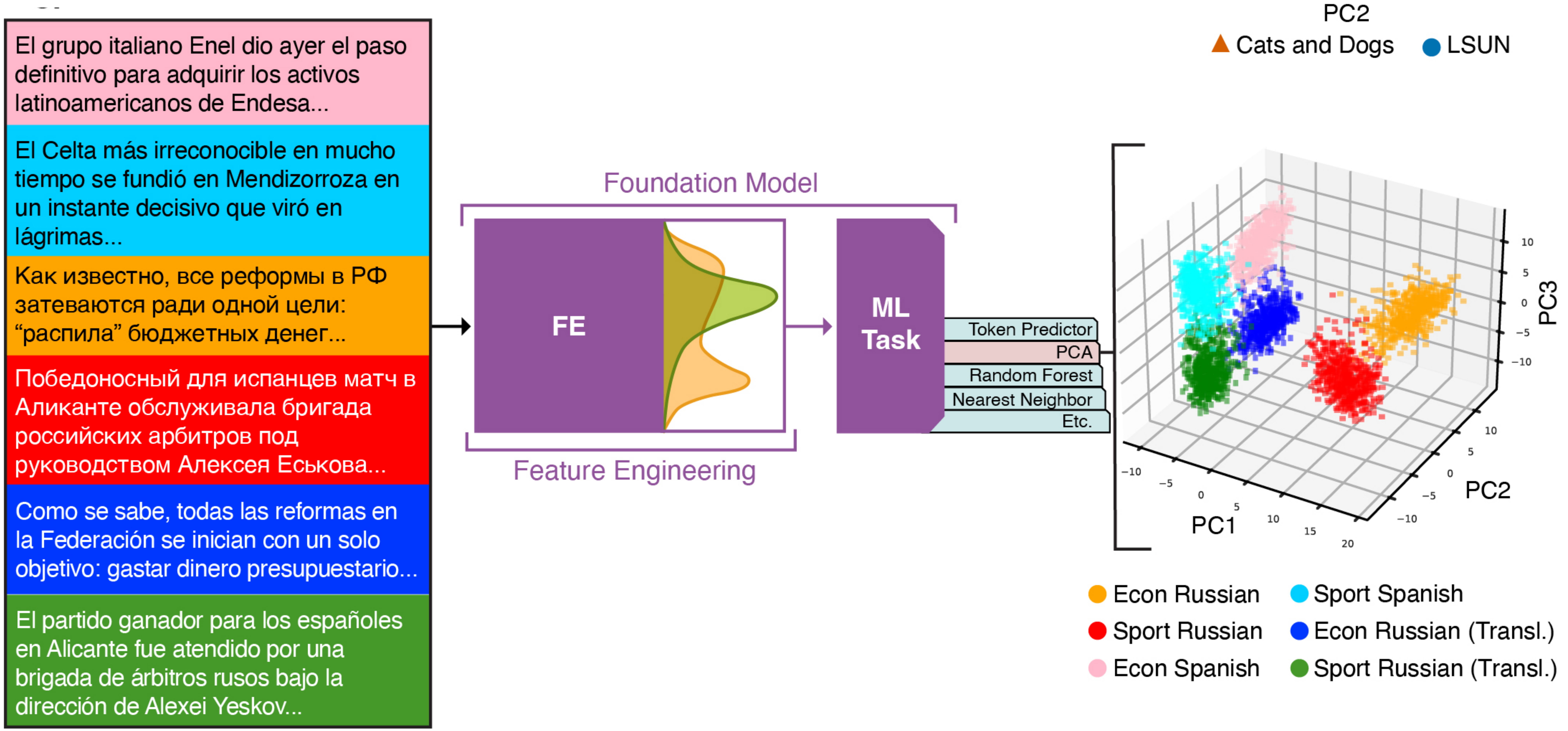
The generic approach in different contexts

The structure is similar, but the models are different



The generic approach in different contexts

The structure is similar, but the models are different



A pre-trained model is a kind of instrument

But used to distinguish between other models

A pre-trained model is a kind of instrument

But used to distinguish between other models

Simple tools (PCA, LDA) applied to the embedding vectors reveal differences between samples generated by other models. Some applications:

A pre-trained model is a kind of instrument

But used to distinguish between other models

Simple tools (PCA, LDA) applied to the embedding vectors reveal differences between samples generated by other models. Some applications:

1. Embed real data and AI-generated data to see if the embedding vectors cluster.

A pre-trained model is a kind of instrument

But used to distinguish between other models

Simple tools (PCA, LDA) applied to the embedding vectors reveal differences between samples generated by other models. Some applications:

1. Embed real data and AI-generated data to see if the embedding vectors cluster.

2. Unsupervised clustering of embedded data recreates the labels in the original.

A pre-trained model is a kind of instrument

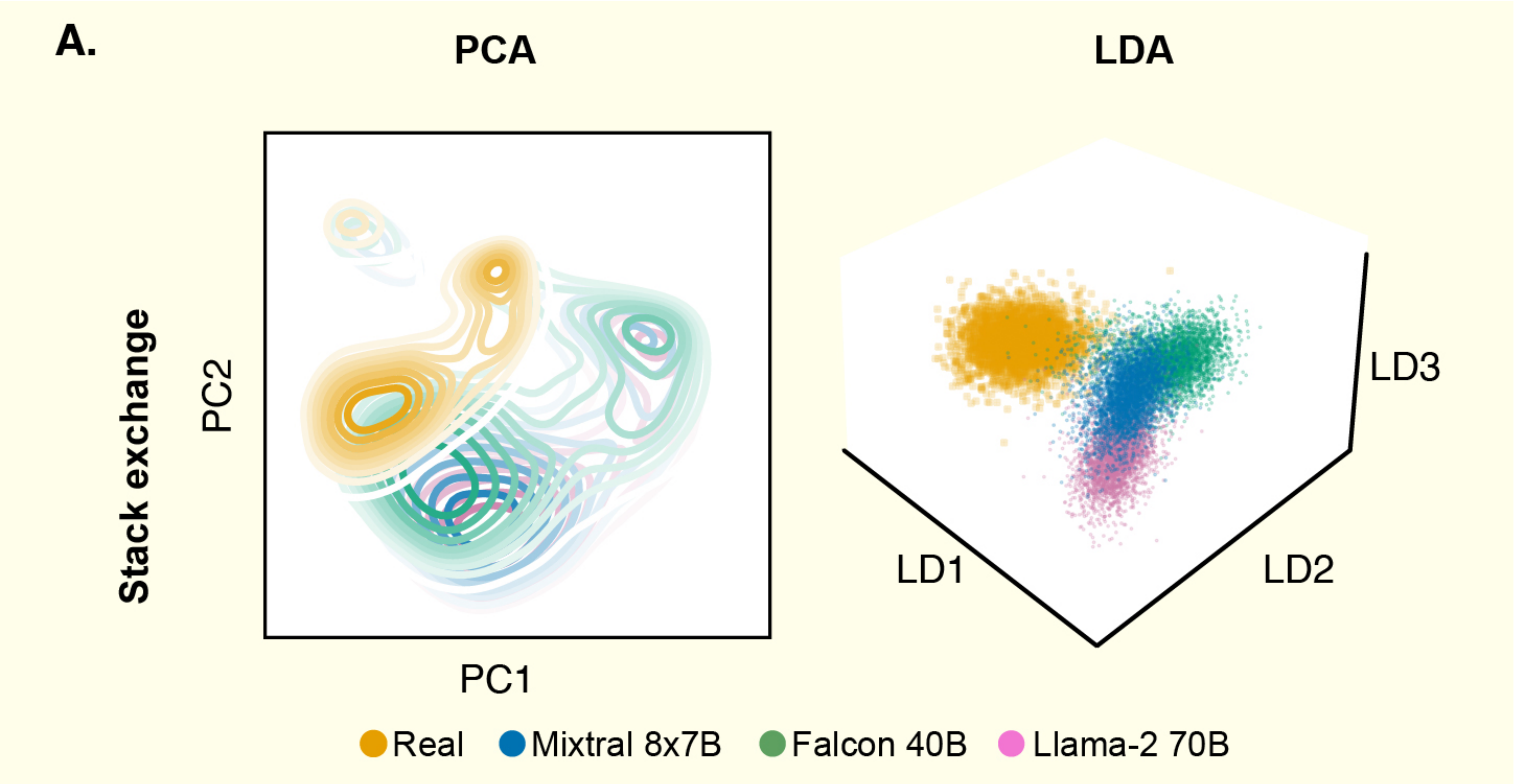
But used to distinguish between other models

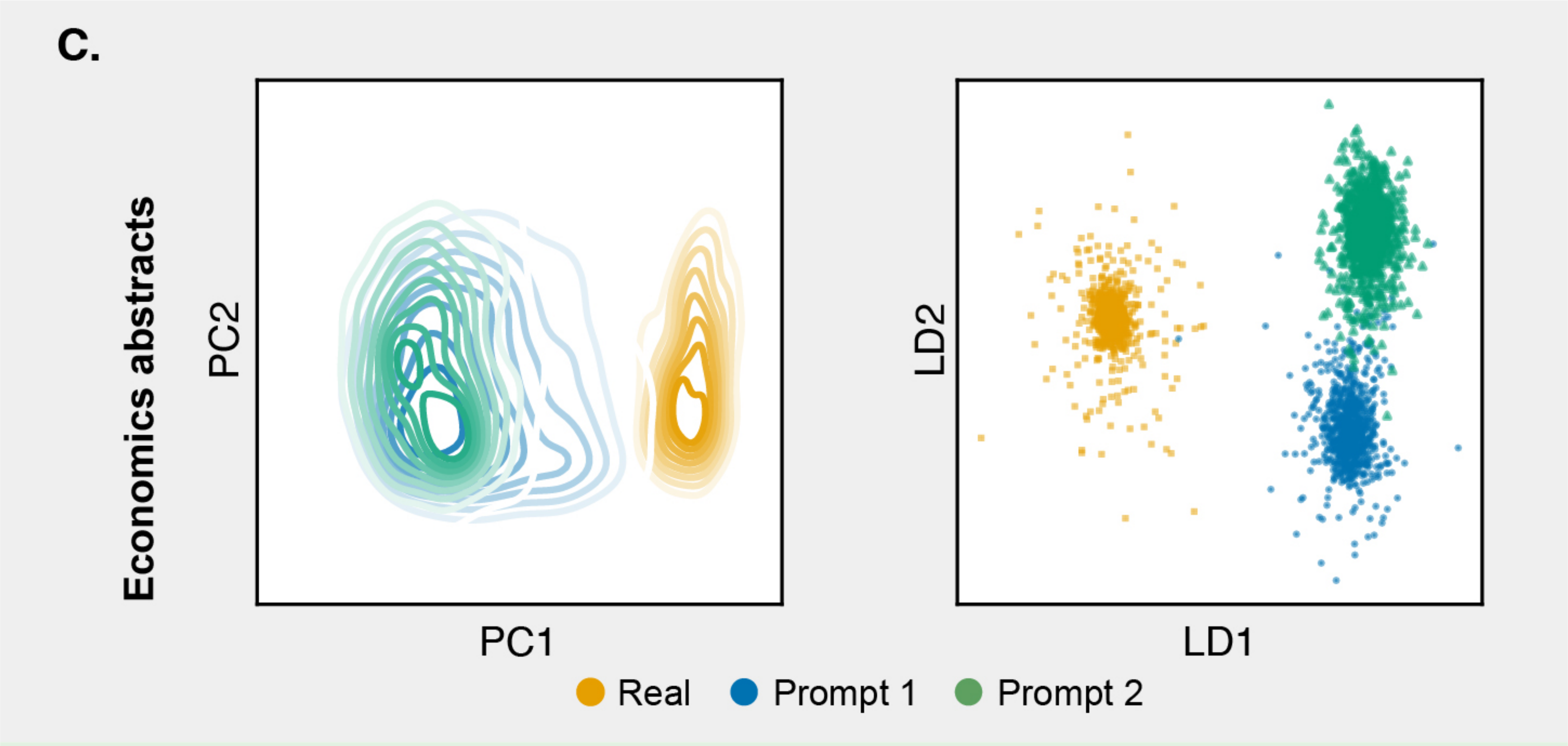
Simple tools (PCA, LDA) applied to the embedding vectors reveal differences between samples generated by other models. Some applications:

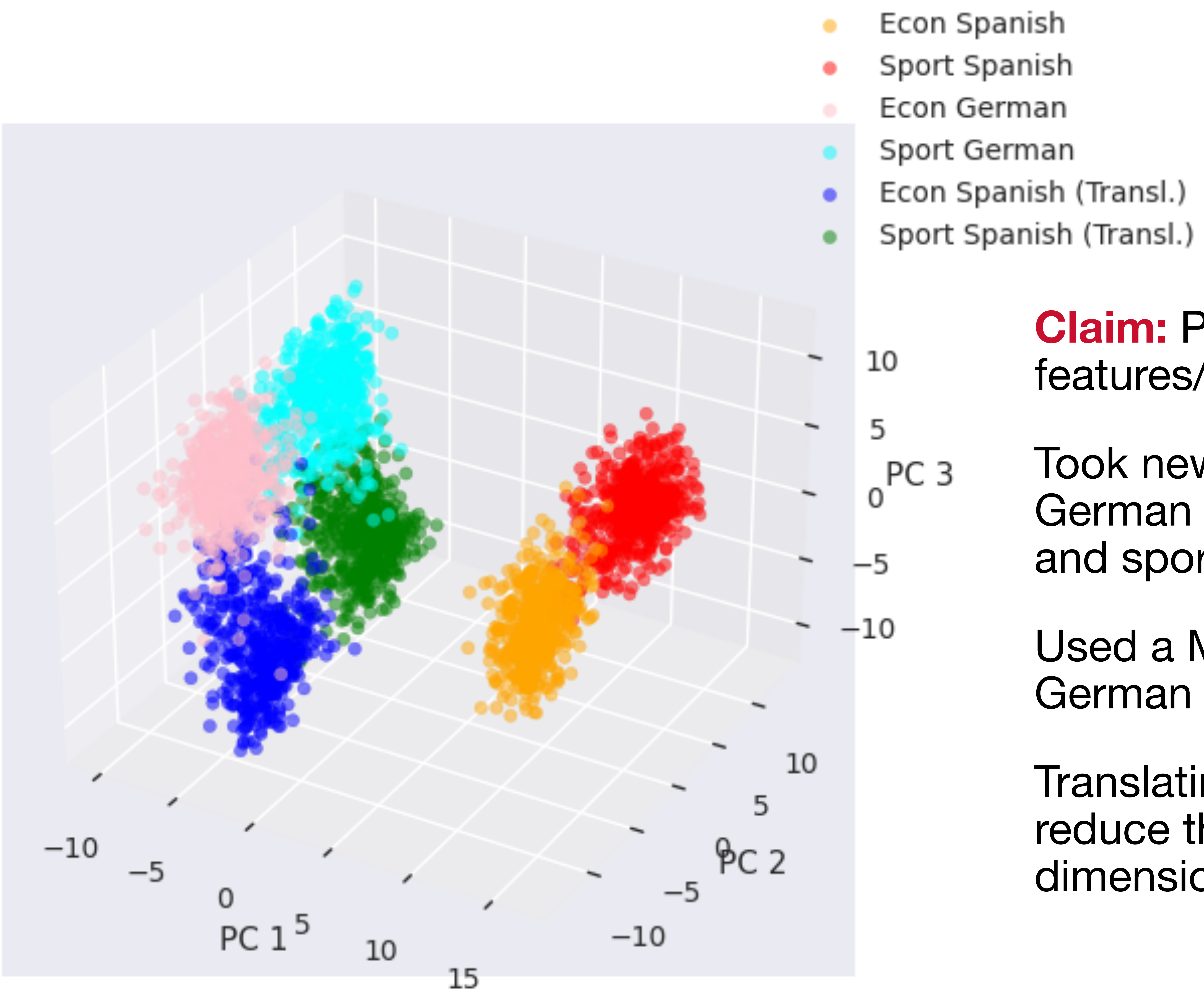
1. Embed real data and AI-generated data to see if the embedding vectors cluster.

2. Unsupervised clustering of embedded data recreates the labels in the original.

3. Detect the difference between real and machine-translated data.


A.
PCA
LDA
Stack exchange
PC2
PC1
LD1
LD2
LD3
Real
Mixtral 8x7B
Falcon 40B
Llama-2 70B


C.
Economics abstracts
PC2
PC1
LD2
LD1
Real
Prompt 1
Prompt 2



Claim: PCs reflect interpretable features/known hidden labels.

Took news articles in Spanish and German in two topics, economics and sports.

Used a ML translator to translate German to Spanish.

Translating news articles helps reduce the variation in one dimension (language).

Some takeaways and ongoing work

Model forensics and model evolution

HarmonyOS 4.0

Samsung UI 7.0

Some takeaways and ongoing work

Model forensics and model evolution

HarmonyOS 4.0

Preliminary experiments show that the embedding spaces of large “foundation models” can separate data generated from different sources.

Samsung UI 7.0

Some takeaways and ongoing work

Model forensics and model evolution

HarmonyOS 4.0

Samsung UI 7.0

Preliminary experiments show that the embedding spaces of large “foundation models” can separate data generated from different sources.

- Forensics applications: comparing models, detecting deepfakes, etc.

Some takeaways and ongoing work

Model forensics and model evolution

HarmonyOS 4.0

Samsung UI 7.0

Preliminary experiments show that the embedding spaces of large “foundation models” can separate data generated from different sources.

- Forensics applications: comparing models, detecting deepfakes, etc.
- “Model DNA”: fine-tuned or “lightly modified” models make minor modifications to the embeddings.

Some takeaways and ongoing work

Model forensics and model evolution

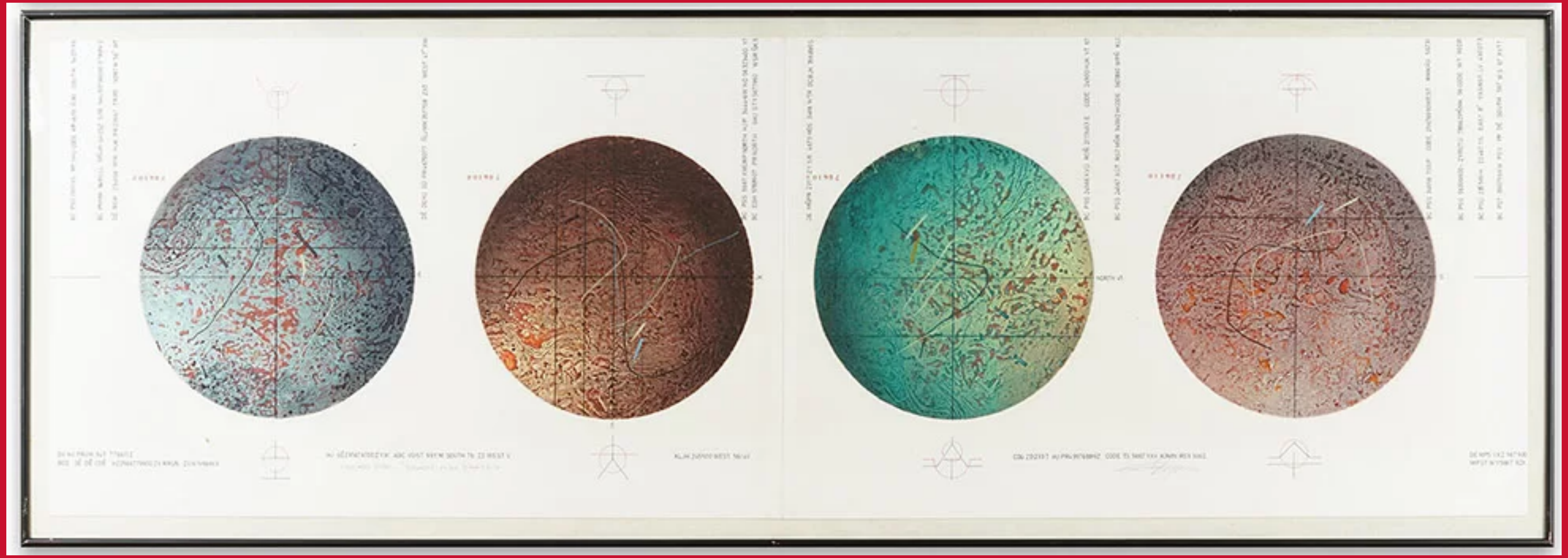
HarmonyOS 4.0

Samsung UI 7.0

Preliminary experiments show that the embedding spaces of large “foundation models” can separate data generated from different sources.

- Forensics applications: comparing models, detecting deepfakes, etc.
- “Model DNA”: fine-tuned or “lightly modified” models make minor modifications to the embeddings.
- Use post processing to “align” embeddings for calibration, ensembling, federated learning, etc.

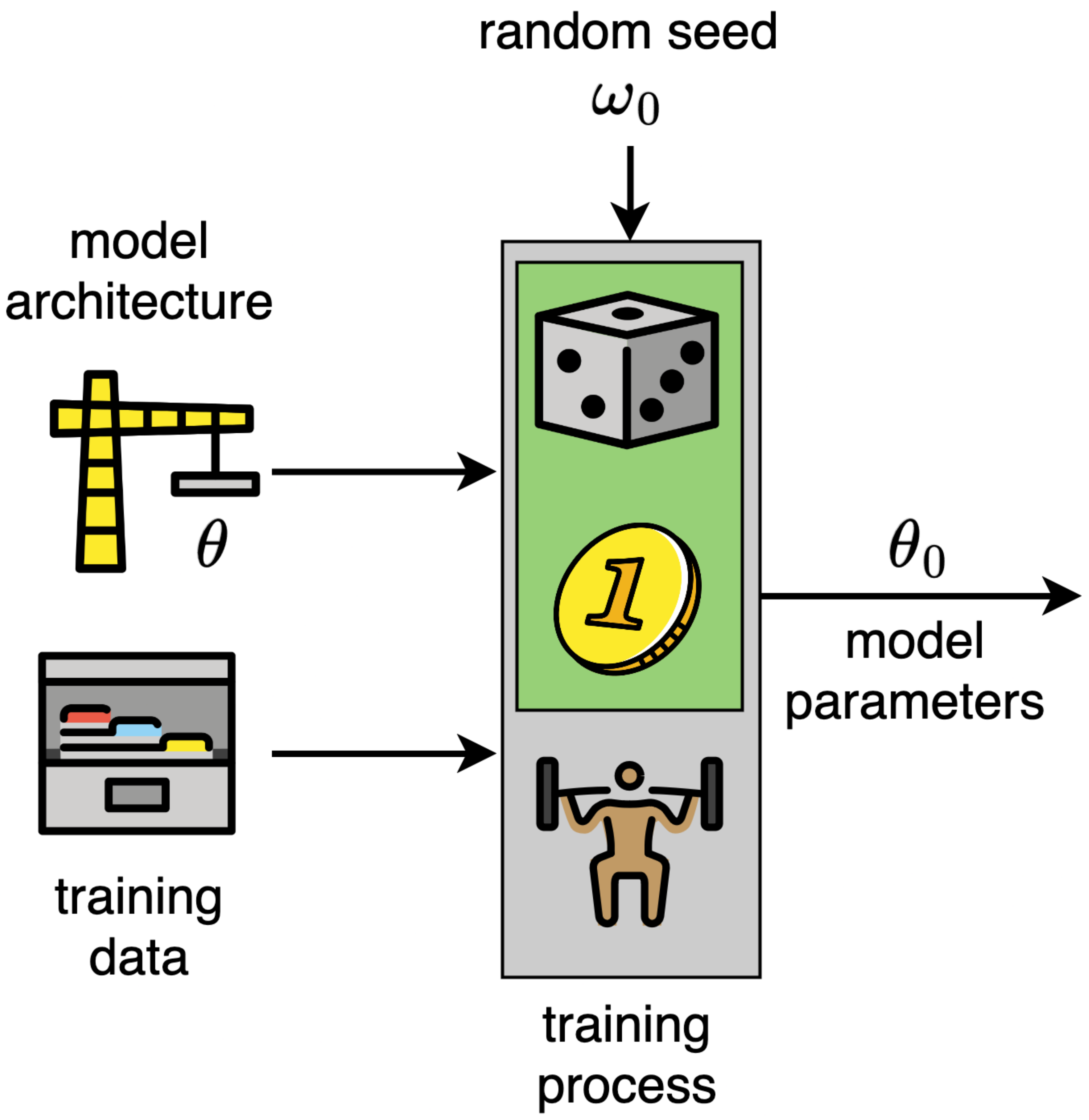
Model comparisons in training



Rm Palaniappan, *Alien Planet-D*
Viscosity, pencil colour and ink on handmade paper

Variability in the training process

Is training reliable?



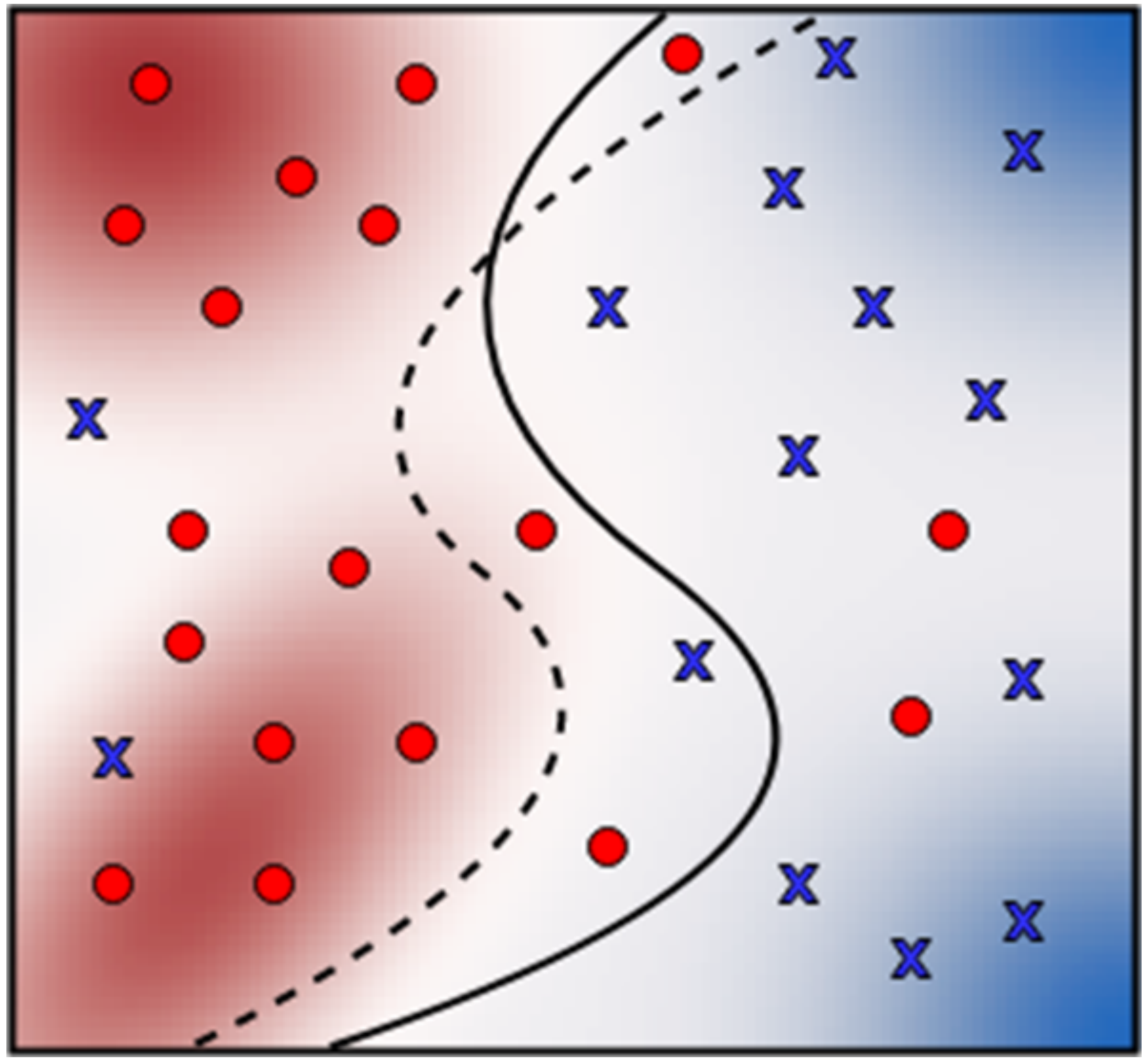
Each time we run the training algorithm on the **same training set**, **same architecture**, **same algorithm**, we still use (pseudo-)**independent randomness**.

- Each training run is a **sample** from $\mathscr{F}$.
- Given samples $f_1, f_2, \ldots, f_M$ are they similar to each other or different?

This is related to how **reproducible** a model is.

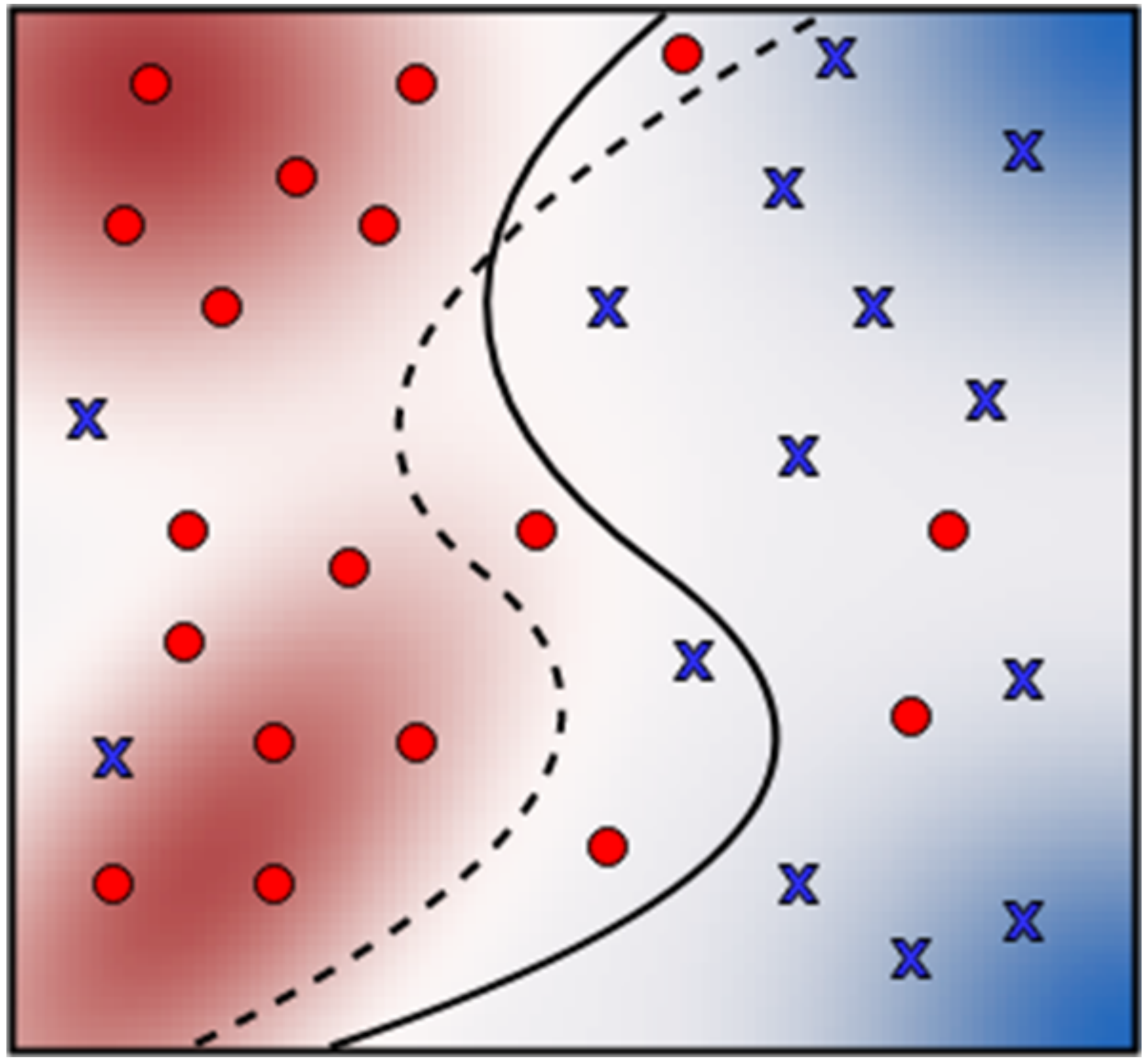
Comparing two runs of training

Model comparisons are ad hoc and waste energy



Comparing two runs of training

Model comparisons are ad hoc and waste energy

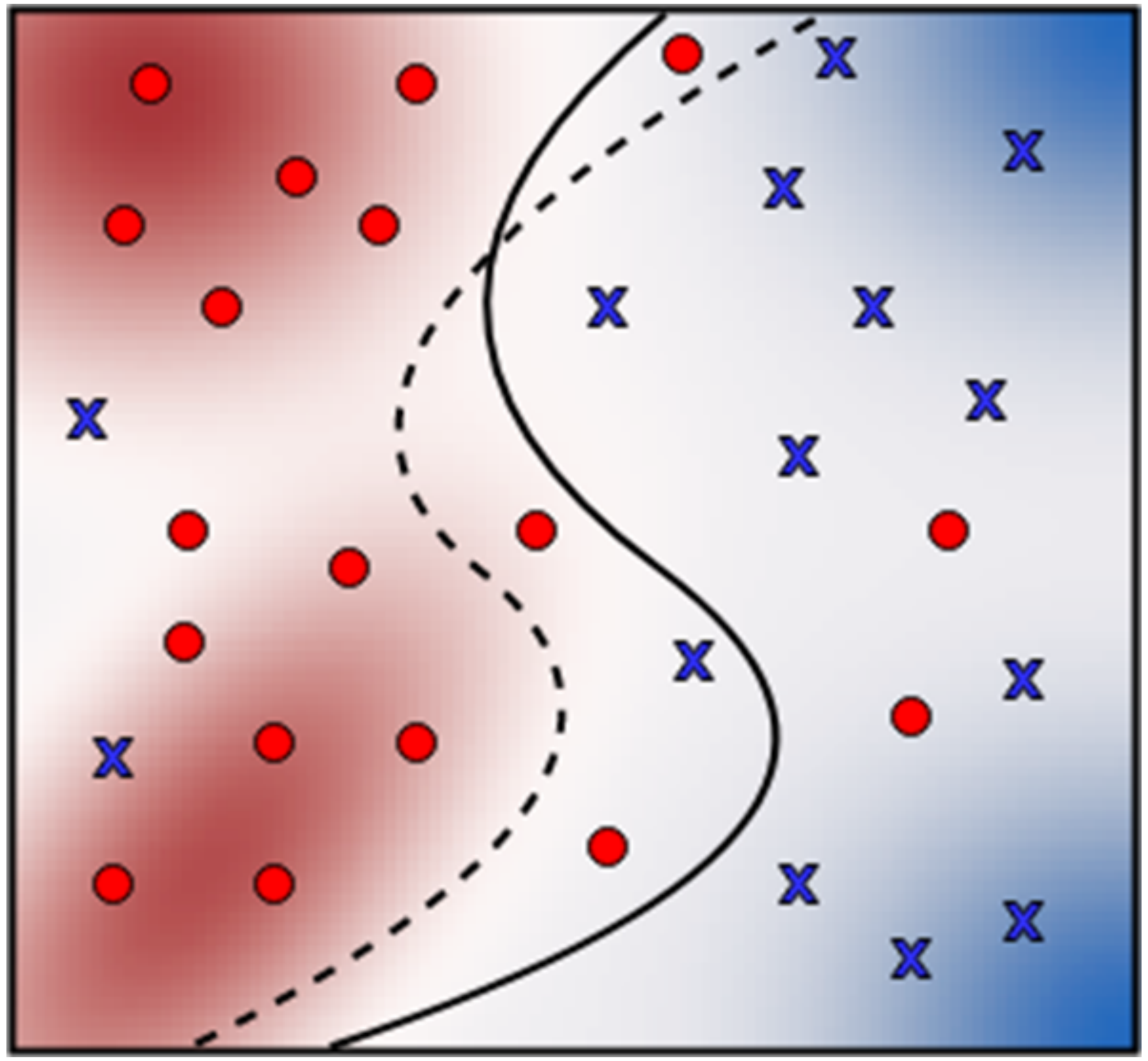


- Determining if one model is "better" than another is **not well-posed**.

Comparing two runs of training

Model comparisons are ad hoc and waste energy

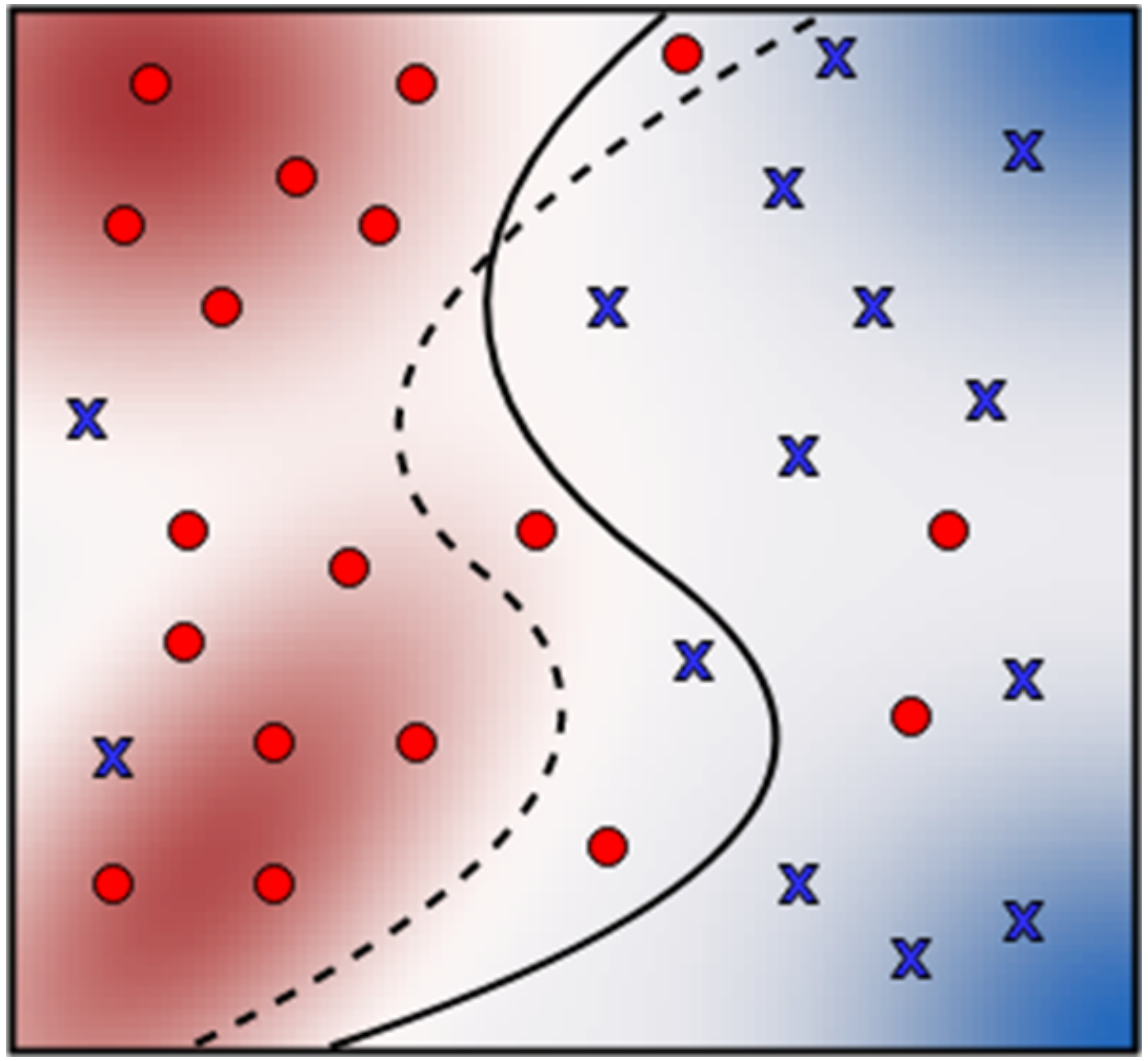
- Determining if one model is "better" than another is **not well-posed**.
- In practice, end up running the training process many times. Wasted computation, time, energy, etc.



Comparing two runs of training

Model comparisons are ad hoc and waste energy

- Determining if one model is "better" than another is **not well-posed**.
- In practice, end up running the training process many times. Wasted computation, time, energy, etc.

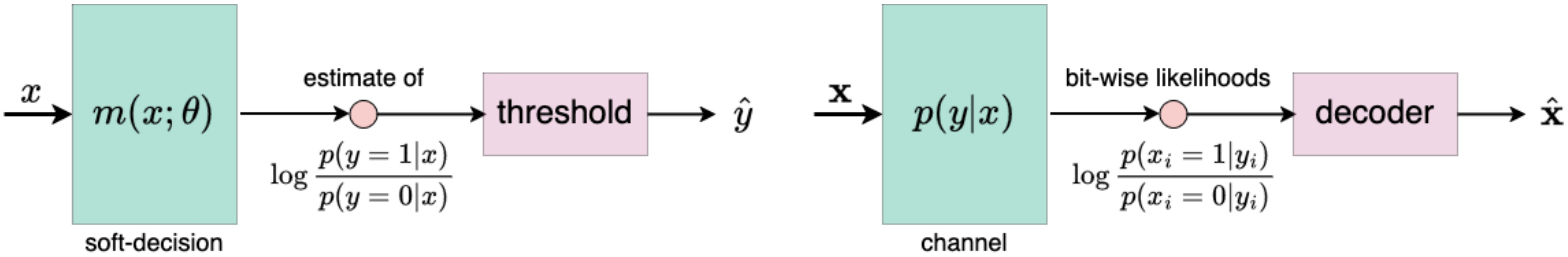


Terms like the **Rashomon effect**[1][2][3], **predictive multiplicity**[4], or **prediction churn**[5] have been used to describe this phenomena.

[1] Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, *16*(3), 199-231
[2] Fisher, A., Rudin, C., & Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research*, *20*(177), 1-81.
[3] Hsu, H., & Calmon, F. (2022). Rashomon capacity: A metric for predictive multiplicity in classification. *Advances in Neural Information Processing Systems*, *35*, 28988-29000.
[4] Milani Fard, M., Cormier, Q., Canini, K., & Gupta, M. (2016). Launch and iterate: Reducing prediction churn. *Advances in Neural Information Processing Systems*, *29*.
[5] Marx, C., Calmon, F., & Ustun, B. (2020, November). Predictive multiplicity in classification. In *International Conference on Machine Learning* (pp. 6765-6774). PMLR.

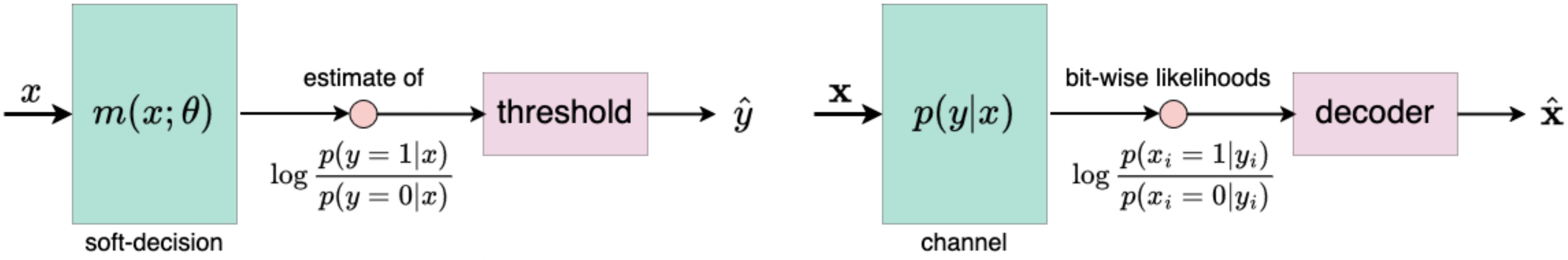
Hard decisions vs. soft decisions

Putting on a communications hat



Hard decisions vs. soft decisions

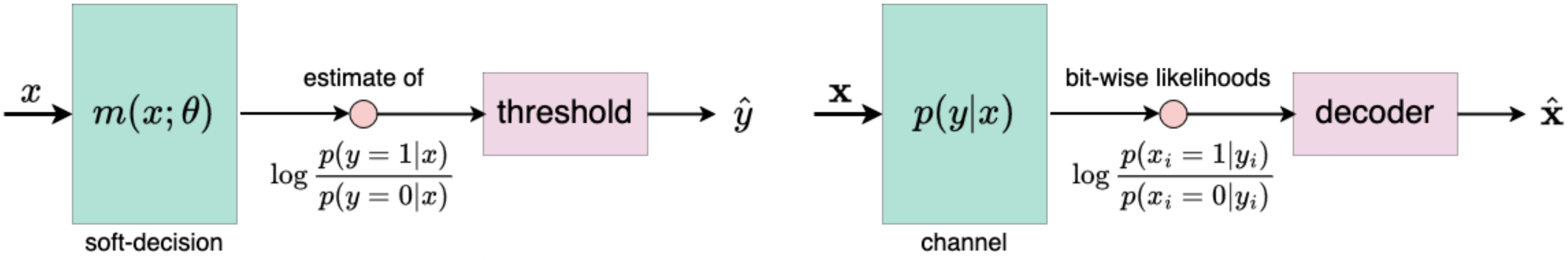
Putting on a communications hat



Test error and **churn** measure differences in “hard decisions” $f : \mathcal{X} \to [L]$.

Hard decisions vs. soft decisions

Putting on a communications hat

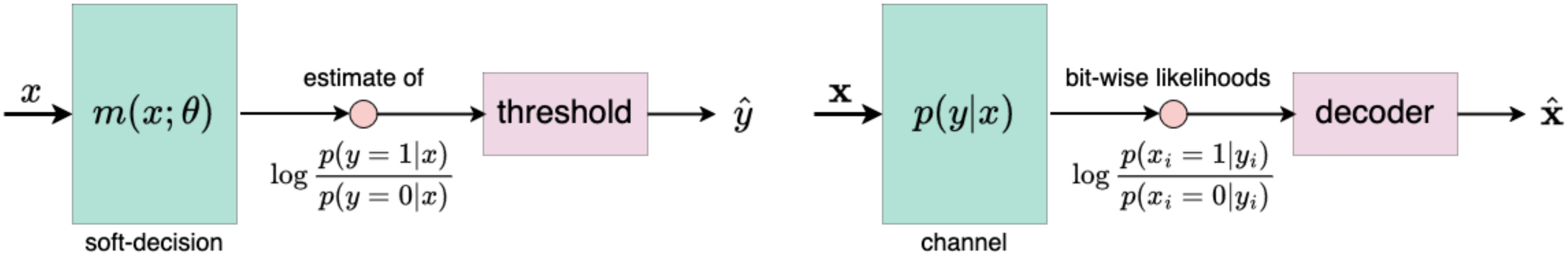


Test error and **churn** measure differences in "hard decisions" $f : \mathcal{X} \to [L]$.

- These are usually made using (softmax) probability estimates $\hat{p}(y \,|\, x, \theta)$.

Hard decisions vs. soft decisions

Putting on a communications hat

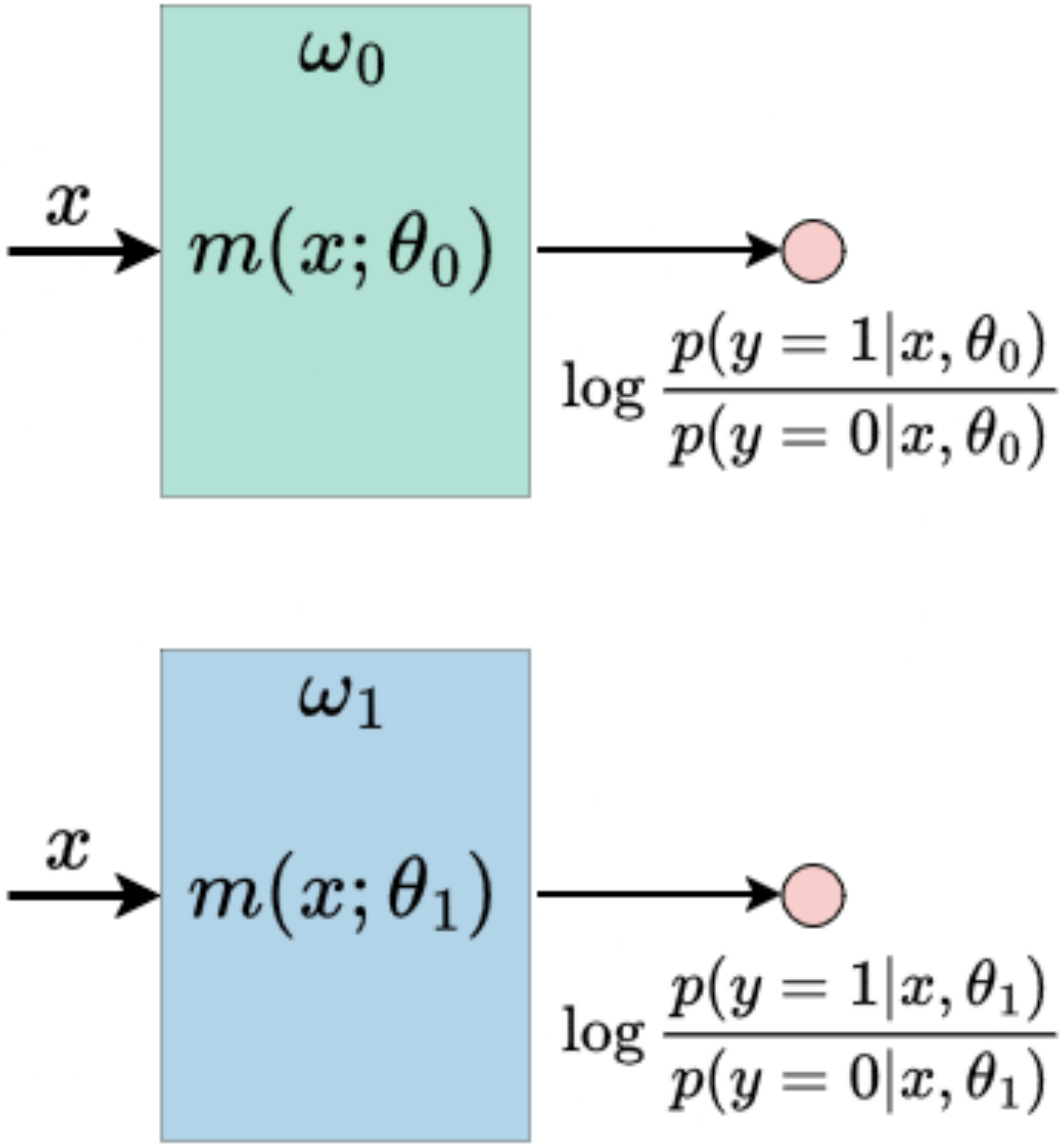


Test error and **churn** measure differences in "hard decisions" $f : \mathscr{X} \to [L]$.

- These are usually made using (softmax) probability estimates $\hat{p}(y \,|\, x, \theta)$.
- Instead look at **pre-threshold "soft decision"** $m(x \,|\, \theta)$ for the model.

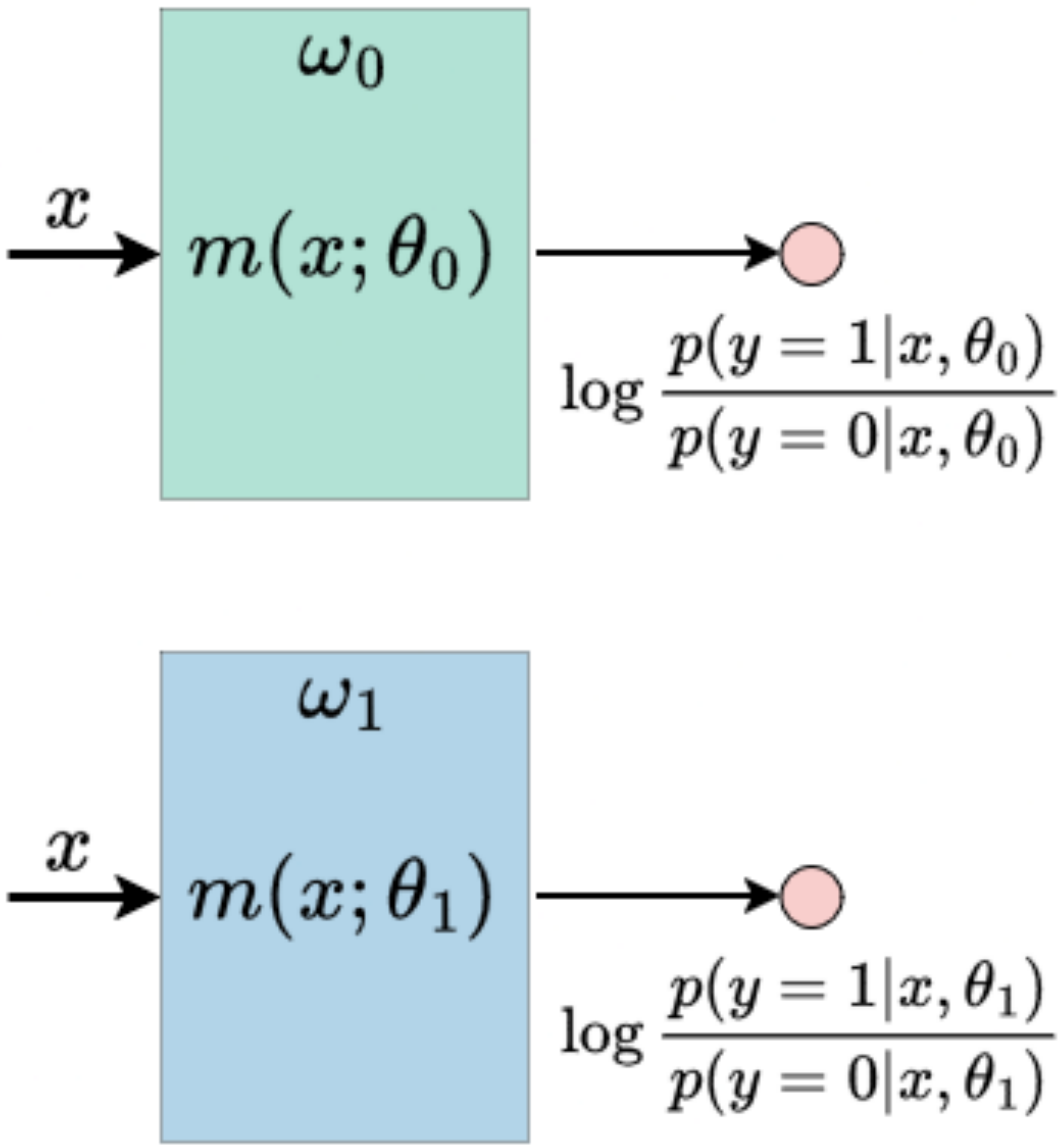
Comparing two binary classifiers

Soft decisions are different even if decisions are the same



Comparing two binary classifiers

Soft decisions are different even if decisions are the same



Measure the difference between the soft decisions/LLRs.

The LLR $m(x \mid \theta)$ of a model is a random variable that depends on x.

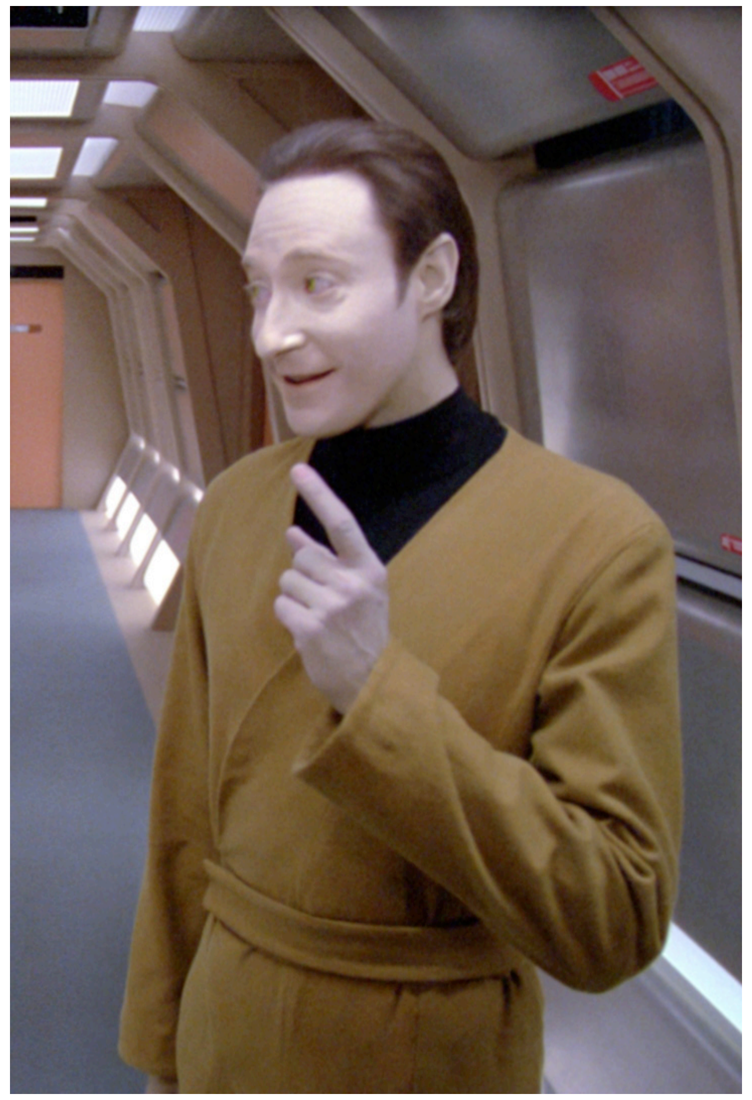
Assume the test set is made of i.i.d. draws from the input distribution.

Turn this into a **hypothesis testing problem**!

Two-sample tests for model similarity

Back to simple tools: hypothesis testing

VS.



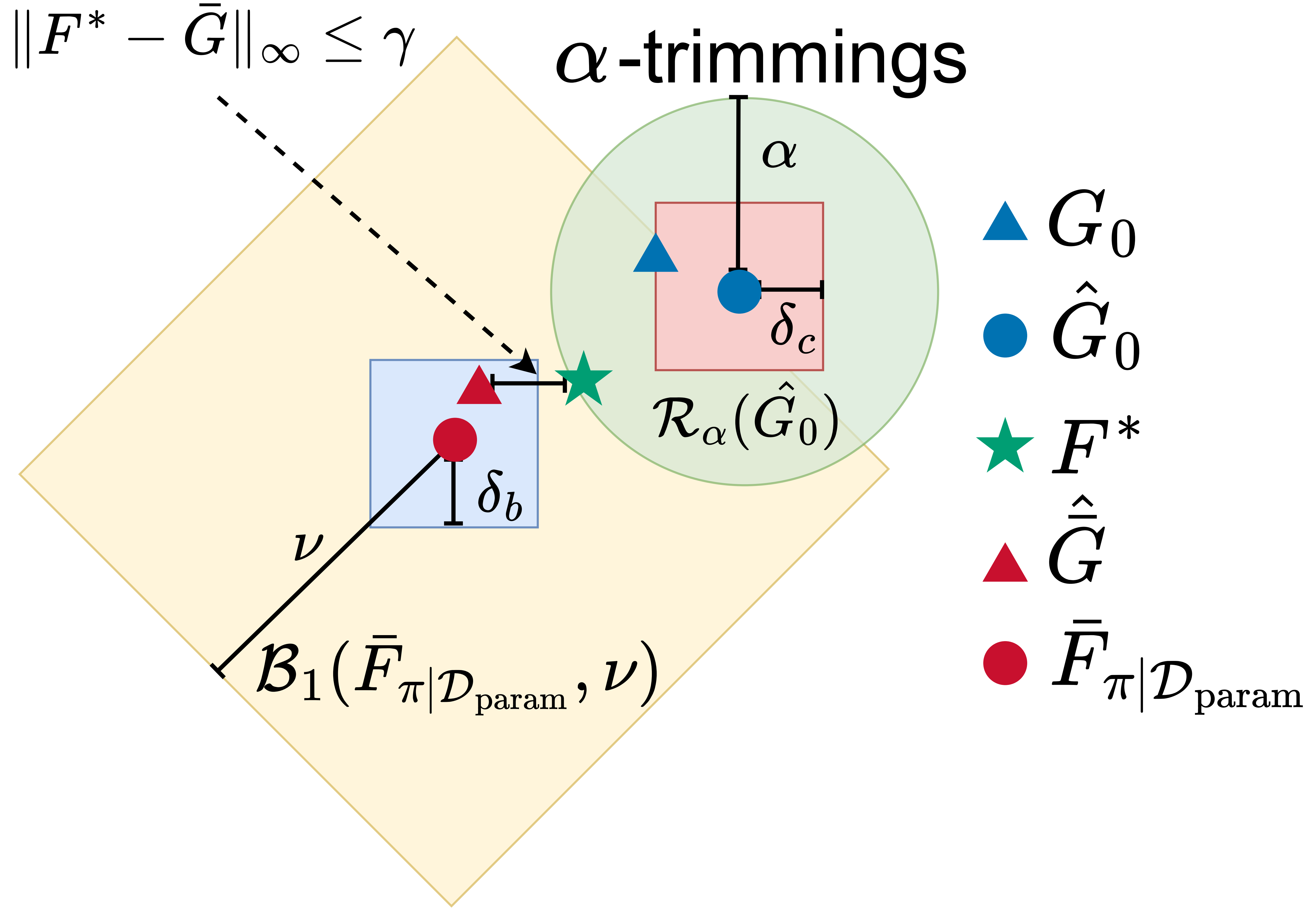
Are the models the same are different? Answer this by testing:

$$\mathscr{H}_0 : m(x; \theta_0) = m(x; \theta_1)$$

$$\mathscr{H}_1 : m(x; \theta_0) \neq m(x; \theta_1)$$

Hypothesis testing for model comparisons

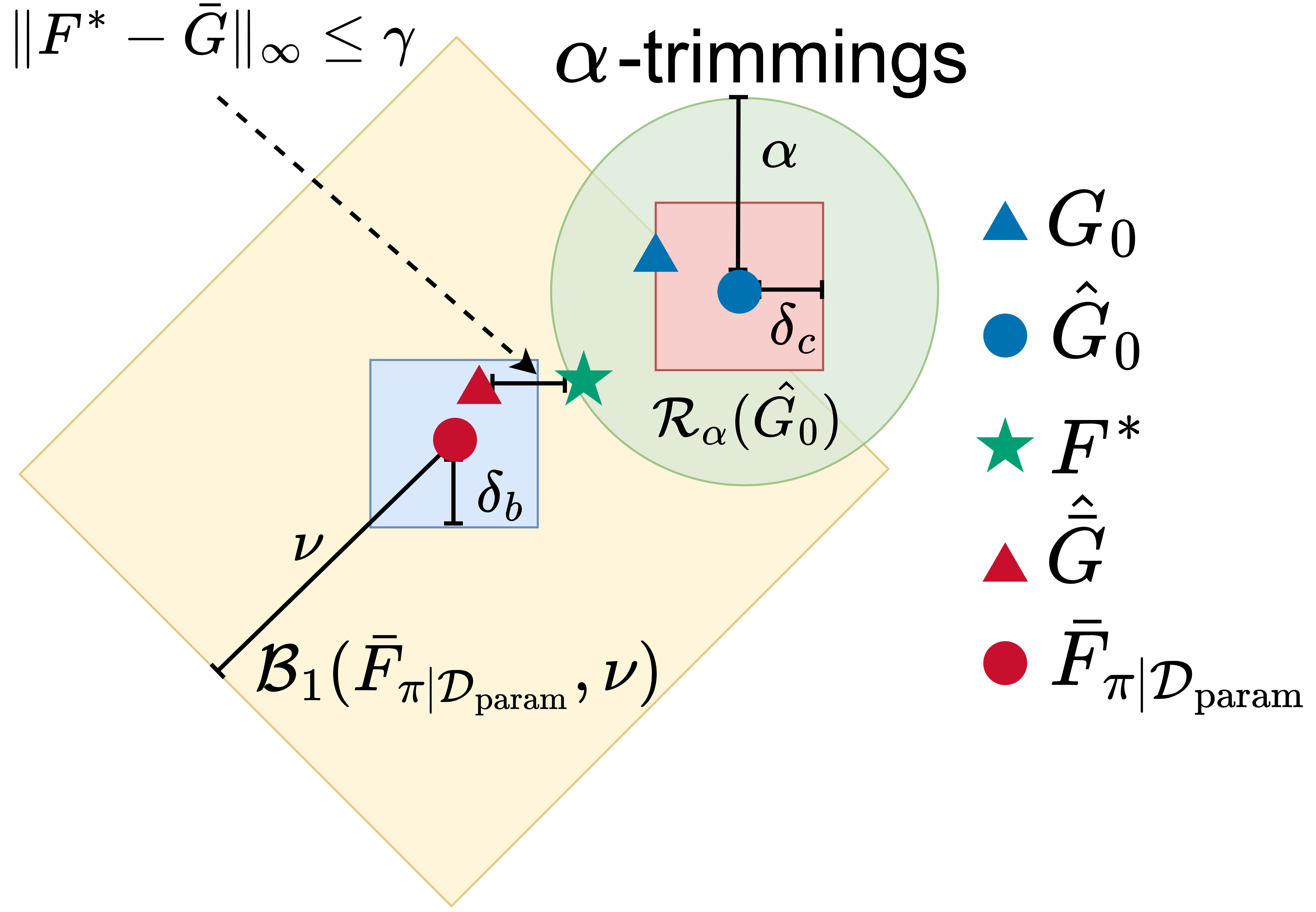
Use the test's threshold as a measure of difference



Hypothesis testing for model comparisons

Use the test's threshold as a measure of difference

Need to use empirical CDFs $\hat{G}_0$ (candidate) and $\widehat{\overline{G}}$ (null).

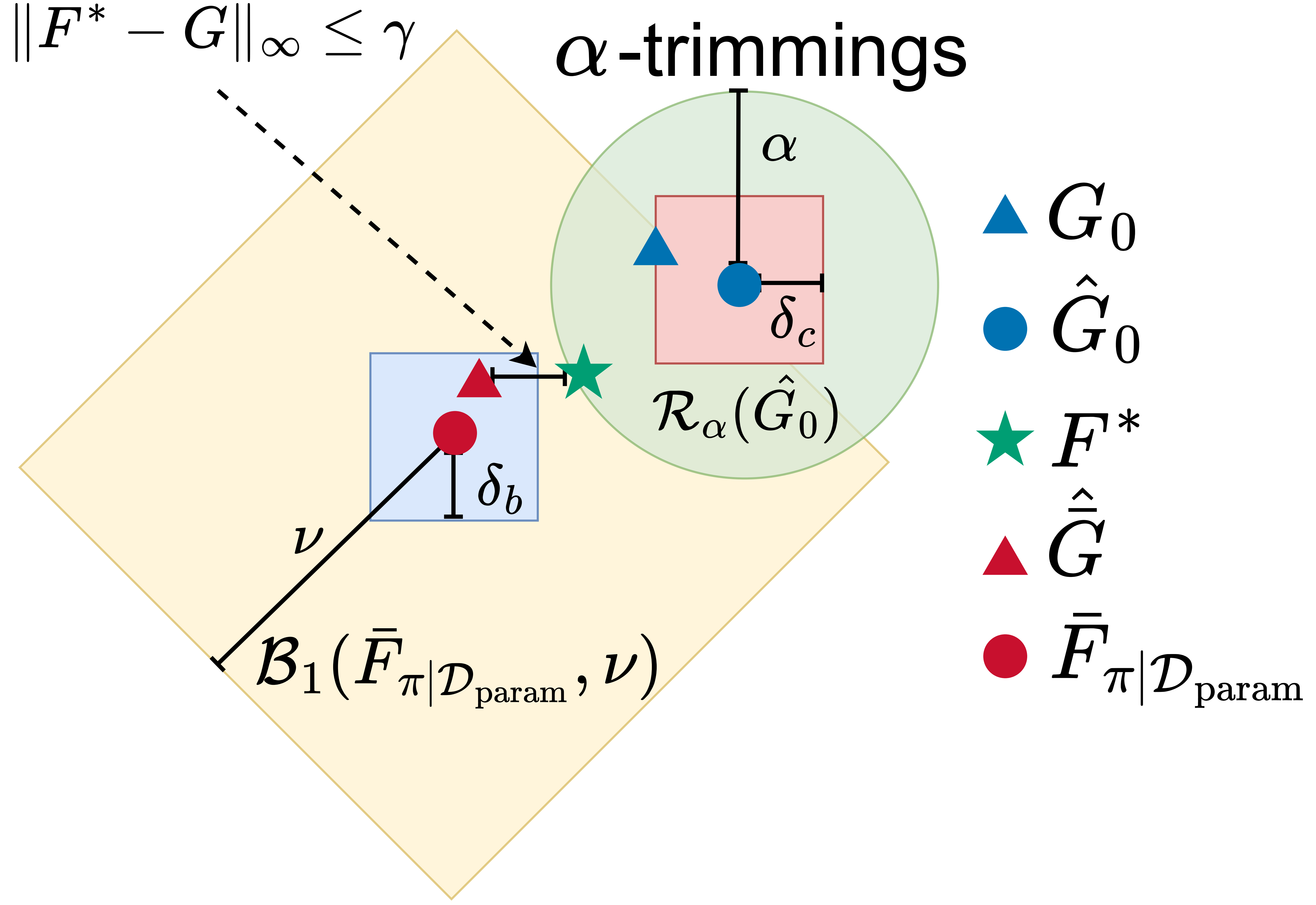


Hypothesis testing for model comparisons

Use the test's threshold as a measure of difference

Need to use empirical CDFs $\hat{G}_0$ (candidate) and $\widehat{\bar{G}}$ (null).

Optimize to find the *closest model to $\widehat{\bar{G}}$ in a ball around $\hat{G}_0$.*



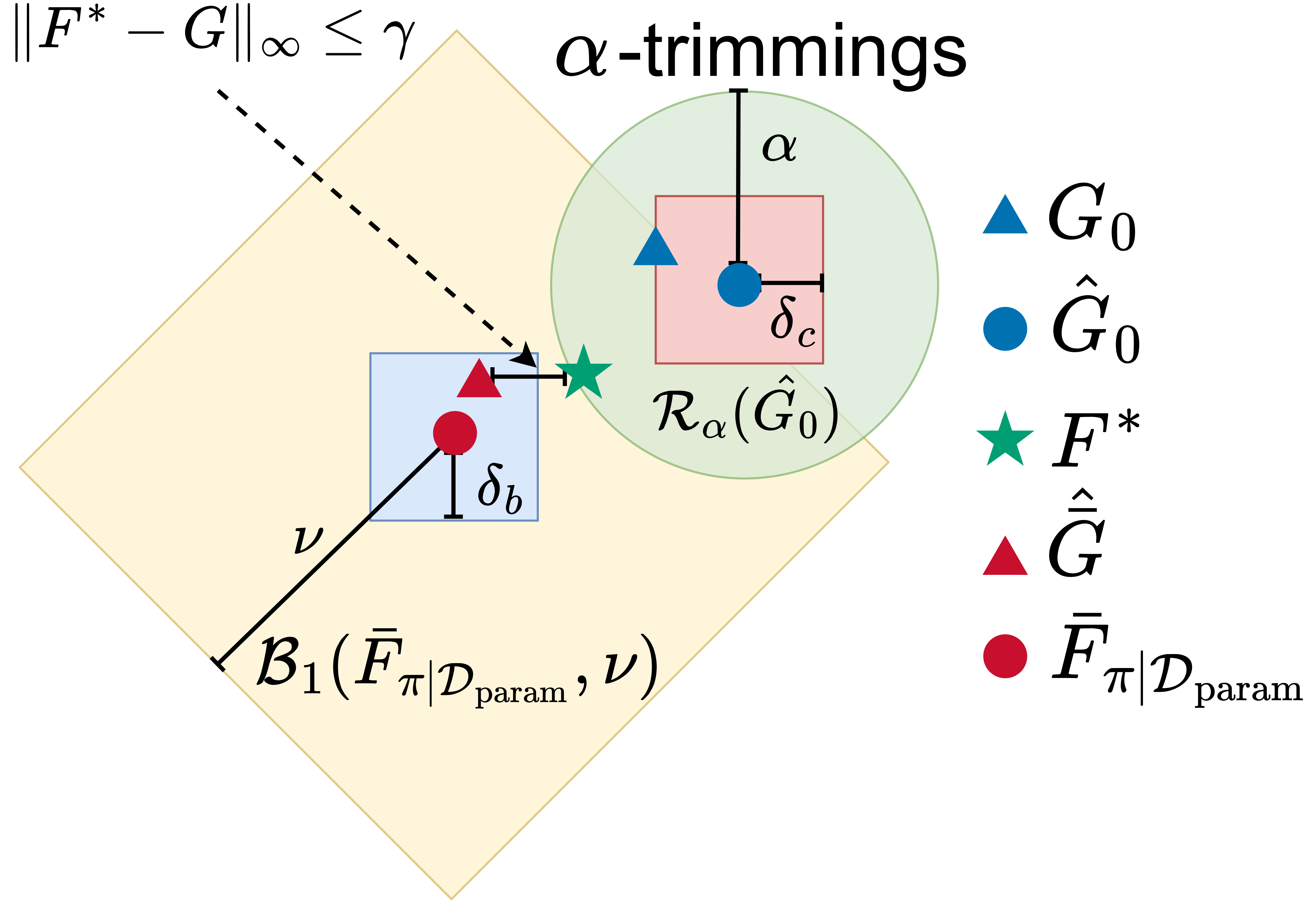
Hypothesis testing for model comparisons

Use the test's threshold as a measure of difference

Need to use empirical CDFs $\hat{G}_0$ (candidate) and $\widehat{\hat{G}}$ (null).

Optimize to find the *closest model to $\widehat{\hat{G}}$ in a ball around $\hat{G}_0$.*

This is a search over "α-trimmings" which can be done efficiently (del Barrio el 2020, Álvarez-Esteban et al. 2011).



Hypothesis testing for model comparisons

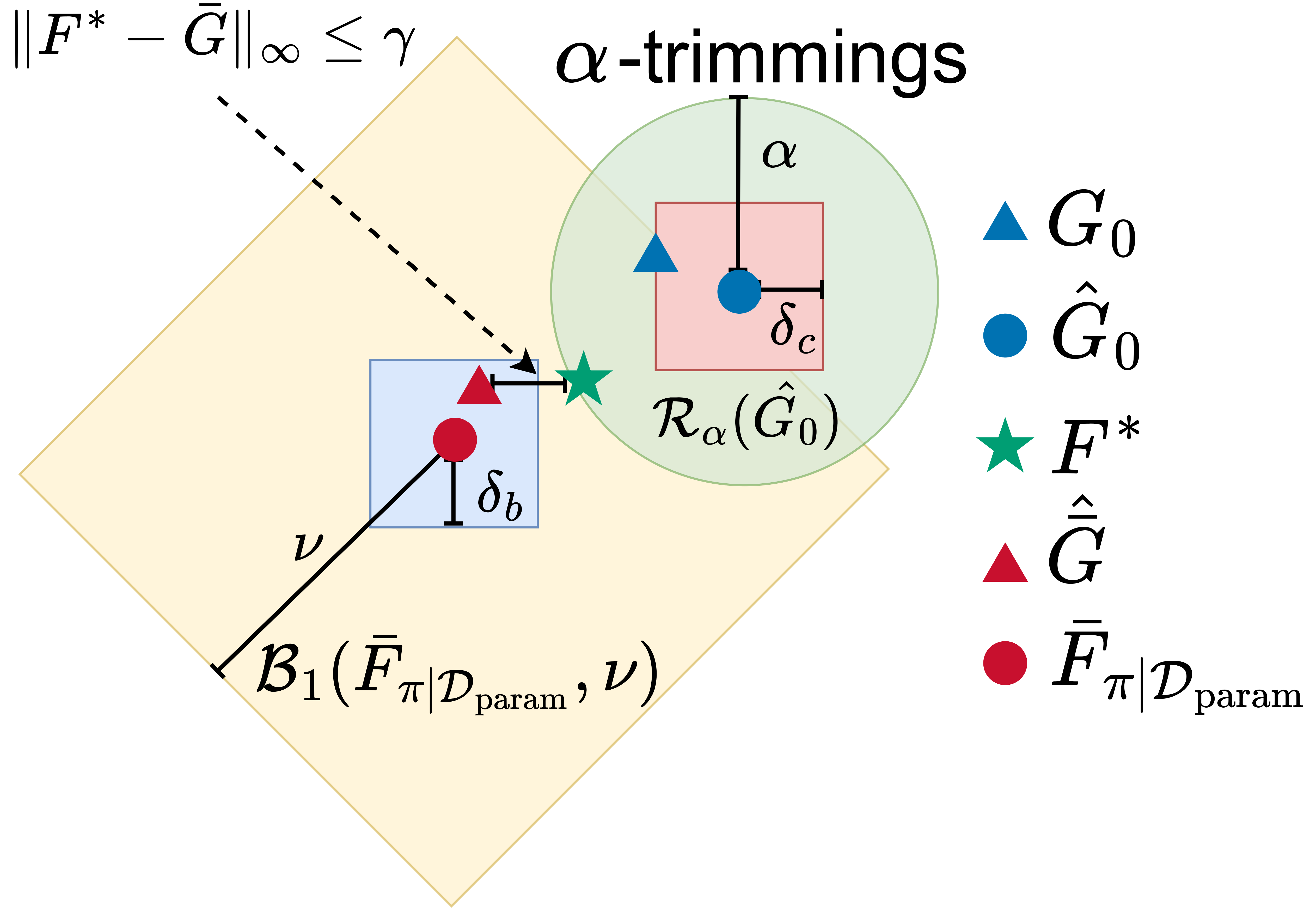
Use the test's threshold as a measure of difference

Need to use empirical CDFs $\hat{G}_0$ (candidate) and $\widehat{\bar{G}}$ (null).

Optimize to find the *closest model to $\widehat{\bar{G}}$ in a ball around $\hat{G}_0$.*

This is a search over "α-trimmings" which can be done efficiently (del Barrio el 2020, Álvarez-Esteban et al. 2011).

Define a new discrepancy measure $\hat{\alpha}$ as the minimum level for the test (= radius of the ball) to accept.



Advantages over other measures

Other measures are pairwise or less information about the models

Advantages over other measures

Other measures are pairwise or less information about the models

1. **Test/validation accuracy**: if two models have similar test performance, “one is as good as the other.”

Advantages over other measures

Other measures are pairwise or less information about the models

1. **Test/validation accuracy**: if two models have similar test performance, “one is as good as the other.”

2. **Churn**: the two models do not disagree on the test set.

Advantages over other measures

Other measures are pairwise or less information about the models

1. **Test/validation accuracy**: if two models have similar test performance, “one is as good as the other.”

2. **Churn**: the two models do not disagree on the test set.

3. **Expected Calibration Error** (ECE) (Naeini et al. 2015): measures the difference between accuracy and expected “confidence” (the LLR).

Advantages over other measures

Other measures are pairwise or less information about the models

1. **Test/validation accuracy**: if two models have similar test performance, “one is as good as the other.”

2. **Churn**: the two models do not disagree on the test set.

3. **Expected Calibration Error** (ECE) (Naeini et al. 2015): measures the difference between accuracy and expected “confidence” (the LLR).

For our new $\hat{\alpha}$ measure:

Advantages over other measures

Other measures are pairwise or less information about the models

1. **Test/validation accuracy**: if two models have similar test performance, "one is as good as the other."

2. **Churn**: the two models do not disagree on the test set.

3. **Expected Calibration Error** (ECE) (Naeini et al. 2015): measures the difference between accuracy and expected "confidence" (the LLR).

For our new $\hat{\alpha}$ measure:

- When $\hat{\alpha}$ is large, at least one of the other metrics is also large.

Advantages over other measures

Other measures are pairwise or less information about the models

1. **Test/validation accuracy**: if two models have similar test performance, “one is as good as the other.”

2. **Churn**: the two models do not disagree on the test set.

3. **Expected Calibration Error** (ECE) (Naeini et al. 2015): measures the difference between accuracy and expected “confidence” (the LLR).

For our new $\hat{\alpha}$ measure:

- When $\hat{\alpha}$ is large, at least one of the other metrics is also large.

- Models with small $\hat{\alpha}$ are generally low on all the other metrics as well.

Connecting back to our story

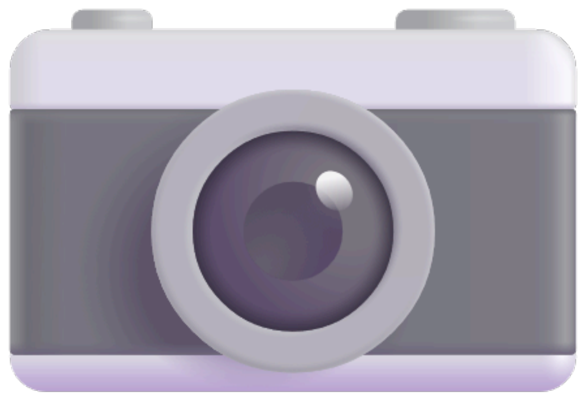
“Reliable” training algorithm should produce “typical” models


iOS 8.3


iOS 18.4


HarmonyOS 4.0


Samsung UI 7.0


MS 3D Fluent


SerenityOS

Connecting back to our story

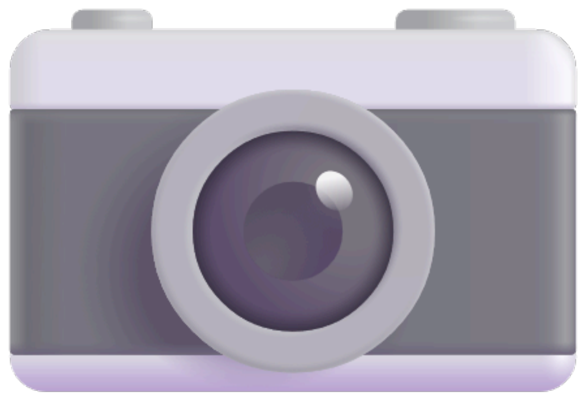
“Reliable” training algorithm should produce “typical” models


iOS 8.3


iOS 18.4


HarmonyOS 4.0


Samsung UI 7.0


MS 3D Fluent


SerenityOS

Measures like $\hat{\alpha}$ (using ℓ_1 balls, Wasserstein balls, etc.) can let us **measure “atypicality.”**

Connecting back to our story

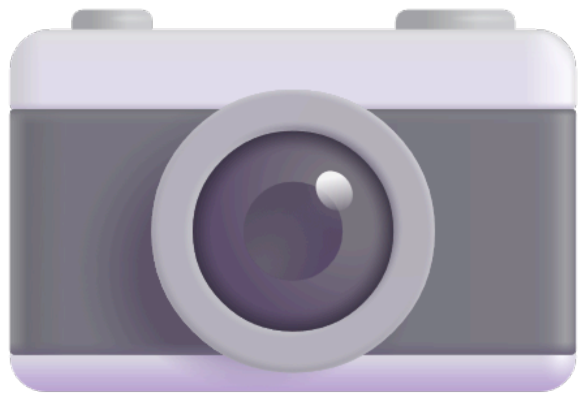
“Reliable” training algorithm should produce “typical” models

iOS 8.3

iOS 18.4

HarmonyOS 4.0

Samsung UI 7.0



MS 3D Fluent

SerenityOS

Measures like $\hat{\alpha}$ (using ℓ_1 balls, Wasserstein balls, etc.) can let us **measure “atypicality.”**

- Use this to design **new methods for model ensembling**.

Connecting back to our story

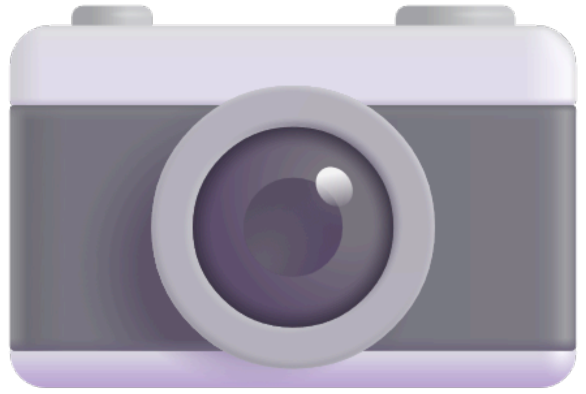
“Reliable” training algorithm should produce “typical” models

iOS 8.3

iOS 18.4

HarmonyOS 4.0

Samsung UI 7.0



MS 3D Fluent

SerenityOS

Measures like $\hat{\alpha}$ (using ℓ_1 balls, Wasserstein balls, etc.) can let us **measure “atypicality.”**

- Use this to design **new methods for model ensembling**.
- Apply it to **other features of trained models** (e.g. NTK spectra) to find model differences.

Connecting back to our story

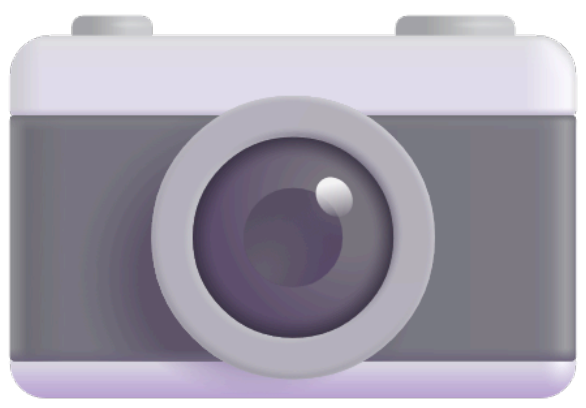
“Reliable” training algorithm should produce “typical” models

iOS 8.3

iOS 18.4

HarmonyOS 4.0

Samsung UI 7.0



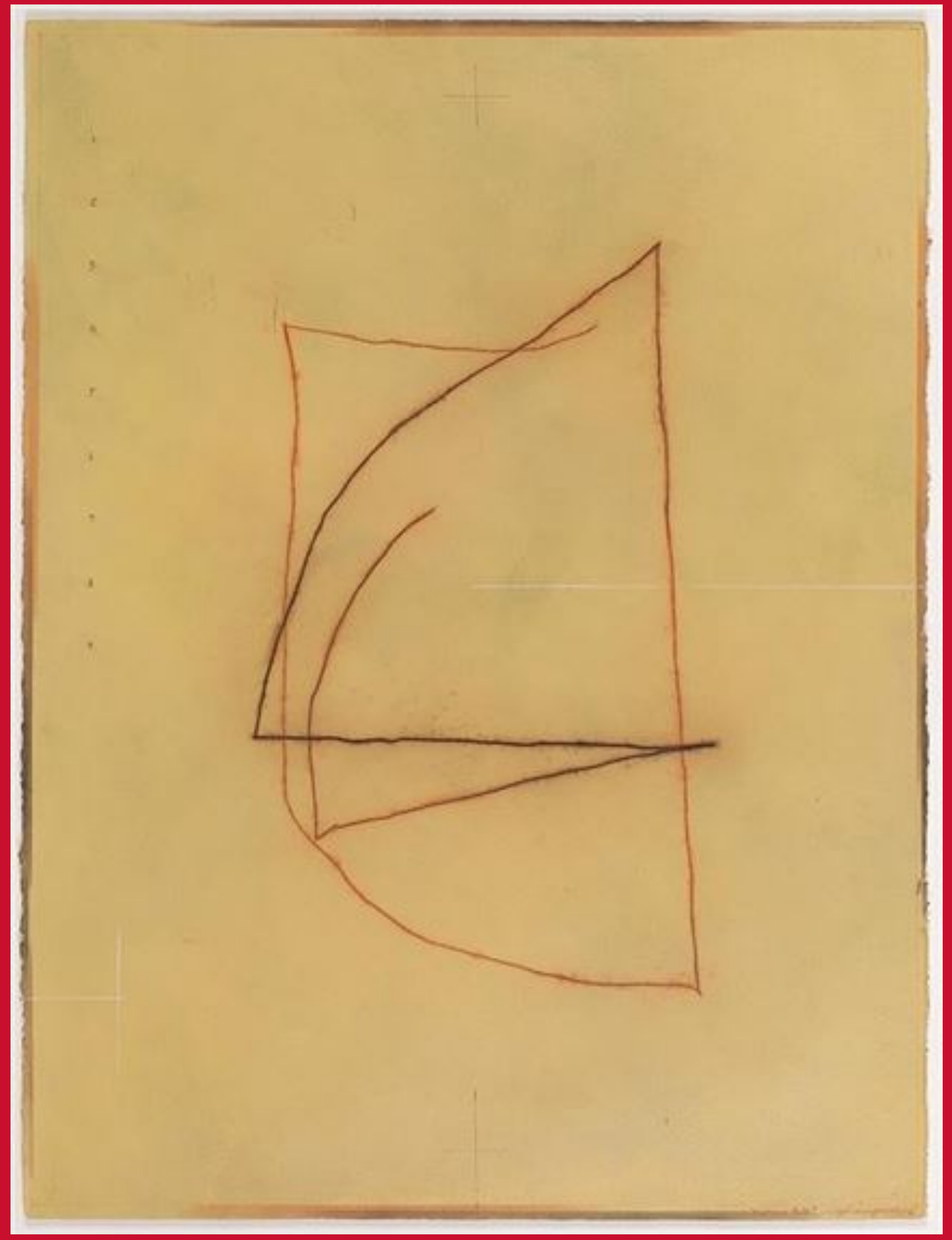
MS 3D Fluent

SerenityOS

Measures like $\hat{\alpha}$ (using ℓ_1 balls, Wasserstein balls, etc.) can let us **measure “atypicality.”**

- Use this to design **new methods for model ensembling**.
- Apply it to **other features of trained models** (e.g. NTK spectra) to find model differences.
- Connect it to **process engineering** and other industrial production ideas.

Some final remarks



Rm Palaniappan, *Intense Talk*
Mixed media on paper pasted on mount board

Back to the original question

What does any of this mean for “AI for Science”?

Back to the original question

What does any of this mean for “AI for Science”?

To use large ML/AI models as part of a scientific workflow, we need “interpretability” and “reliability.”

Back to the original question

What does any of this mean for "AI for Science"?

To use large ML/AI models as part of a scientific workflow, we need "interpretability" and "reliability."

We also need to understand "reliability" for the training/fine-tuning processes.

Back to the original question

What does any of this mean for “AI for Science”?

To use large ML/AI models as part of a scientific workflow, we need “interpretability” and “reliability.”

We also need to understand “reliability” for the training/fine-tuning processes.

It’s more important to **compare models directly** and not just their **performance**.

Where is this all going?

Maybe some strange new worlds

Developing a good set of techniques for model comparisons requires thinking from several different directions:

Where is this all going?

Maybe some strange new worlds

Developing a good set of techniques for model comparisons requires thinking from several different directions:

- **Theory:** can we instead compare surrogate models like “faithful” NTK representations (Engel et al. 2024)?

Where is this all going?

Maybe some strange new worlds

Developing a good set of techniques for model comparisons requires thinking from several different directions:

- **Theory:** can we instead compare surrogate models like “faithful” NTK representations (Engel et al. 2024)?
- **Experiment:** can we do these comparisons cheaply (e.g. using academic-level resources)?

Where is this all going?

Maybe some strange new worlds

Developing a good set of techniques for model comparisons requires thinking from several different directions:

- **Theory:** can we instead compare surrogate models like "faithful" NTK representations (Engel et al. 2024)?
- **Experiment:** can we do these comparisons cheaply (e.g. using academic-level resources)?
- **Application:** how do we use model comparisons in forensics, process engineering, ensembling, and beyond?

谢谢大家的关注!