CERFACS

CENTRE EUROPÉEN DE RECHERCHE ET DE FORMATION AVANCÉE EN CALCUL SCIENTIFIQUE

INP TOULOUSE ENSEEIHT

Intelligence Artificielle pour la Prévision

Corentin Lapeyre - CERFACS

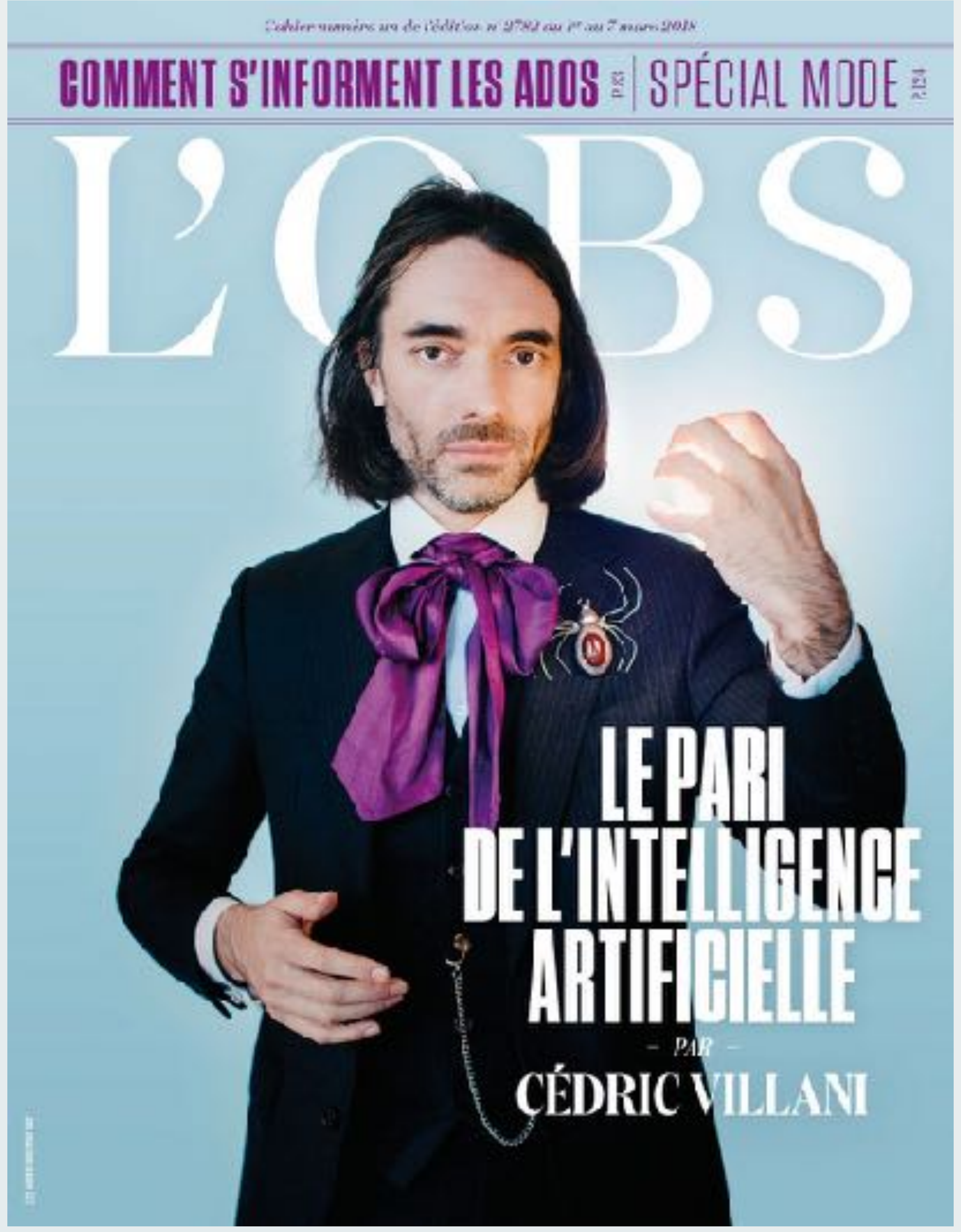
ENSEEIHT • Module UIAP

2023.10.03

Comment l'Intelligence Artificielle va changer nos vies
Libération
France Inter
N°1
VOYAGE AU CŒUR DE L'IA
HORS-SÉRIE
REPORTAGES
RENCONTRES
Comment l'Intelligence Artificielle change nos vies
Libération
France Inter
N°2
L'IA AU CŒUR DE L'HUMAIN
Avec Alain Damasio, Yuval Noah Harari, Cathy O'Neil...

Federico Capasso Le designer du monde quantique
La Recherche
Astronomie
Météorologie
Chimie
Santé-médecine
Génétique...
L'INTELLIGENCE ARTIFICIELLE transforme les sciences
NOVEMBRE 2017

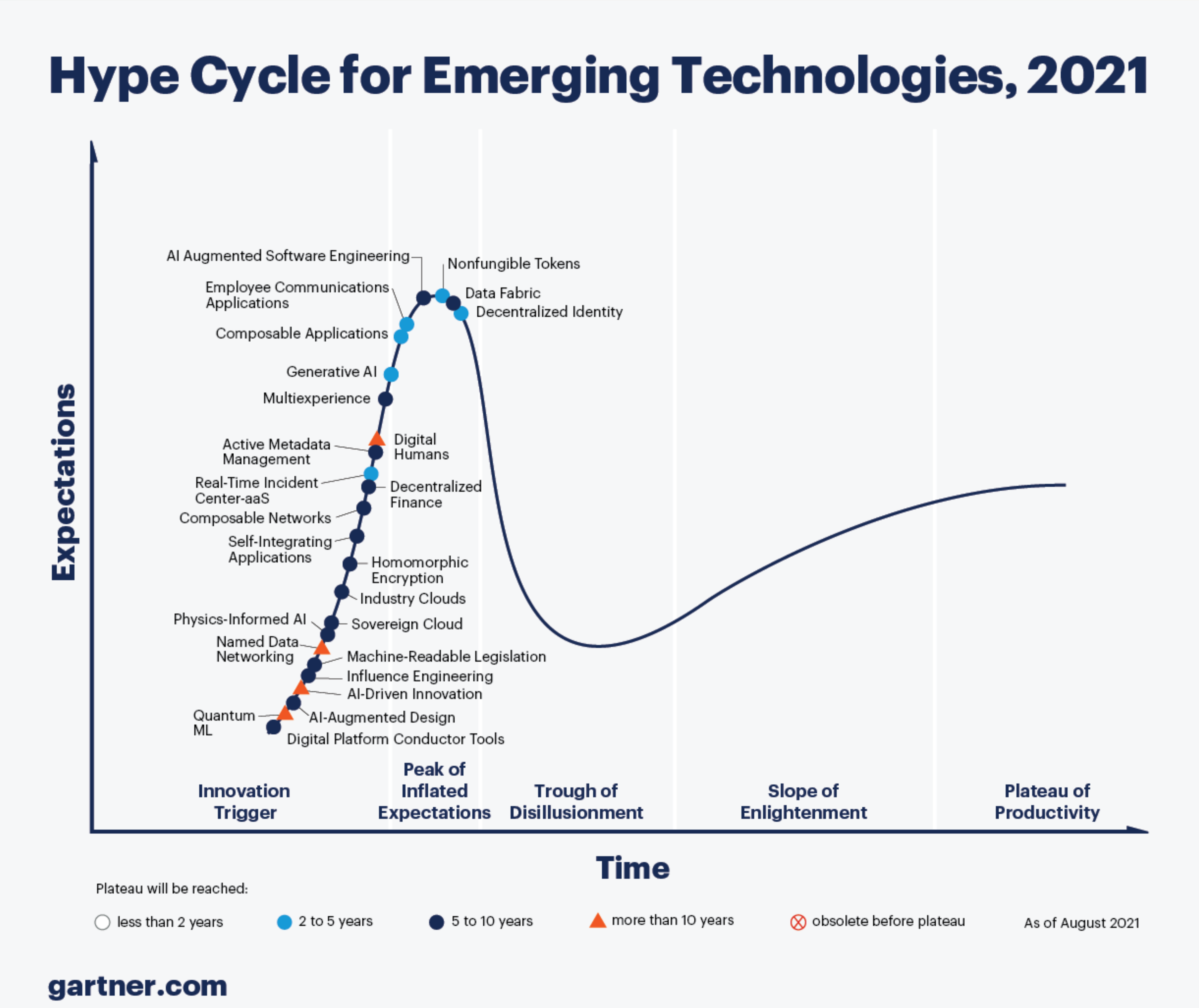
Le Point
Hebdomadaire d'information du jeudi 16 mars 2017 n° 2323
Intelligence artificielle
Comment profiter du monde nouveau
La révolution qui devrait être au cœur de la présidentielle
Avec Thomas L. Friedman, Charles-Edouard Bouée et Yann Le Cun


COMMENT S'INFORMENT LES ADOS
SPÉCIAL MODE
L'OBS
LE PARI DE L'INTELLIGENCE ARTIFICIELLE
PAR
CÉDRIC VILLANI

Courrier international
INTELLIGENCE ARTIFICIELLE
LA MACHINE À FANTASMES
EN SUPPLÉMENT 2017, L'ANNÉE EN CARTOONS
INFLUENCE — LE BEL AUTOMNE FRANÇAIS
ÉCONOMIE — BITCOIN, UN FLÉAU ÉCOLOGIQUE
Les robots vont-ils détruire nos emplois et nos vies ? Pour en finir avec les idées reçues, le décryptage de la MIT Technology Review
Voyage
Acapulco, mon paradis perdu

...

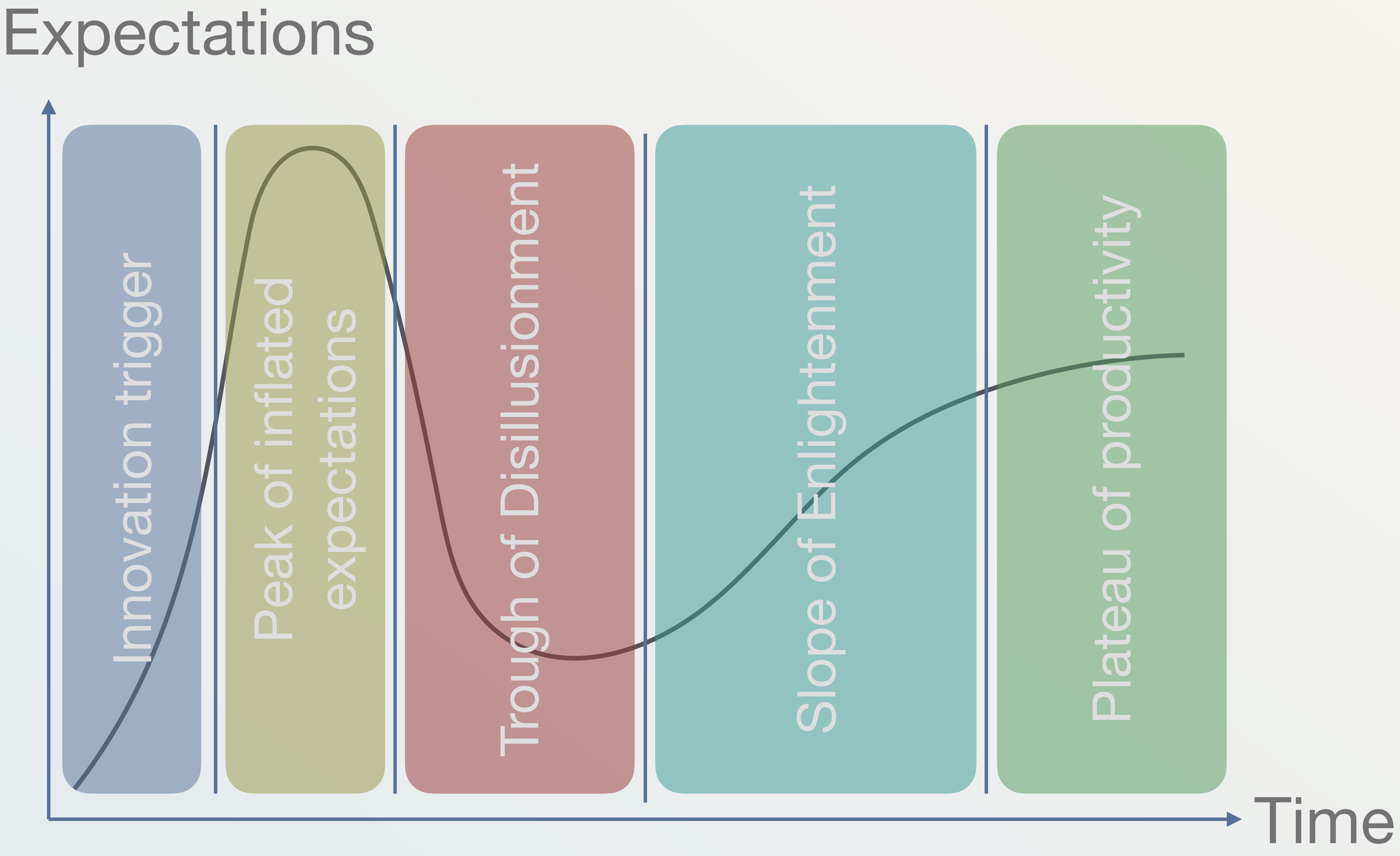
The hype



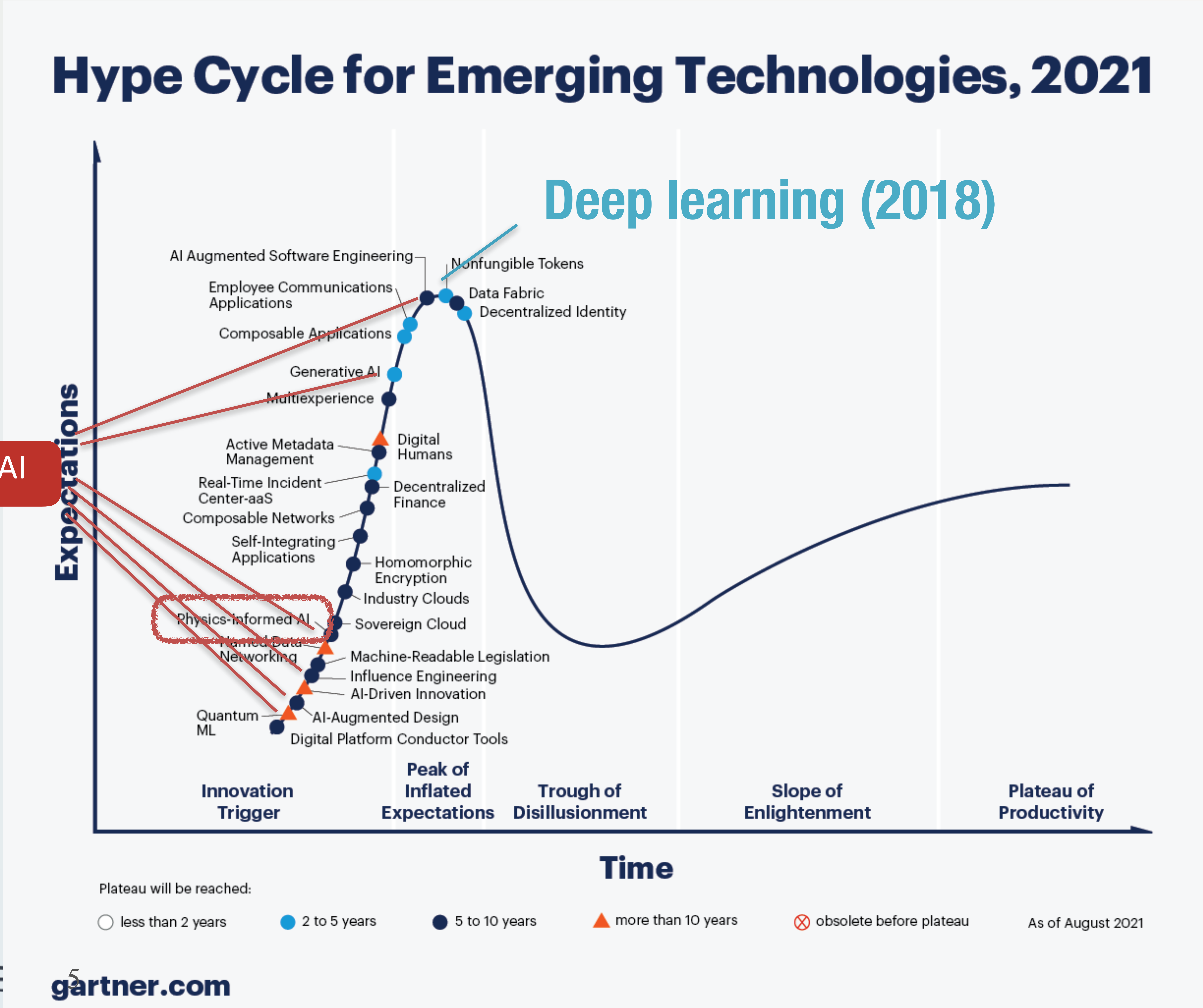
Hype Cycle for Emerging Technologies, 2021

gartner.com

The hype


Expectations
Innovation trigger
Peak of inflated expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of productivity
Time

The hype



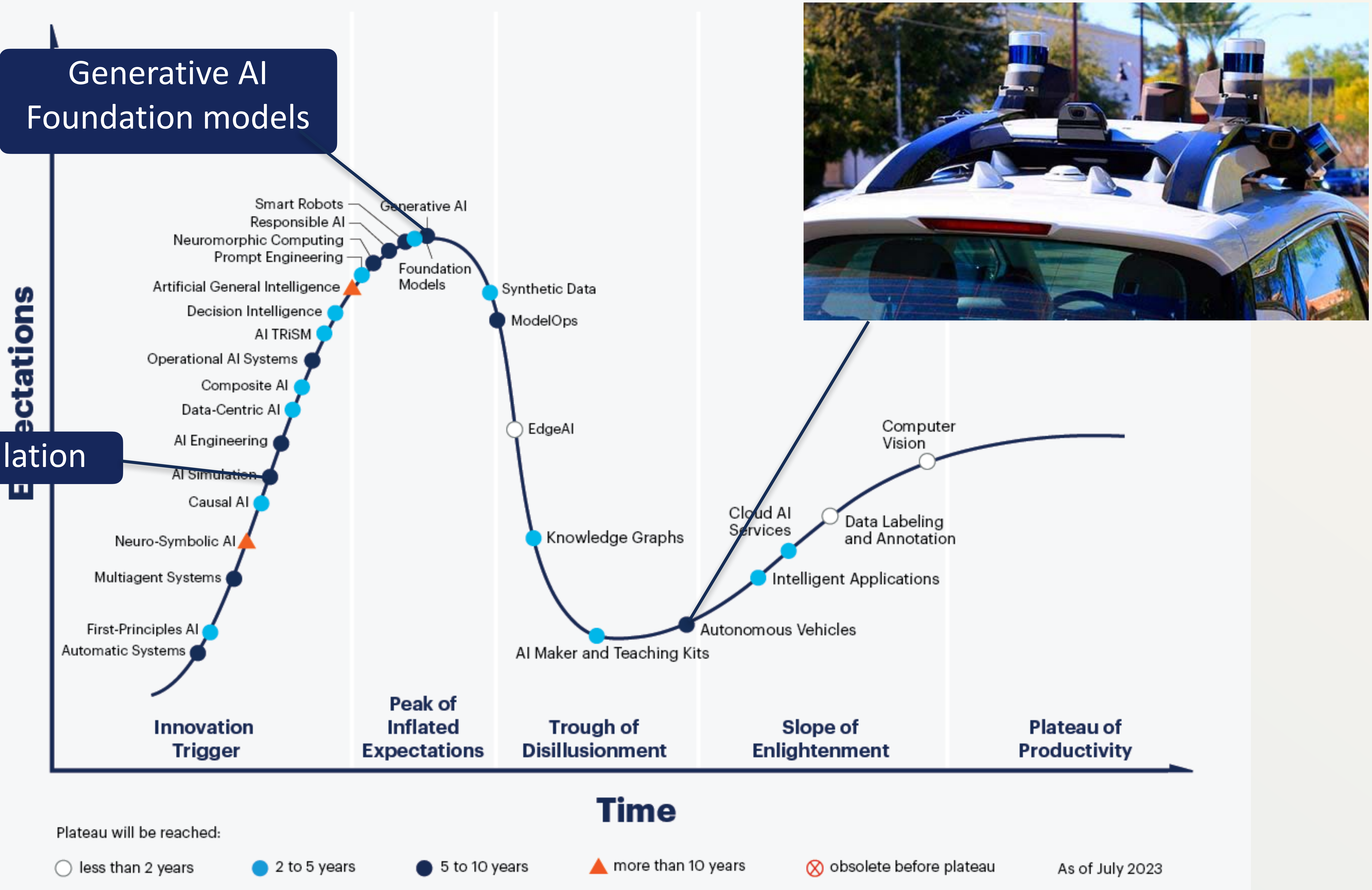
Deep learning (2018)

AI

Hype Cycle for Artificial Intelligence, 2023

Generative AI
Foundation models

AI Simulation



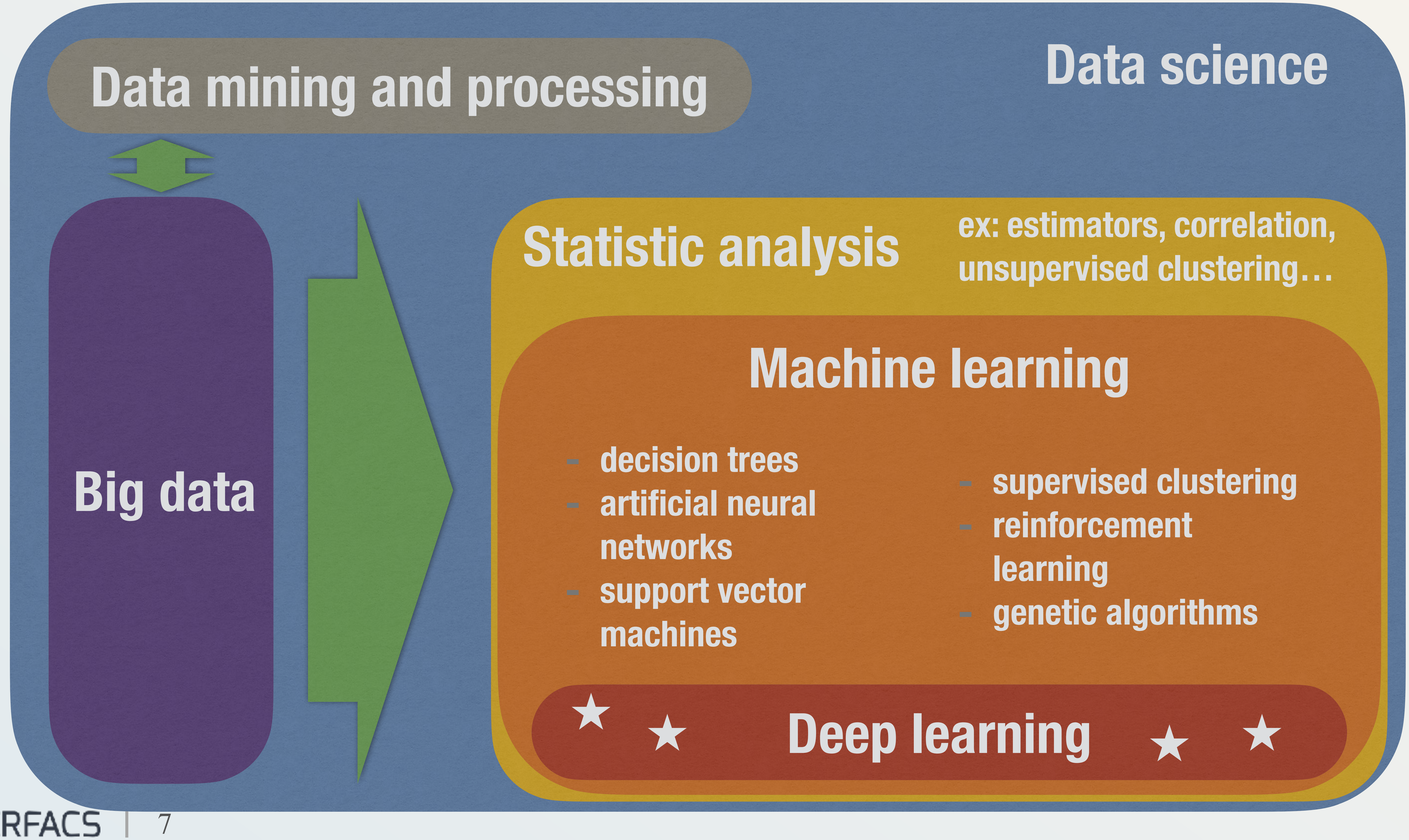
gartner.com

Source: Gartner

Data Science

Statistics: The science of collecting, displaying, and analysing data

oxfordreference.com



Data science

Data mining and processing

Big data

Statistic analysis

ex: estimators, correlation, unsupervised clustering…

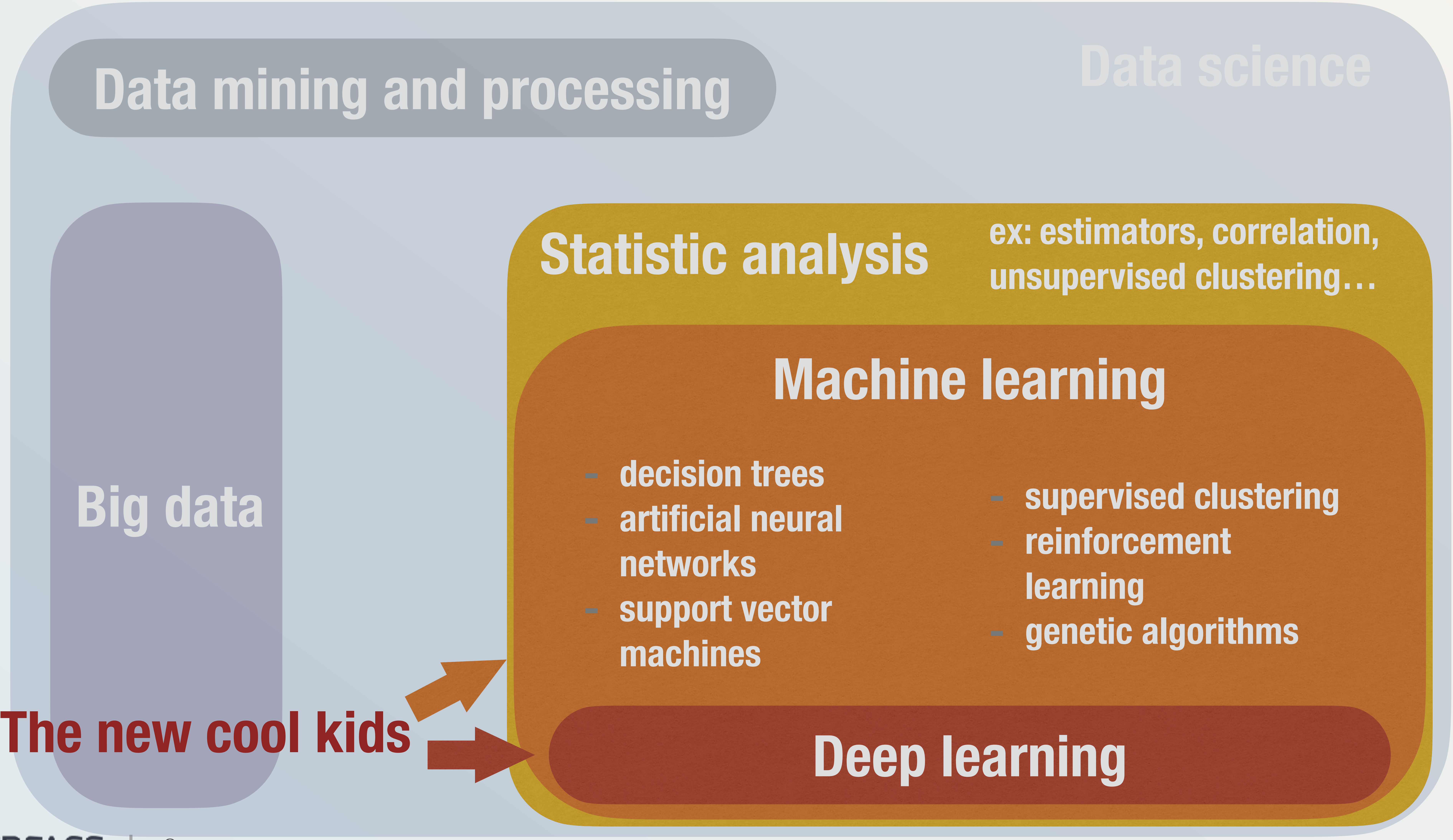
Machine learning

- decision trees
- artificial neural networks
- support vector machines

- supervised clustering
- reinforcement learning
- genetic algorithms

★ ★ Deep learning ★ ★

Data Sciences



Data science

Data mining and processing

Big data

Statistic analysis

ex: estimators, correlation, unsupervised clustering…

Machine learning

- decision trees
- artificial neural networks
- support vector machines
- supervised clustering
- reinforcement learning
- genetic algorithms

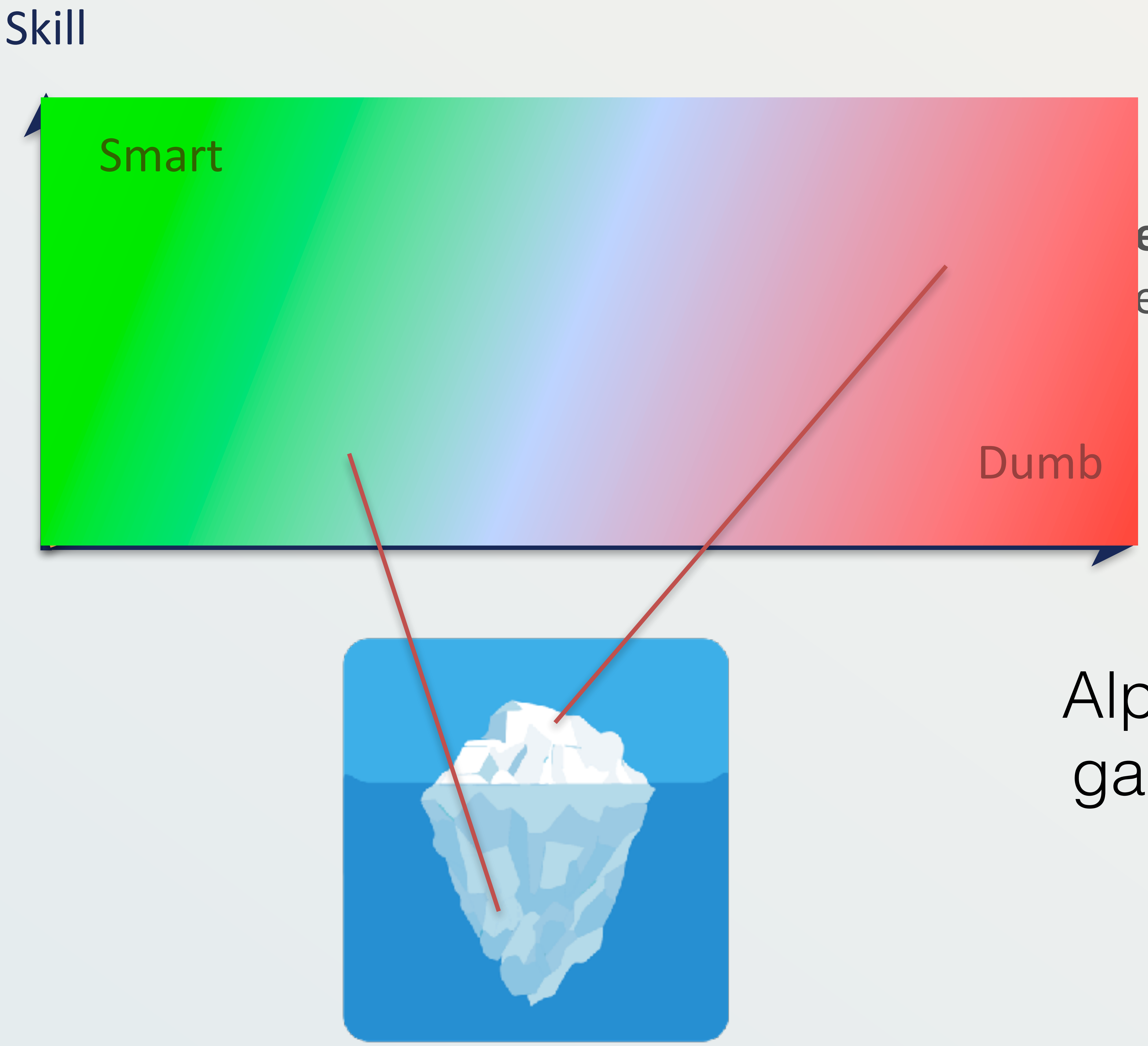
Deep learning

The new cool kids

Intelligence vs Experience

- One definition of intelligence:
  (from F. Chollet)

$$\text{Intelligence} = \frac{\text{Skill}}{\text{Experience}}$$



enough experience can e gained, ML eventually eats humans

Alpha Zero[1] needs 21 Million games of Go during training

but

training takes ≈24h

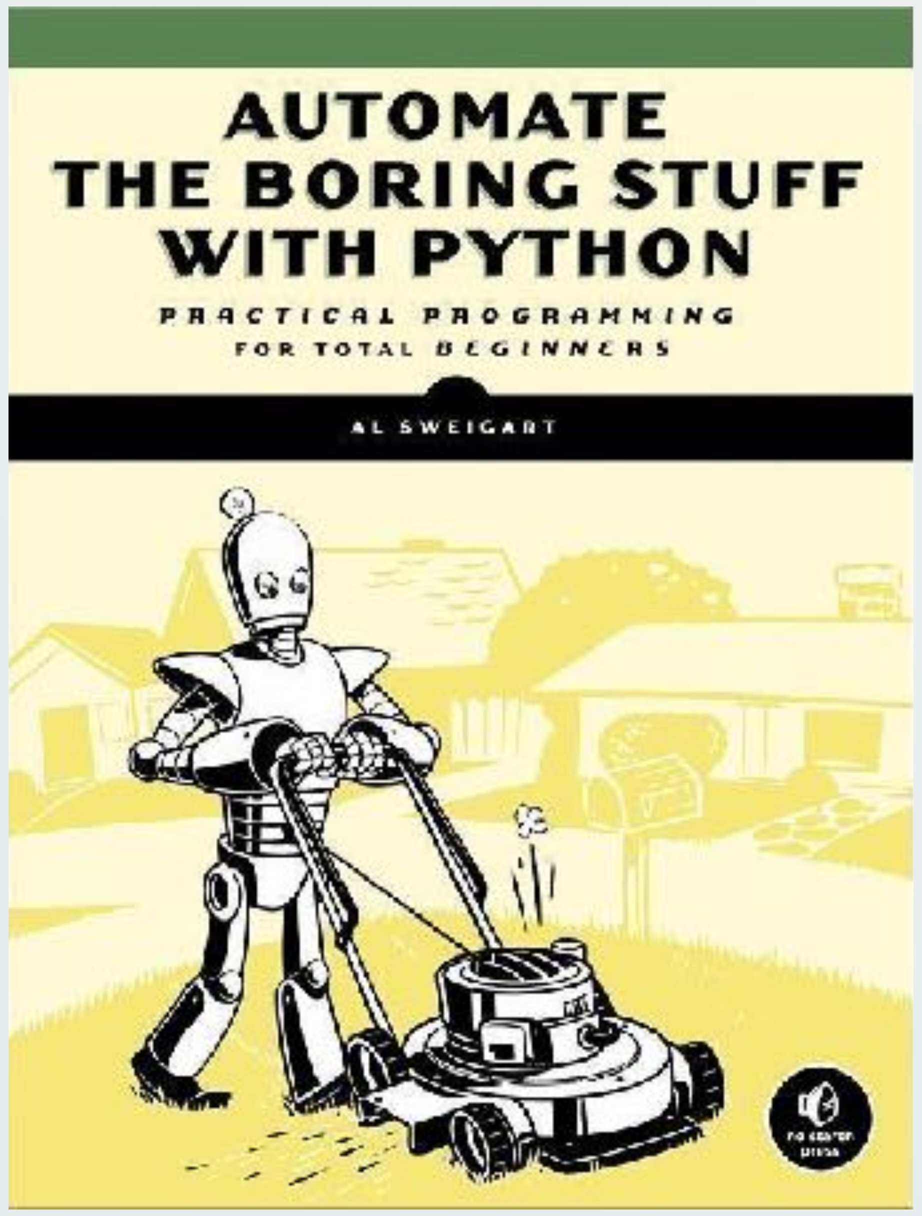
[1]: Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., ... & Lillicrap, T. (2017). Mastering chess and shogi by self-play with a general reinforcement learning algorithm.

Uses of AI

- Shiny « superhuman » algorithms make headlines

- But most applications « automate the boring stuff ».

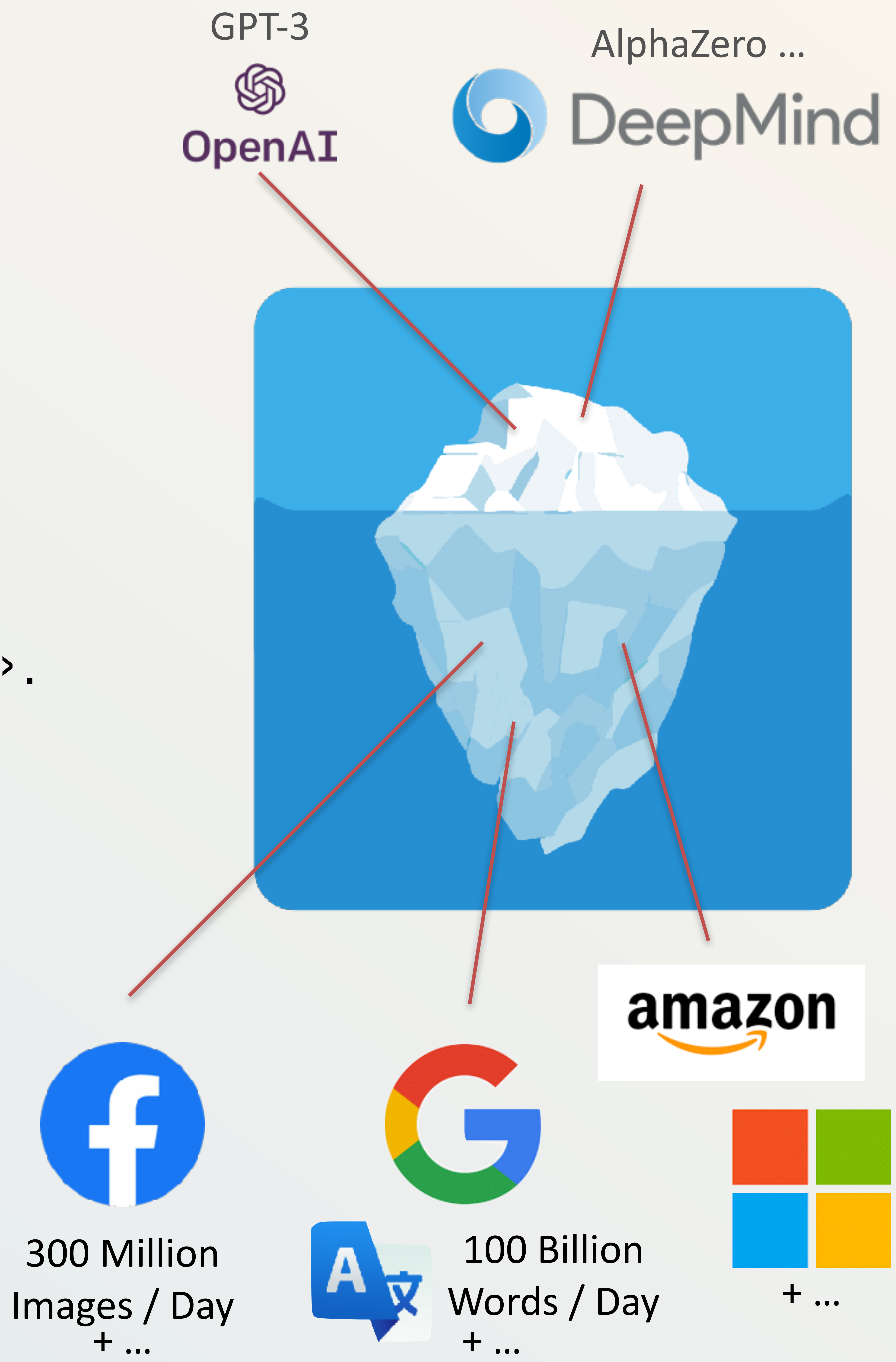
Just like regular programming does!



AUTOMATE THE BORING STUFF WITH PYTHON

PRACTICAL PROGRAMMING FOR TOTAL BEGINNERS

AL SWEIGART

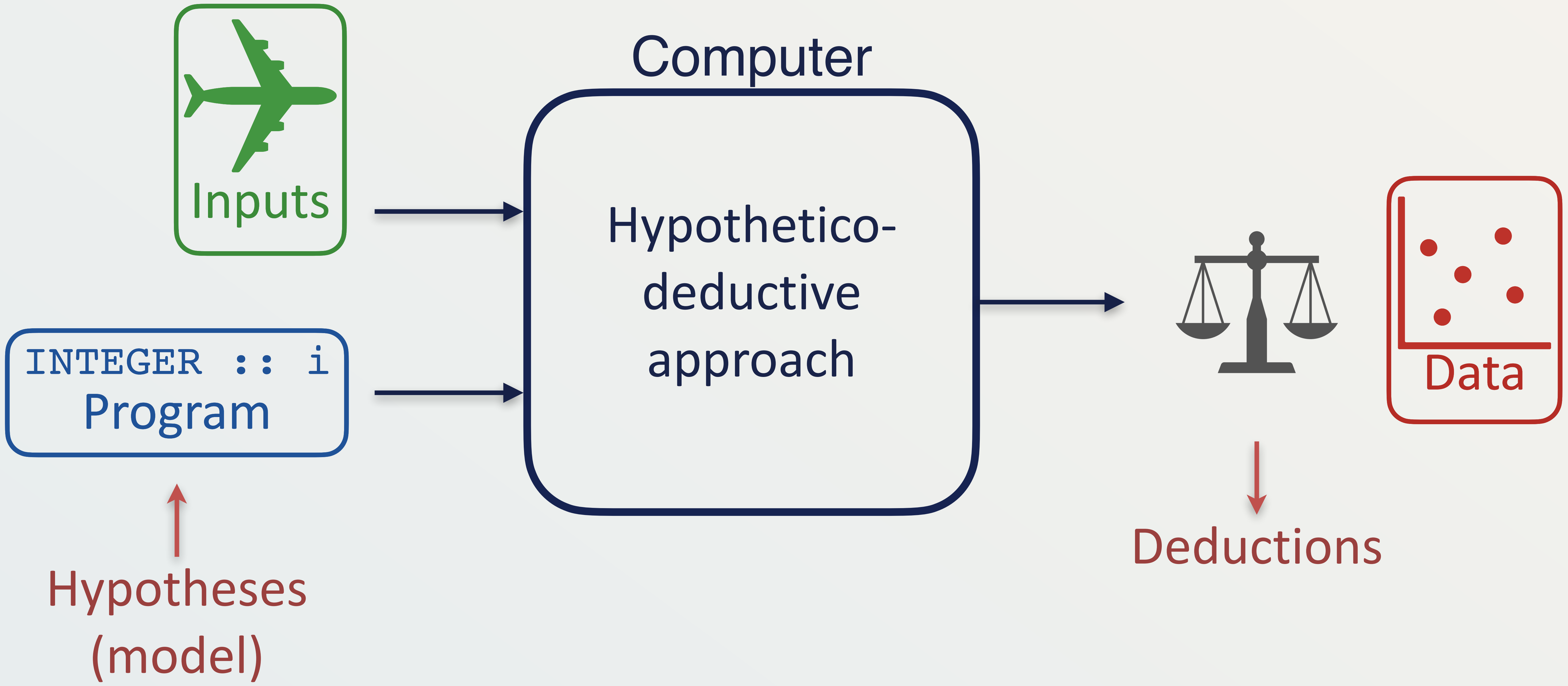


Just hype?

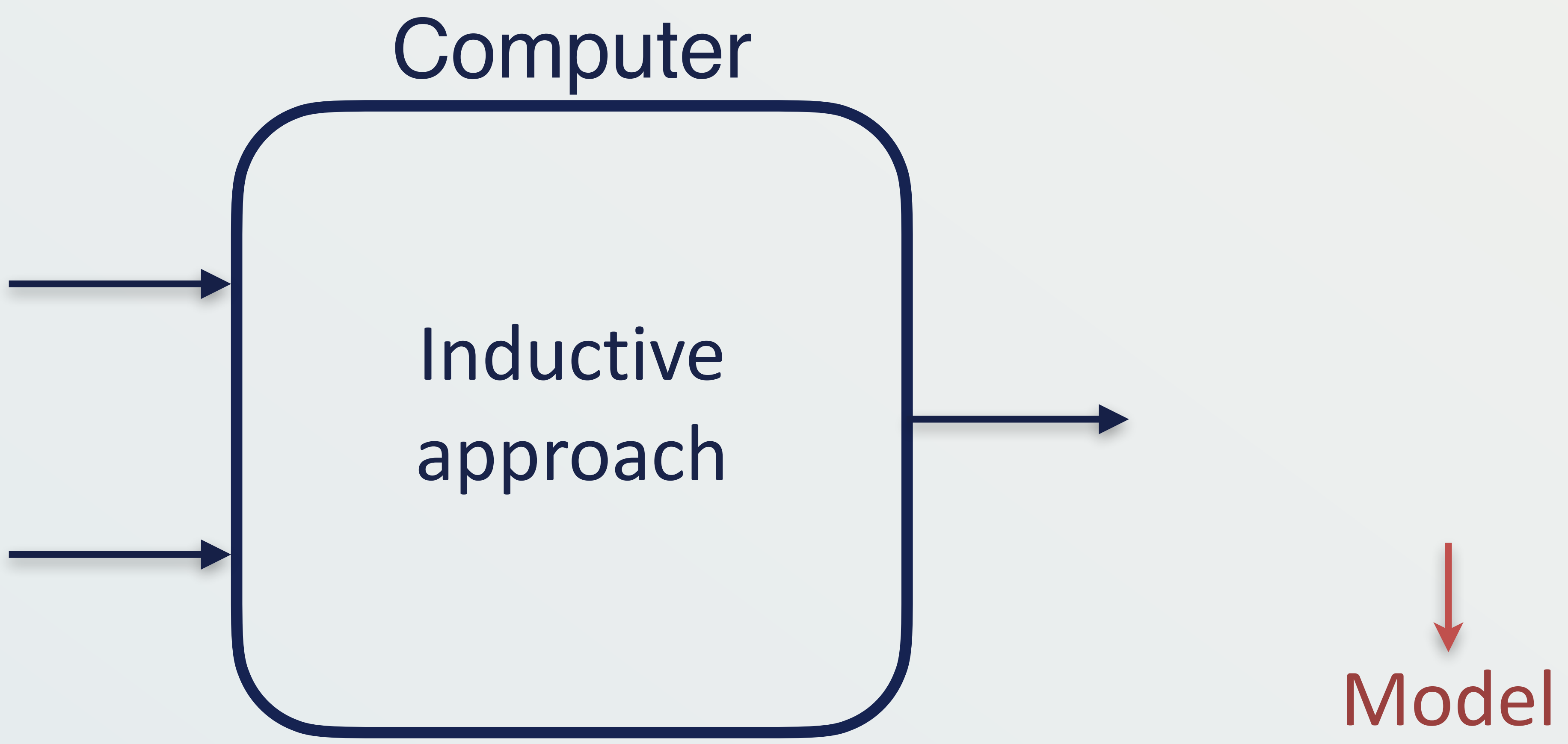
LA REVANCHE DES NEURONES
L'invention des machines inductives et la controverse de l'intelligence artificielle

Dominique CARDON
Jean-Philippe COINTET
Antoine MAZIÈRES

La Découverte | « Réseaux » 2018/5 n° 211 | pages 173 à 220



The scientific method is historically a deductive approach. **The data validates the model.**



Data-driven approaches are *inductive*. **The model is the output.**

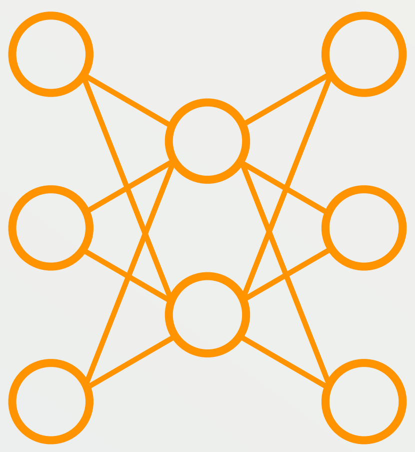
Breiman, L. (2001). Statistical modeling: The two cultures. *Statistical science*, *16*(3), 199-231.

A very brief history of AI

Turing

Hinton

LeCun



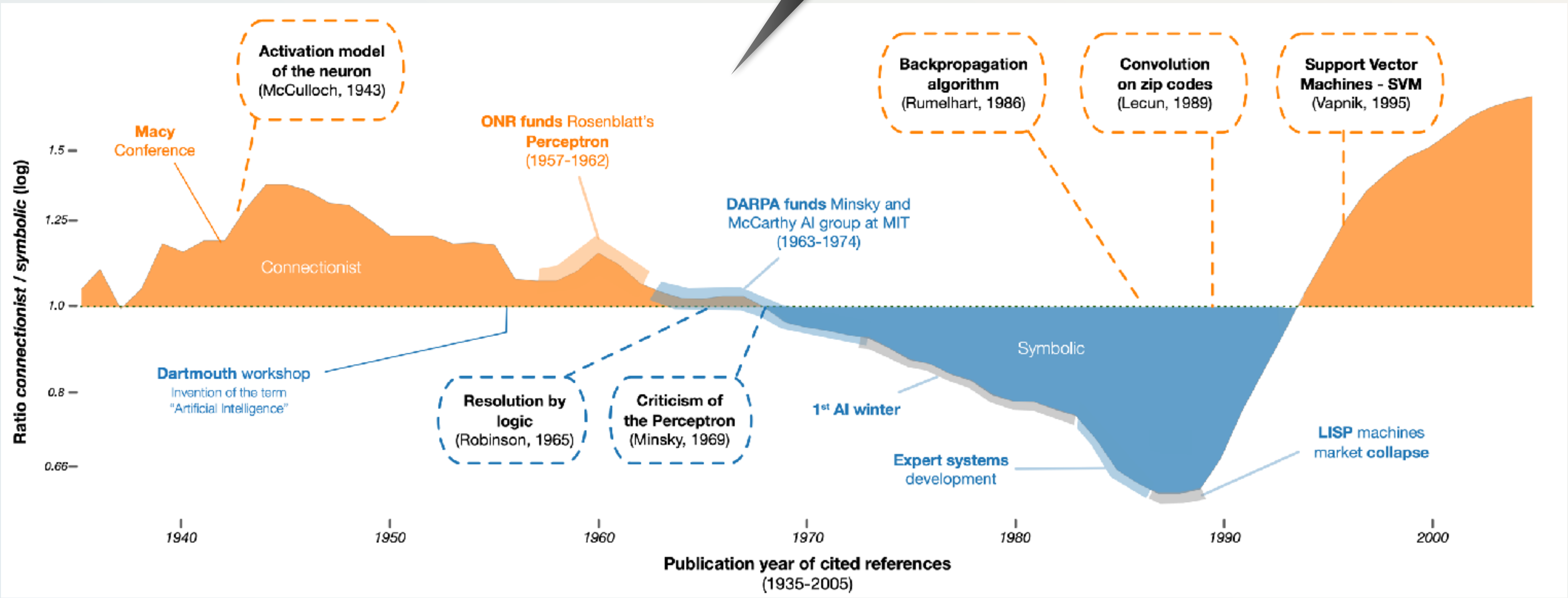
Connexionnist

Σ ∫ √

Symbolic

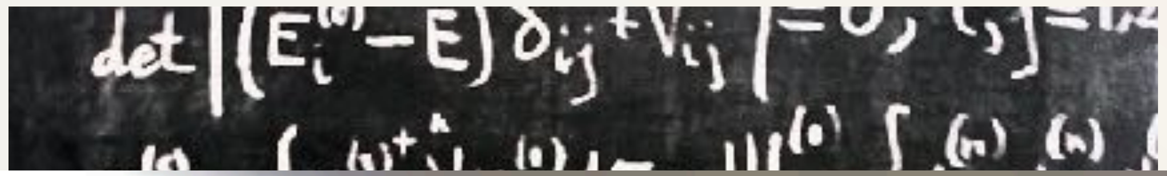
McCarthy

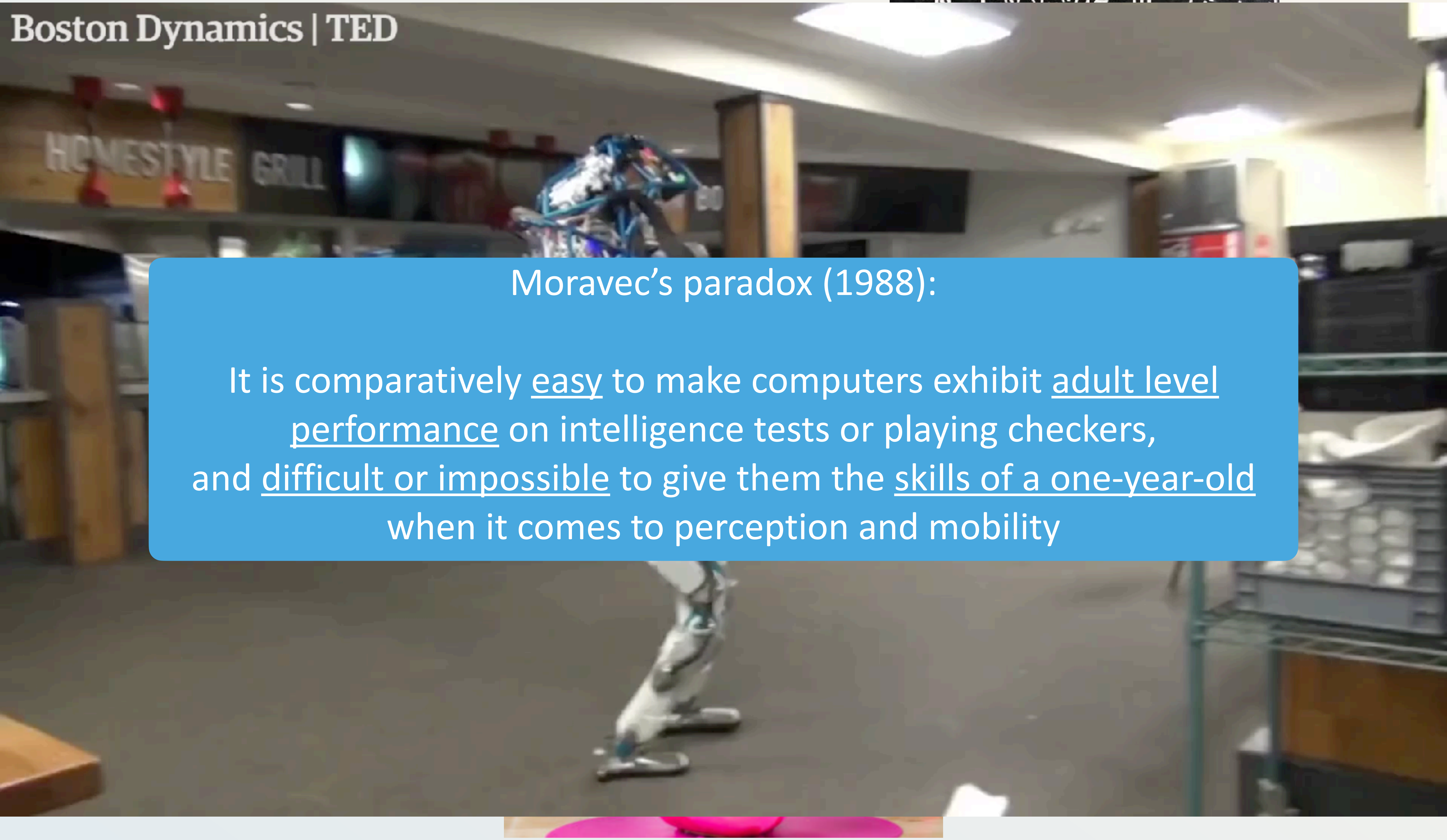
Minsky



Cardon, D., Cointet, J.-P. & Mazières, A (2018). La revanche des neurones. *Réseaux*, 211, 173-220.

What is *learning*?





Moravec’s paradox (1988):

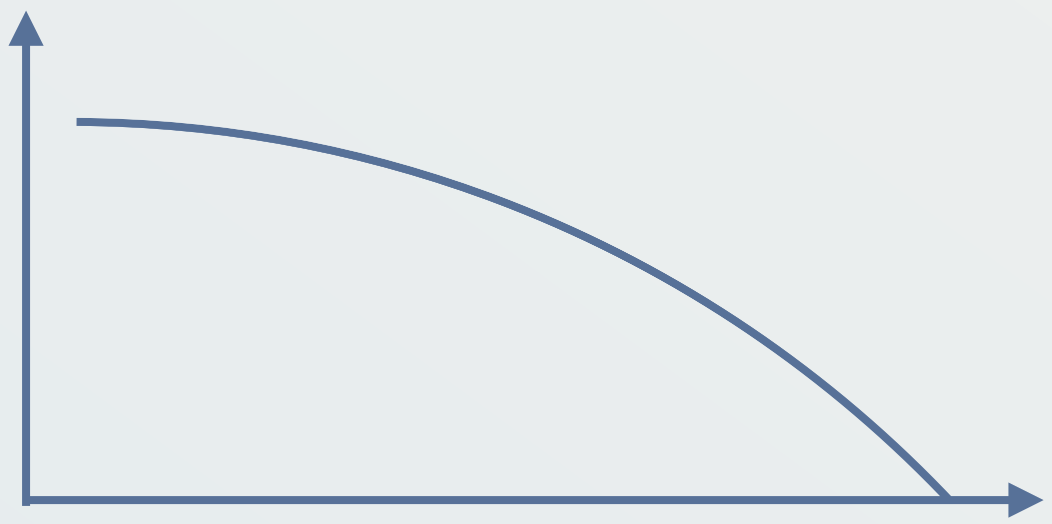
It is comparatively easy to make computers exhibit adult level performance on intelligence tests or playing checkers, and difficult or impossible to give them the skills of a one-year-old when it comes to perception and mobility

Learn motor skills

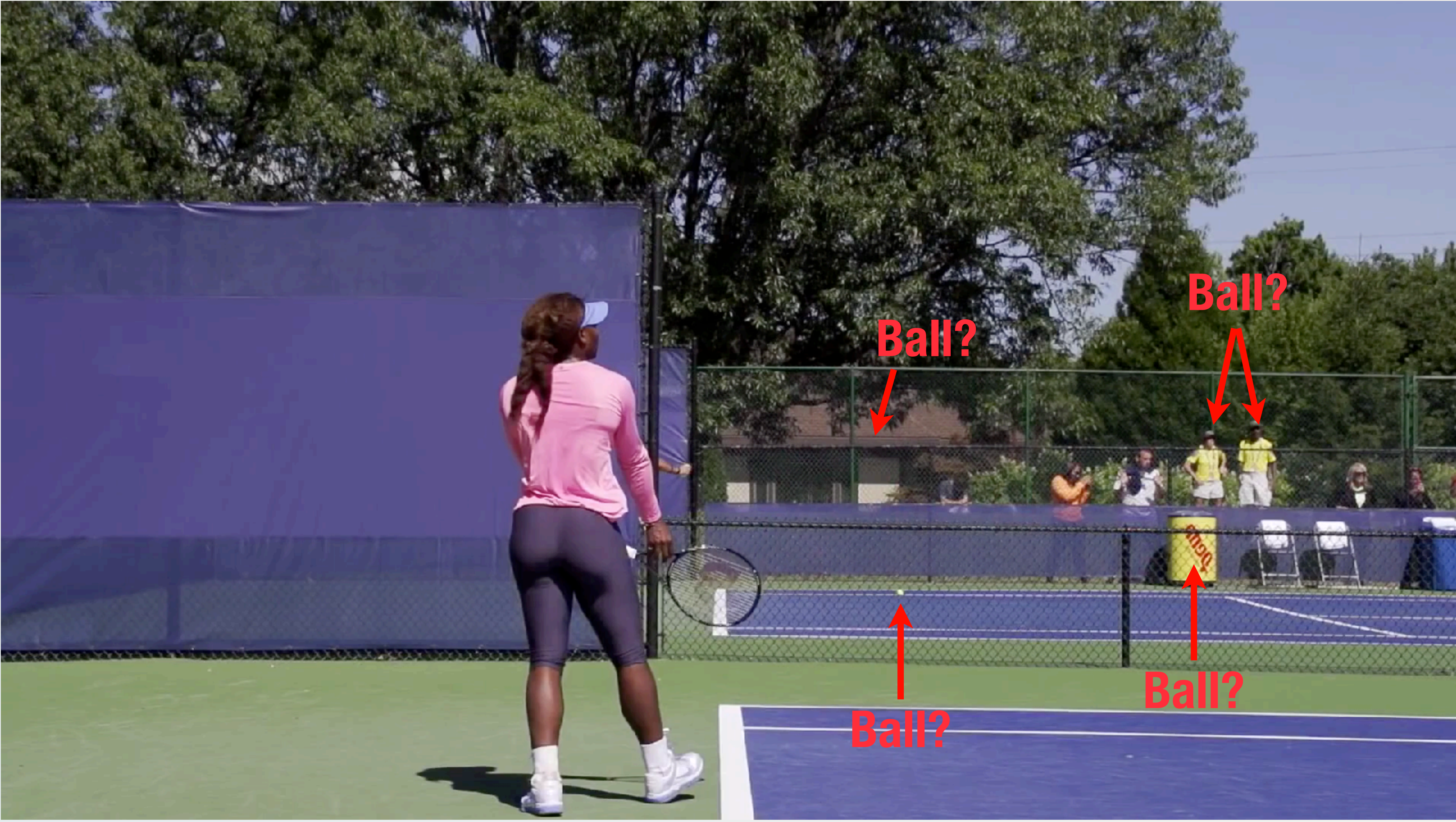
$$m\vec{a} = \vec{W} = m\vec{g}$$

or

$$m\vec{a} = \vec{W} + \vec{D}$$



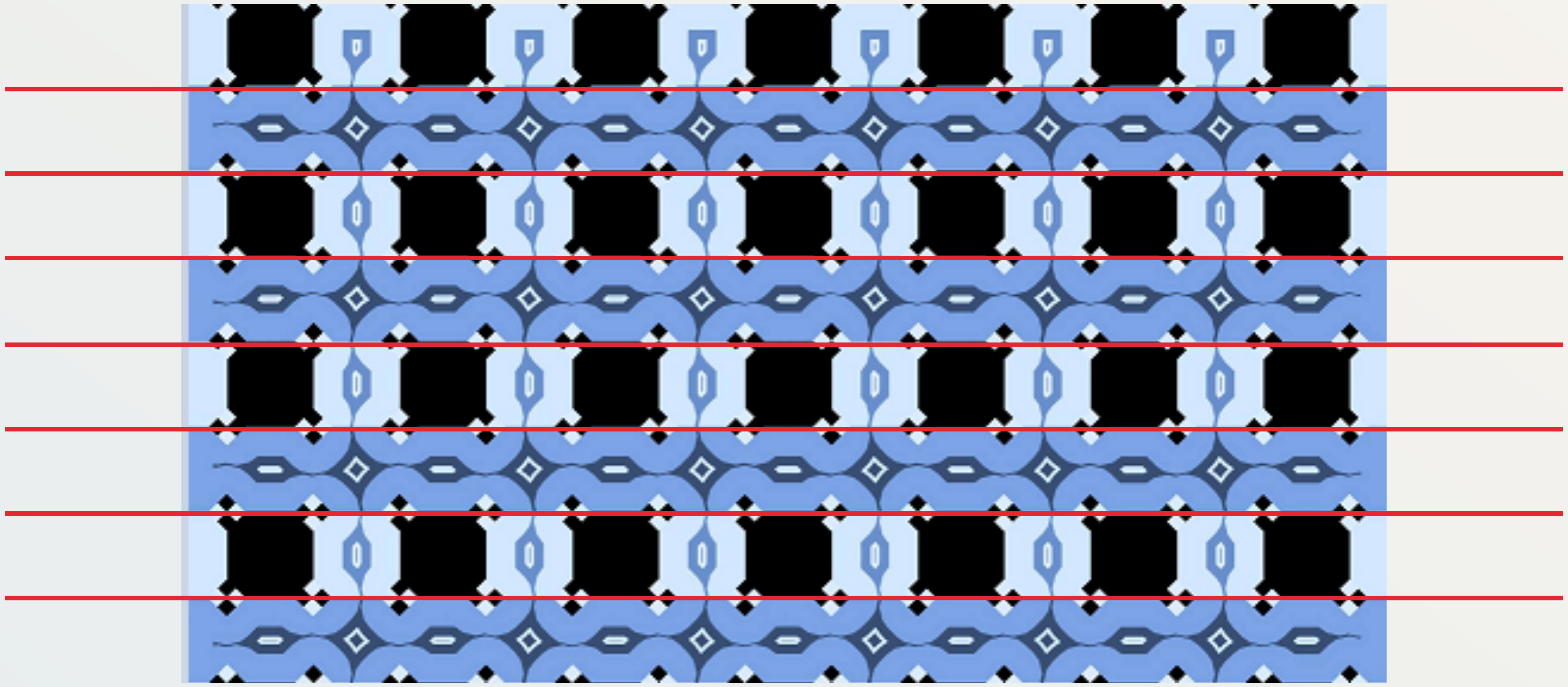
What a powerful tool!



Where's the ball?

- It's yellow
- I has texture
- It's round
- It's moving

Are these stripes straight?

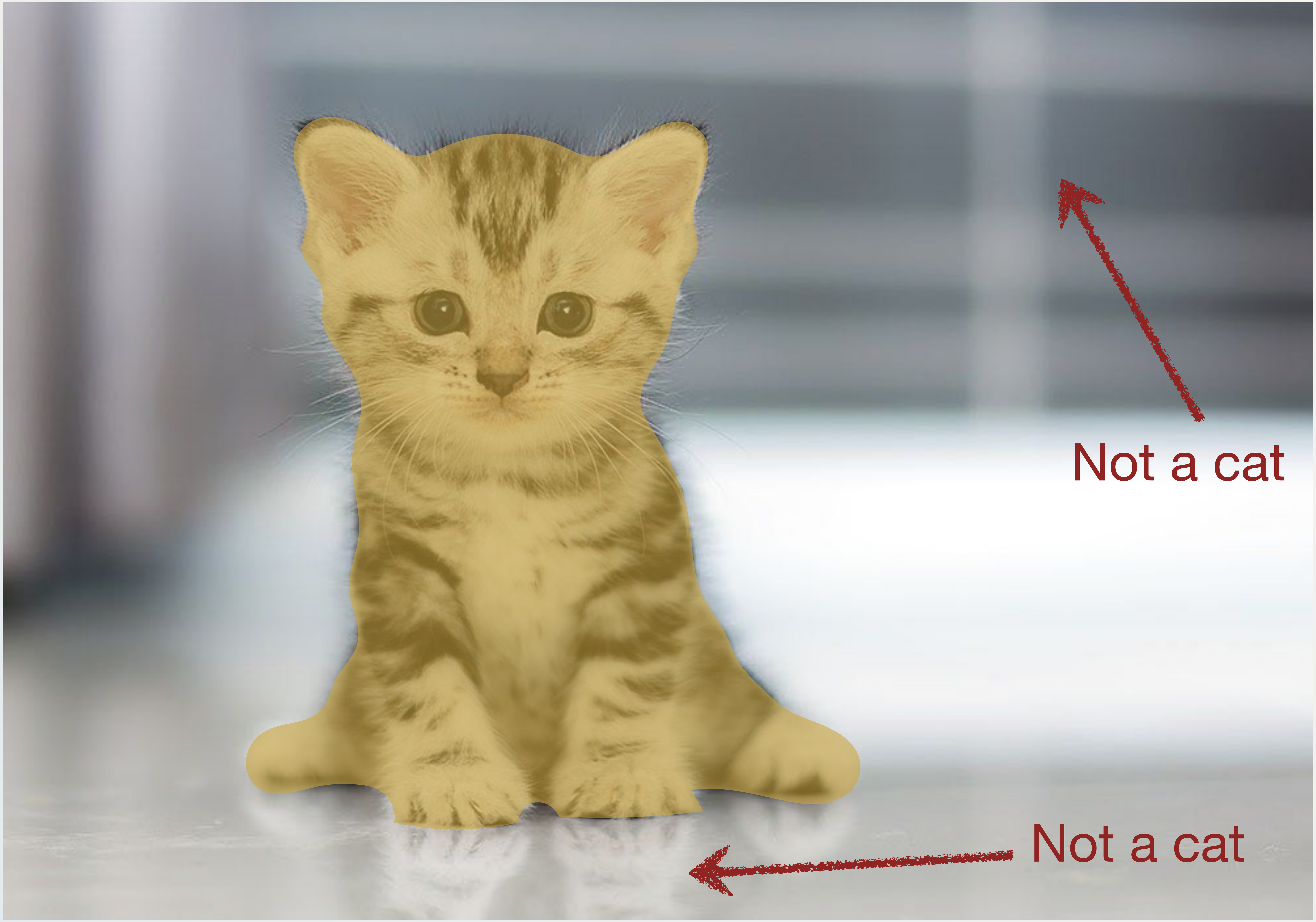


YES !

But do you <u>*see*</u> them straight?

No matter how long you watch, they'll never *look* straight…

Your brain performs complex unconscious processing.



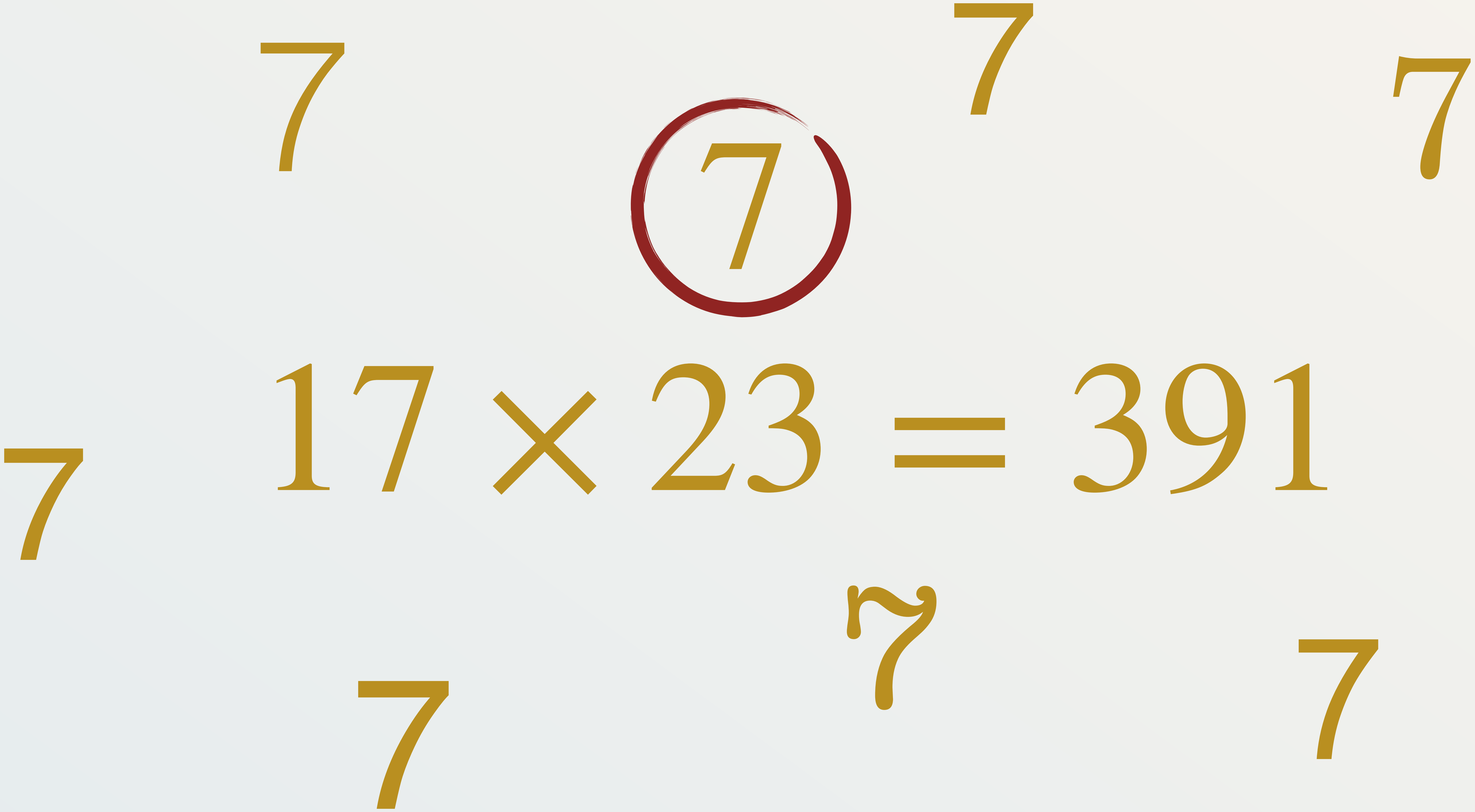
Where’s the cat?

$$17 \times 23 = ?$$

How much is this?

$$17 \times 23 = 391$$

How much is this?



$$17 \times 23 = 391$$

Which 7 did I use?

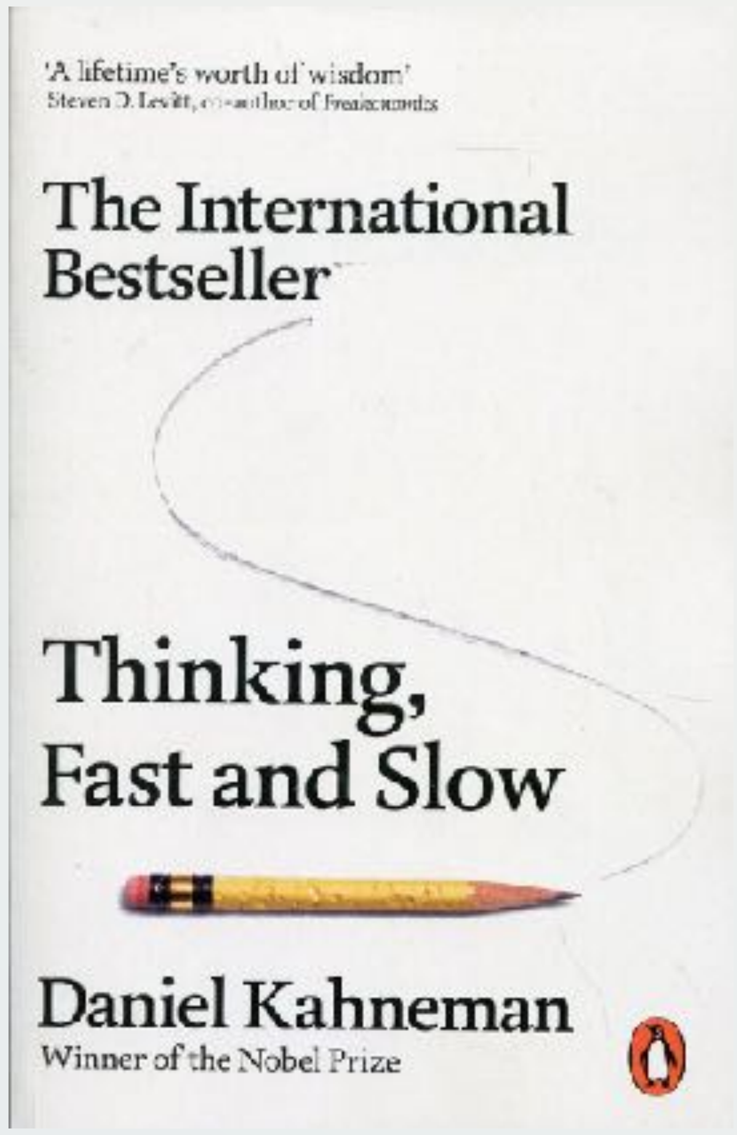
Coding works well for conscious tasks

but

We lack an approach to replicate tasks that we learn intuitively

Curious? You can start here:

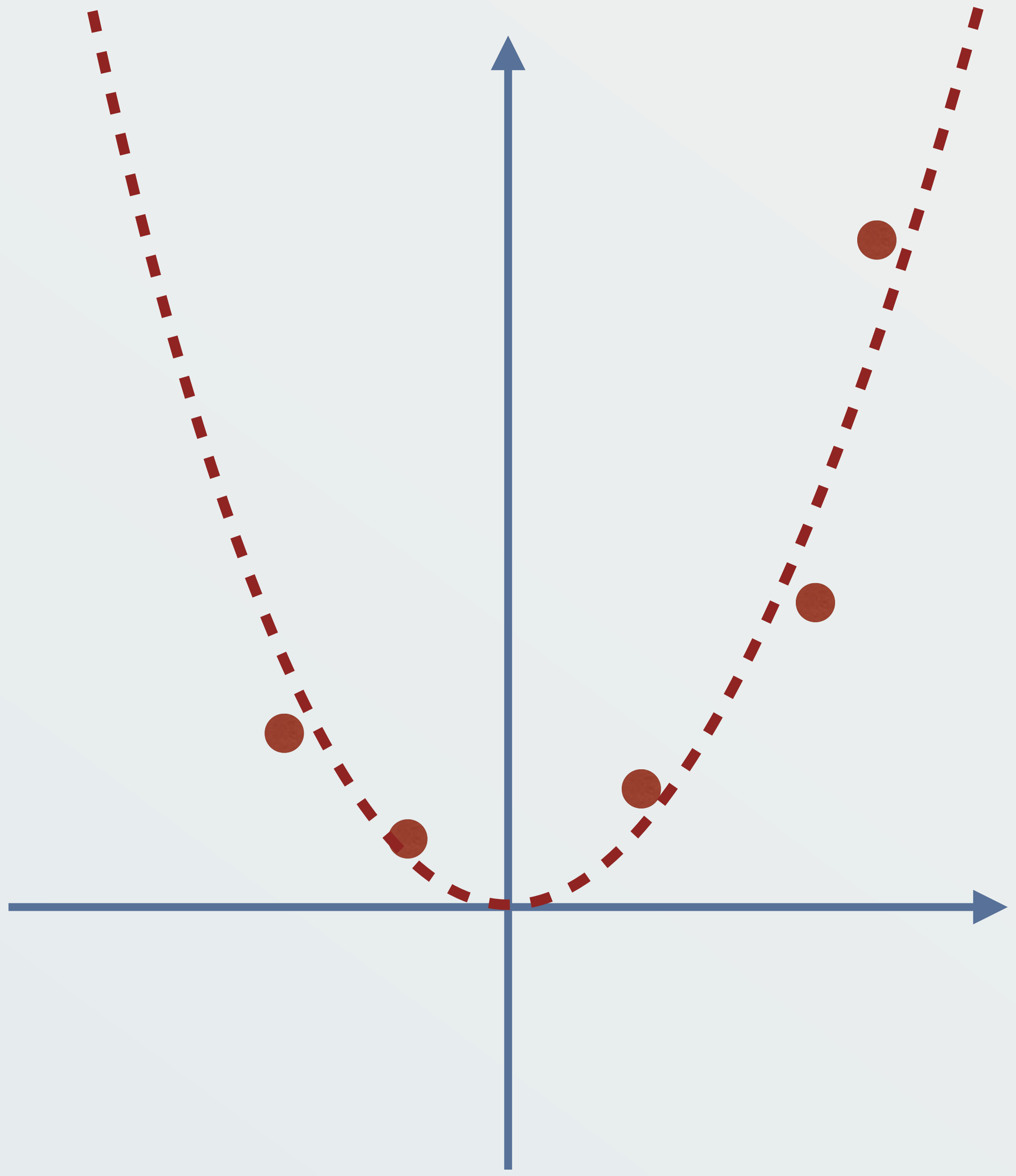
Kahneman, Daniel. Thinking, fast and slow. Macmillan, 2011.


'A lifetime's worth of wisdom'
Steven D. Levitt, co-author of Freakonomics
The International Bestseller
Thinking, Fast and Slow
Daniel Kahneman
Winner of the Nobel Prize

Nobel prize 2002

Machines that learn ?

Machine learning



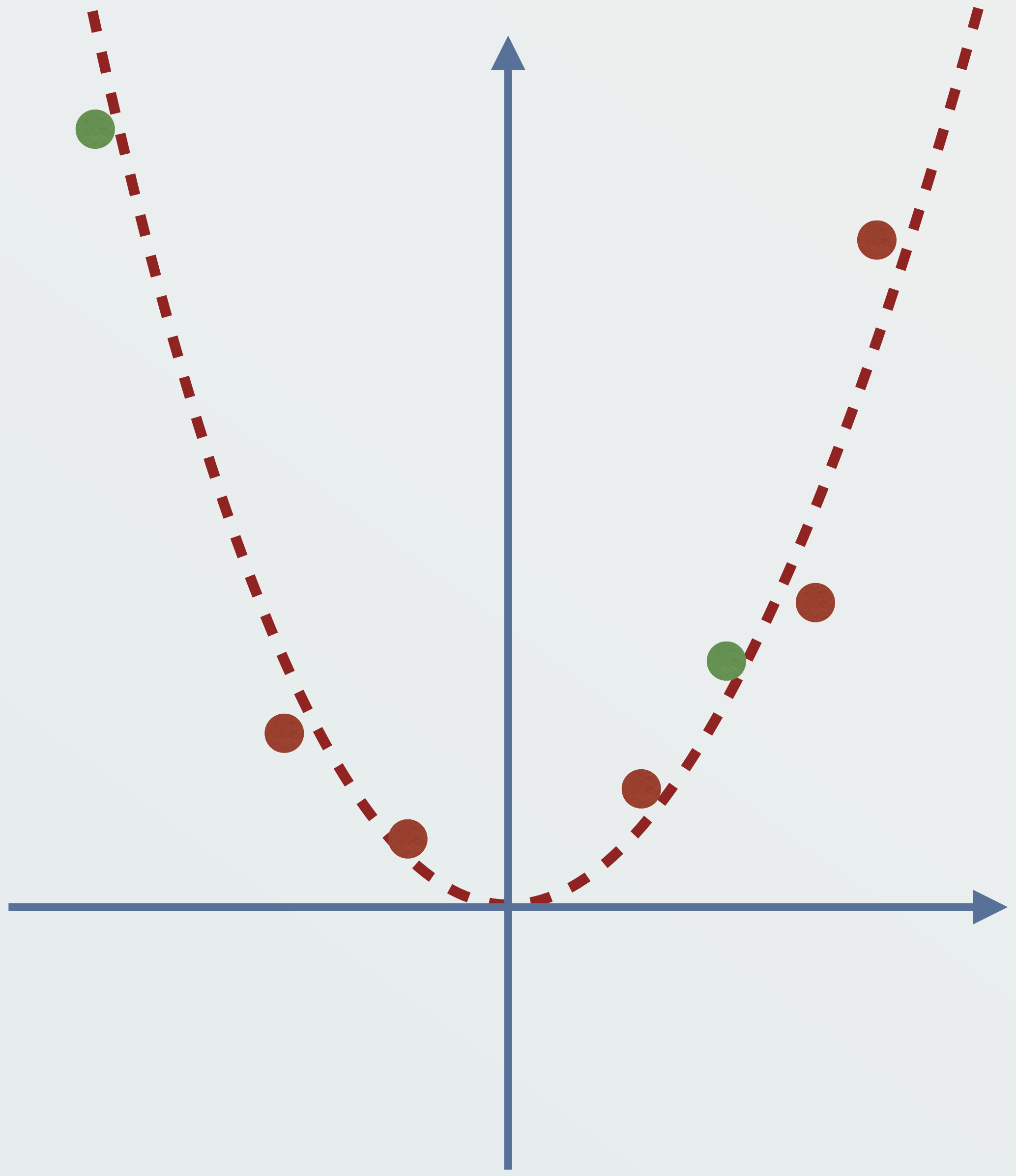
Rule based systems $y = 3x^2$

Knowledge based models $y = ax^2 + b$

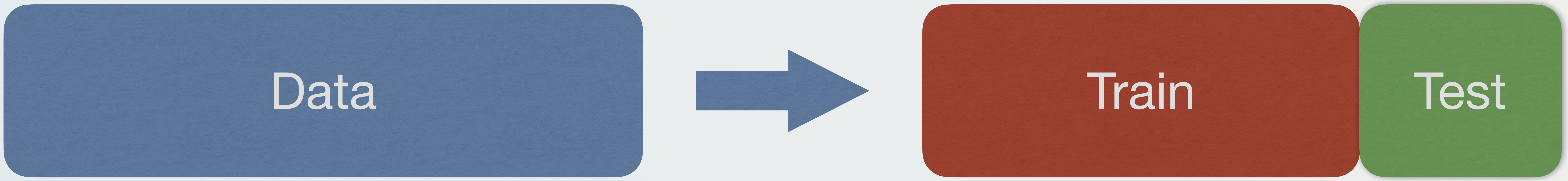
Machine learning $y = \sum_{i=0}^{N} a_i x^i$

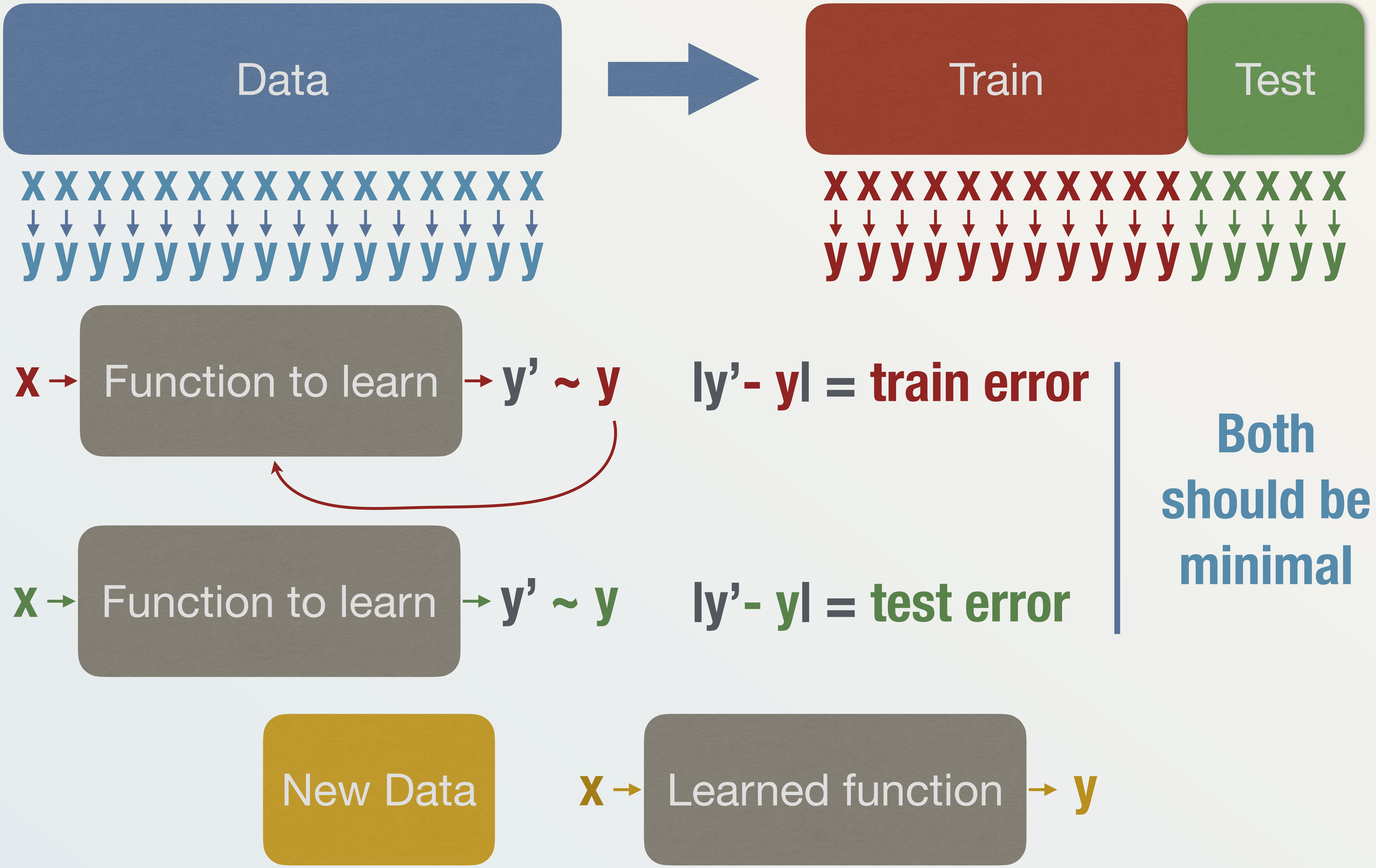
Machine learning means selecting a *class of functions*.
The exact shape (not just coefficients) is learned from the data

Objective: generalization



- Machine learning strategy:
  - observe the **training** data to learn the **features**
  - **do not** learn the **noise**
  - generalize well to the **test** data (good prediction)





Data

Train

Test

x x x x x x x x x x x x x x x x

y y y y y y y y y y y y y y y y

x → Function to learn → y’ ~ y

|y’- y| = train error

x → Function to learn → y’ ~ y

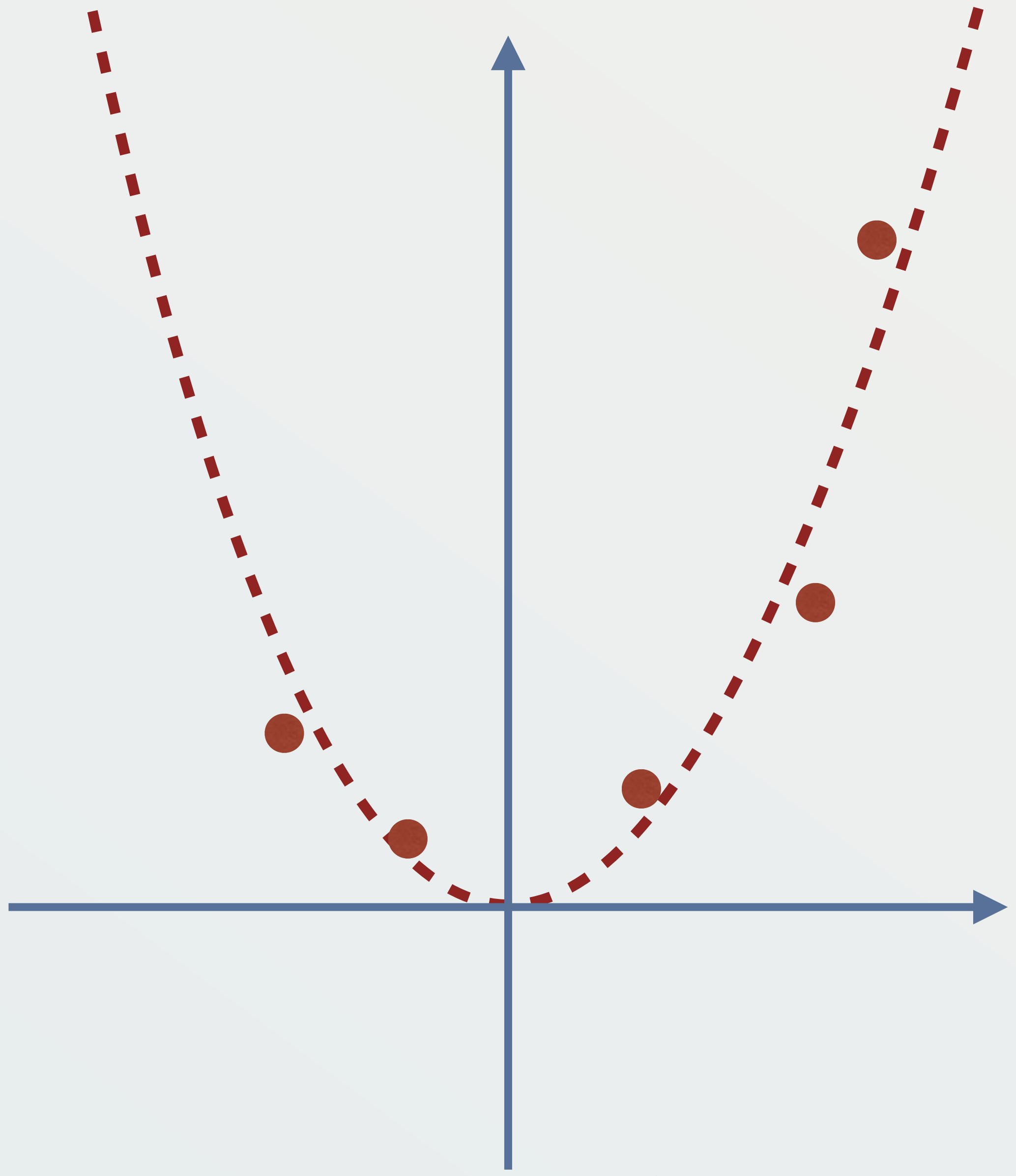
|y’- y| = test error

Both should be minimal

New Data

x → Learned function → y

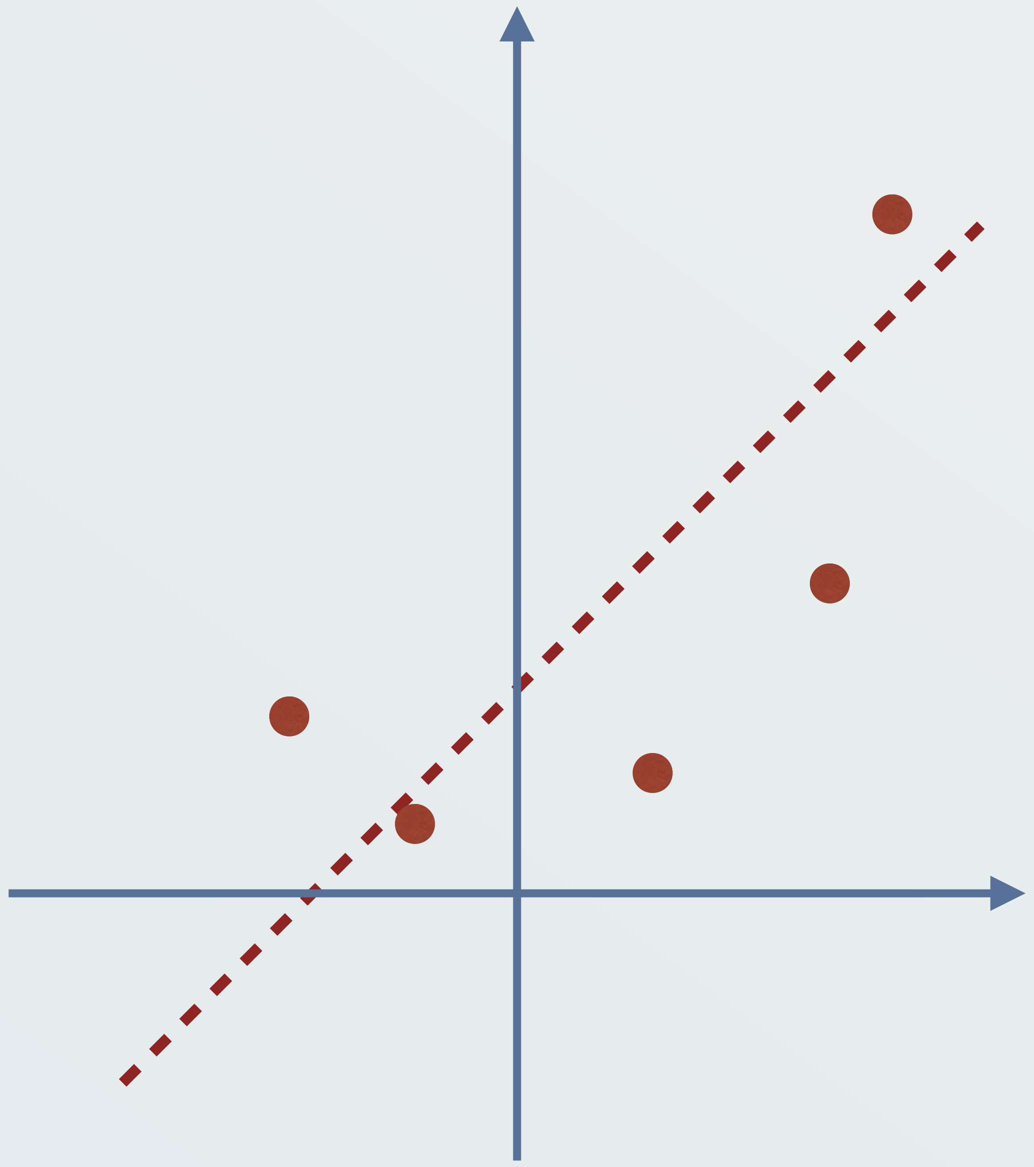
Under / Overfitting



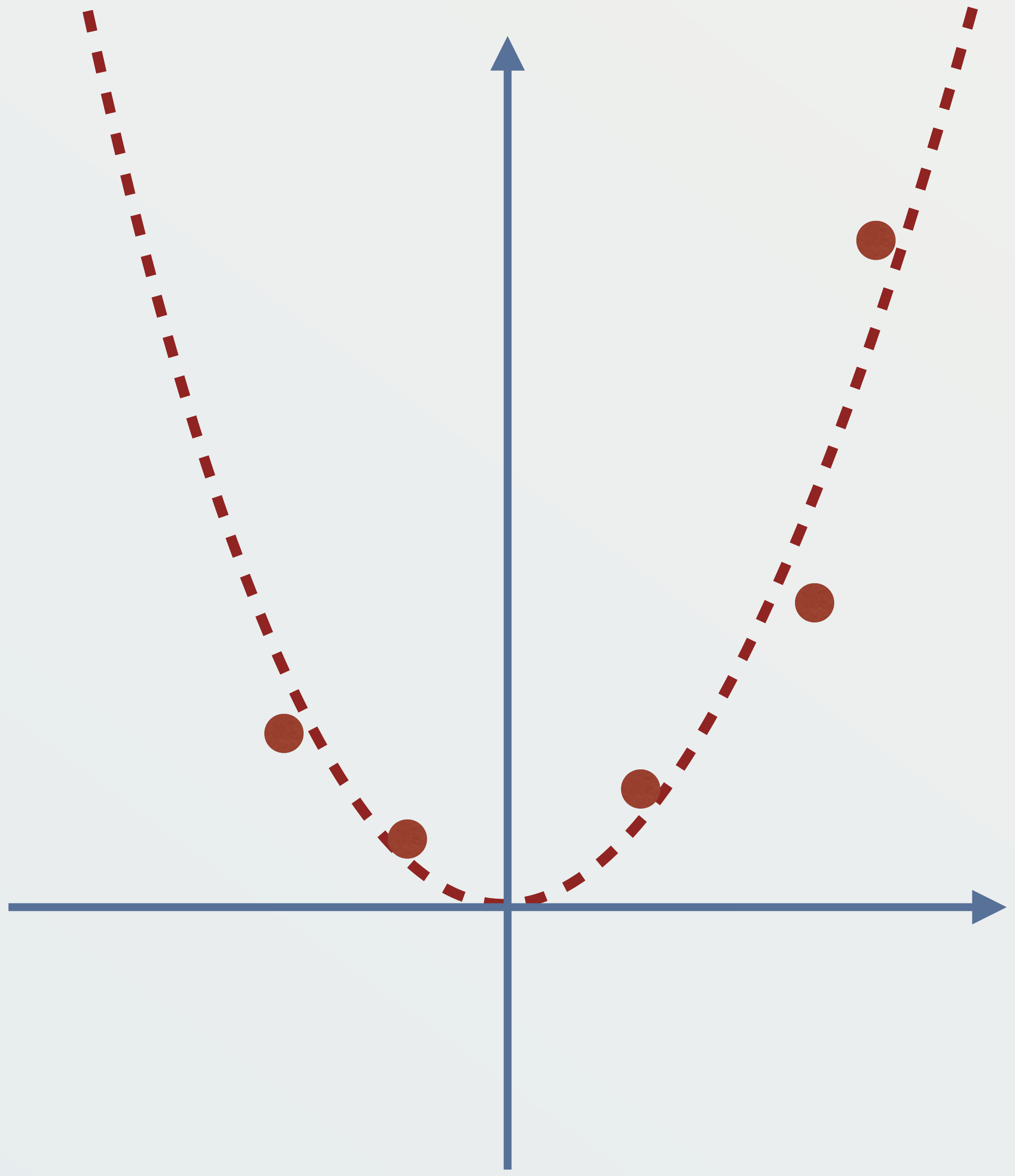
Human fitting :

« Hey, this looks like a 2nd order polynomial »

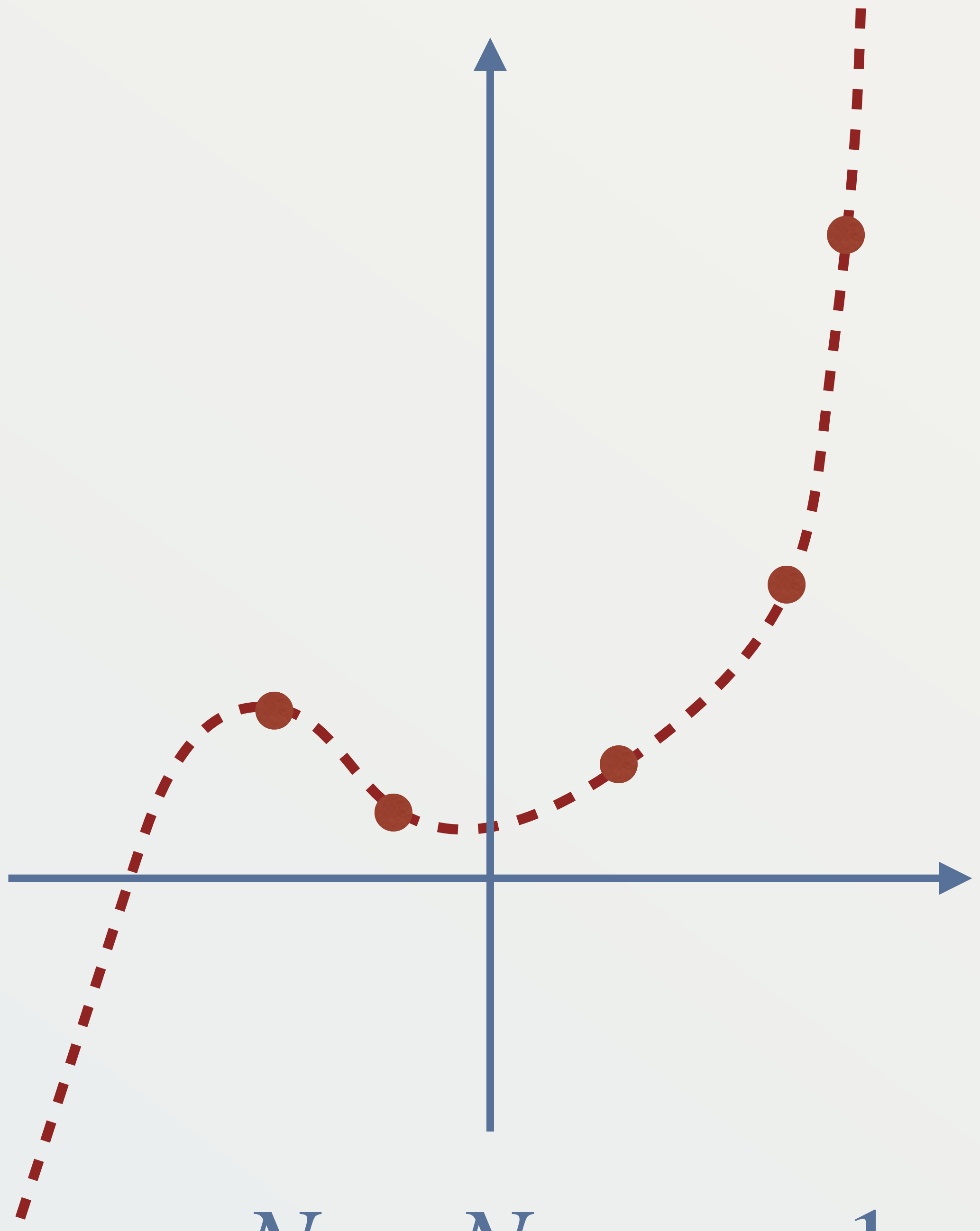
Learned fit : $y = \sum_{i=0}^{N} a_i x^i \qquad N = ?$



$N = 1$



$N = 2$



$N = N_{points} - 1$

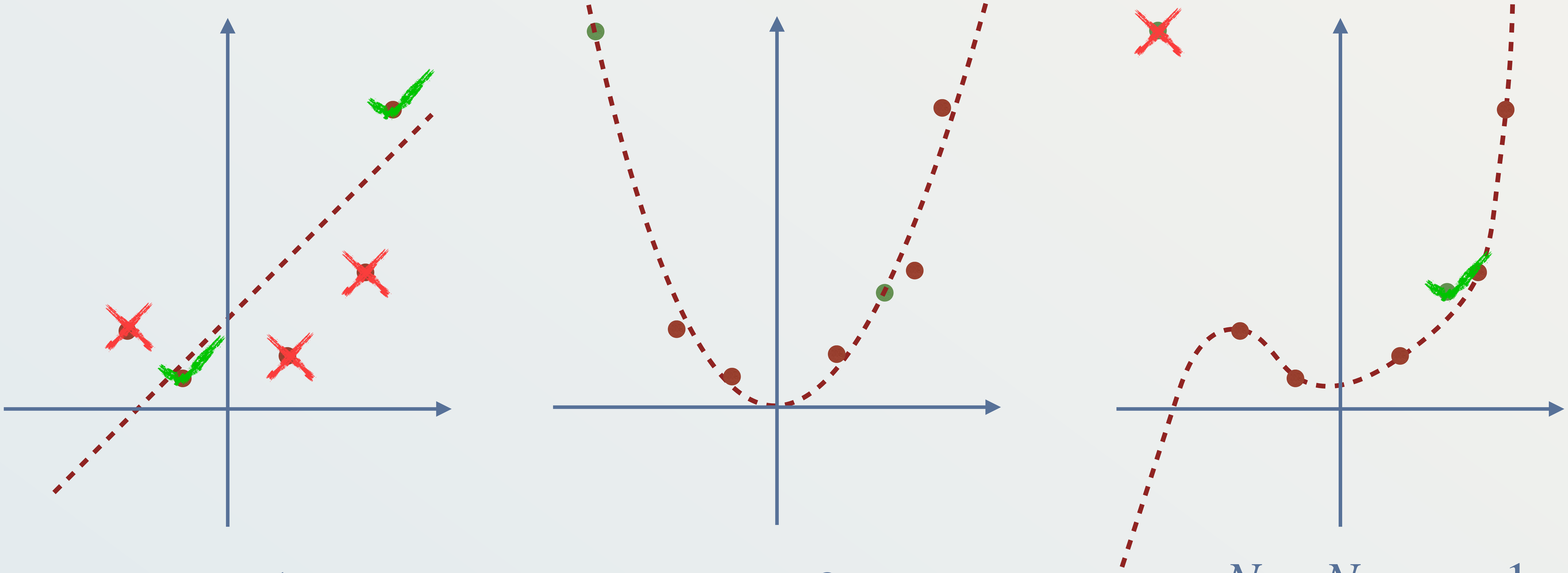
Under / Overfitting

To know what works, you measure the error on both the train and test sets. Here it seems the right *capacity* N is 2.

Training error

Correct capacity

Testing error

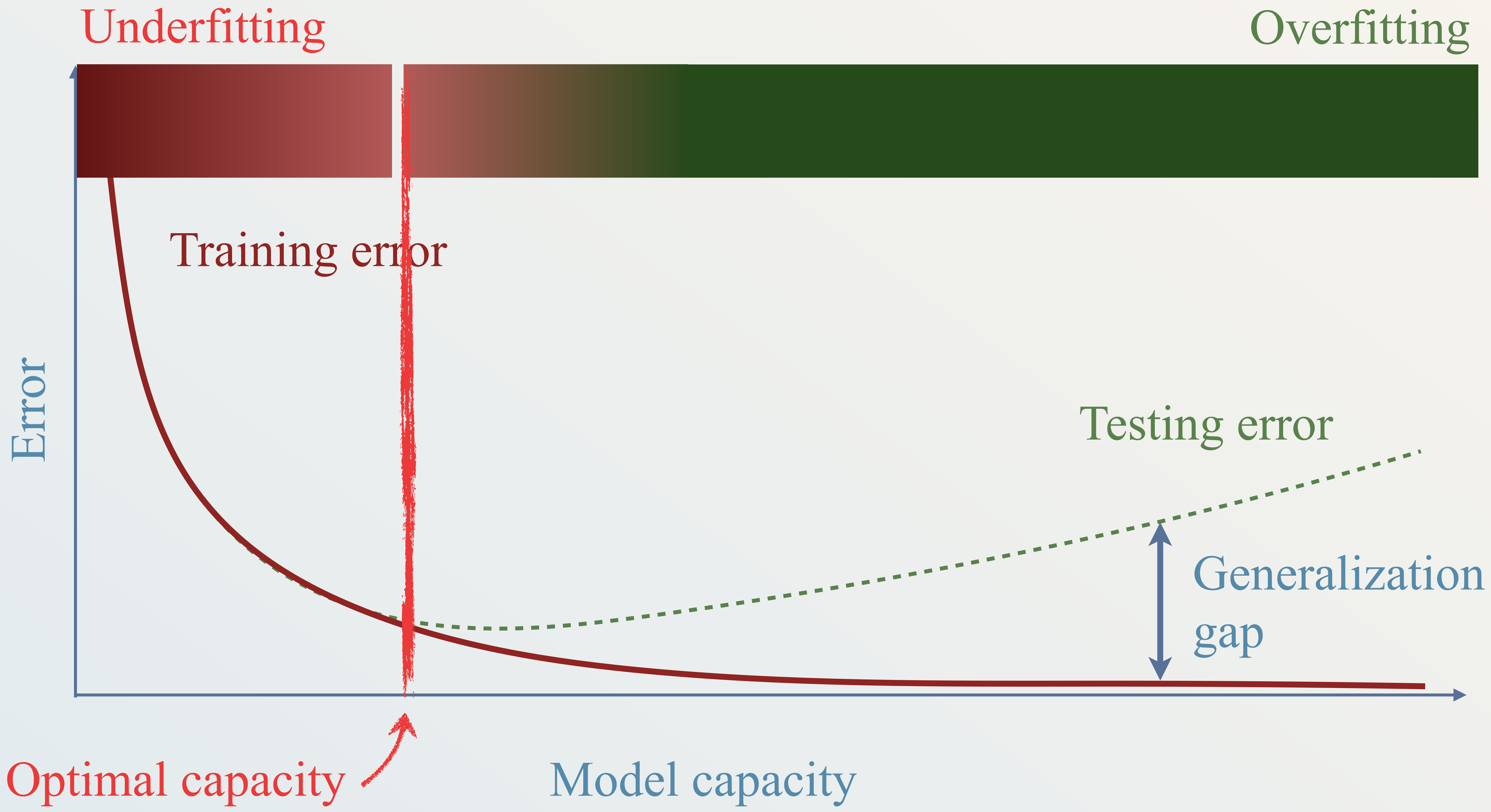


$N = 1$

$N = 2$

$N = N_{points} - 1$

Under / Overfitting



At it’s heart: *Bayesian Inference:*

(just like humans! [1])

Hypothesis H

Evidence E

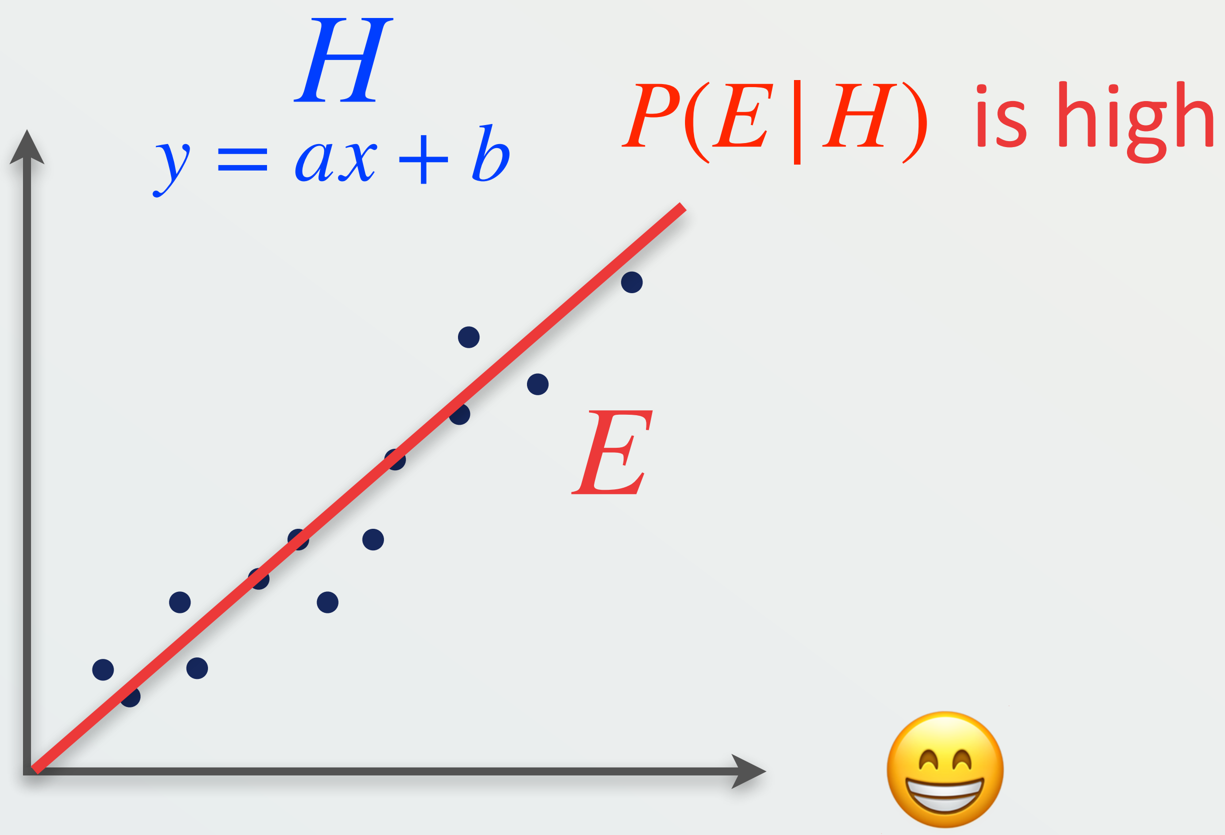
$$\underbrace{P(H|E)}_{\text{Posterior}} \propto \underbrace{P(E|H)}_{\text{Likelihood}} \cdot \underbrace{P(H)}_{\text{Prior}}$$

- Procedure:
  - ◉ Start with *prior beliefs* (H)
  - ◉ Compute the *likelihood of evidence* (E) given (H)
  - ◉ Using Bayes’ rule, obtain new posterior beliefs of hypothesis given evidence P(H|E)

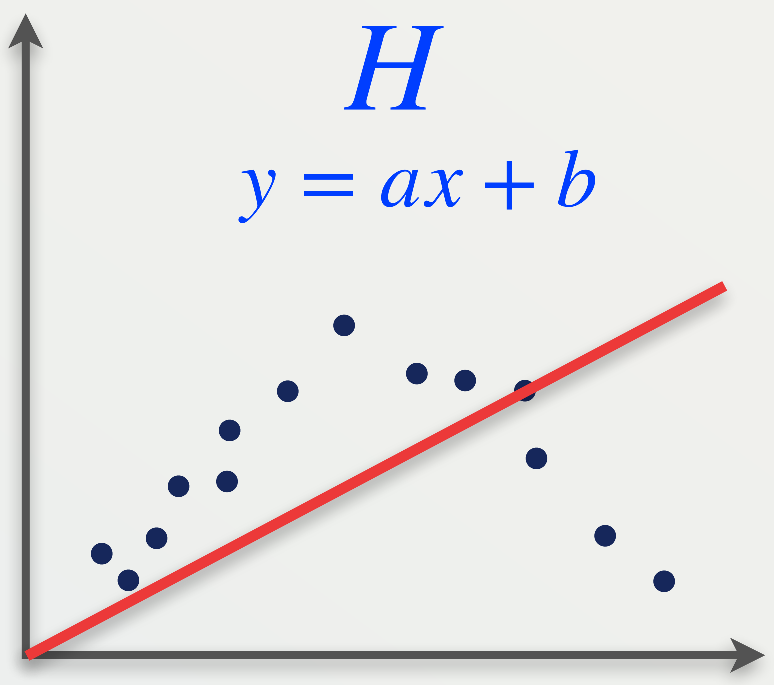
- Automating this process leads to selecting the *beliefs* best supported by *evidence*, *i.e. learning*

[1] Dehaene, S. (2020). How We Learn: Why Brains Learn Better Than Any Machine... for Now. Penguin.

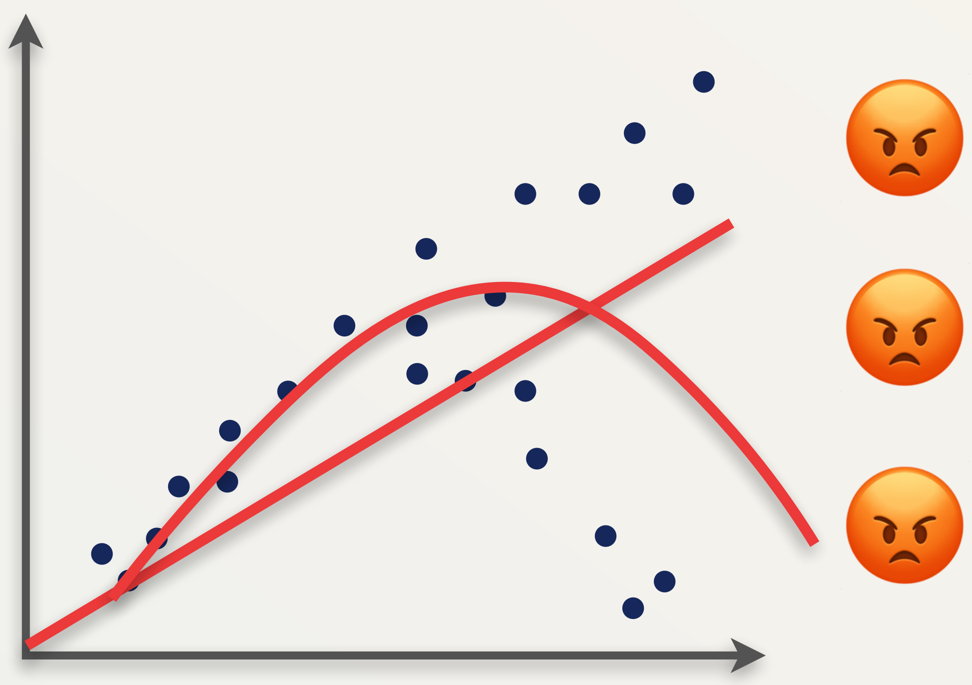
Machine Learning is the automated process of searching for the *hypothesis* H that leads to the *maximum likelihood* of the evidence E given H: P(E|H), hence the best posterior P(H|E)



Conclusion: my hypothesis is supported by the data, so I'm now more confident in it

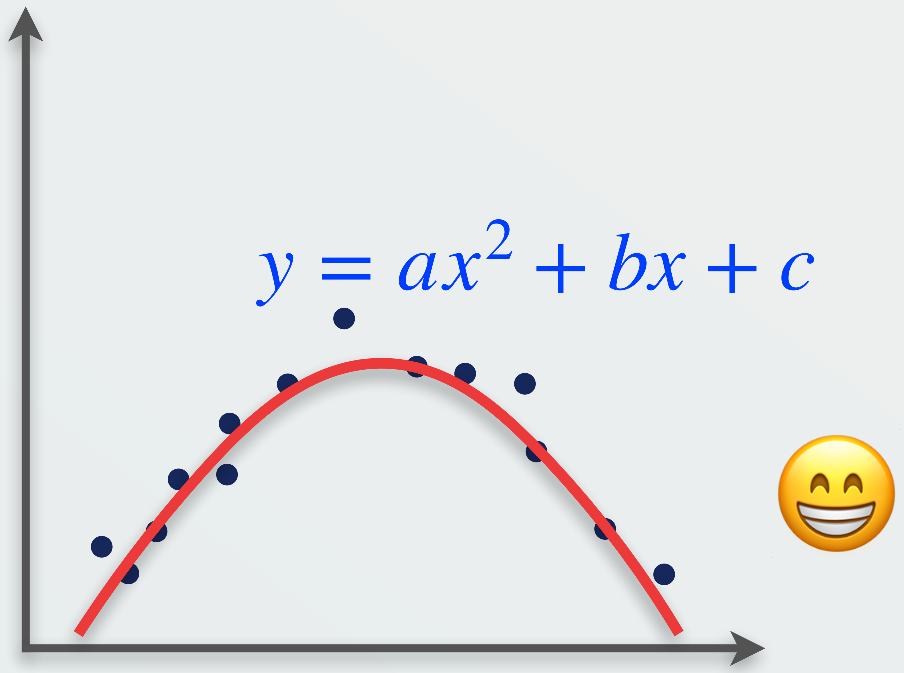


Conclusion: 😡
The data doesn't support H



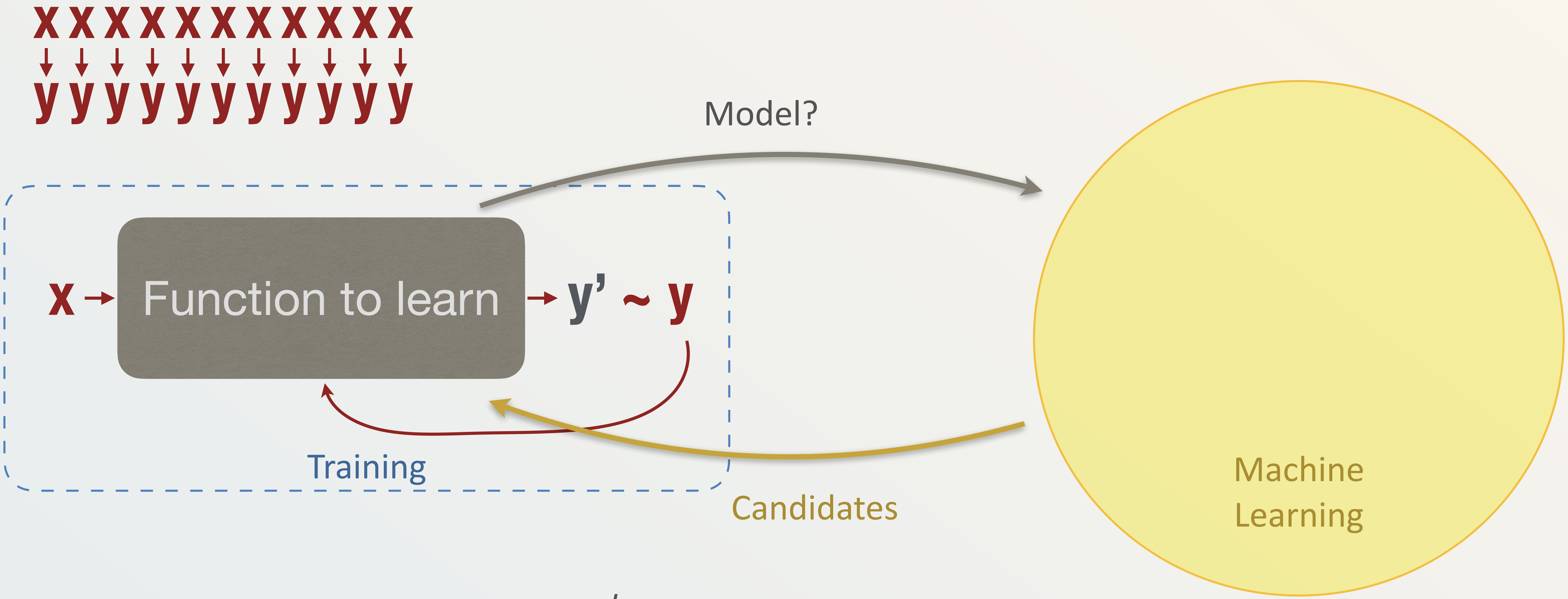
Conclusion:
Nothing works!

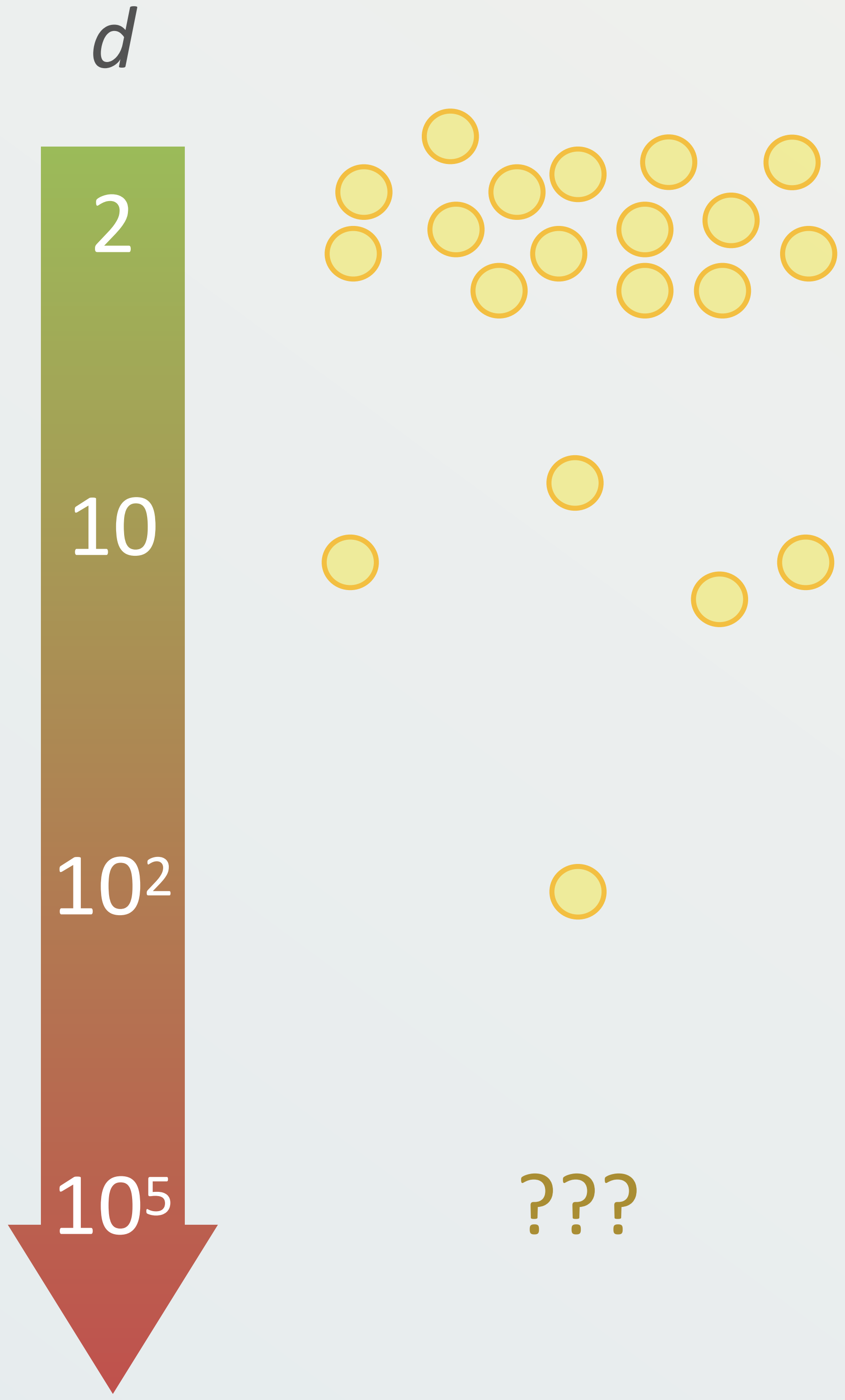
The no-free lunch theorem [1]
« there are no a priori distinctions between learning algorithms »



Some problems are ill-posed:
There is a fundamental ambiguity that cannot be resolved

[1] Wolpert, David H. "The lack of a priori distinctions between learning algorithms." Neural computation 8.7 (1996): 1341-1390.





<u>Model?</u>
N number of samples
d dimension of *X*
…

Can I interest you in a dimensionality reduction technique?

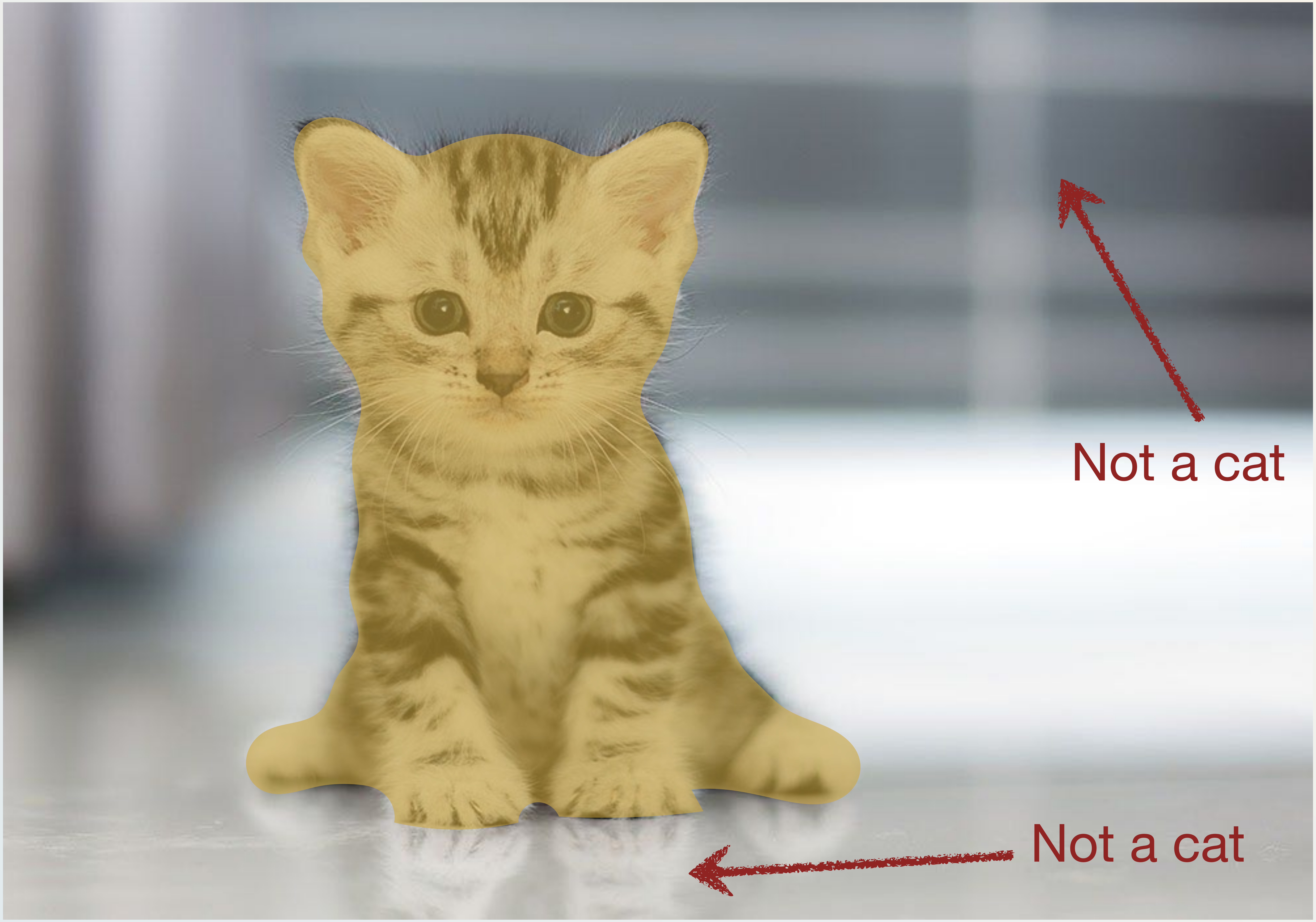
Clustering

PCA

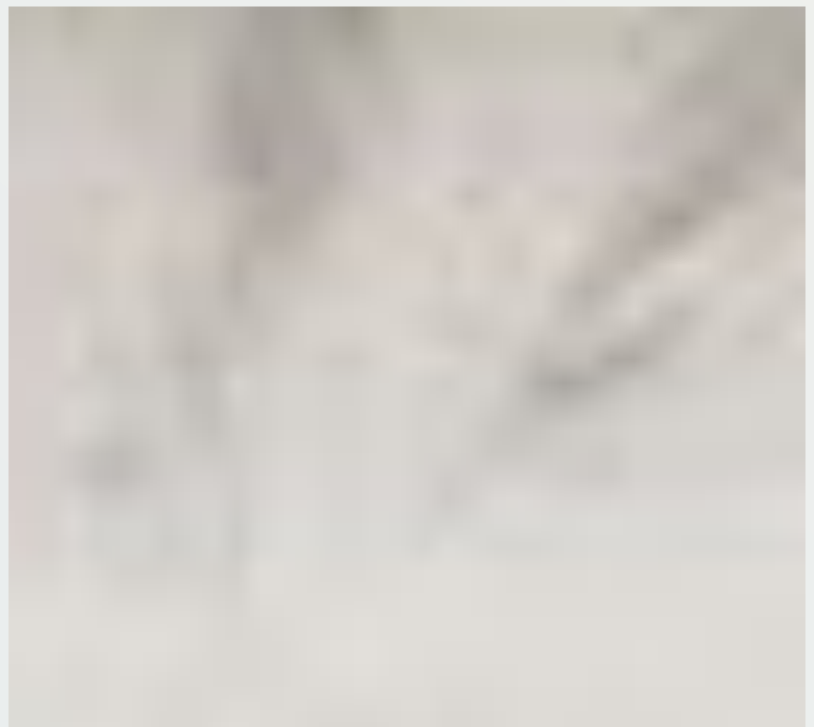
SOM

POD

…



What pixels belong to the cat?



Cat? No

Jeez, that’s hard!

How come you knew before?

Cat? Yes

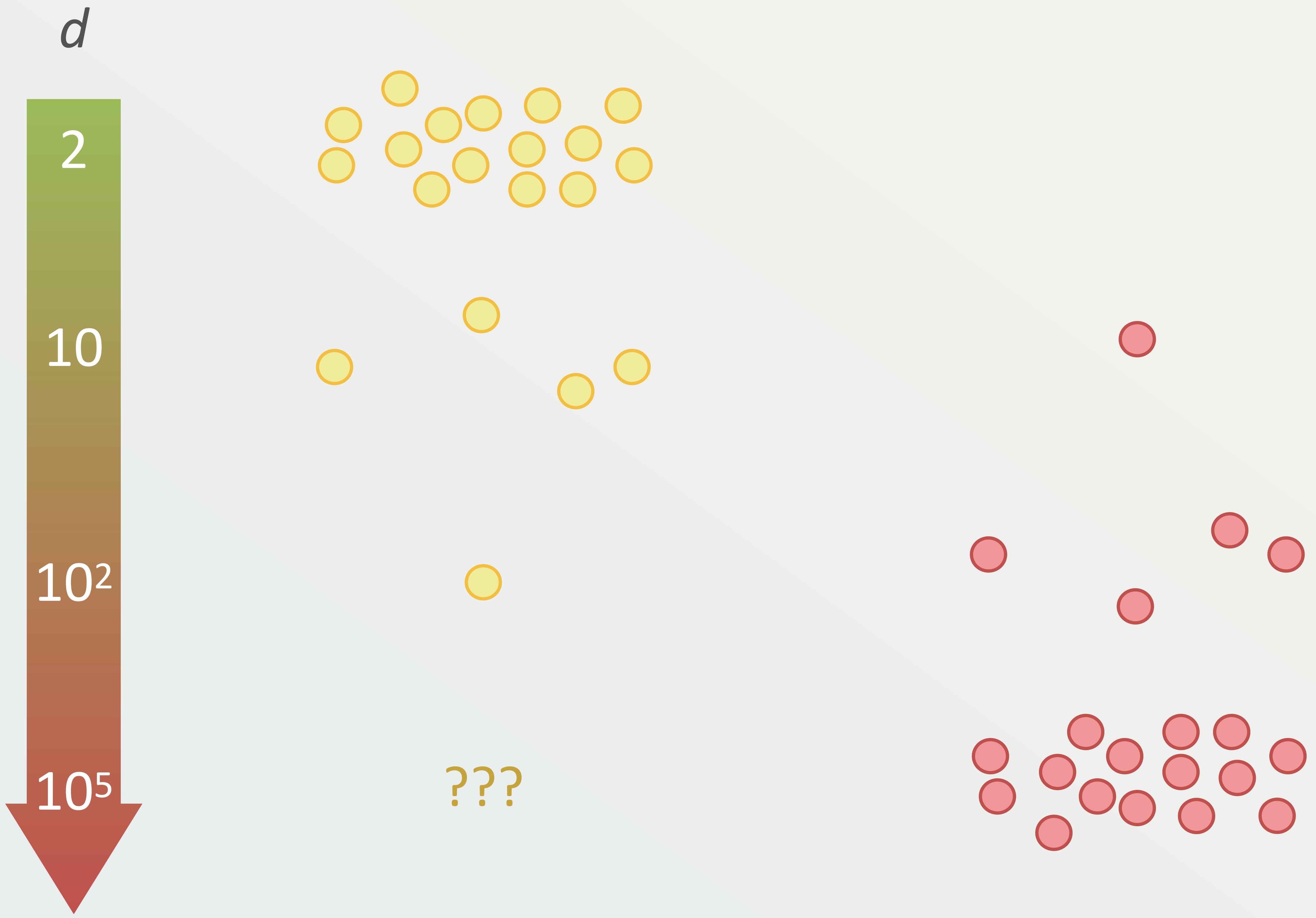
Context !

50 x 50 pixels x 3 (RGB) colors = 7500 dimensions!

Machine Learning

Feature / Representation / Deep Learning

<u>Model?</u>
N number of samples
d dimension of *X*
…



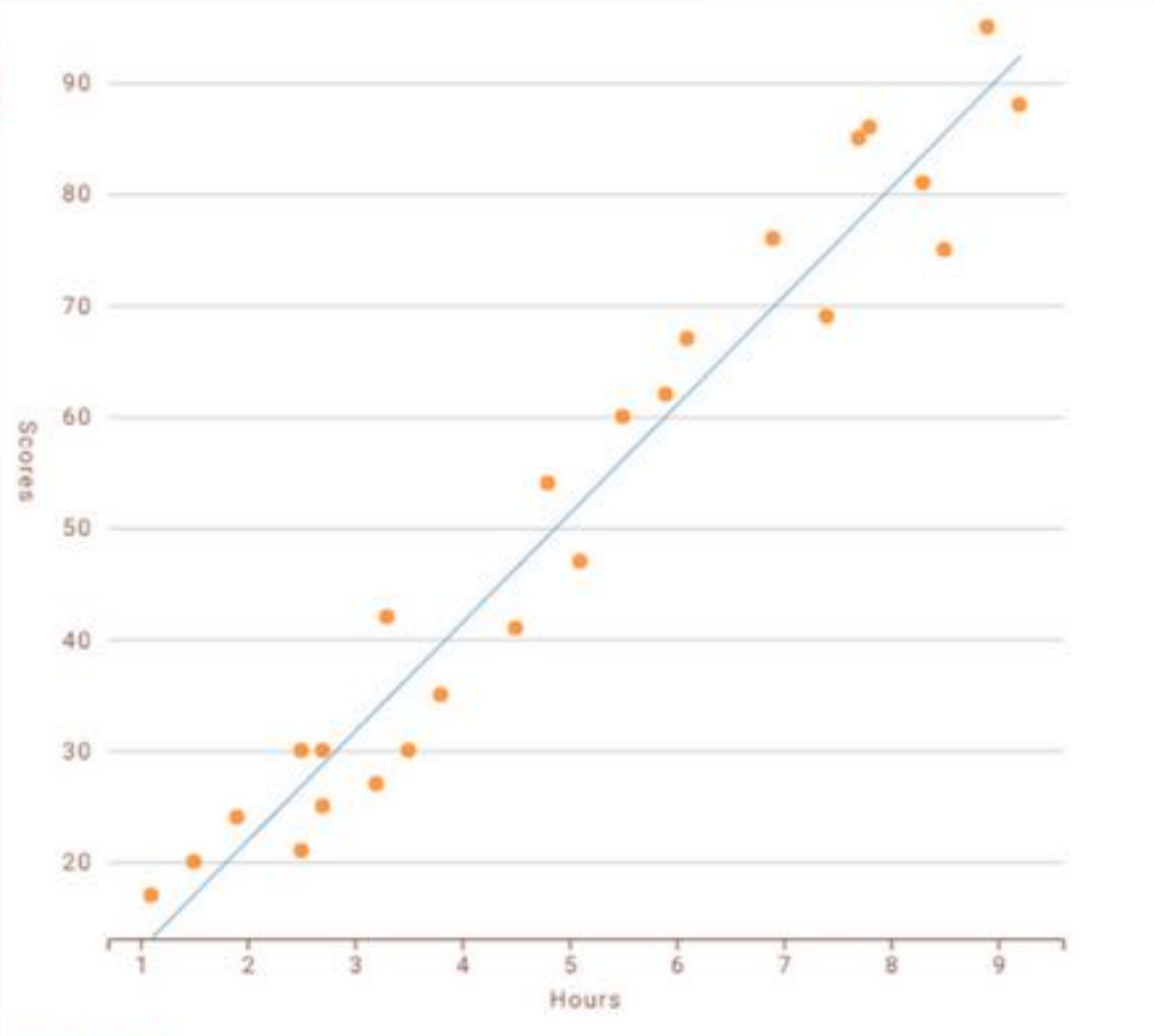
Automatic discovery of representations needed from raw data

Some Machine learning algorithms

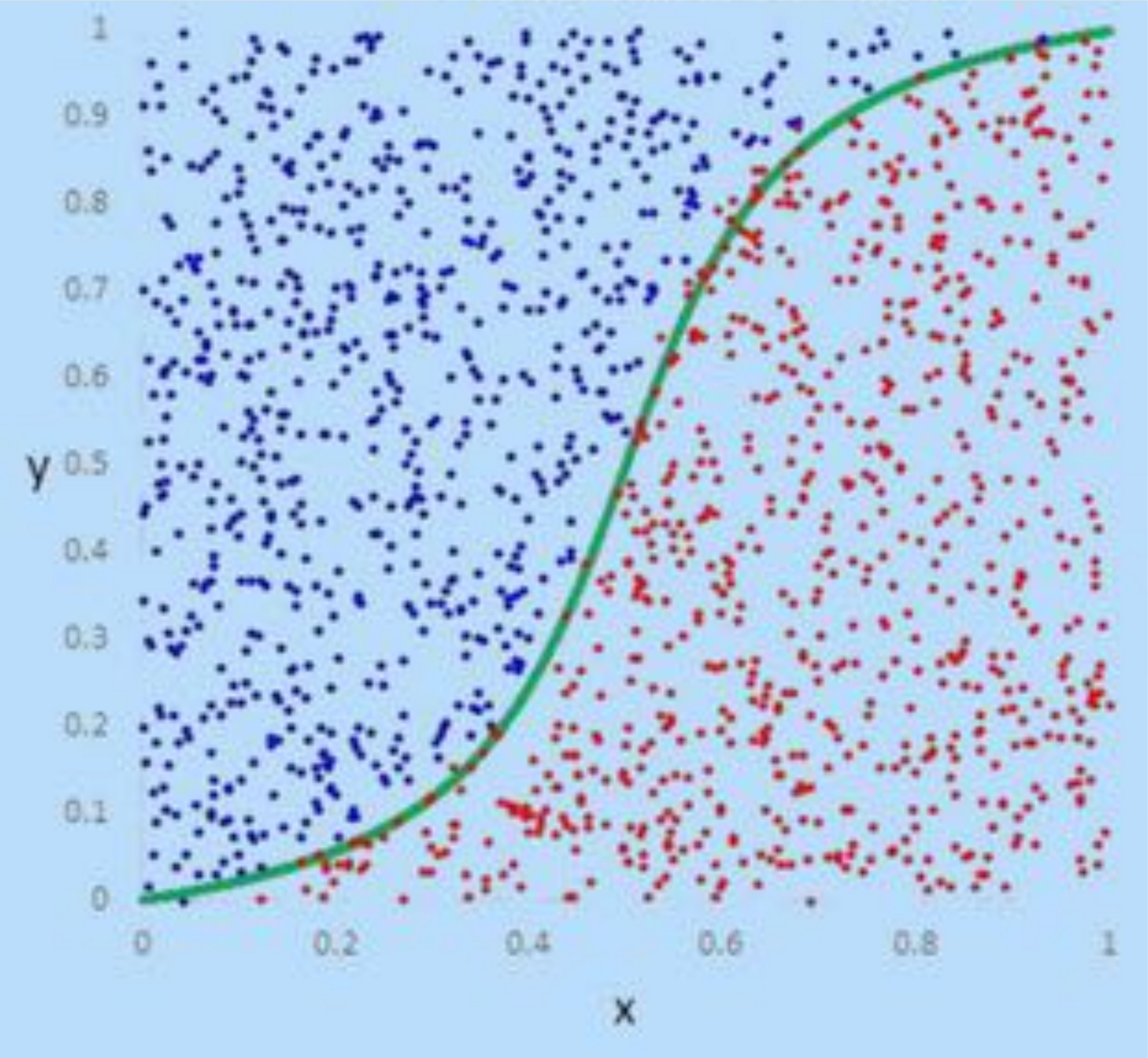
Regression

$$f(x) = ax + b$$

$$f(x) = \frac{a}{1 + e^{-b(x-x_0)}}$$



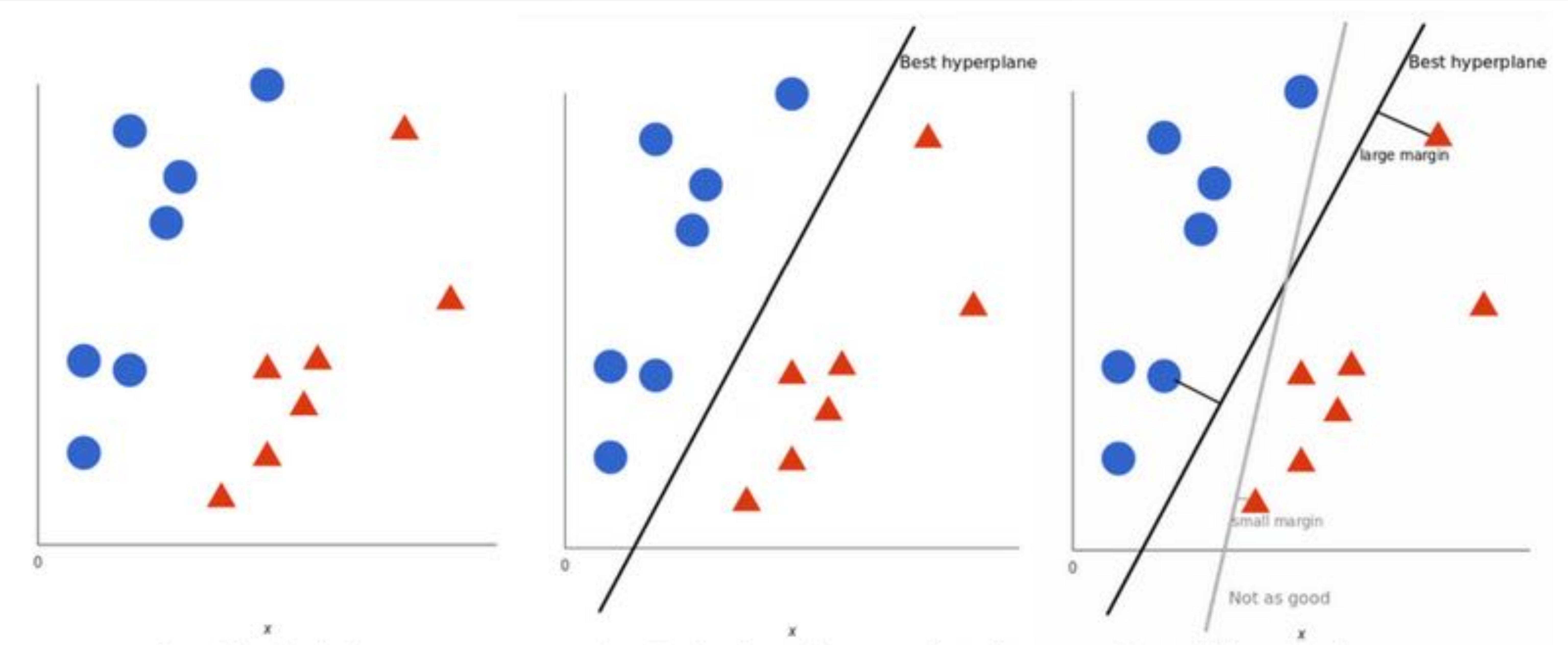
Linear



Logistic

Support Vector Machines (SVM)

(linear)

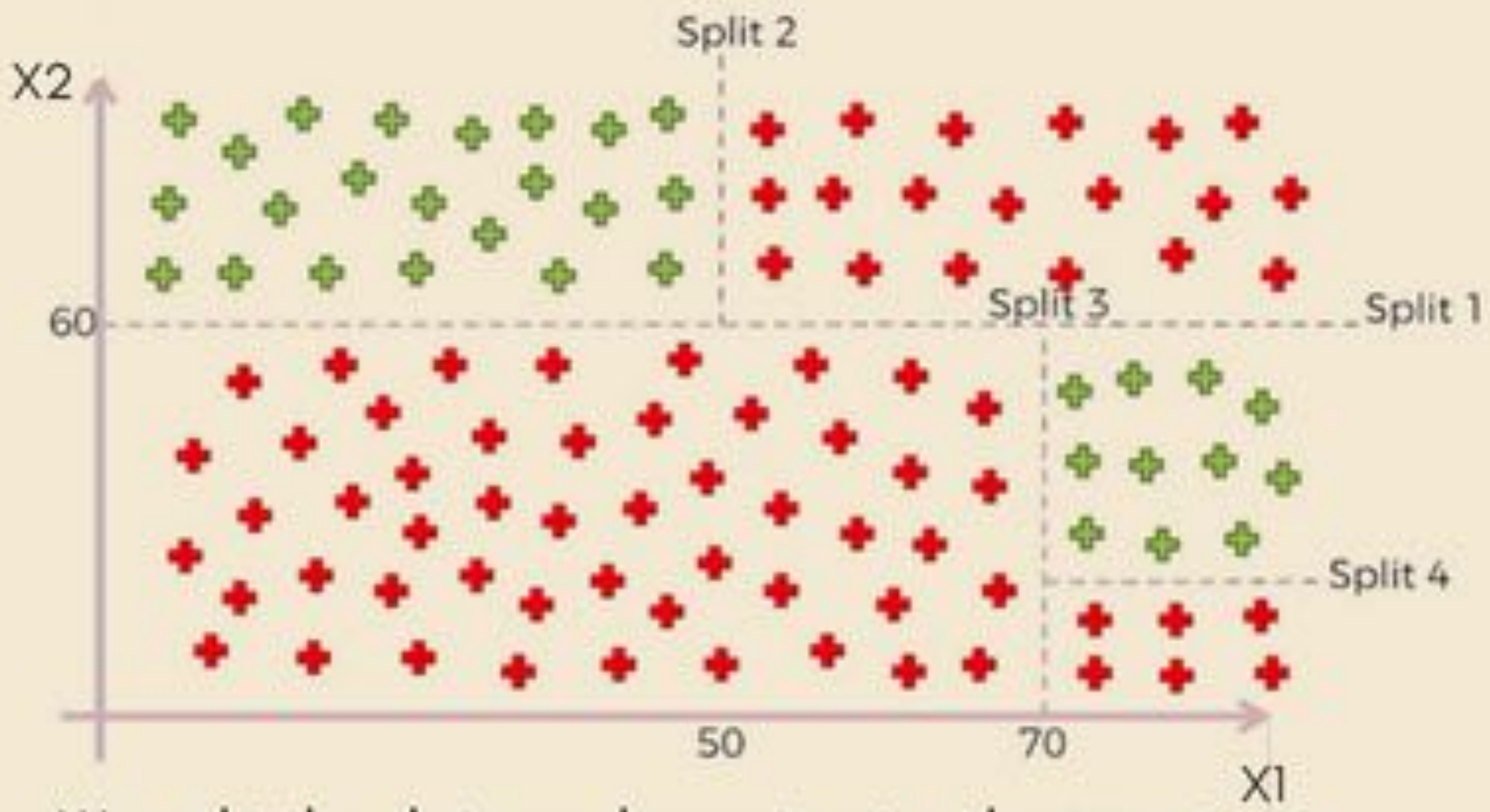


Our labeled data

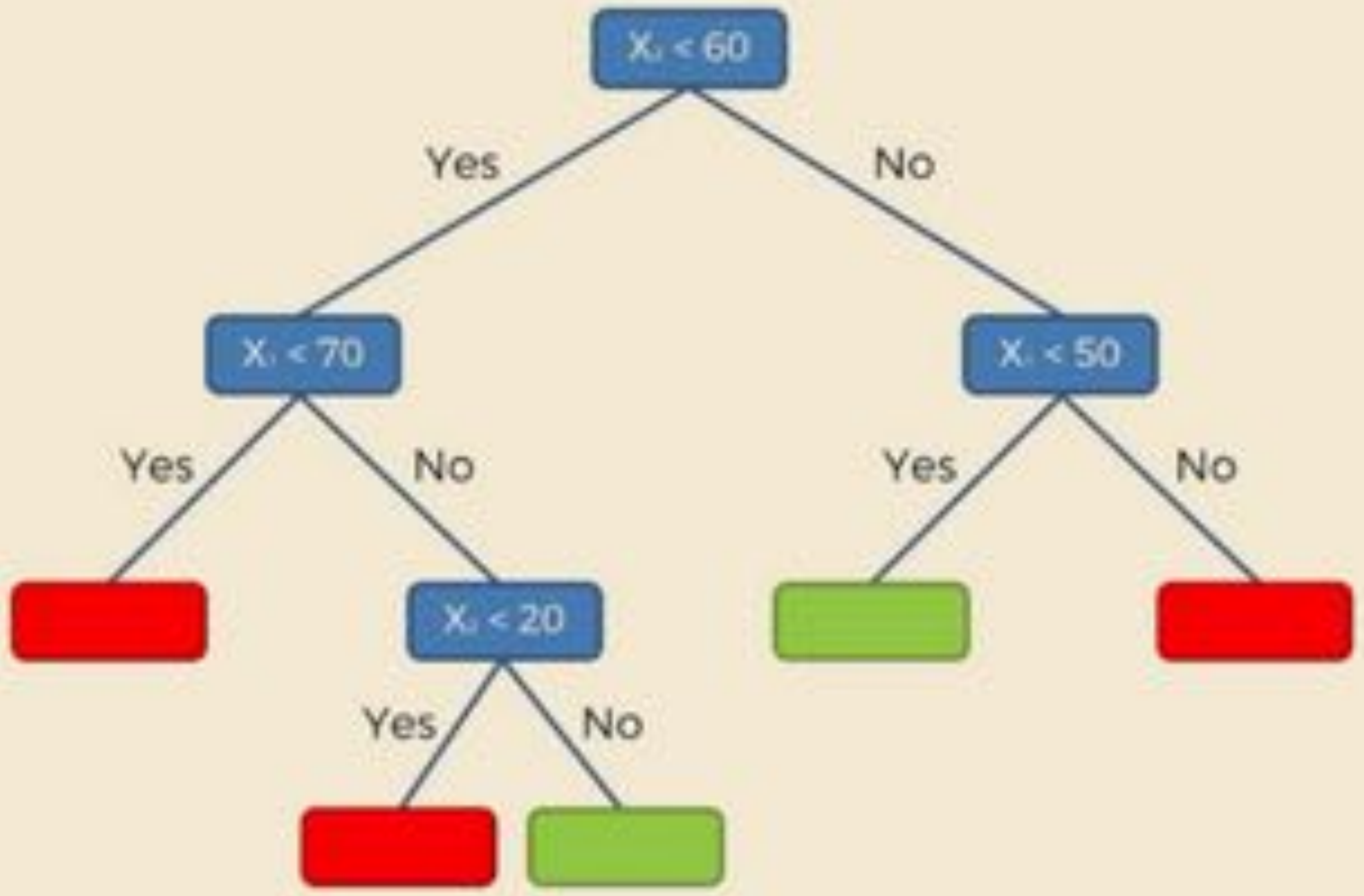
In 2D, the best hyper-plane is simply a line

Not all hyper-planes are created equal

Decision trees

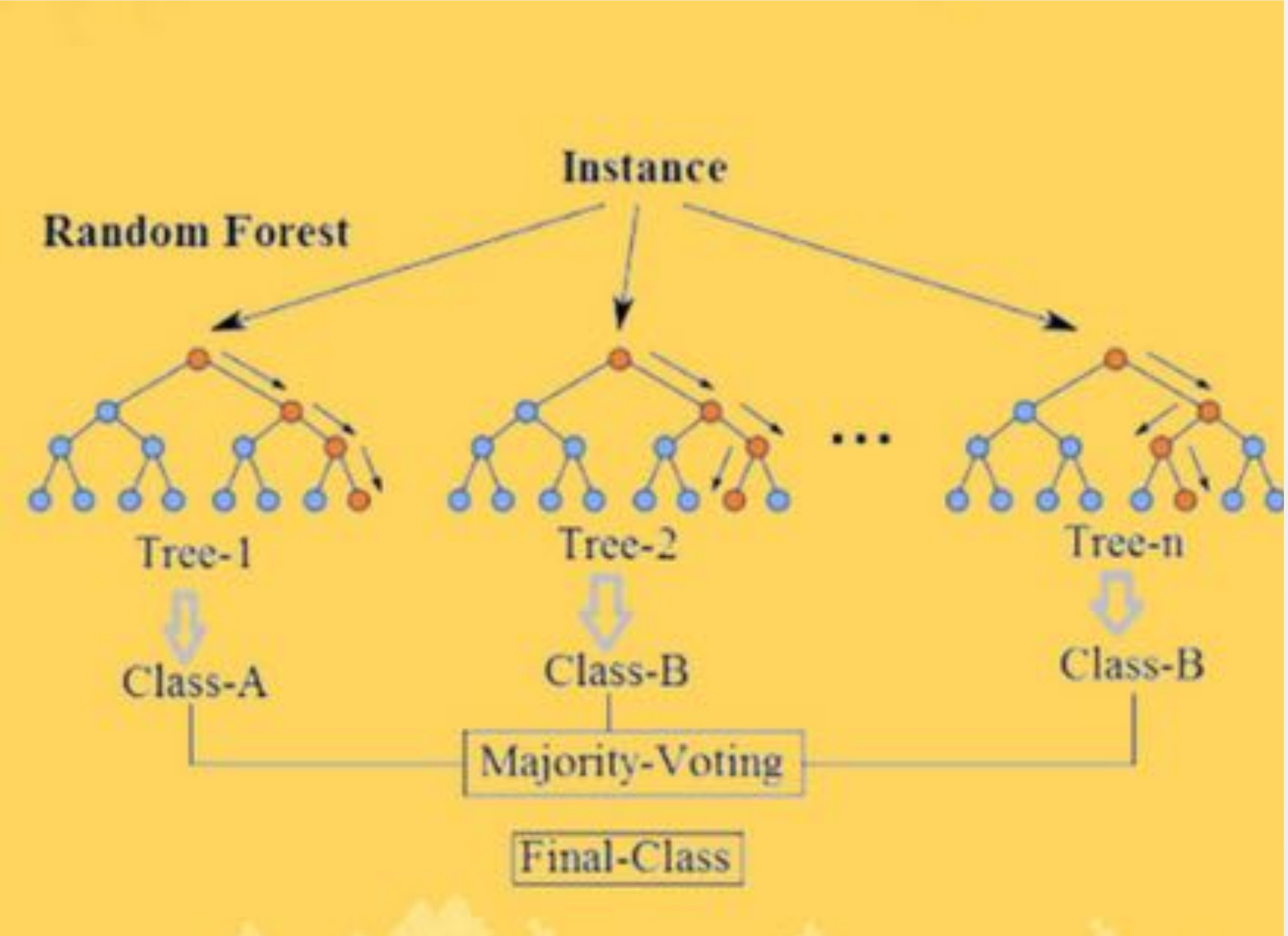


We split the data and construct a decision tree side by side which we will use later. This very task is achieved by using various algorithms. It builds a decision tree from a fixed set of examples and the resulting tree is used to classify future samples.



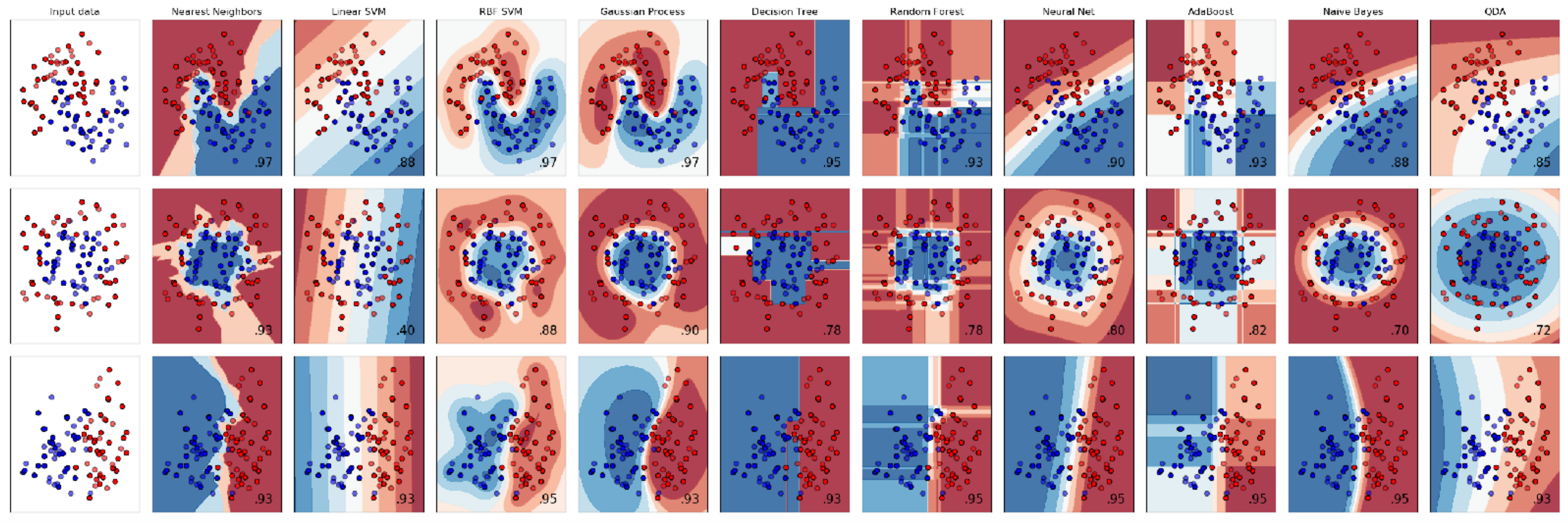
The resulting Tree (obtained by applying algorithms like CART, ID3) which will be later used to predict the outcomes

Random forests



Machine learning algorithms

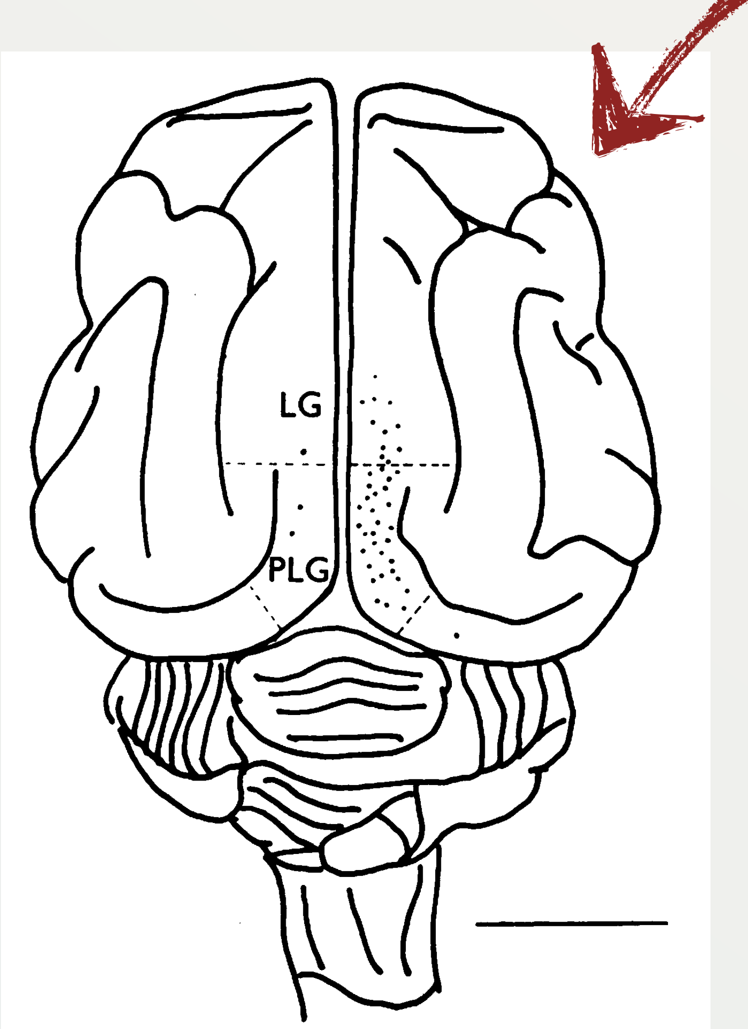
http://scikit-learn.org



What about neural networks?

« Neural » Networks?

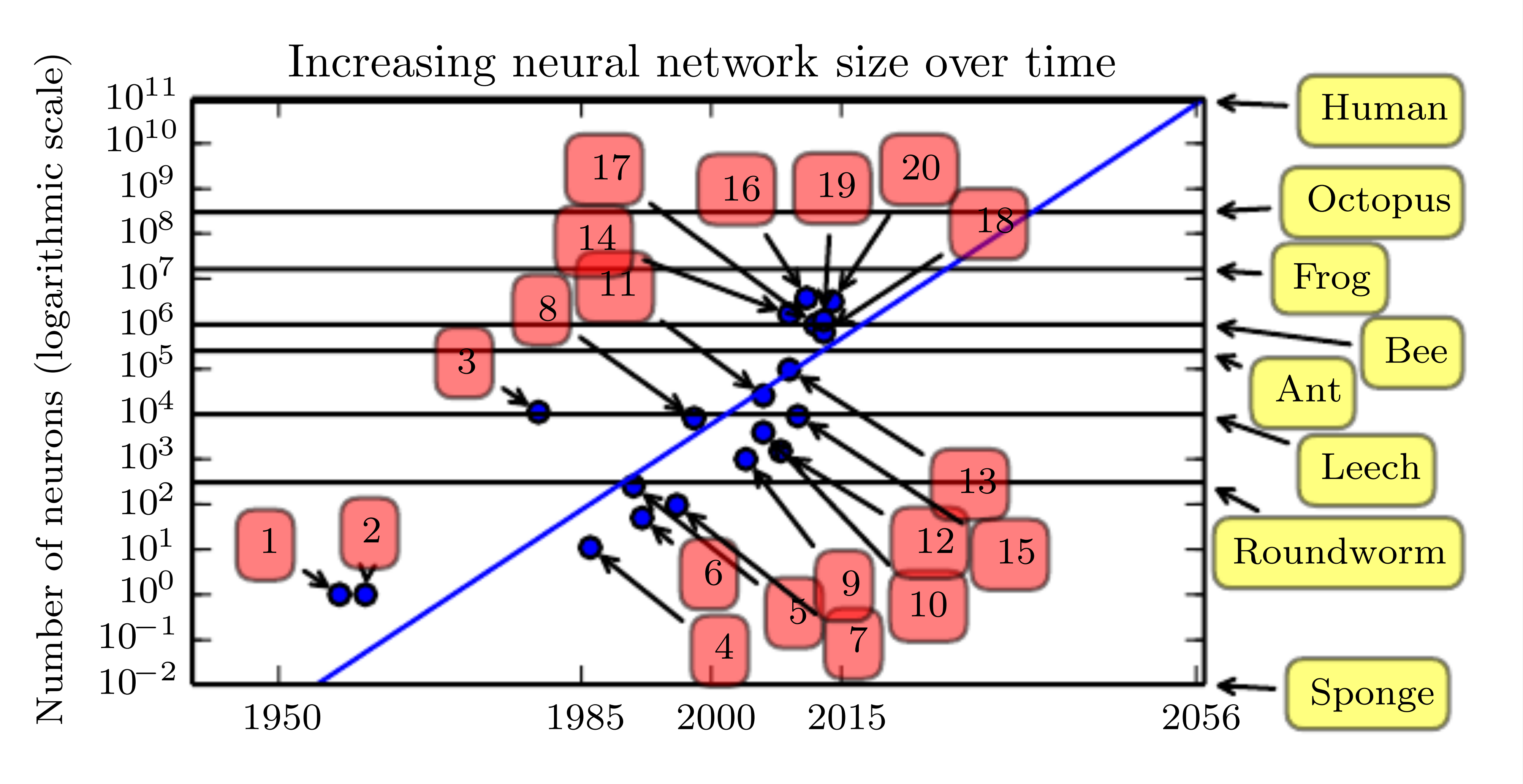
- Loosely inspired from biological systems
- Number of neurons and connections now reaching mammalian values

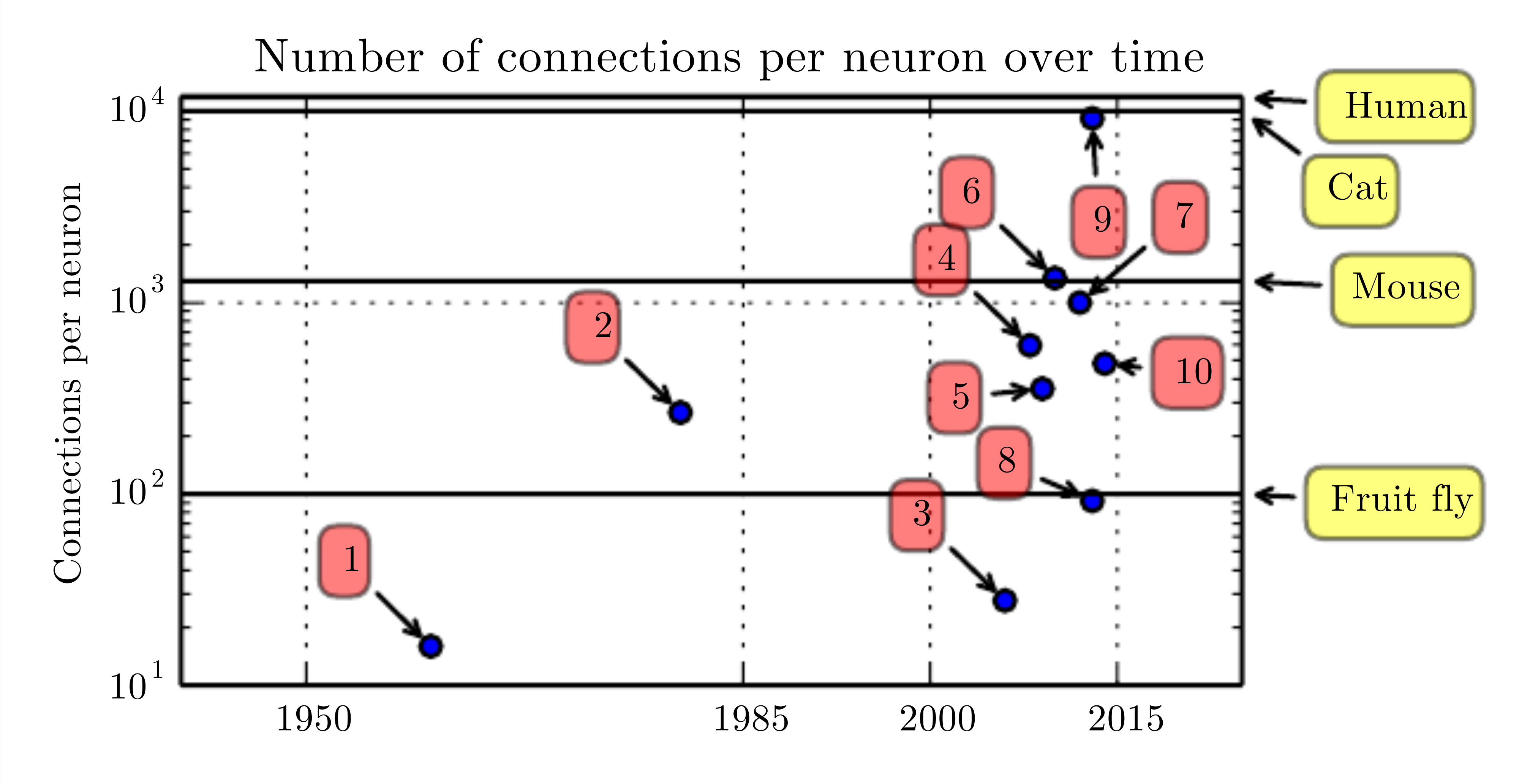


Cat brain (sorry…)

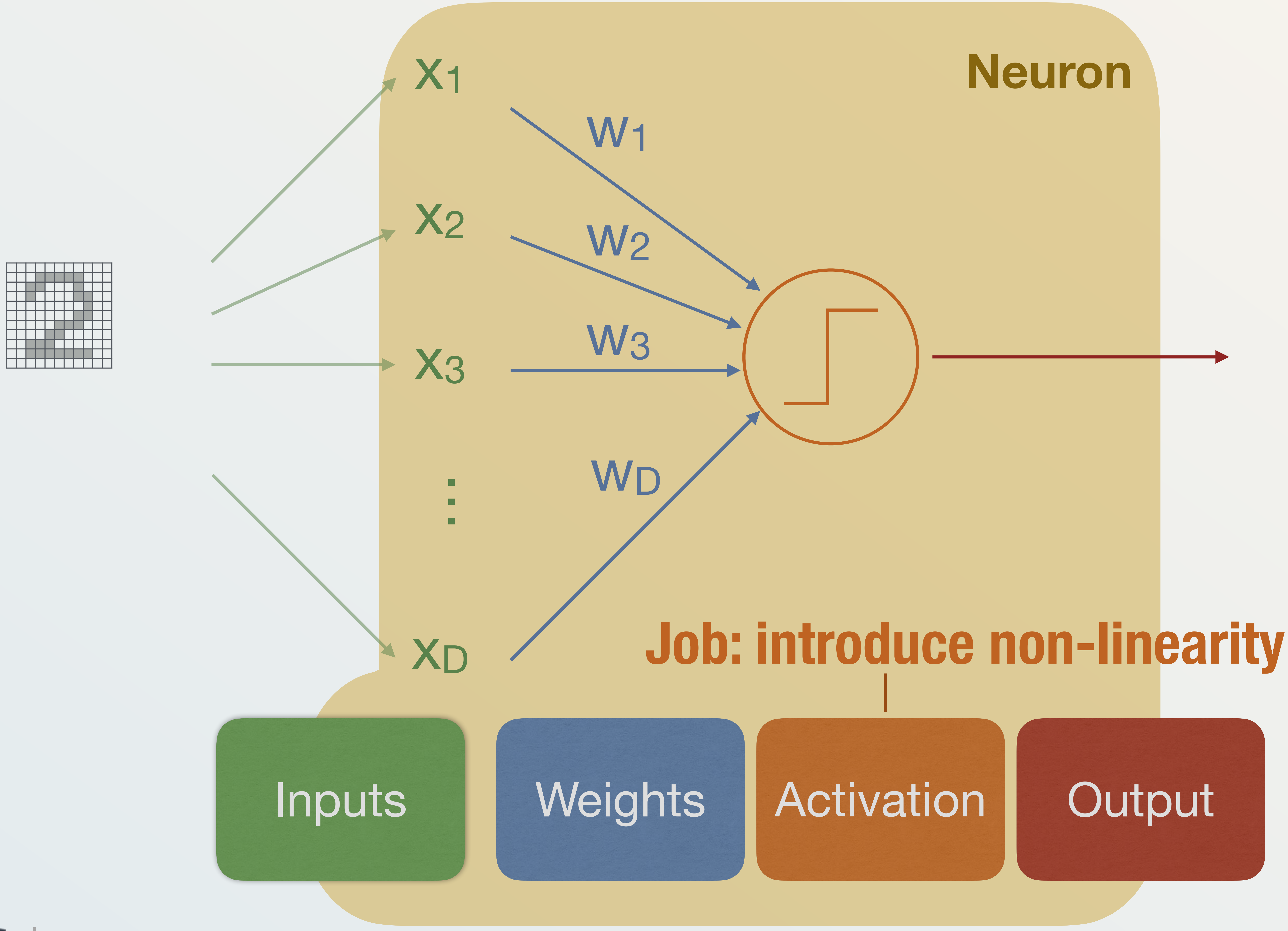
Hubel, David H., and Torsten N. Wiesel. "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex." The Journal of physiology 160.1 (1962): 106-154.

Nobel prize 1981

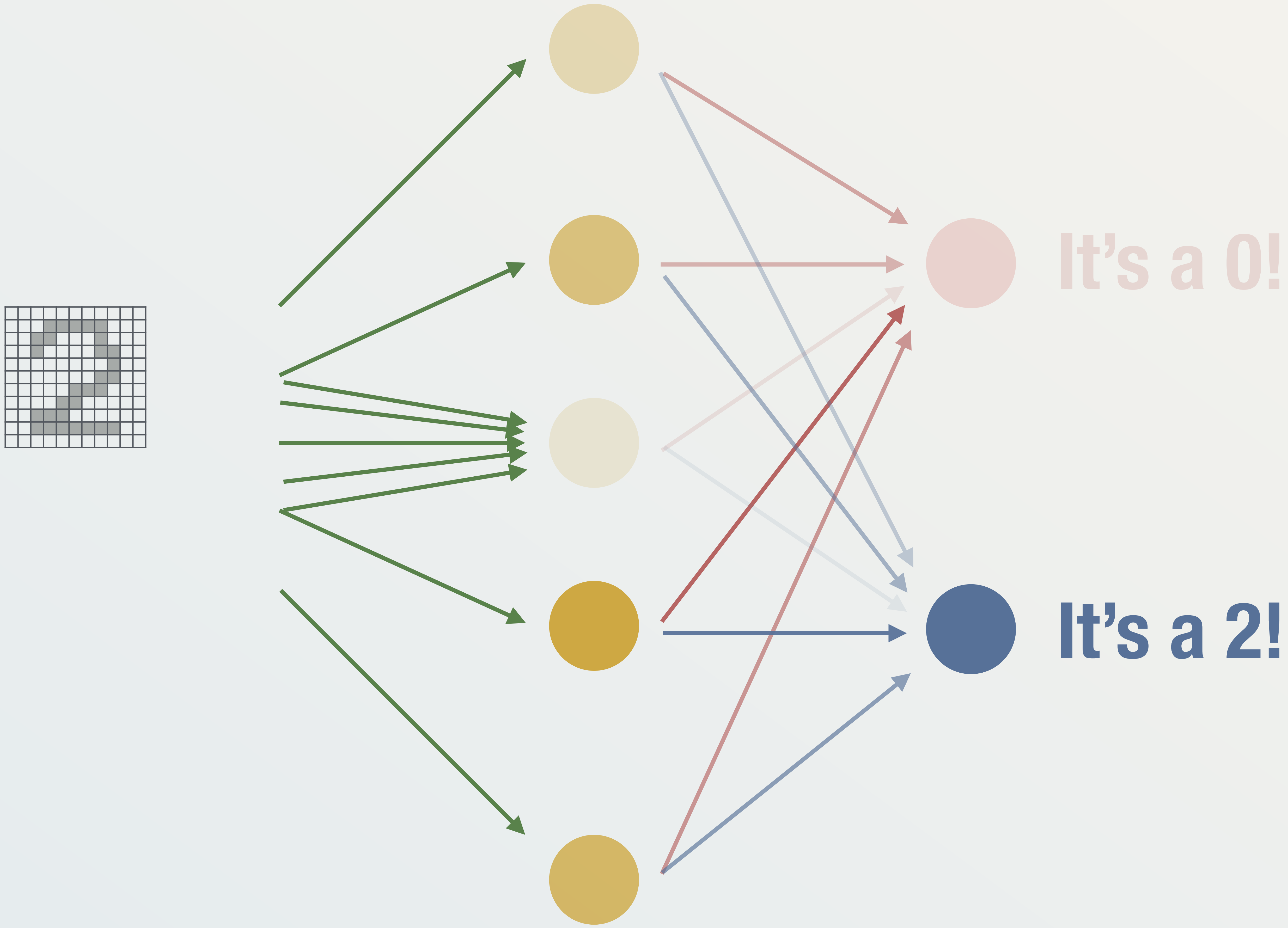




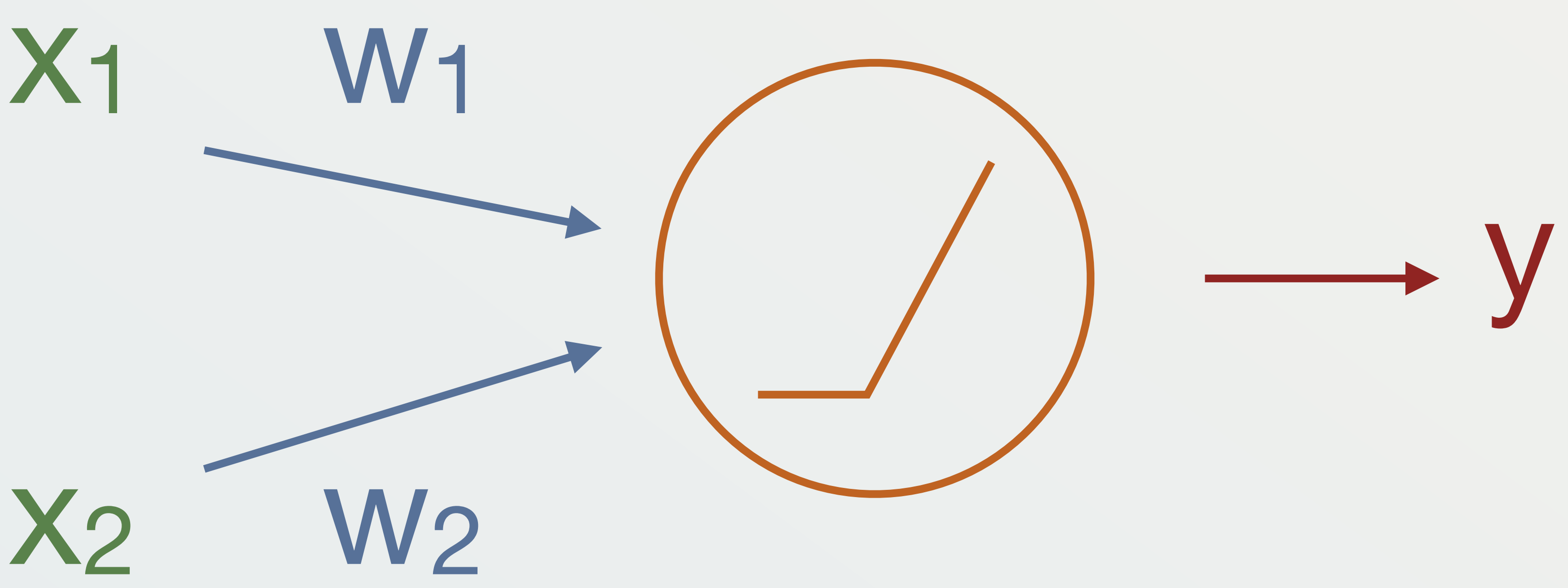
What is a « neuron »?



Assembling neurons



Is my function a neuron?

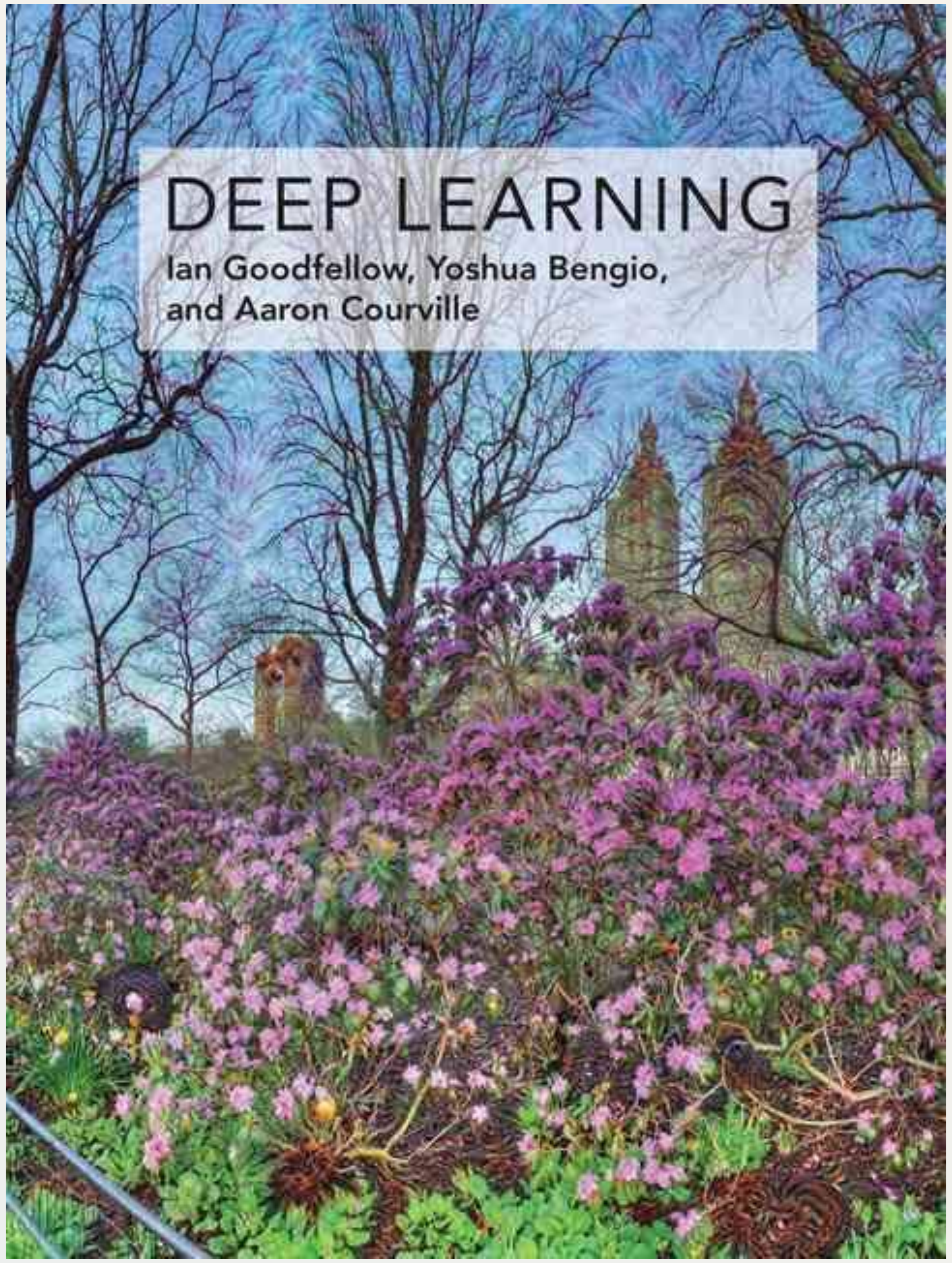


$$y = \max(x_1 w_1 + x_2 w_2, 0)$$

- Neural nets are just computational graphs
- Neurons are *universal approximators*: you can represent ***any*** operation with neural nets
- But not all neural nets perform deep learning…

Deep learning

For more, check out the “Deep Learning Bible”

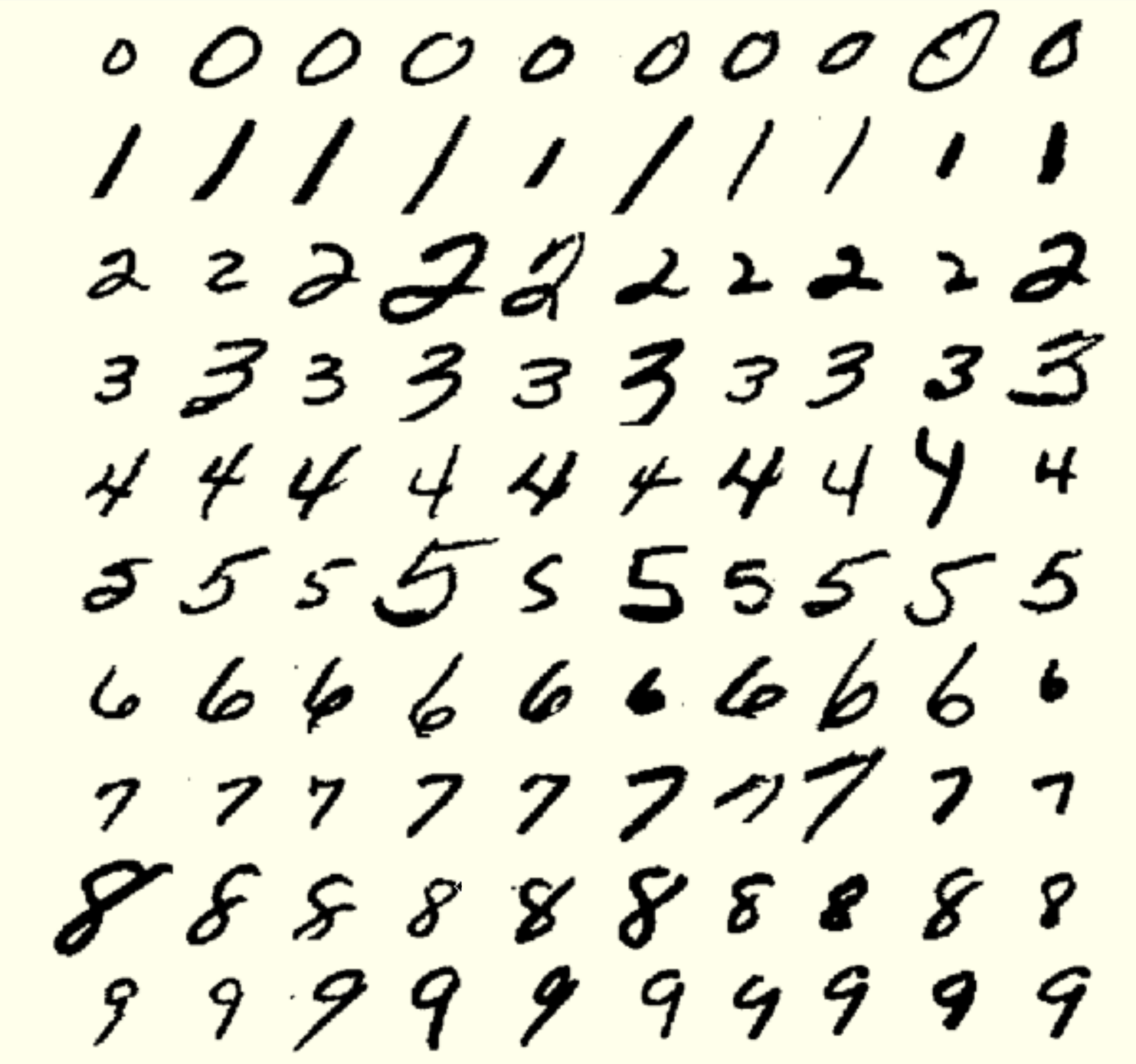


Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). Deep learning (Vol 1). Cambridge: MIT press.

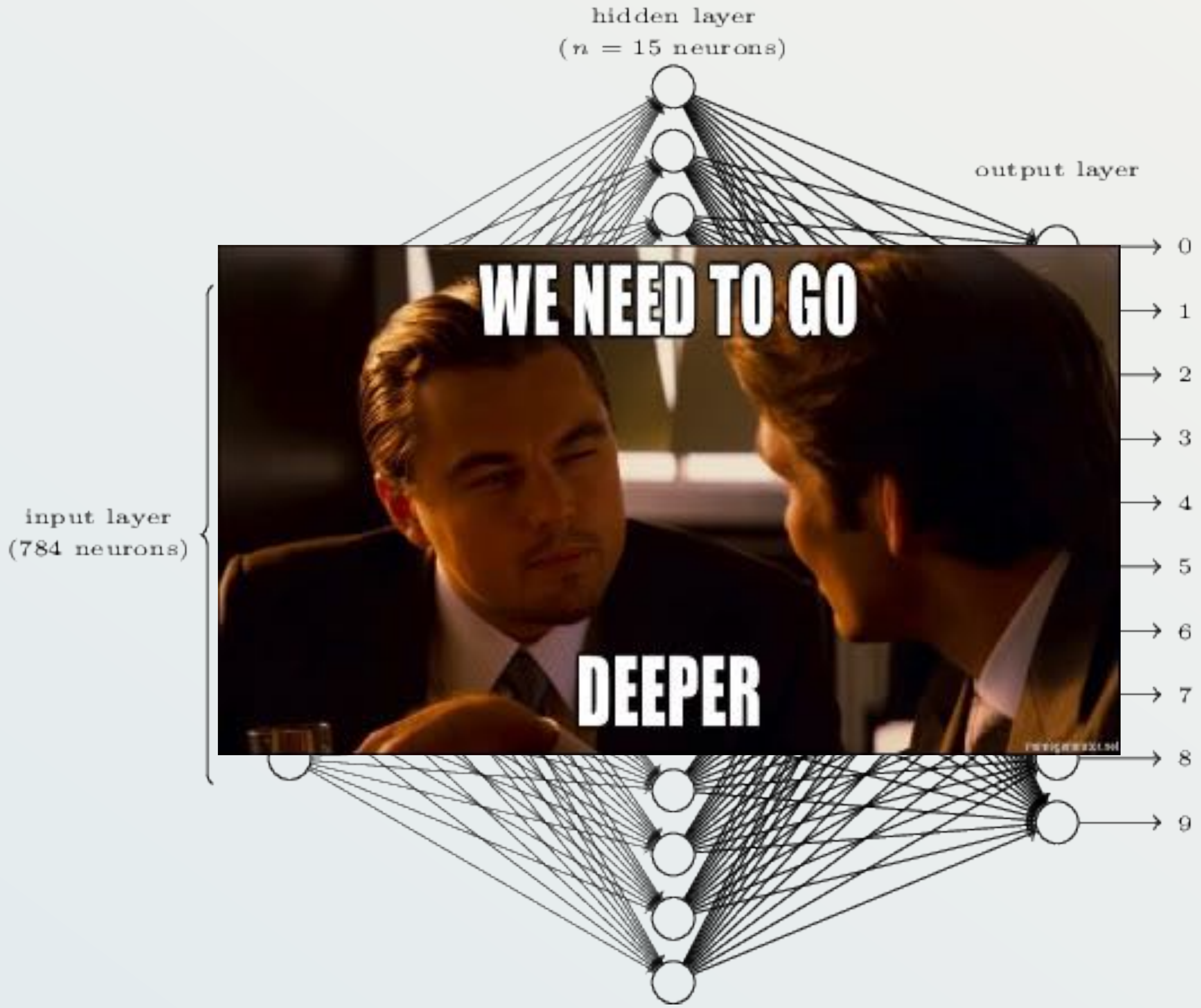
What’s so great about neural networks for machine learning?

Deep classifiers

Back to MNIST



A simple neural net

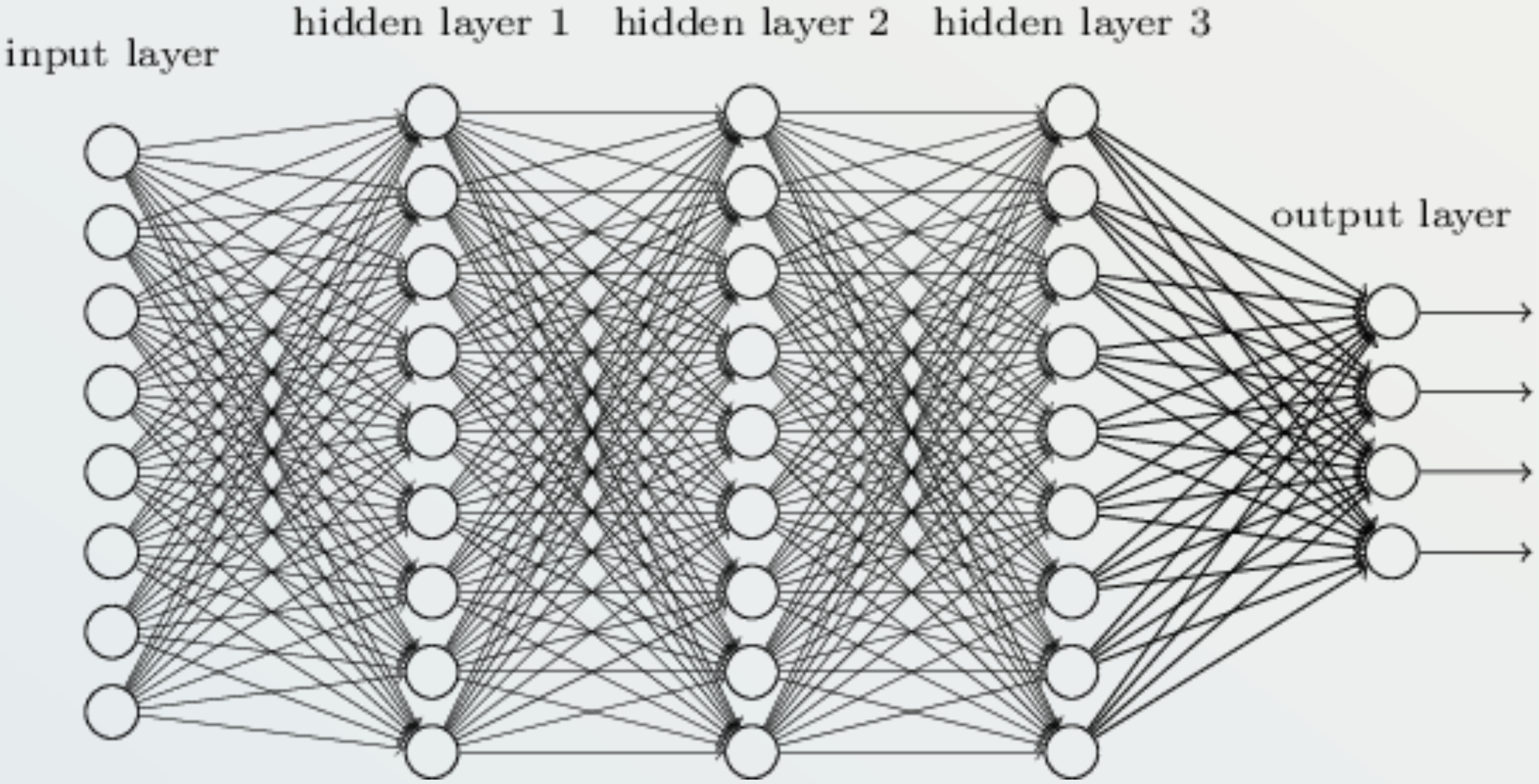


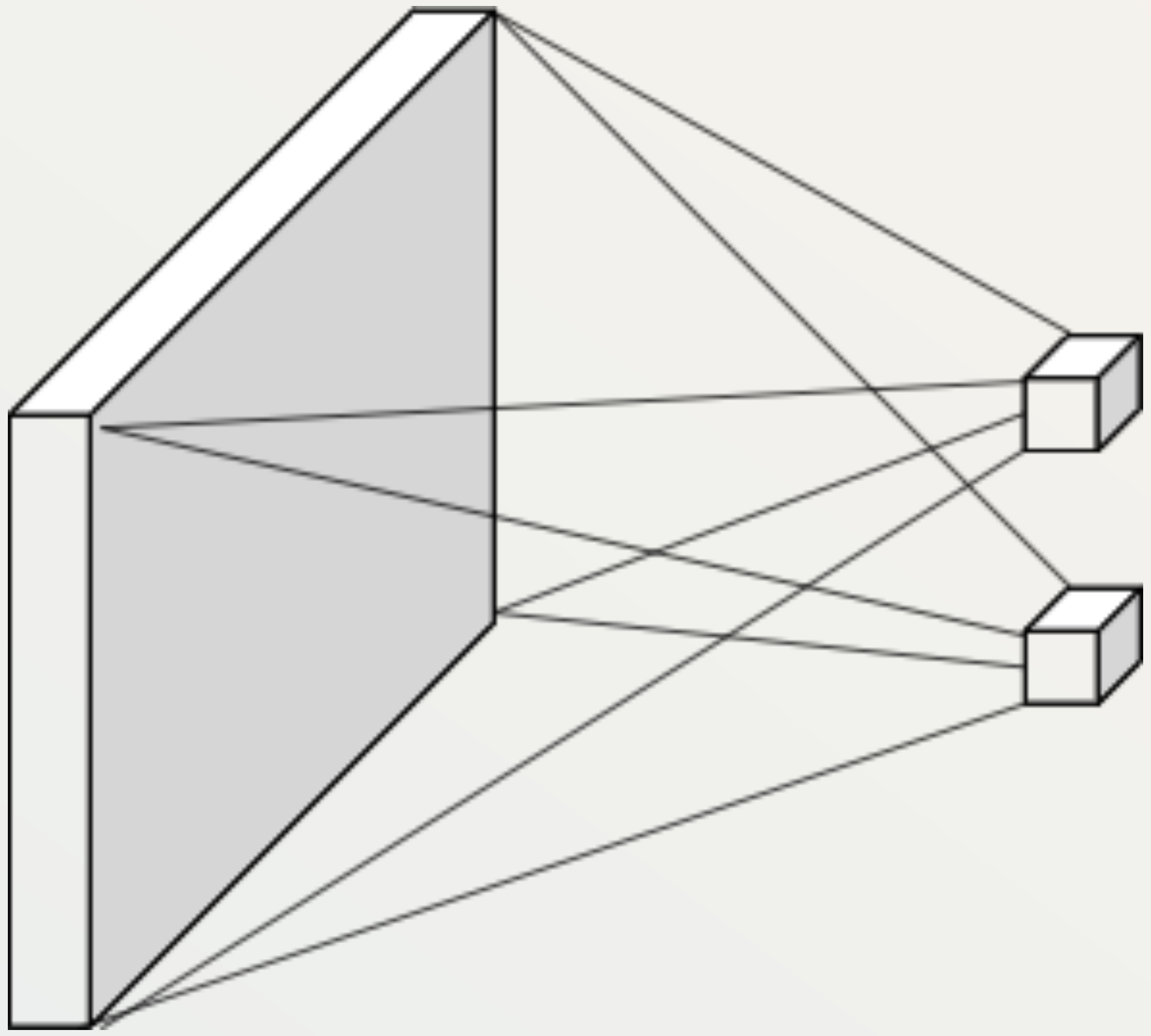
- Simple « Multi-Layer Perceptron » (MLP)
  - 28x28 = 784 pixels on input
  - 0 → 9: 10 outputs
  - 1 hidden layer

91.5% accuracy
= 8.5% error

Easy right?

With enough neurons and depth, you can replicate any function!



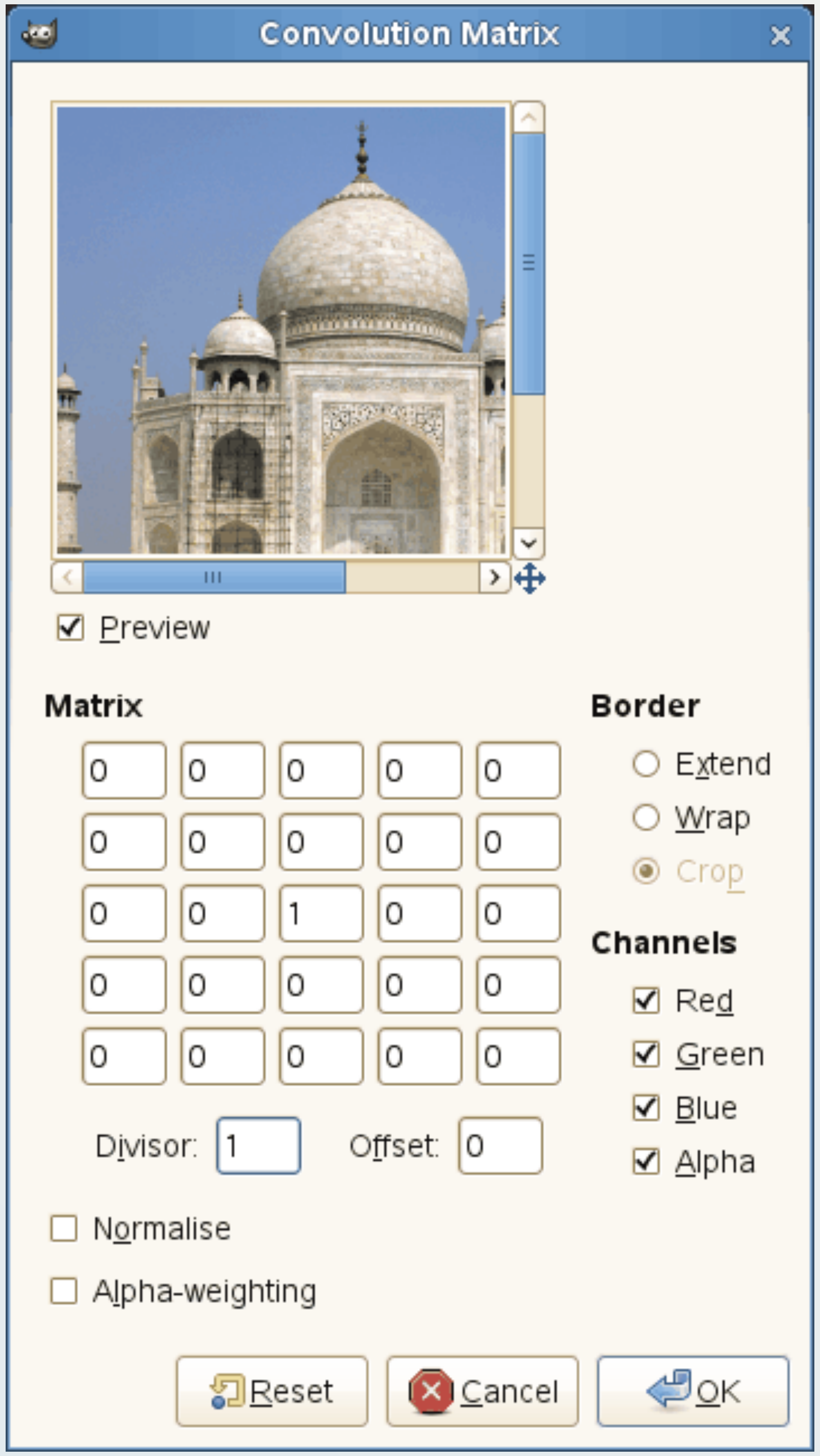


Yes. But, this becomes cumbersome fast…

Especially for big input images (e.g. 256x256 px)

To improve this approach, the neurons can be connected differently

Image filters (Gimp docs)



☑ Preview

Matrix

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Divisor: 1 Offset: 0

☐ Normalise

☐ Alpha-weighting

Border

○ Extend

○ Wrap

● Crop

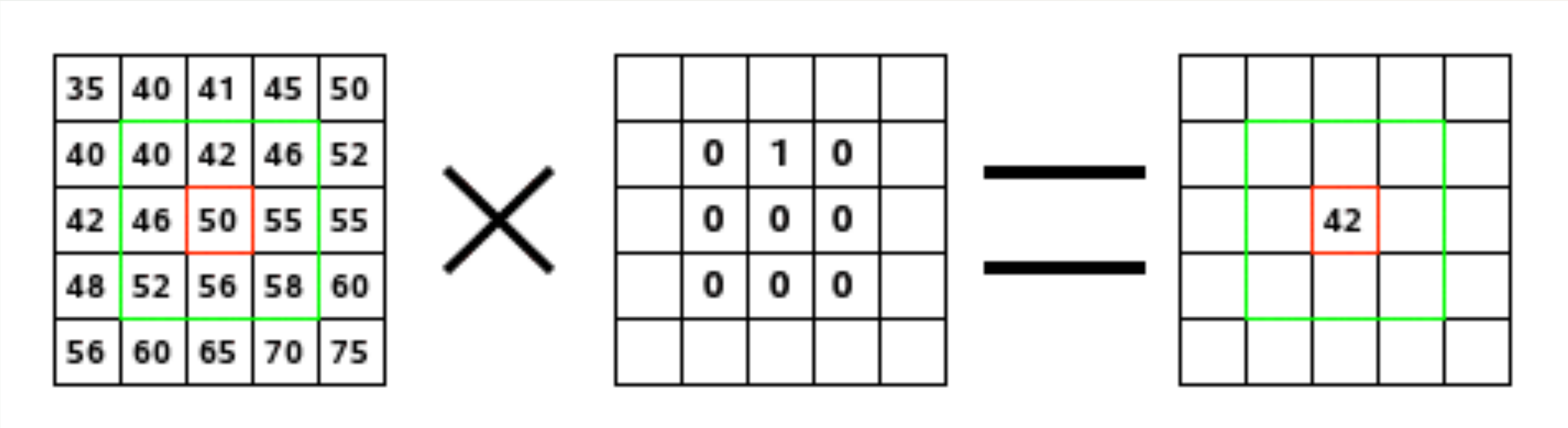
Channels

☑ Red

☑ Green

☑ Blue

☑ Alpha



35	40	41	45	50
40	40	42	46	52
42	46	50	55	55
48	52	56	58	60
56	60	65	70	75

×

	0	1	0	
	0	0	0	
	0	0	0	

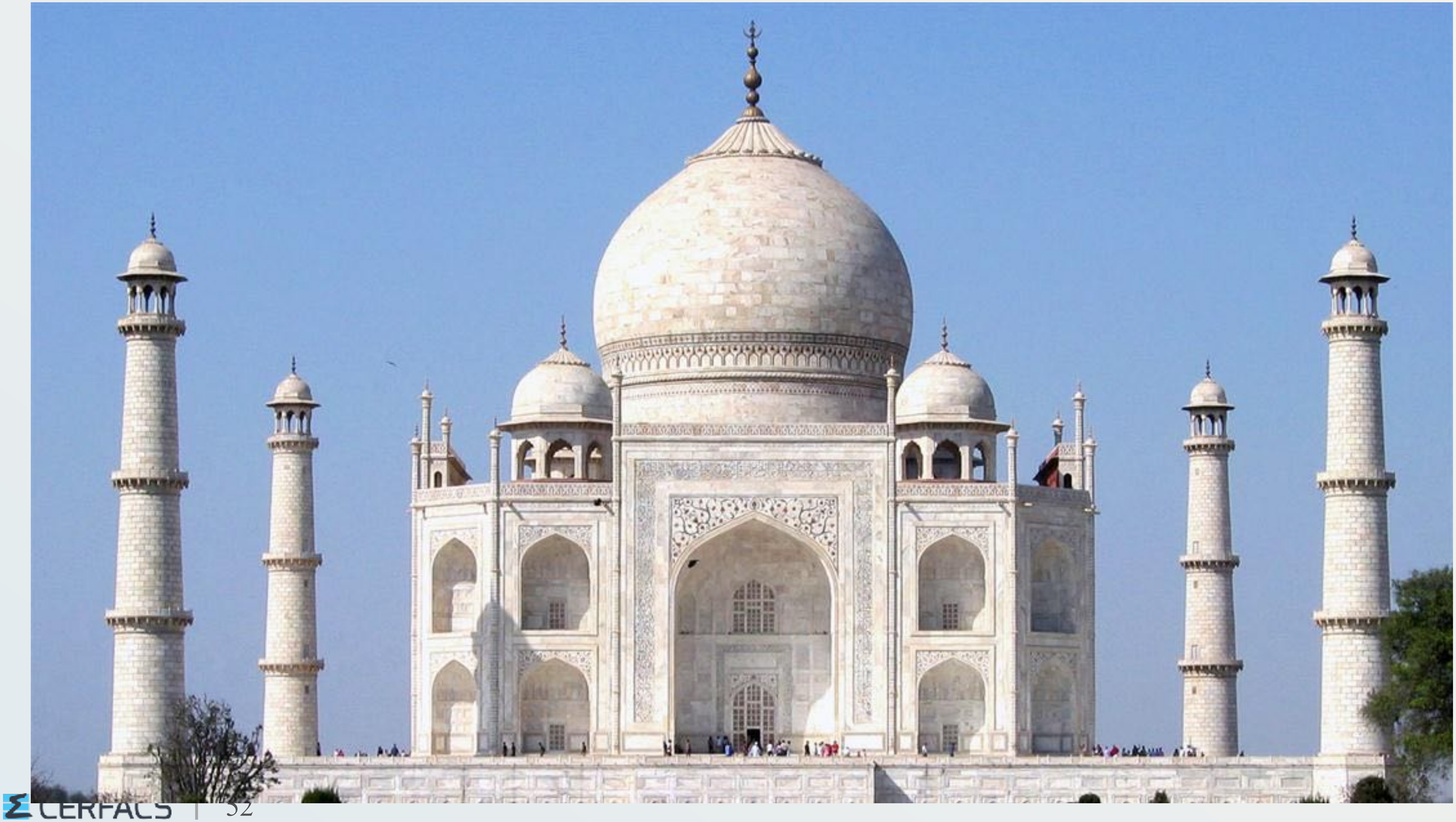
=

		42		

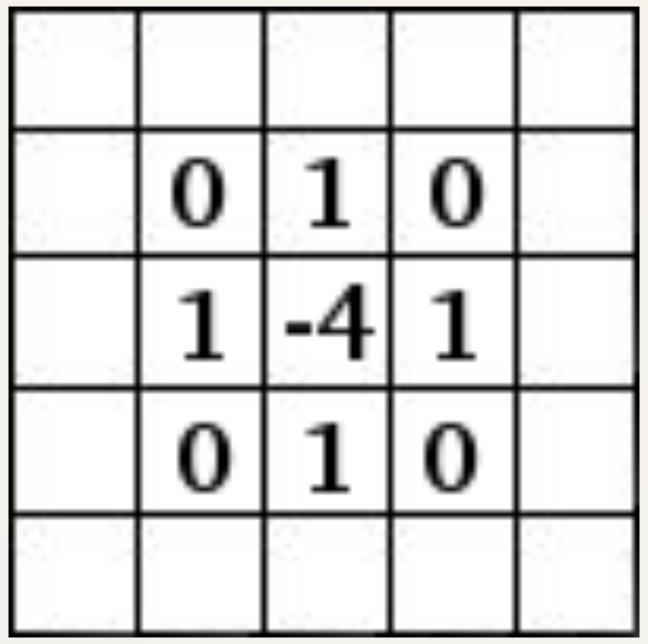
Image Kernel Result

This is in effect a convolution of the image by the filter

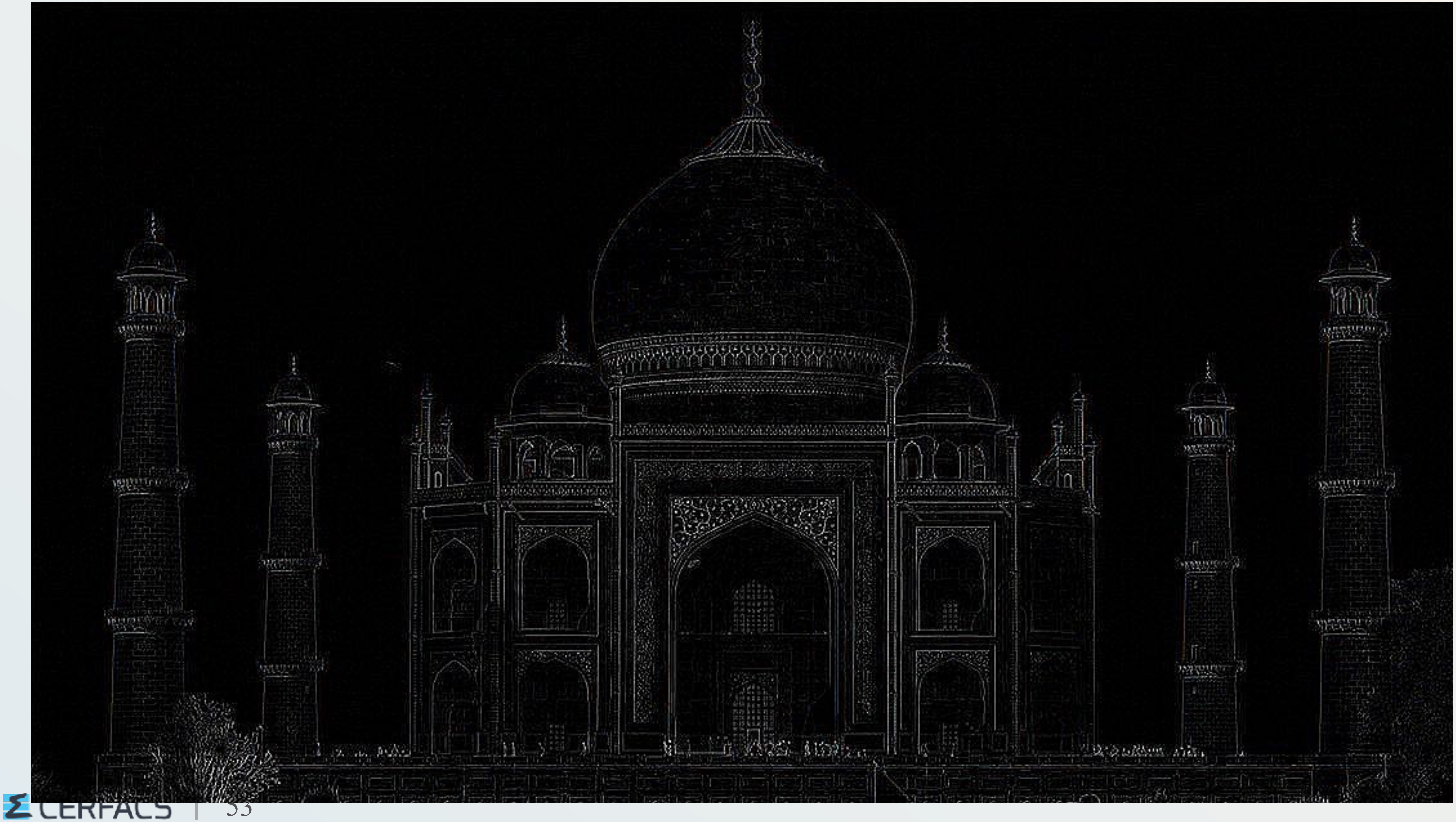
Base



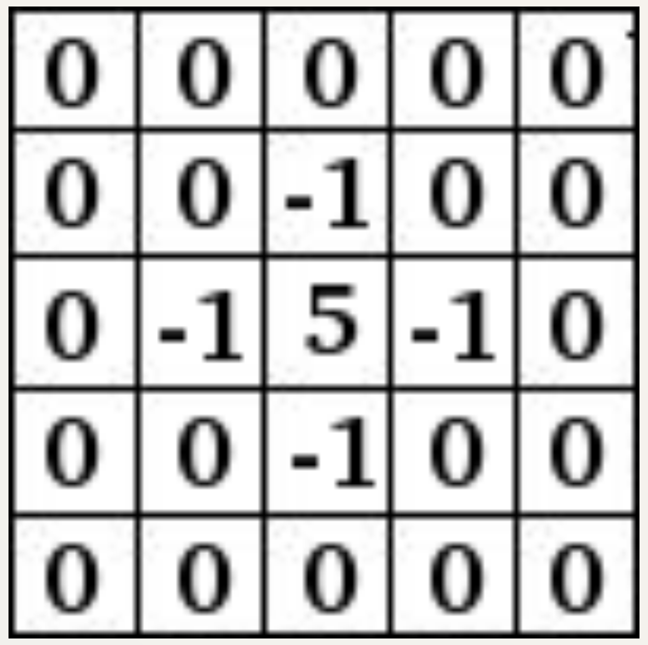
Edge detection



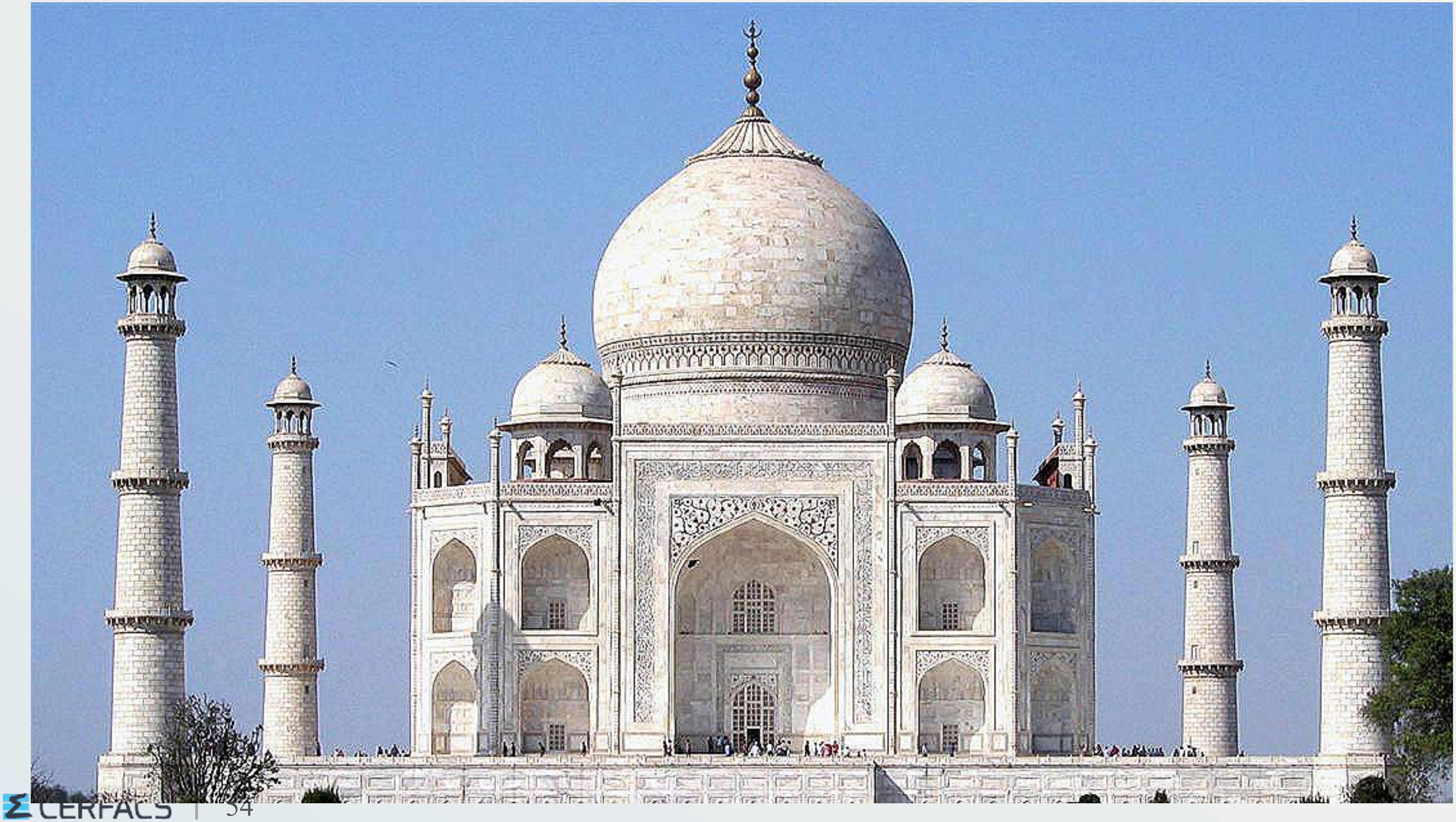
	0	1	0	
	1	-4	1	
	0	1	0	



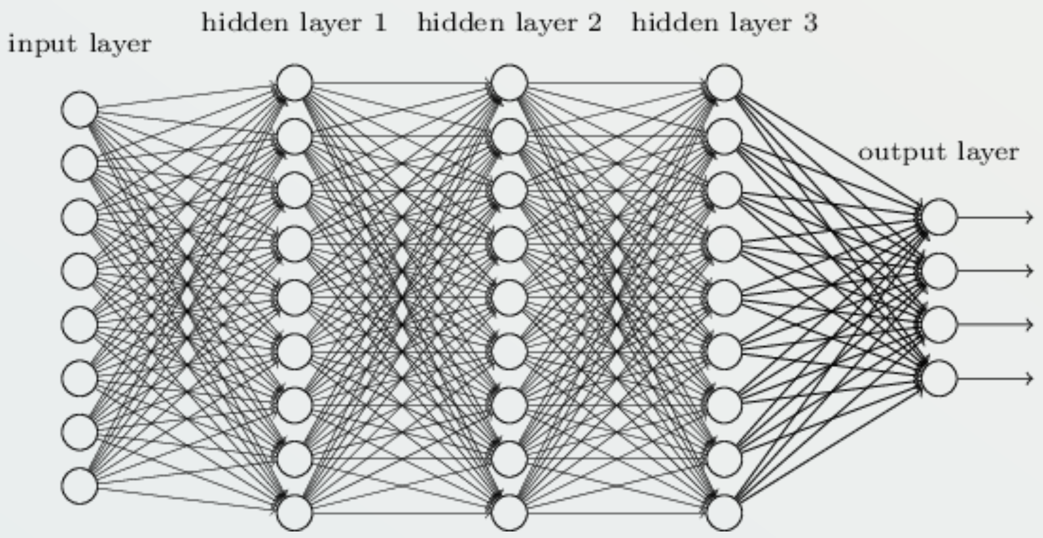
Sharpen



0	0	0	0	0
0	0	-1	0	0
0	-1	5	-1	0
0	0	-1	0	0
0	0	0	0	0



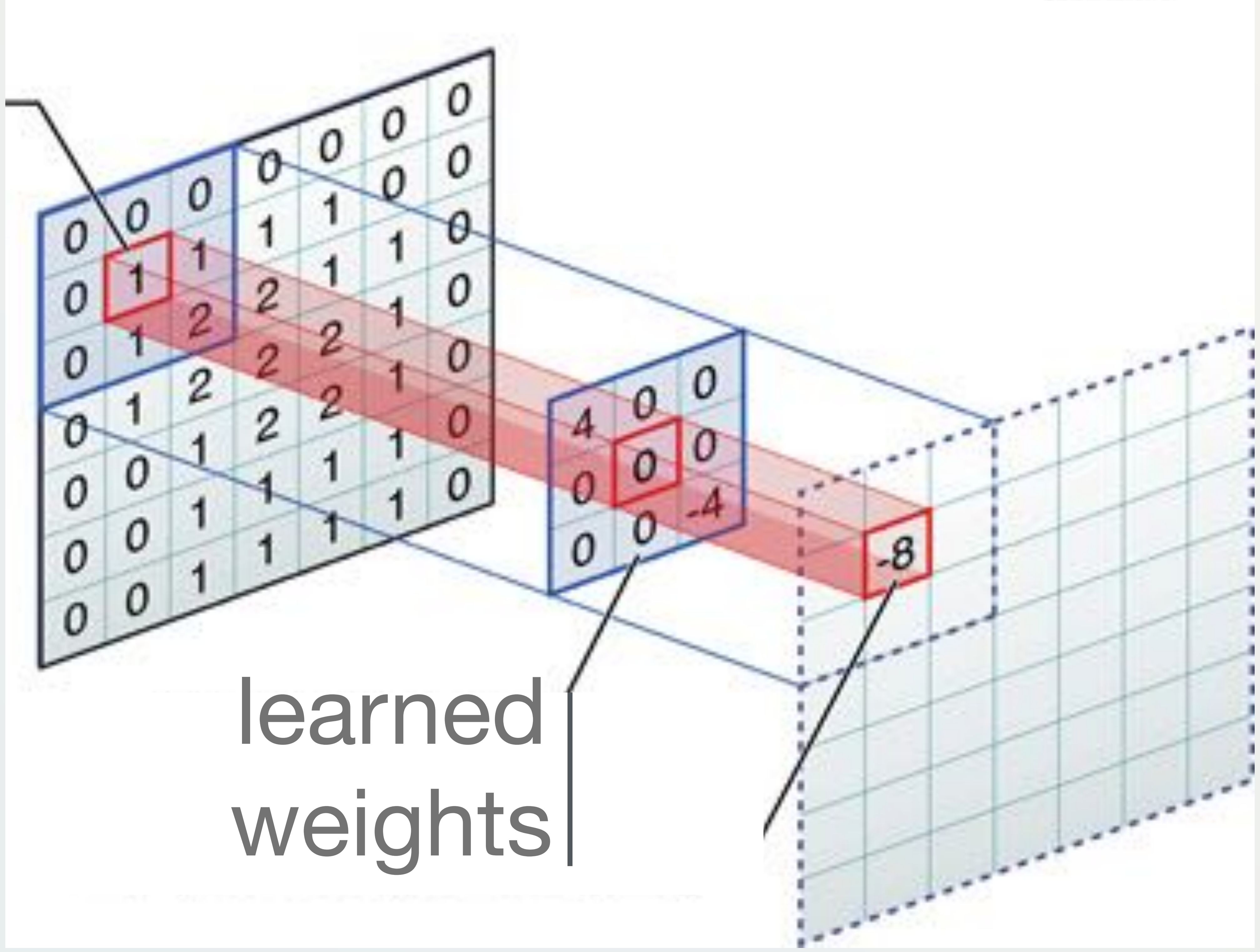
Building smarter layers



Fully Connected Layer

Shared weights using <u>convolution</u> :

You learn the kernel weights, then share over the full input



learned weights

Deep MNIST

(less than humans: ~2 %!)

LeNet-5 (1998) : ~1% error

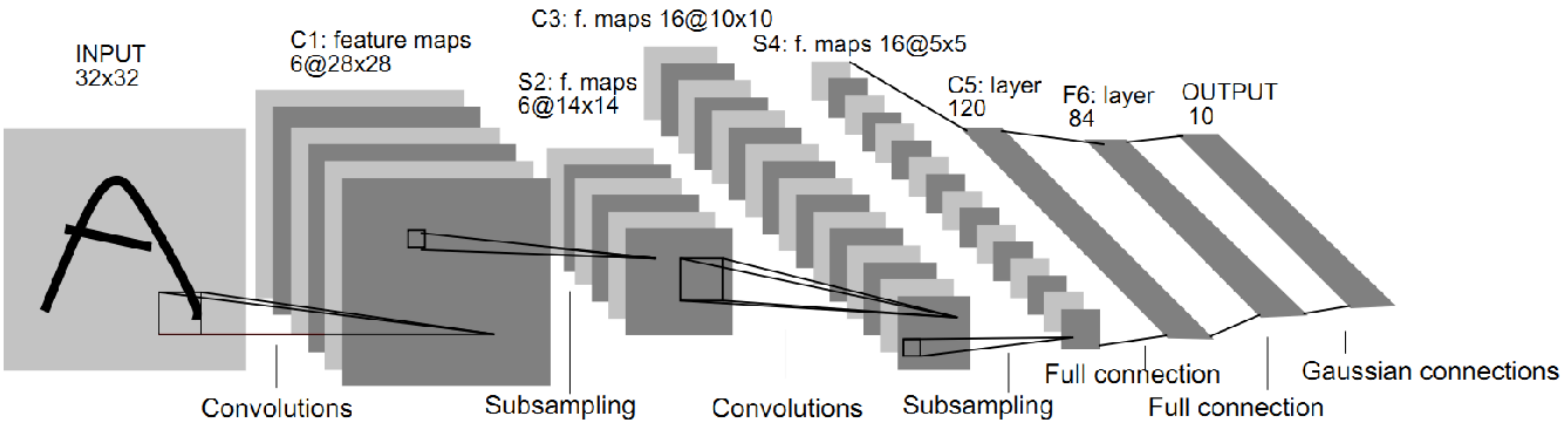


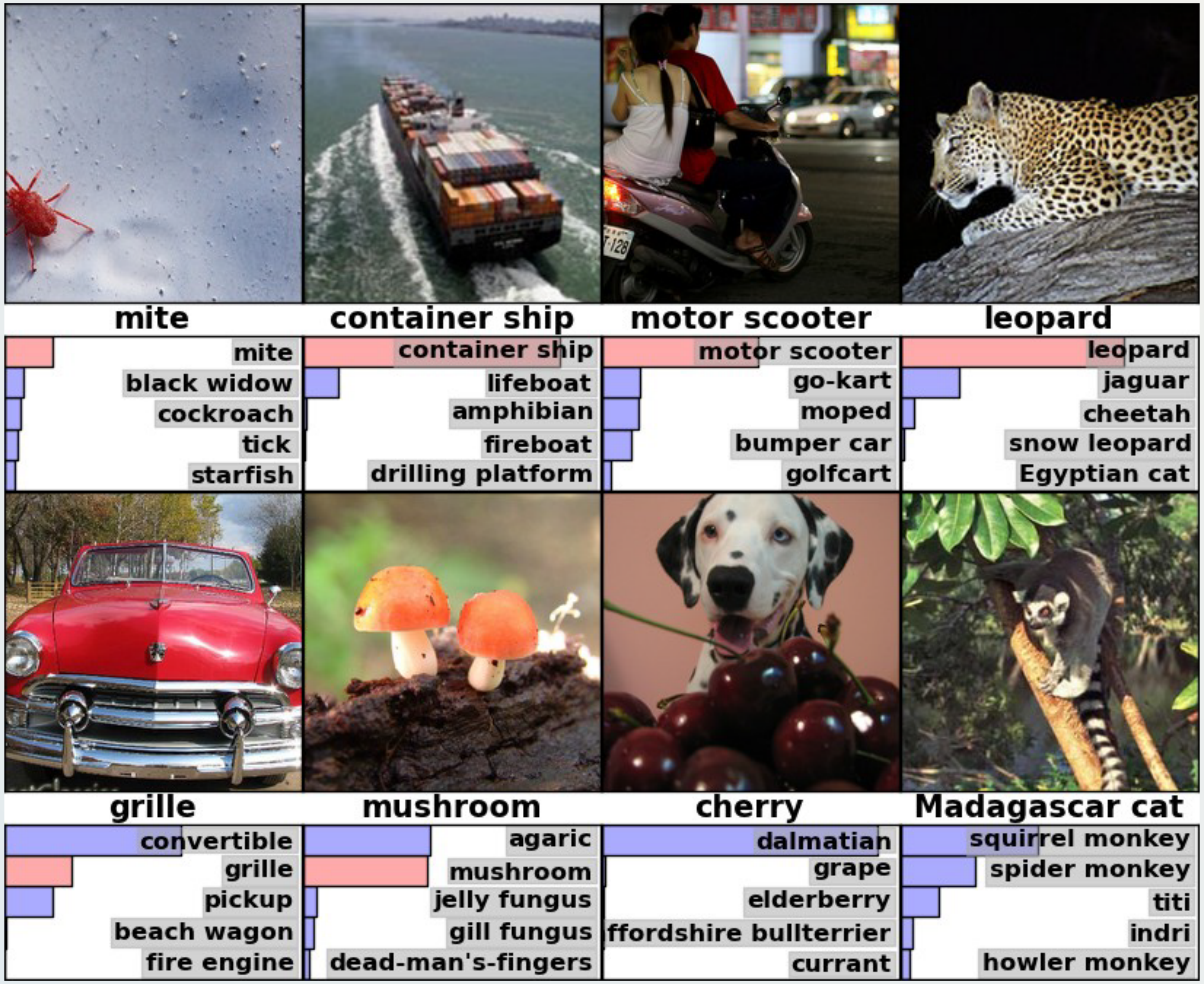
Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE 86(11): 2278–2324, 1998

Best result today: 0.21% error

Li Wan, Matthew Zeiler, Sixin Zhang, Yann LeCun, Rob Fergus Regularization of Neural Network using DropConnect, International Conference on Machine Learning 2013

ImageNet

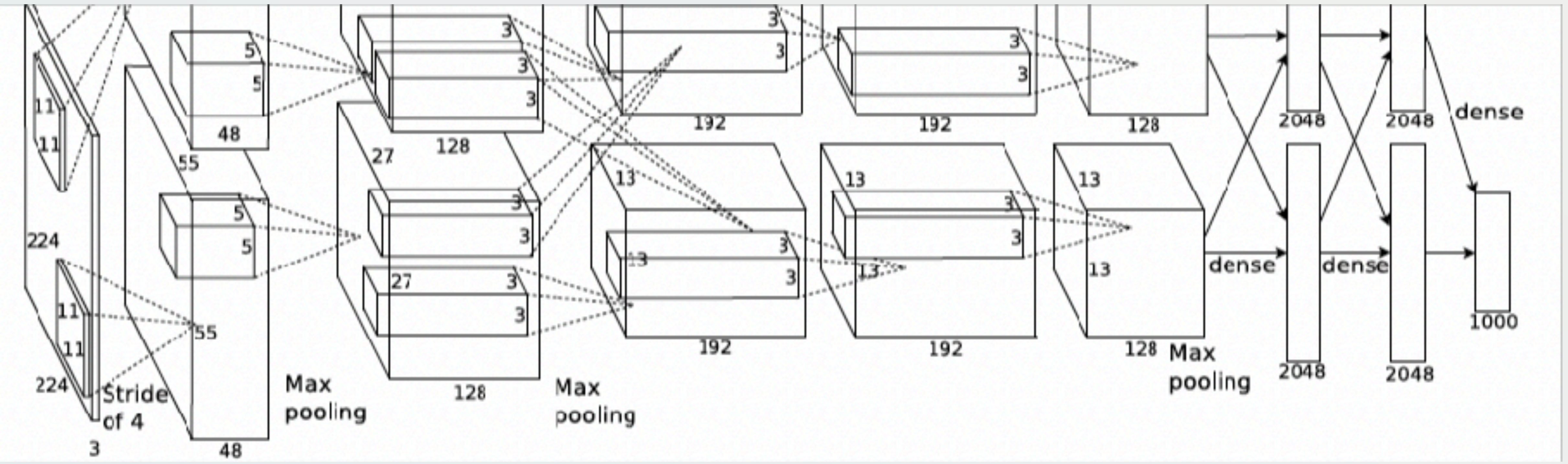


Popular AI challenge:

- **Crowdsourced labeling of image database (14 million labeled images)**
- **Competing algorithms try to classify them**

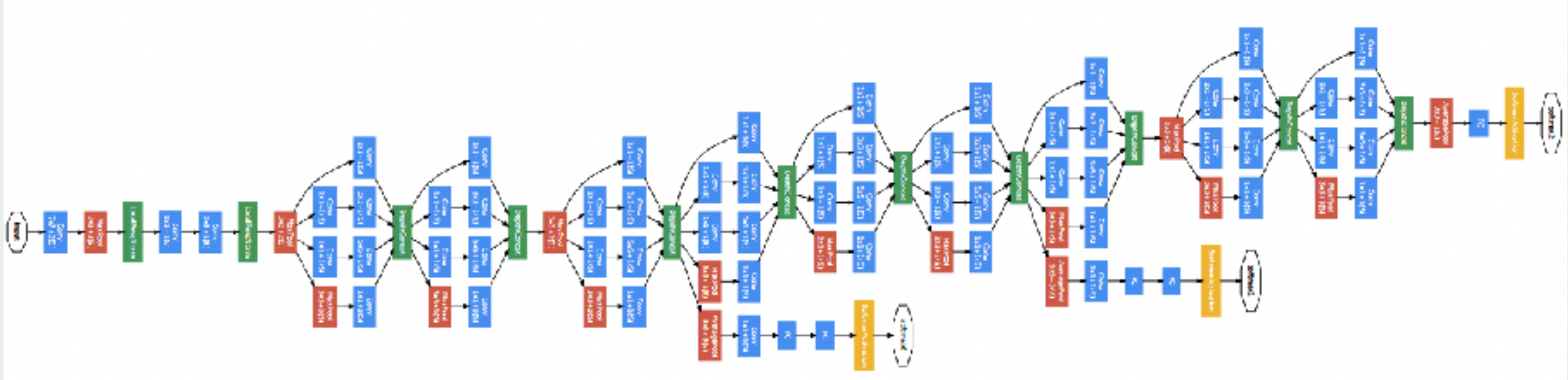
ImageNet

- Images are **Big Data** compared to MNIST

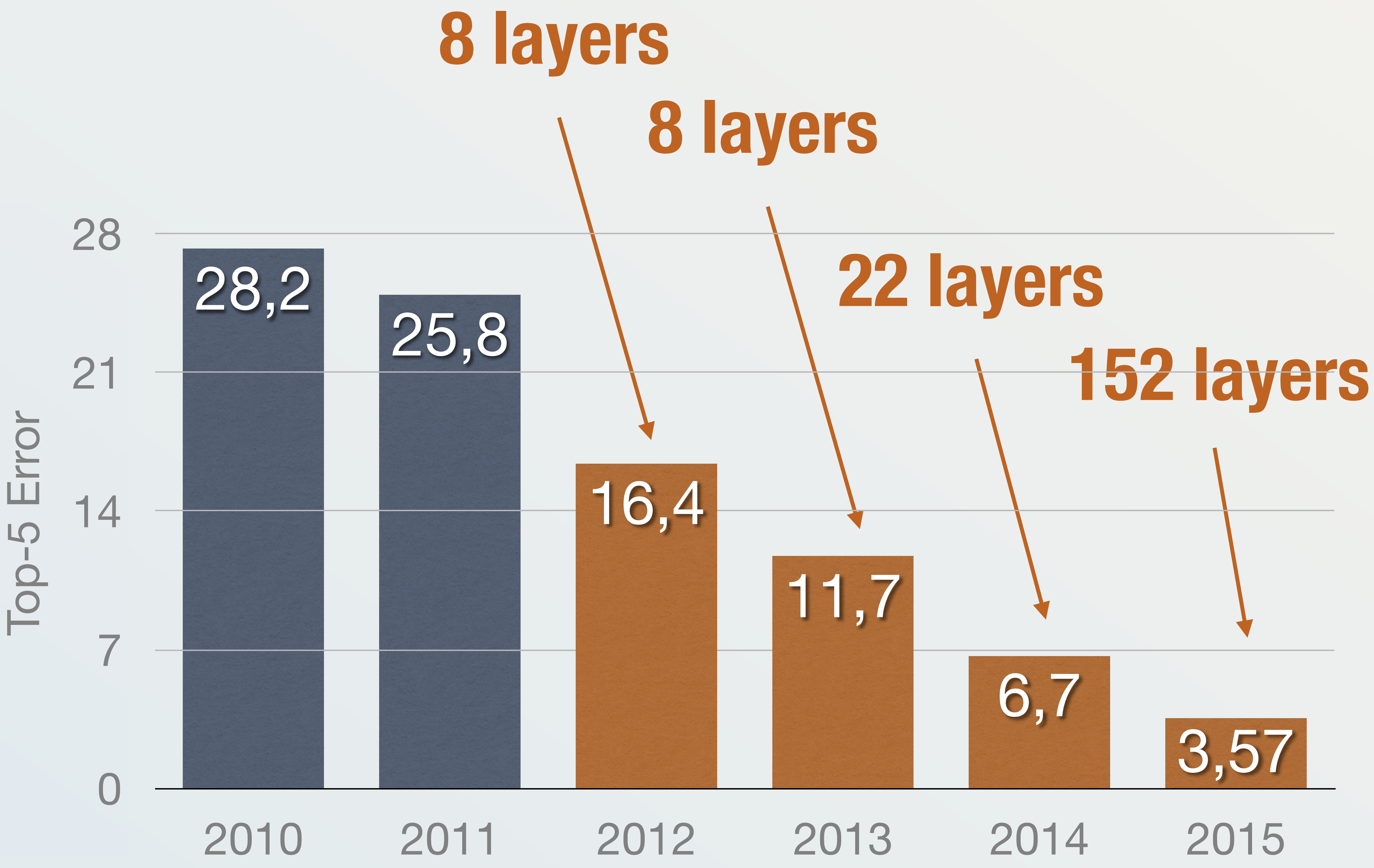


AlexNet: ImageNet 2012 winner

GoogLeNet: ImageNet 2014 winner



Deeper and deeper…



8 layers

8 layers

22 layers

152 layers

How do I use this?

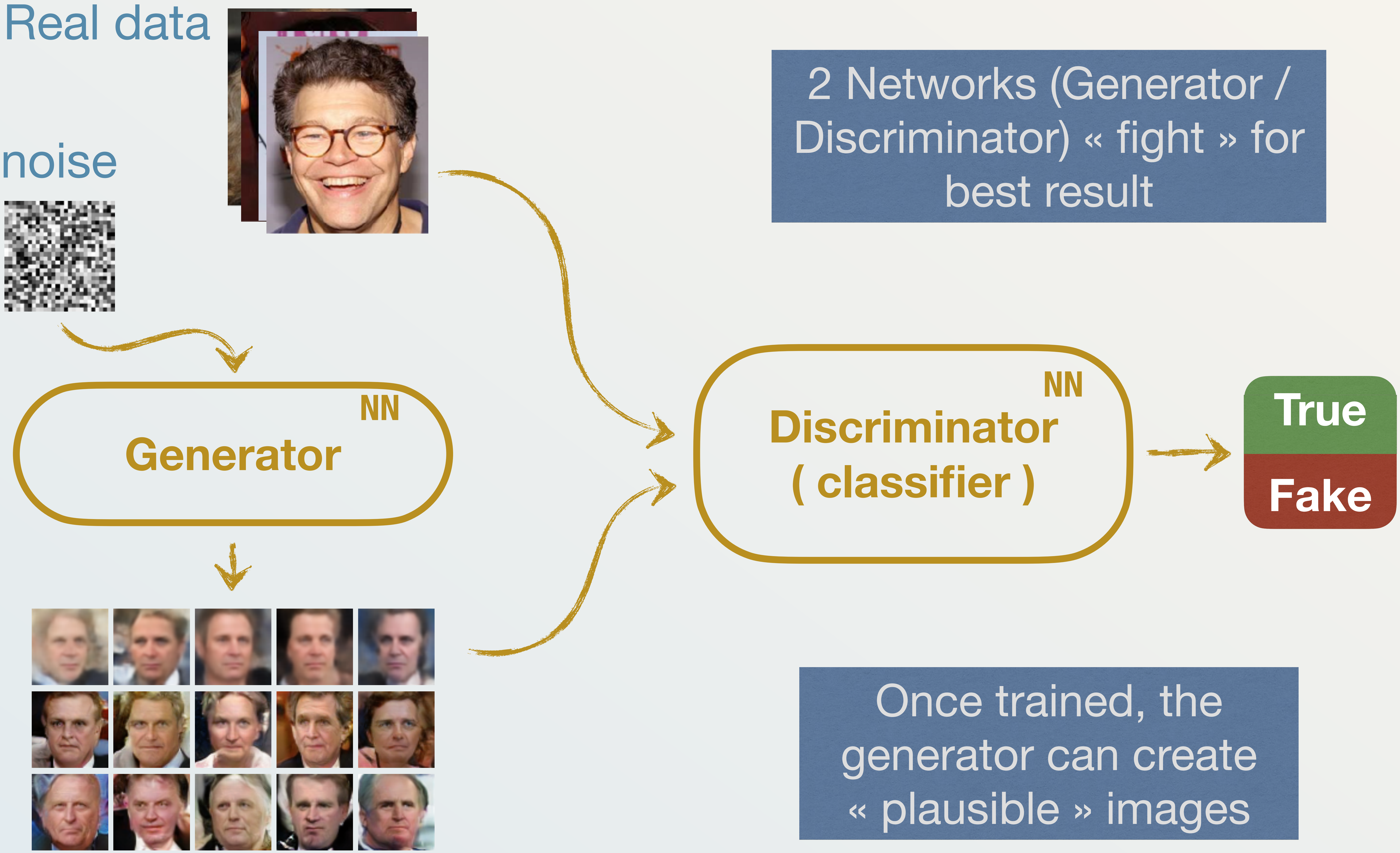
- Do **not** expect to make sense of the function.
  - ◉ GoogLeNet (22 layers) = 11,193,984 parameters
  - ◉ ResNet (153 layers) = 25,636,712 parameters

- Deep neural **classifiers** are high performers

- But deep learning is not just about classification! Can it do anything else?

A few modern networks

Adverserial generators

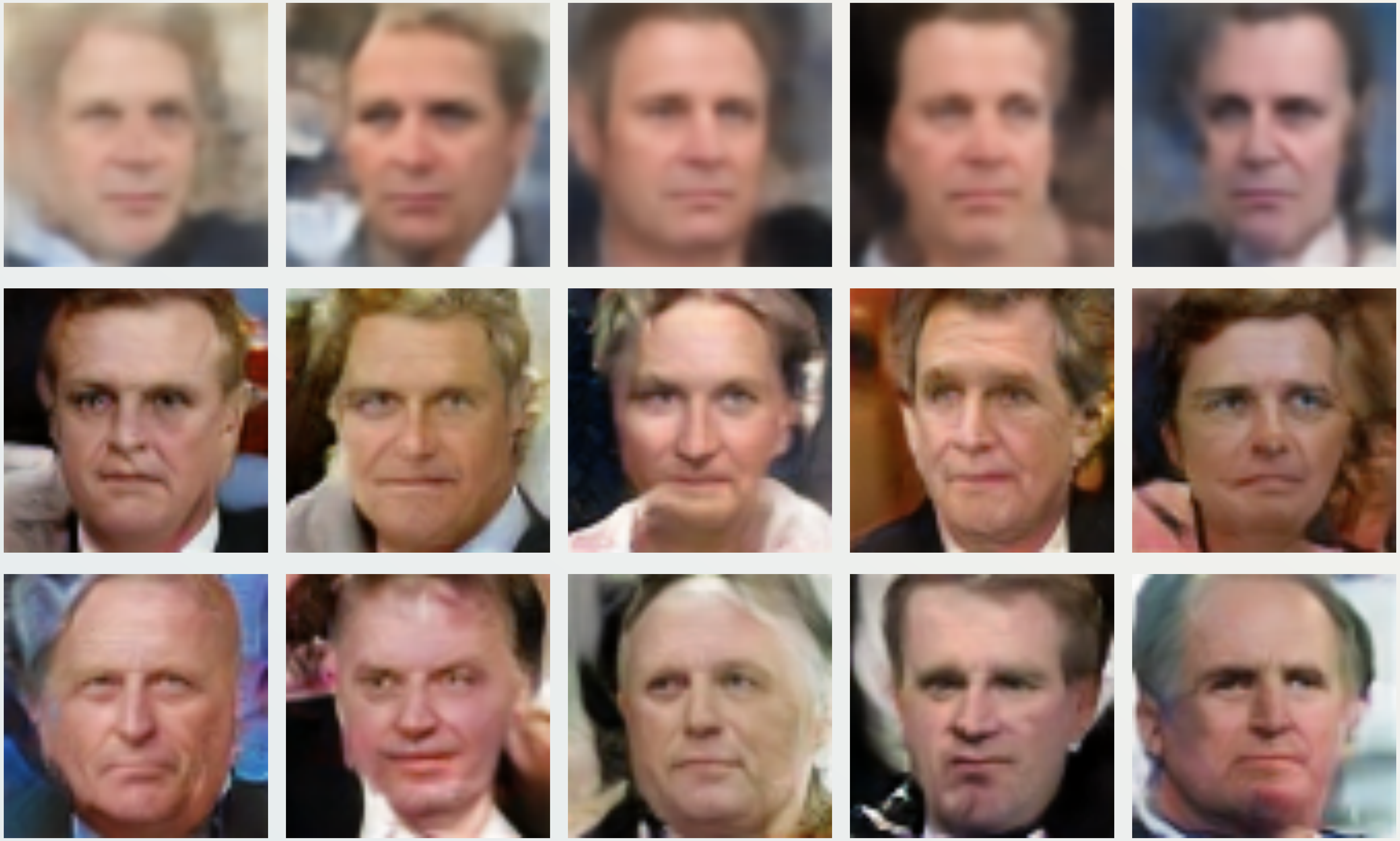


2 Networks (Generator / Discriminator) « fight » for best result

Once trained, the generator can create « plausible » images

GAN examples

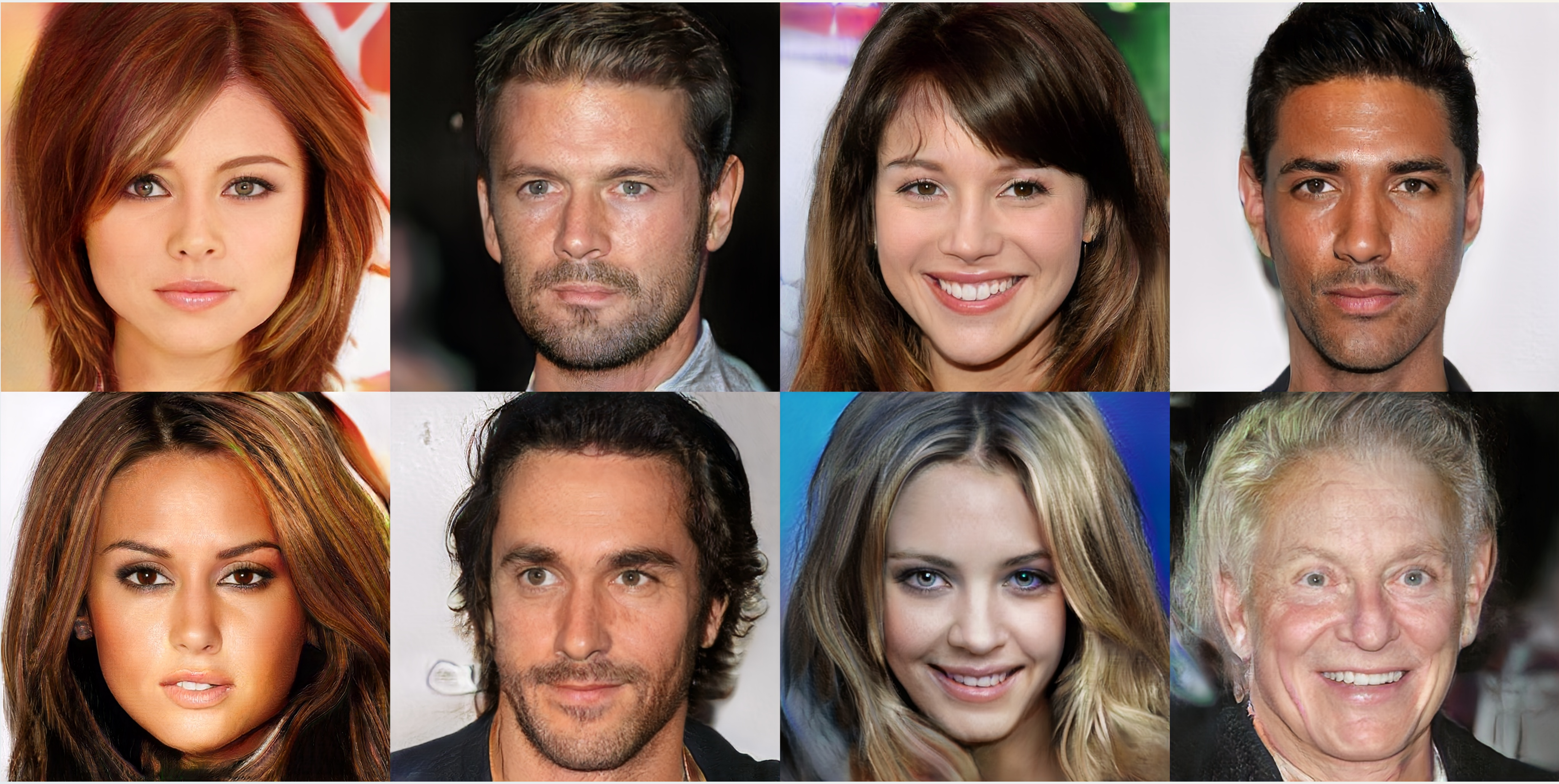
State of art 2015



Larsen, Anders Boesen Lindbo, et al. "Autoencoding beyond pixels using a learned similarity metric." arXiv preprint arXiv:1512.09300 (2015).

GAN examples

State of art 2017



Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196.

https://thispersondoesnotexist.com

GAN examples

State of art 2019

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of stylegan.
In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8110-8119).

GAN examples

https://thispersondoesnotexist.com

State of art 2019

Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T. (2020). Analyzing and improving the image quality of stylegan.
In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 8110-8119).

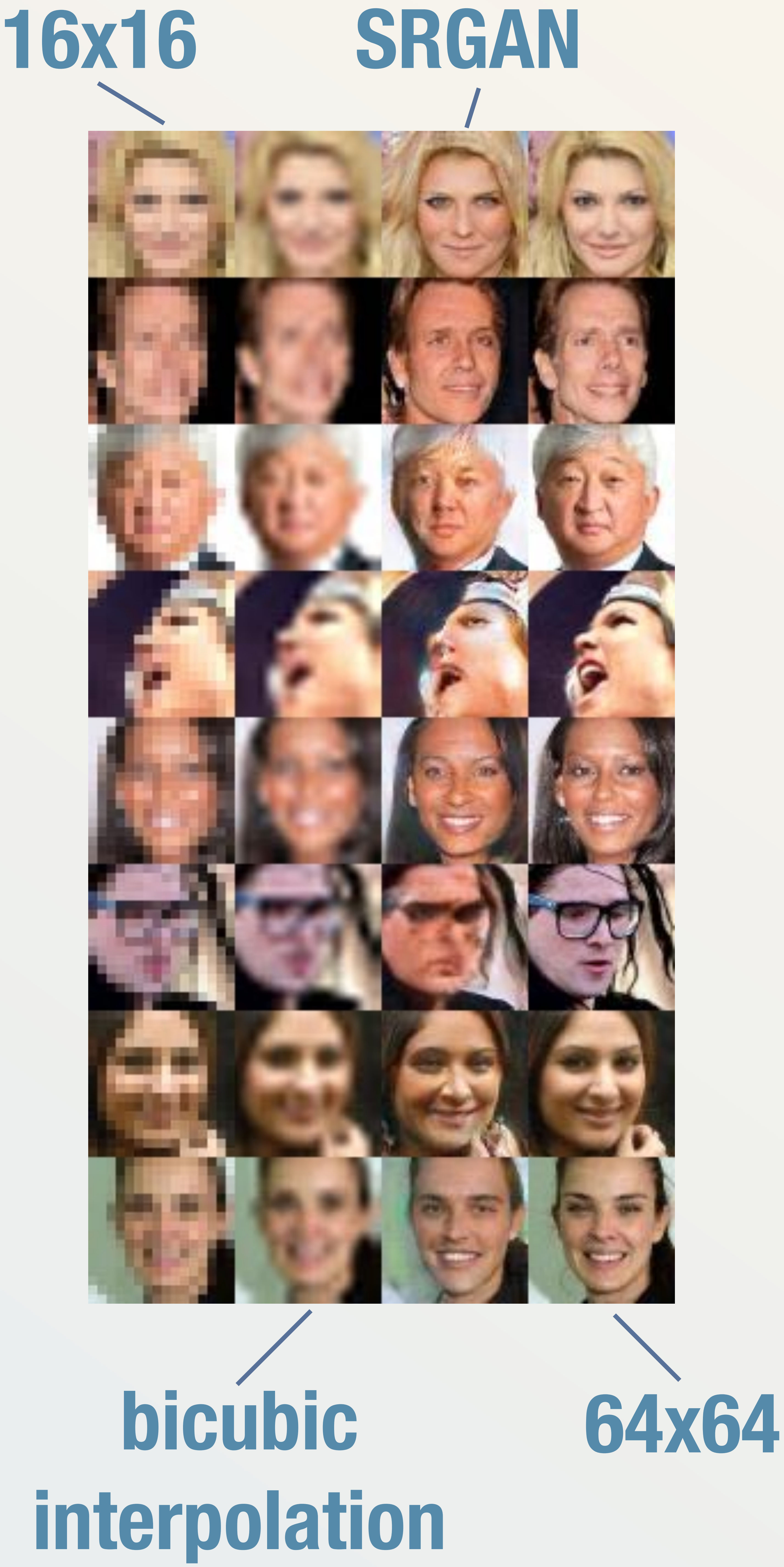
SRGAN

Super - Resolution GAN

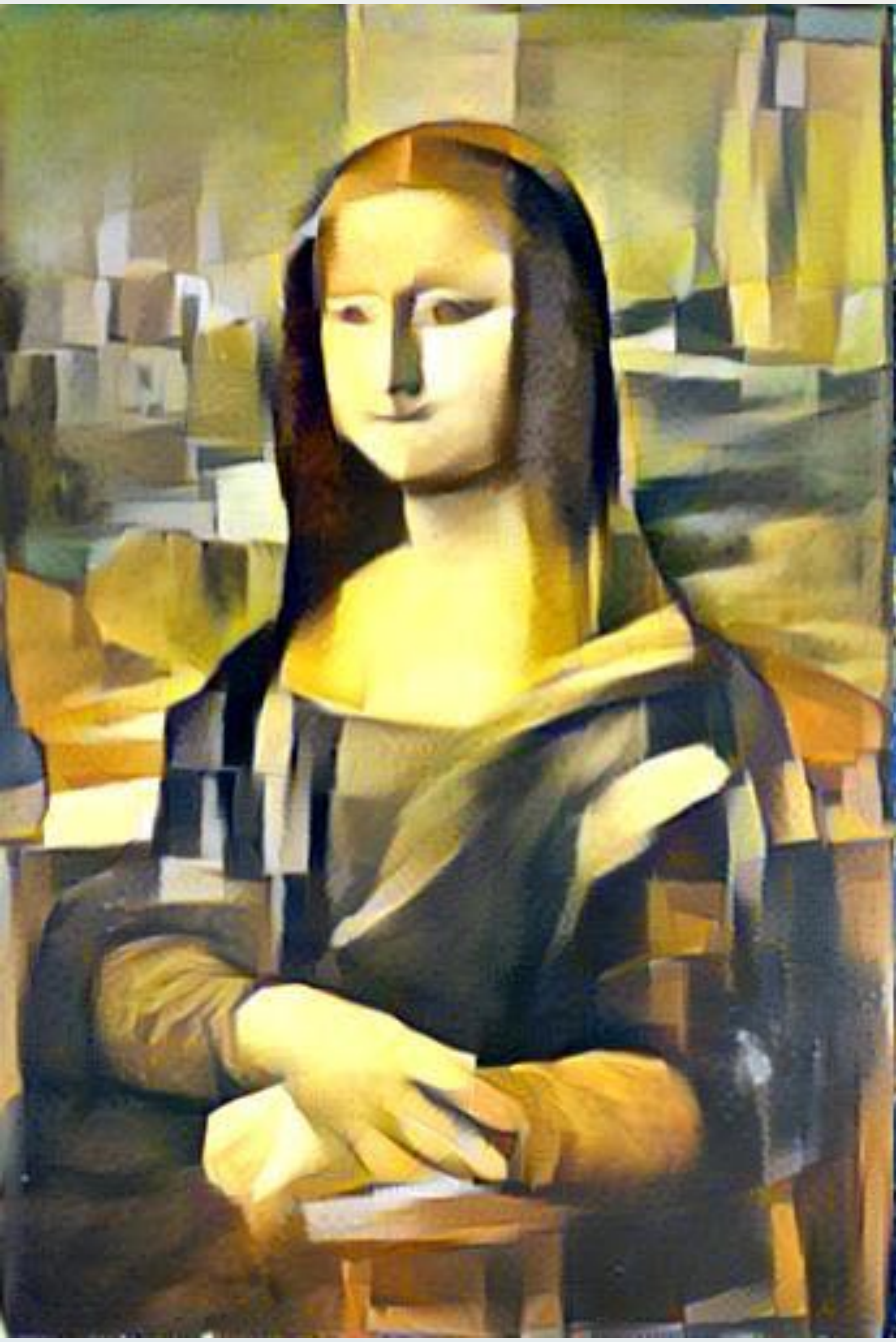
- Output: 64x64 images (from the `Large-scale CelebFaces Attributes` dataset)

- Input: degraded 16x16 image

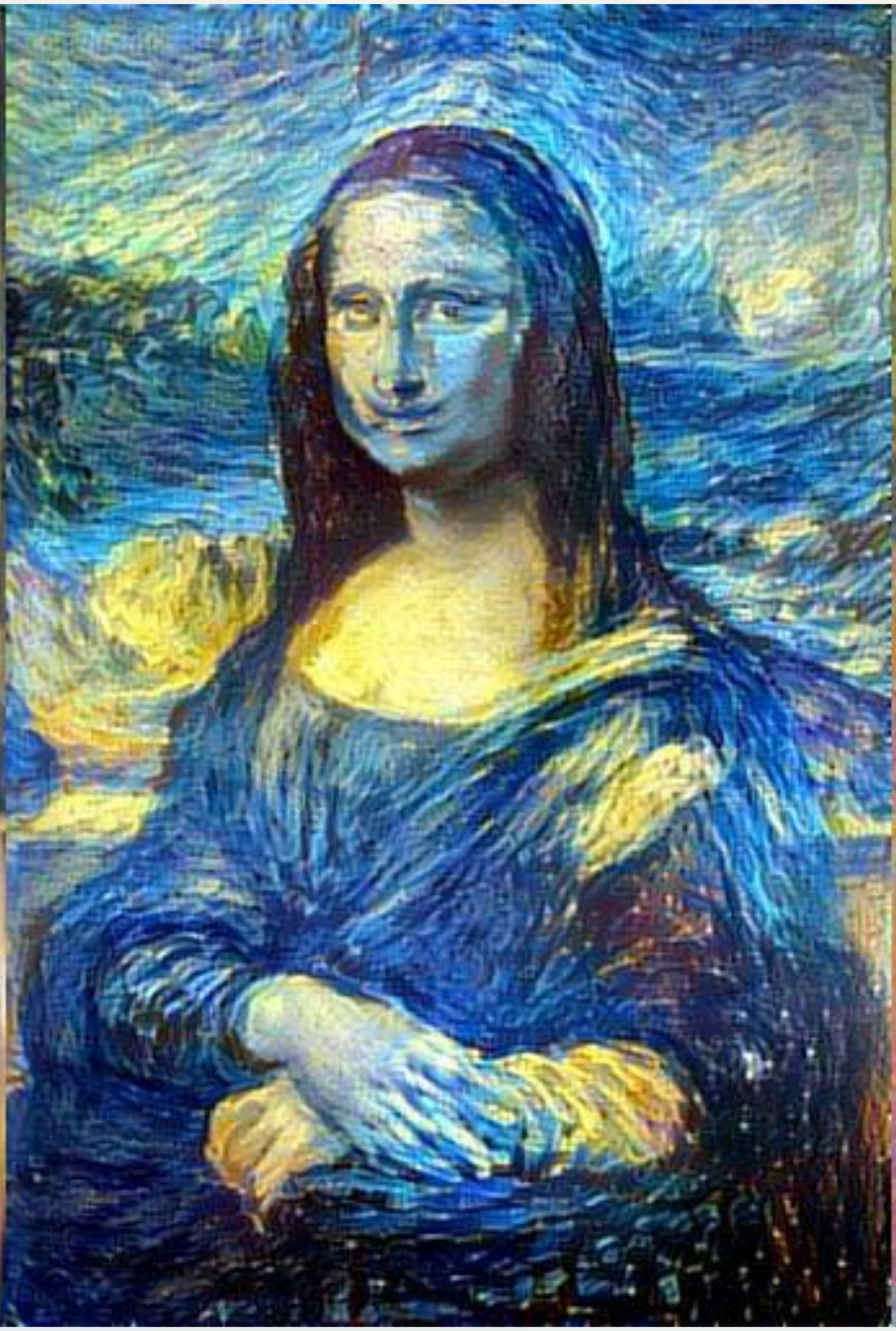
- GAN learns to reproduce « credible » images



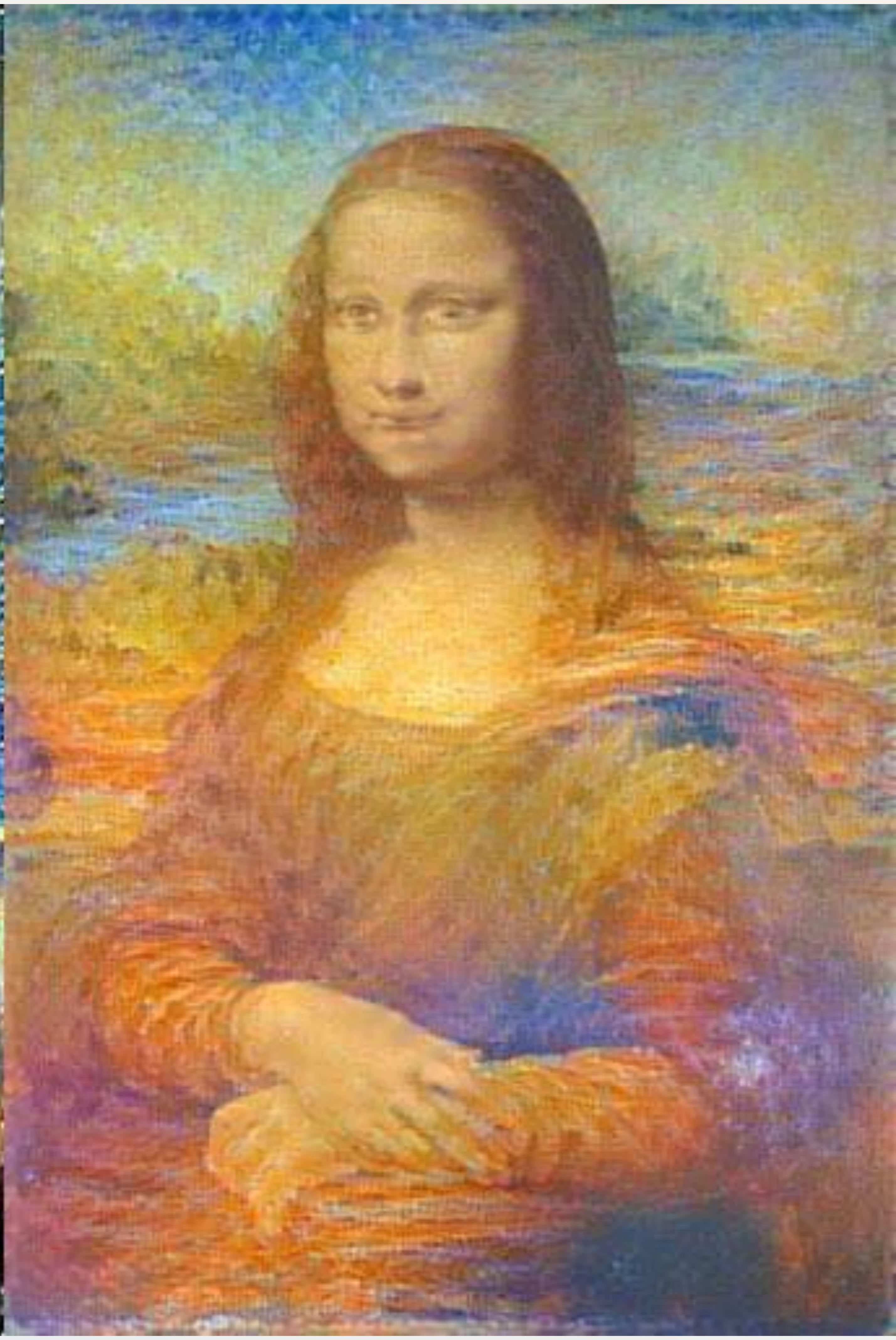
Style Transfer



Picasso

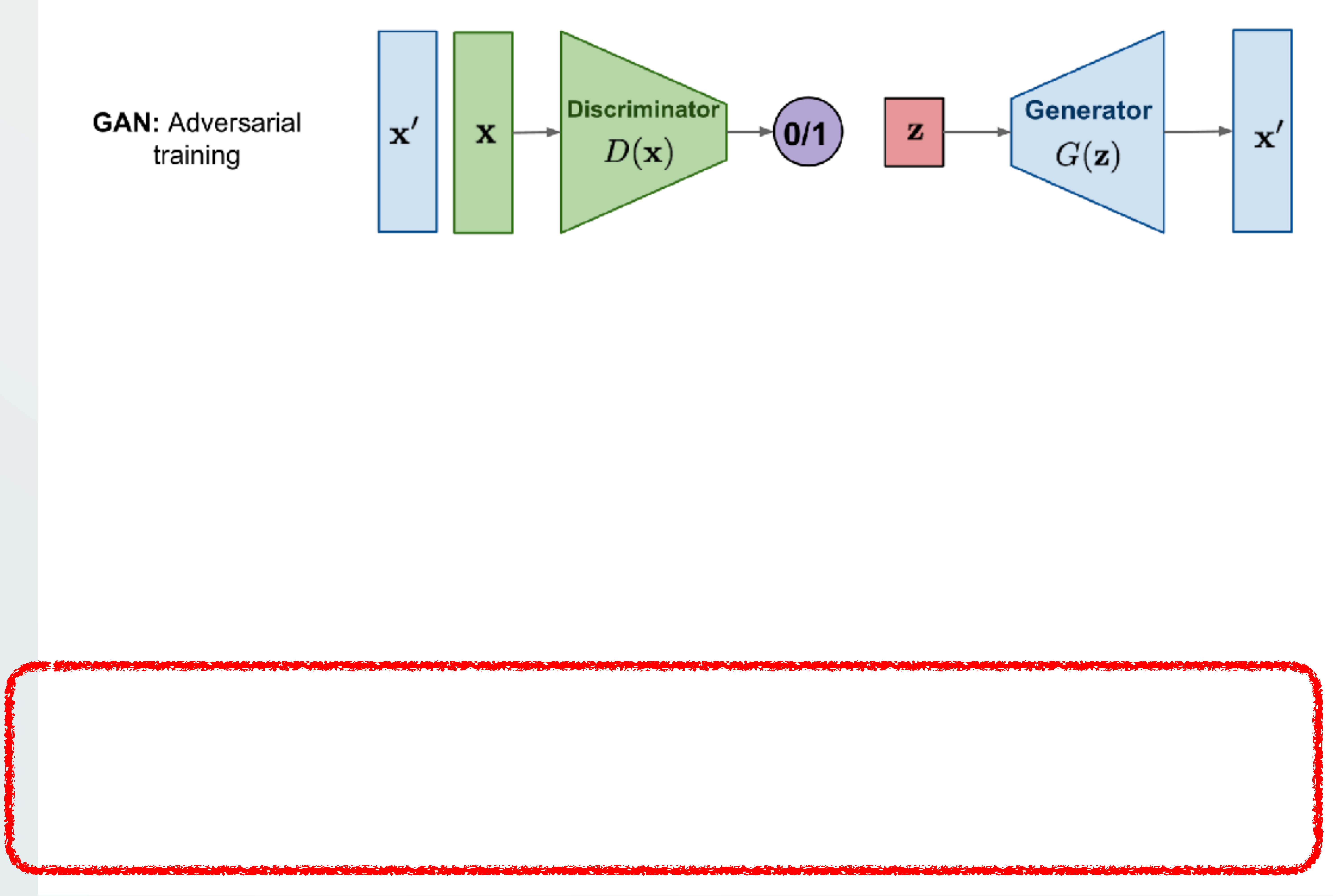


van Gogh



Monnet

Many kinds of generators



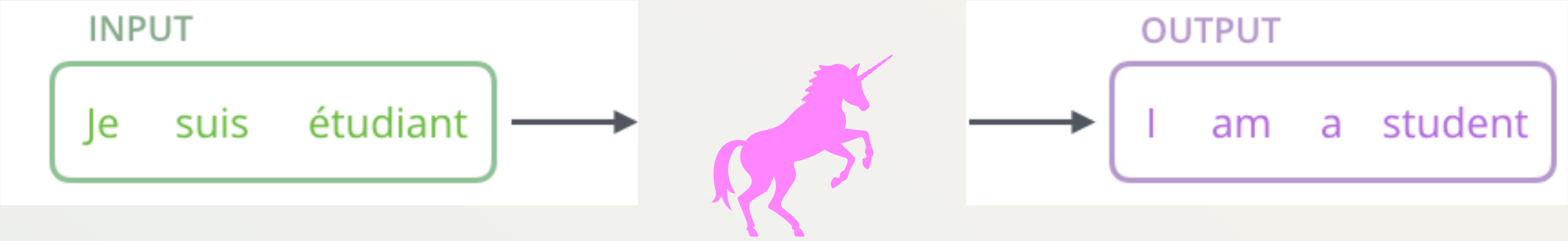
https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

Natural Language Processing

« A language model is a probability distribution over sequences of words. »

Jurafsky, Dan; Martin, James H. (2021). "N-gram Language Models”
https://web.stanford.edu/~jurafsky/slp3/

Learning language tasks



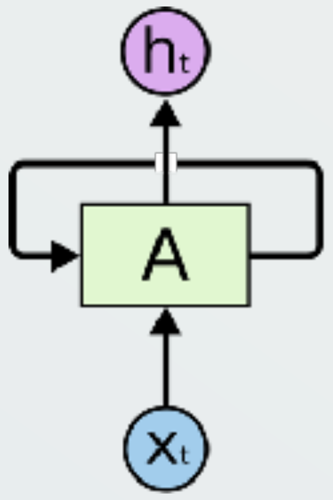
INPUT

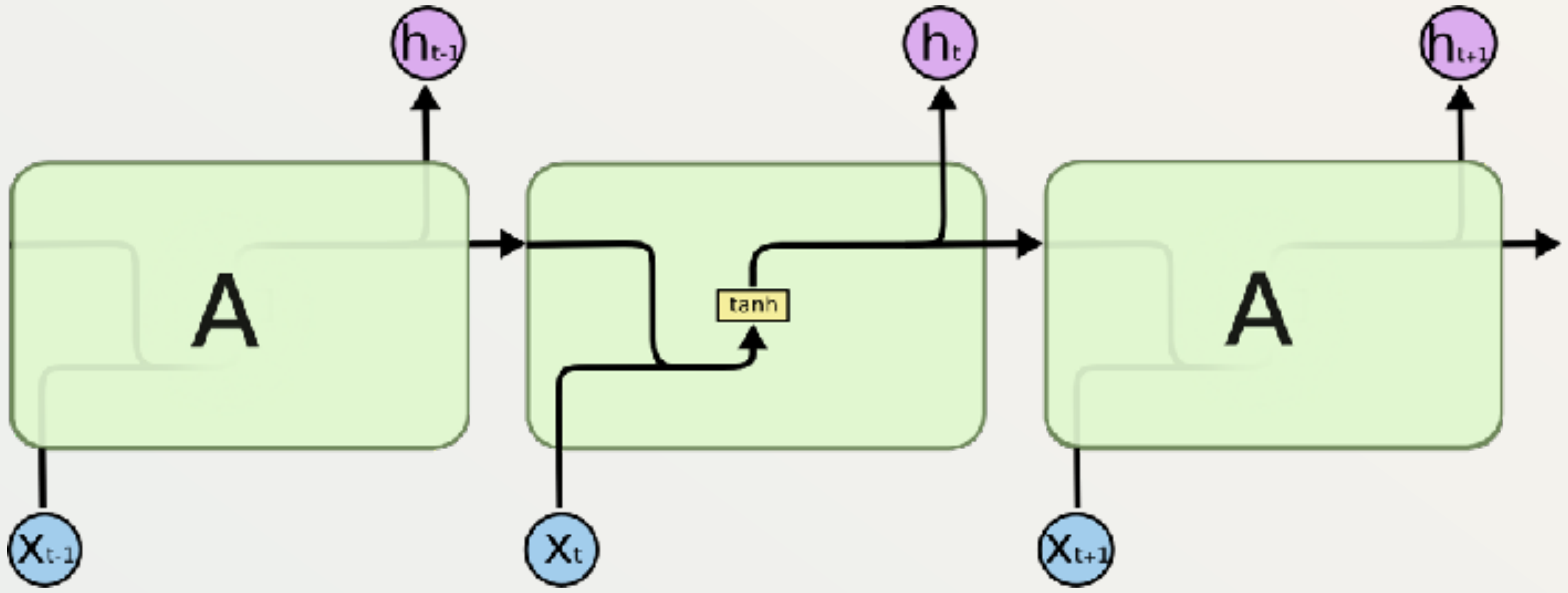
Je suis étudiant

OUTPUT

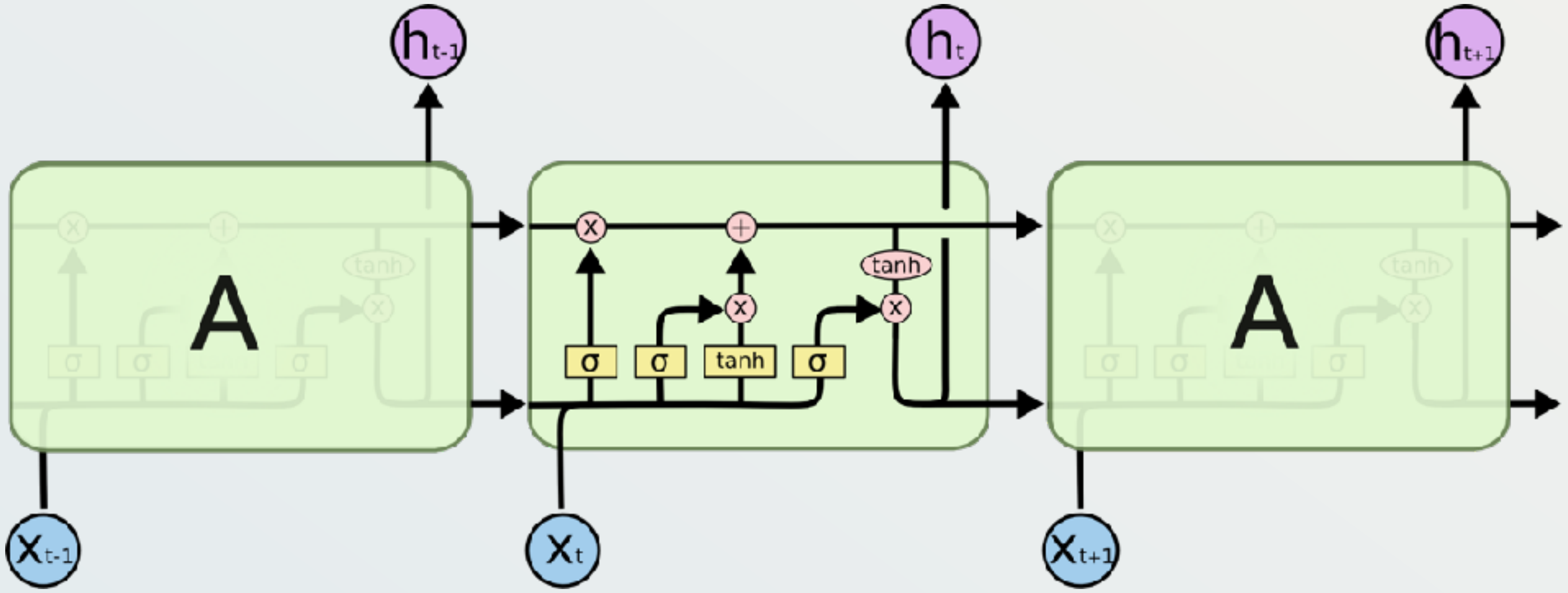
I am a student

Recurrent models



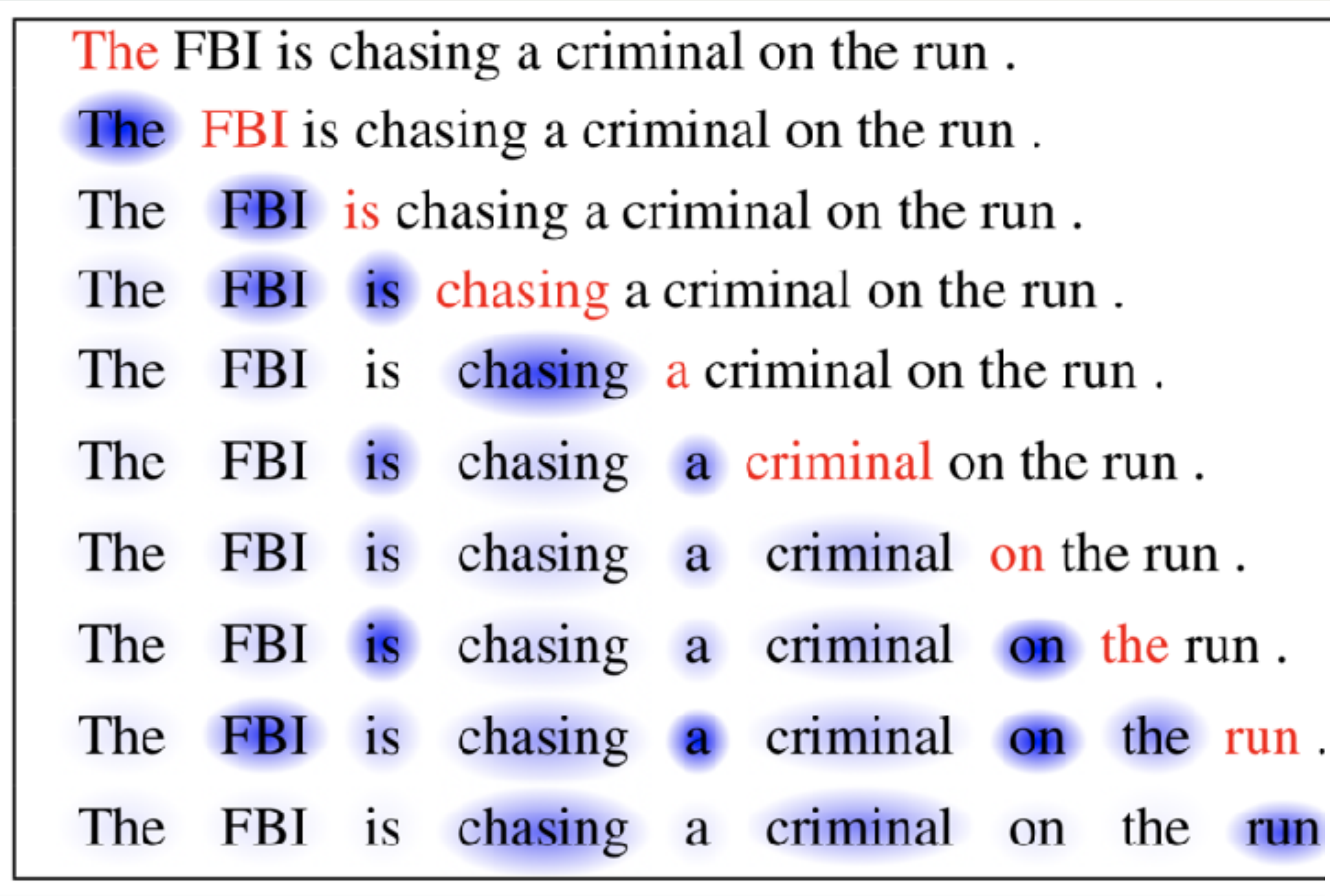


LSTM = Long Short Term Memory

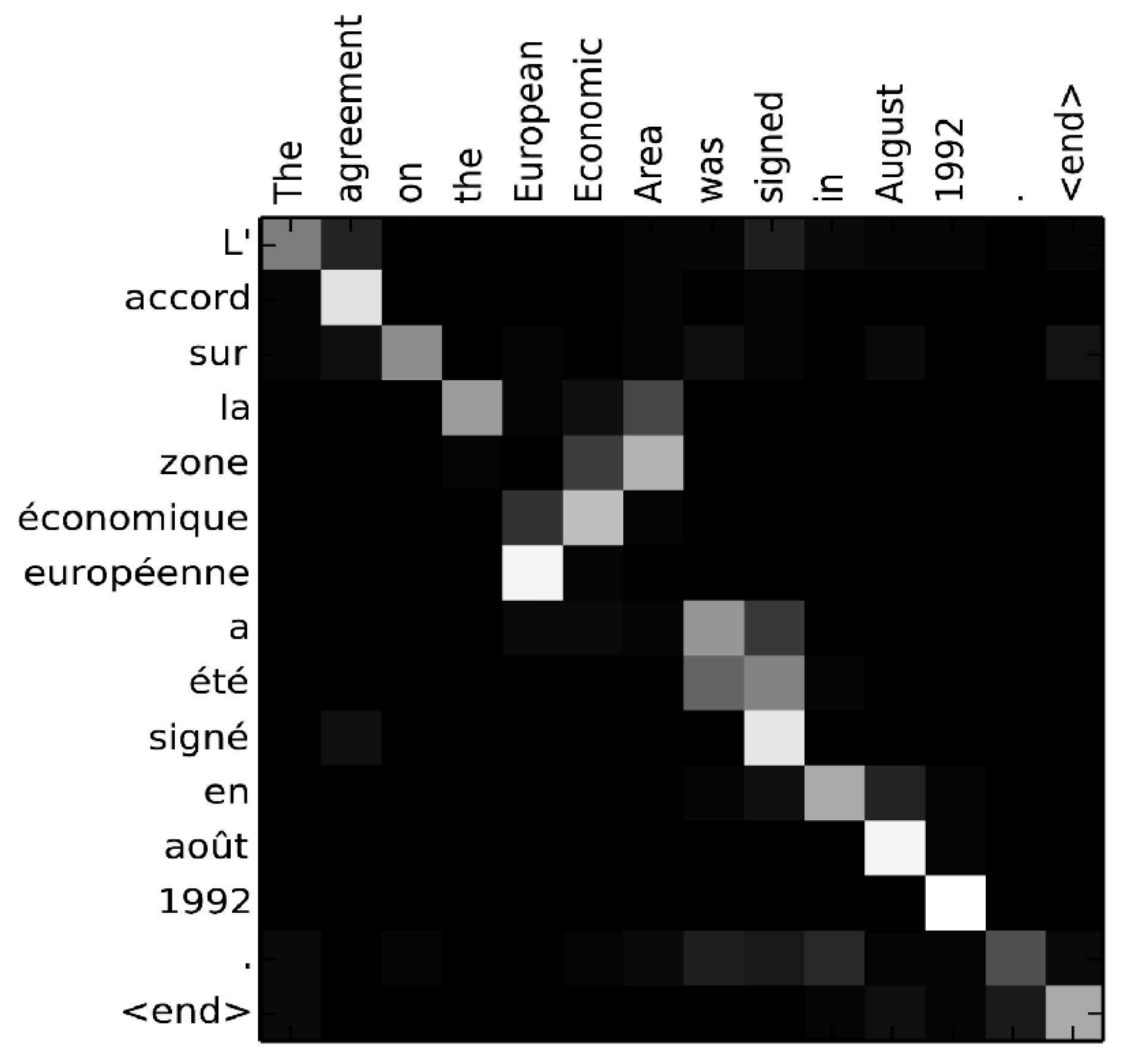


https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Attention



The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run



Jianpeng Cheng, Li Dong, and Mirella Lapata. "Long short-term memory-networks for machine reading." EMNLP 2016.

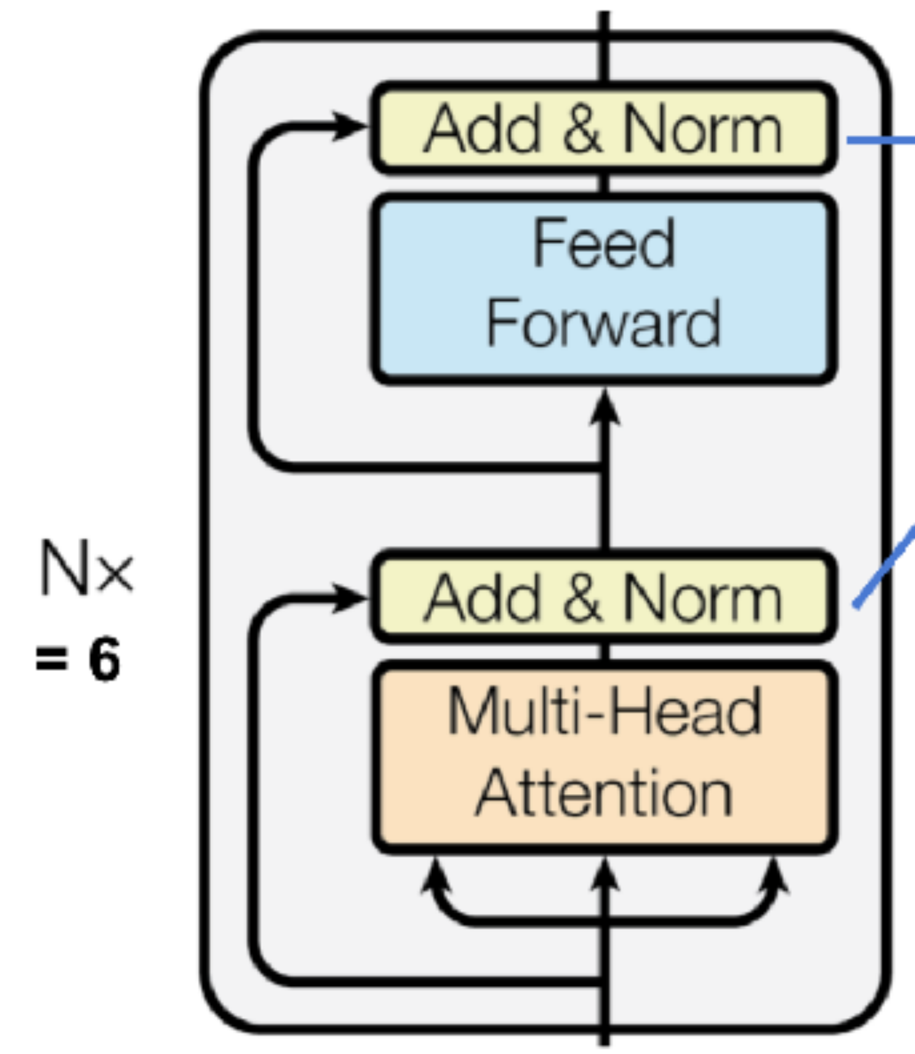
Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." ICLR 2015.

https://lilianweng.github.io/posts/2018-06-24-attention/

The Transformer

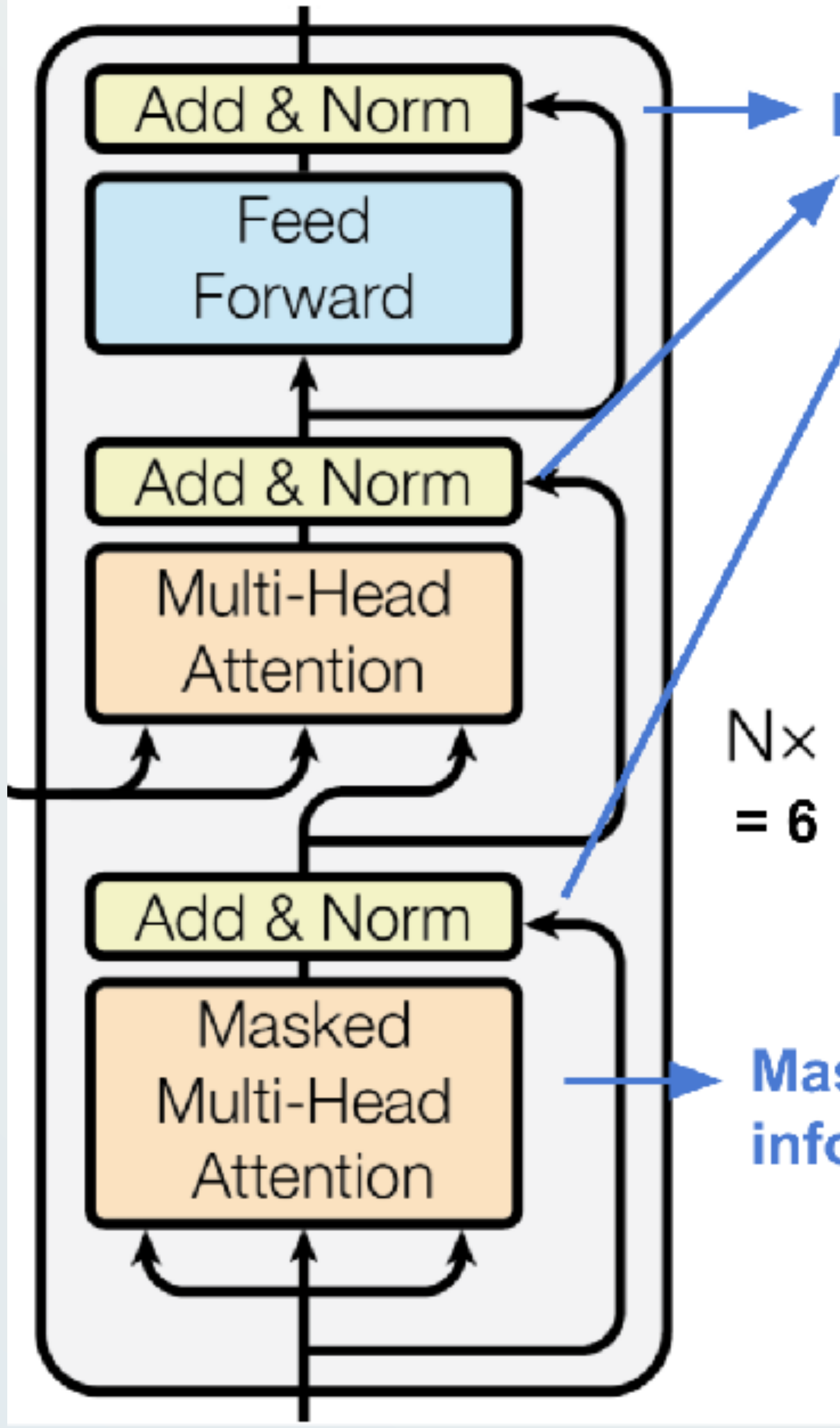
The Transformer

Encoder



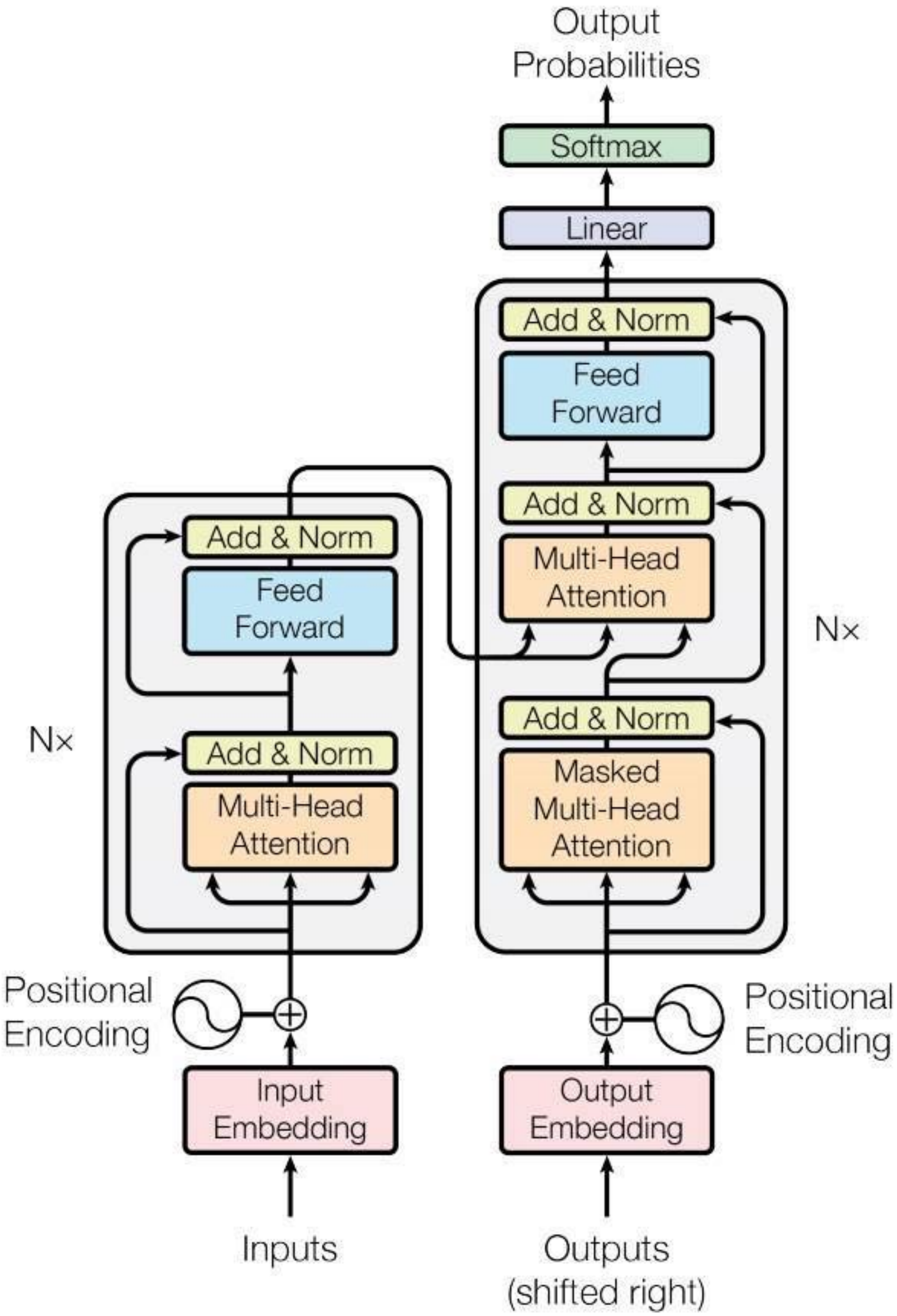
LayerNorm(x + SubLayer(x))

Decoder



LayerNorm(x + SubLayer(x))

Masked: not to use the information in the future.



s

ChatGPT

Pro

Expl

Joke
thei
diff

M

Pred
for
grou
comm
pret
grou

ne

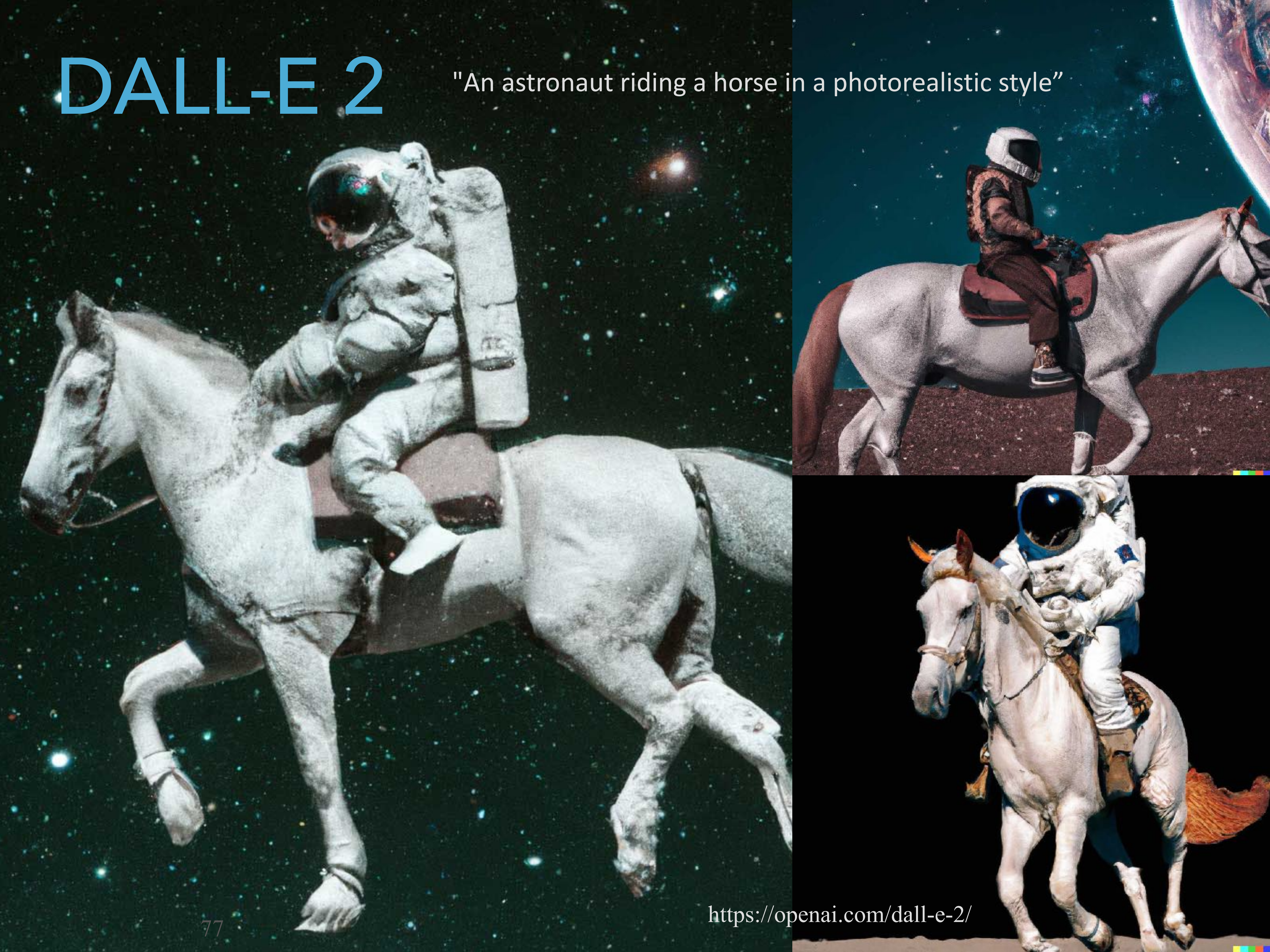
```
mo
        for j in range(i, (n + 1)):
            sm = (sm + (i * j))
        return sm
```

```
# 3.
print(total_length)
```

“AI system[s] that can create realistic images and art from a description in natural language.”

https://openai.com/dall-e-2/

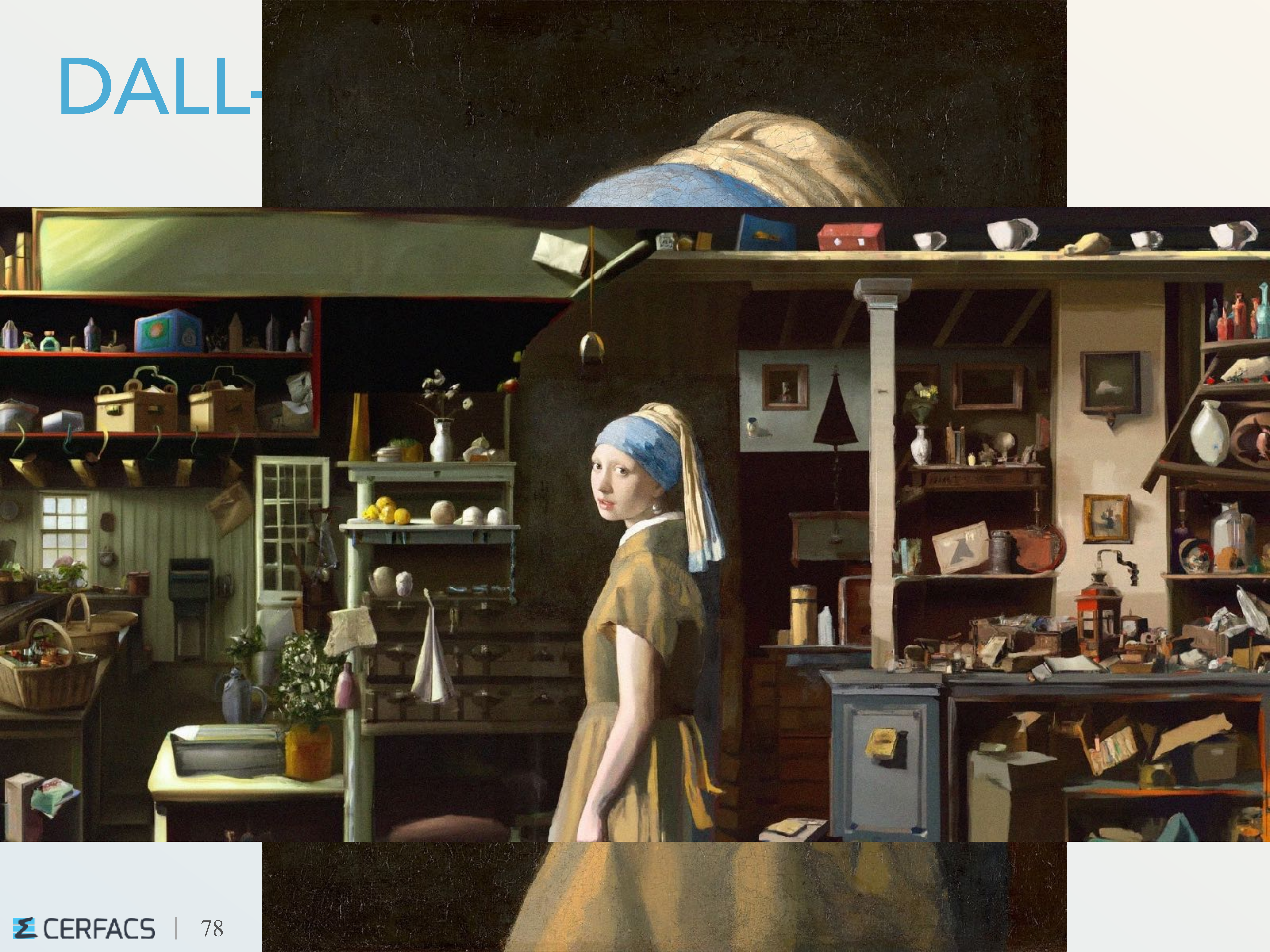
DALL-E 2



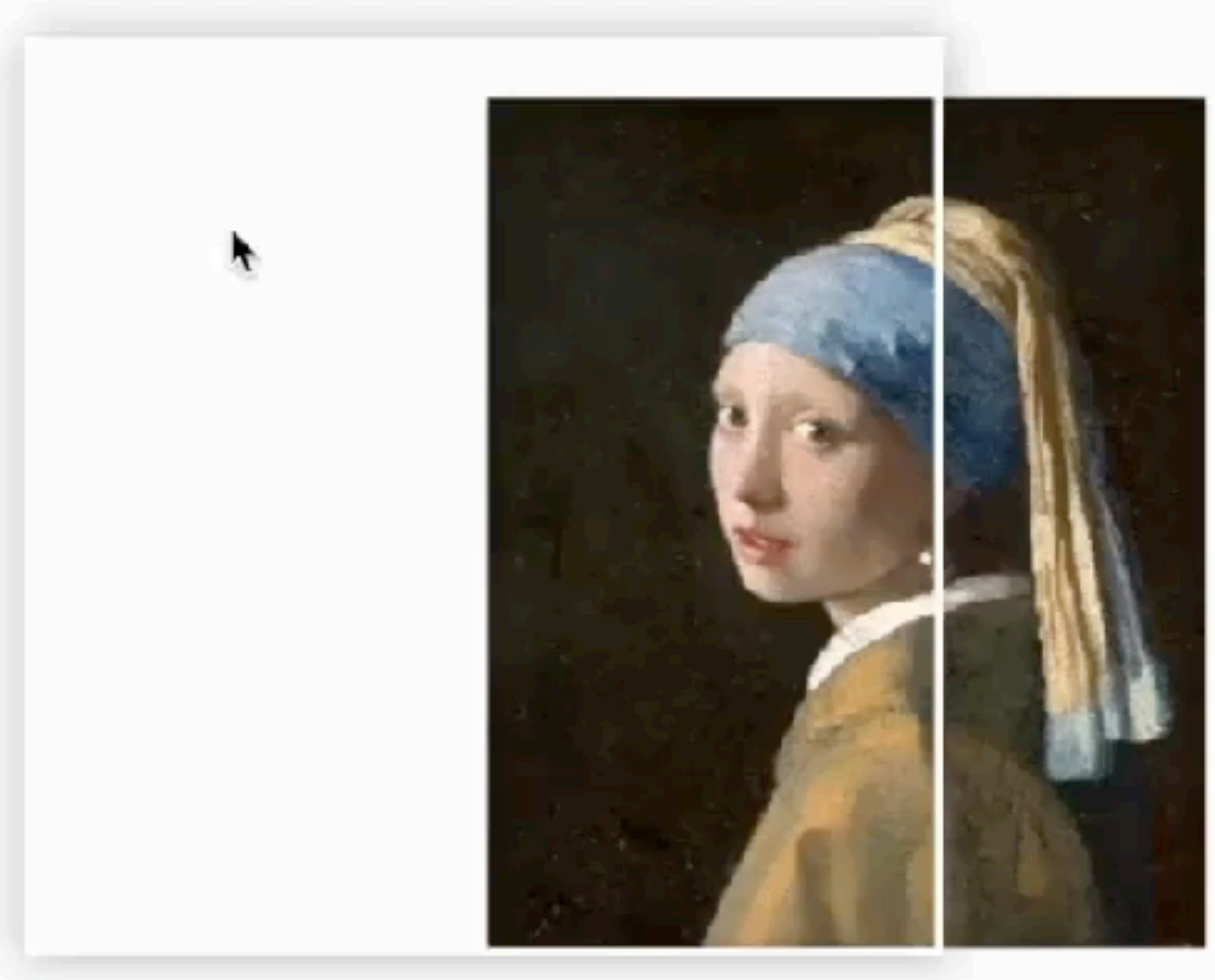
"An astronaut riding a horse in a photorealistic style”

https://openai.com/dall-e-2/

DALL-



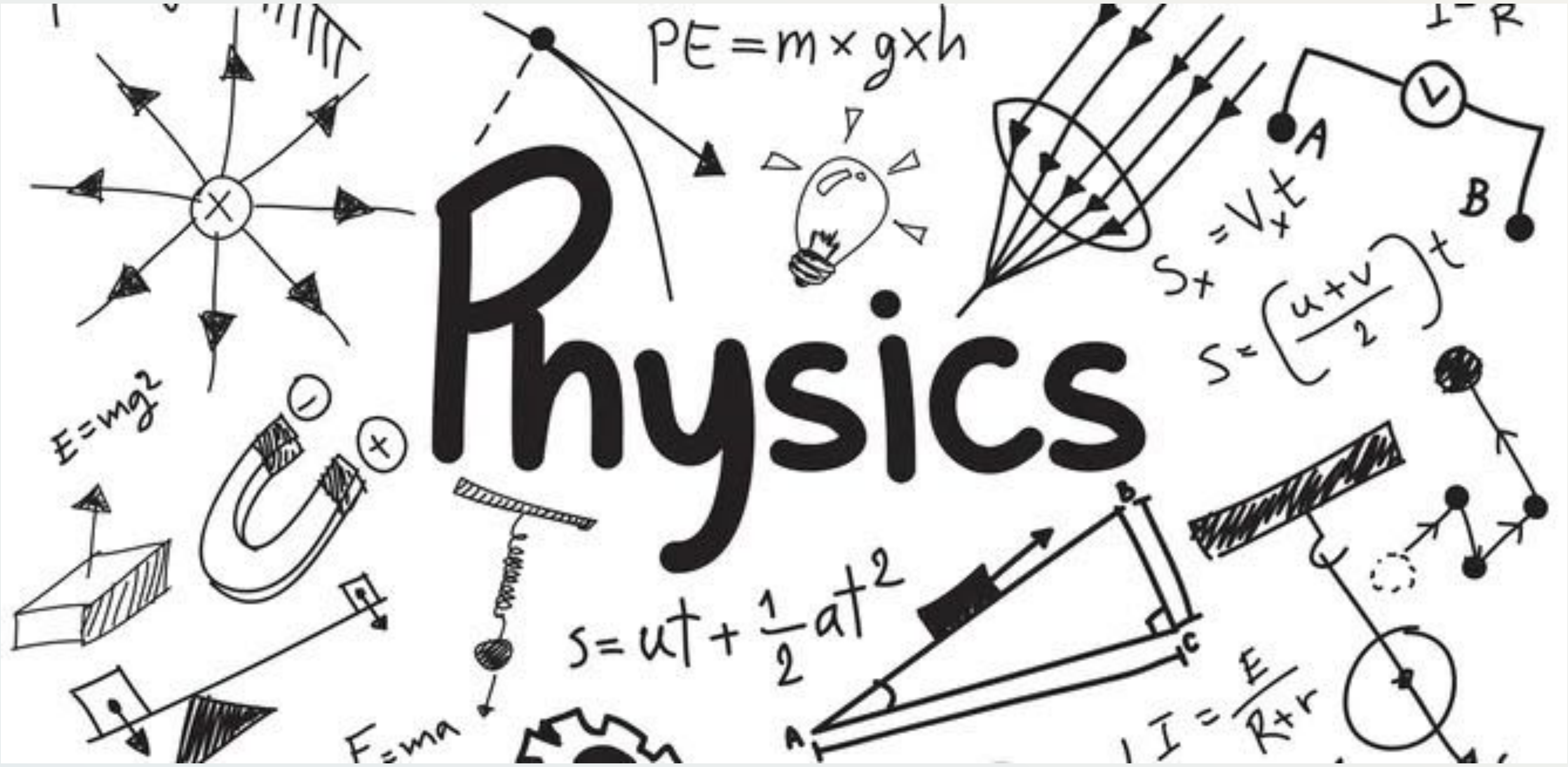
DALL-E 2 : Outpainting

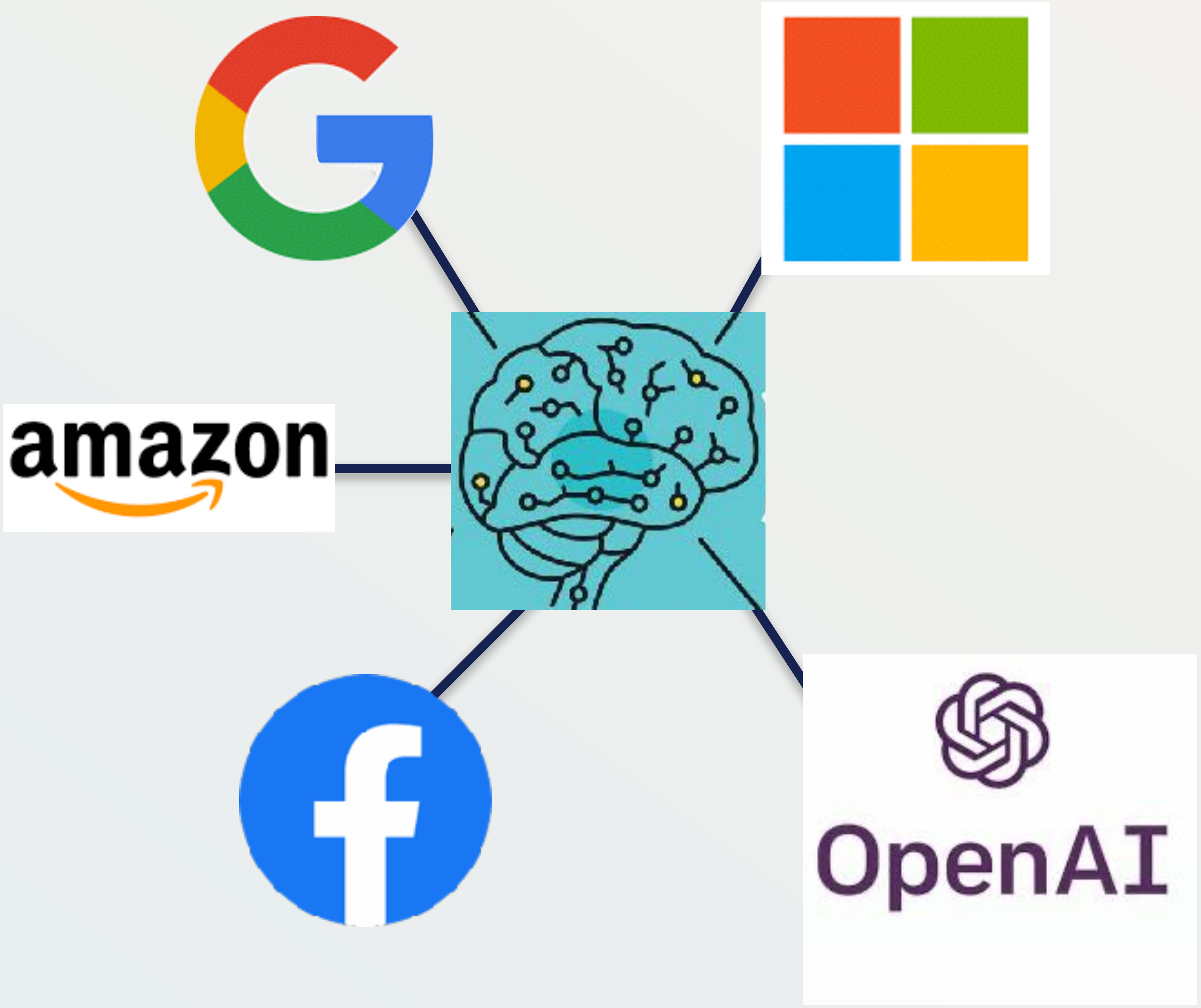


https://openai.com/blog/dall-e-introducing-outpainting/

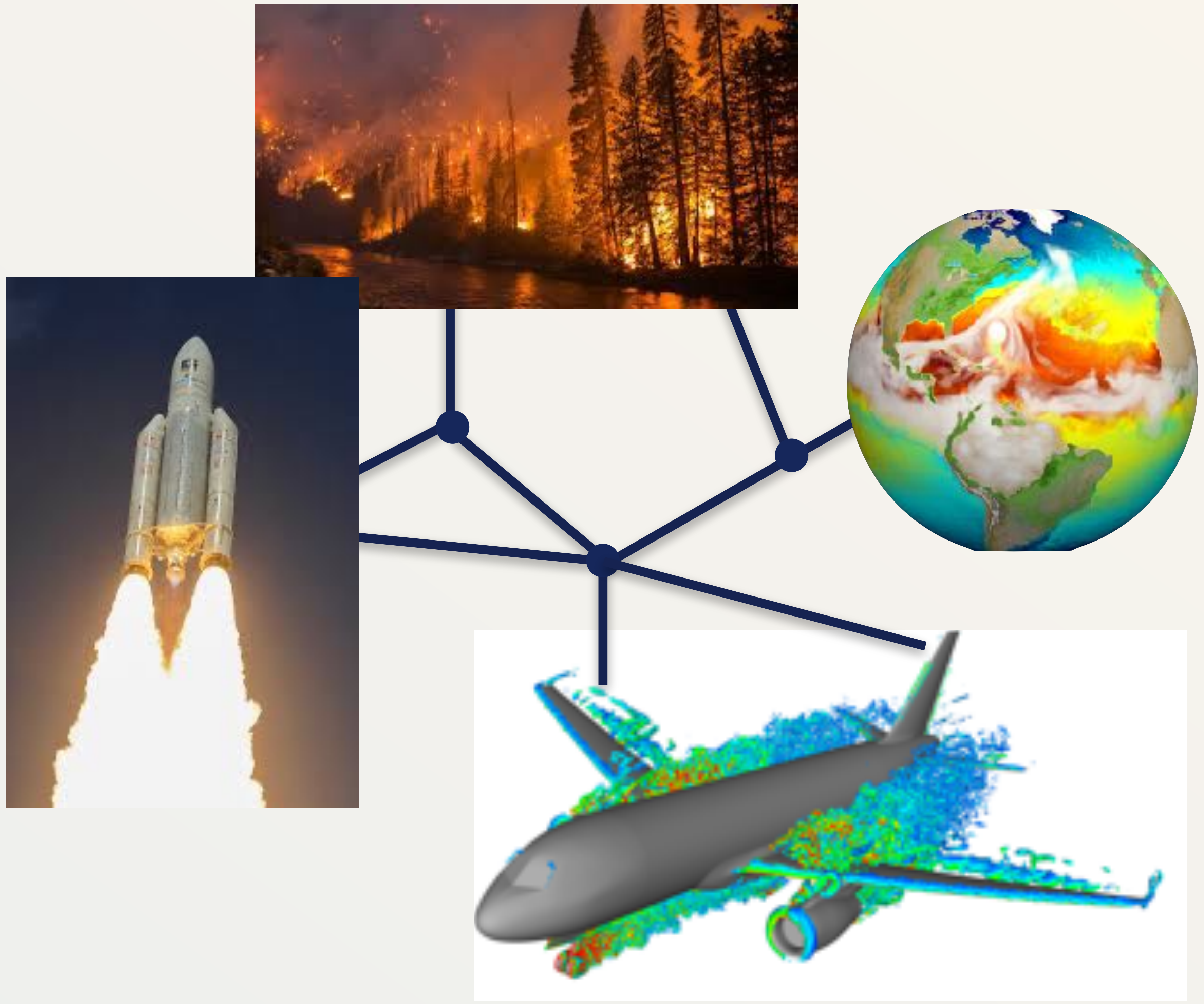
Try it yourself! :

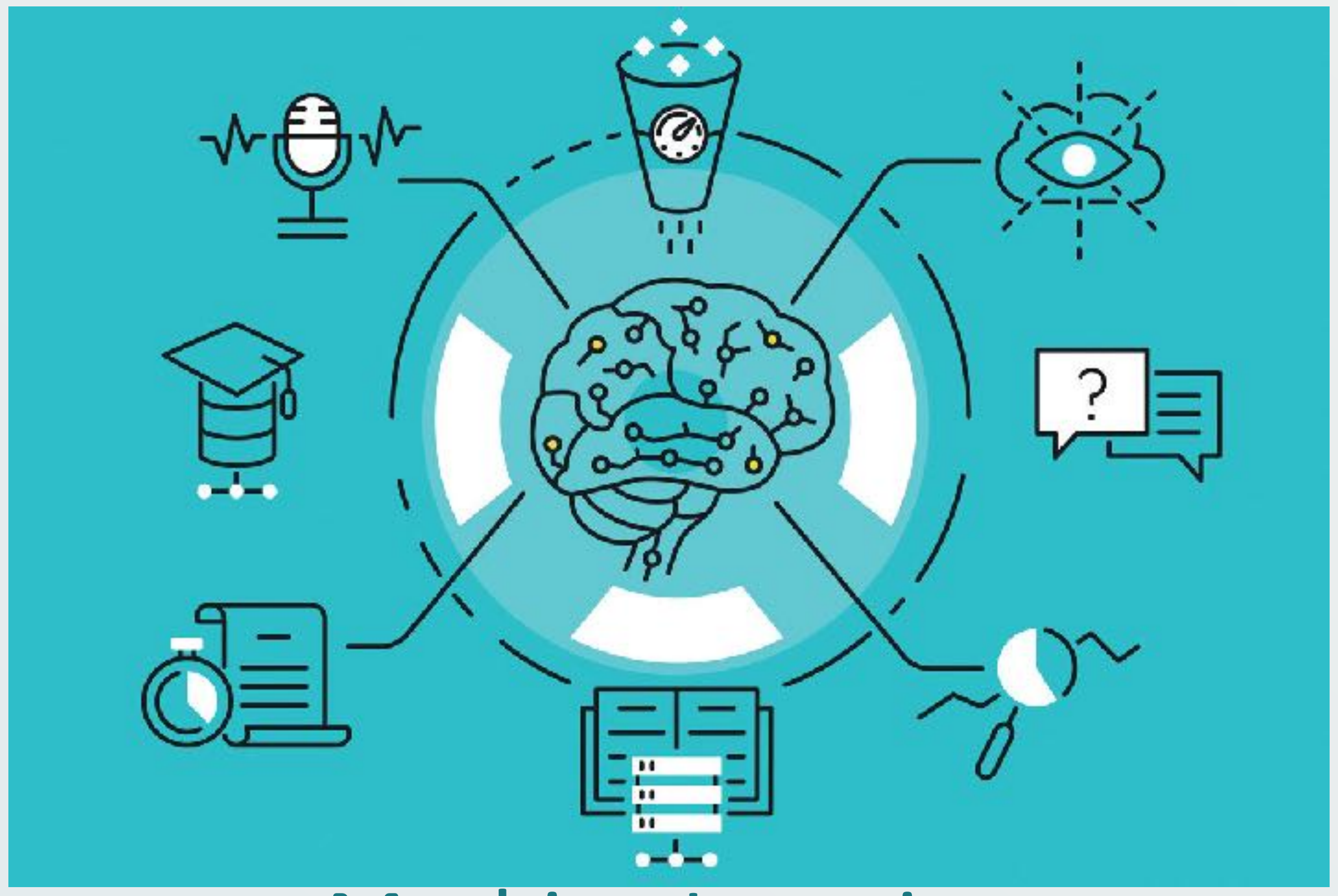
Back to


Physics
PE=m×g×h
E=mg²
S=ut+$\frac{1}{2}$at²
F=ma
I=$\frac{E}{R+r}$
Sx=Vxt
S=($\frac{u+v}{2}$)t
A
B
C



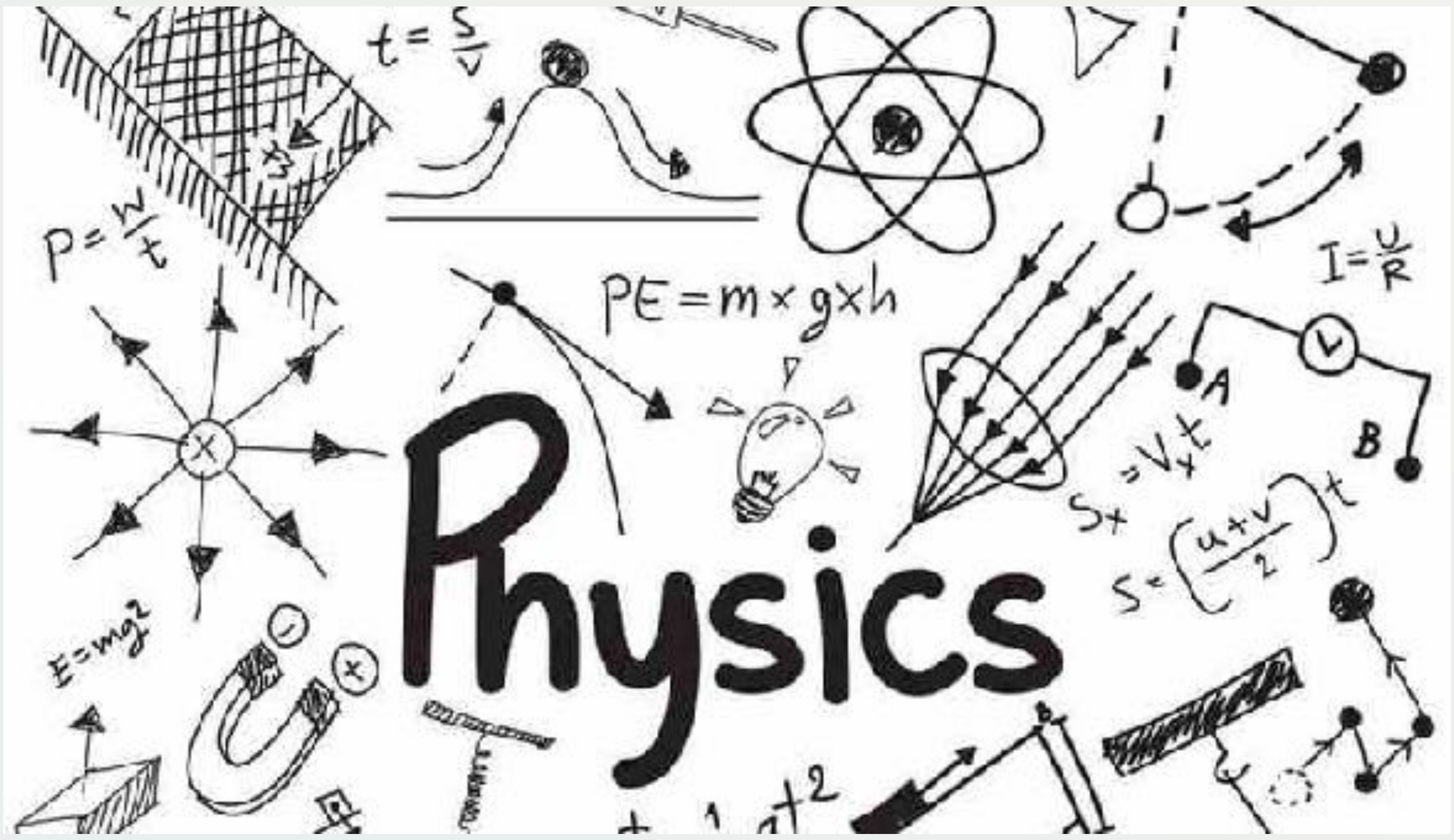
?

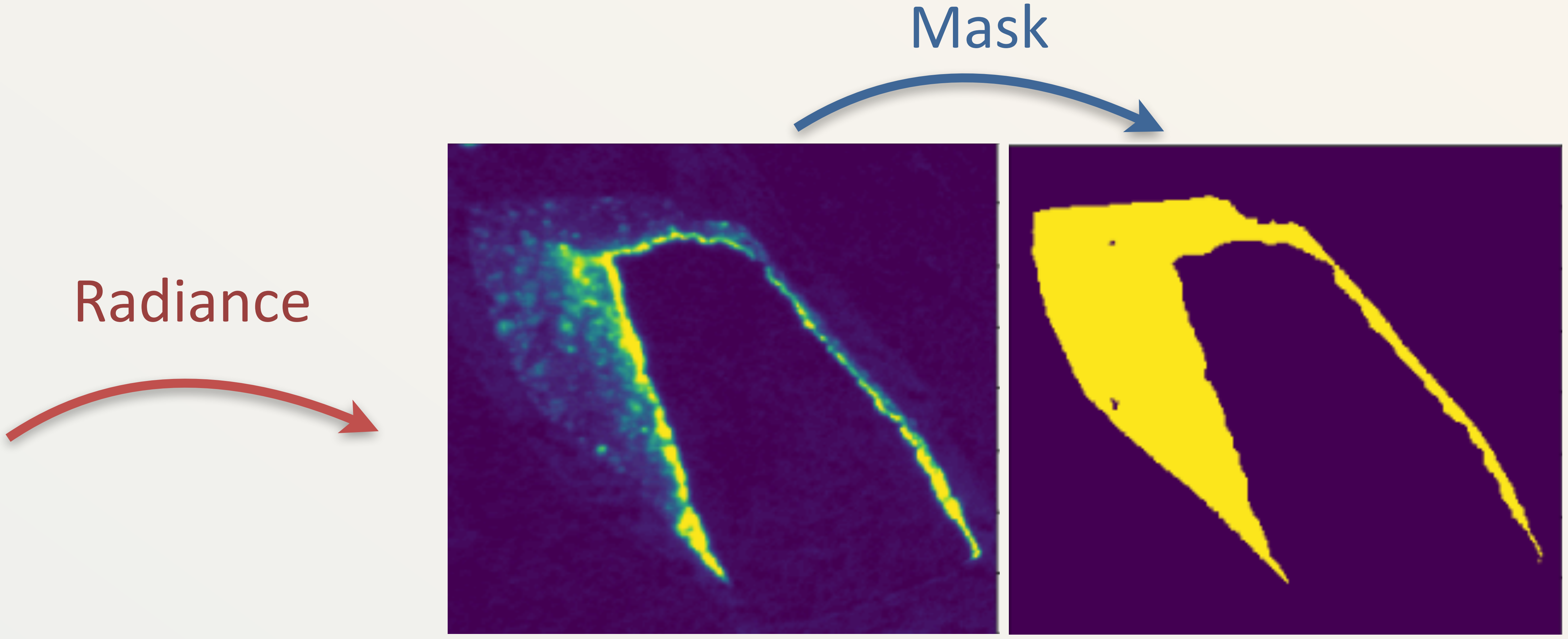




Machine Learning

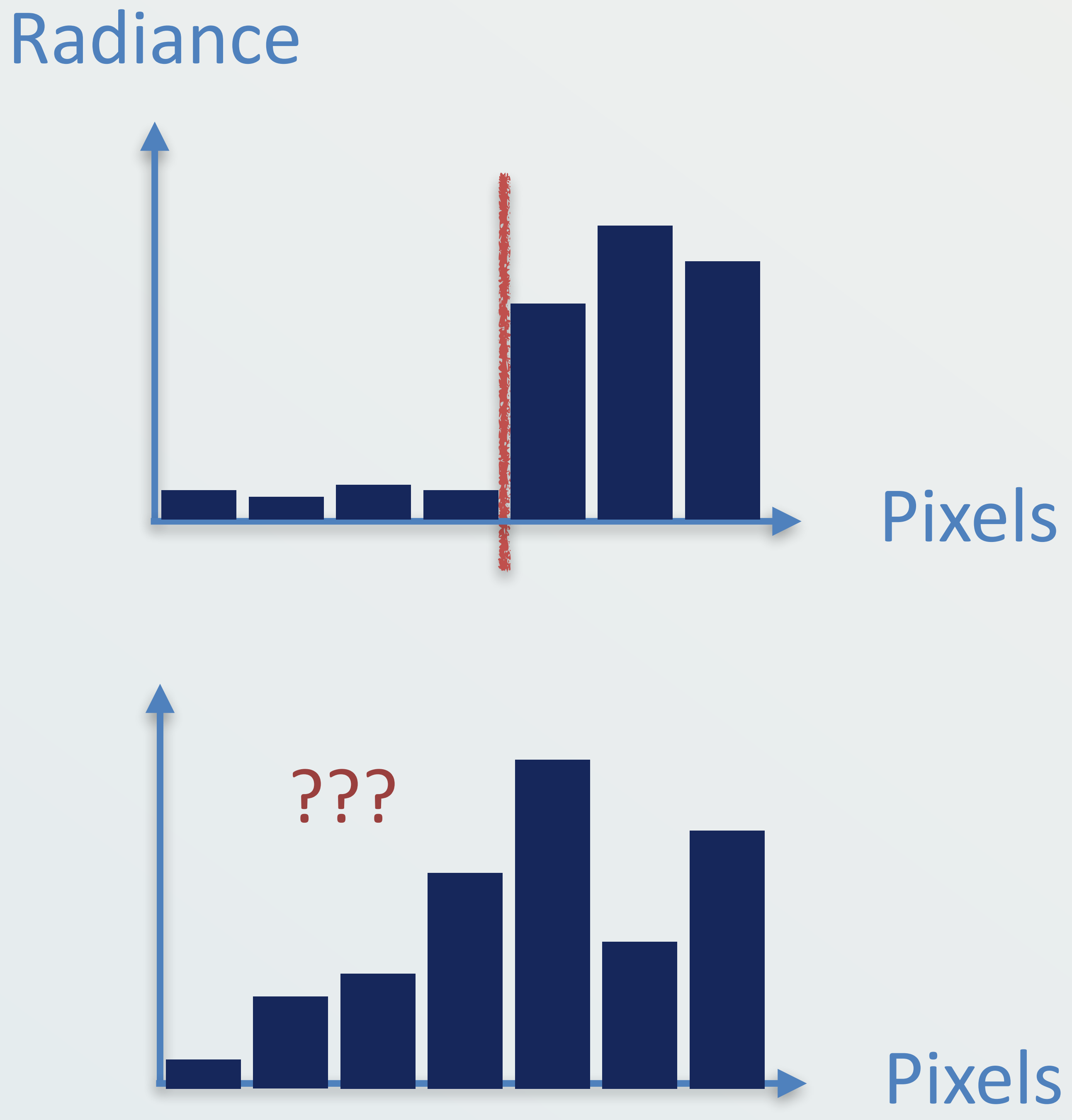
?

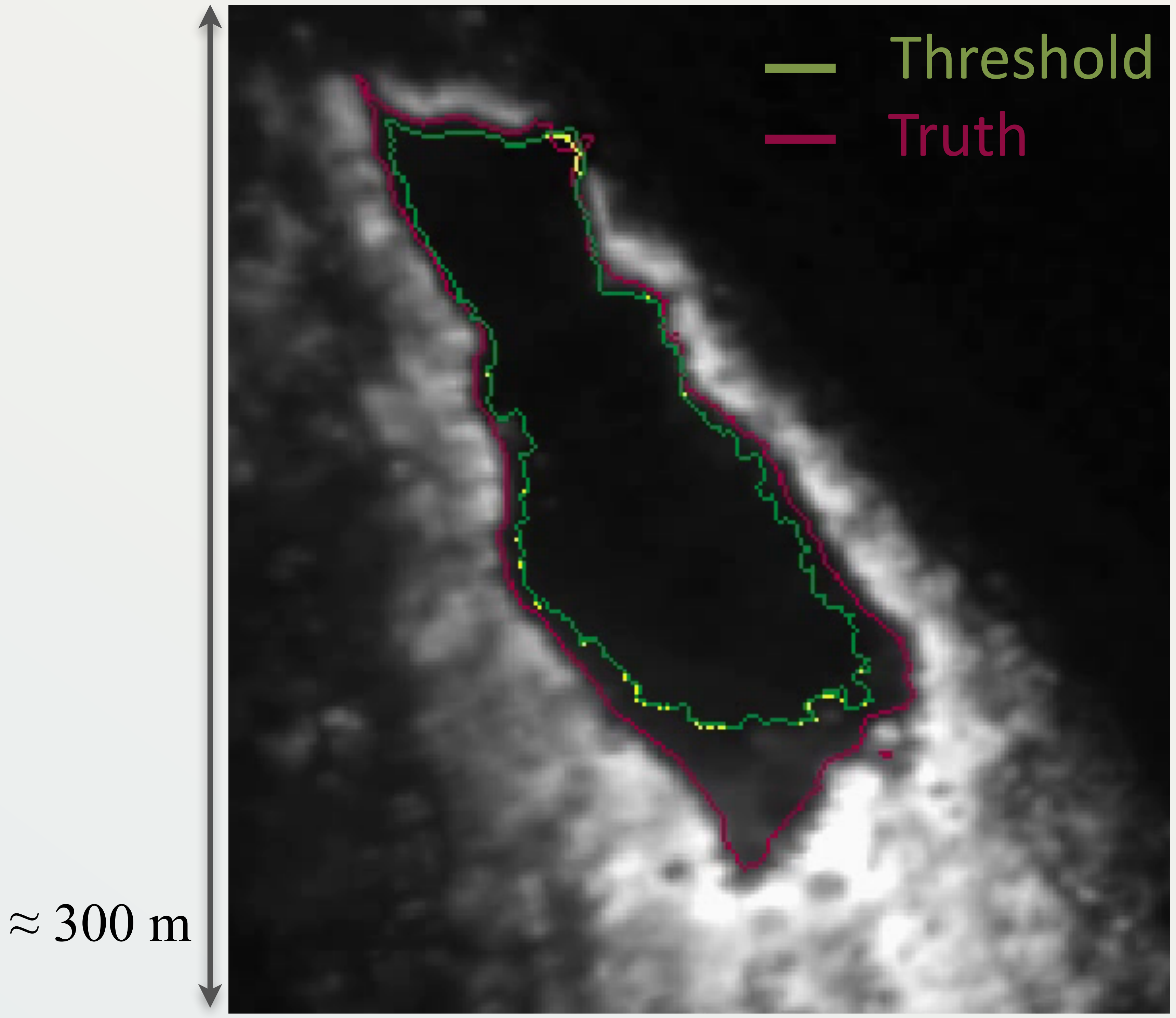




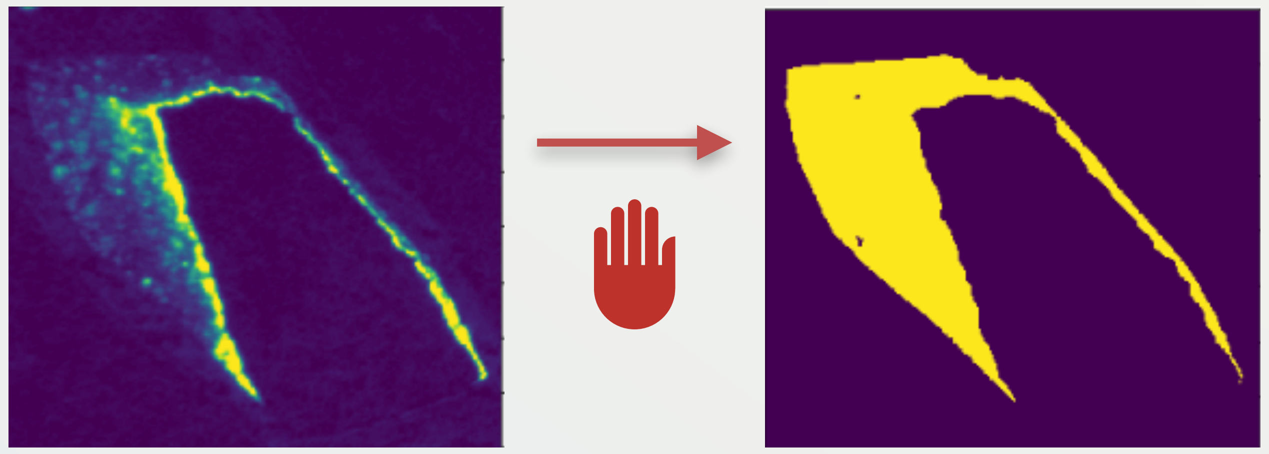
Radiance

Mask

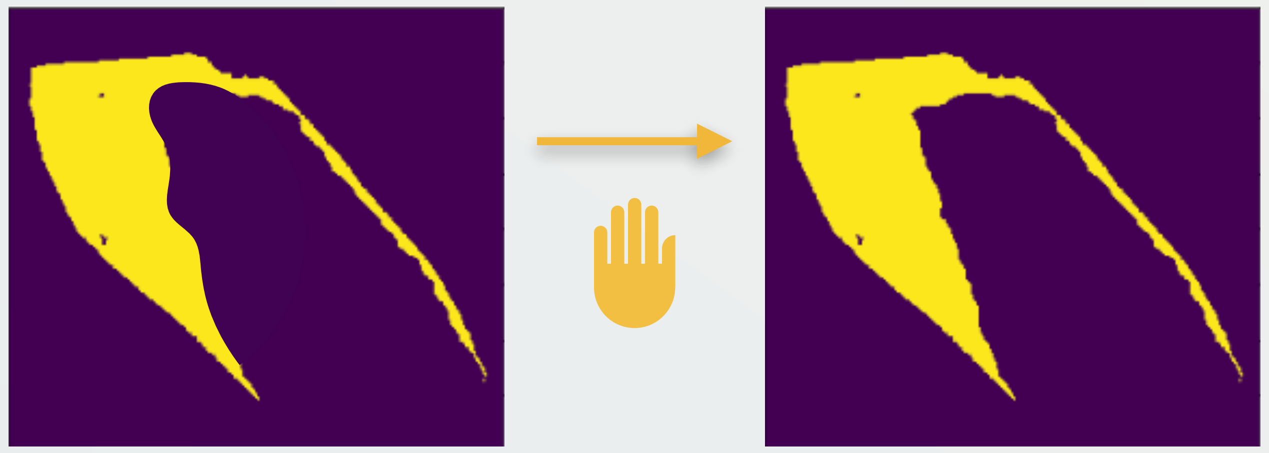




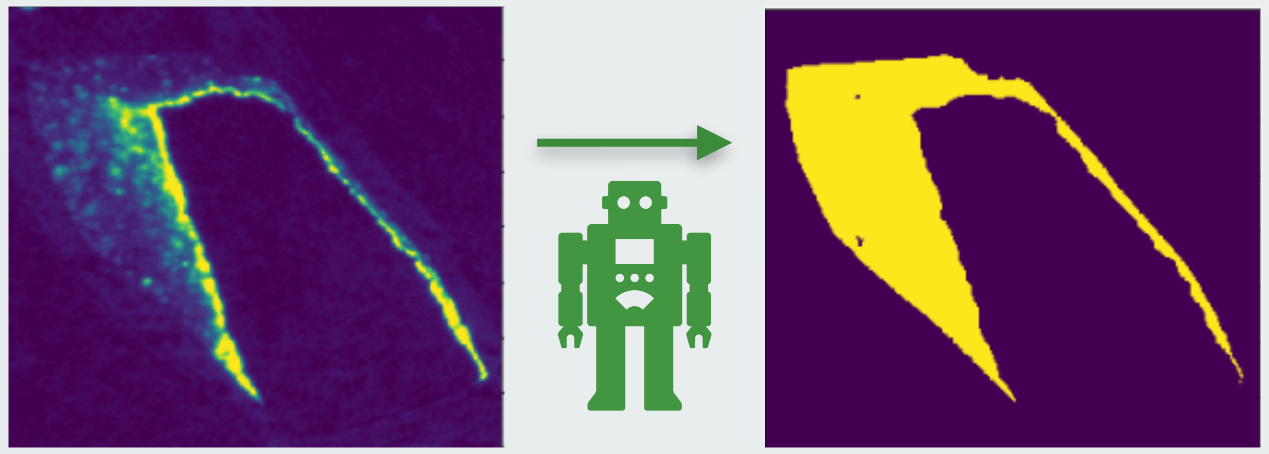
Fieldwork campaign organized by Prof. Martin Wooster (Dept of Geog. University College London) in Kruger National Park, 2014 South Africa.
Ongoing work performed at Cerfacs by R. Paugam, N. Cazard, M. Rochoux



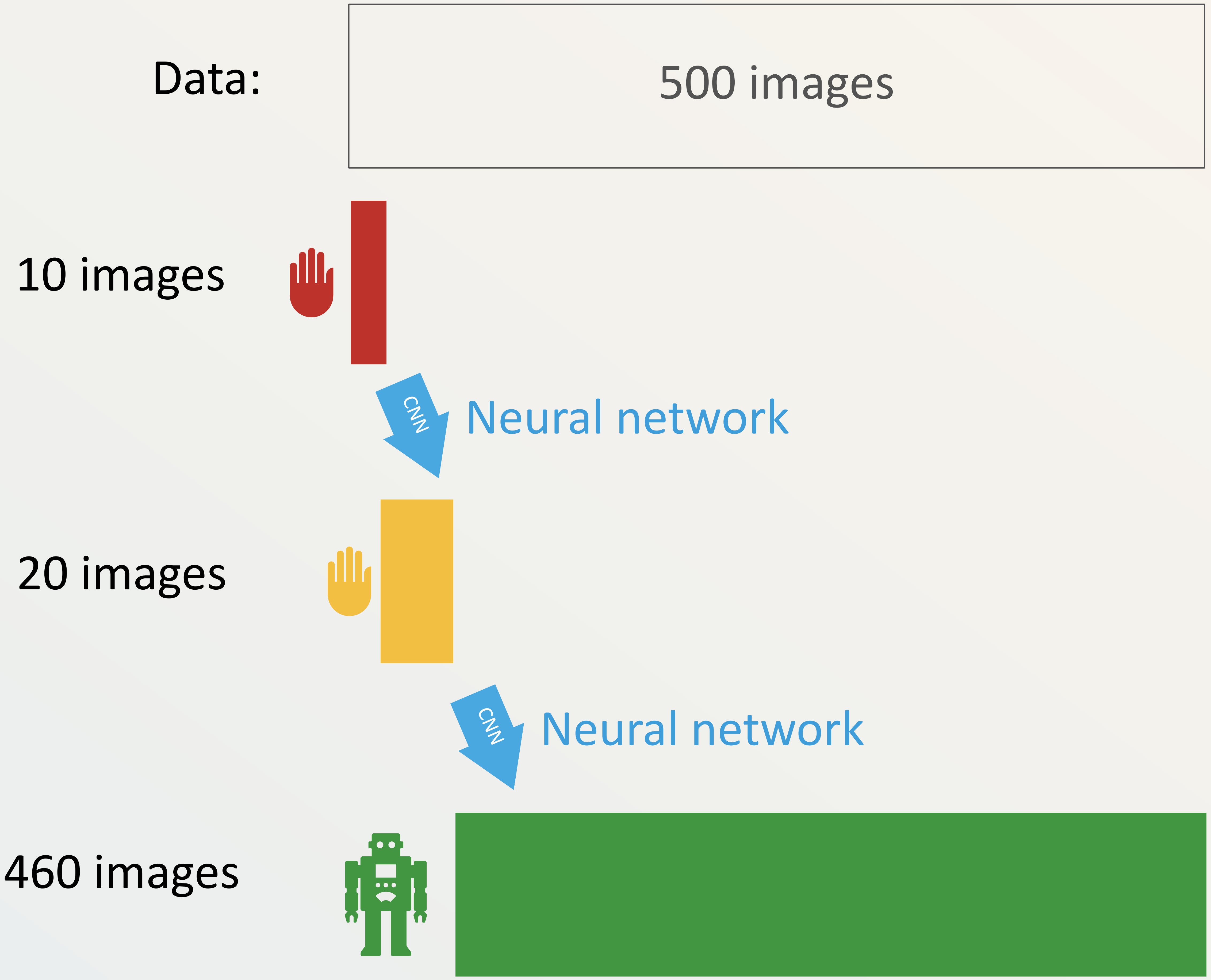
20 minutes



2-3 minutes



0 minutes



Data:

500 images

10 images

Neural network

20 images

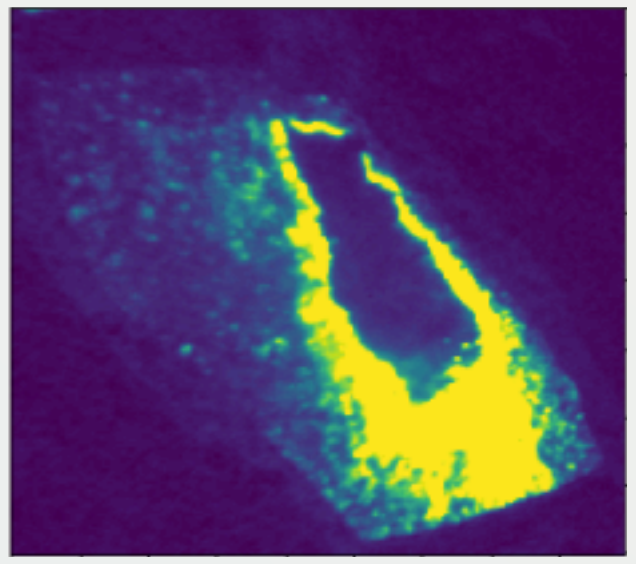
Neural network

460 images

Human time: ≈ 4h instead of 160h !

Transfer learning

Fire 1

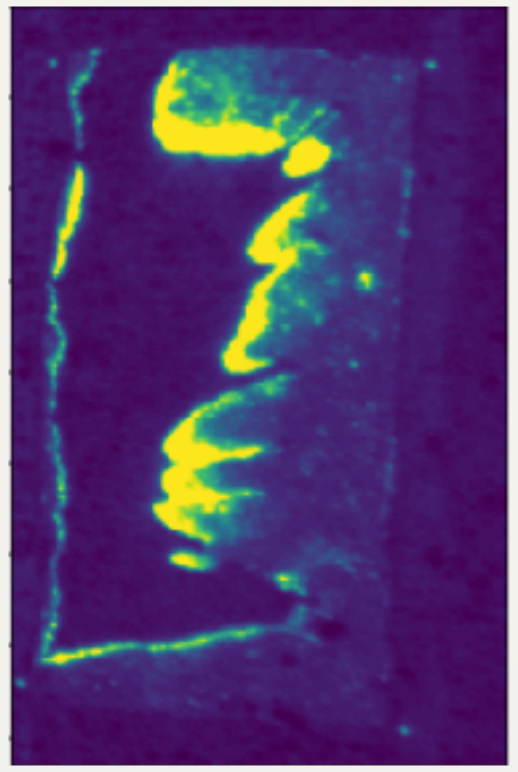


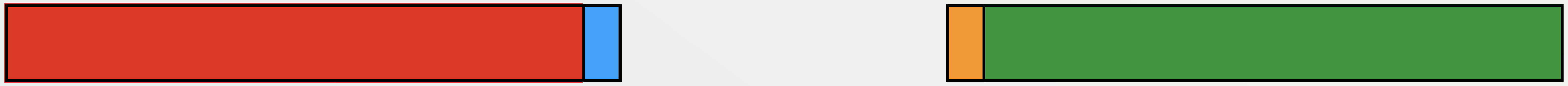
🟥 Train

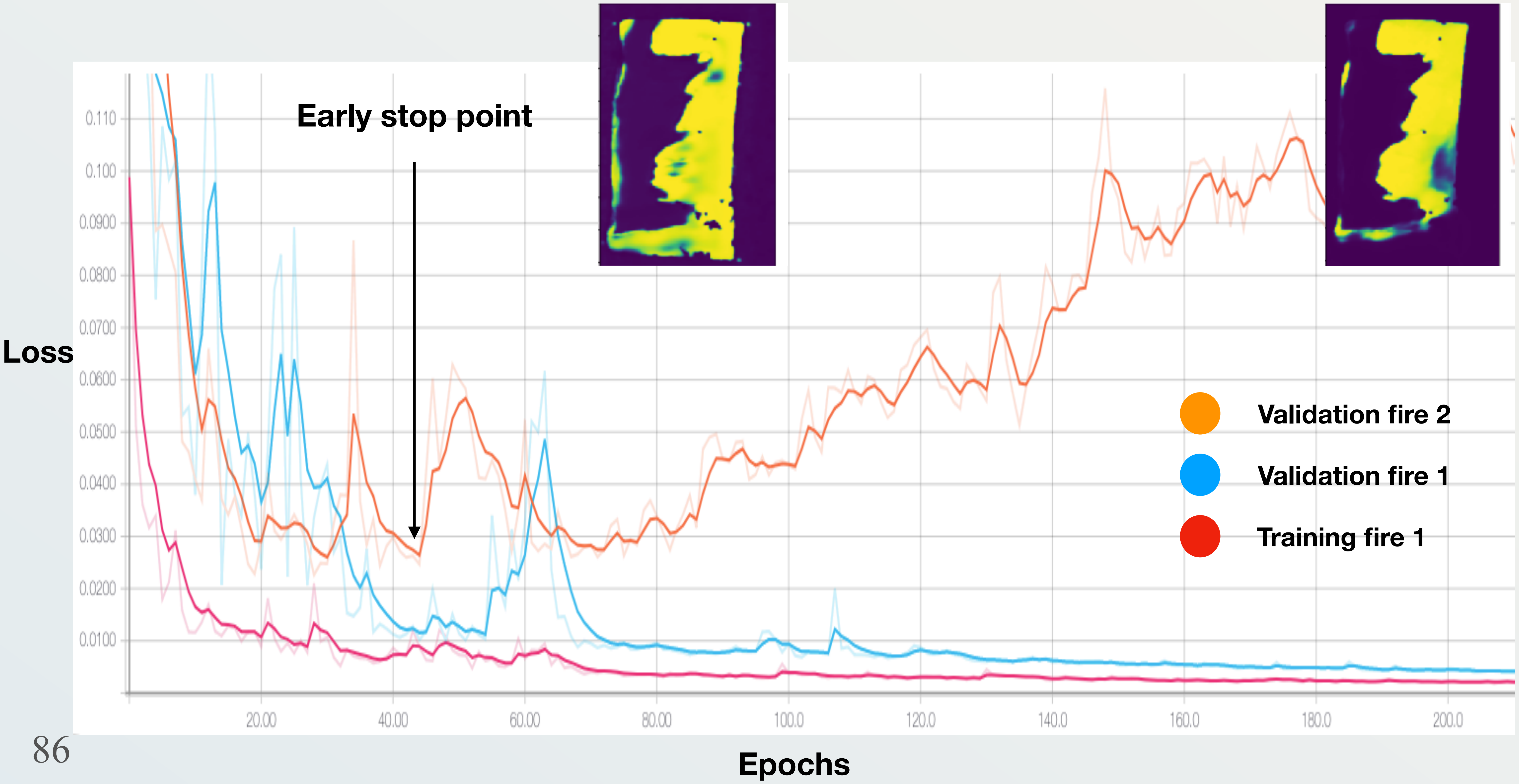
🟦🟧 Validation

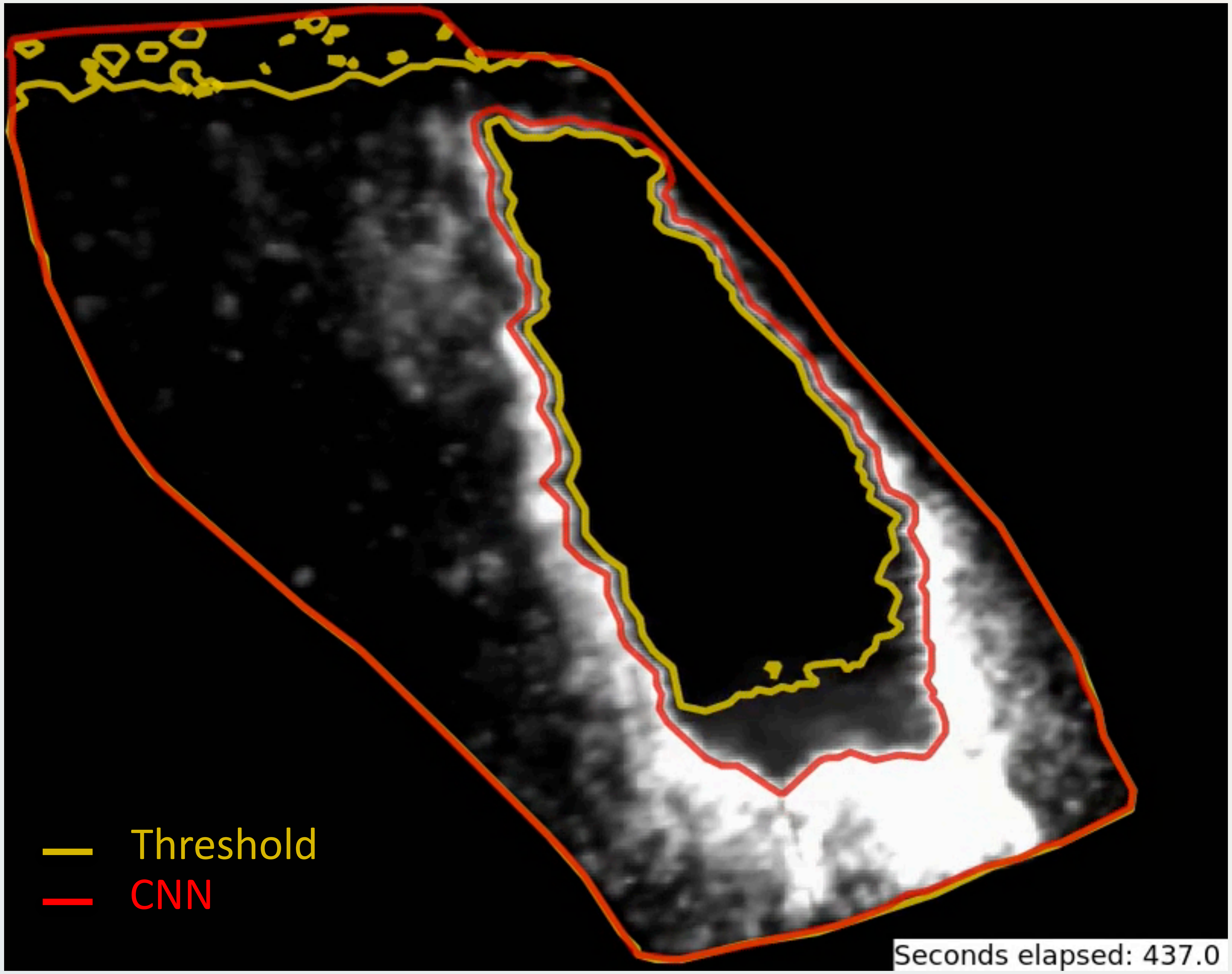
🟩 Test

Fire 2









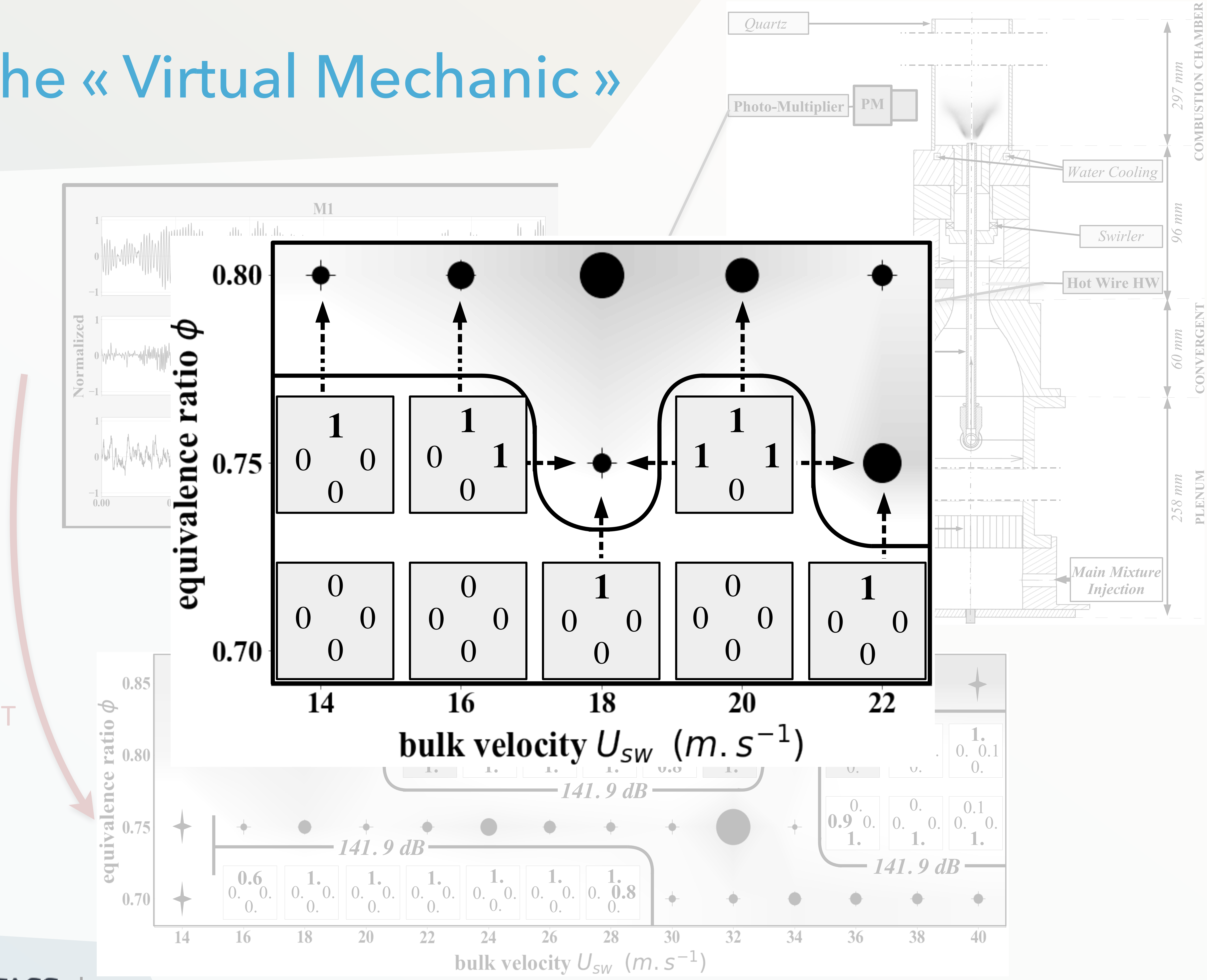
Context awareness is crucial here

Deep Learning enables automatic extraction of features from context

Fieldwork campaign organized by Prof. Martin Wooster (Dept of Geog. University College London) in Kruger National Park, 2014 South Africa.
Work performed at Cerfacs by R. Paugam, N. Cazard, M. Rochoux

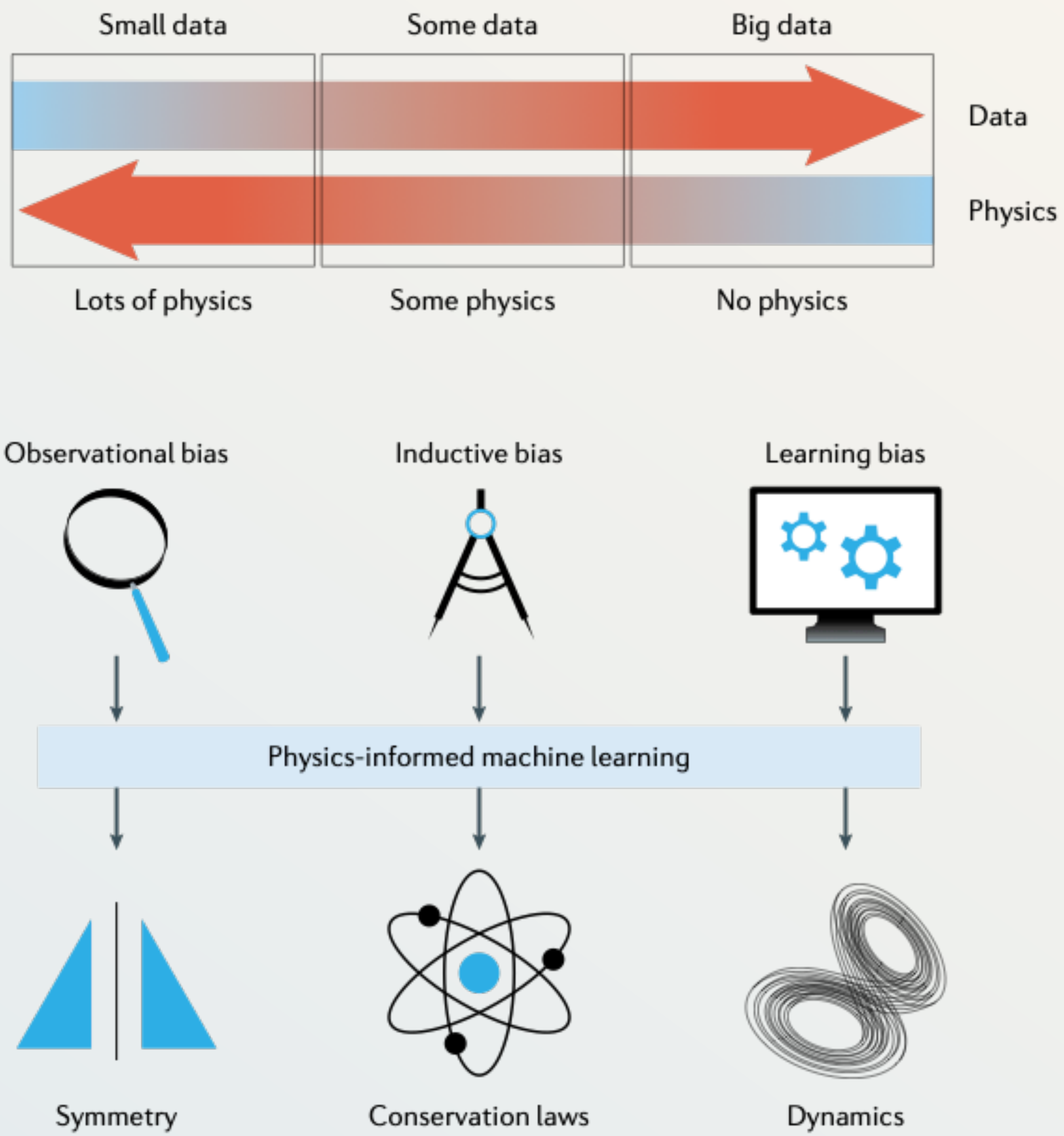
The « Virtual Mechanic »

PREDICT



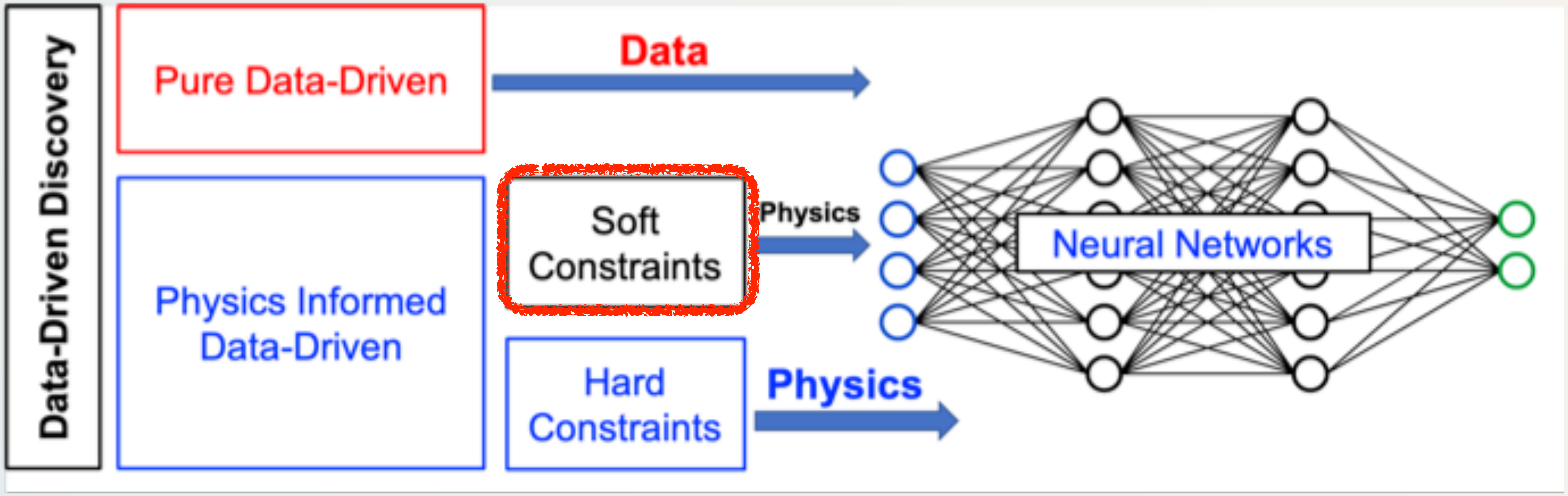
Cellier, A. et al. “Detection of Precursors of Combustion Instability using Convolutional Recurrent Neural Networks” Combustion and Flame (accepted for publication) (2021).

Physics-Informed Learning



Karniadakis, George Em, et al. "Physics-informed machine learning." *Nature Reviews Physics* 3.6 (2021): 422-440.

Constraints



PINNs (soft constraints)

Generic PDE form: $u_t + \mathcal{N}[u;\lambda] = 0$

Example: Burgers' equation $\dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} = \nu\dfrac{\partial^2 u}{\partial x^2}$ $\mathcal{N}[u;\lambda] = \lambda_1 u u_x - \lambda_2 u_x x$

Suppose you have imperfect observations of the system (e.g. noisy measurements).

- Question 1: given fixed λ, can we estimate u(t, x) ? (Data driven PDE solving)
- Question 2: what λ best describes the data ? (System identification)

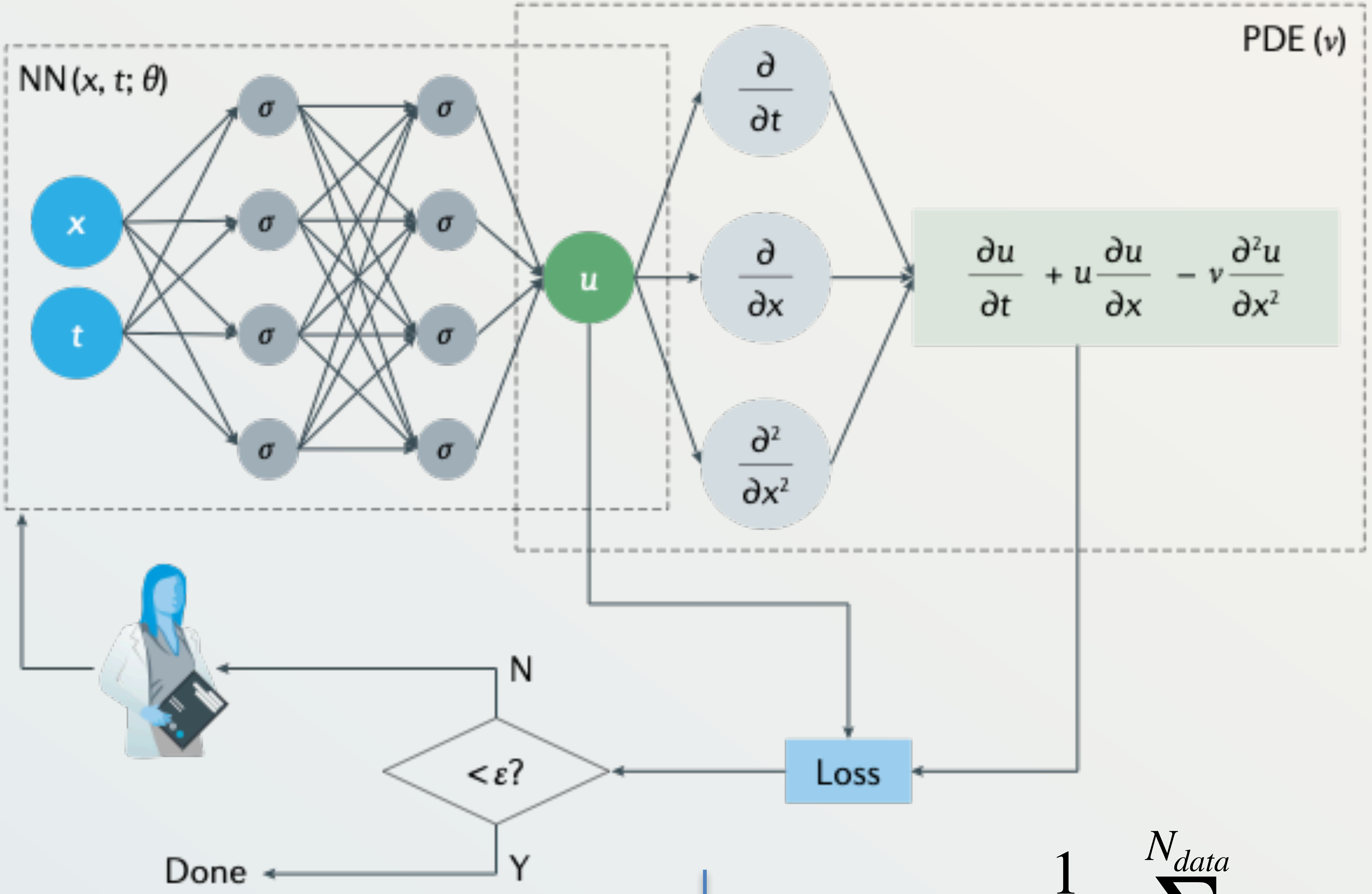
Focus on Q1: $f := u_t + \mathcal{N}[u]$

$u(t,x)$ -> Deep Neural Network

$f(t,x)$ -> Physics-Informed Neural Network

Raissi, Maziar, Paris Perdikaris, and George Em Karniadakis. "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations." arXiv preprint arXiv:1711.10561 (2017).

PINNs

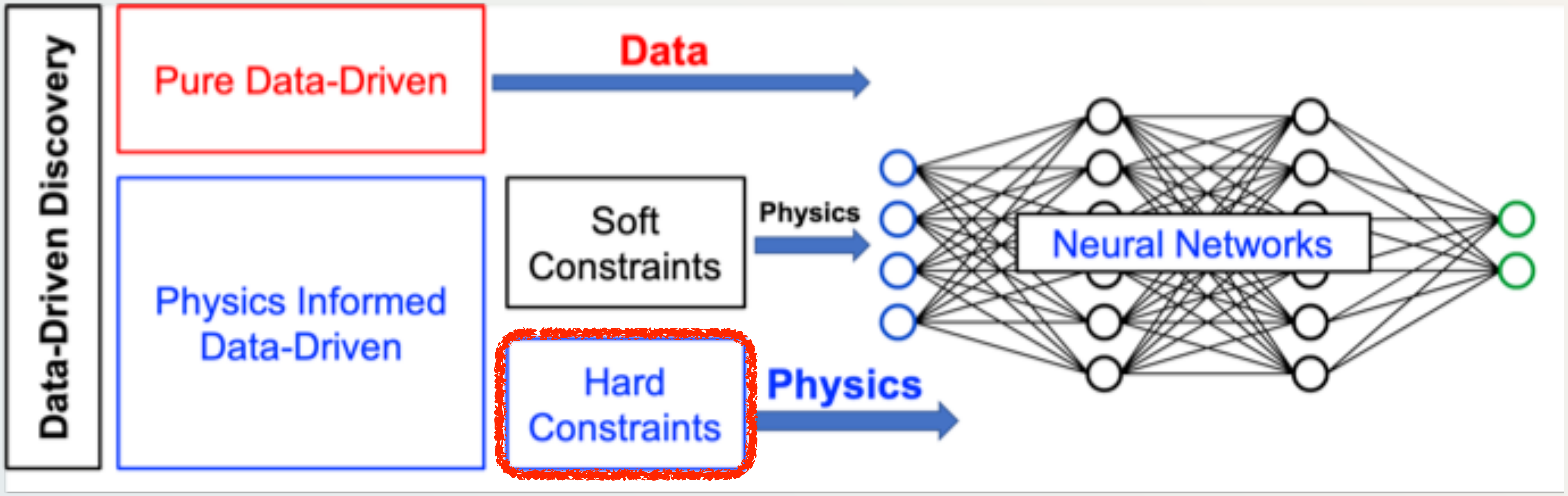


$$L = w_{data} L_{data} + w_{PDE} L_{PDE}$$

$$L_{data} = \frac{1}{N_{data}} \sum_{i=1}^{N_{data}} (u(t_i, x_i) - u_i)^2$$

$$L_{PDE} = \frac{1}{N_{PDE}} \sum_{j=1}^{N_{PDE}} \left(\frac{\partial u}{\partial t} + \mathcal{N}(u) \right)^2 \Bigg|_{t_j, x_j}$$

Constraints



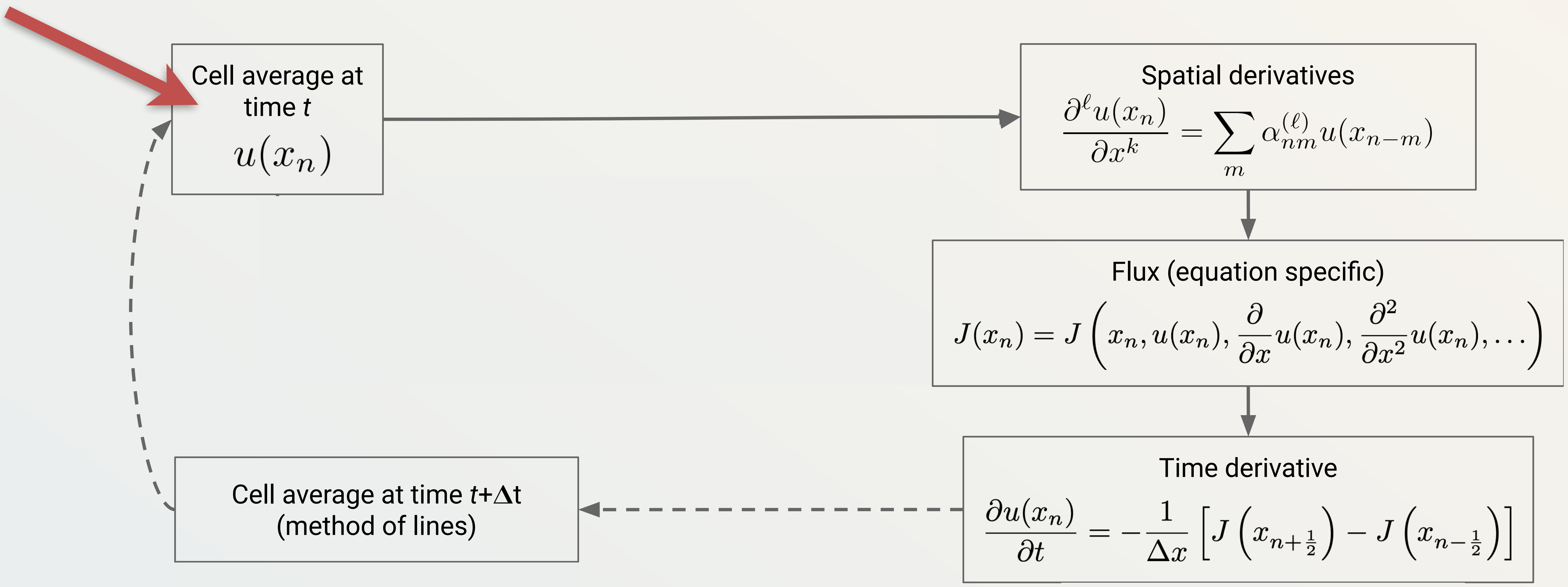
Data-Driven Discovery

Pure Data-Driven

Physics Informed Data-Driven

Soft Constraints

Hard Constraints



Cell average at time *t*

$u(x_n)$

Spatial derivatives

$$\frac{\partial^\ell u(x_n)}{\partial x^k} = \sum_m \alpha_{nm}^{(\ell)} u(x_{n-m})$$

Flux (equation specific)

$$J(x_n) = J\left(x_n, u(x_n), \frac{\partial}{\partial x}u(x_n), \frac{\partial^2}{\partial x^2}u(x_n), \ldots\right)$$

Time derivative

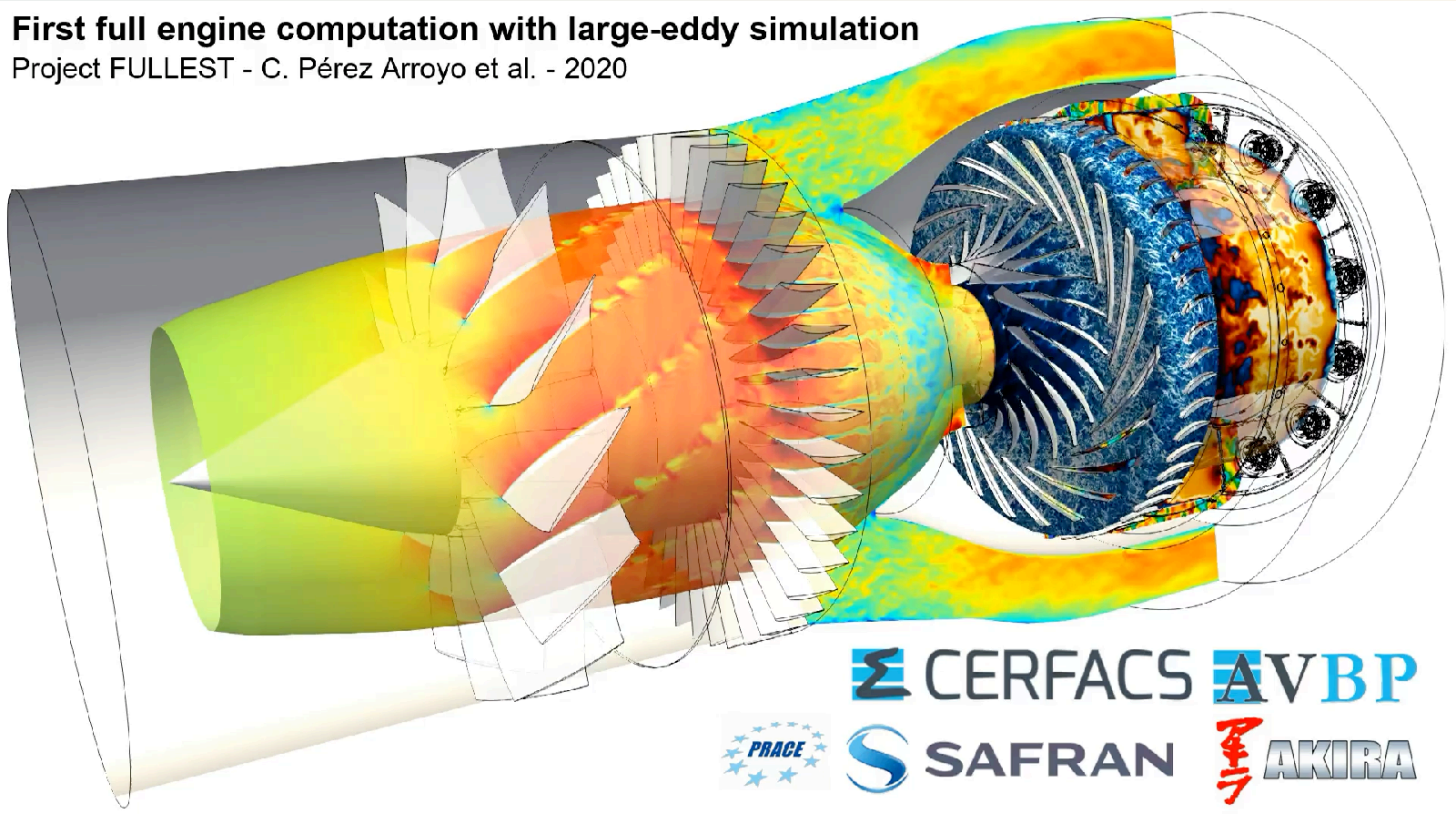
$$\frac{\partial u(x_n)}{\partial t} = -\frac{1}{\Delta x}\left[J\left(x_{n+\frac{1}{2}}\right) - J\left(x_{n-\frac{1}{2}}\right)\right]$$

Cell average at time $t+\Delta$t
(method of lines)

To train *through* the solver,
it must be *differentiable.*

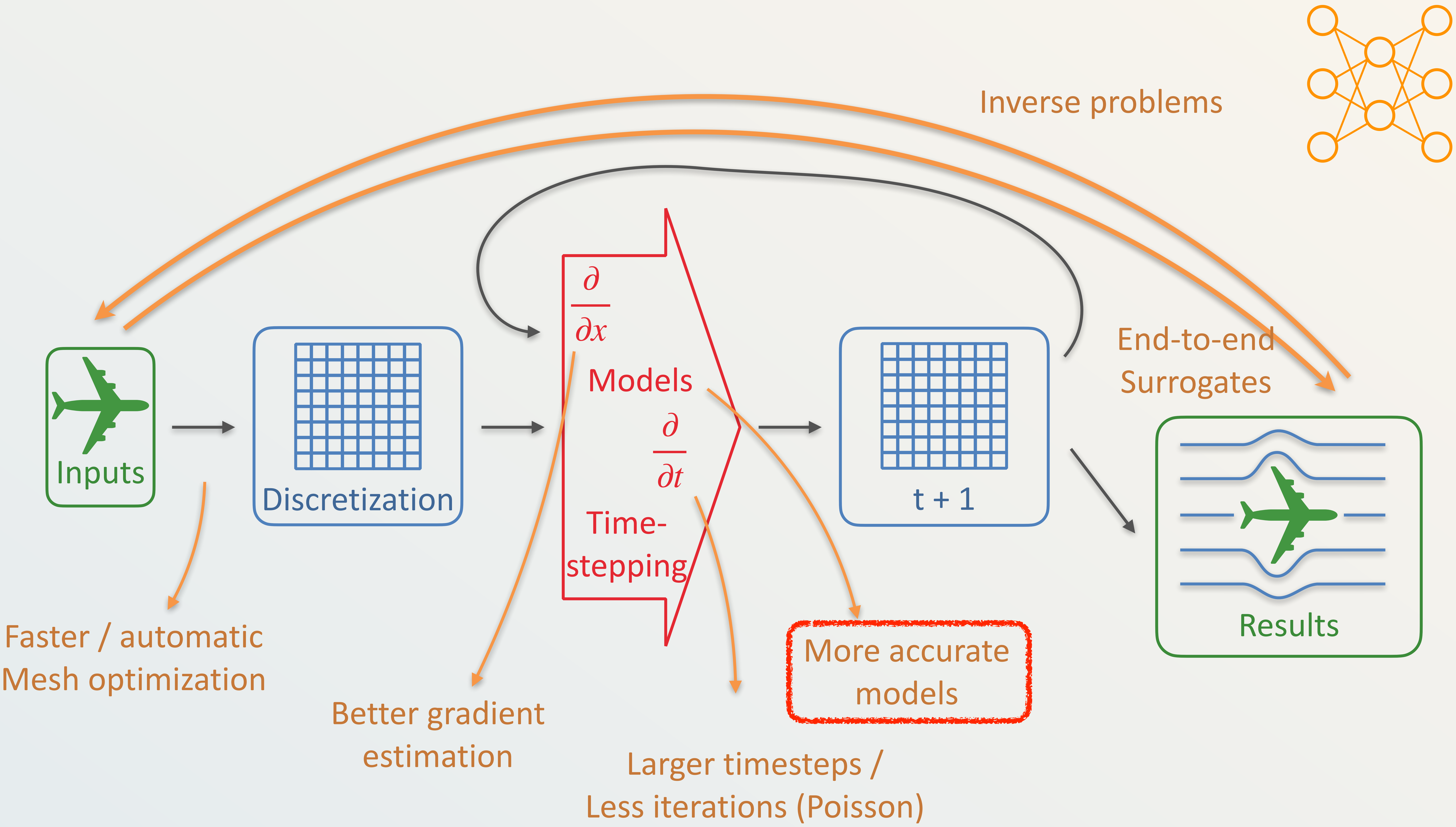
Bar-Sinai, Y., Hoyer, S., Hickey, J., & Brenner, M. P. (2019). Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, *116*(31), 15344-15349.

What about CFD?



First full engine computation with large-eddy simulation

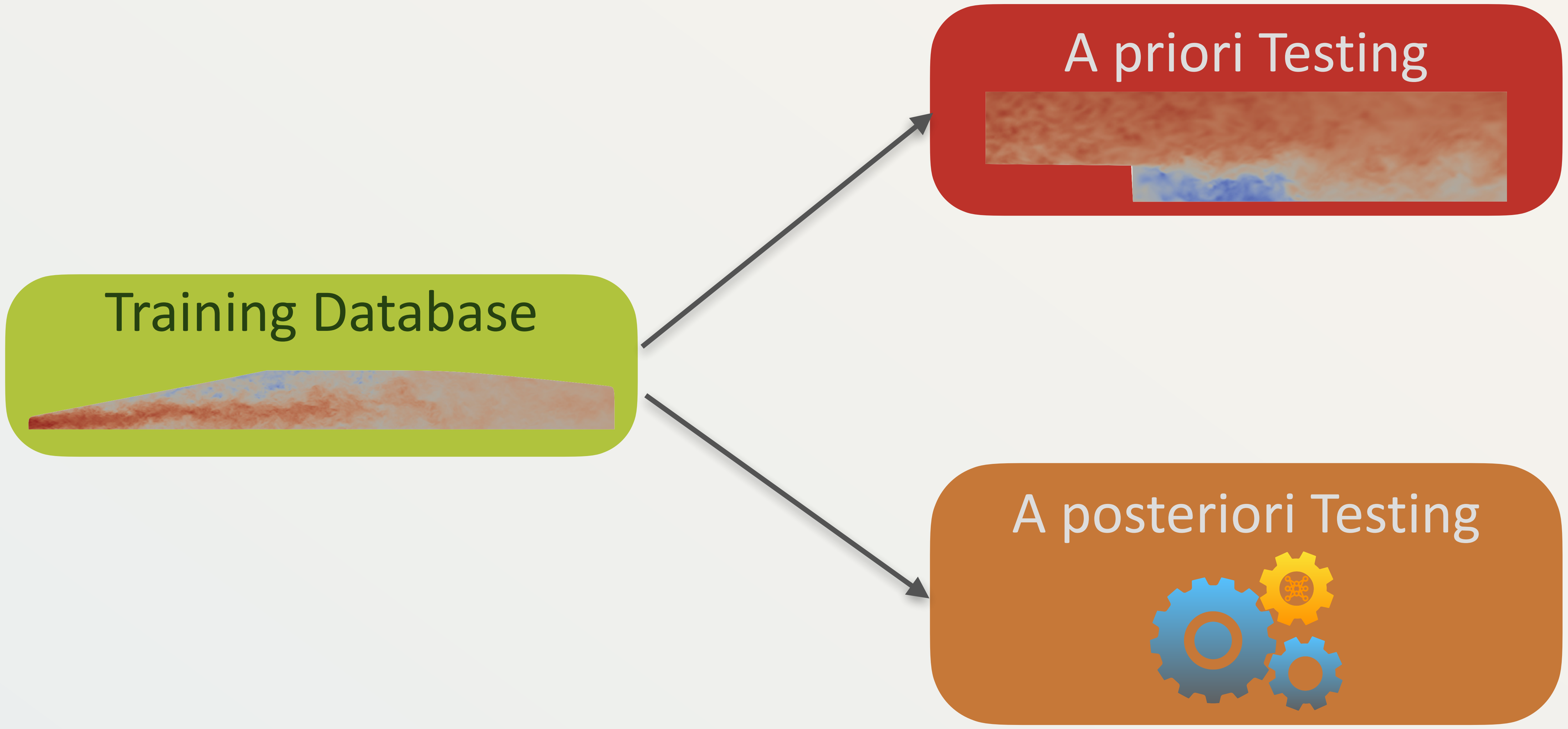
Project FULLEST - C. Pérez Arroyo et al. - 2020



Where should we look?
Unclear: literature still hesitant

Much research on hybrid techniques

Post-mortem



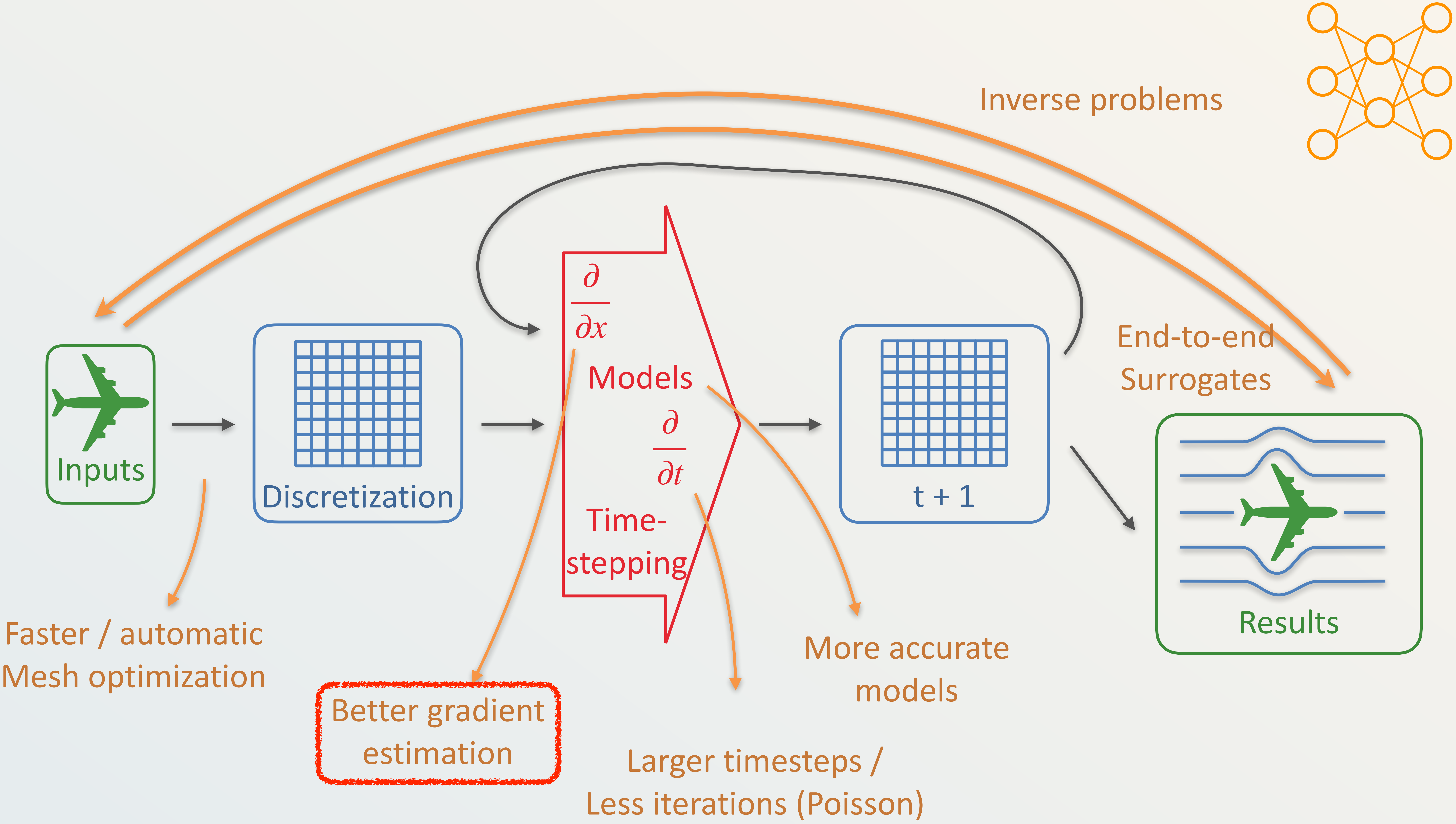
In the loop

Reinforcement Learning

(c.f. talk this afternoon)

=> Search for behaviour that maximises your true objective
(here: in code performance)

Train for your objective directly *through* the solver


Inverse problems

Inputs

Discretization

$\frac{\partial}{\partial x}$

Models

$\frac{\partial}{\partial t}$

Time-stepping

t + 1

End-to-end Surrogates

Results

Faster / automatic Mesh optimization

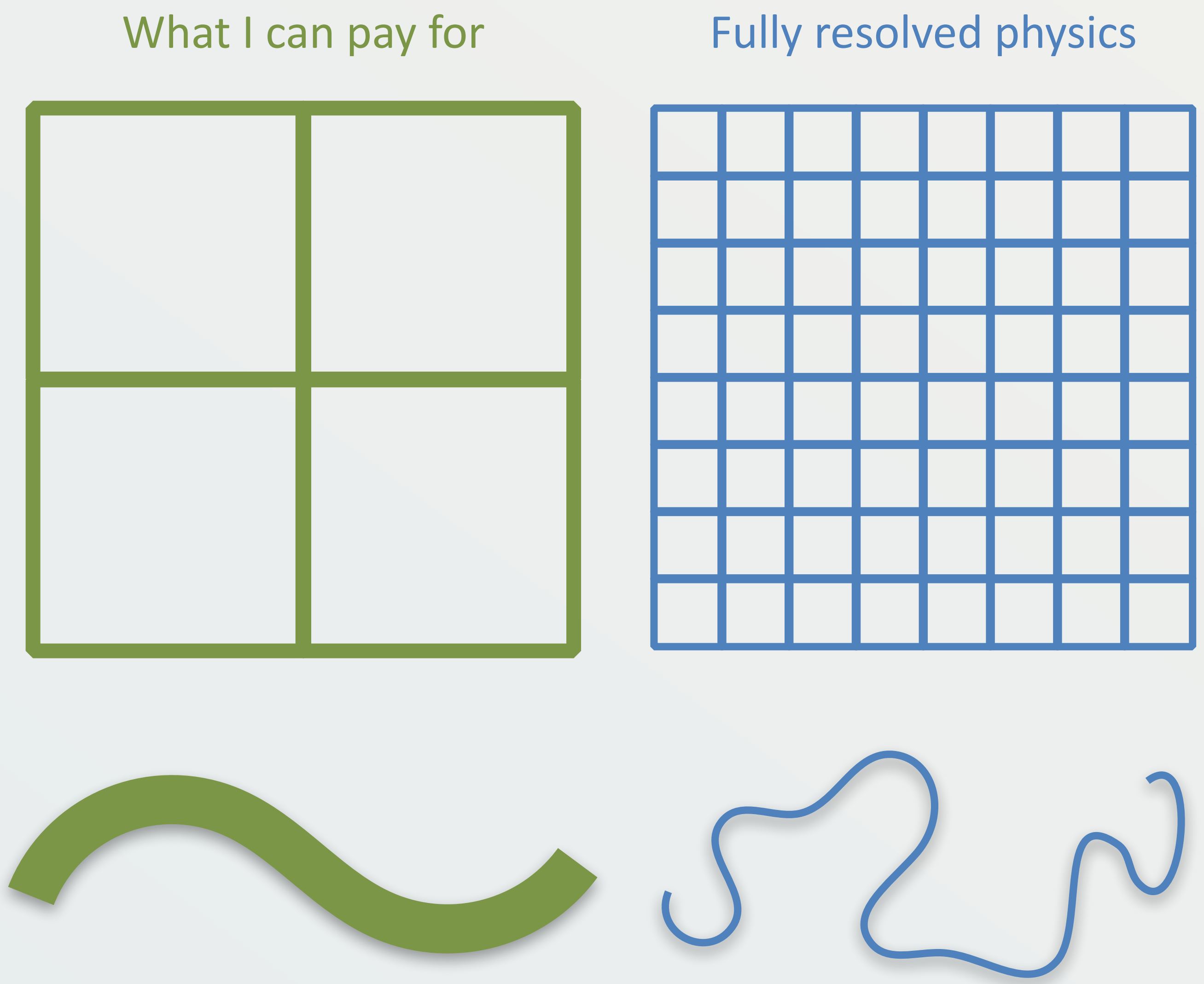
Better gradient estimation

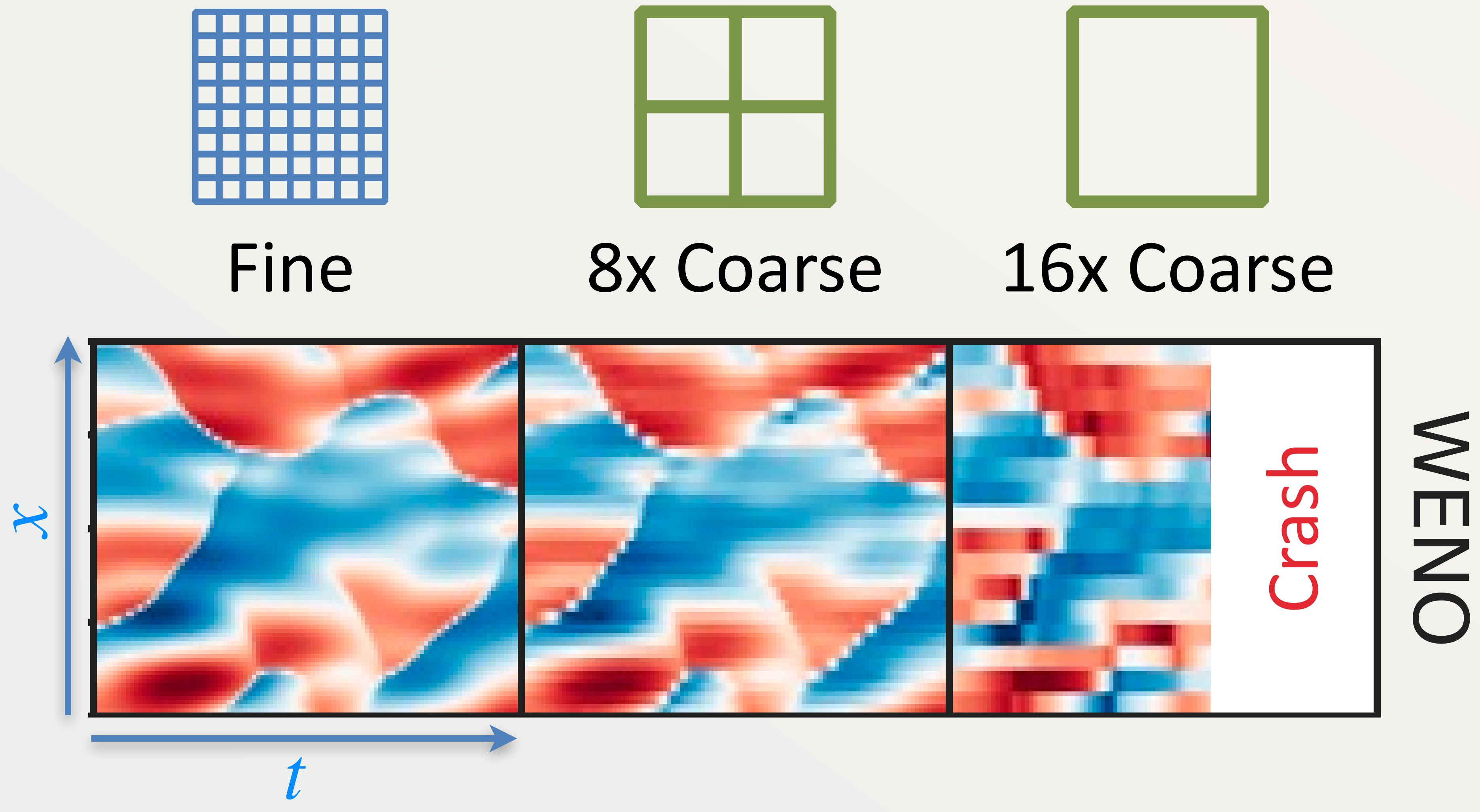
Larger timesteps / Less iterations (Poisson)

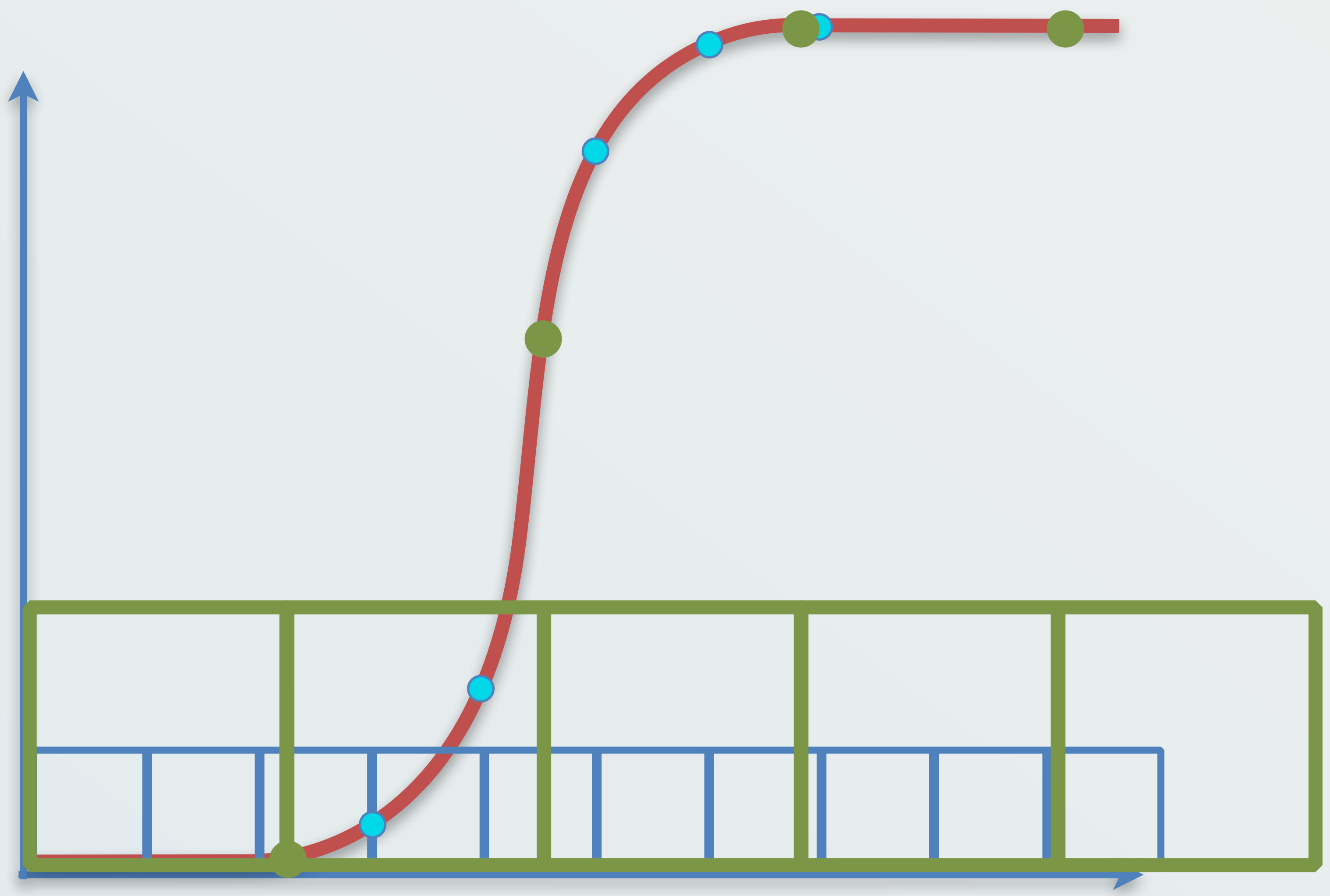
More accurate models

An example of training through the solver

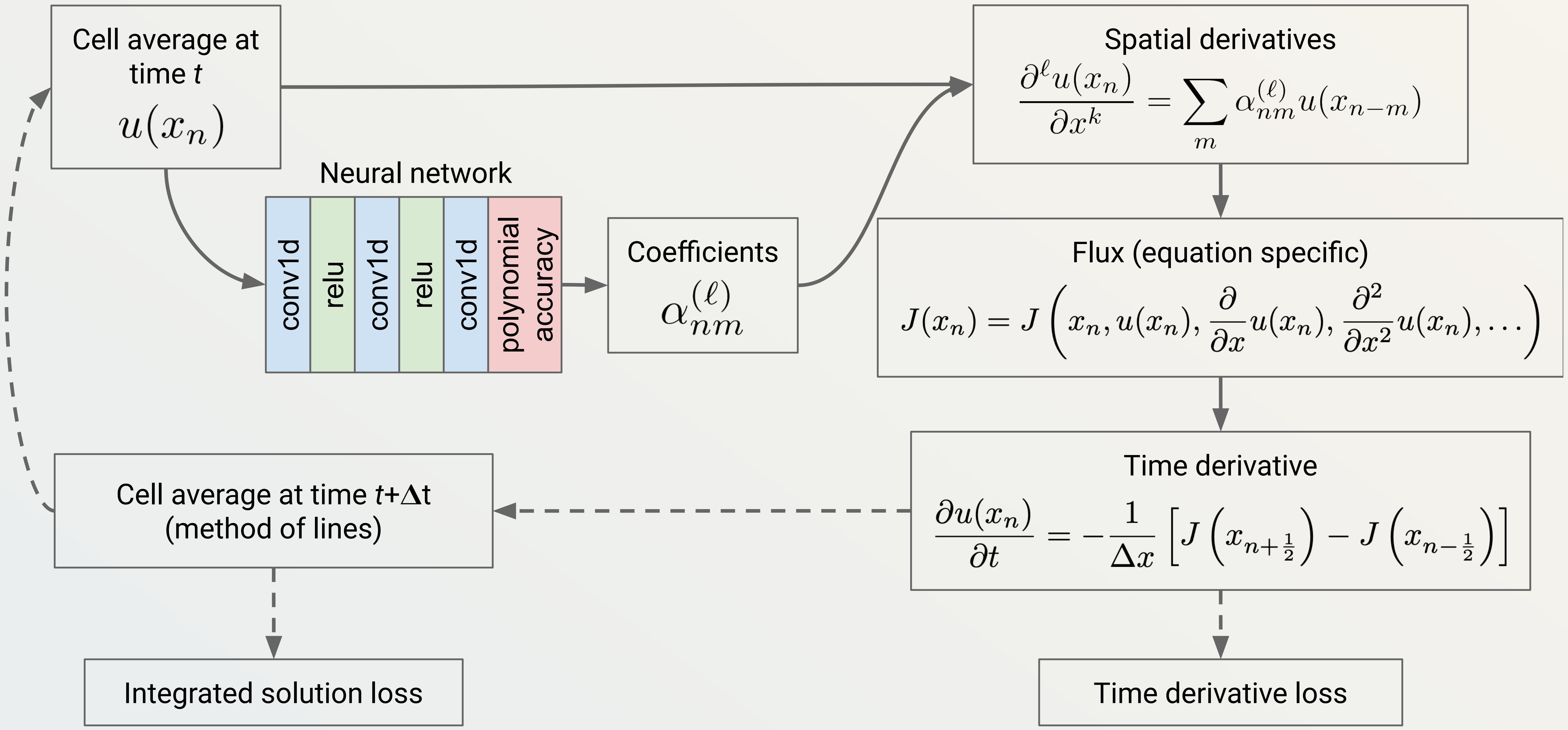
Solving fine structures





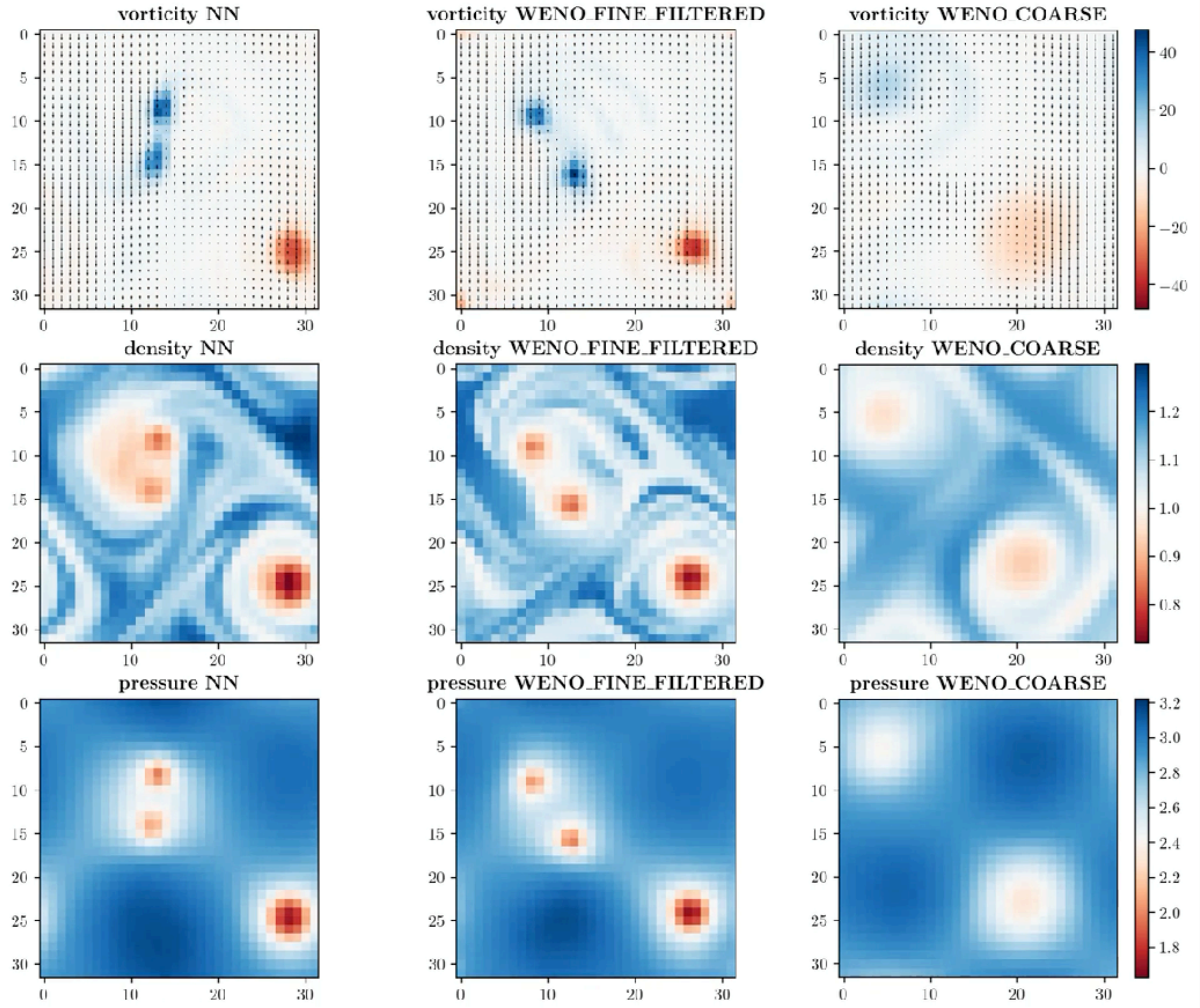


Bar-Sinai, Y., Hoyer, S., Hickey, J., & Brenner, M. P. (2019). Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, *116*(31), 15344-15349.


Cell average at time t
$u(x_n)$
Neural network
conv1d
relu
conv1d
relu
conv1d
polynomial accuracy
Coefficients
$\alpha_{nm}^{(\ell)}$
Spatial derivatives
$\frac{\partial^\ell u(x_n)}{\partial x^k} = \sum_m \alpha_{nm}^{(\ell)} u(x_{n-m})$
Flux (equation specific)
$J(x_n) = J\left(x_n, u(x_n), \frac{\partial}{\partial x}u(x_n), \frac{\partial^2}{\partial x^2}u(x_n), \ldots\right)$
Time derivative
$\frac{\partial u(x_n)}{\partial t} = -\frac{1}{\Delta x}\left[J\left(x_{n+\frac{1}{2}}\right) - J\left(x_{n-\frac{1}{2}}\right)\right]$
Cell average at time $t+\Delta$t
(method of lines)
Integrated solution loss
Time derivative loss

To train *through* the solver,
it must be *differentiable.*

Bar-Sinai, Y., Hoyer, S., Hickey, J., & Brenner, M. P. (2019). Learning data-driven discretizations for partial differential equations. *Proceedings of the National Academy of Sciences*, *116*(31), 15344-15349.



vorticity NN

vorticity WENO_FINE_FILTERED

vorticity WENO_COARSE

density NN

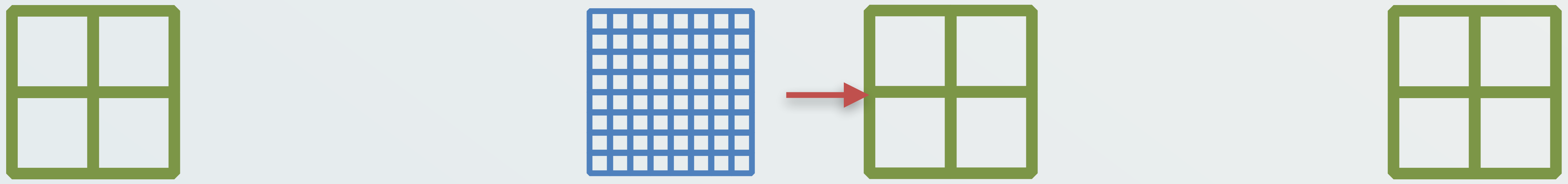
density WENO_FINE_FILTERED

density WENO_COARSE

pressure NN

pressure WENO_FINE_FILTERED

pressure WENO_COARSE



CFD + NN

CFD

Filtered

CFD

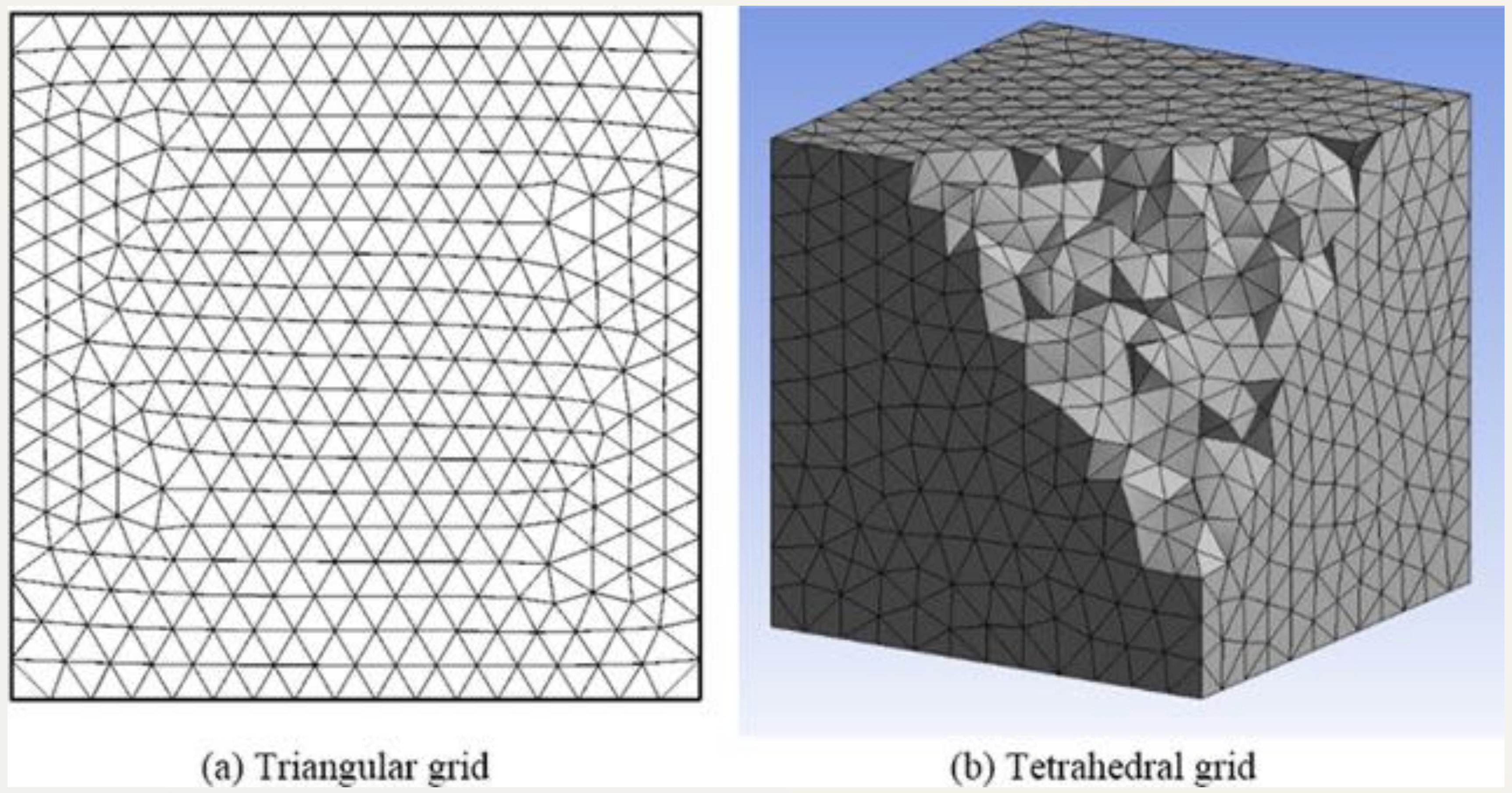
CFD solver rewritten in Julia (fully differentiable framework)

Ongoing PhD of Luciano Drozda, Cerfacs

Unstructured numerical scheme:

Two-step Taylor-Galerkin type C

TTGC

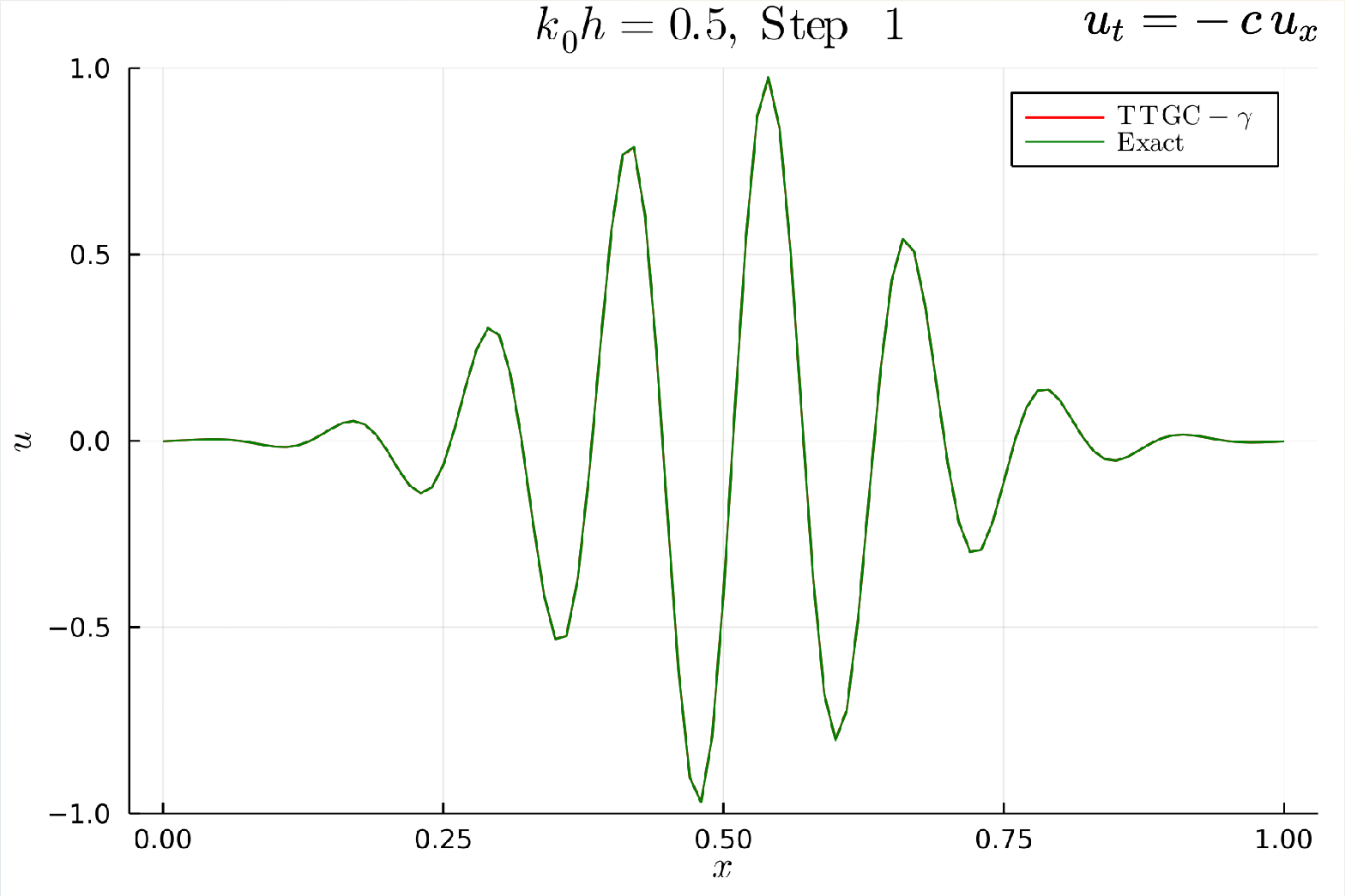

(a) Triangular grid

(b) Tetrahedral grid

$$\tilde{u}^n = u^n + (0.5 - \gamma)\,\Delta t\, u_t^n + (1/6)\,\Delta t^2\, u_{tt}^n$$

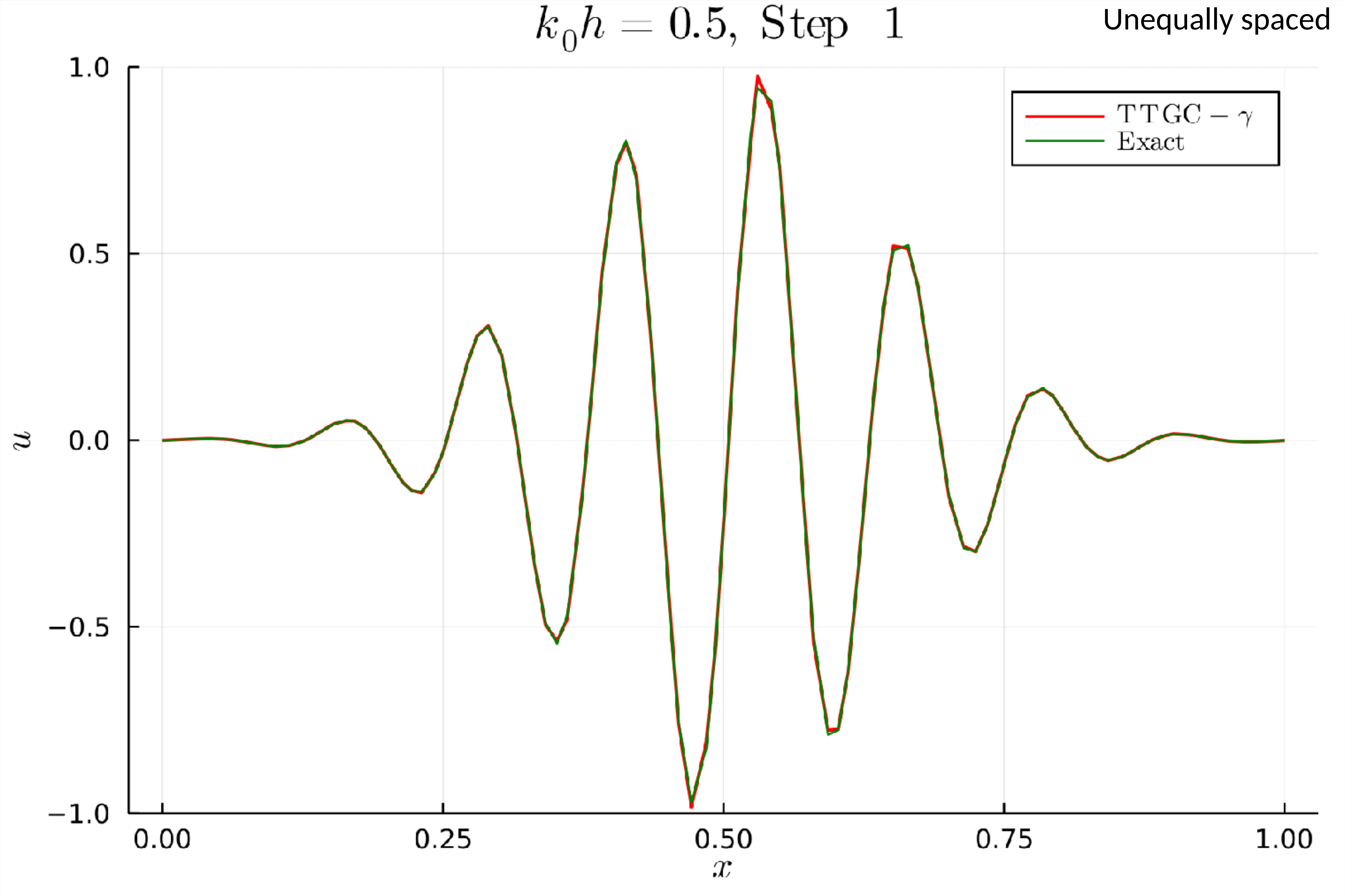
$$u^{n+1} = u^n + \Delta t\, \tilde{u}_t^n + \gamma\, \Delta t^2\, u_{tt}^n$$

$$\boxed{\gamma = 0.01}$$


$k_0 h = 0.5,$ Step 1
$u_t = -c\,u_x$
TTGC − γ
Exact
1.0
0.5
0.0
−0.5
−1.0
u
0.00
0.25
0.50
0.75
1.00
x

$$u_t = -c\,u_x$$

Unequally spaced



Global optimal is *not* a local optimal

$$\boxed{\gamma = 0.01}$$

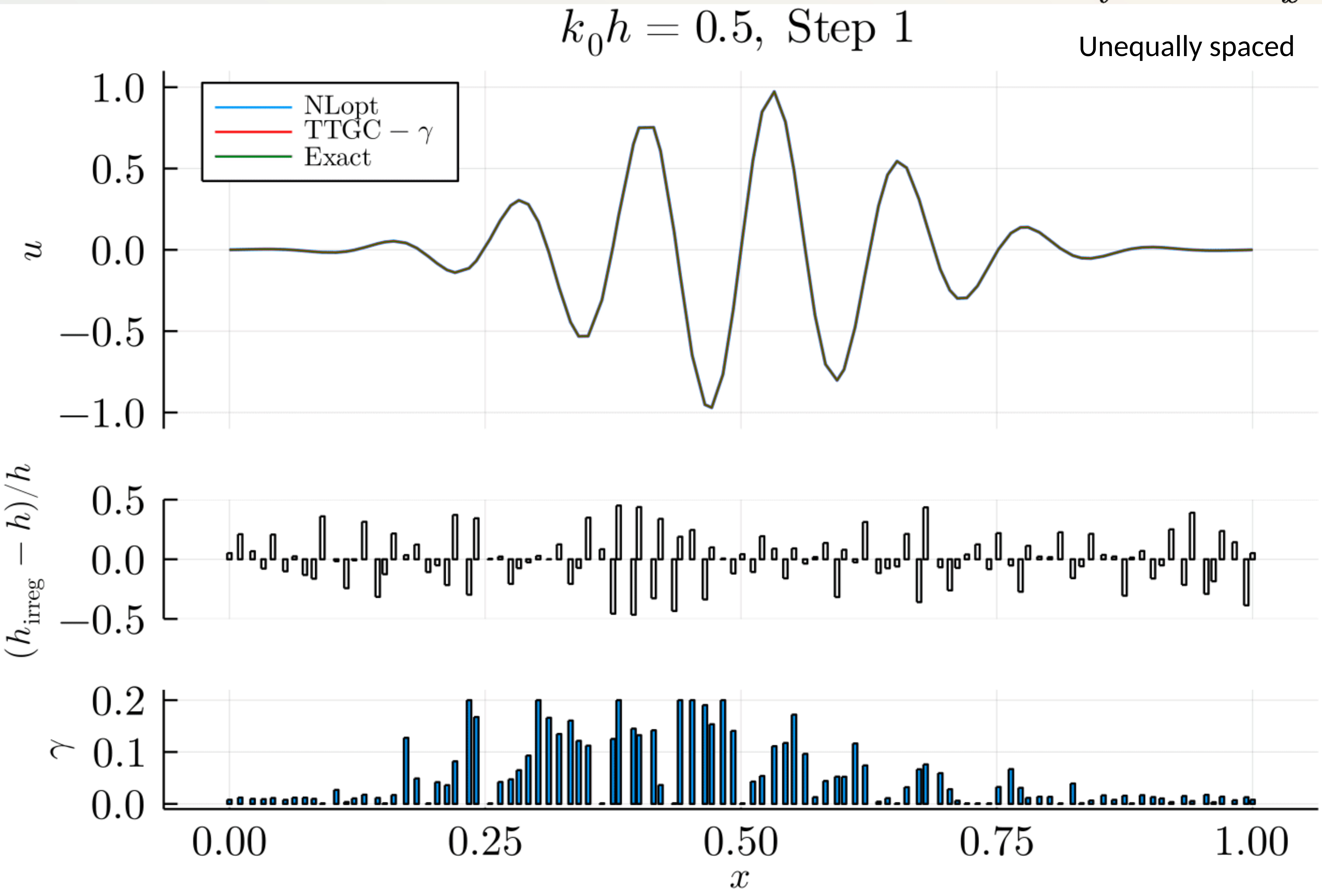
Allow γ to change **locally** in the mesh

Differentiate TTGC solver

Supply gradients to **NLopt** (optimizer)

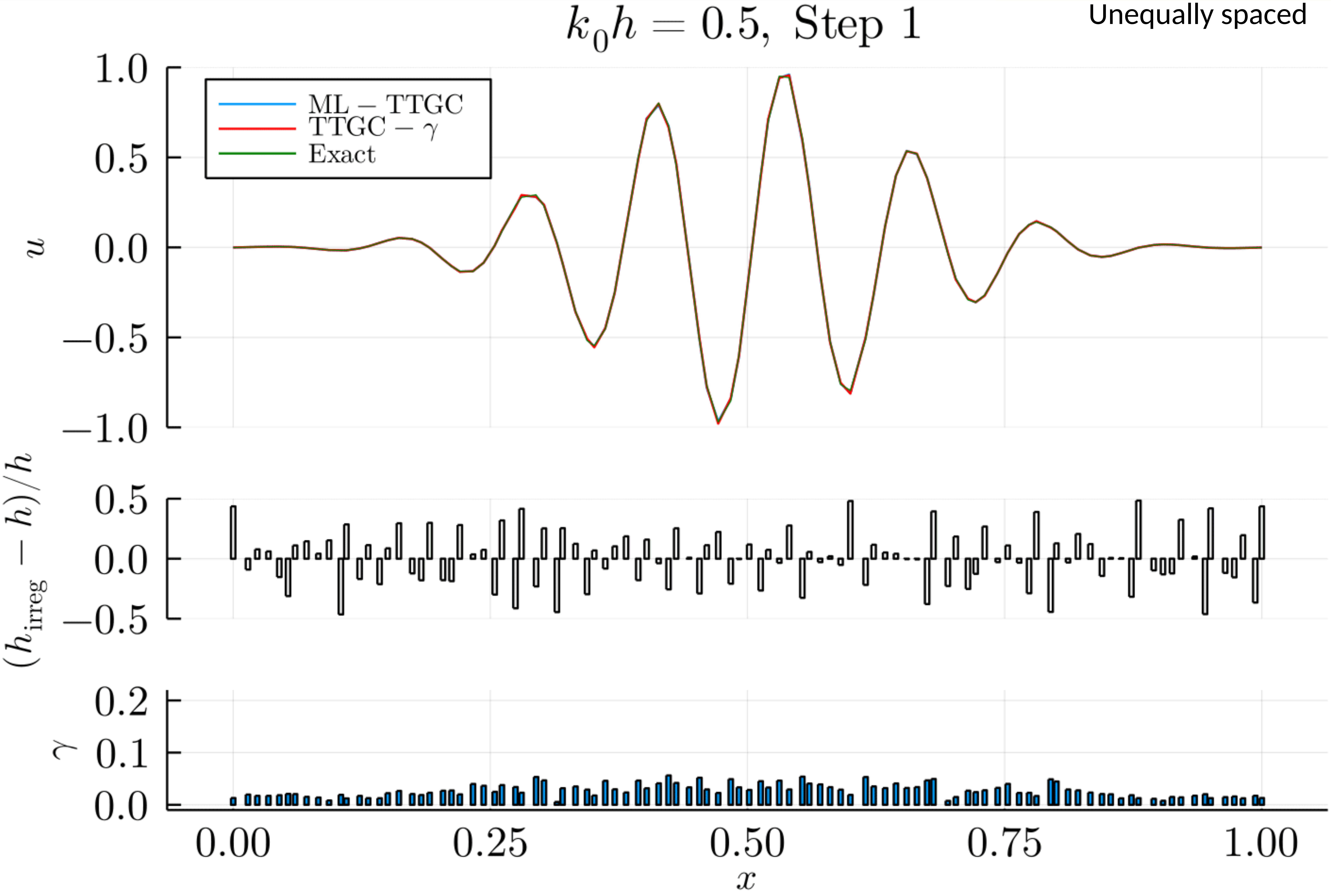
$u_t = -c\,u_x$

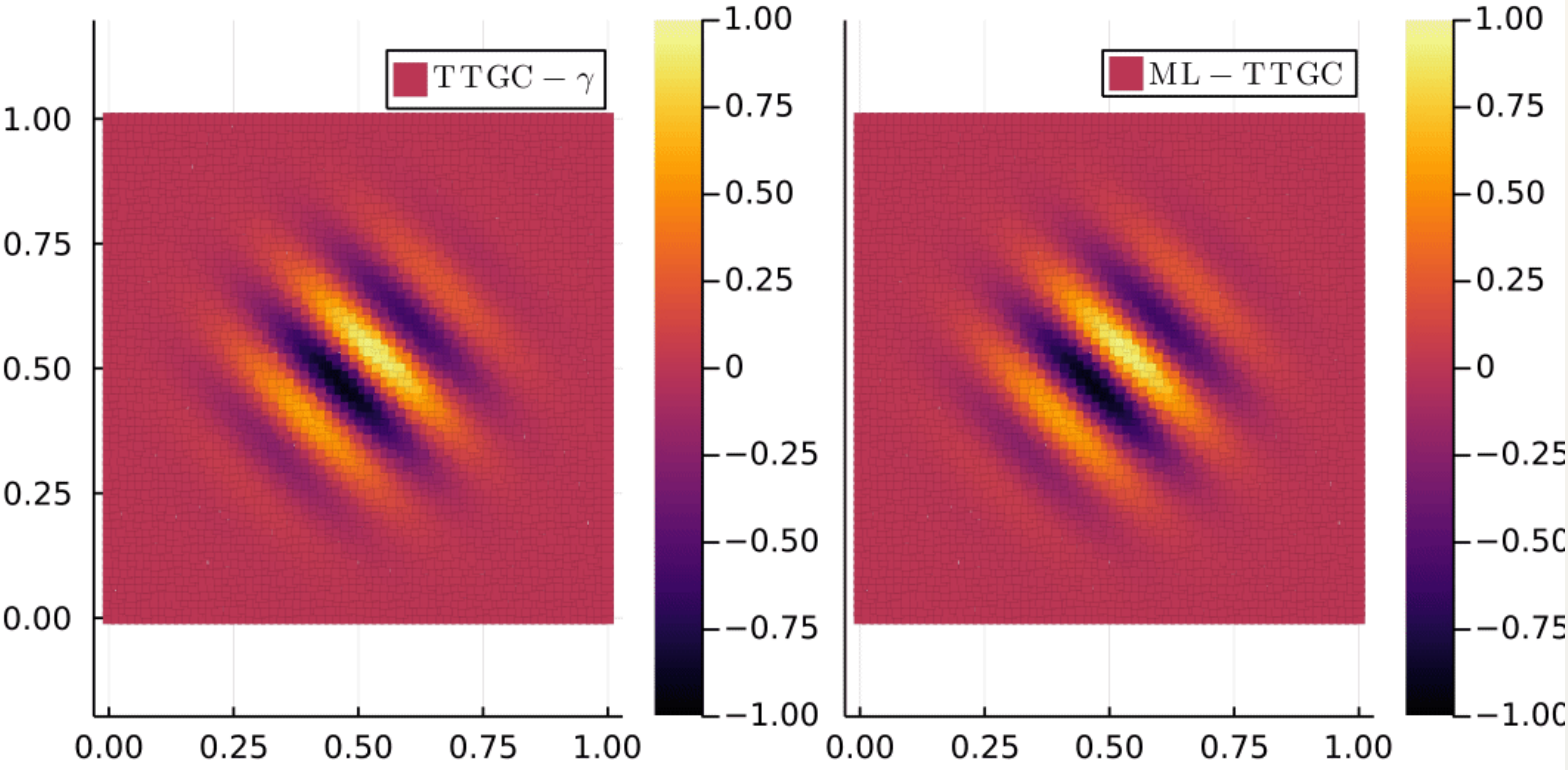
Unequally spaced



$$u_t = -c\,u_x$$

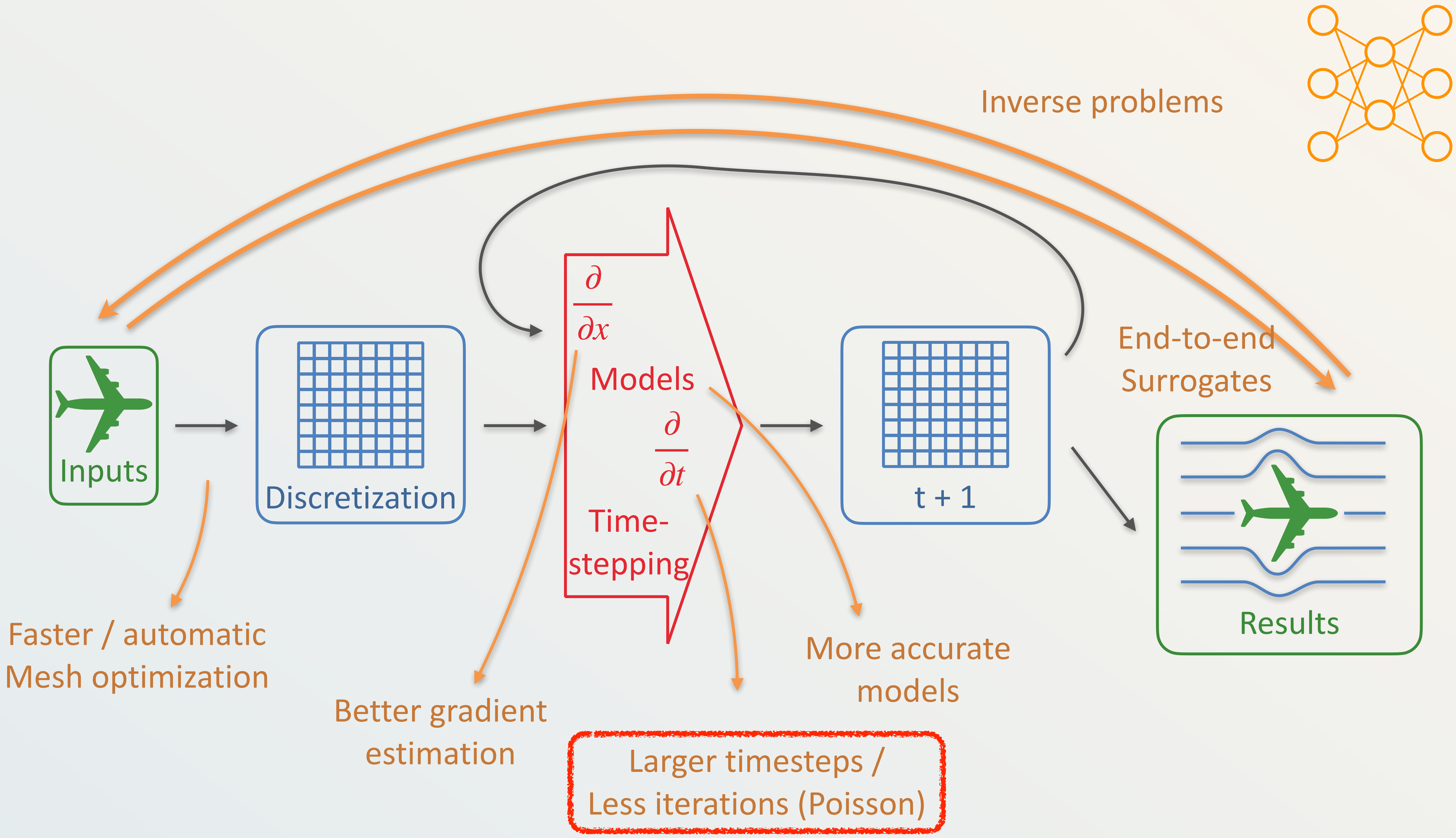
Unequally spaced





WIP: extension to higher dimension

Ongoing PhD of Luciano Drozda, Cerfacs



Inverse problems

Inputs

Discretization

$\frac{\partial}{\partial x}$

Models

$\frac{\partial}{\partial t}$

Time-stepping

t + 1

End-to-end Surrogates

Results

Faster / automatic
Mesh optimization

Better gradient
estimation

Larger timesteps /
Less iterations (Poisson)

More accurate
models

An example of training through the solver

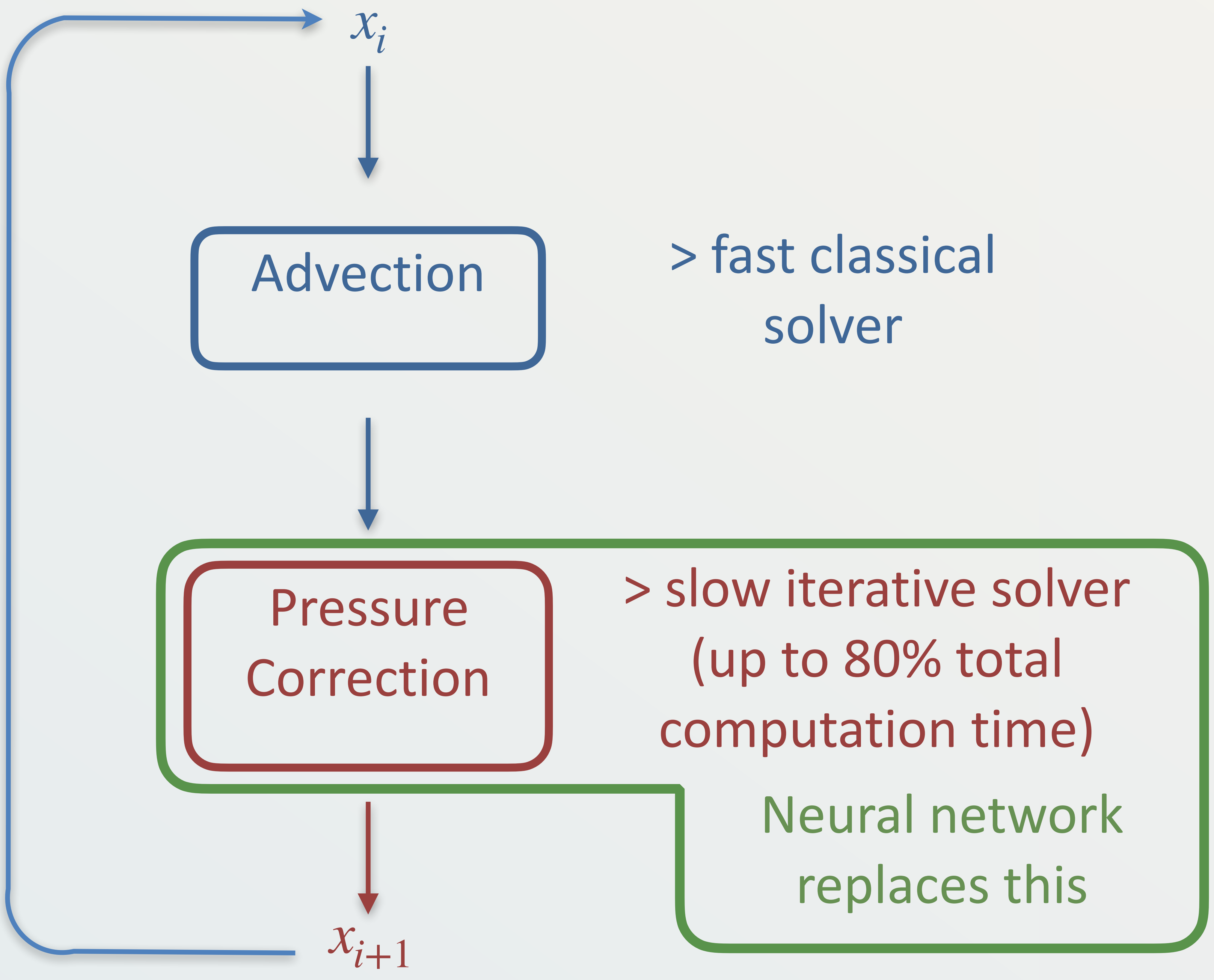
Context

- Second order linear Elliptic Partial Differential Equations equations

- $A\,\frac{\partial^2 u}{\partial x^2} + B\frac{\partial^2 u}{\partial y^2}$ + C$\frac{\partial^2 u}{\partial x \partial y} + D\frac{\partial u}{\partial x} + E\frac{\partial u}{\partial y}$+ F u + G= 0 where
  $B^2 - 4\ AC < 0$

- Simplest cases:
  - Poisson and Laplace equation: $\Delta\ \varphi = f$
  - Frequently seen in multiple physics problems:
    - Incompressible Navier Stokes $\nabla^2 p = f(v,\ V)$
    - Electrostatics $\Delta\,\varphi = -\frac{\rho}{\varepsilon}$ (φ = Scalar electric potential field)
    - Newton Gravity $\Delta\,\phi = 4\,\pi\,G\,\rho$ (ϕ = Scalar gavitational potential)
    - …

Strategy



x_i

Advection

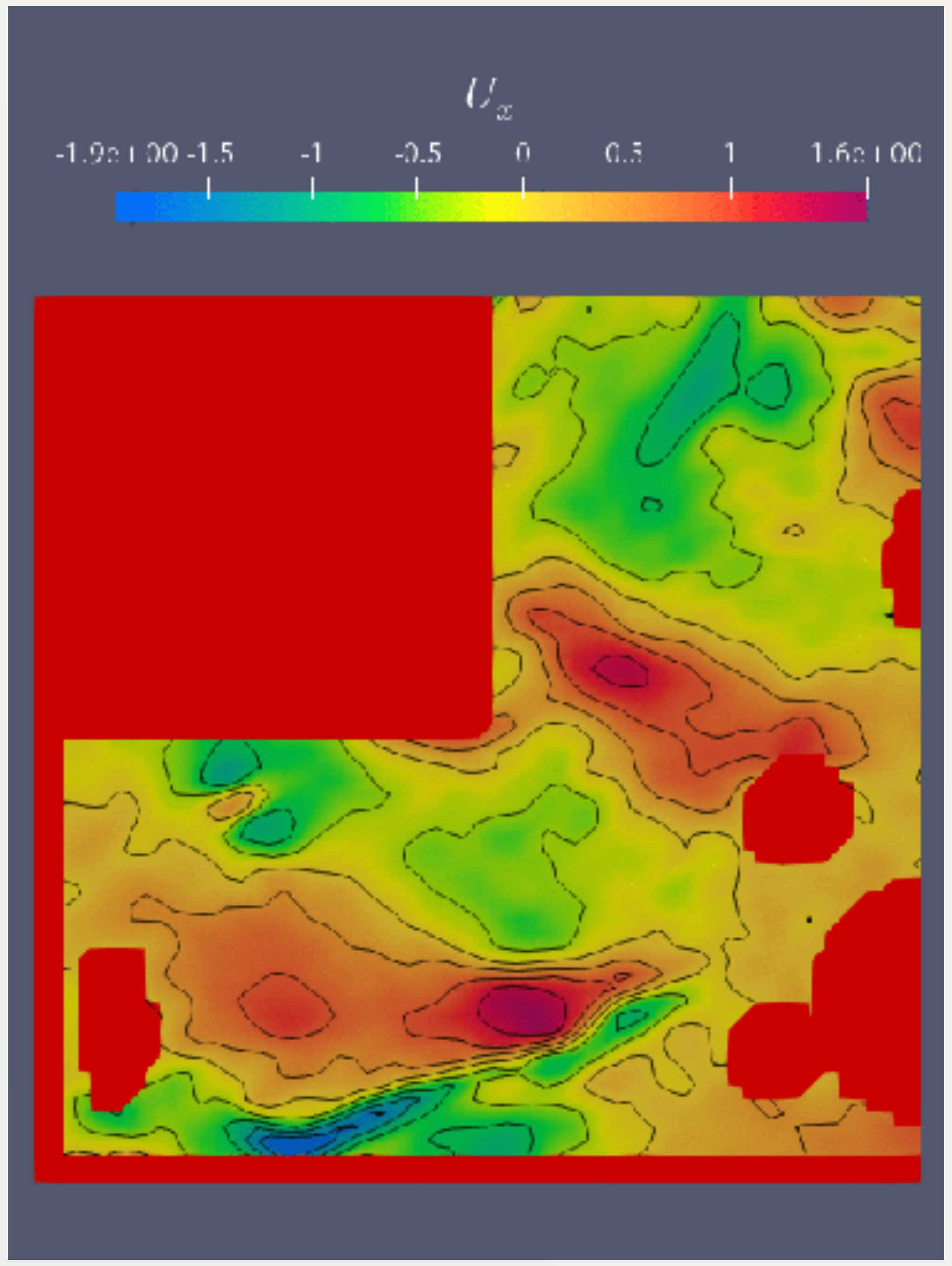
> fast classical solver

Pressure Correction

> slow iterative solver (up to 80% total computation time)

Neural network replaces this

x_{i+1}

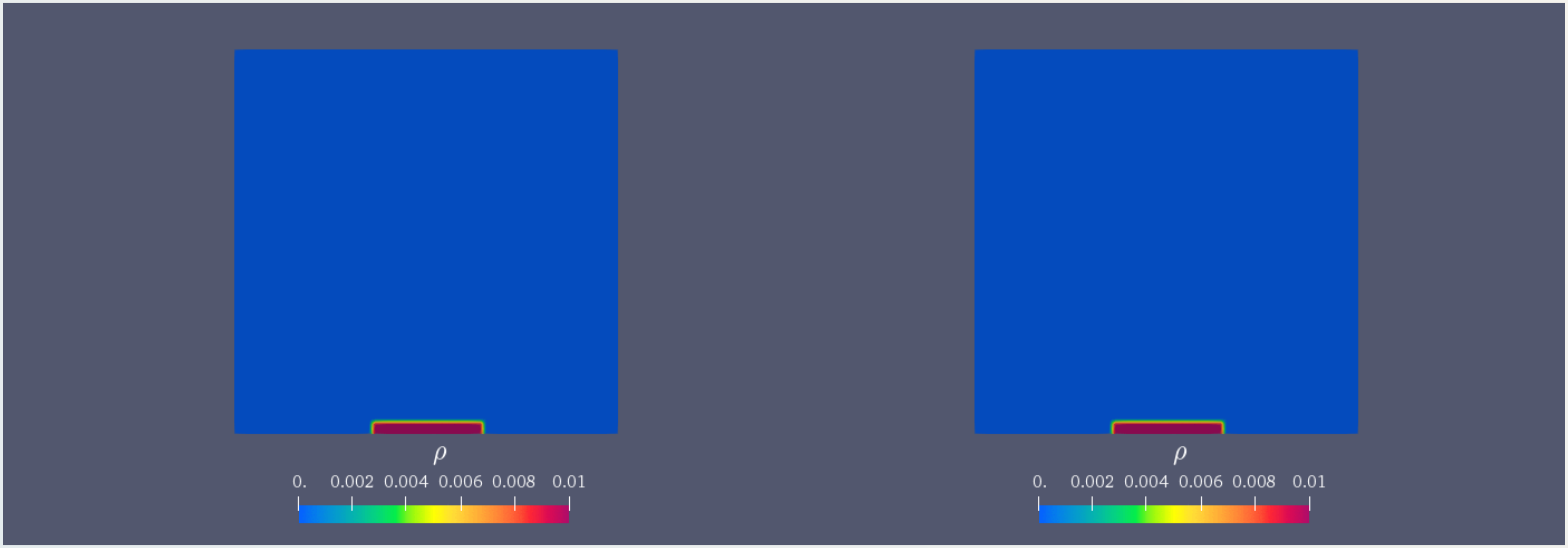


Strategy originally proposed as « FluidNet » [1]

[1] Tompson, Jonathan, et al. "Accelerating eulerian fluid simulation with convolutional networks." *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.

Fully neural pressure

- A fully neural pressure strategy (à la FluidNet)

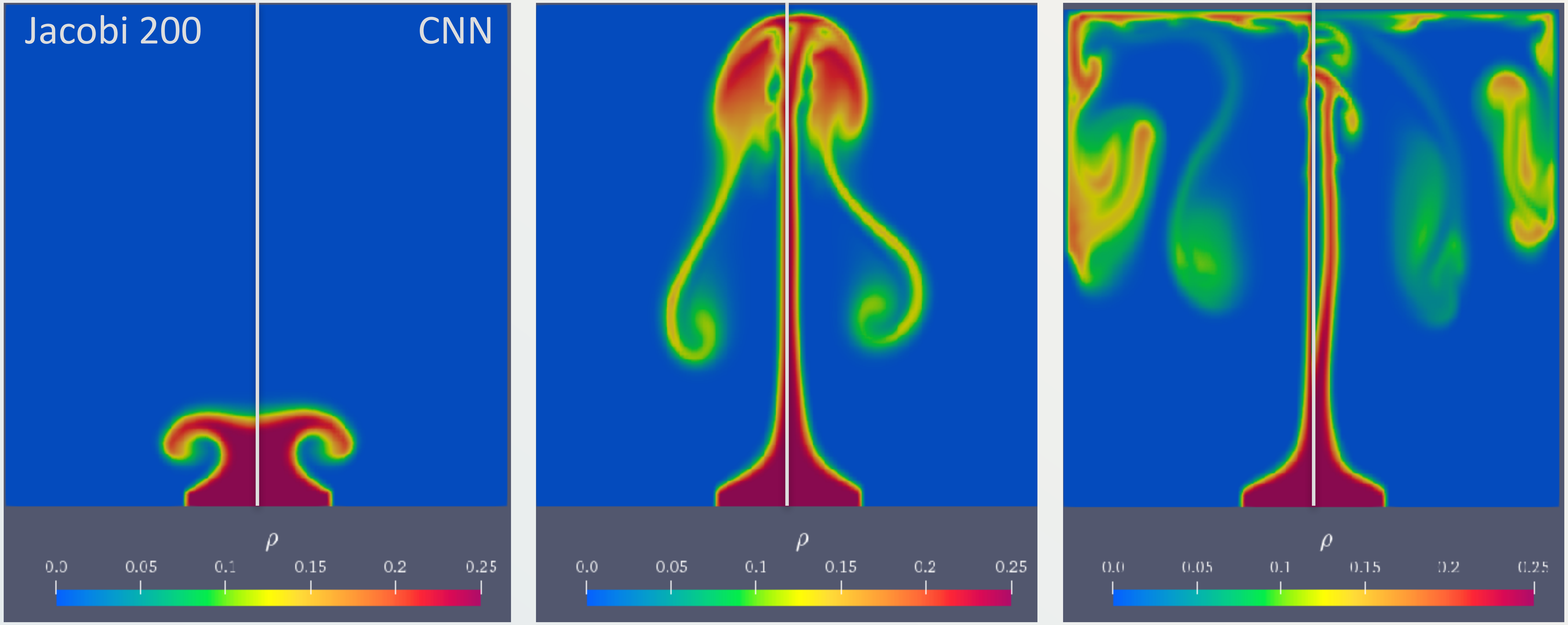


CNN

Jacobi 200

Fully neural pressure

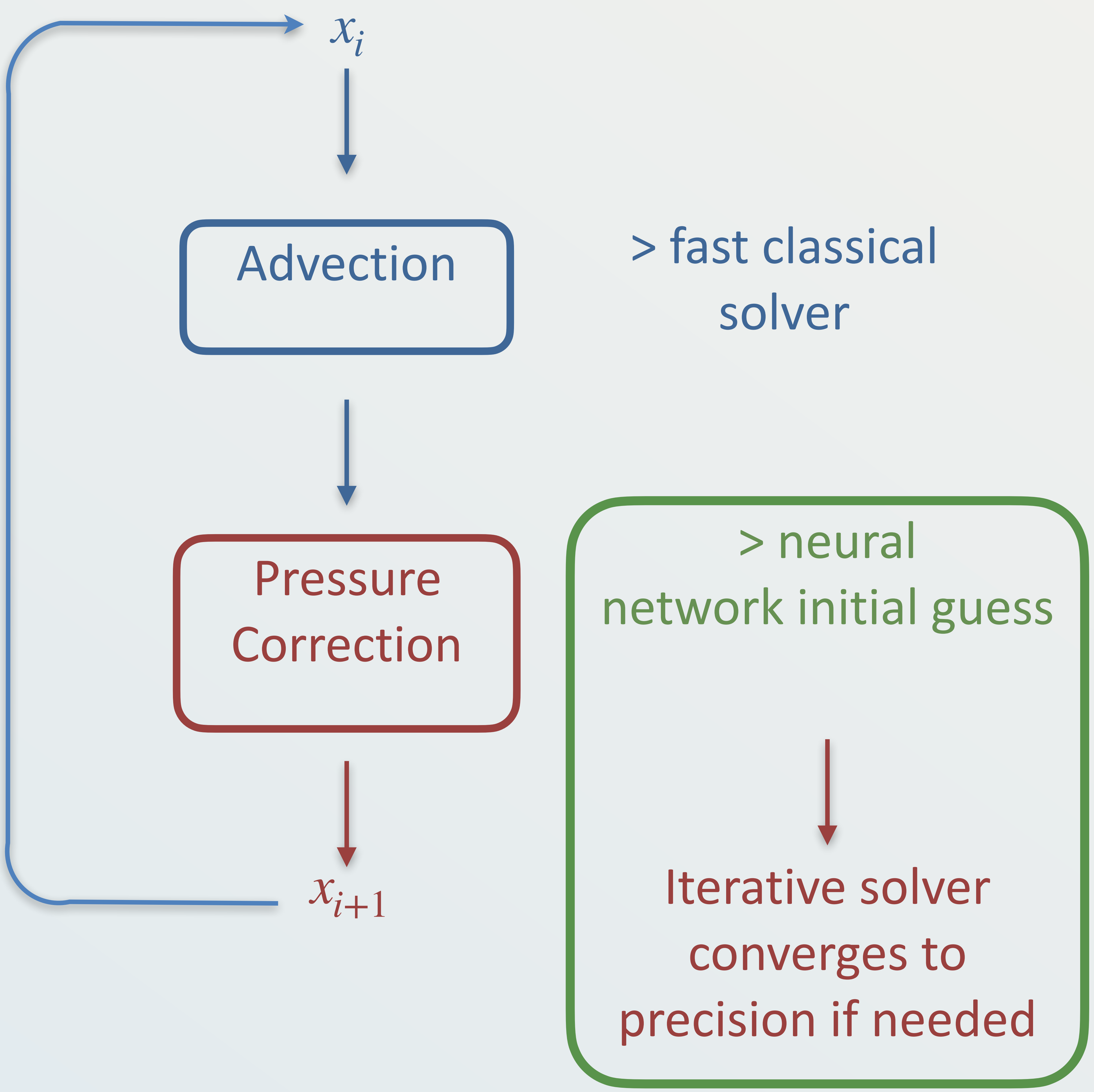
- Interesting results but robustness problems



- How could we guarantee the accuracy of the pressure correction? => Hybrid

Hybrid strategy

Ongoing PhD work of Ekhi Ajuria. ISAE-Supaéro - CERFACS
Supervision: Bénédicte Cuenot (CERFACS), Michaël Bauerheim (ISAE)



x_i

Advection

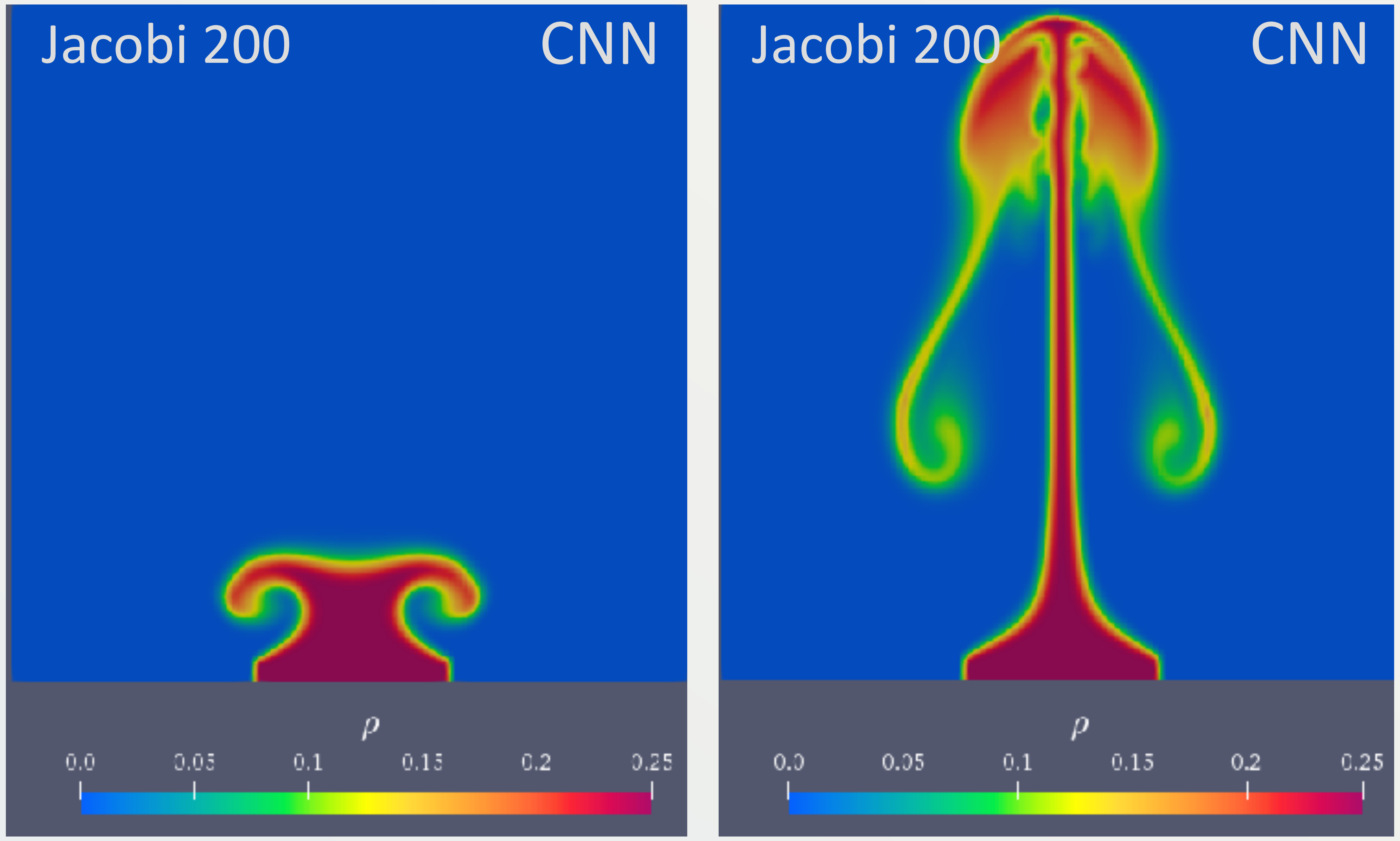
> fast classical solver

Pressure Correction

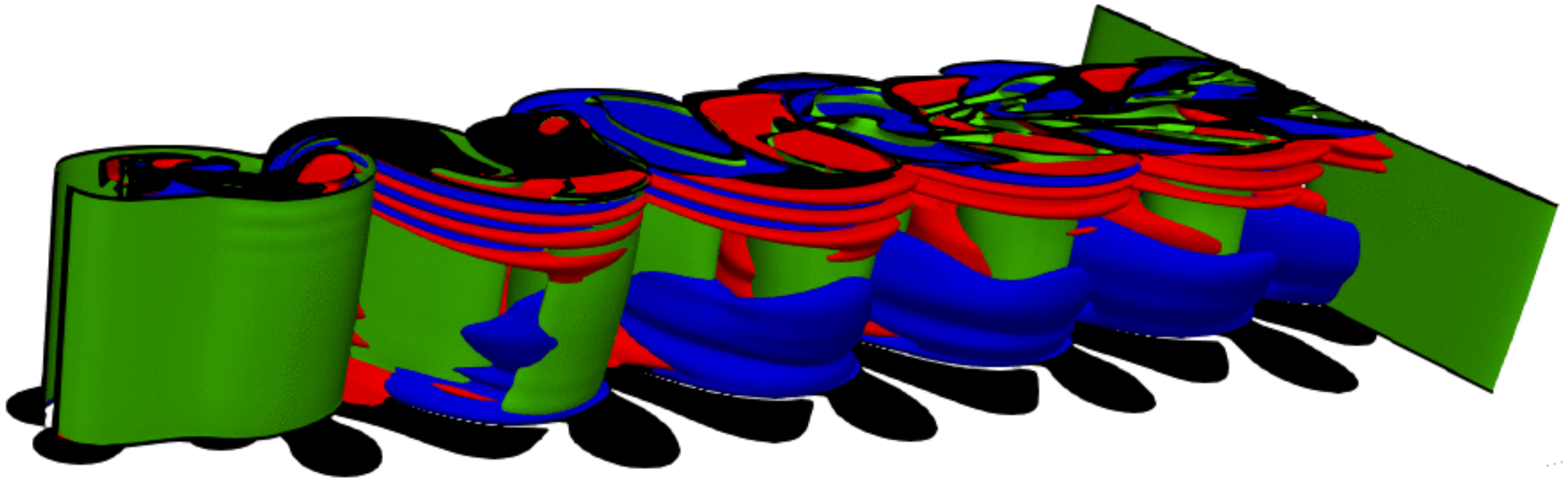
> neural network initial guess

Iterative solver converges to precision if needed

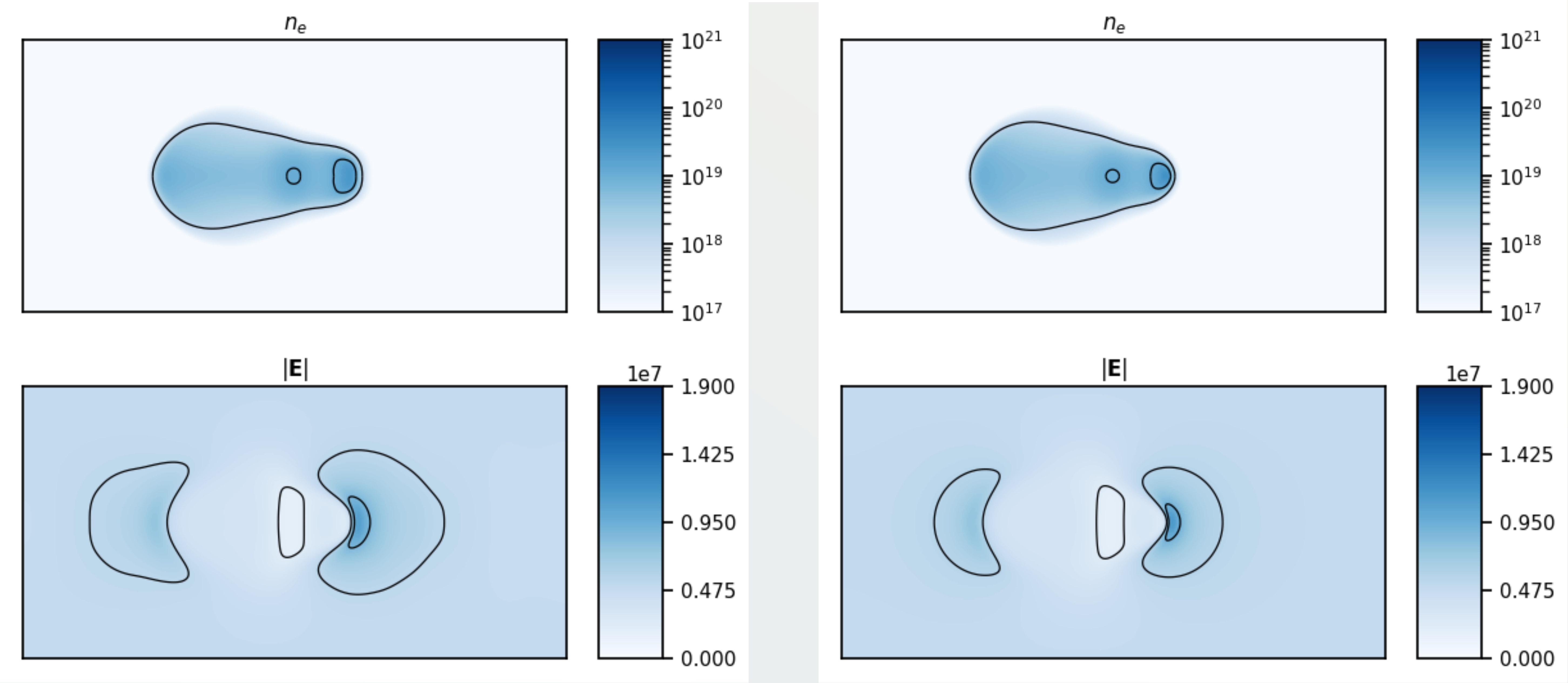
x_{i+1}



Re 6000 3D Von Karman vortex street



electron density and electric field norm

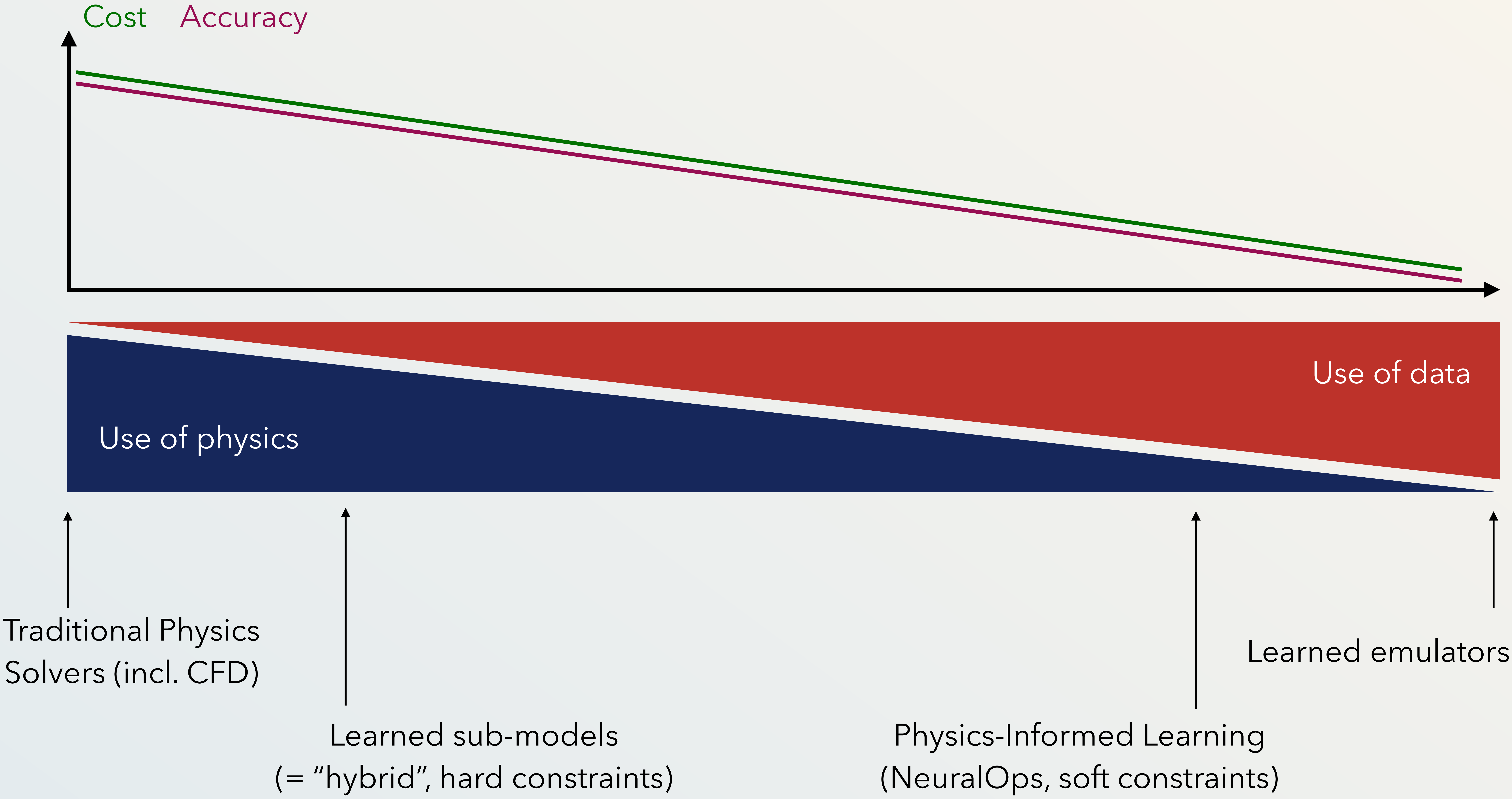


(a) Neural network

(b) Linear system

Ongoing PhDs of E. Ajuria (Cerfacs - ISAE-Supaéro), L. Cheng and G. Bogopolski (Cerfacs)

Weather forecasting: a paradigm shift?



Cost Accuracy

Use of data

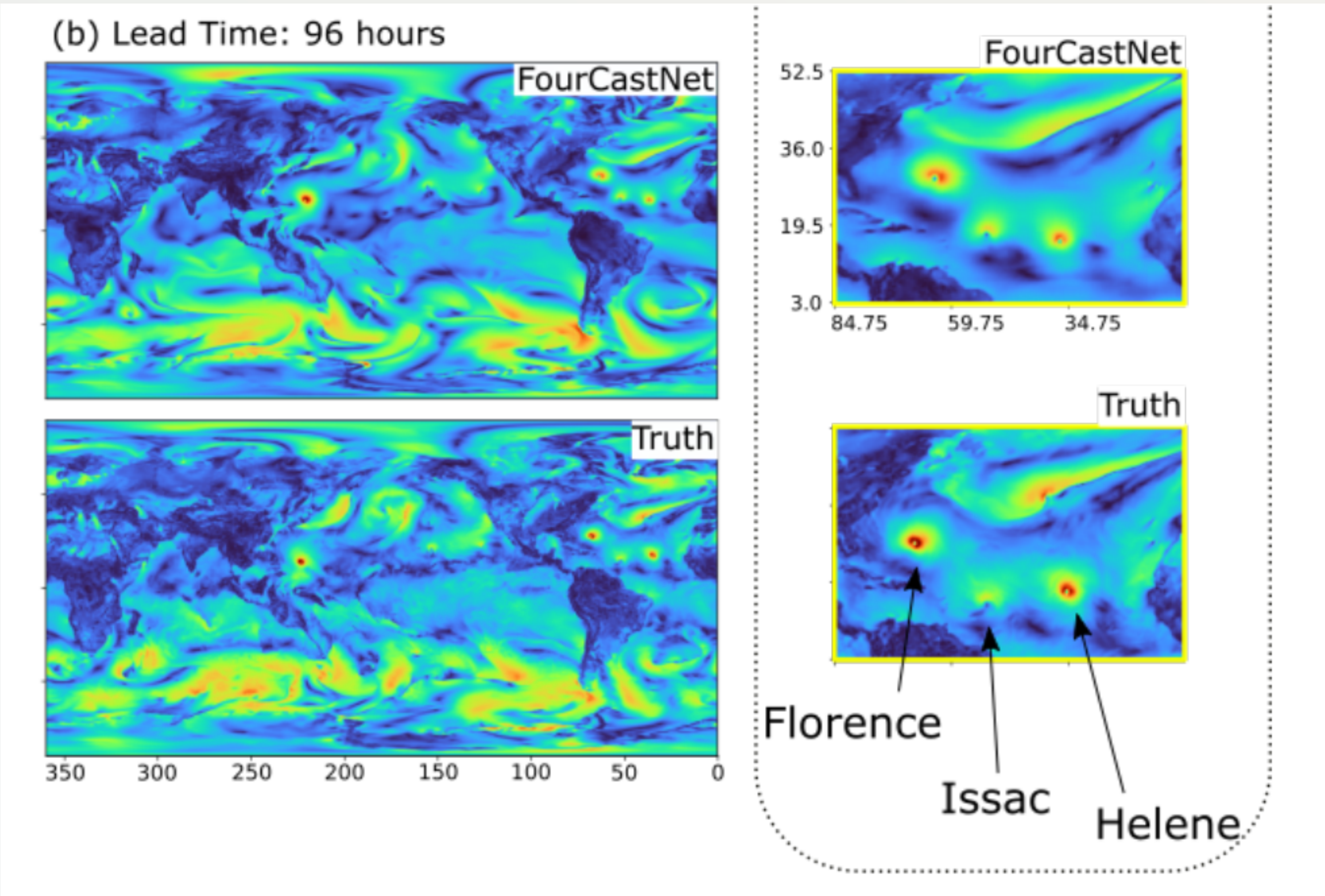
Use of physics

Traditional Physics
Solvers (incl. CFD)

Learned sub-models
(= "hybrid", hard constraints)

Physics-Informed Learning
(NeuralOps, soft constraints)

Learned emulators



FourCastNet
Fourier Neural Operator
Physics-informed

GraphCast
Graph Neural Network
No physics

FengWu
Transformer
No physics

2022

2023

Feb

Nov

Dec

Jan

Apr

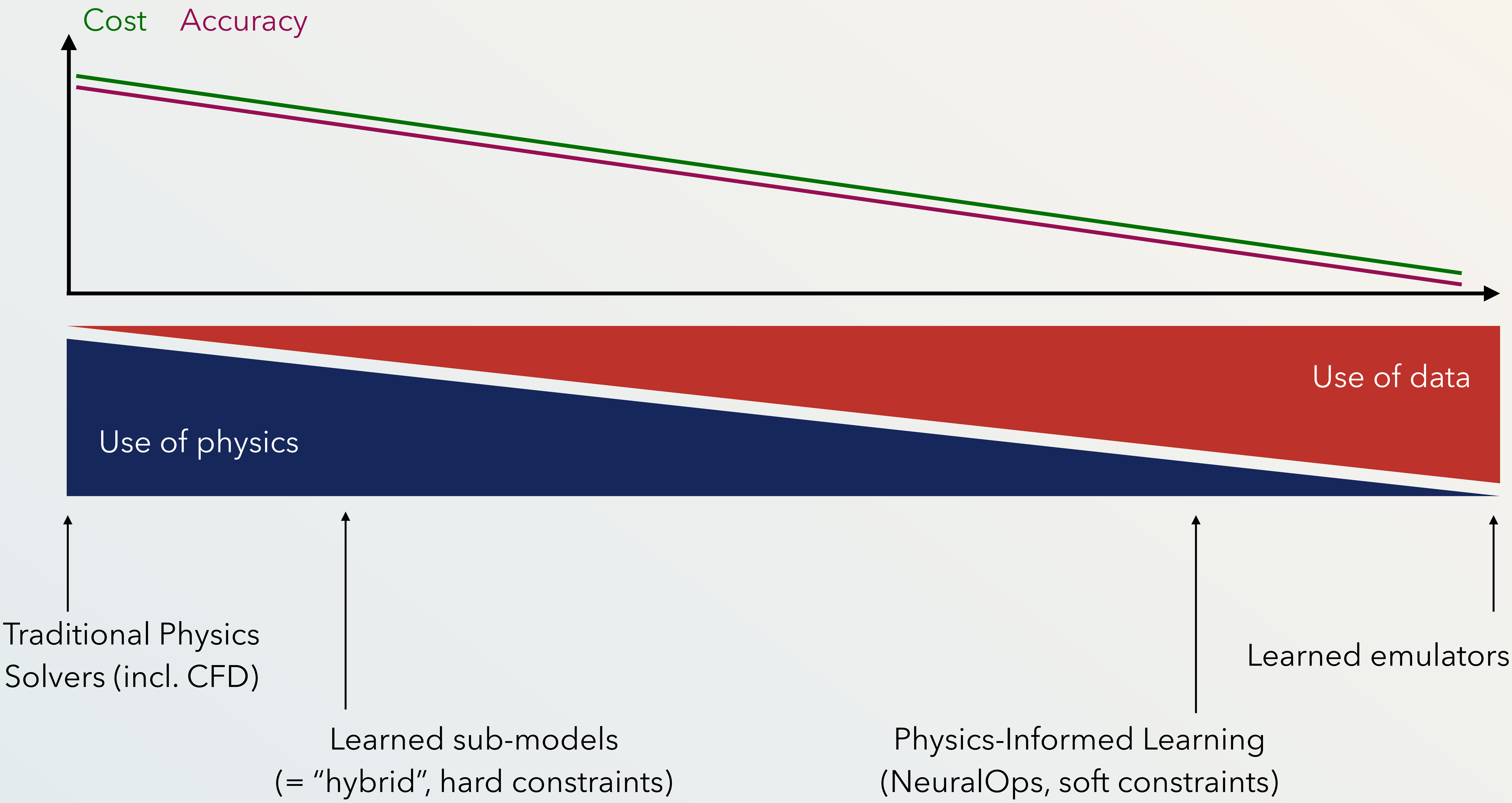
Jun

PanguWeather
Transformer
No physics

ClimaX
Transformer
Foundation model
Multi-resolution
No physics

SwinRDM
Diffusion model
No Physics

https://www.ecmwf.int/en/about/media-centre/science-blog/2023/rise-machine-learning-weather-forecasting



Cost Accuracy

Use of physics

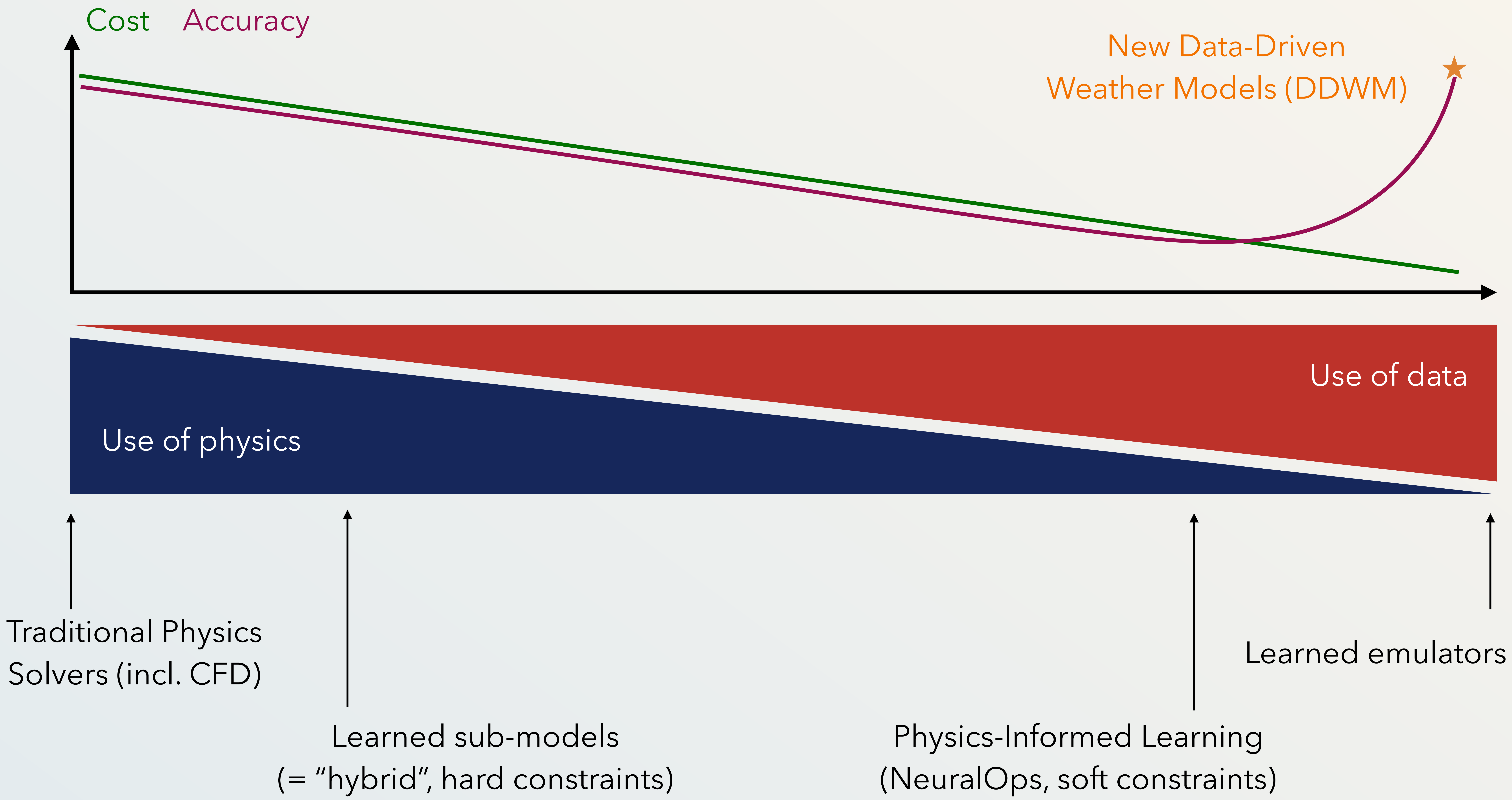
Use of data

Traditional Physics
Solvers (incl. CFD)

Learned sub-models
(= "hybrid", hard constraints)

Physics-Informed Learning
(NeuralOps, soft constraints)

Learned emulators



Cost Accuracy

New Data-Driven
Weather Models (DDWM)

Use of data

Use of physics

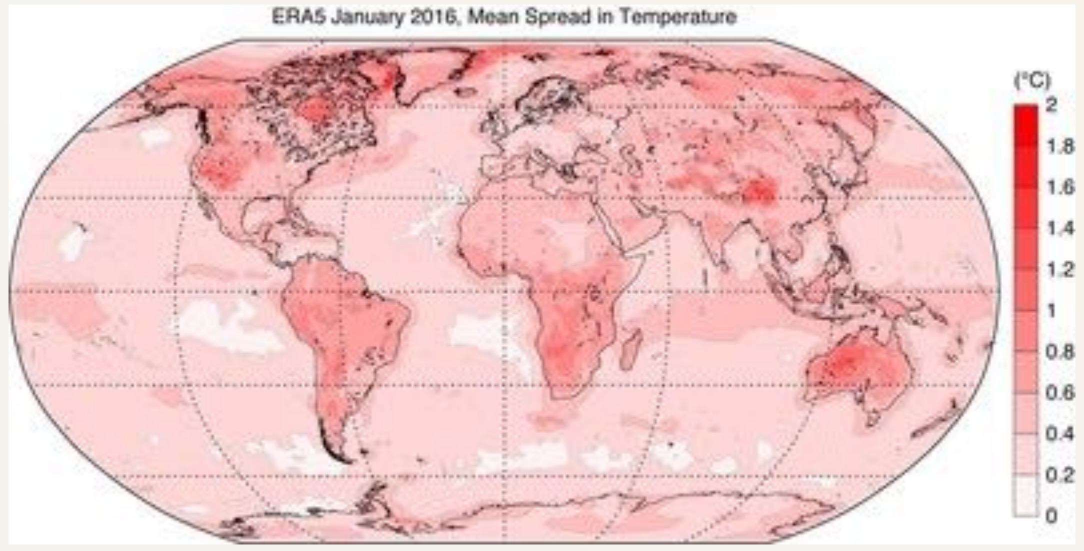
Traditional Physics
Solvers (incl. CFD)

Learned sub-models
(= "hybrid", hard constraints)

Physics-Informed Learning
(NeuralOps, soft constraints)

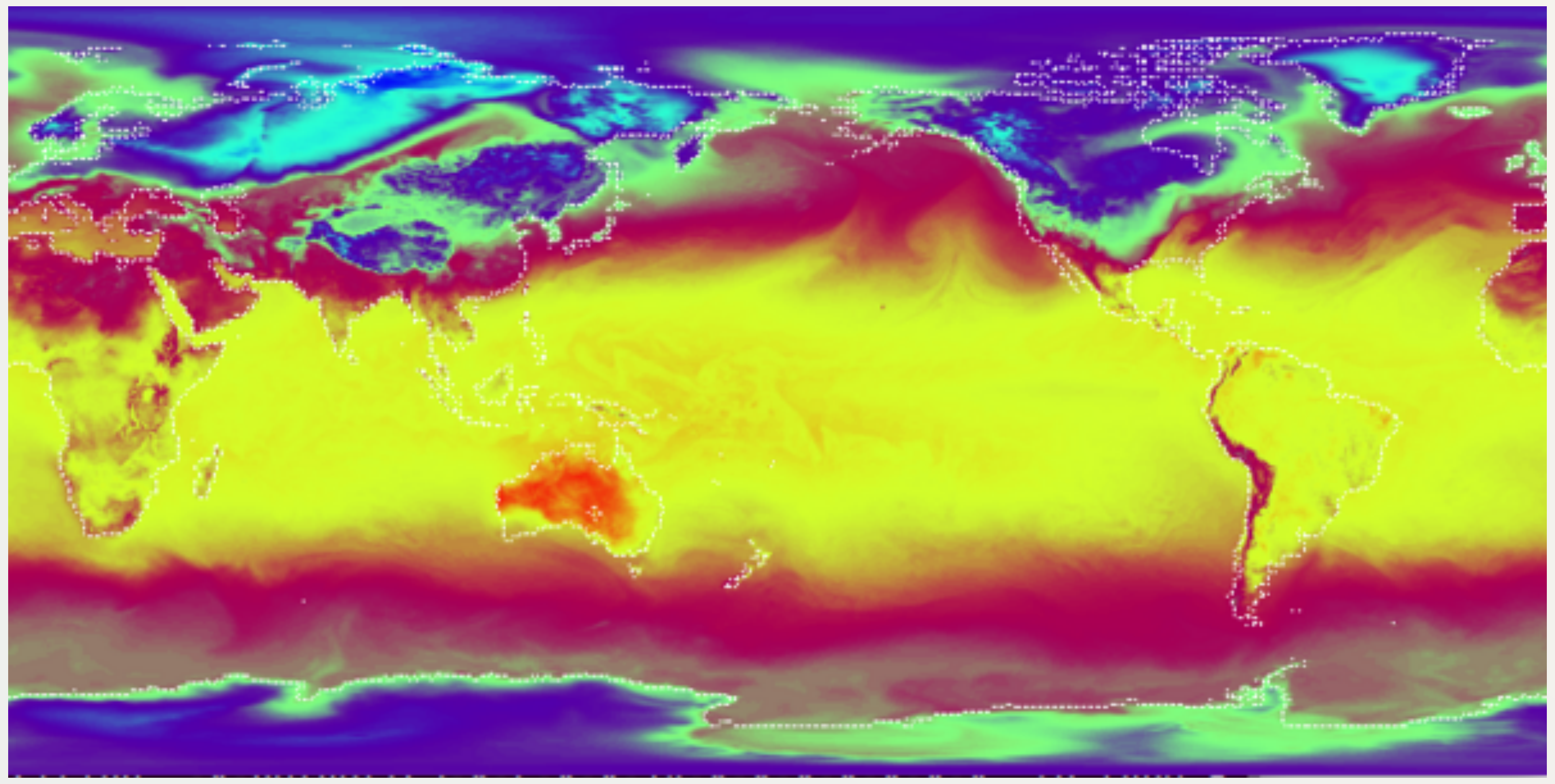
Learned emulators

Key Enabler:
ECMWF Reanalysis v5 (ERA5)



IMPLEMENTED BY ECMWF

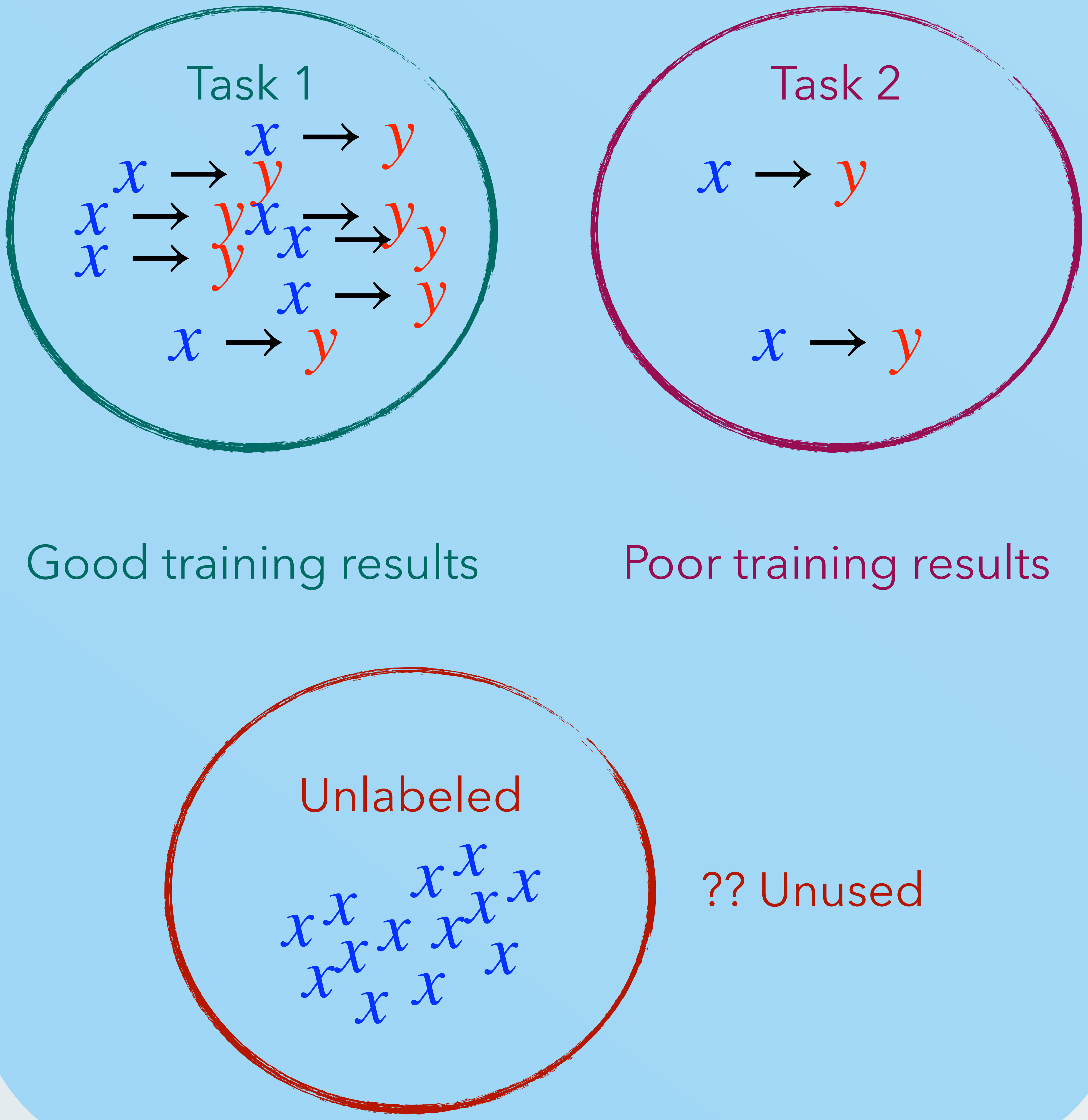
Copernicus
Europe's eyes on Earth



10 PB high-quality reanalysis

- *Hourly* estimates of a large number of atmospheric, land and oceanic climate variables *1940 - present*.
- The data cover the Earth on a *30km grid* (1 million nodes) and resolve the atmosphere using *137 levels* from the surface up to a height of 80km

Supervised training paradigm



Task 1

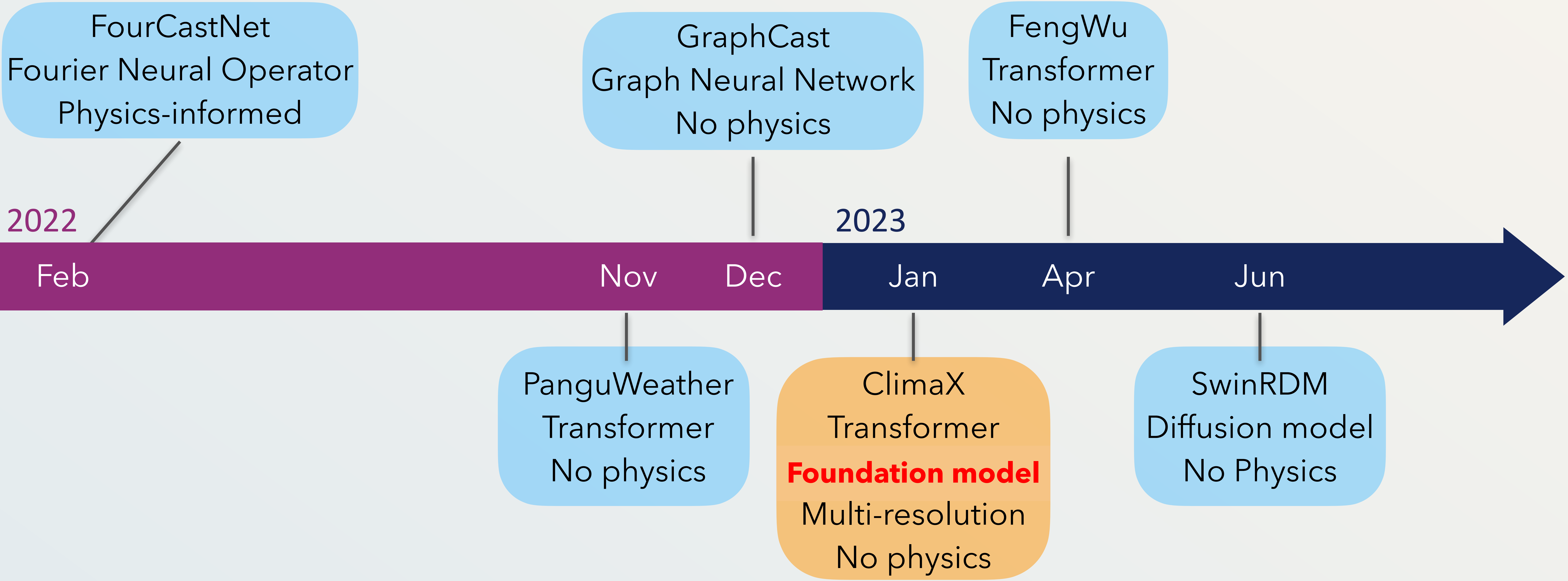
Task 2

Good training results

Poor training results

Unlabeled

?? Unused



2022

Feb

FourCastNet
Fourier Neural Operator
Physics-informed

Nov

PanguWeather
Transformer
No physics

Dec

GraphCast
Graph Neural Network
No physics

2023

Jan

ClimaX
Transformer
Foundation model
Multi-resolution
No physics

Apr

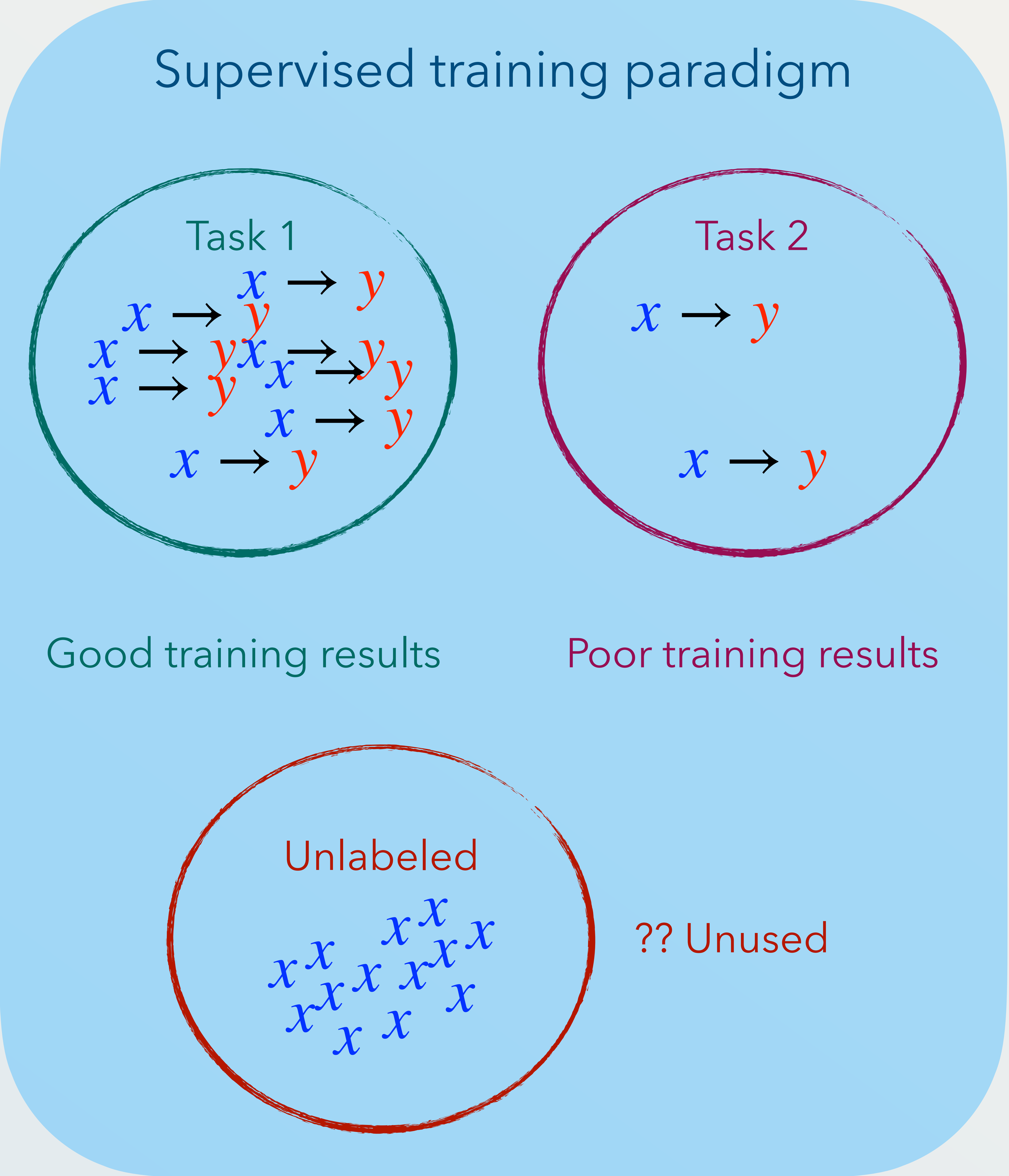
FengWu
Transformer
No physics

Jun

SwinRDM
Diffusion model
No Physics

https://www.ecmwf.int/en/about/media-centre/science-blog/2023/rise-machine-learning-weather-forecasting a.k.a "foundation models"

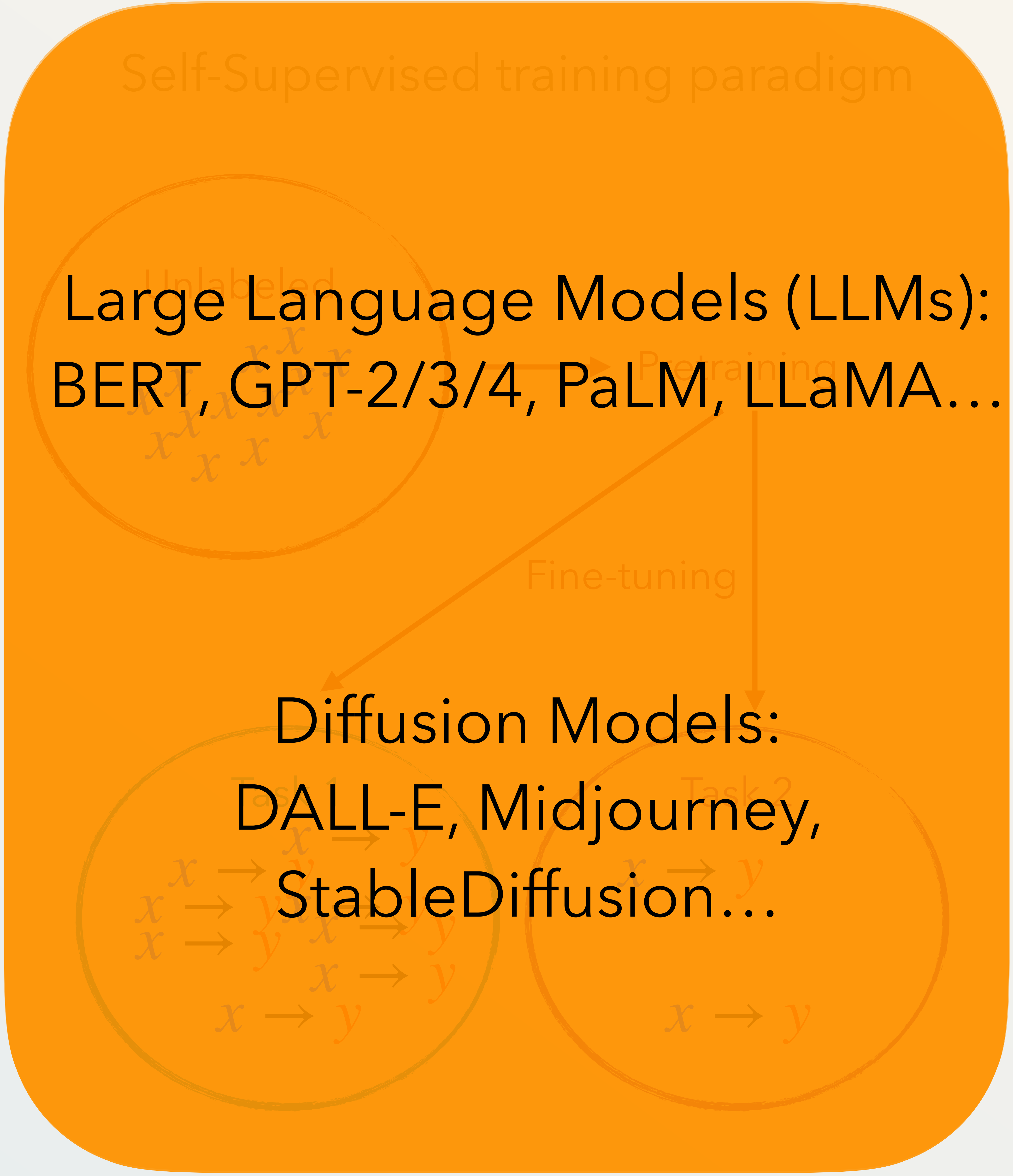
Supervised training paradigm



Good training results

Poor training results

?? Unused



Large Language Models (LLMs): BERT, GPT-2/3/4, PaLM, LLaMA…

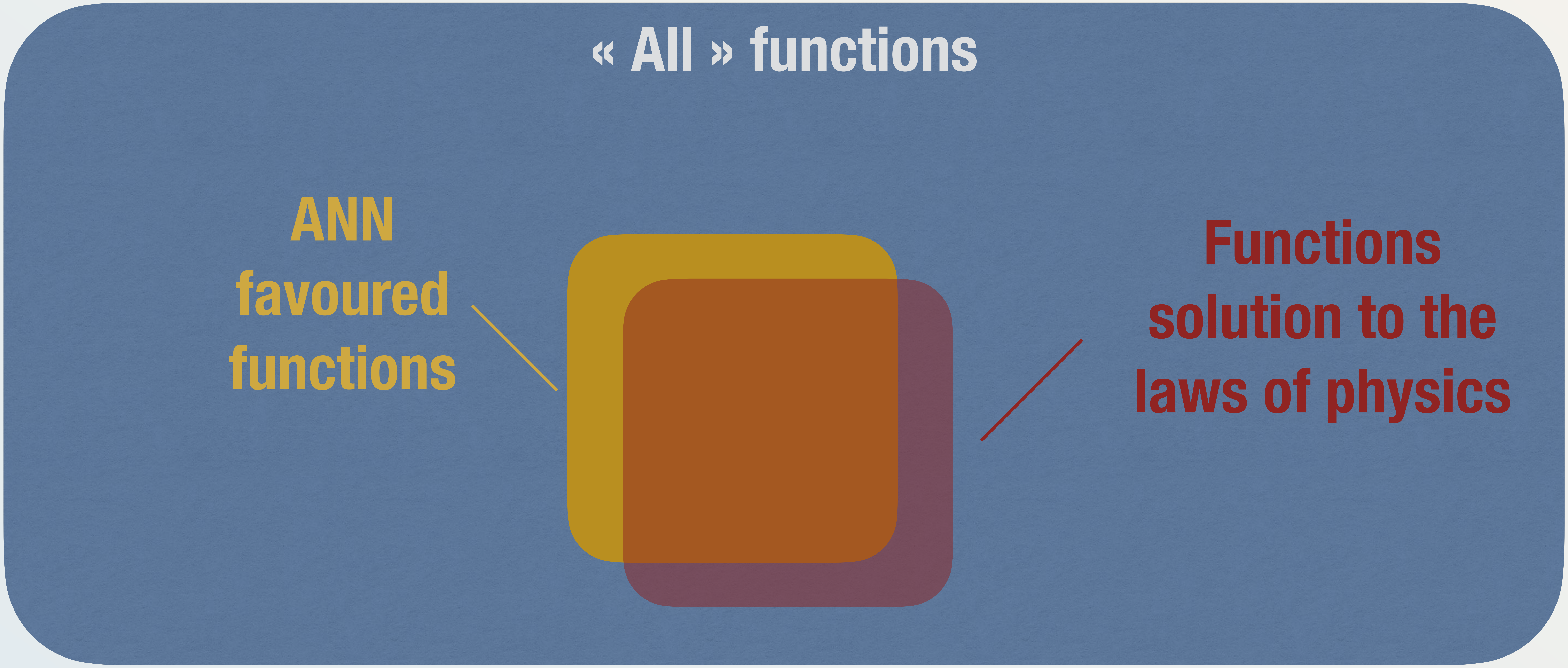
Diffusion Models: DALL-E, Midjourney, StableDiffusion…

<u>Shameless ad</u> internship + PhD open between CERFACS / La Sorbonne to apply this to general CFD. Come see me if interested!

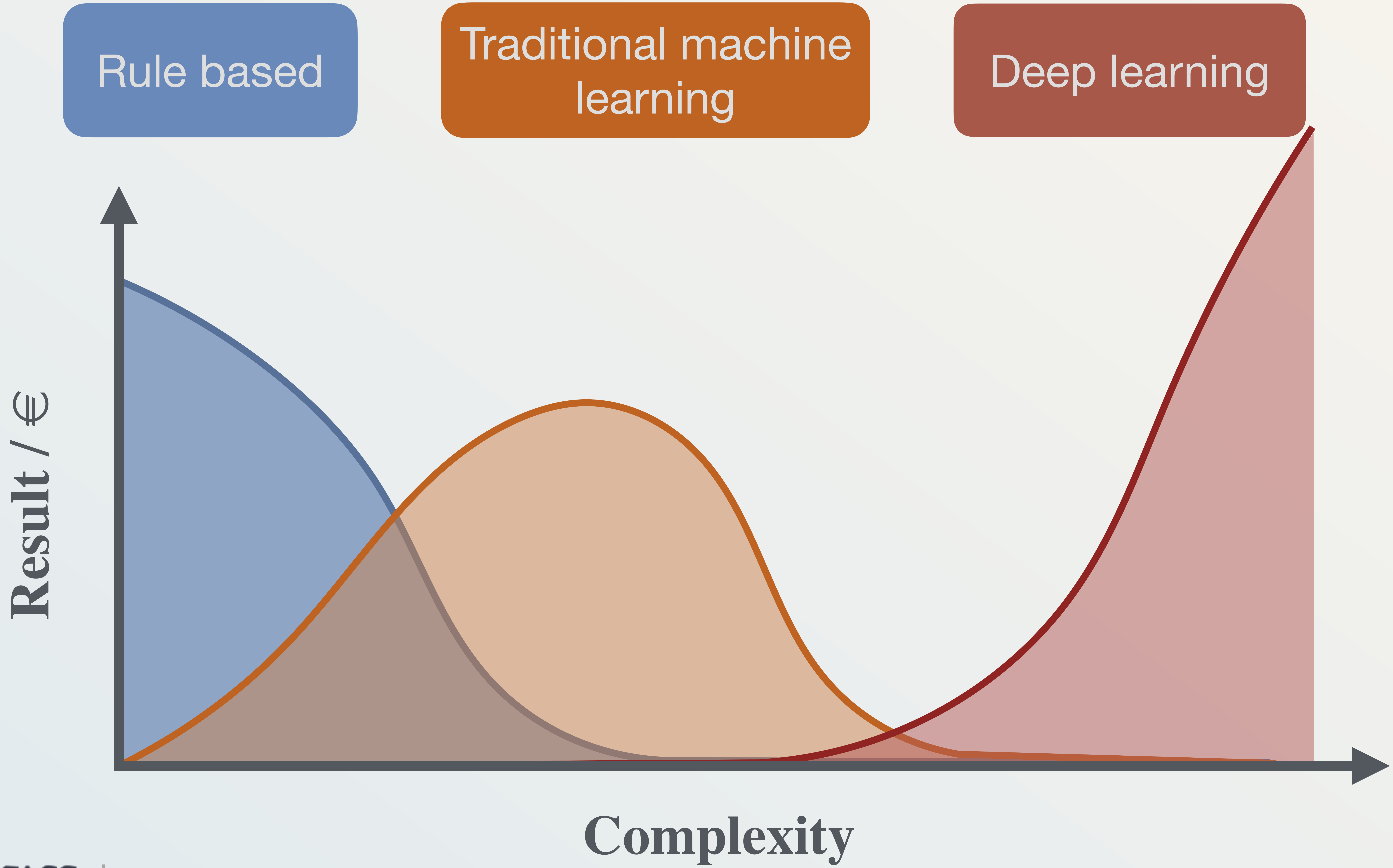
Conclusion

Why does it work?

Henry W. Lin, Max Tegmark, and David Rolnick, *Why does deep and cheap learning work so well?* Journal of Statistical Physics (2017)



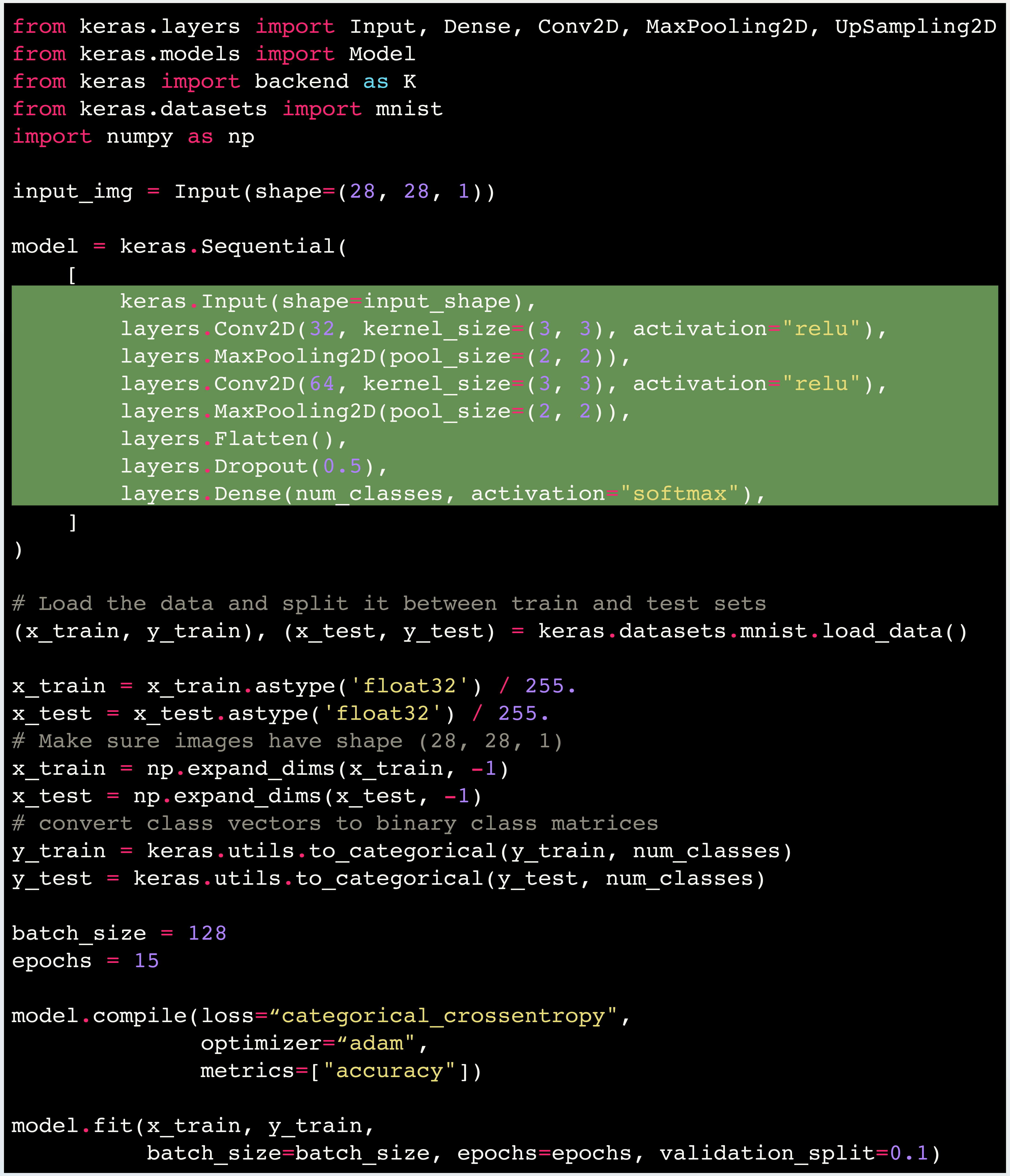
When should we use it?



For what problems?

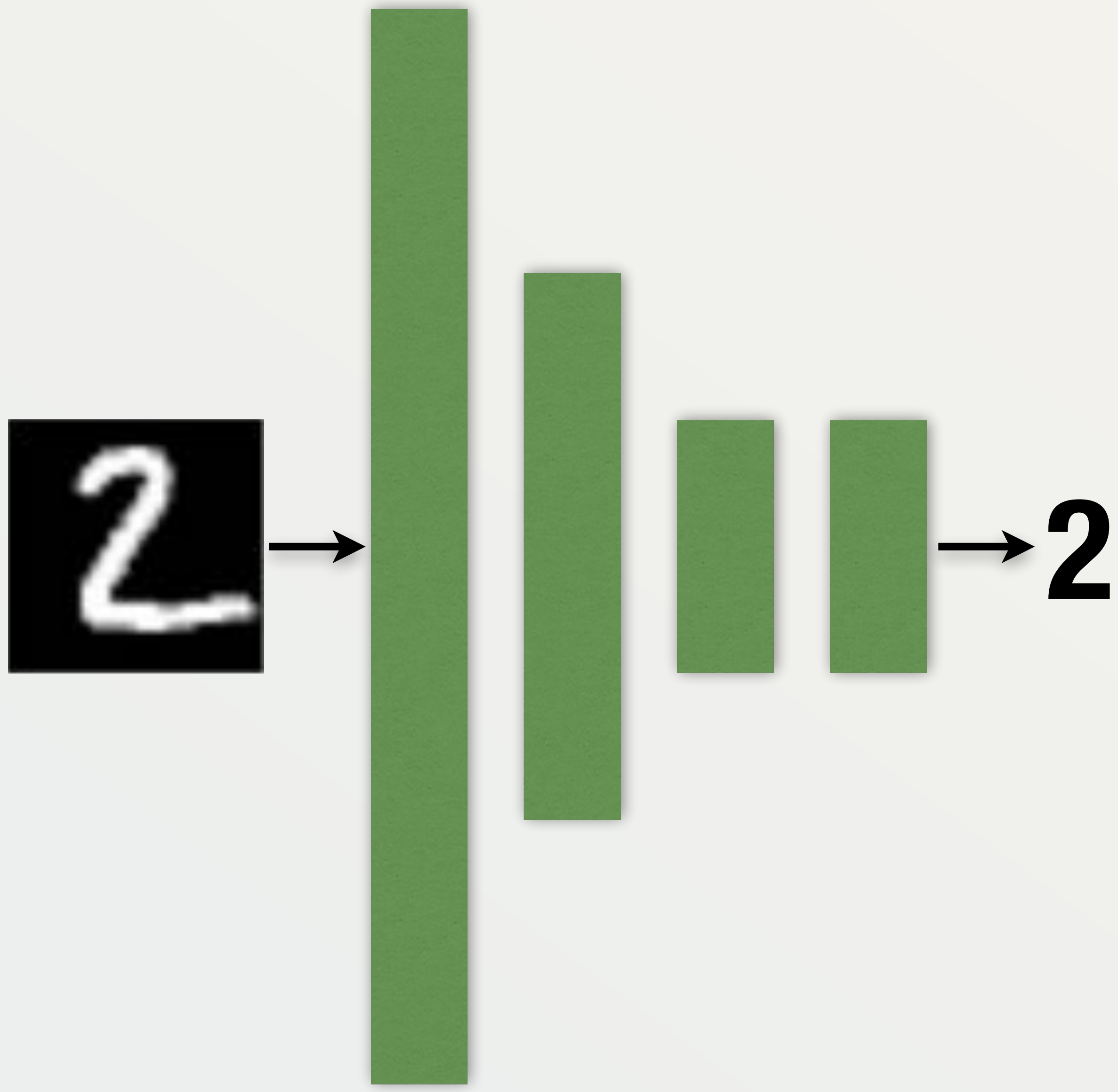
- When you have an analytical solution, use it!
- But many problems (current and to be formulated) don't. Then, if you have data, there might be hope…

What does it look like?



```
from keras.layers import Input, Dense, Conv2D, MaxPooling2D, UpSampling2D
from keras.models import Model
from keras import backend as K
from keras.datasets import mnist
import numpy as np

input_img = Input(shape=(28, 28, 1))

model = keras.Sequential(
    [
        keras.Input(shape=input_shape),
        layers.Conv2D(32, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Flatten(),
        layers.Dropout(0.5),
        layers.Dense(num_classes, activation="softmax"),
    ]
)

# Load the data and split it between train and test sets
(x_train, y_train), (x_test, y_test) = keras.datasets.mnist.load_data()

x_train = x_train.astype('float32') / 255.
x_test = x_test.astype('float32') / 255.
# Make sure images have shape (28, 28, 1)
x_train = np.expand_dims(x_train, -1)
x_test = np.expand_dims(x_test, -1)
# convert class vectors to binary class matrices
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)

batch_size = 128
epochs = 15

model.compile(loss=“categorical_crossentropy",
              optimizer=“adam",
              metrics=["accuracy"])

model.fit(x_train, y_train,
          batch_size=batch_size, epochs=epochs, validation_split=0.1)
```

A simple convolutional classifier



A few lines of python / lua / java… and you're off

2023 - UIAP Course Summary

	26/09	*Python*
	3/10	*Introduction to the course*
	17/10 - 5/12	*ML and the Rocket Combustion Project*
20%	*12/12*	*Oral presentation on Rocket Project*

12/12	*Intro to Neural Networks*	
19/12 - 23/01	*Satellite Project*	
30/01	*Oral presentation on Satellite Project*	20%
30/01	*Conference*	
4/02	*Written report on Satellite Project*	60%

Evaluation