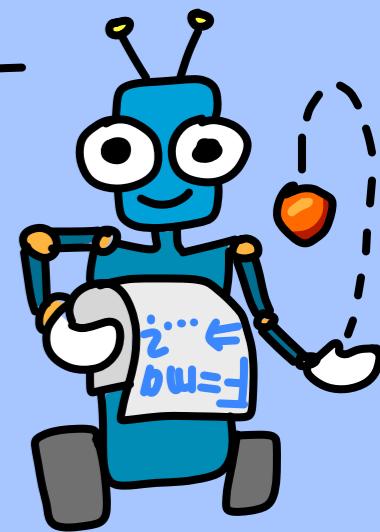


# ADVANCED MACHINE LEARNING FOR PHYSICS, SCIENCE, AND ARTIFICIAL SCIENTIFIC DISCOVERY

LECTURE SERIES 2021/22

BY FLORIAN MARQUARDT  
(FRIEDRICH-ALEXANDER UNIVERSITÄT  
ERLANGEN-NÜRNBERG &  
MAX PLANCK INSTITUTE FOR THE  
SCIENCE OF LIGHT, ERLANGEN, GERMANY)

MONDAY 6:00 PM-7:30 CET (GERMANY)  
WEDNESDAY 6 PM - 7:30



# ALL INFO ON WEBSITE

- DISCUSSION GROUP
- CODE
- LINKS TO LITERATURE
- LINK TO YOUTUBE CHANNEL
- HOMEWORK PROBLEMS  
(POSTED EACH WEDNESDAY)
- INFO ABOUT EXAM FOR  
FAU STUDENTS
- OTHER STUDENTS: CAN DO  
"MINI PROJECT" AT END &  
GET CERTIFICATE

## 1.

## INTRODUCTION

## 1.1

"MACHINE LEARNING"  
/ COMPUTER

$$y = F(x)$$

$$y = F(x, \text{EXPERIENCE})$$

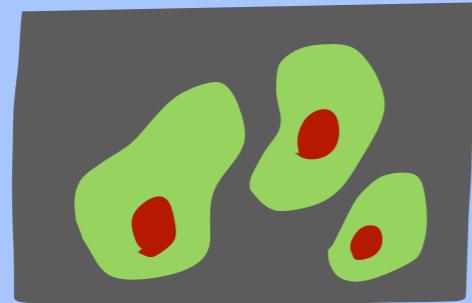
MEMORY STATE  
DEPENDS

→ IMPROVE! (WITH RESPECT  
TO A TASK)

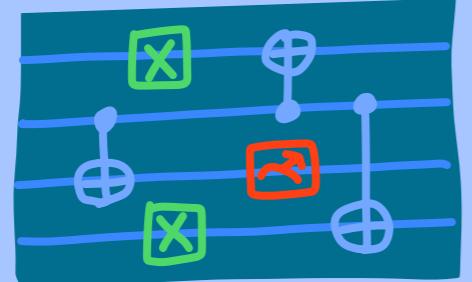
→ NO HUMAN DESIGN OF  $F(x)$  NEEDED

→ GENERAL LEARNING ALGORITHMS

## SCIENCE



MICROSCOPY



QUANTUM CONTROL  
STRATEGIES

## TECHNOLOGY



SELF-DRIVING CARS  
& ROBOTICS



SPEECH RECOGNITION

THIS MOVIE IS  
HARDLY WORTH  
WATCHING...

NATURAL LANGUAGE  
PROCESSING



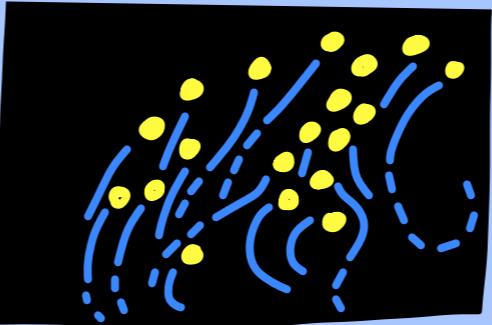
IMAGE RESTORATION

JE SUIS UNE  
MACHINE...  
→ I AM...

TEXT TRANSLATION



PLAYING GAMES

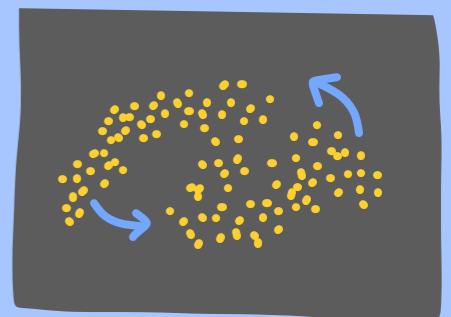


MANY-PARTICLE  
DYNAMICS

ACTIVITY PREDICTION  
FOR CHEMICALS



PHASE TRANSITIONS



PROTEIN FOLDING

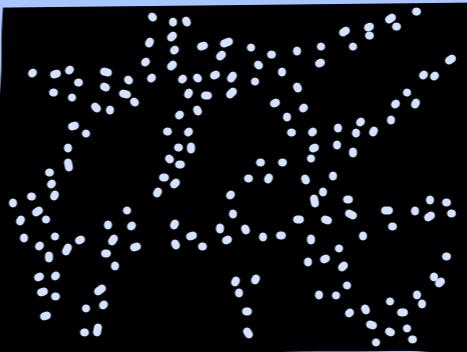
1.2

## SCIENTIFIC DISCOVERY & COMPUTERS

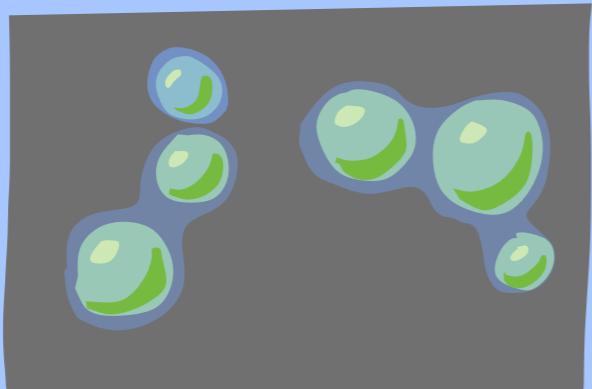
ALREADY SINCE 1950s : HUMAN SCIENTIFIC  
DISCOVERY AIDED BY  
COMPUTERS



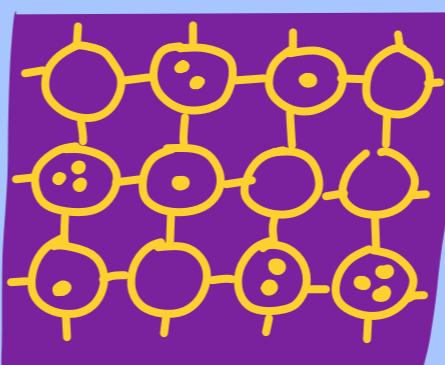
MANY-PARTICLE DYNAMICS,  
NONLINEAR DYNAMICS,  
STATISTICAL PHYSICS



COSMOLOGY



QUANTUM CHEMISTRY  
& MATERIALS SCIENCE



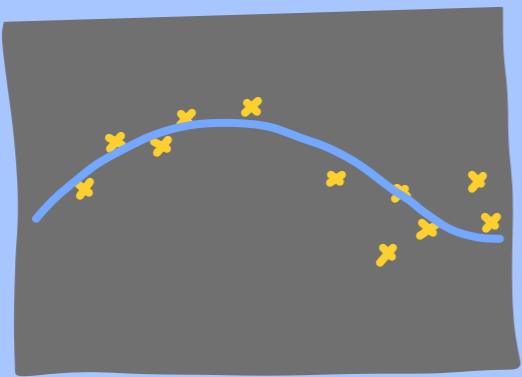
LATTICE QUANTUM  
FIELD THEORY



CLIMATE  
MODELS

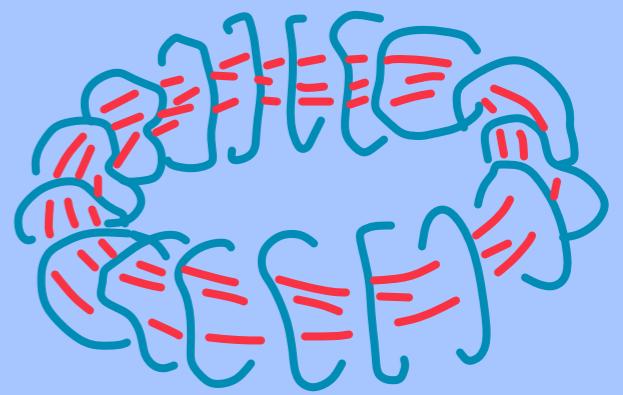
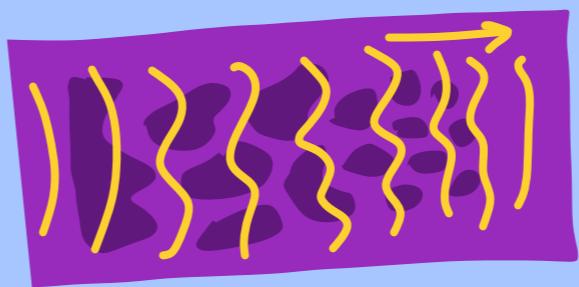


4-COLOR THEOREM



GCTAATGACT  
AGCTTACAGA  
TGGATCCATG  
ACTGGAGTAA

## DATA ANALYSIS



## OPTIMIZATION

→ NATURAL TRANSITION TO  
MACHINE LEARNING METHODS

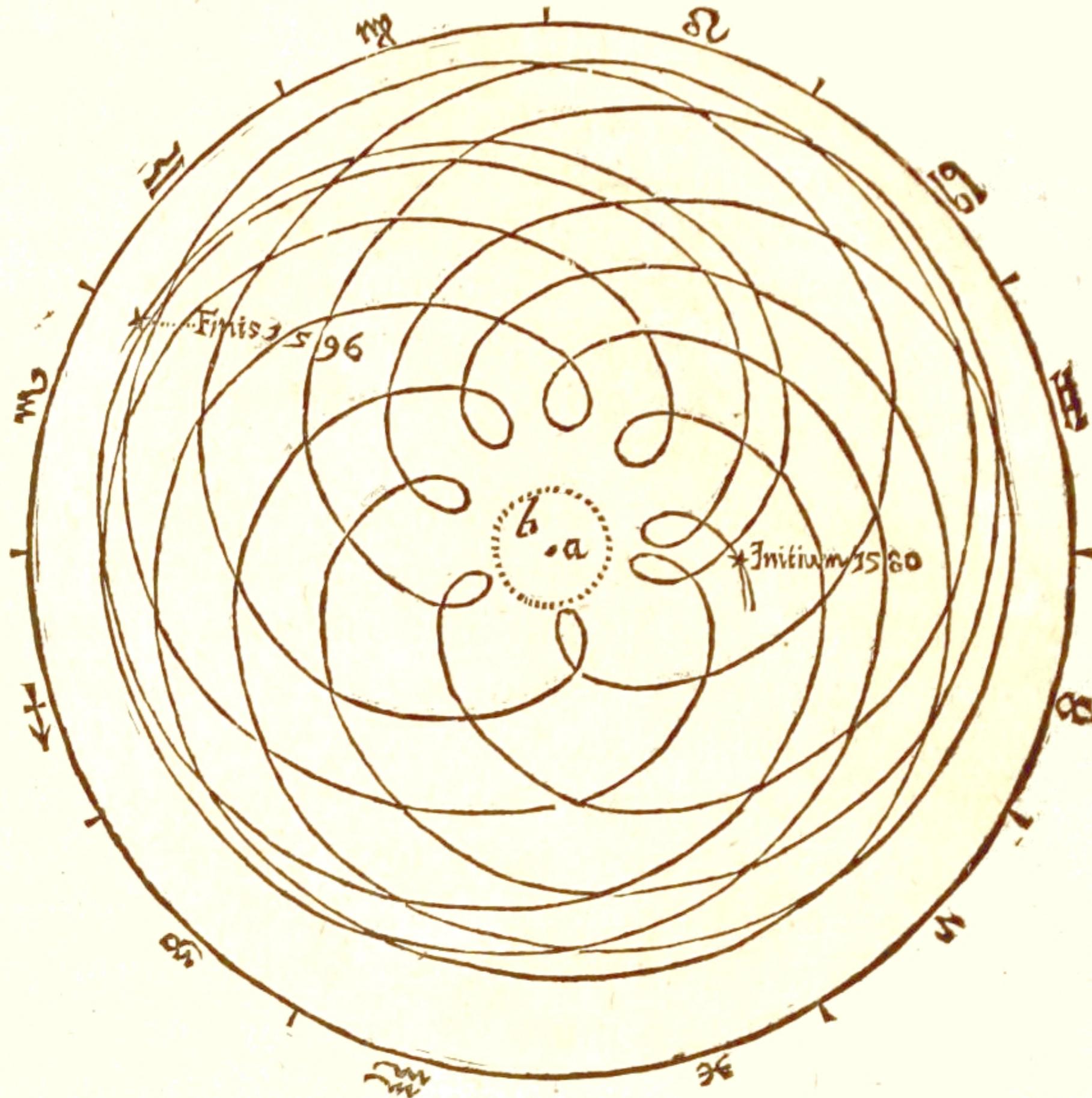
BUT: CENTRAL STEPS  
IN SCIENCE ARE  
STILL TAKEN BY HUMANS...

CENTRAL EXAMPLE:  
MODEL BUILDING

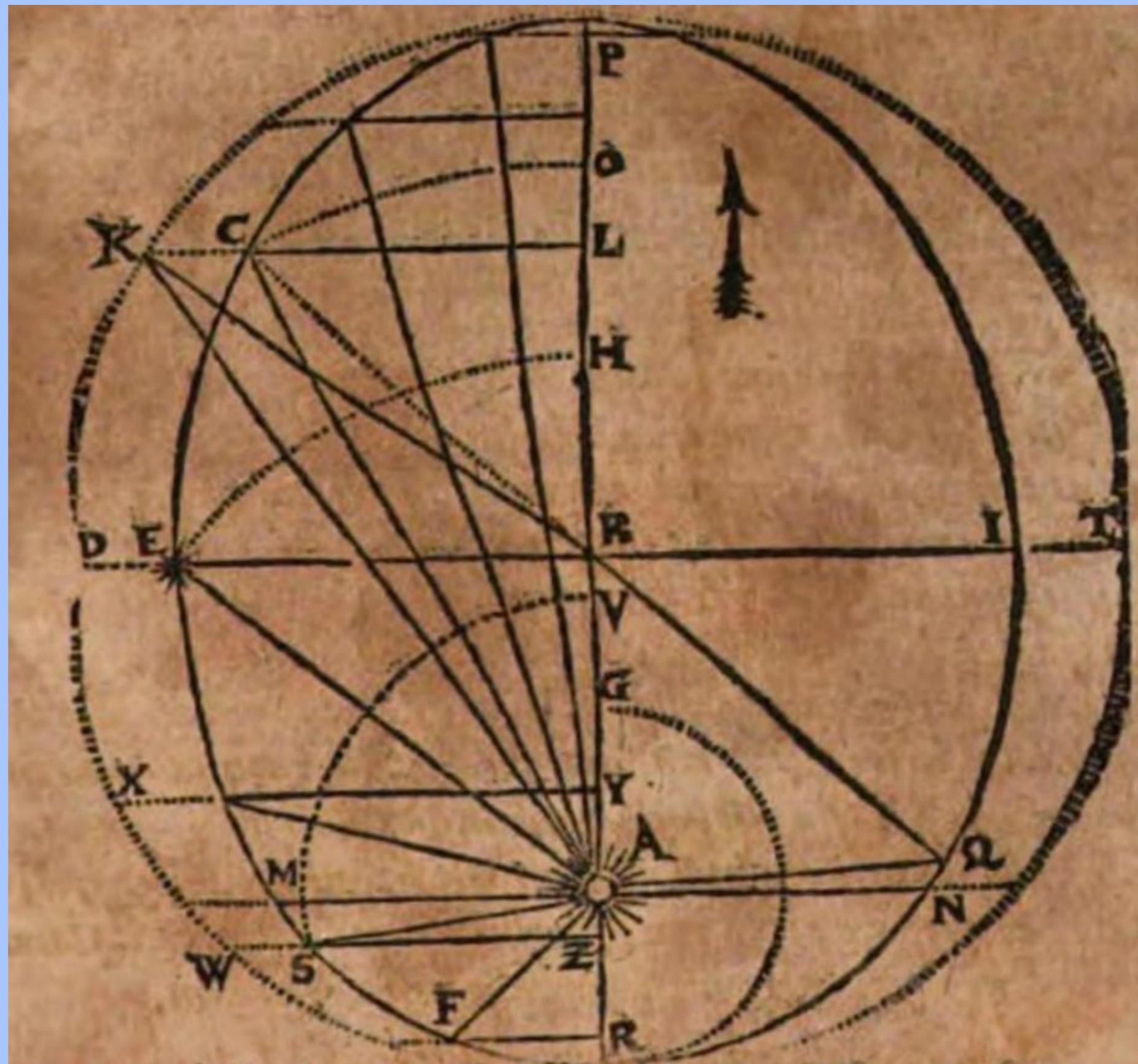


WIKIMEDIA COMMONS/ANDY

DE MOTIB. STELLÆ MARTIS



EPITOME ASTRONOMICAE COPERNICANAЕ  
KEPLER, 1635



Variis tant modi, sed compendiosissimus est, qui vi-  
D additur



BUILD A MODEL, FROM OBSERVATIONS

- ... FROM FEW OBSERVATIONS (DATA EFFICIENCY)
- ... SUCH THAT IT LIKELY GENERALIZES WELL
- ... POSSIBLY USING ANALOGIES
- ... FOCUS ON MOST RELEVANT FEATURES
- ... FOCUS ON MOST PREDICTABLE FEATURES

TEST THAT MODEL VIA WELL-CHOSEN NEW OBSERVATIONS  
... IN WELL-CHOSEN EXPERIMENTS

& EXPLORE ITS CONSEQUENCES IN  
HYPOTHETICAL SITUATIONS

## MAKE EFFICIENT PREDICTIONS

... DEAL WITH LARGE NUMBERS  
OF VARIABLES

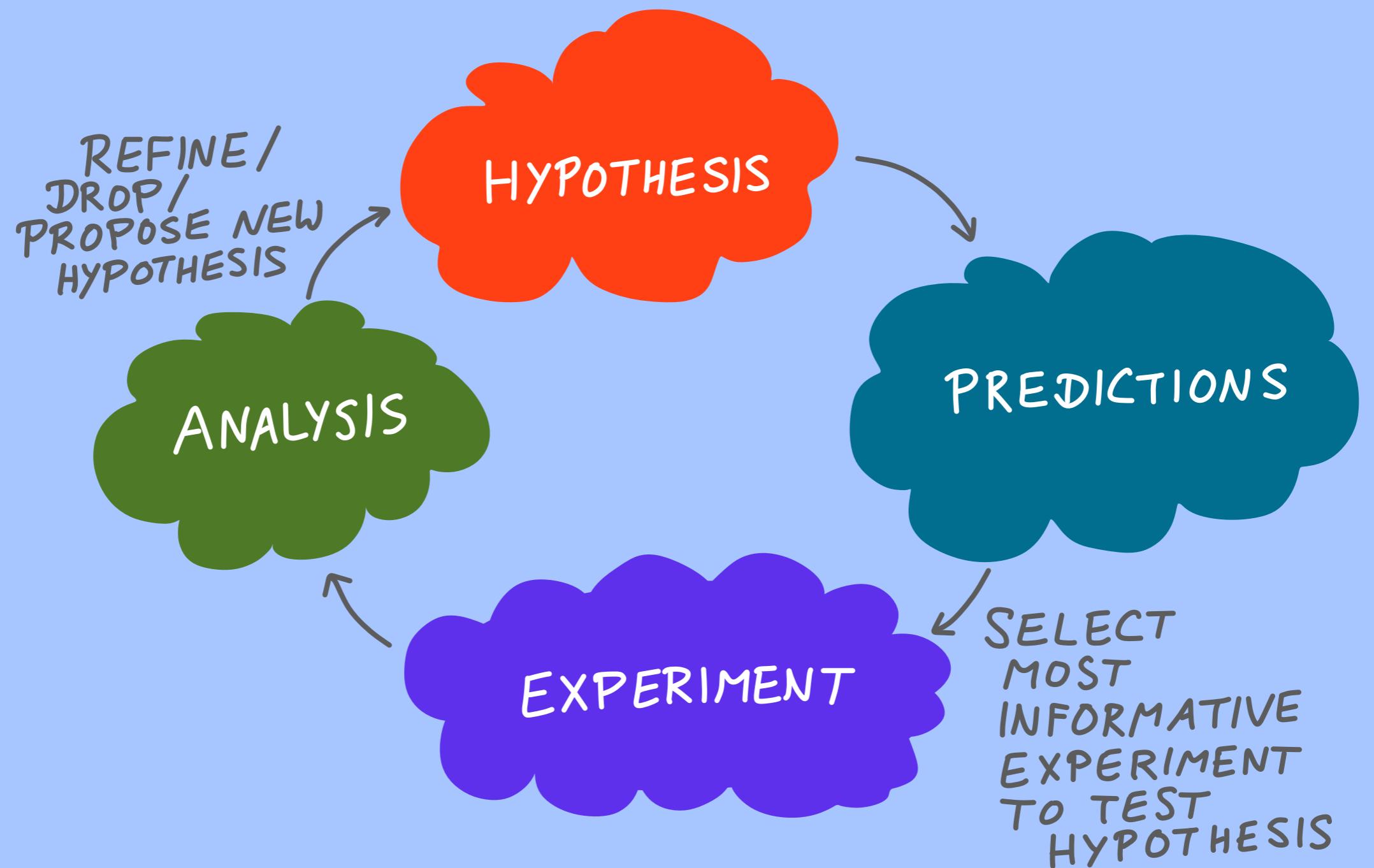
... INVENT APPROXIMATIONS

... INVENT EFFECTIVE MODELS

(AT LARGER SCALES, EVEN  
IF MICROSCOPIC MODEL KNOWN)

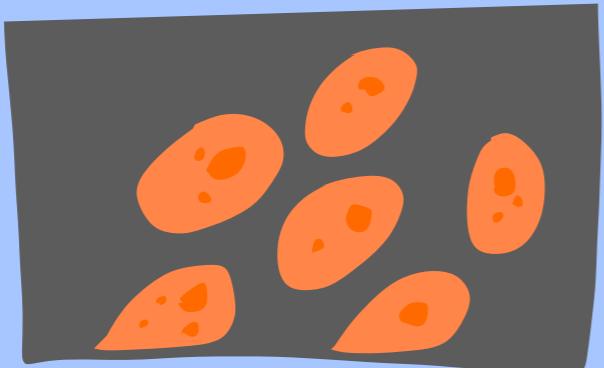
## SOME QUESTIONS:

- SEVERAL APPROXIMATELY VALID MODELS  
→ HOW TO CHOOSE? ("OCCAM'S RAZOR"?)
- ROLE OF ANALYTICAL EXPRESSIONS?
- HOW DOES THE COMPUTER UNDERSTAND  
WHICH INSIGHTS ARE ALREADY KNOWN?
- HOW CAN THE COMPUTER EFFECTIVELY  
COMMUNICATE ITS INSIGHTS?

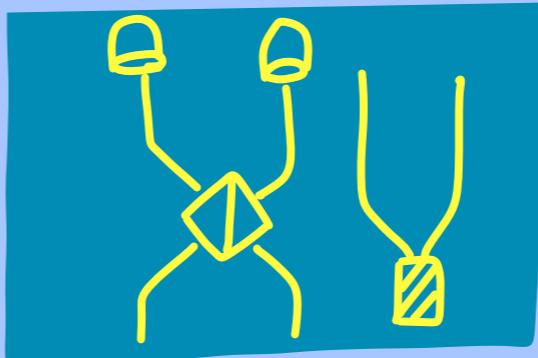


"INSIGHTS" GENERATED TO A  
CONSIDERABLE DEGREE BY A COMPUTER  
→ "ARTIFICIAL SCIENTIFIC DISCOVERY"

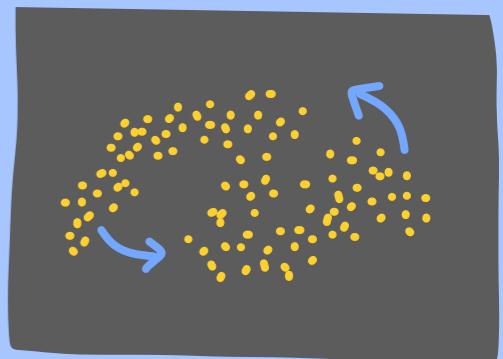
SOME EXAMPLES:



"ROBOT SCIENTISTS"  
FOR FUNCTIONAL  
GENOMICS & DRUG  
SCREENING



INVENTING NEW  
QUANTUM OPTICS  
EXPERIMENTS



PROTEIN FOLDING

(REFERENCES: SEE COURSE WEBSITE)

LONG-TERM GOALS:

- MORE GENERAL
- REDUCE HUMAN INPUT

GENERAL ARTIFICIAL  
SCIENTIFIC DISCOVERY  
SHOULD BE EASIER  
THAN GENERAL ARTIFICIAL  
INTELLIGENCE!

- QUANTITATIVE & LOGICAL  
REASONING FORMS PART  
OF IT → "EASY" FOR COMPUTERS
- PROMISE: DIRECT INTERFACE  
TO SIMULATIONS &  
EXPERIMENTS

1.3

## THIS LECTURE SERIES

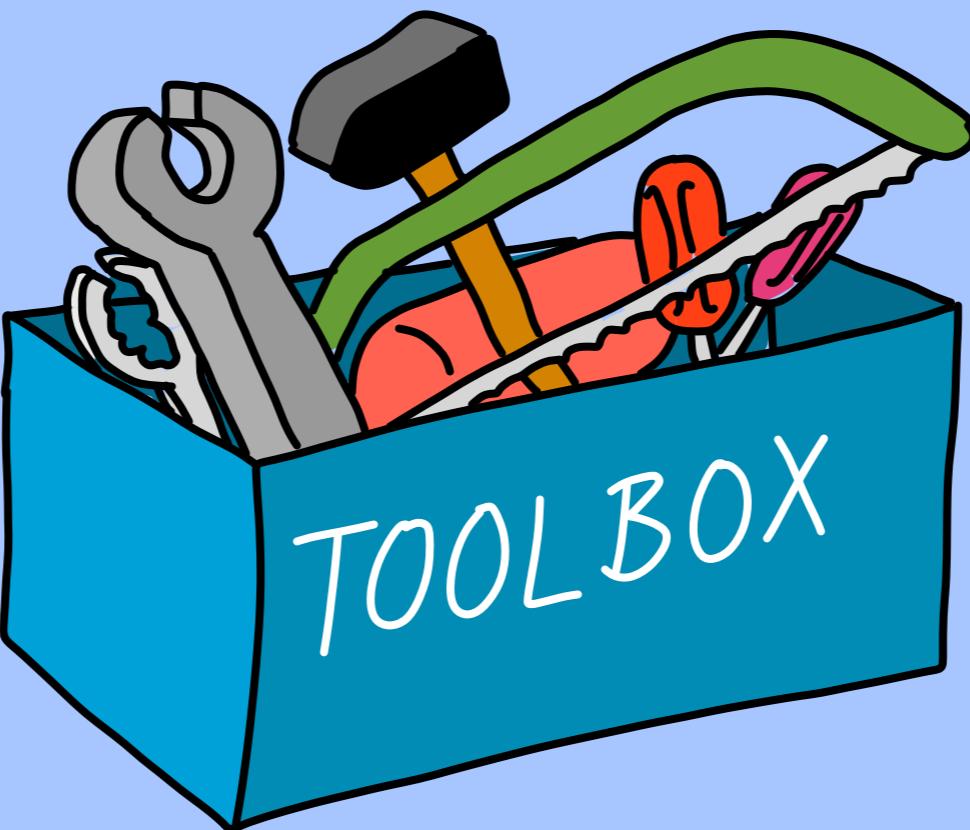
ARTIFICIAL  
NEURAL  
NETWORKS

BAYES

INFORMATION  
THEORY

REPRESENTATION  
LEARNING

ADVANCED  
NN STRUCTURES

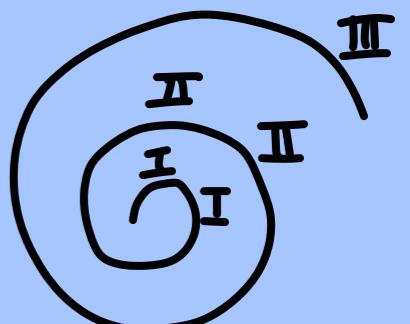


DISCOVERING  
STRATEGIES

ADAPTIVE  
OBSERVATIONS

MEASURING  
COMPLEXITY

LEARNING  
PROBABILITY  
DISTRIBUTIONS

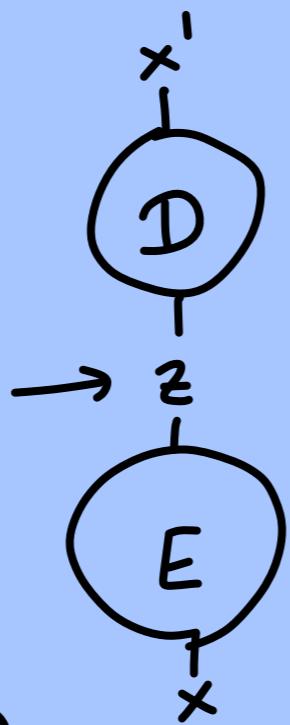
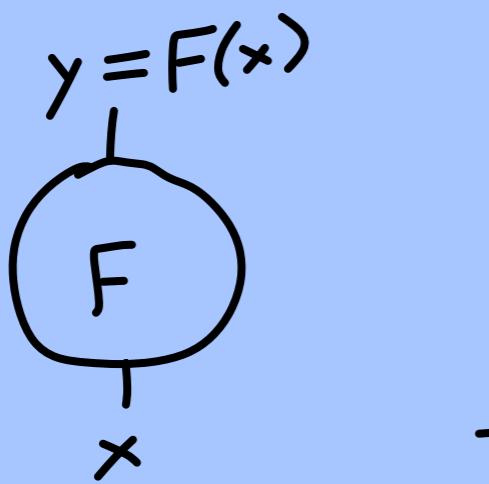


2.

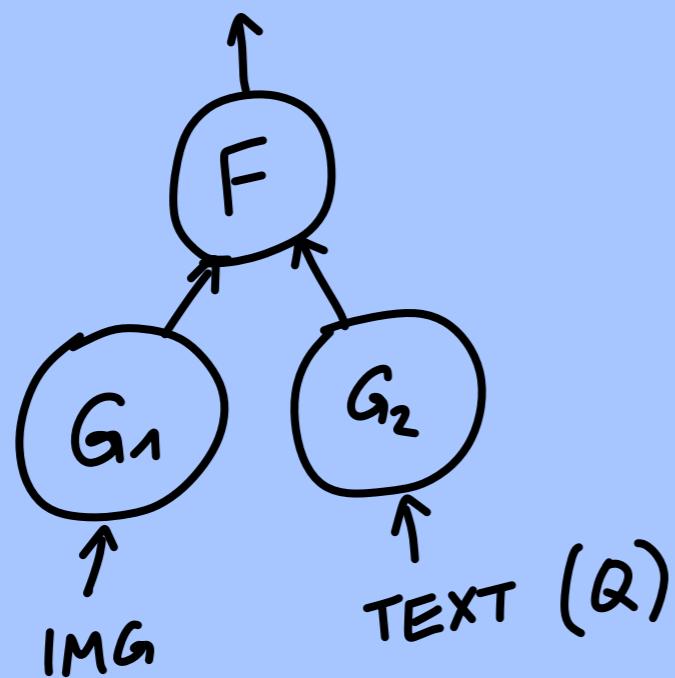
## ARTIFICIAL NEURAL NETWORKS I: BASICS

2.1

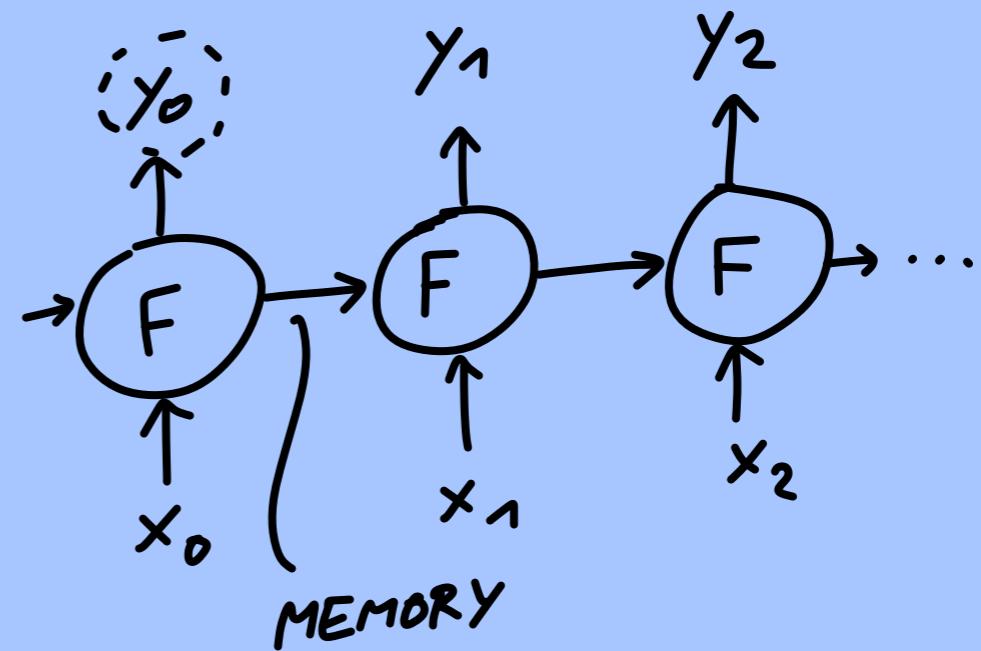
MOTIVATION: LEARNABLE  
FUNCTIONS AS BUILDING BLOCKS  
FOR MACHINE LEARNING



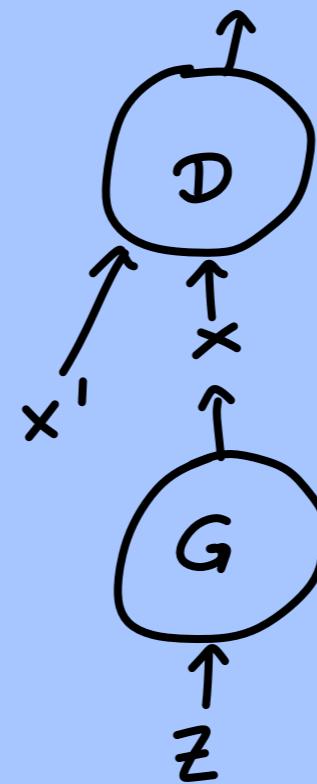
TEXT (A)



TEXT (Q)



MEMORY



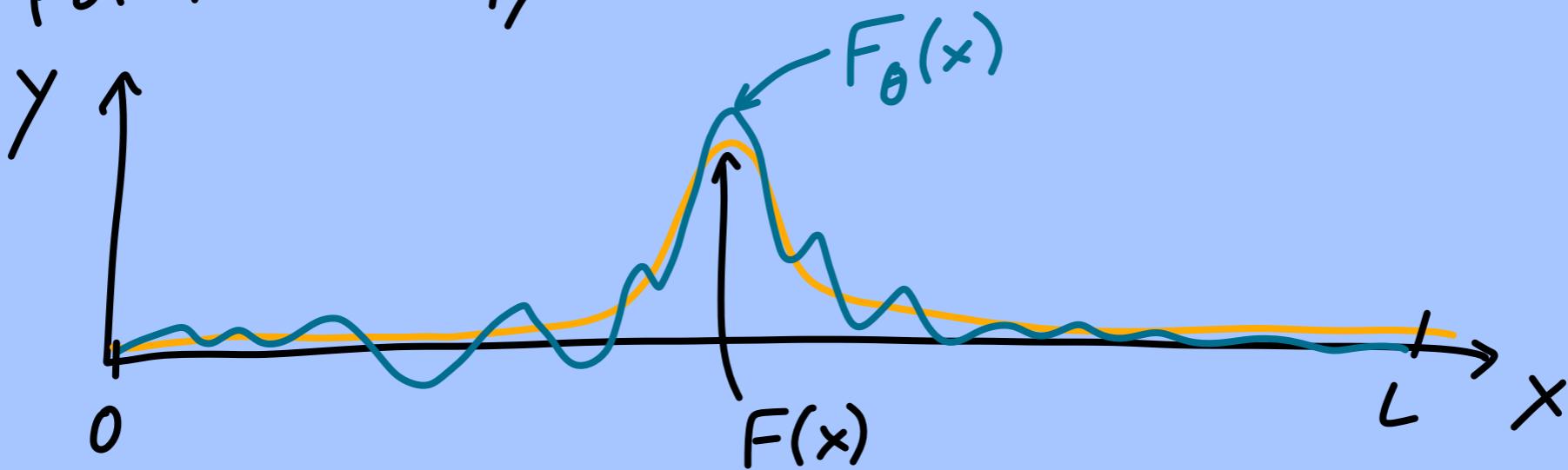
2.2

## FUNCTION APPROXIMATION

$$y = F_{\theta}(x)$$

INPUT  $\in \mathbb{R}^{d_x}$       !  
OUTPUT       $\approx F(x)$   
PARAMETERS  
 $\in \mathbb{R}^{d_\theta}$       |  
 $\mathbb{R}^{d_y}$       TARGET

FOR NOW:  $x, y \in \mathbb{R}^1$



MINIMIZE  $\int_0^L (F_\theta(x) - F(x))^2 dx$

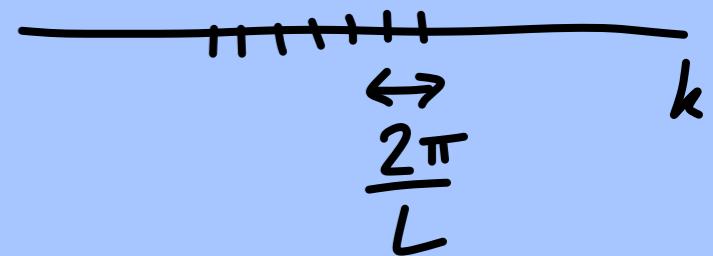
OPTIONS FOR  $F_\theta$ ?

POLYNOMIAL

$$F_\theta(x) = \sum_{n=0}^N \theta_n x^n$$

FOURIER

$$F_\theta(x) = \sum_k \theta_k e^{ikx}$$
$$|k| \leq K$$



GENERAL :

LINEAR SUPERPOS.  
OF NONLIN. TERMS

$$F_\theta(x) = \sum_n \theta_n \phi_n(x)$$

MINIMIZ. OF SQ. DEV.  
→ LIN. ALG. ✓

$\phi_n$  ORTHONORM. BASIS

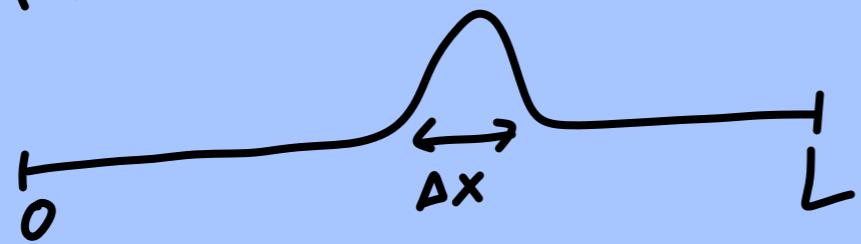
$$\Rightarrow \theta_n = \langle \phi_n | F \rangle$$

# TERMS  $\rightarrow \infty \Rightarrow F_\theta(x) \rightarrow F(x) \checkmark$

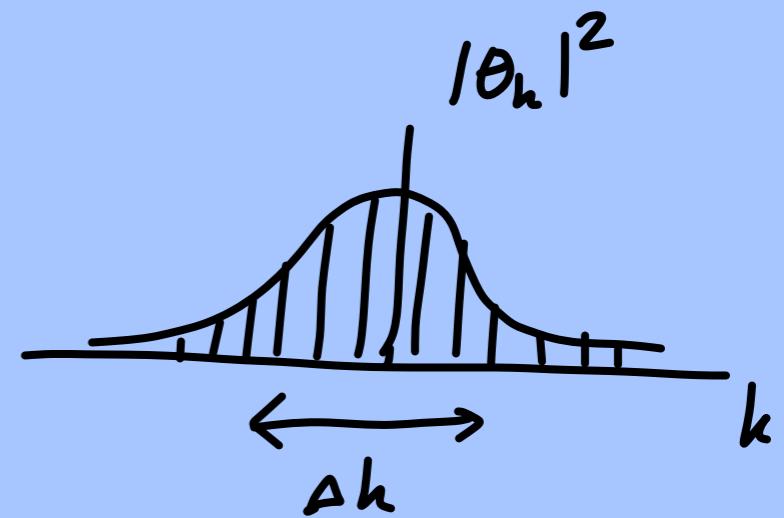
"EXPRESSIVENESS"  $\Rightarrow$  APPROX.  
ARBITR. FCTS.  $\checkmark$

BUT: EFFICIENT?

FOURIER:

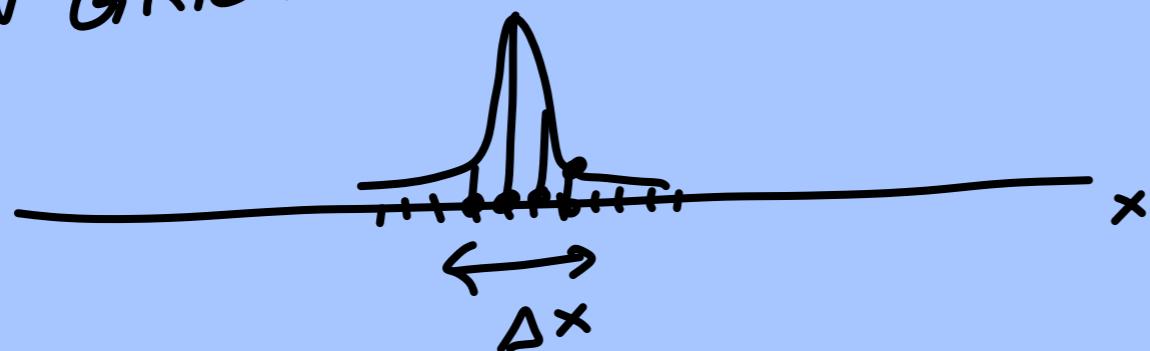


$$\Delta k \sim \frac{1}{\Delta x}$$

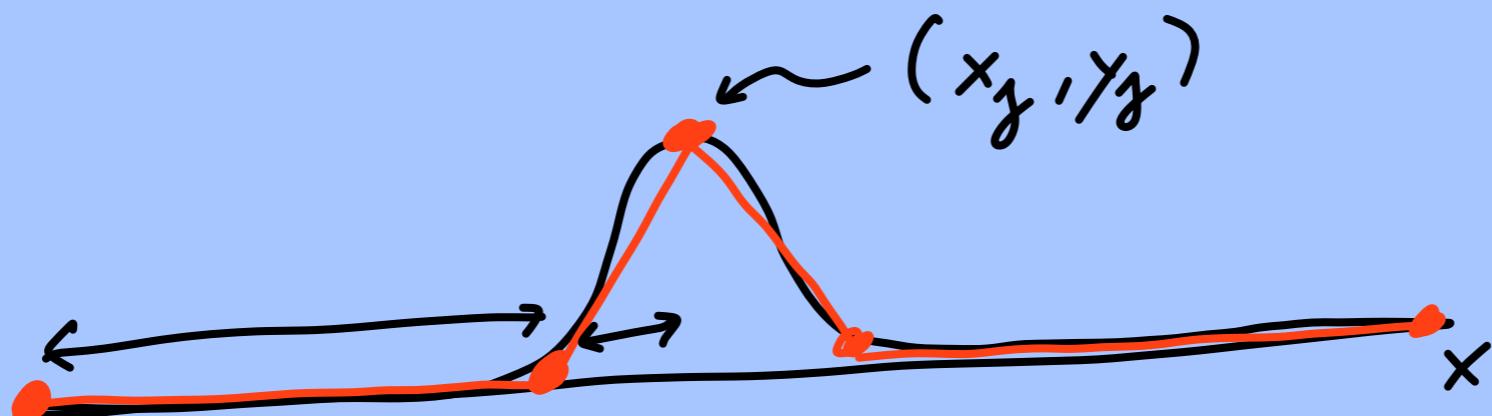


NEED  $\frac{L}{\Delta x}$  COEFF.!

ON GRID:



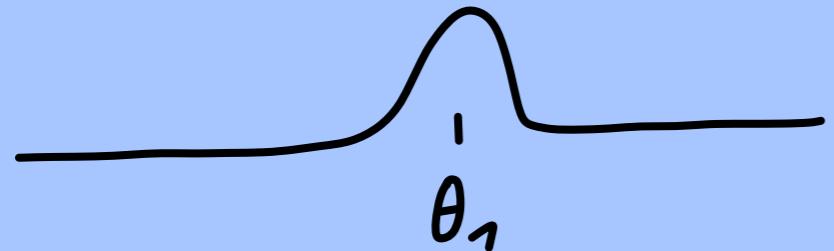
NEED  $\frac{\Delta x}{\alpha}$  GRID PTS.  
LATTICE SPACING



ADAPTIVE GRID!

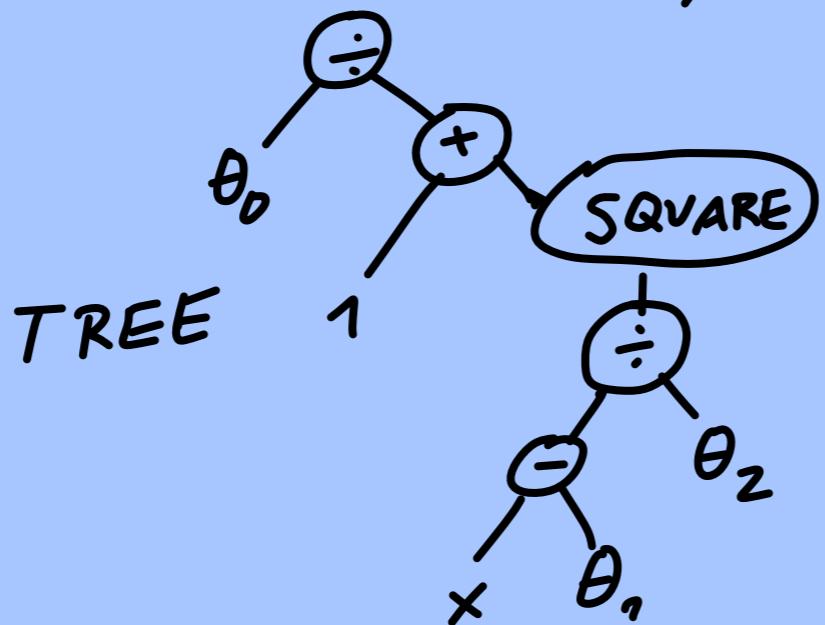
⇒ N.N. WILL DO THIS!

ANOTHER OPTION:  
ANSATZ



$$F_\theta(x) = \frac{\theta_0}{1 + \left(\frac{x - \theta_1}{\theta_2}\right)^2}$$

LESSON: HIERARCHICAL/RECURSIVE STRUCTURE



⇒ GENERAL IDEA:

$$F_\theta(x) = F_{\theta_3}^{(3)} \left( F_{\theta_2}^{(2)} \left( F_{\underline{\theta_1}}^{(1)}(x) \right) \right)$$

# WISHLIST

SIMPLE BUILDING BLOCKS

GENERALITY: EASILY SCALED UP

CAN APPROXIMATE ANY FUNCTION

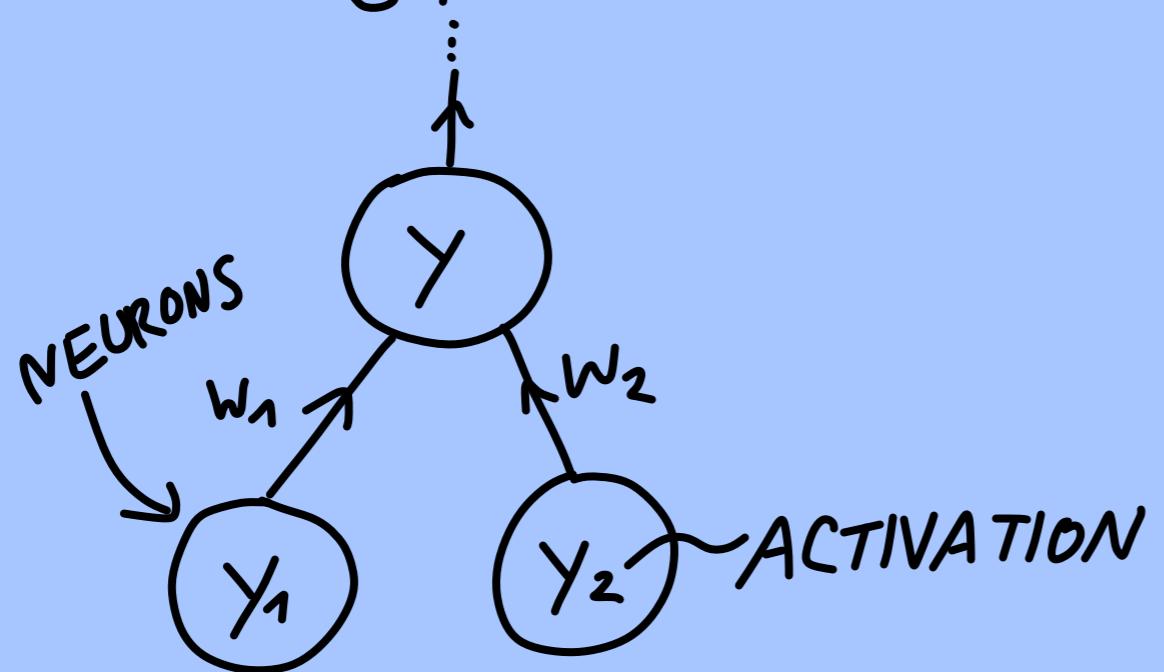
EFFICIENT APPROXIMATION

RECURSIVE

WORKS WELL FOR HIGH DIMENSIONS

CAN LEARN EFFICIENTLY

## 2.3

ARTIFICIAL NEURAL NETWORKS:  
STRUCTURE & EXPRESSIVITY

## 1. LINEAR STEP

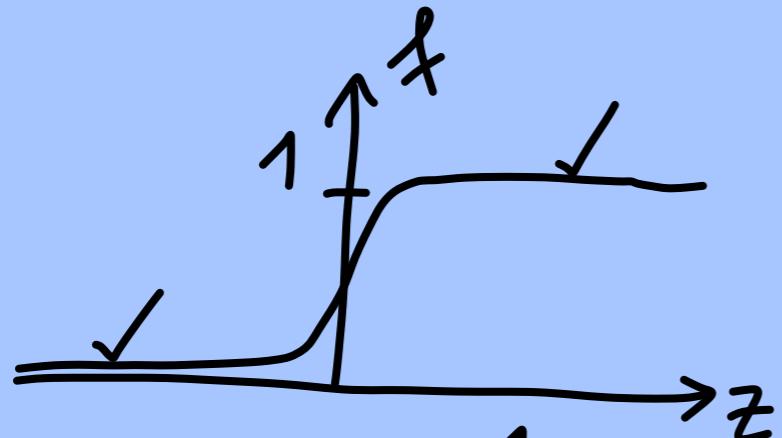
$$z = w_1 y_1 + w_2 y_2 + b$$

$\swarrow$        $\swarrow$        $\searrow$   
WEIGHTS           BIAS

## 2. NONLINEAR STEP

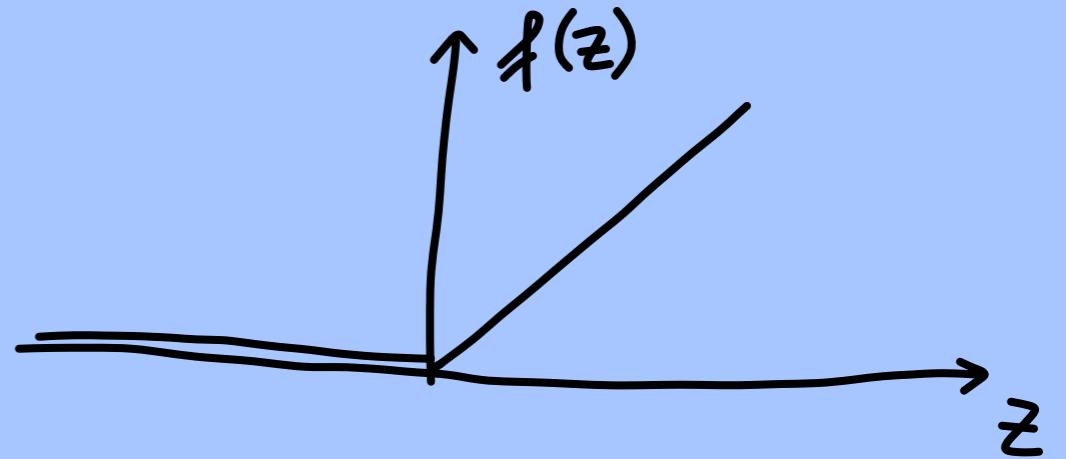
$$y = f(z)$$

$\swarrow$   
ACTIVATION  
FUNCTION



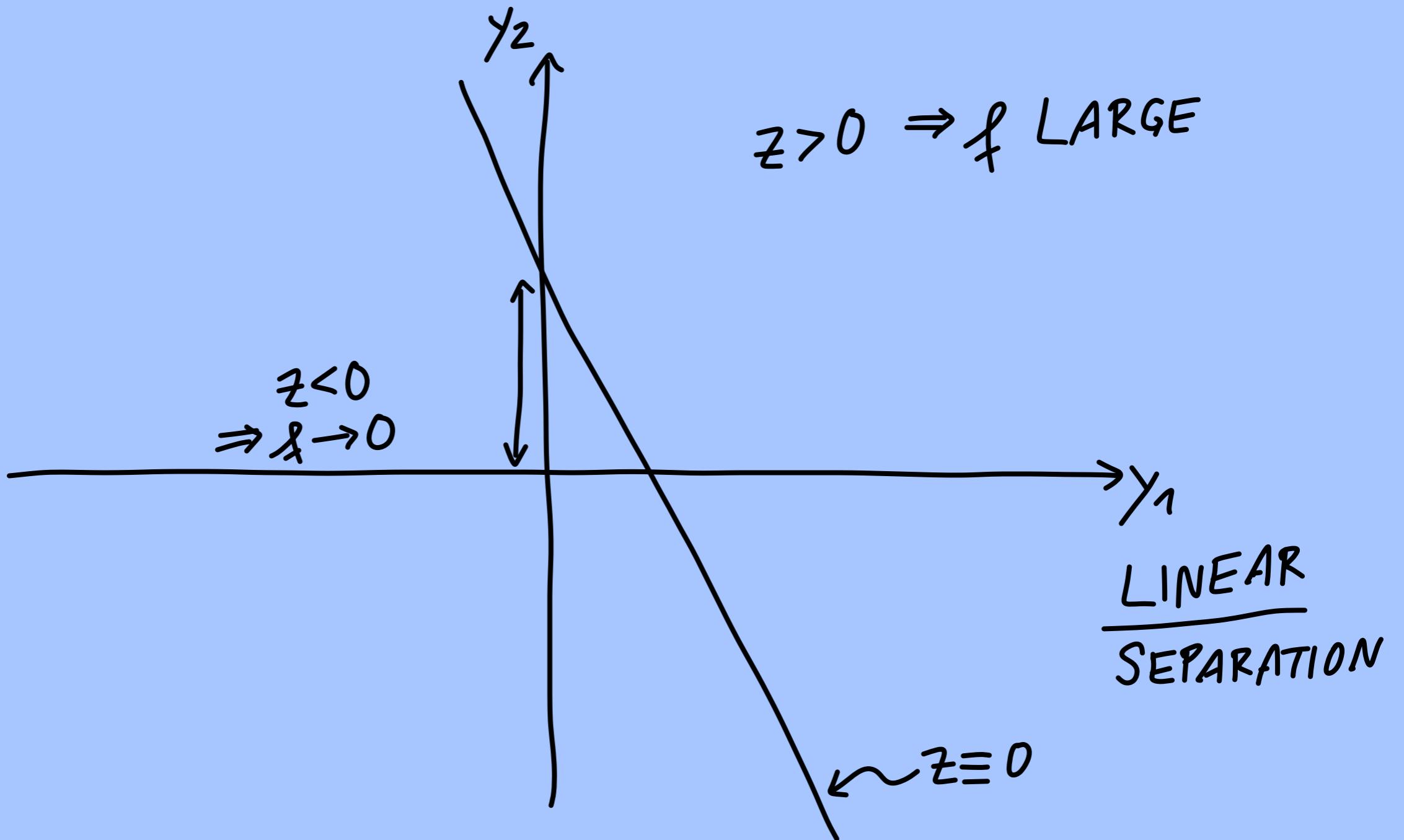
$$f(z) = \frac{1}{1 + e^{-z}}$$

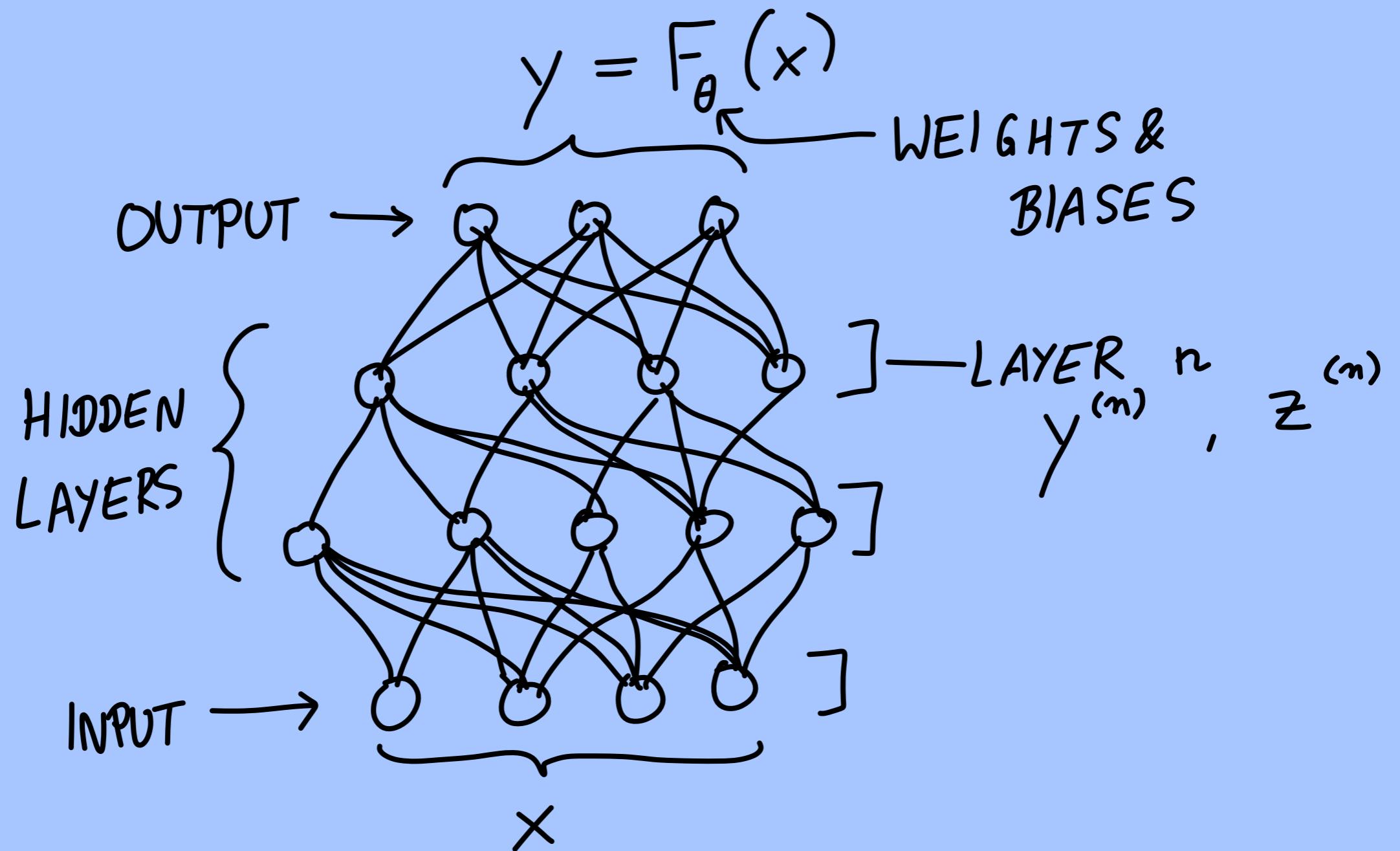
SIGMOID



$$f(z) = \begin{cases} z & \text{FOR } z \geq 0 \\ 0 & \text{FOR } z < 0 \end{cases}$$

RECTIFIED LINEAR UNIT  
"RELU"





FOR ALL LAYERS:

1.  $z^{(n)} = \underbrace{W^{(n)} y^{(n-1)}}_{\text{MATRIX } d_n \times d_{n-1}} + b^{(n)}$

$\mathbb{R}^{d_n}$   $\mathbb{R}^{d_{n-1}}$   $\mathbb{R}^{d_n}$

$b^{(n)}$   $\leftarrow$  # NEURONS IN LAYER  $n-1$

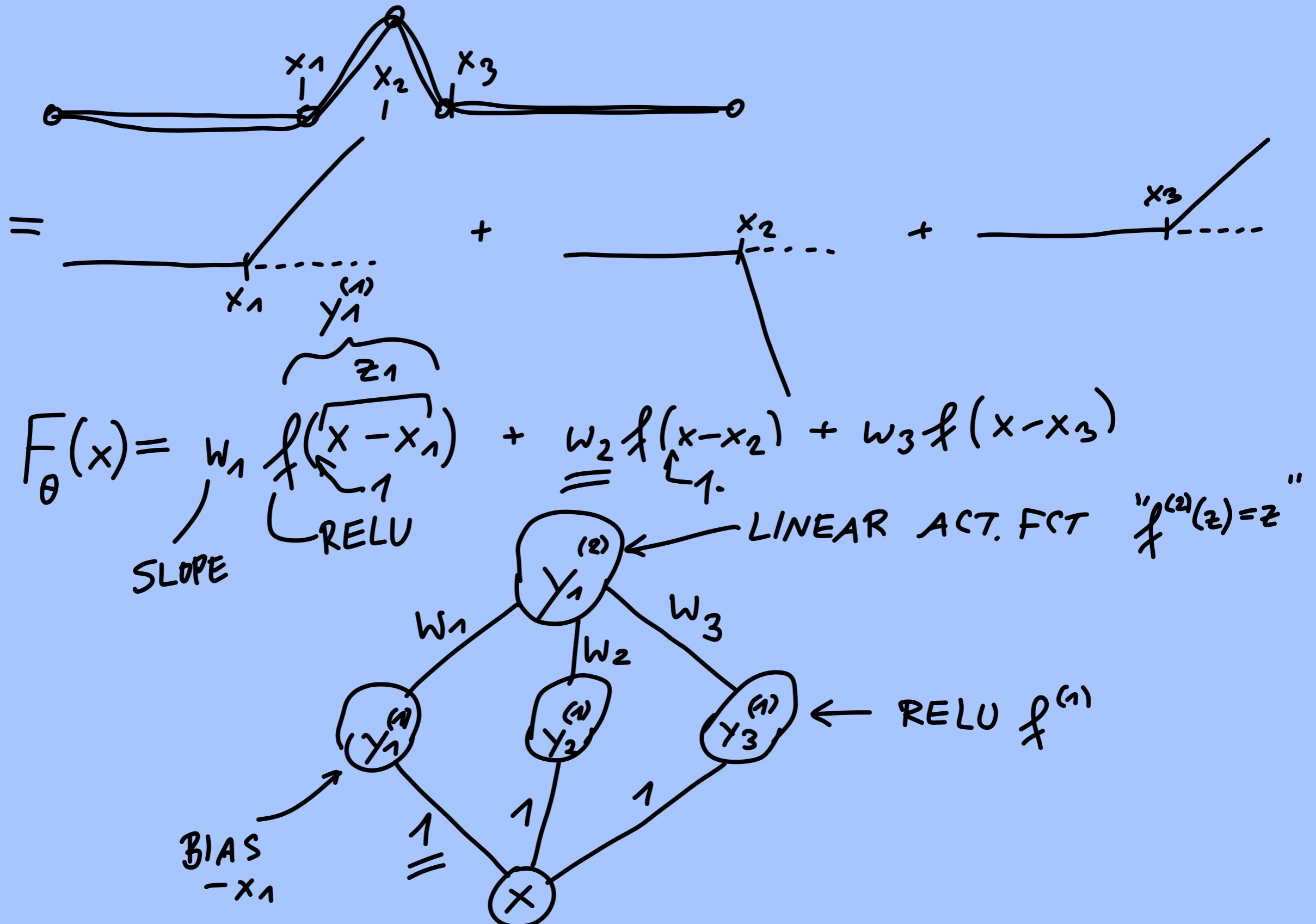
2.  $y^{(n)} = f(z^{(n)})$

$\hookrightarrow$  "POINTWISE" APPLICATION

$$y_j^{(n)} = f(z_j^{(n)}) \quad \forall j$$

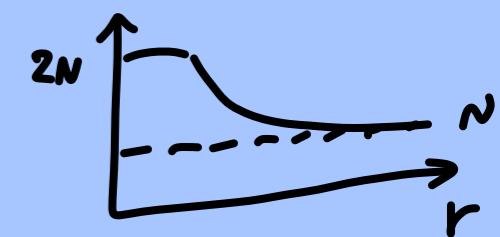
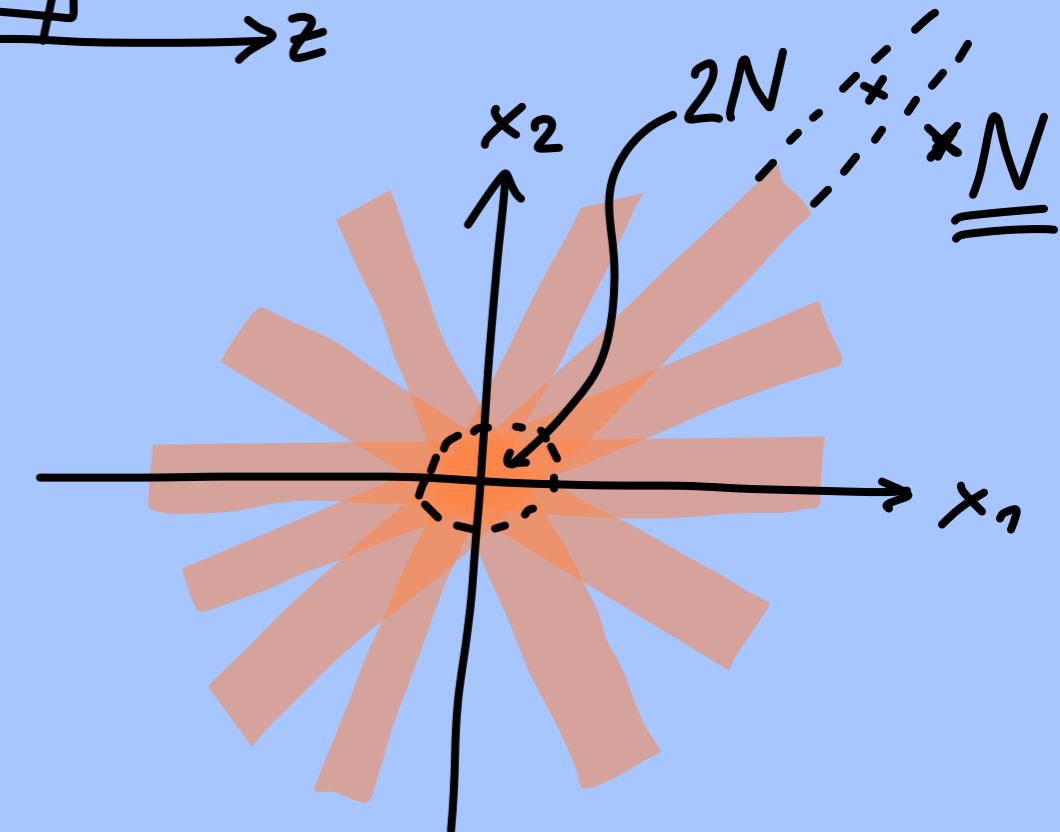
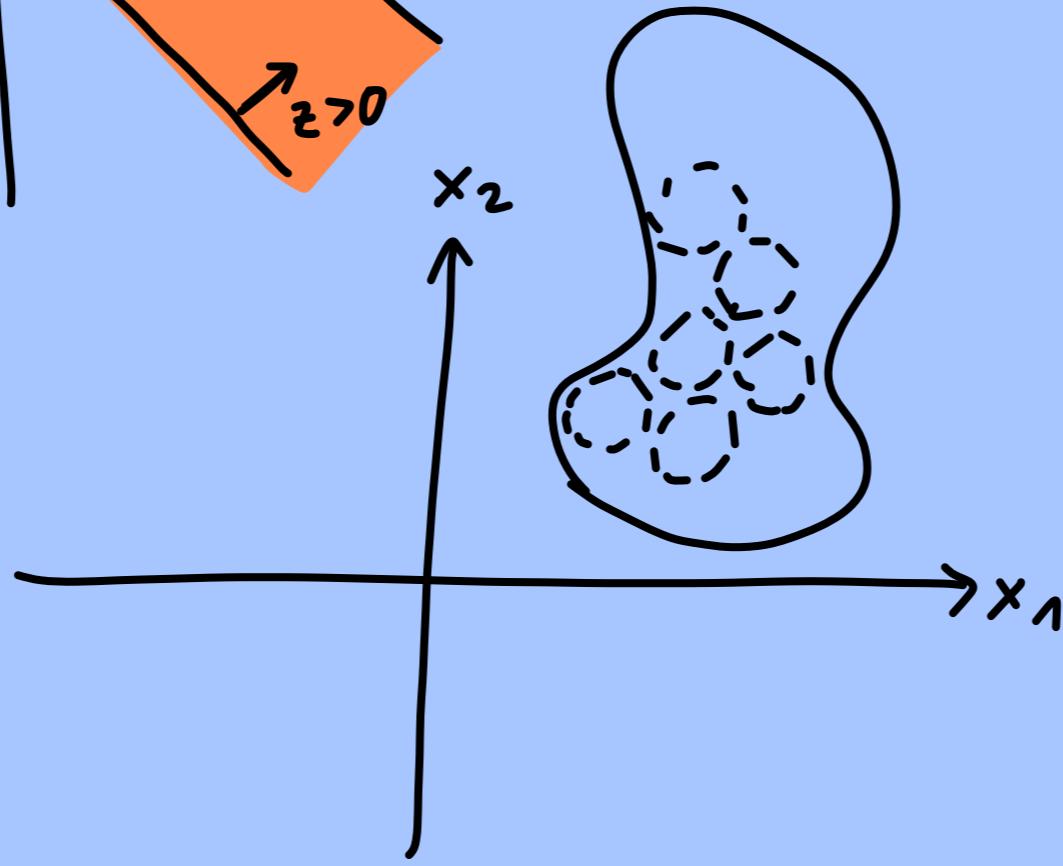
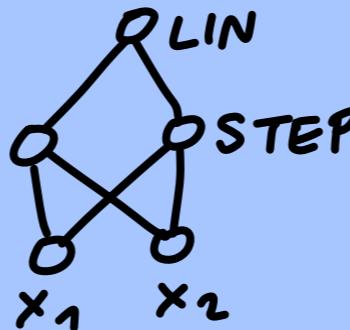
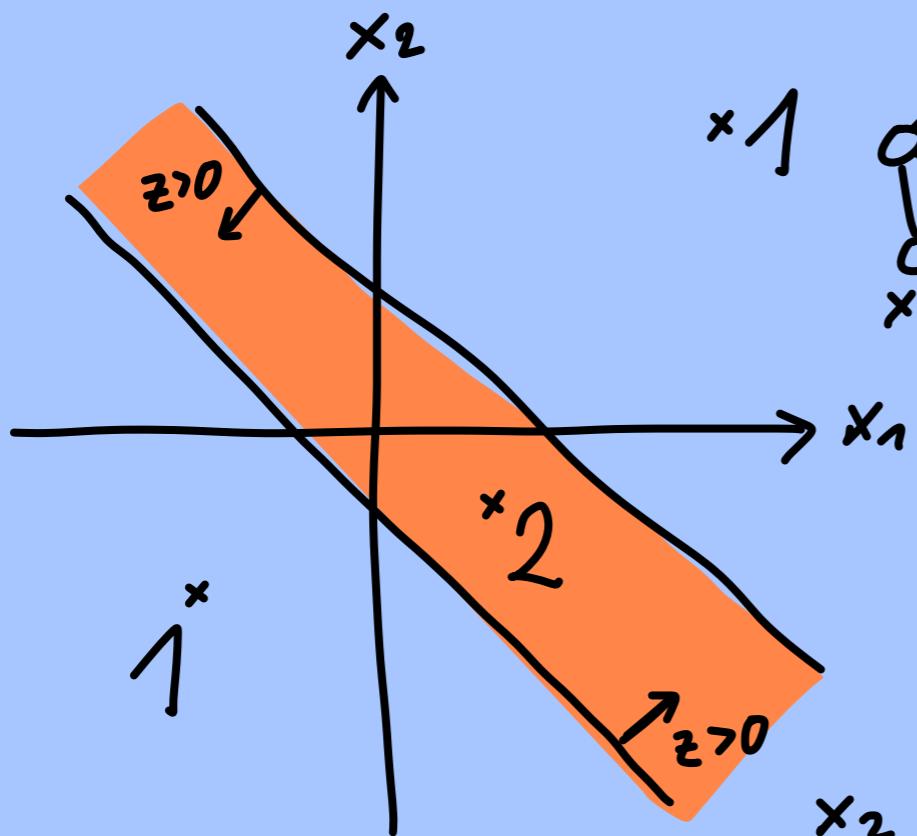
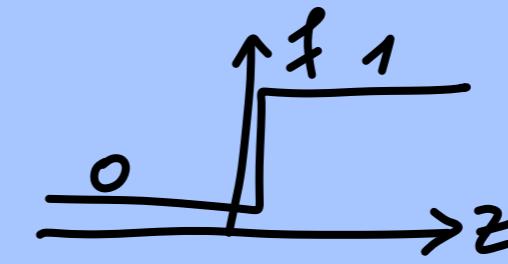
$$\theta_n = (w^{(n)}, b)$$

$$F_{\theta_n}^{(n)}(y^{(n-1)}) = f(w^{(n)}y^{(n-1)} + b^{(n)})$$



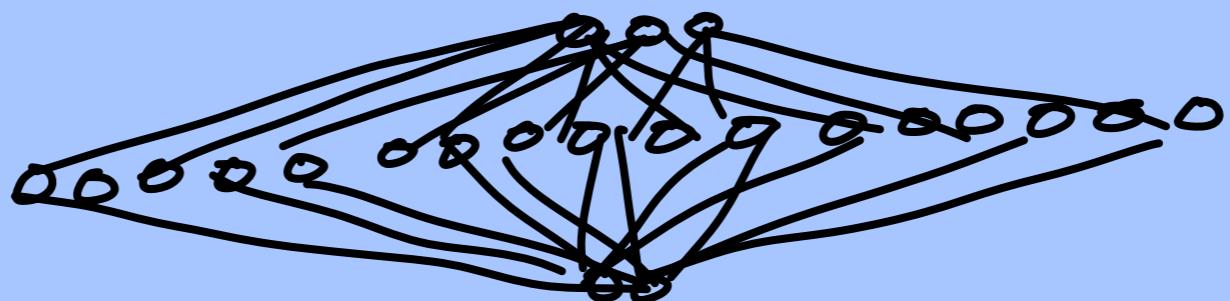
2D?

$$f(z) = H(z)$$



CYBENKO (1987)

'ANY' (SUFFICIENTLY SMOOTH) FUNCTION  
 $F: \mathbb{R}^k \rightarrow \mathbb{R}^d$  CAN  
BE APPROXIMATED "ARBITRARILY WELL"  
BY A NEURAL NETWORK WITH  
A SINGLE HIDDEN LAYER (WITH SIGMOID)  
WITH  
(SUFFIC. MANY NEURONS)



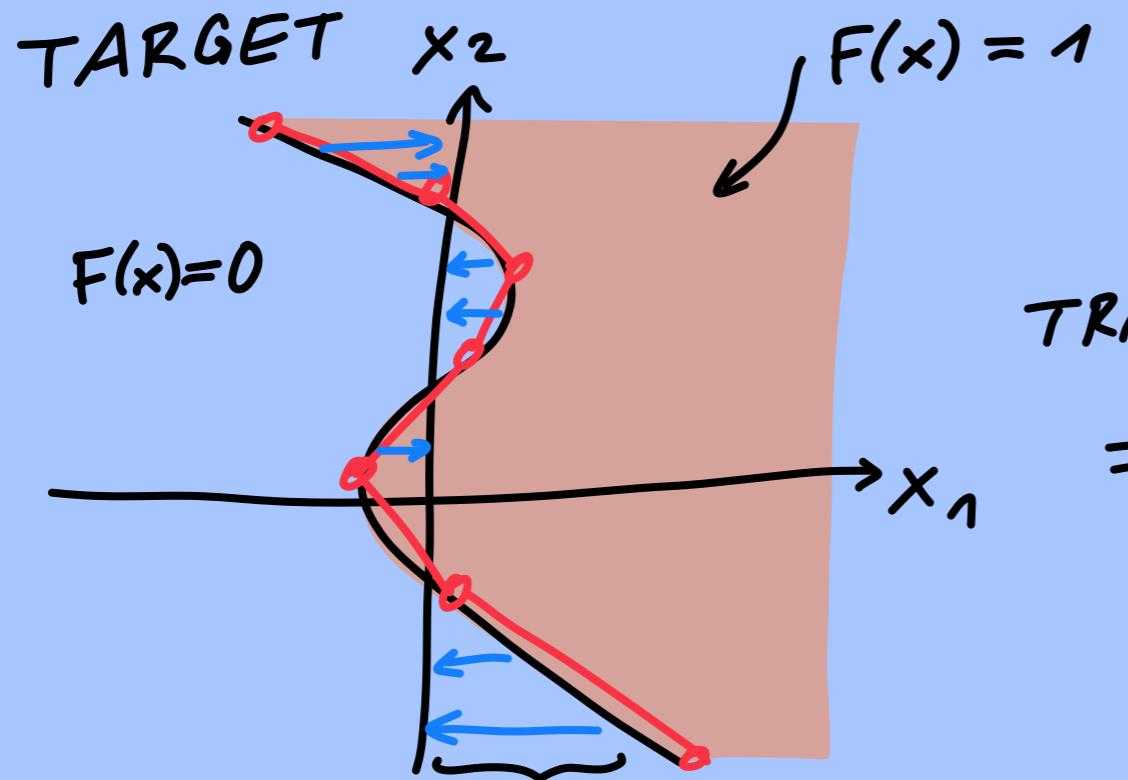
BUT: NOT NECESSARILY EFFICIENT!



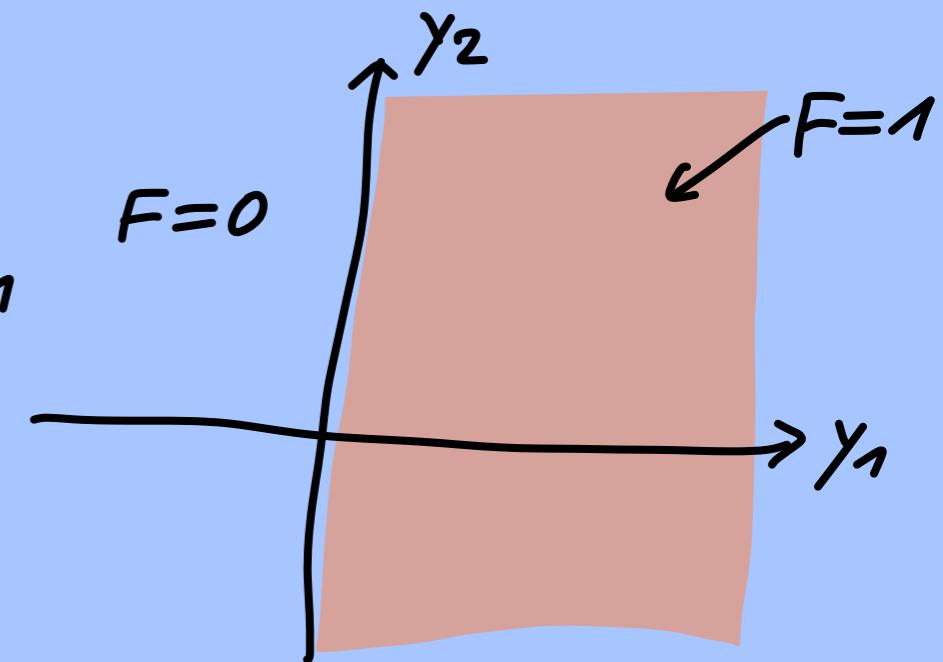
NEED MORE HIDDEN  
LAYERS



DEEP NEURAL NETWORKS !

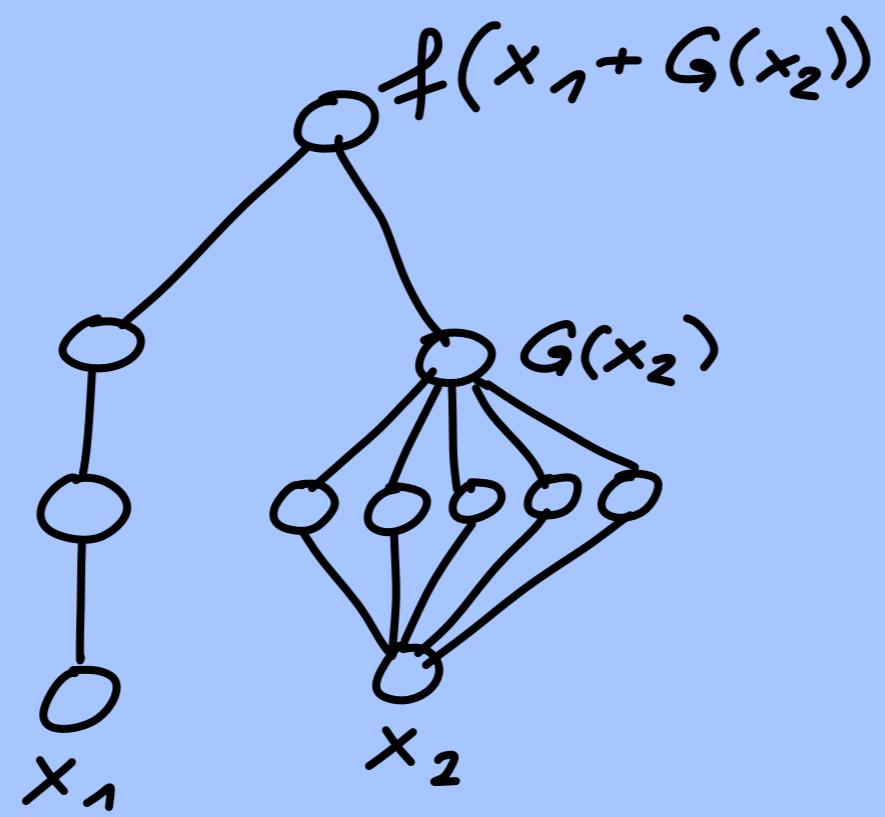


TRANSFORM



$$y_2 = x_2$$

$$y_1 = x_1 + G(x_2)$$



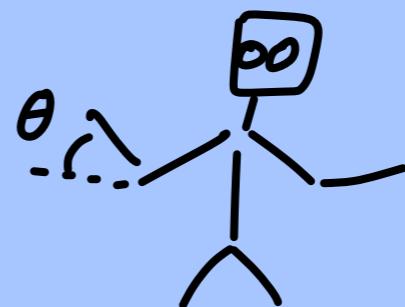
"COORDINATE TRANSFORMATION" VIEWPOINT

LOWER LAYERS "PRE PROCESS" INPUT  
→ EASIER FOR HIGHER  
LAYERS!

WHY "DEEP" MAY BE BETTER

- ALGORITHMS : SEQUENTIAL PROCESSING
- PHYSICS/DATA GENERATION :  
MANY STEPS
- HIGH-DIM. "COMPLEX" DATA

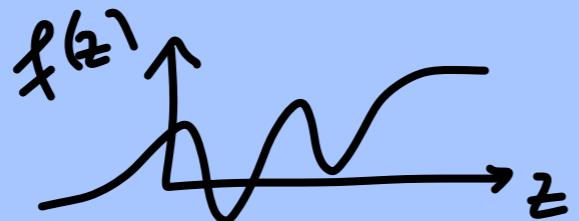
= LOW-DIM. MANIFOLD  
OFTEN



- BIOLOGY

# FAQ NEURAL NETWORKS — STRUCTURE

NON-MONOTONIC ACTIVATION FUNCTIONS?



MULTIPLE ACTIV. FUNCTIONS IN A SINGLE LAYER?

$$\underbrace{00000}_{f(z) = Z(z)} \quad \underbrace{0000}_{f(z) = \sin z} \quad \underbrace{0000}_{f(z) = \frac{1}{1+z^2}}$$

$$\sin\left(\frac{1}{1+z^2}\right)$$

→ SYMBOLIC  
REGRESSION  
NETWORK

ACTIVATION FUNCTIONS THAT DO NOT ACT POINTWISE?

MOST IMPORTANT EXAMPLE: "SOFTMAX"

$$y_j^{(n)} = \frac{e^{z_j^{(n)}}}{\sum_k e^{z_k^{(n)}}} \quad \left. \begin{array}{l} \{ \geq 0 \\ \{ \text{NORMALIZATION} \end{array} \right.$$

$\sum_k y_k^{(n)} = 1$

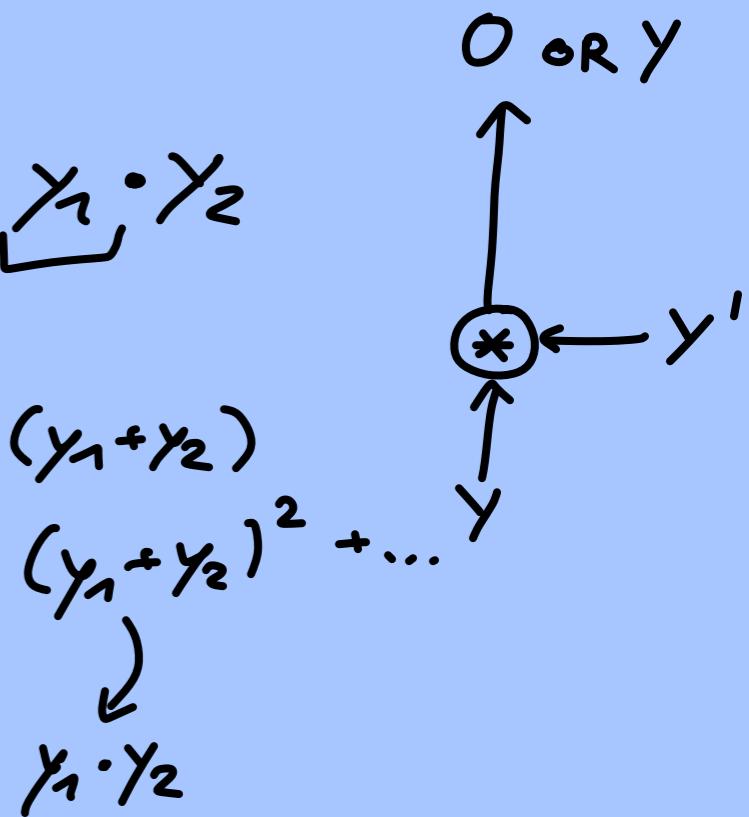
$0000 \quad z_j^{(n)}$

$\Rightarrow$  INTERPRET AS PROBAB. DISTRIBUTION

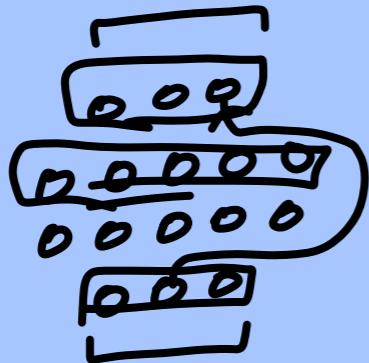
PRODUCT INSTEAD OF SUM?

$$\left( \sum_k w_{kj} y_k \right) \mapsto \text{YES: } y_1 \cdot y_2$$

$$\begin{aligned} f(y_1 + y_2) &= f(0) + f'(y_1 + y_2) \\ &\quad + \frac{f''}{2} (y_1 + y_2)^2 + \dots \end{aligned}$$



CONNECTIONS BETWEEN DISTANT LAYERS ?



"SKIP CONNECTIONS"  
(RESNETS, U-NETS)

INPUT / OUTPUT OF VARIABLE SIZE ?

NO

(EXCEPT USING  
SOME KIND OF SYMMETRY)

## 2.4

THE COST FUNCTION  $\equiv$  LOSS  
CENTRAL QUANTITY!

GOAL: MEASURE DEVIATION  
BETWEEN TRUERESULT  $y^{\text{true}(x)}$   
AND  $F_{\theta}(x)$

$$\mathcal{L}(\theta) = \left\langle \mathcal{L}(F_{\theta}(x), y^{\text{true}(x)}) \right\rangle_x$$

↓  
LOSS

↓  
SAMPLE-SPECIFIC LOSS  $\mathcal{L}$

↓  
NN RESULT

↓  
TRUE RESULT

→  
AVG OVER SAMPLES

$\langle \cdot \rangle_x$  = AVERAGE OVER ALL  
SAMPLES

↪ IDEALIZED :  $\langle \cdot \rangle_x = \frac{1}{L} \int_0^L \cdot dx$

↪ HYPOTHETICAL  
"ALL SAMPLES IN THE WORLD"

↪ ALL SAMPLES IN  
LARGE DATABASE

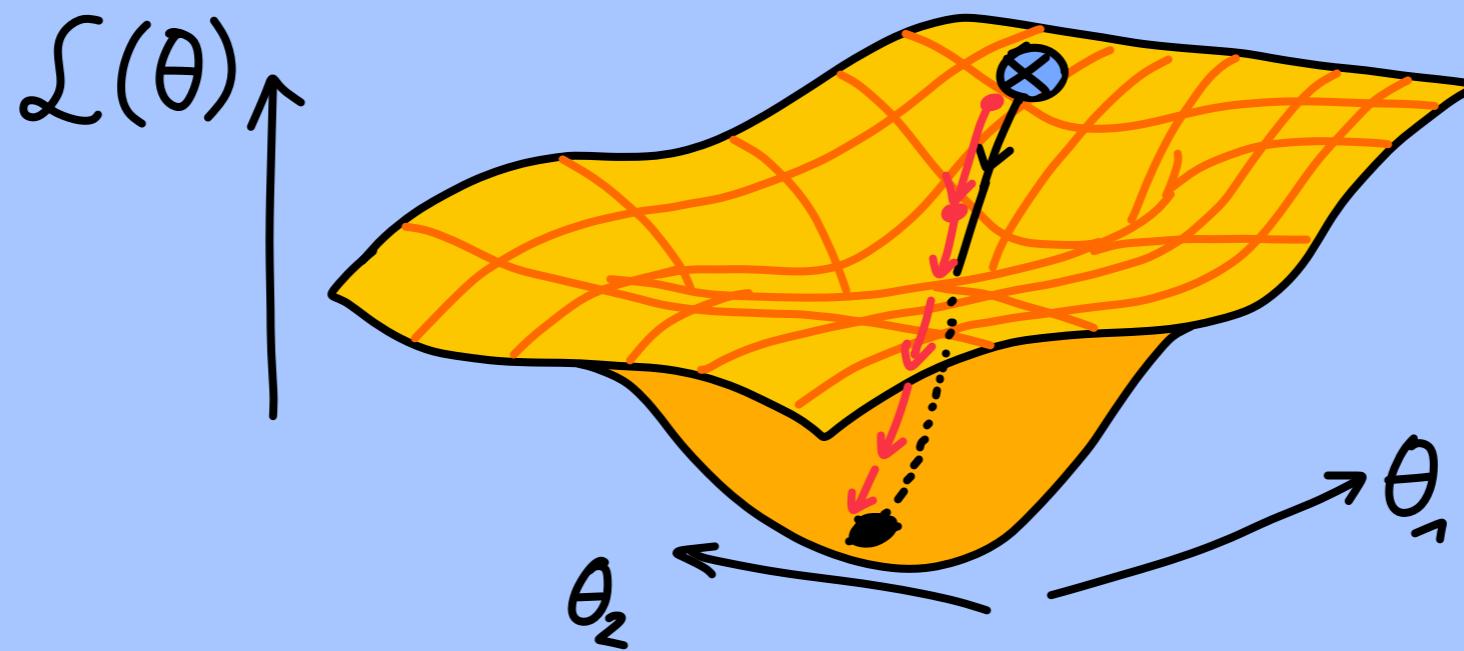
SIMPLEST EXAMPLE:

$$\begin{aligned} L(y^{NN}, y^{\text{true}}) &= (y^{NN} - y^{\text{true}})^2 \\ &= \|y^{NN} - y^{\text{true}}\|_2^2 \\ &= \sum_z (y_z^{NN} - y_z^{\text{true}})^2 \end{aligned}$$

2.5

## STOCHASTIC GRADIENT DESCENT

GOAL : MINIMIZE  $\mathcal{L}(\theta)$



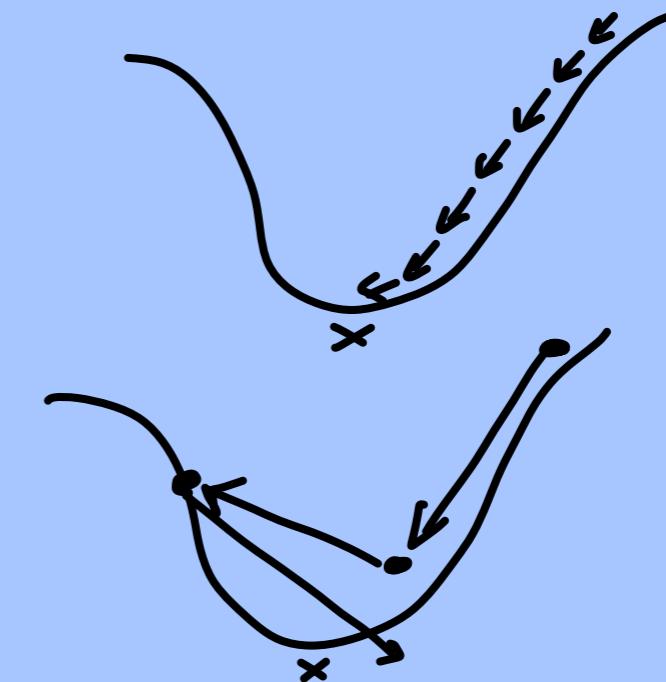
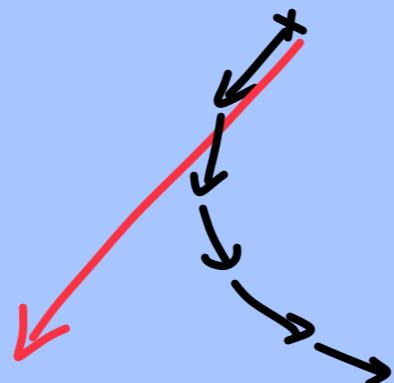
STEPWISE GRADIENT DESCENT:

$$\delta\theta = -\gamma \nabla_{\theta} \mathcal{L}(\theta)$$

$\underbrace{\gamma}_{\text{"LEARNING RATE"}}$

$\gamma$  SMALL : SLOW

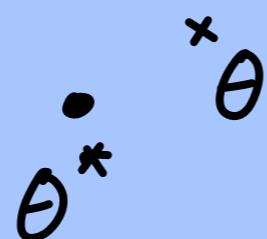
$\gamma$  LARGE: OVERSHOOT



NEAR FIXED POINT  $\theta^*$  [WHERE  $\nabla_{\theta} \mathcal{L}(\theta^*) = 0$ ]:

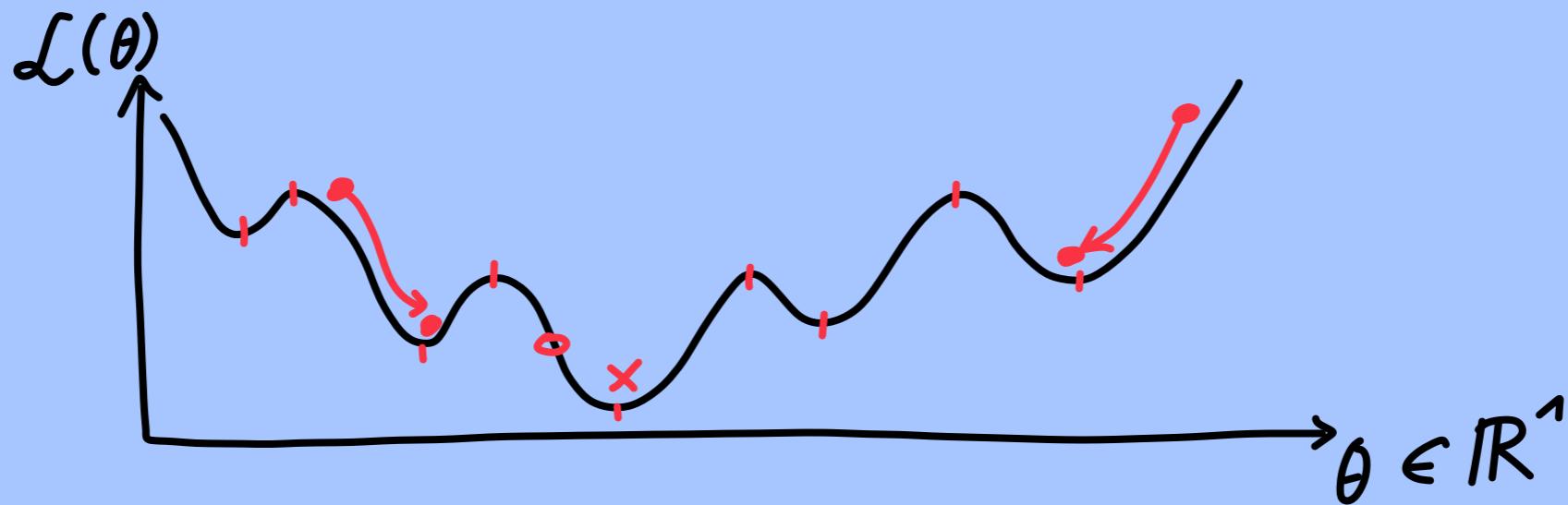
$$-\nabla_{\theta} \mathcal{L}(\theta) = 0 - M (\underline{\theta - \theta^*}) + \dots$$

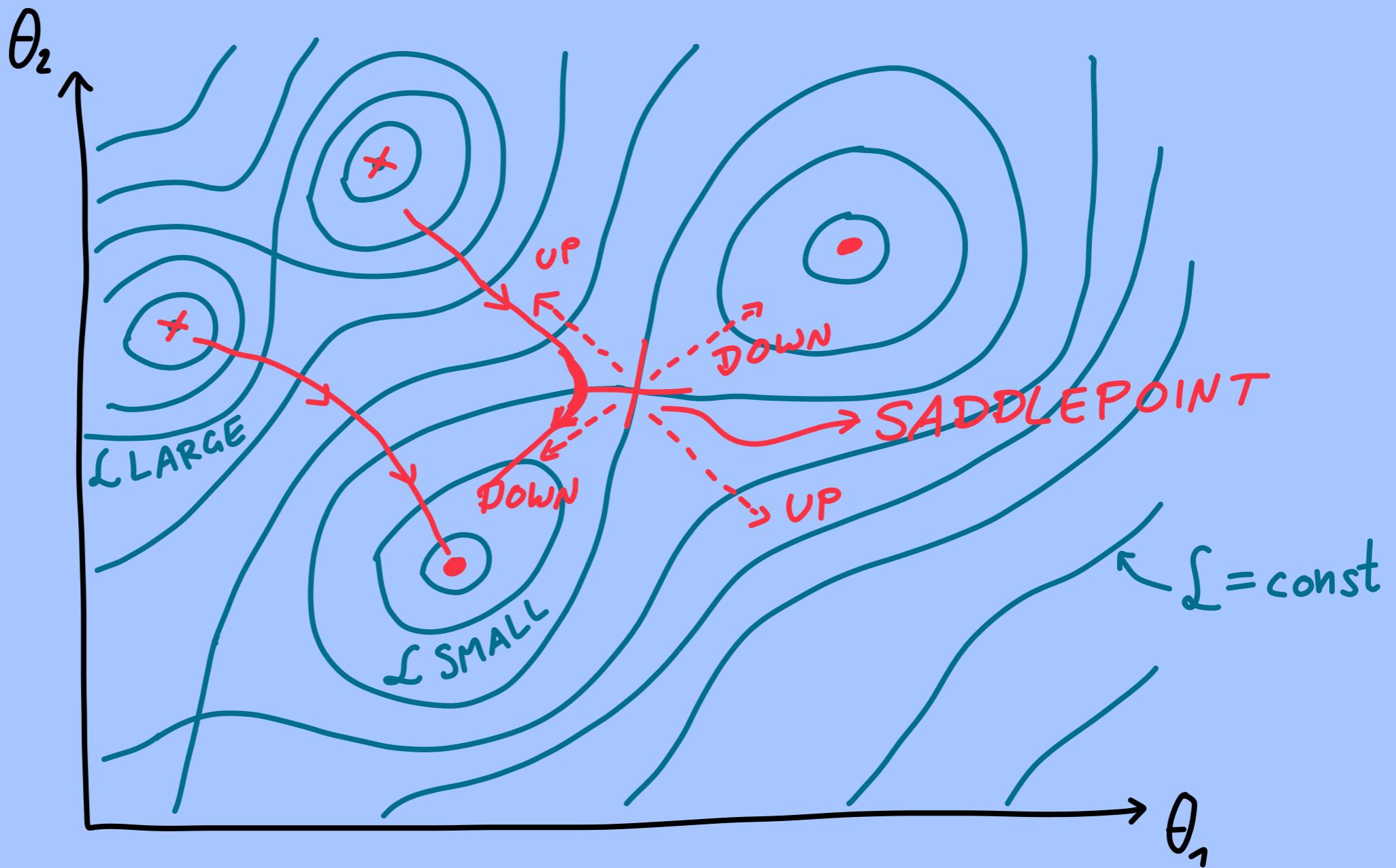
AT MINIMUM:  
M POSITIVE  
DEFINITE



$$\|\theta^{(t)} - \theta^*\|_2 \sim e^{-\lambda t} \xrightarrow{\text{SMALLEST (!) EIGENVALUE OF } M}$$

LOCAL MINIMA ?



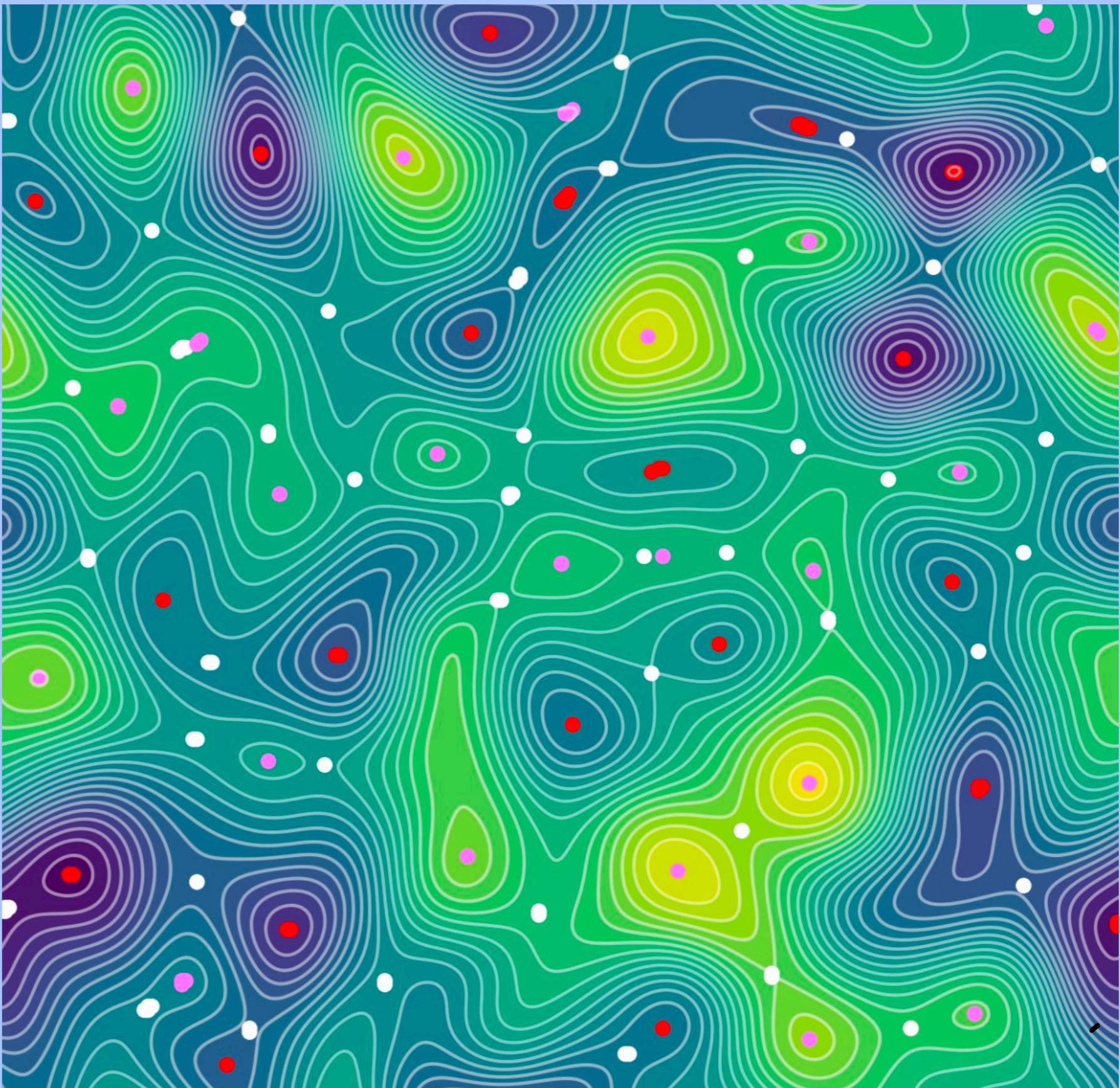


# GAUSSIAN RANDOM FIELD IN 2D

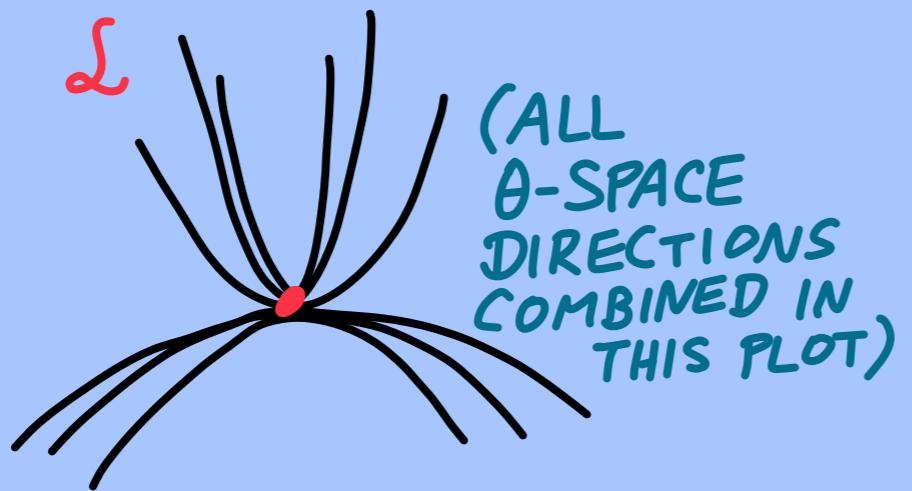
FRACTION  
OF:

- MINIMA :  $\frac{1}{4}$
- MAXIMA :  $\frac{1}{4}$
- SADDLE  
POINTS :  $\frac{1}{2}$

SEE CODE:  
Extrema And..  
.. Saddle points  
NOTEBOOK



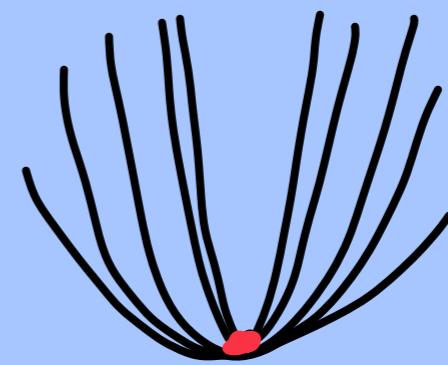
SADDLE POINT  
(MANY!)



BOTH POSITIVE &  
NEGATIVE CURVATURES  
= EIGENVALUES OF  
HESSIAN MATRIX

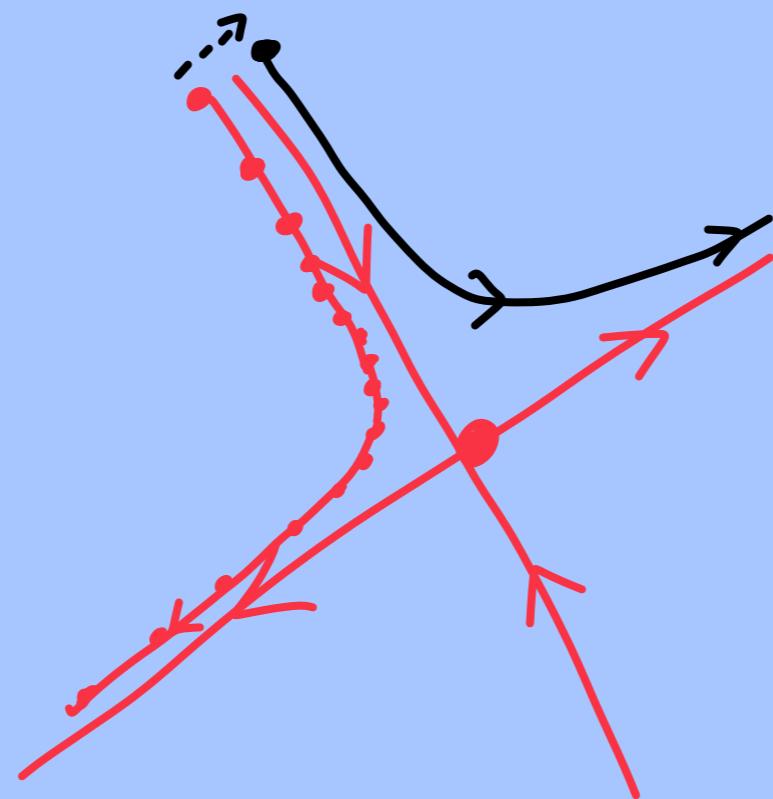
$$\left[ \frac{\partial^2}{\partial \theta_j \partial \theta_k} \mathcal{L}(\theta) \right]_{jk}$$

MINIMUM  
(MUCH FEWER!)



ONLY POSITIVE  
CURVATURE

RESULT:  
FRACTION OF  
MINIMA IS  
EXPONENTIALLY  
SMALL IN THE  
θ-SPACE DIMENSION!



CHALLENGE:

$$\mathcal{L}(\theta) = \langle \mathcal{L}(F_\theta(x), y^{\text{true}(x)}) \rangle_x$$

↓  
TOO  
COSTLY!

⇒ SAMPLING:

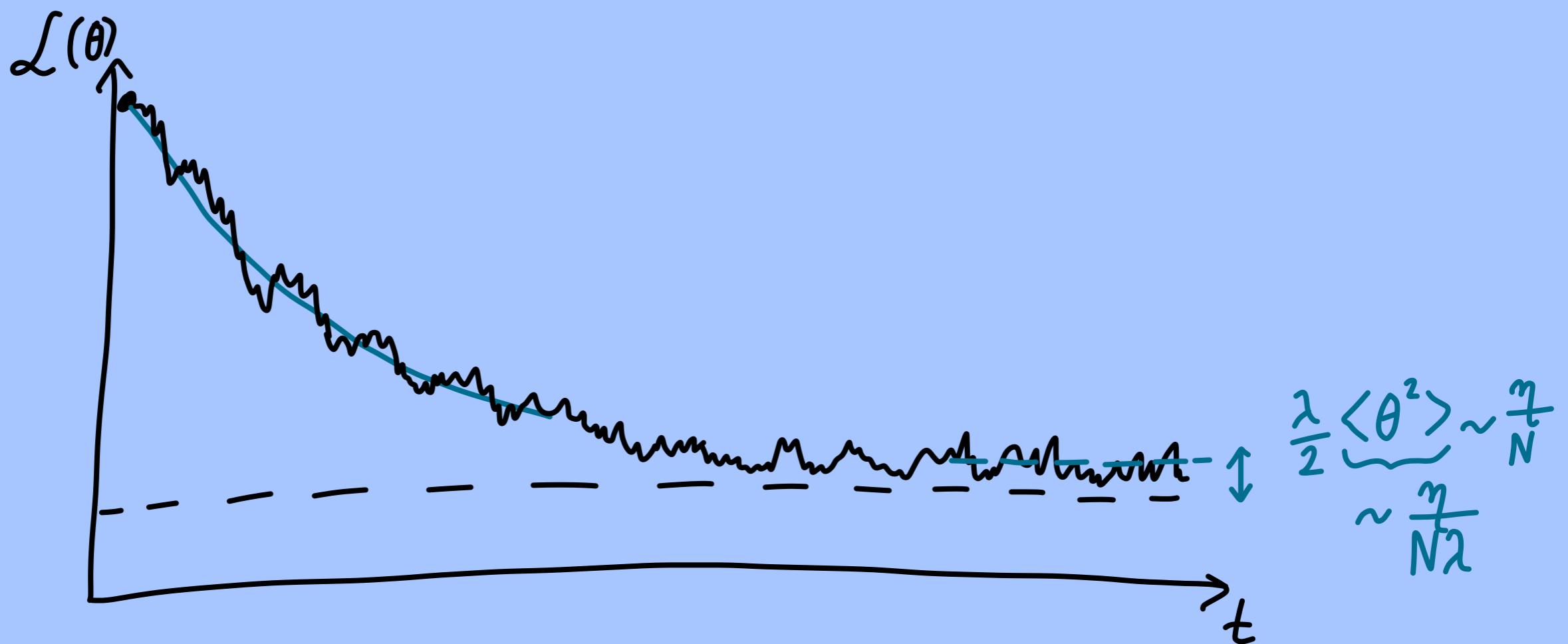
$$\mathcal{L}(\theta) = \langle \mathcal{L}(\dots) \rangle_x \approx \frac{1}{N} \sum_{j=1}^N \mathcal{L}(F_\theta(x_j), y^{\text{true}(x_j)}) =: \bar{\mathcal{L}}(\theta, \{x\})$$

BATCH SIZE → "BATCH"  $\{x\}$   
 $= \{x_1, x_2, \dots, x_N\}$

SAMPLES DRAWN FROM DISTR. OF  $x$

$$\bar{\mathcal{L}}(\theta, \{\cdot\}) = \mathcal{L}(\theta) + \underbrace{\text{SAMPLING NOISE}}$$

SAME FOR GRADIENT



PHYSICS: DYNAMICS OF  
AN OVERDAMPED  $(\dot{\theta} \sim -\nabla_{\theta} \mathcal{L})$   
PARTICLE WITH  
FLUCTUATIONS

NEAR FIXED POINT:

$$\theta^* = 0$$

$\theta \in \mathbb{R}^1$

$$S\theta^{(t)} = -\eta(\lambda\theta^{(t)} + \xi^{(t)}) \quad t = \frac{\text{TIME}}{\text{STEP}}$$

$$\theta^{(t+1)} = \underbrace{(1-\eta\lambda)\theta^{(t)}}_{-\eta\lambda} - \underbrace{\eta\xi^{(t)}}_{\xi^{(t)}}$$

$$\theta^{(t)} = -\eta \sum_{t'=0}^t e^{-\eta\lambda(t-t')} \xi^{(t')} + e^{-\eta\lambda t} \theta^{(0)}$$

FOR  $t \rightarrow \infty$

$$\langle [\theta^{(t)}]^2 \rangle = \eta^2 \sum_{t'_1, t'_2} e^{-\eta\lambda(t-t'_1)} e^{-\eta\lambda(t-t'_2)} \underbrace{\langle \xi^{(t'_1)} \xi^{(t'_2)} \rangle}_{S_{t'_1, t'_2} \langle \xi^2 \rangle}$$

$$\begin{aligned} \langle [\theta^{(t)}]^2 \rangle &\approx \gamma^2 \sum_{t'=-\infty}^t e^{-2\gamma\lambda \frac{(t-t')}{n}} \langle \xi^2 \rangle \\ &= \gamma^2 \langle \xi^2 \rangle \underbrace{\sum_{n=0}^{\infty} e^{-2\gamma\lambda n}}_{\approx \frac{1}{2\gamma\lambda}} \\ &\approx \frac{1}{2\gamma\lambda} \end{aligned}$$

$$\langle \theta^2 \rangle = \gamma^2 \langle \zeta^2 \rangle \frac{1}{2\gamma_2}$$

$\downarrow$

$$\frac{1}{N} \operatorname{Var}_x \partial_\theta \mathcal{L}(\theta, x)$$

$\sim \frac{\eta}{N\lambda} \underbrace{\operatorname{Var}_x \partial_\theta \mathcal{L}}_{\text{CURVATURE OF } \mathcal{L}}$   
 $N \uparrow \Rightarrow \langle \theta^2 \rangle \downarrow$ 


## 2.6

# CALCULATING GRADIENTS: AUTOMATIC DIFFERENTIATION AND BACKPROPAGATION

GOAL : CALCULATE  $\nabla_{\theta} \mathcal{L}(\theta, x)$   
 DEPENDS ON  $y^{nn}(x) = F_{\underline{\theta}}(x)$

$$\underline{y}^{(n)} = F^{(n)}(y^{(n-1)}, \underline{\theta})$$

$$\frac{\partial y_k^{(n)}}{\partial \theta_j} = \underbrace{\frac{\partial F_k}{\partial \theta_j}}_{\text{EXPLICIT DEPENDENCE}} + \sum_l \frac{\partial F_k}{\partial y_l^{(n-1)}} \frac{\partial y_l^{(n-1)}}{\partial \theta_j}$$

GO DOWN RECURSIVELY  
 $n \mapsto n-1$

INDIRECT  
 DEPENDENCE

GENERAL STRATEGY:

"FORWARD PASS":

- CALCULATE  $y^{(1)}, y^{(2)}, \dots, y^{(n)}$  <sup>OUTPUT</sup>  
FROM  $y^{(0)} \equiv x$
- STORE  $\frac{\partial F_k}{\partial y_e^{(n-1)}}$  FOR LATER

$$\frac{\partial \bar{L}}{\partial \theta_j} = \left\langle \sum_k \frac{\frac{\partial L(y^{NN}(x), y^{\text{true}}(x))}{\partial y_k^{(N)}} \frac{\partial y_k^{(N)}}{\partial \theta_j}}{\partial y_k^{(N)}} \right\rangle_{\xi \times \xi}$$

$\Delta_k$   
"ERROR SIGNAL"

GRADIENT  $g_j$

$$[ L = (y^{NN} - y^{\text{true}})^2 \Rightarrow \Delta = \frac{\partial L}{\partial y^{NN}} = 2(y^{NN} - y^{\text{true}}) ]$$

START  $g \equiv 0$

FOR EACH  $n = N, N-1, \dots$  :  
"BACKWARD PASS"

ACCUMULATE GRADIENT:

$$g_j^{\text{NEW}} = g_j + \sum_k \Delta_k \frac{\partial F_k^{(n)}}{\partial \theta_j}$$

MATRIX-VECTOR PROD.

UPDATE ERROR SIGNAL:

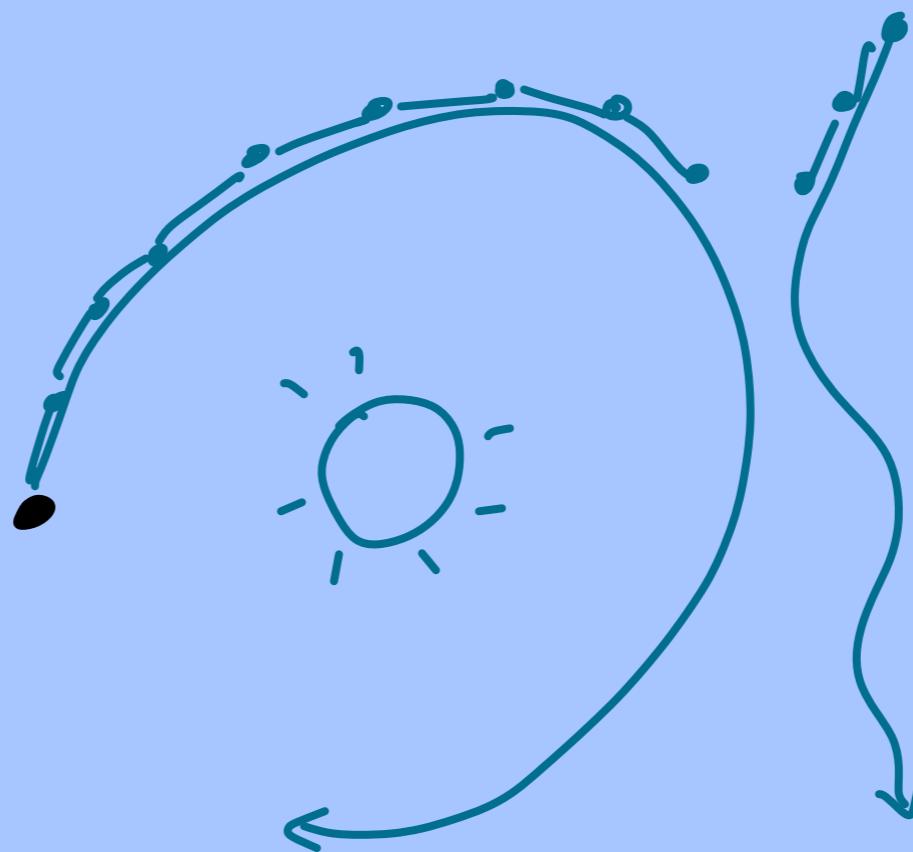
$$\Delta_\ell^{\text{NEW}} = \sum_k \Delta_k \frac{\partial F_k^{(n)}}{\partial y_\ell^{(n-\ell)}}$$

EFFORT  $\hat{=}$  FORWARD PASS !  
"BACK PROPAGATION"

# "AUTOMATIC DIFFERENTIATION"

→ TENSORFLOW, JAX, PYTORCH,...

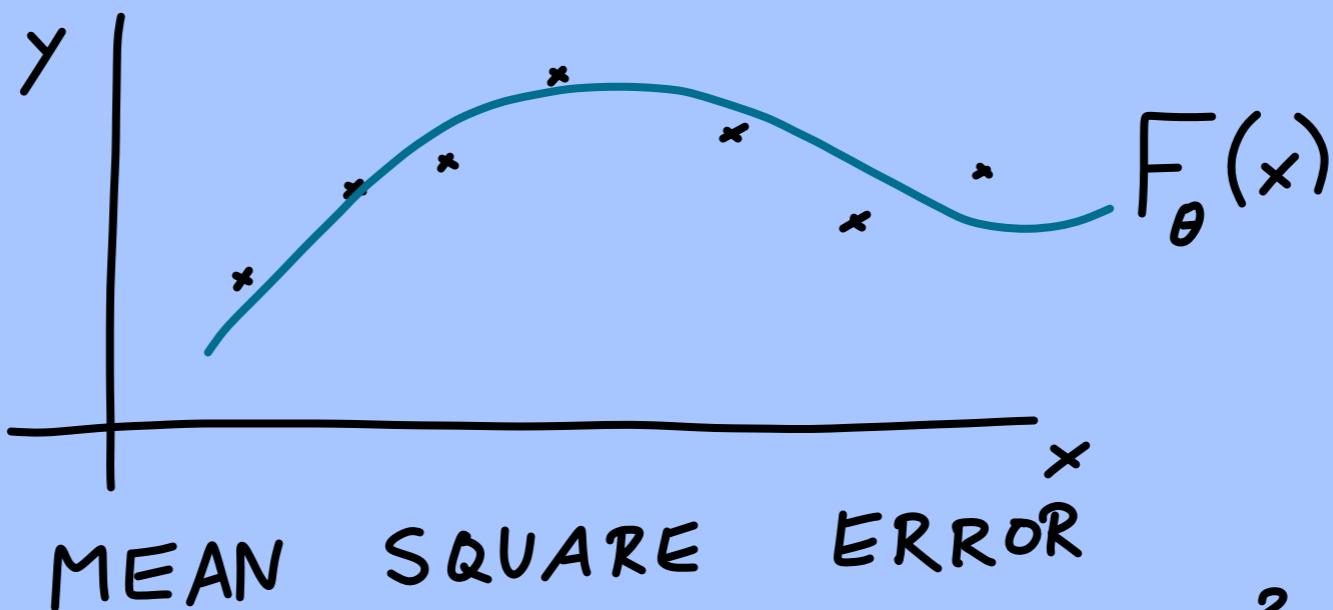
NOTE: - ALL NUMERICAL!  
- NOT SYMBOLIC!



## 2.7

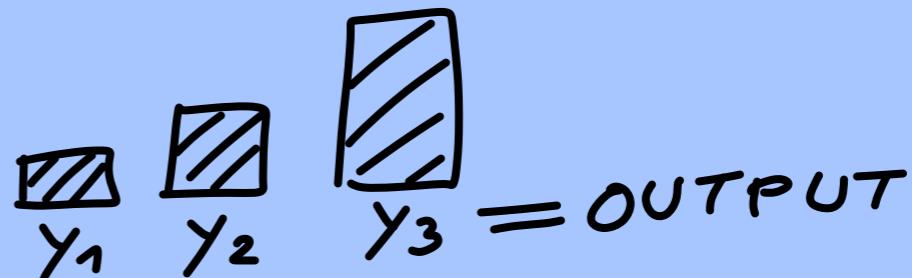
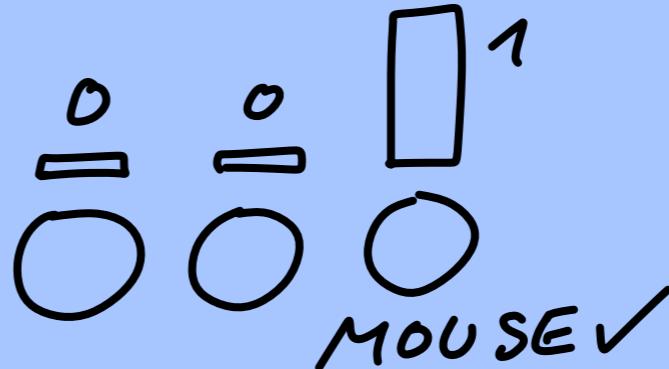
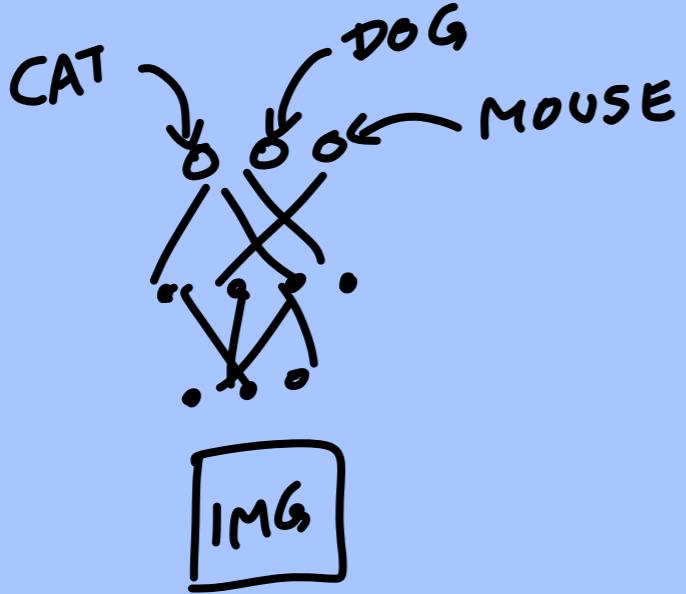
## SOME LOSS FUNCTIONS

"REGRESSION TASKS"



$$\mathcal{L}(y_{(x)}^{NN}, y_{(x)}^{\text{true}}) = (y^{NN} - y^{\text{true}})^2$$

# CLASSIFICATION TASKS:



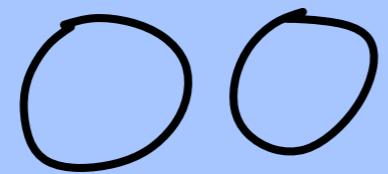
$$y_j = p_j^{NN} \quad (\text{SOFTMAX})$$

"CATEGORICAL CROSS-ENTROPY"

$$\mathcal{L}(p^{NN}, p^{\text{true}}) = - \sum_j p_j^{\text{true}} \ln p_j^{NN}$$

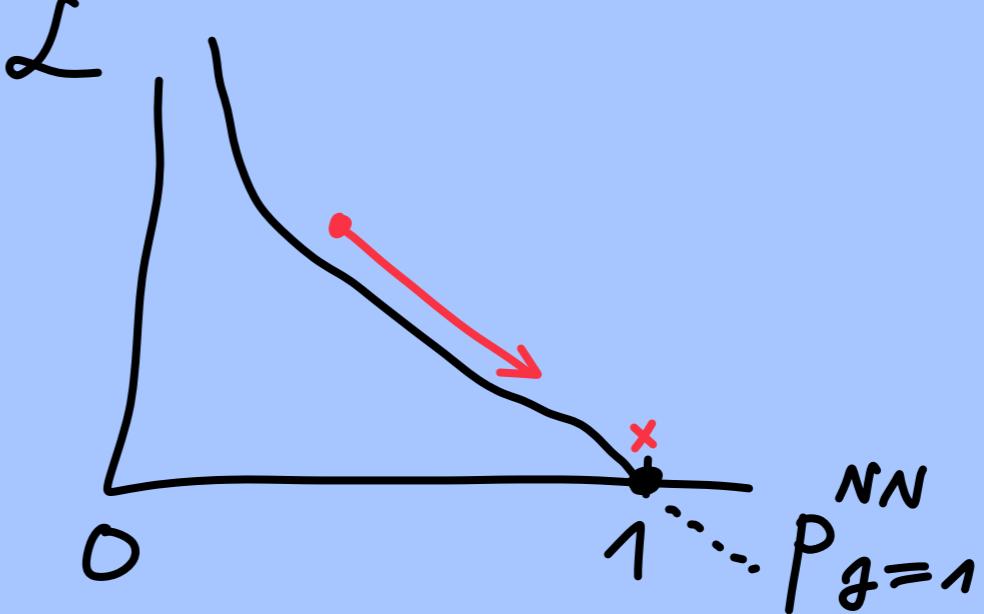
$\mathcal{L}$  IS MINIMIZED BY  $p_j^{NN} \equiv p_j^{\text{true}}$

EXAMPLE: 2 OUTPUT NEURONS

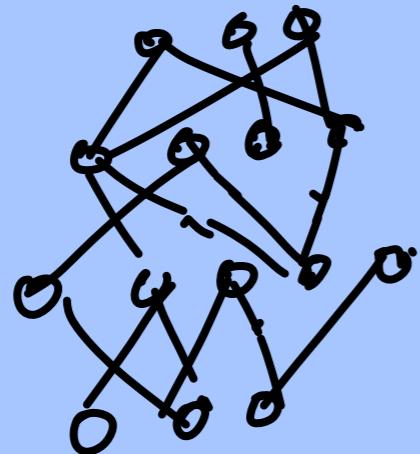


$$P_{g=1}^{\text{true}} = 1$$

$$\Rightarrow L = -\ln P_{g=1}^{NN}$$



# LOSS CONTRIBUTIONS FOR ACTIVATIONS OR WEIGHTS



GOAL : SPARSE WEIGHTS (OR ACT.)

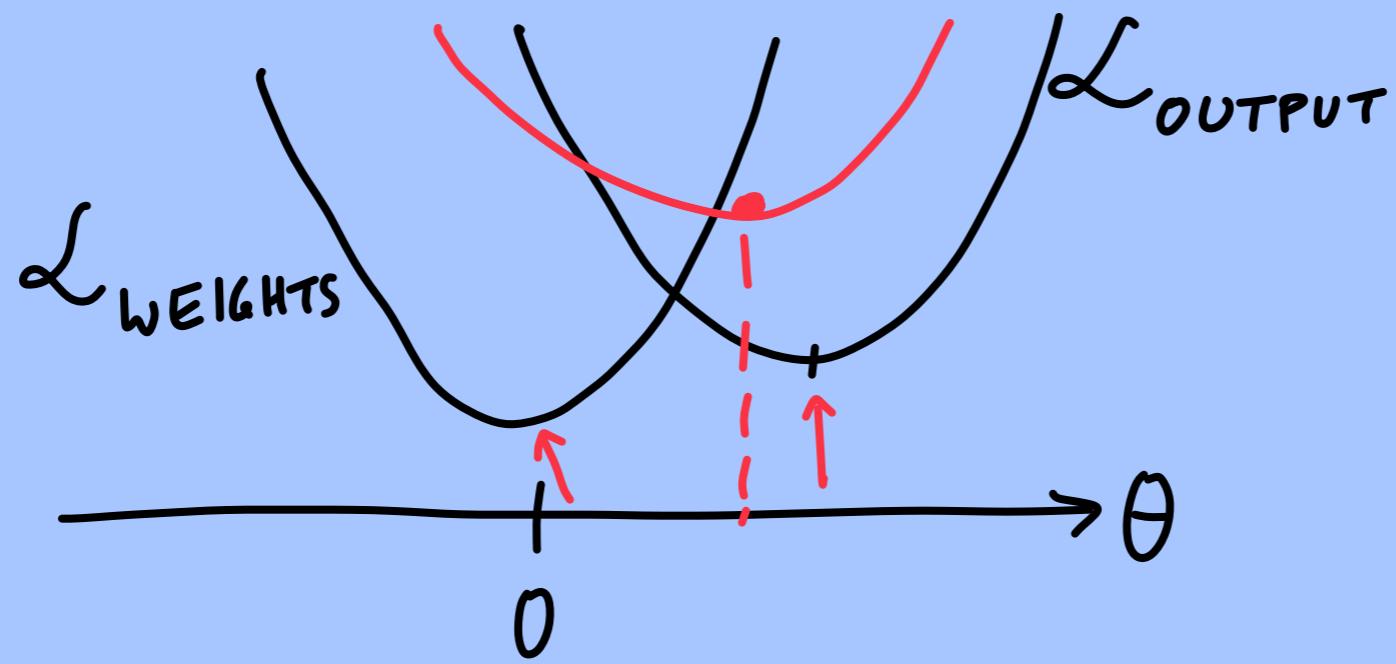
⇒ CAN "PRUNE" THE NN!

"REGULARIZATION"

$$\mathcal{L}_{\text{TOTAL}} = \underbrace{\mathcal{L}_{\text{OUTPUT}}}_{+} + \underbrace{\mathcal{L}_{\text{WEIGHTS}}}_{}$$

$$\sum_{h,e} (w_{he}^{(n)})^2$$

COULD ALSO USE  $\sum_{h,e} \lambda_{h,e} (w_{he}^{(n)})^2$



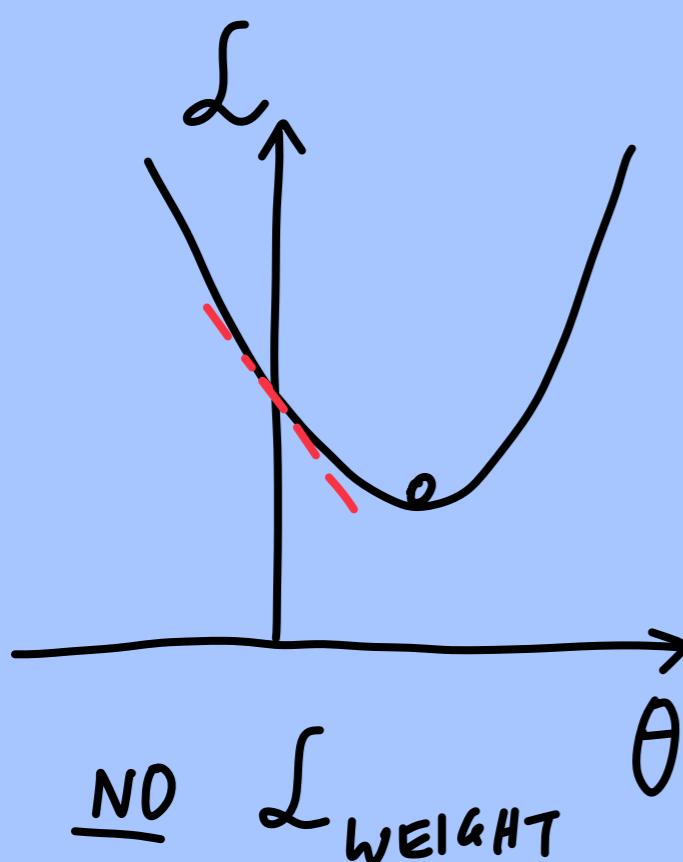
EVEN BETTER FOR SPARSITY:

"L<sub>1</sub>-LOSS"

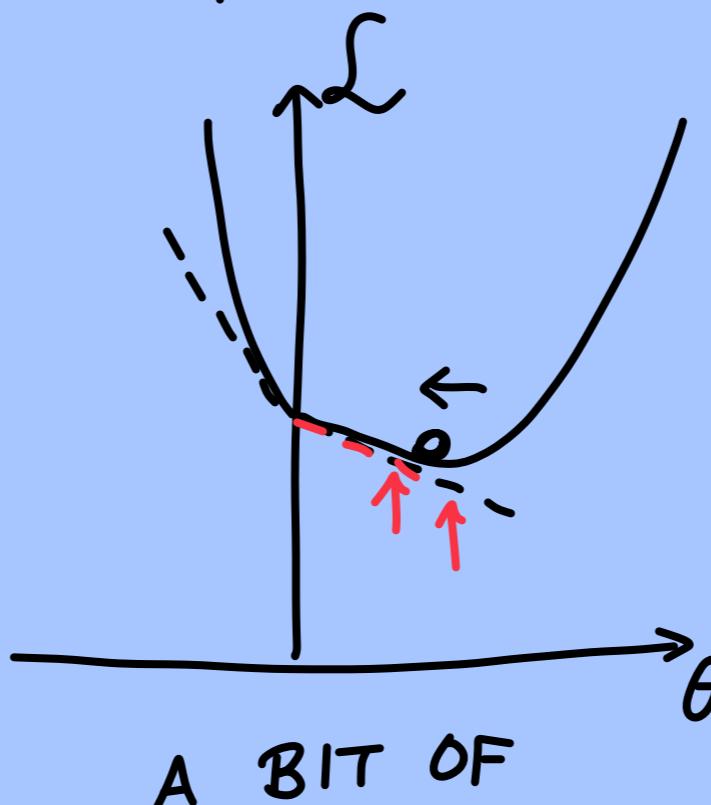
$$L_{\text{WEIGHT}} = \lambda \sum_{k, l} |w_{k,l}^{(n)}|$$

$$\text{"L}_2\text{" } \|v\|_2 = \sqrt{\sum v_j^2}$$

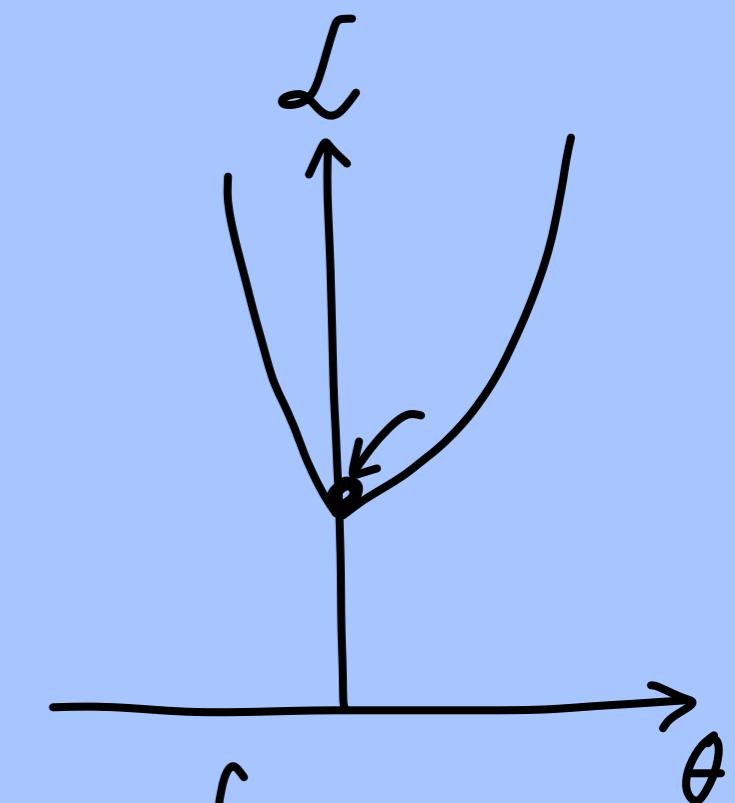
$$\text{"L}_1\text{" } \|v\|_1 = \sum |v_i|$$



NO  $L_{\text{WEIGHT}}$



A BIT OF  
 $L_{\text{WEIGHT}}$

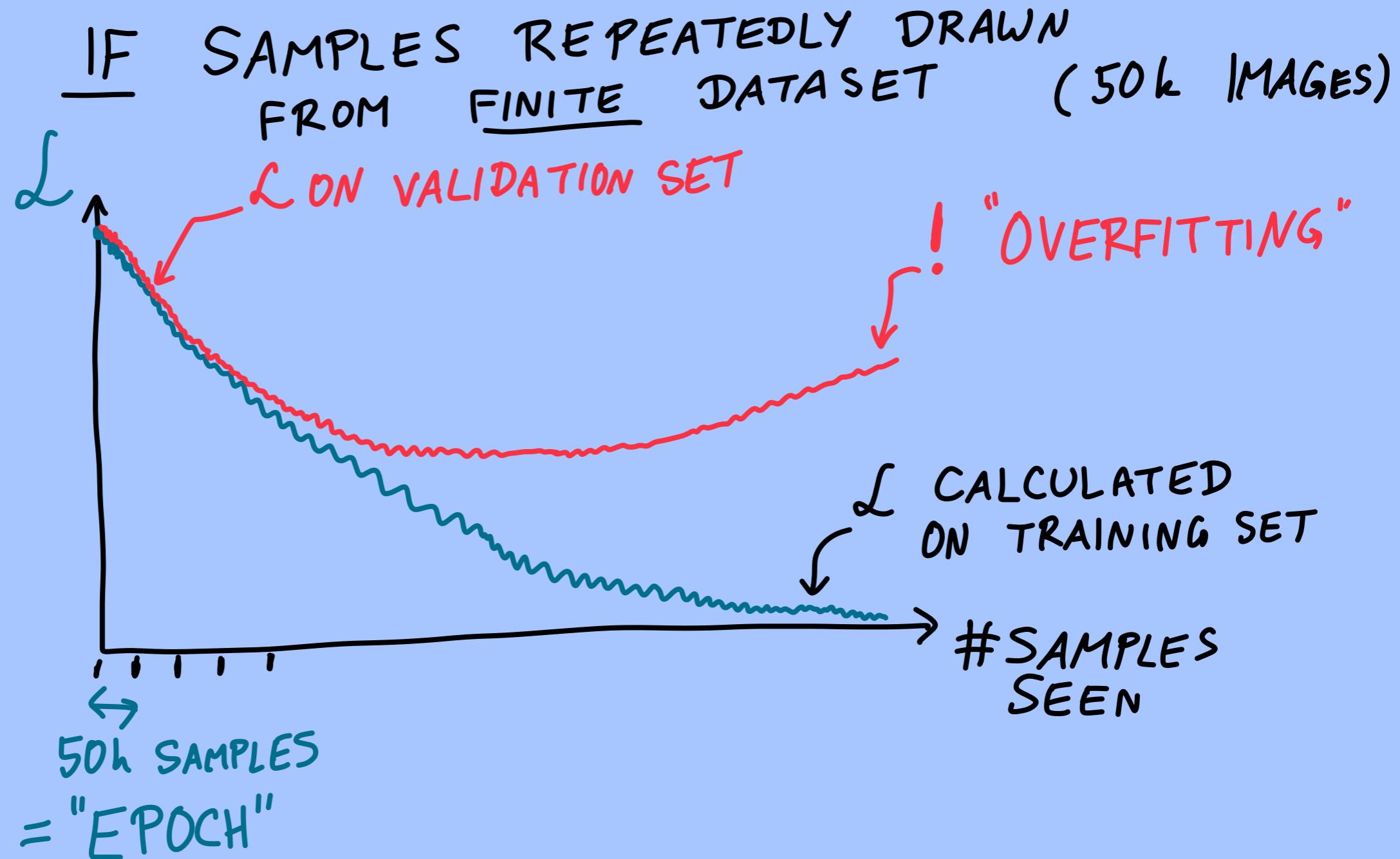


$L_{\text{WEIGHT}}$   
STRONG  
( $\lambda$  LARGE)

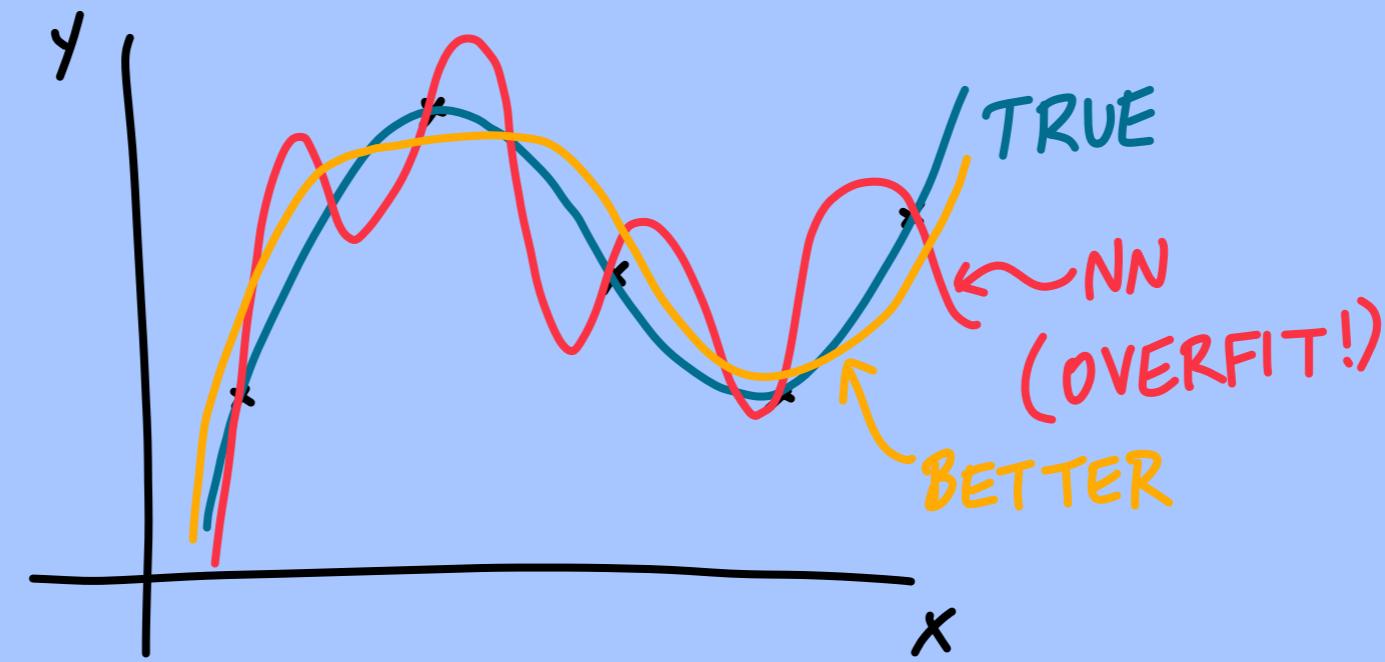
⇒  $\theta$  CAN BECOME EXACTLY ZERO!  
GOOD!

## 2.8

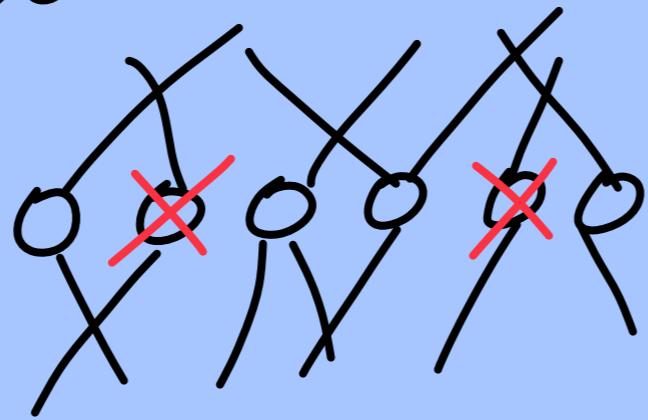
## OVERFITTING AND DROPOUT



# 1D EXAMPLE



DROPOUT:



RANDOMLY "DROP OUT" NEURONS DURING TRAINING (OR ADD NOISE)

⇒ PREVENTS OVERFITTING

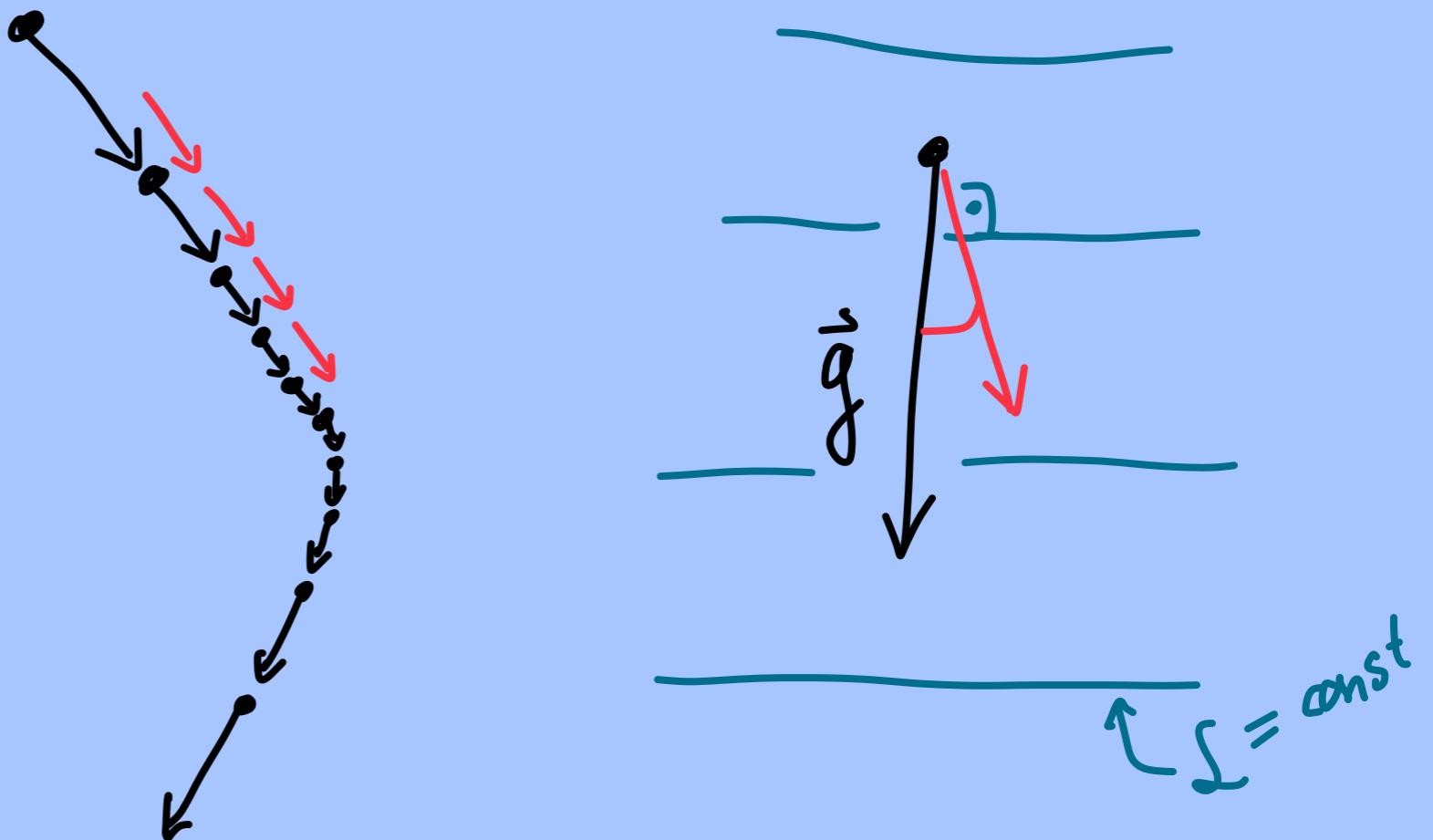
2.9

## ADAPTIVE GRADIENT DESCENT

NOW:

$$\delta\theta_j = - \underbrace{\gamma_j g_j}_{\substack{\text{ADAPTIVE} \\ \text{LEARNING RATE}}} \quad \text{GRADIENT} \quad \frac{\partial L}{\partial \theta_j}$$

[COMPARE OTHER NUMERICAL TECHNIQUES,  
LIKE RUNGE-KUTTA WITH ADAPTIVE STEPSIZE]



MANY VARIANTS: ADAGRAD, ADADELTA, RMSPROP, ...

## ADAM (ONE OF THE MOST POPULAR)

$$m^{(t)} = \beta m^{(t-1)} + (1-\beta) g^{(t)}$$

**"MOMENTUM"**

$0 < \beta < 1$   $\Rightarrow$  "RUNNING AVERAGE" OF  $g$  OVER  $\sim \frac{1}{1-\beta}$  TIME STEPS

$$V^{(t)} = \gamma V^{(t-1)} + (1-\gamma) [g^{(t)}]^2$$

$$\hat{m}^{(t)} = \frac{m^{(t)}}{1-\beta^t}$$

$$\hat{V}^{(t)} = \frac{V^{(t)}}{1-\gamma^t} \Rightarrow$$

CORRECTIONS IN BEGINNING

$$\theta_g^{(t)} = -\gamma \frac{\hat{m}_g^{(t)}}{\sqrt{V_g^{(t)}} + \epsilon}$$

EXAMPLE:

$$\beta \approx 0.9$$

$$\gamma = 0.999$$

$$\epsilon \approx 10^{-8}$$

## 2.10

SUMMARY: BASICS OF  
NEURAL NETWORKS

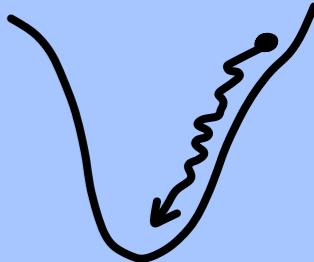
- FUNCTION APPROXIMATION  $y = F_\theta(x)$
- LAYER STRUCTURE, LINEAR & NONLINEAR STEPS

- GOAL: MINIMIZE LOSS

$$\mathcal{L}(\theta) = \langle \mathcal{L}(F_\theta(x), y^{\text{true}}(x)) \rangle_x$$

- STOCHASTIC GRADIENT DESCENT

$$\delta\theta = -\eta \frac{\partial \mathcal{L}}{\partial \theta} \approx -\eta \left\langle \frac{\partial \mathcal{L}}{\partial \theta} \right\rangle_{\text{BATCH } \{x\}}$$

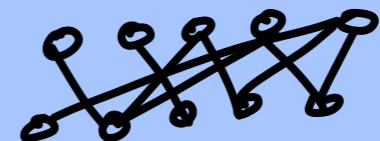


- CALCULATE GRADIENT EFFICIENTLY VIA BACKPROPAGATION
- CHOICES OF: NETWORK STRUCTURE (#LAYERS, #NEURONS)  
ACTIVATION FUNCTIONS,  
LOSS, BATCHSIZE, LEARNING RATE
- DANGER OF OVERRFITTING

2.11

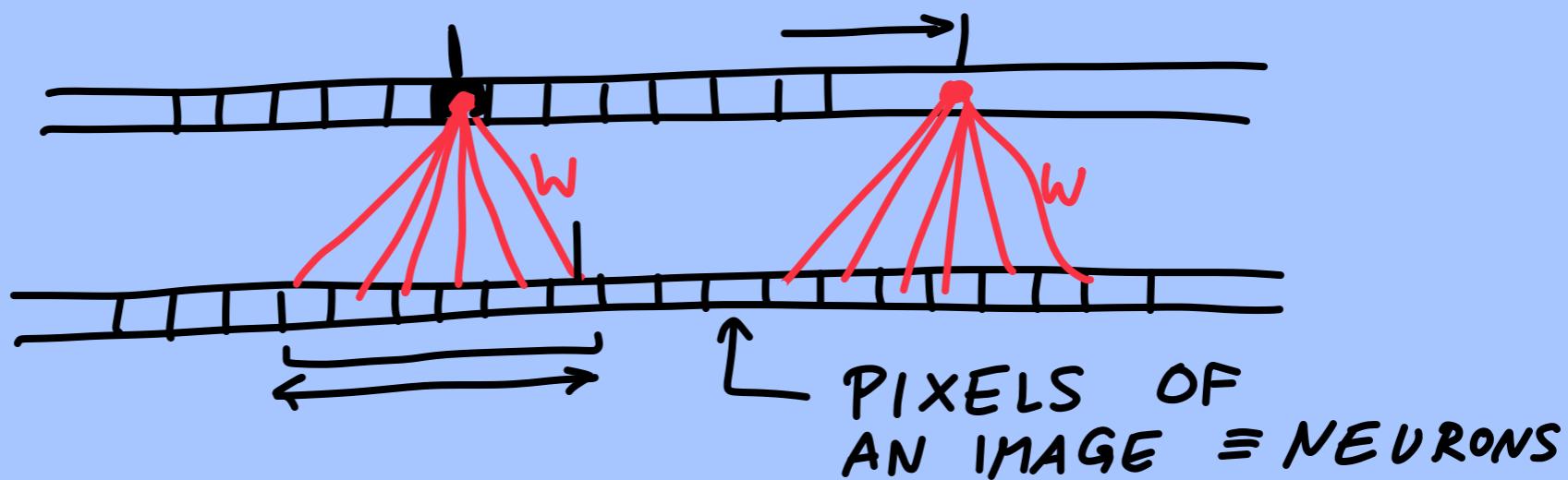
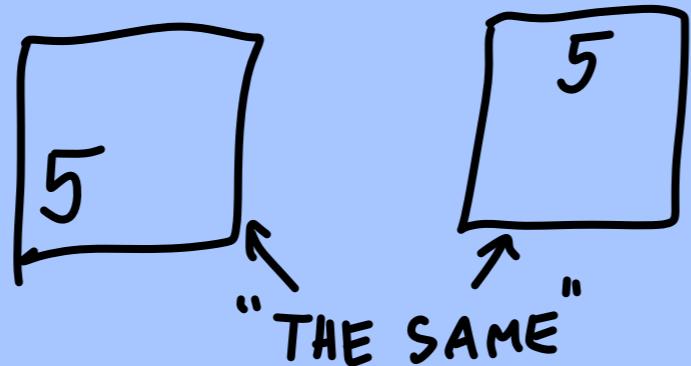
## CONVOLUTIONAL NEURAL NETWORKS

SO FAR: "FULLY CONNECTED NN"



NOW: CONSTRAIN NN STRUCTURE BY SYMMETRY

HERE: TRANSLATIONAL SYMMETRY



## DISCRETE CONVOLUTION

$$z_j^{(n)} = \sum_{j'} w^{(n)}(j-j') y_{j'}^{(n-1)} + b^{(n)}$$

$|j-j'| \leq d$

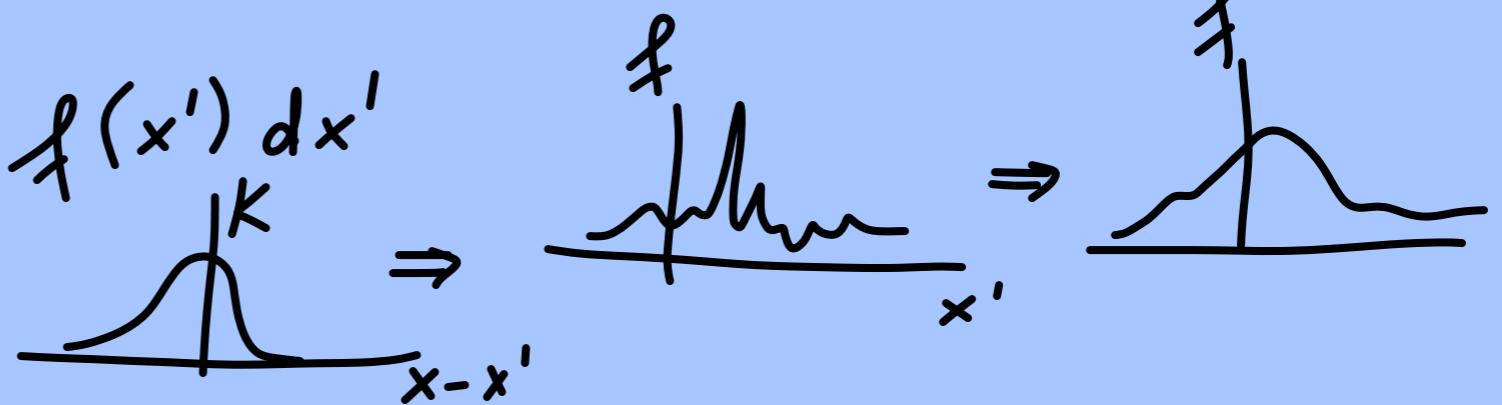
"KERNEL",  
DEPENDS ONLY  
ON DISTANCE  $j-j'$

$$y_j^{(n)} = f(z_j^{(n)})$$

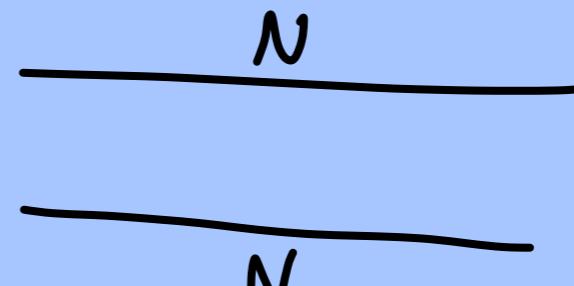
## CONVOLUTION

$$f^{\text{NEW}}(x) = \int K(x-x') f(x') dx'$$

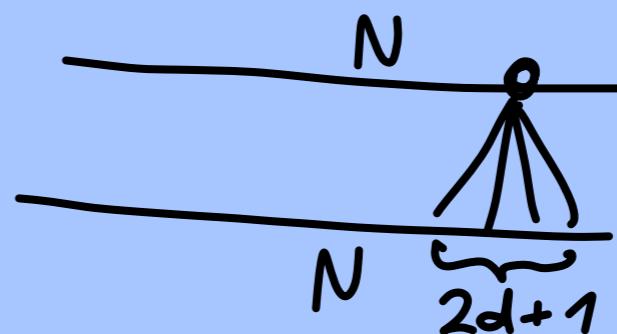
E.G. SMOOTHING



FULLY CONNECTED

$$\Rightarrow w \in \mathbb{R}^{N \times N}$$
$$N^2$$


CONV. NN



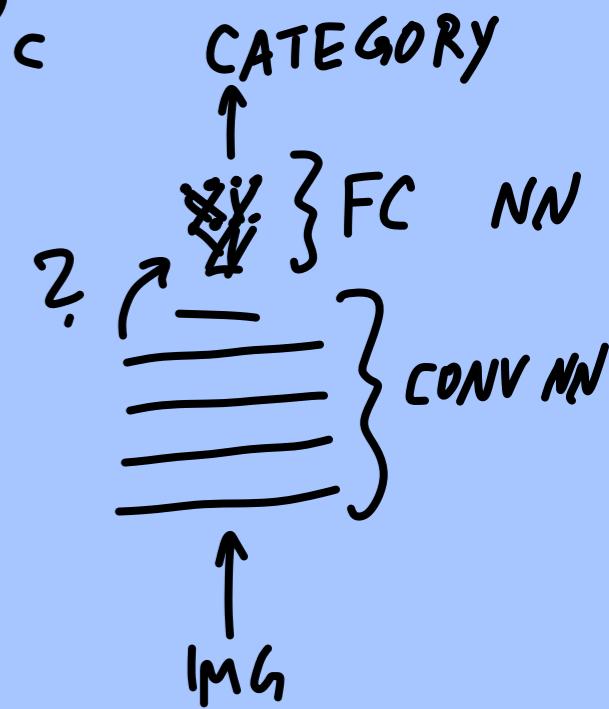
$w : \underline{2d+1}$  NUMBERS

$\Rightarrow$  # PARAMS IS  
STRONGLY REDUCED!

$$z_{j,c}^{(n)} = \sum_{j',c'}^{(n)} w_{cc'}^{(n)}(j-j') y_{j',c'}^{(n-1)} + b_c^{(n)}$$

↑ CHANNEL NUMBER

TRANSITION CONV → FC



E.G. GLOBAL POOLING ≡ SUM

$$y_c^{(n)} = \sum_j y_{j,c}^{(n-1)}$$

"FLATTEN"

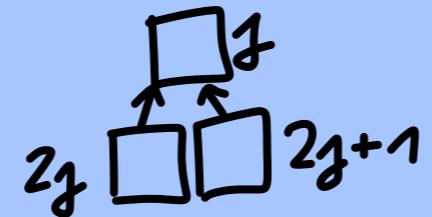
$$y_{j'}^{(n)} = y_{j \rightarrow \text{POSITION}}^{(n-1)}$$

NEURON INDEX

AFTERWARDS :

$$\sum_{j'} w_{jj'}^{(n+1)} y_{j'}^{(n)}$$

"DOWNSAMPLING"  
(AVERAGE  
POOLING)

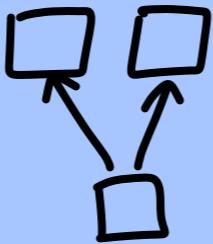


$$y_g^{(n)} = \frac{1}{2} (y_{2g}^{(n-1)} + y_{2g+1}^{(n-1)})$$

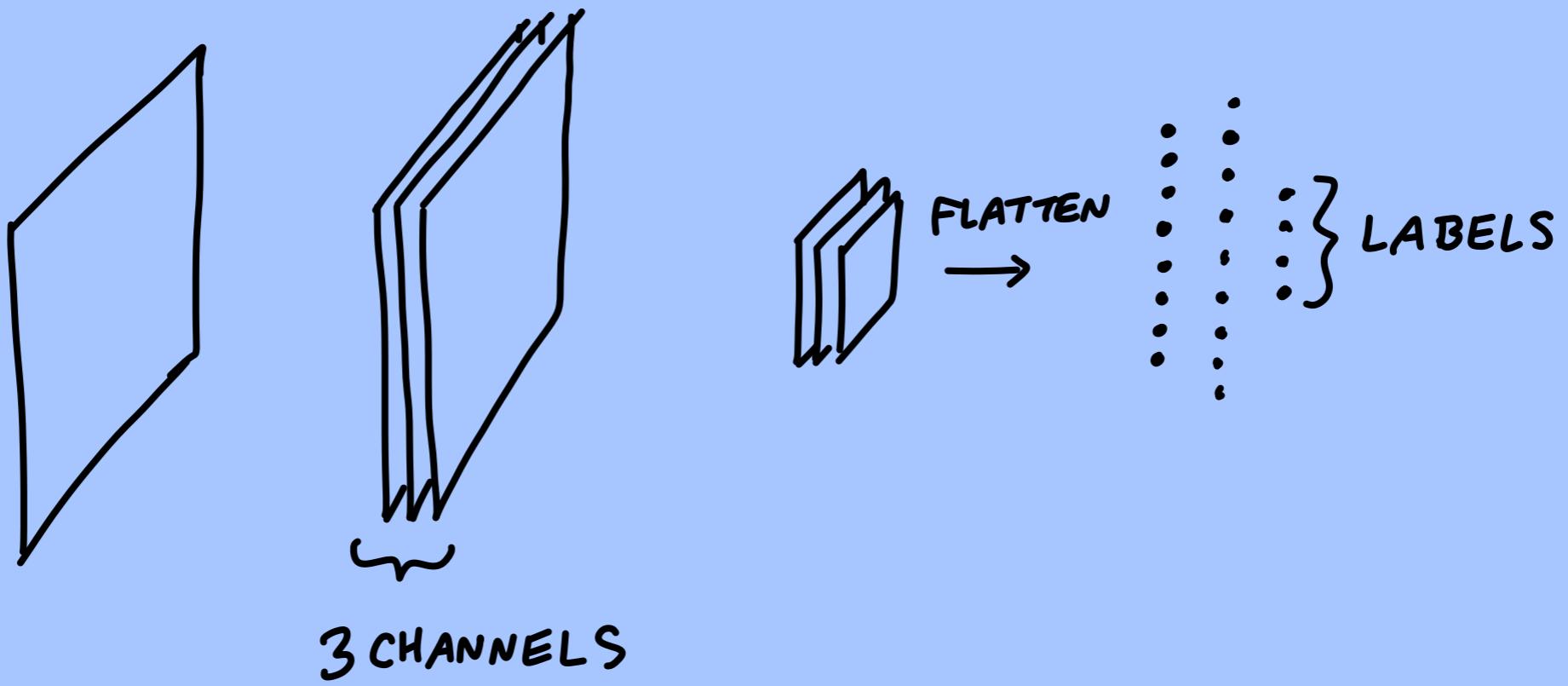
$\Rightarrow \frac{1}{2}$  OF NEURONS



"UPSAMPLING"

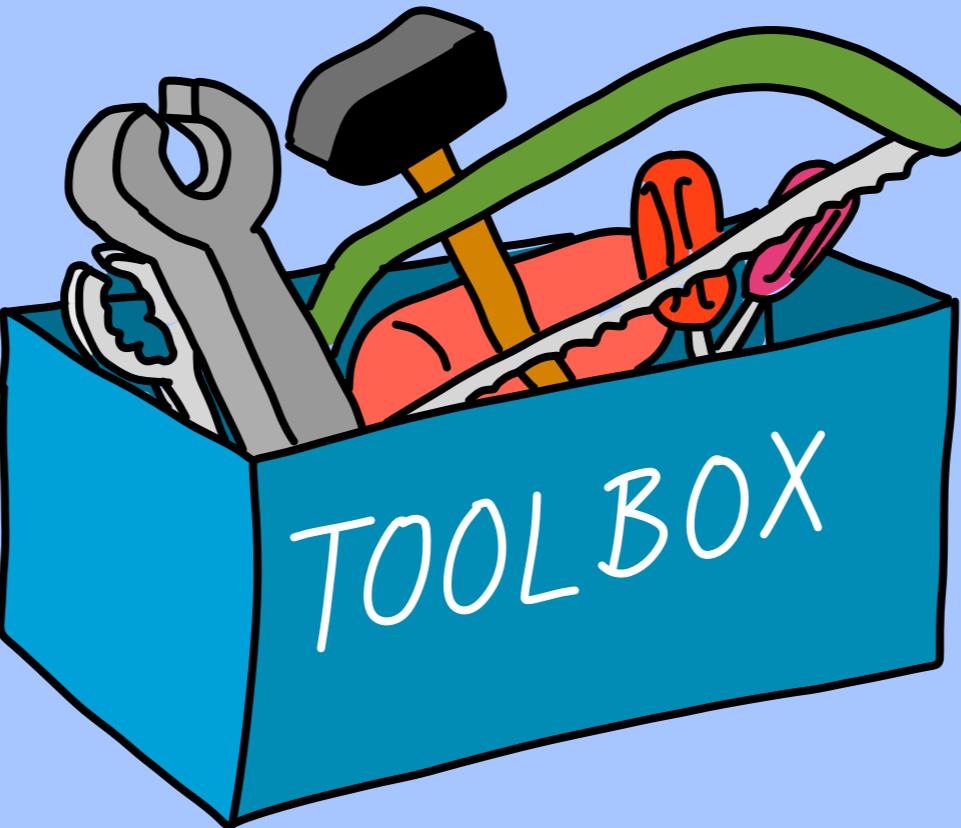


$$y_{2g}^{(n)} = y_g^{(n-1)}$$
$$y_{2g+1}^{(n)} = y_g^{(n-1)}$$



ARTIFICIAL  
NEURAL  
NETWORKS

X



BAYES

INFORMATION  
THEORY

REPRESENTATION  
LEARNING

ADVANCED  
NN STRUCTURES

LEARNING  
PROBABILITY  
DISTRIBUTIONS

DISCOVERING  
STRATEGIES

ADAPTIVE  
OBSERVATIONS

MEASURING  
COMPLEXITY

3

# REPRESENTATION LEARNING I

3.1

## MOTIVATION & GOALS

SCIENCE ≈ COMPRESSION //  
EXTRACT ESSENCE //

↓

PREDICT

*Paris Terria.* 89

**TABULARUM  
RUDOLPHI  
ASTRONOMI  
CARUM  
PARS TERTIA,  
DE ECLIPSIBUS SOLIS ET LUNAE, ALIASQUE  
PLANETARUM CONGREGATIONIBUS ET CON-  
figurationibus.**

*Typus Astronomicus, neque Politicus, neque Ecclesiasticus utilitas, sed  
metu Astronomico, servientibus ad Mundum Eclipticis  
in Methodo Annis Julianis.*

Numerus	Ianuarii	Februarii	Martii	Aprilis	Mayi	Junii	July	Augus.	Sept.	Octobri	Novem.	Decem.	Anno	Hora		
I	1	2	1	10	19	28	27	16	25	14	23	1	1	10:00		
II	1	2	1	10	19	28	27	16	25	14	23	1	1	10:00		
III	1	2	1	10	19	28	27	16	25	14	23	1	1	10:00		
XI	3	4	3	12	21	30	29	18	27	16	25	14	1	10:00		
XIX	4	5	4	13	22	31	30	19	28	17	26	15	1	10:00		
VIII	6	7	6	4	13	22	31	30	19	18	17	15	1	10:00		
XVI	7	6	7	5	3	2	1	31	19	18	17	15	1	10:00		
V	9	8	9	7	6	5	4	3	2	1	31	19	1	10:00		
XIII	10	9	10	8	6	5	4	3	2	1	31	19	1	10:00		
II	11	10	11	9	7	6	5	4	3	2	1	31	19	1	10:00	
X	13	12	11	10	8	7	6	5	4	3	2	1	31	19	1	10:00
XVIII	15	14	13	12	10	9	8	7	6	5	4	3	2	1	10:00	
VII	16	15	14	13	12	11	10	9	8	7	6	5	4	3	10:00	
XV	18	17	16	15	14	13	12	11	10	9	8	7	6	5	10:00	
XII	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	10:00
I	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	10:00
XI	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	10:00
XVII	26	25	24	23	22	21	20	19	18	17	15	13	12	11	10	10:00
VI	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	10:00
XIII	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	10:00
IV	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	10:00

\* Et fidei Ruffini, quoniam non Resonans non agnoscitur, sed in longiora pronuntiatur.

M Cyclo.Obv

6 Tabularum Radiophilis

Tabula Latitudinis LV N.B. implicata, una cum Reductione loci Orbitae  $\oplus$  ad Eclipticam, que videtur ab Nodo  $\ominus$  in Quadrante excentrico.

Luminos.	Gradi.	Min.	Sec.	Tabula Latitudinis LV N.B. implicata, una cum Reductione loci Orbitae $\oplus$ ad Eclipticam, que videtur ab Nodo $\ominus$ in Quadrante excentrico.	Gradi.	Min.	Sec.
0 0 0 0	15	10 29 52	1	10 29 52	15	10 29 52	1
0 0 1 1	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 10 29	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 10 30	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 21	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 22	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 23	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 24	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 25	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 26	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 27	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 28	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 29	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 30	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 31	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 32	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 33	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 34	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 35	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 36	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 37	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 38	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 39	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 40	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 41	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 42	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 43	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 44	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 45	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 46	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 47	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 48	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 49	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 50	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 51	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 52	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 53	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 54	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 55	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 56	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 57	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 58	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 59	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 40	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 41	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 42	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 43	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 44	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 45	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 46	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 47	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 48	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 49	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 50	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 51	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 52	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 53	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 54	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 55	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 56	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 57	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 58	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 59	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 40	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 41	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 42	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 43	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 44	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 45	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 46	21	1 24 21	1	1 24 21	21	1 24 21	1
0 0 11 47	21	1 24 21	1	1 24 2			

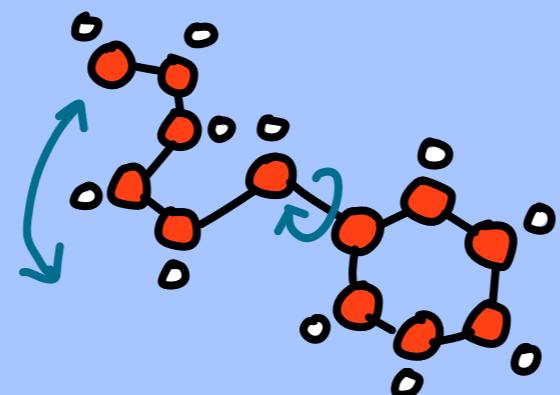
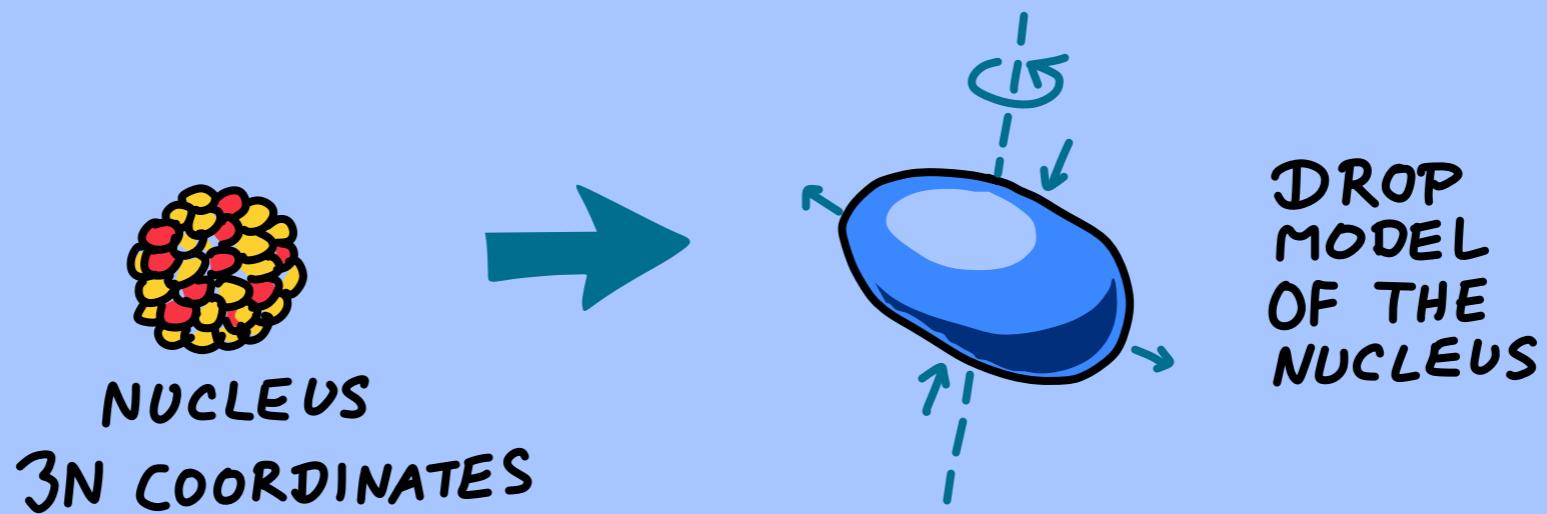
# LOSSY COMPRESSION



IMAGE → ... → PLANET  
POSITIONS  
( $1000 \times 1000$  PIXELS!)

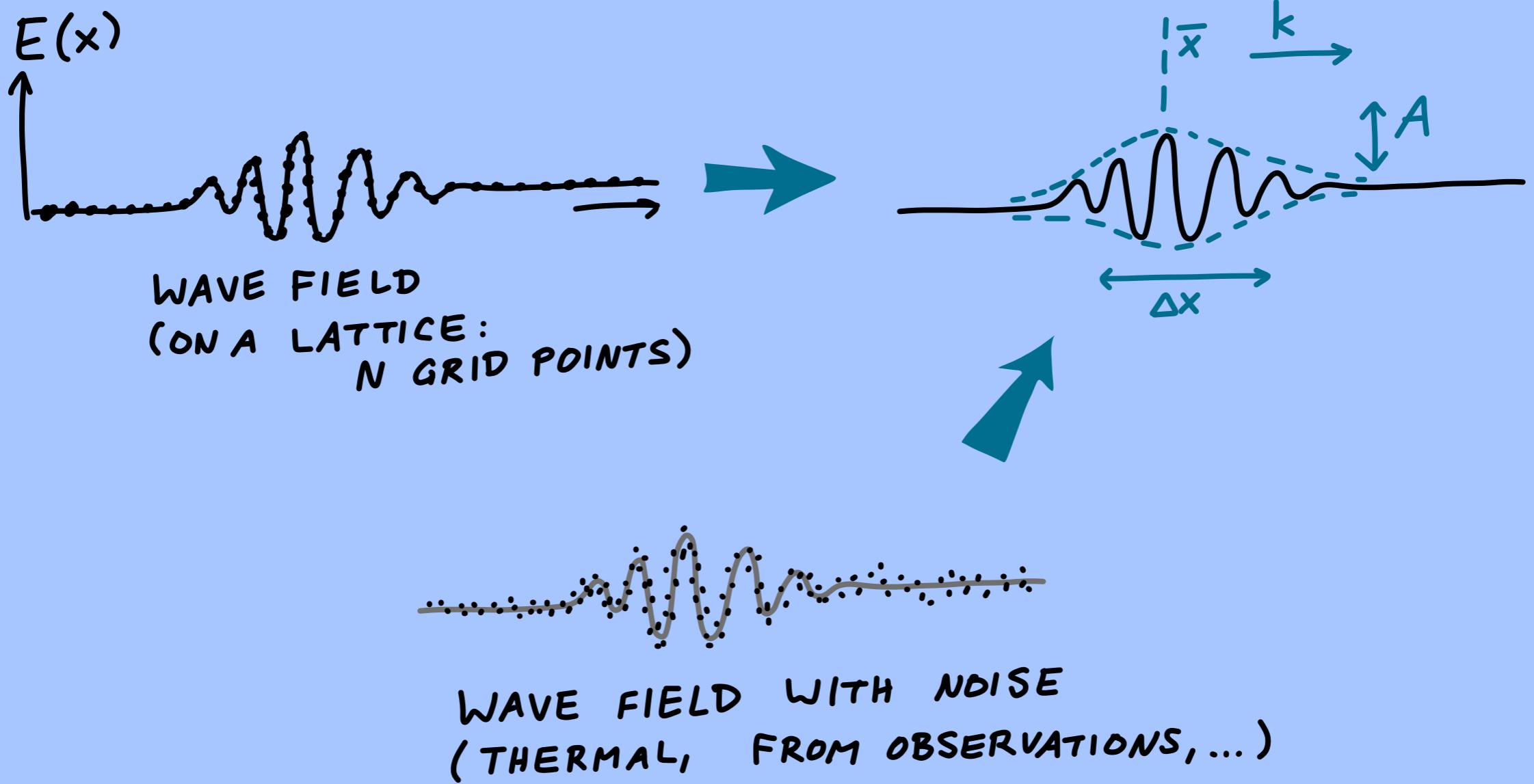
LEAVE AWAY "UNIMPORTANT"  
INFORMATION (ATMOSPHERIC  
BLUR, FIXED STARS, GROUND,...)

# COLLECTIVE COORDINATES

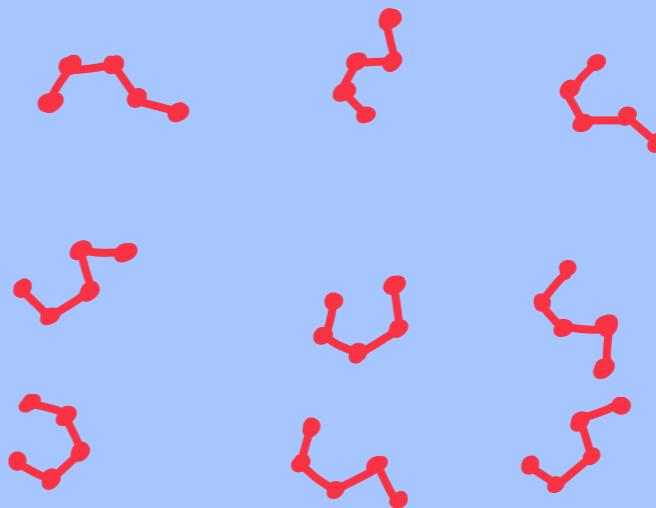
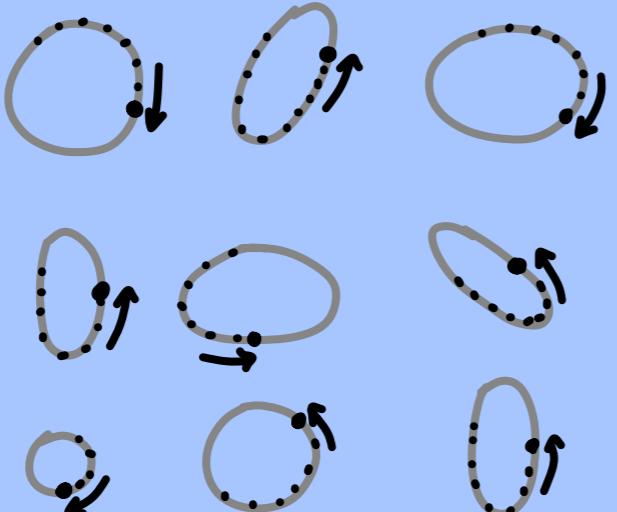
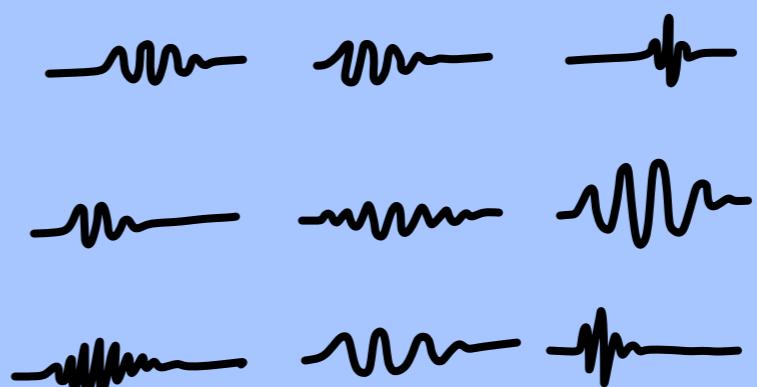
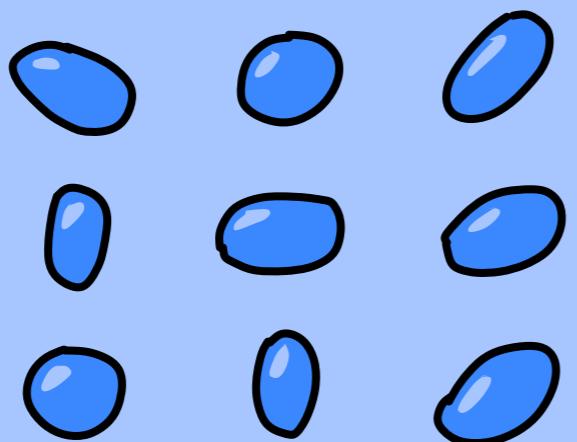


MOLECULE:  
ROTATIONS AROUND  
SOME BONDS , ...

# COLLECTIVE COORDINATES

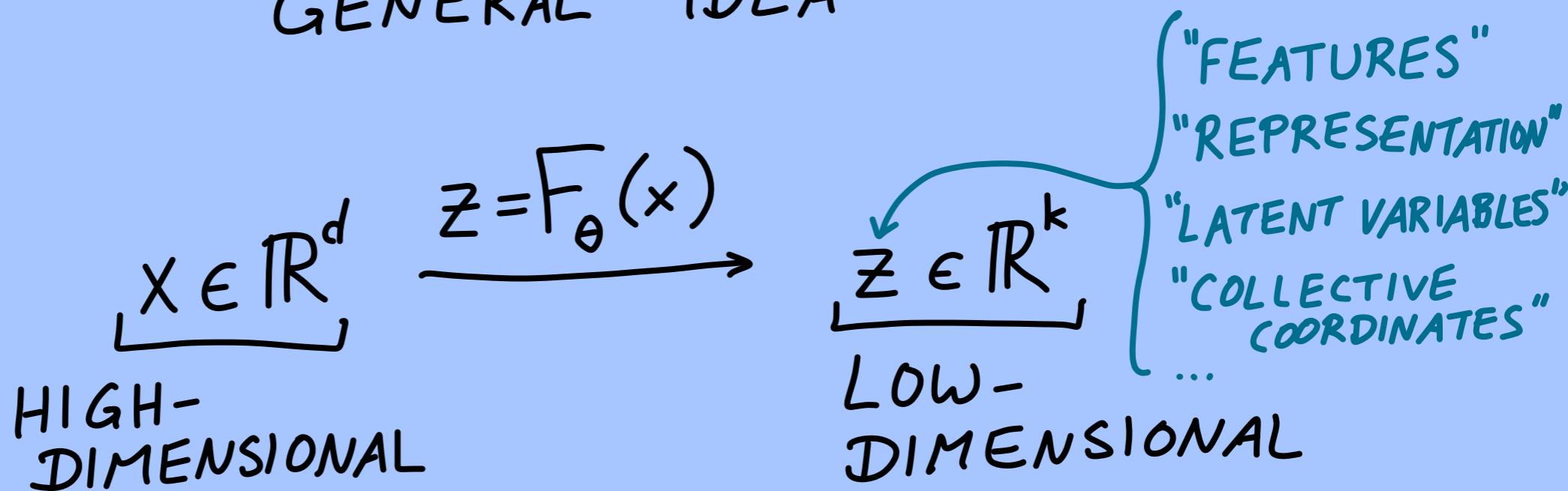


NEED MANY SAMPLES TO  
EXTRACT IMPORTANT FEATURES



FOCUS ON FEATURES THAT VARY !  
(FIXED FEATURES ARE PART OF THE MODEL)

# FEATURE EXTRACTION: GENERAL IDEA



ALL PARTICLE COORDINATES

$$x = \begin{bmatrix} \text{blue oval} \end{bmatrix} \longrightarrow z = (2.5, 3.2, -1.7, \dots, 5.0)$$

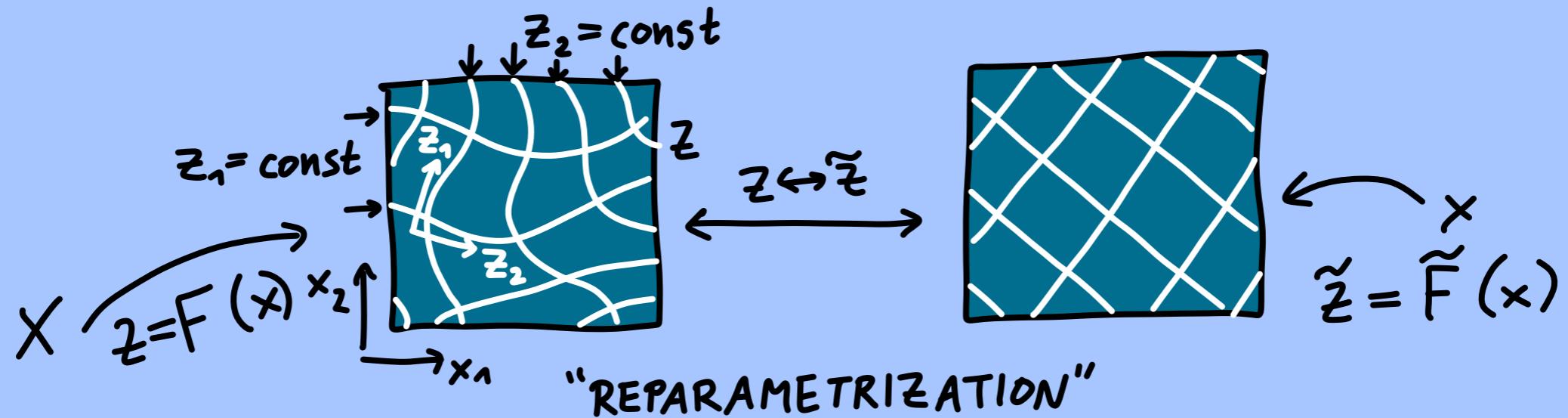
$$x = \text{blue oval} \longrightarrow z = (1.2, 2.4, -0.4, \dots, 3.2)$$

$$x = \text{blue oval} \longrightarrow z = (0.7, 1.3, 0.2, \dots, 2.4)$$

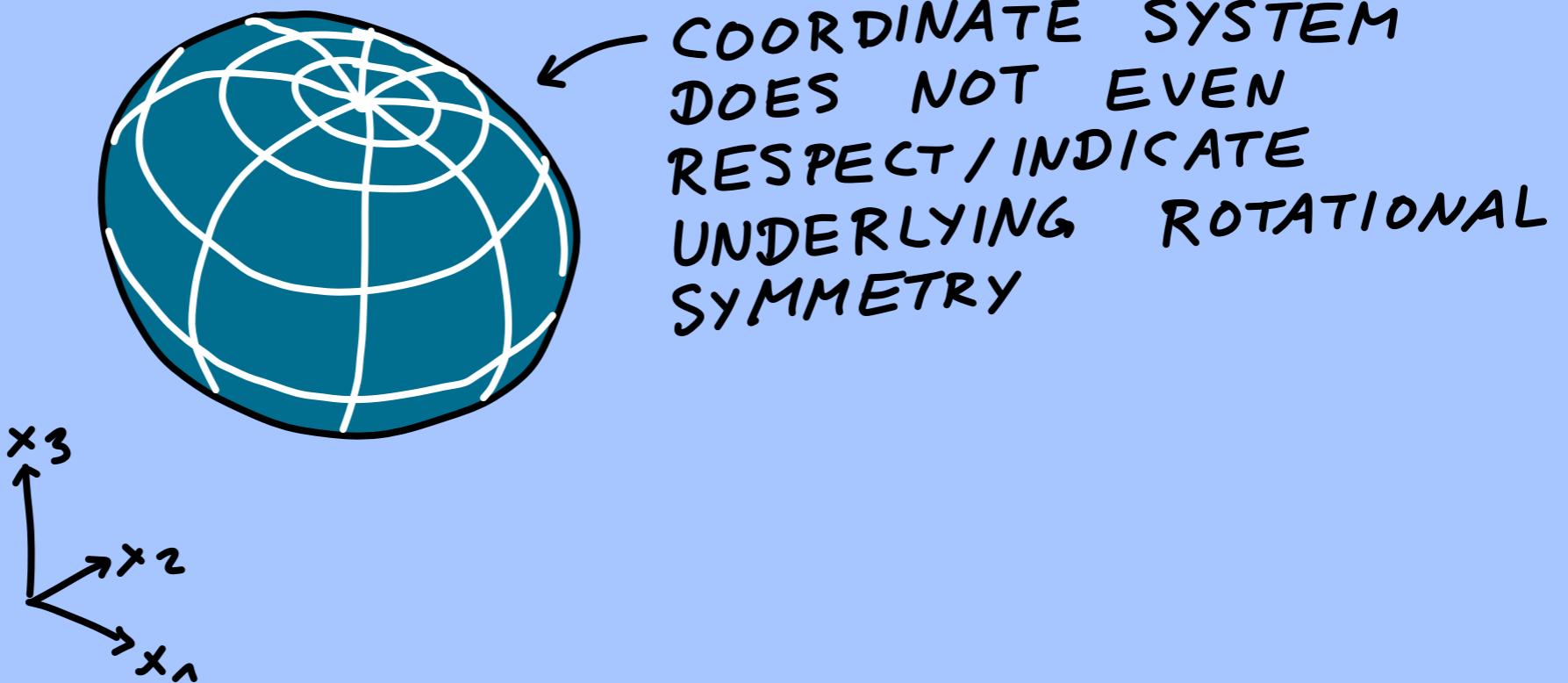
⋮

COLLECTIVE COORDINATES

# COORDINATE CHOICES



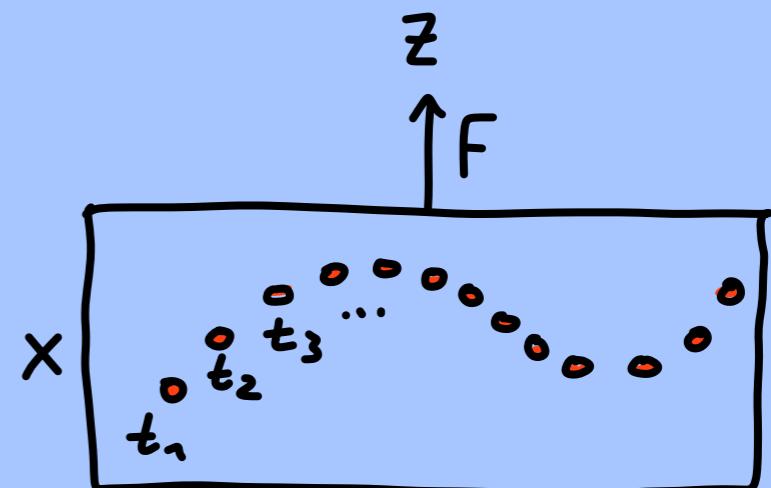
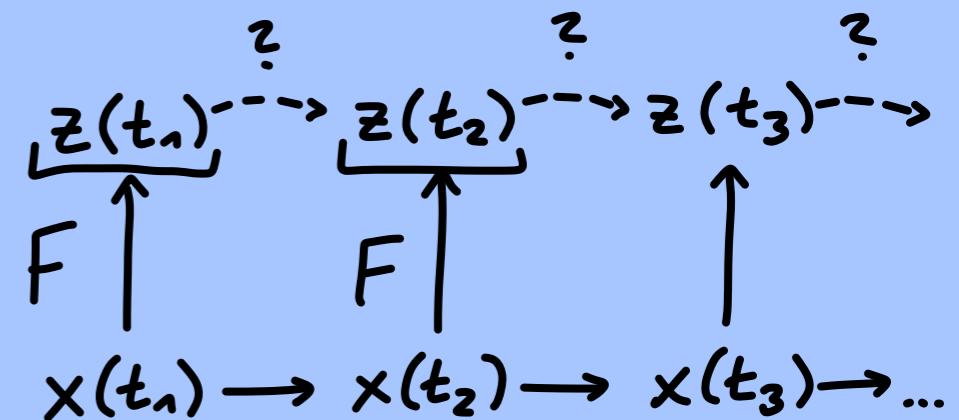
OFTEN: NO "OPTIMAL" CHOICE !



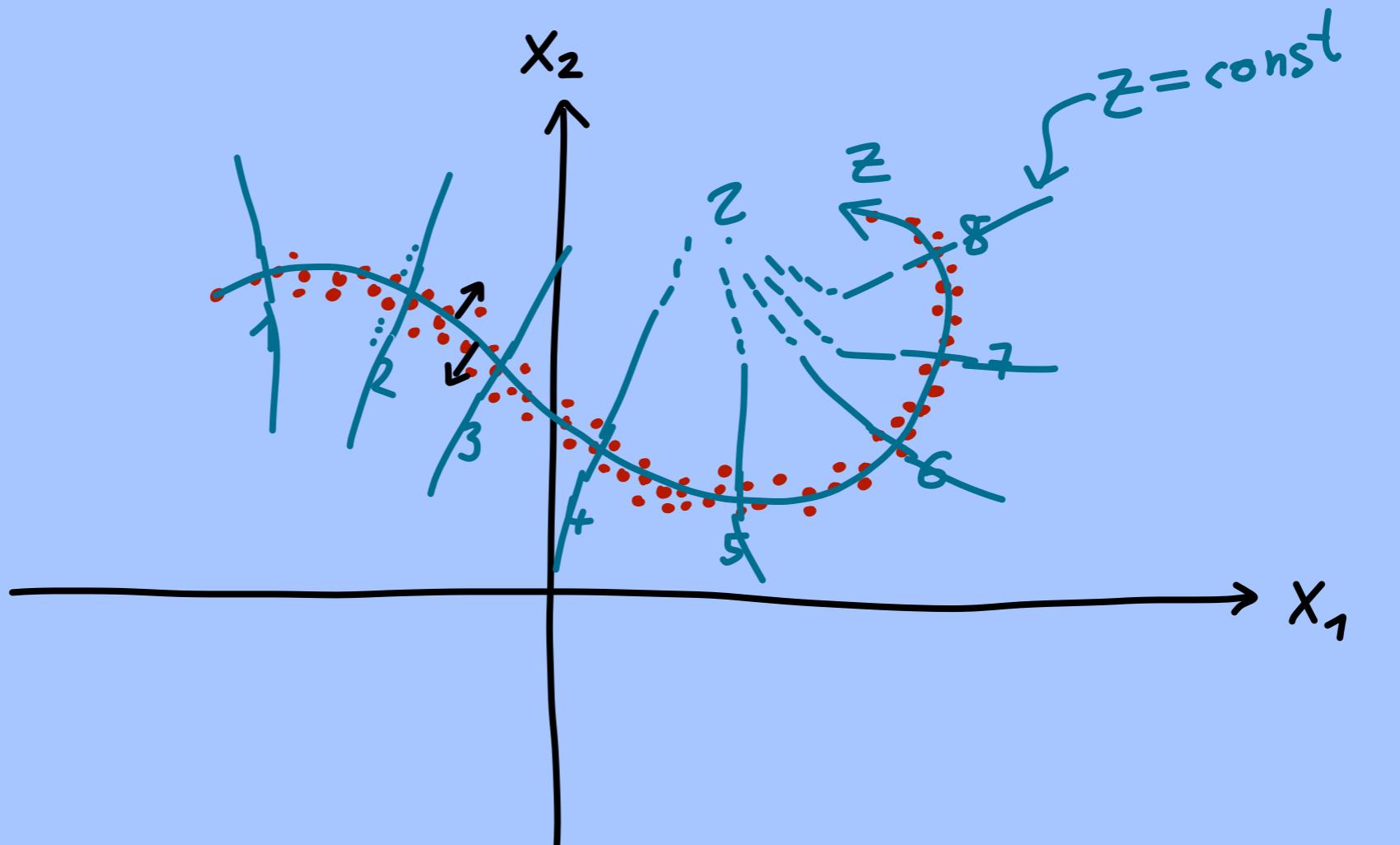
# DYNAMICS : TWO CHOICES

$x$  = CONFIGURATION  
AT ANY TIME  
POINT

$x$  = WHOLE  
TRAJECTORY  
(SEQUENCE  
OF CONFIGURATIONS)



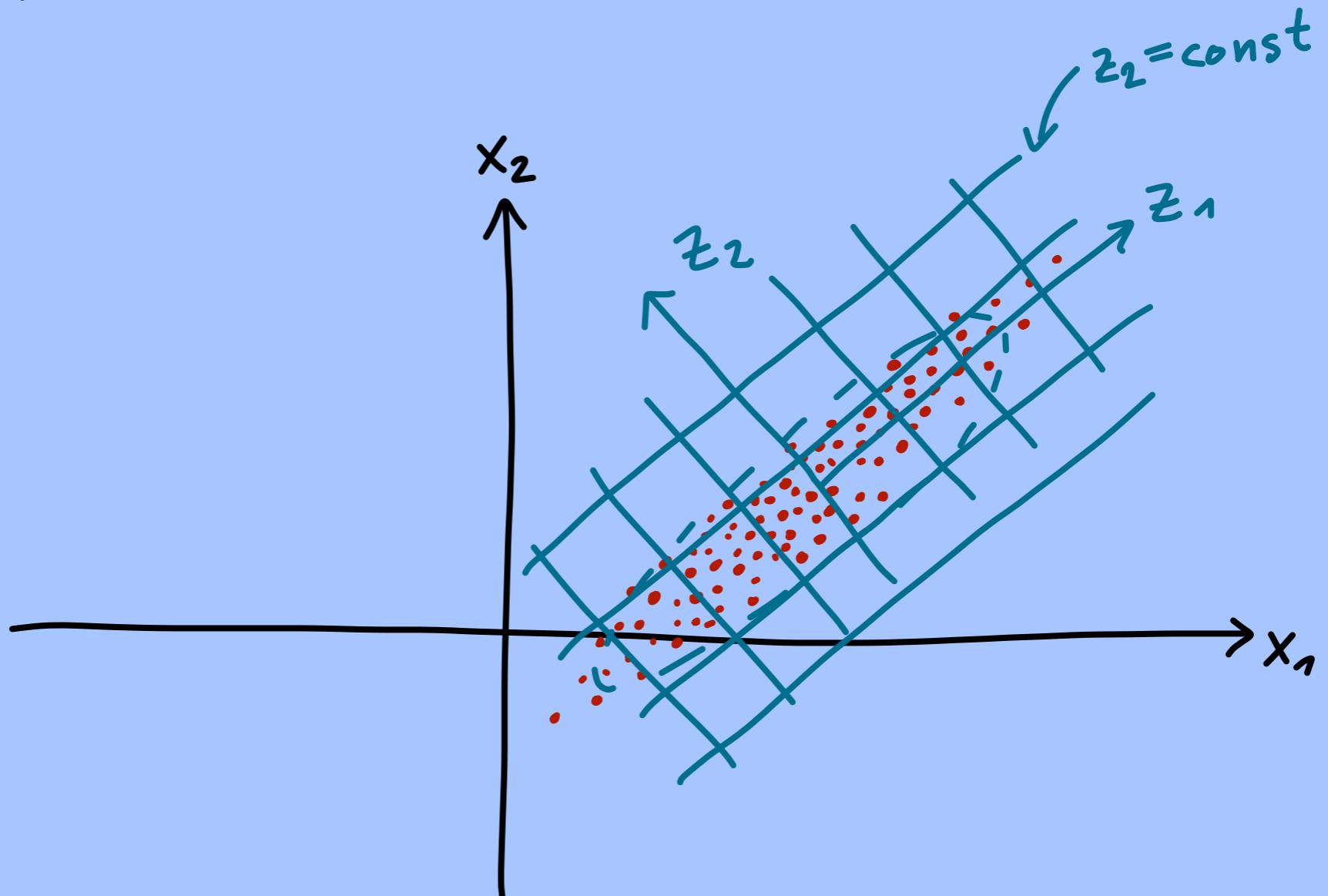
# DROPPING "UNIMPORTANT" ASPECTS

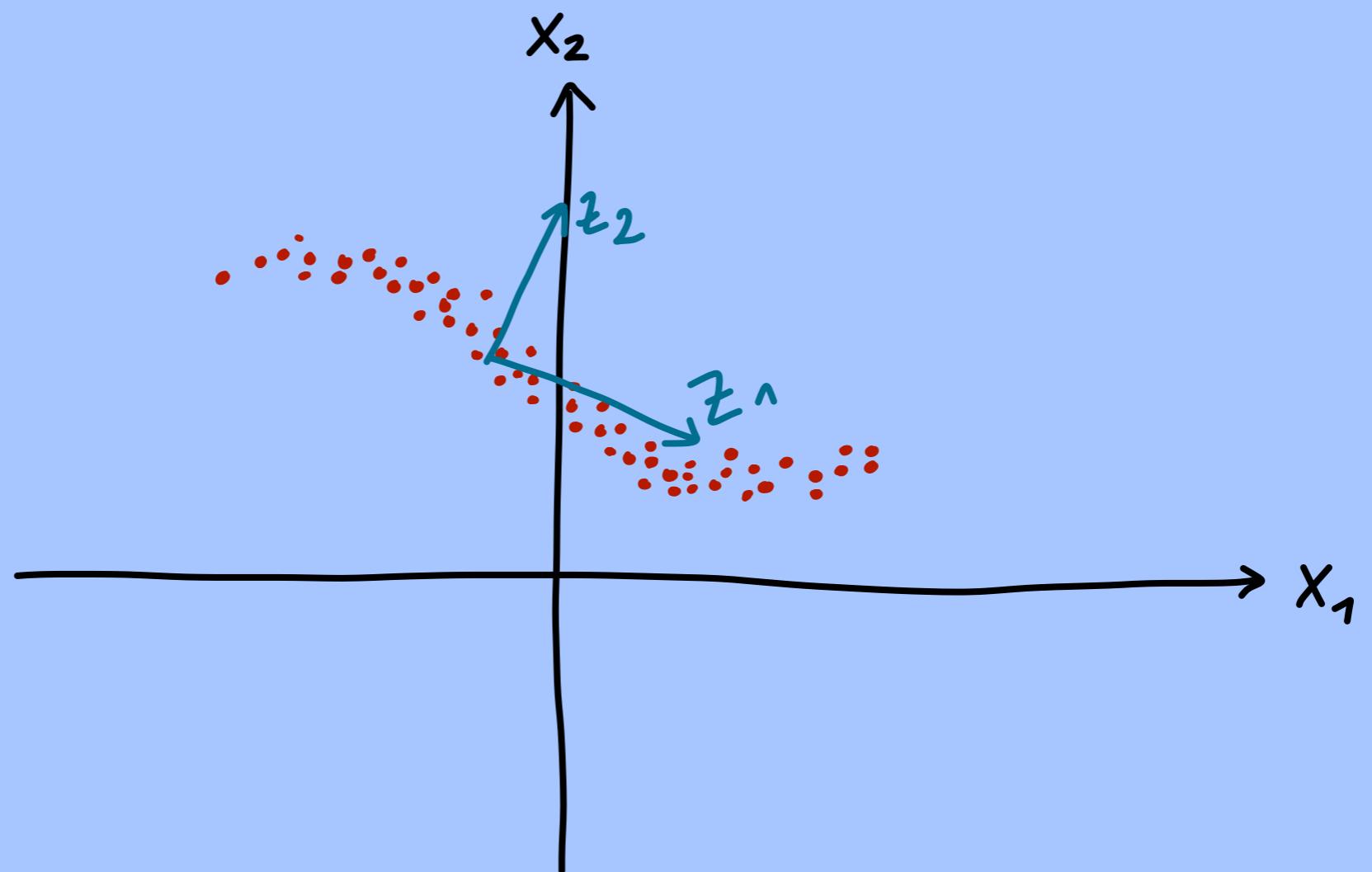


$$z = F(x_1, x_2)$$

3.2

## PRINCIPAL COMPONENT ANALYSIS





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

$$\bar{x} = \langle x \rangle_x = \frac{1}{N} \sum_{\ell=1}^N x^{(\ell)}$$

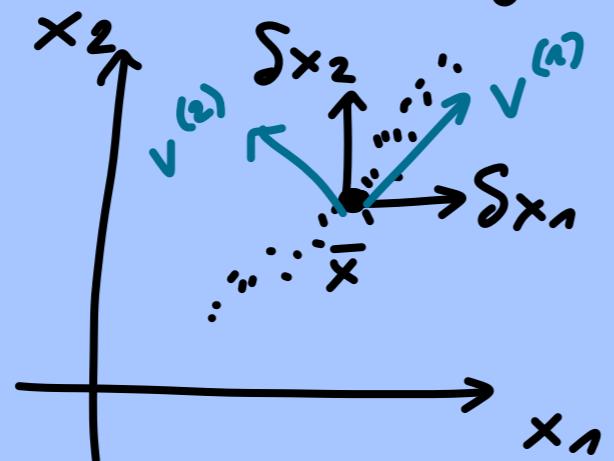
ALL SAMPLES  
IN DATASET

$$\delta_x = x - \bar{x} \quad \text{FLUCTUATIONS}$$

$$1D: \text{Var } x = \langle \delta_x^2 \rangle \geq 0$$

$d > 1$ : COVARIANCE MATRIX

$$[\text{Cov}(x_i, x_j)]_{ij} = \boxed{\langle \delta_{x_i} \delta_{x_j} \rangle = C_{ij}}$$



(SYMMETRIC:  
 $C_{ij} = C_{ji}$ )

HERE:

$$\underbrace{\langle \delta_{x_1} \delta_{x_2} \rangle}_{C_{12}} >> 0$$

$C$  IS POSITIVE SEMIDEFINITE

$$\forall u \in \mathbb{R}^d \quad u^t C u = \sum_{i,j} u_i c_{ij} u_j \geq 0$$

PROOF:

$$\begin{aligned} \dots &= \sum_{i,j} u_i \underbrace{\langle \delta_{x_i} \delta_{x_j} \rangle}_{\geq 0} u_j \\ &= \left\langle \left( \underbrace{\sum_k u_k \delta_{x_k}}_k \right)^2 \right\rangle \geq 0 \end{aligned}$$

DIAGONALIZE  $C$ :

$$C v^{(k)} = \lambda_k v^{(k)}$$

EIGENVECTORS

EIGENVALUES

$$\lambda_k \in \mathbb{R} \quad (C \text{ SYMM.})$$

$$\lambda_k \geq 0 \quad (C \text{ POS. S.D.})$$

$$\{v^{(k)}\} = \{v^{(1)}, v^{(2)}, \dots, v^{(d)}\}$$

CAN BE CHOSEN AS  
ORTHONORMAL BASIS

$$\langle v^{(k')} | v^{(k)} \rangle = \sum_j v_j^{(k')} v_j^{(k)} = \underbrace{\delta_{k,k'}}$$

SCALAR  
PRODUCT

$\Rightarrow$  NEW COORDINATE SYSTEM:

$$z_k = \langle v^{(k)} | x \rangle = \sum_j v_j^{(k)} x_j$$
$$\Rightarrow x = \sum_k z_k v^{(k)}$$

NEW FLUCTUATIONS:

$$\delta z = z - \bar{z}$$

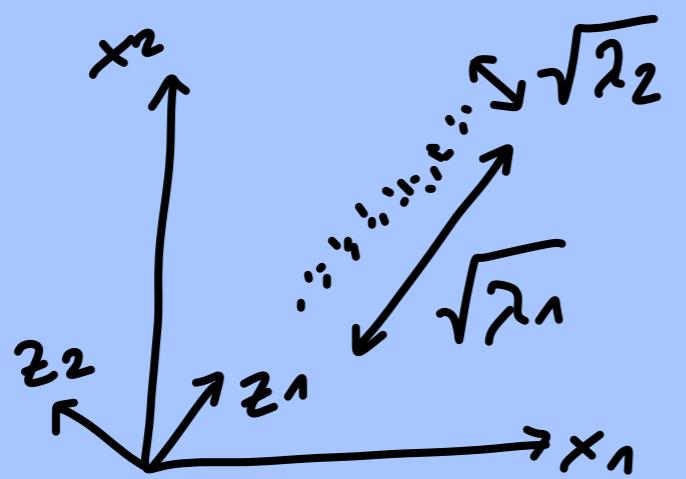
$\Rightarrow$  THESE ARE UNCORRELATED:

$$\langle \delta z_i \delta z_j \rangle = \sum_{i', j'} v_{i'}^{(i)} v_{j'}^{(j)} \underbrace{\langle \delta x_{i'}, \delta x_{j'} \rangle}_{C_{i' j'}}$$

$$C_{v^{(j)}} = \lambda_j v^{(j)}$$

$$= \sum_{i'} v_{i'}^{(i)} \lambda_j v_{i'}^{(j)} = \lambda_j \delta_{ij}$$

$$\langle \delta z_j^2 \rangle = \lambda_j = \text{VARIANCE ALONG EIGEN-DIRECTION}$$



DROP  $z_j$  WITH SMALL  $\lambda_j$

TOTAL VARIANCE

$$\sum_j \langle \delta x_j^2 \rangle = \sum_k \langle S z_k^2 \rangle = \sum_k \lambda_k$$

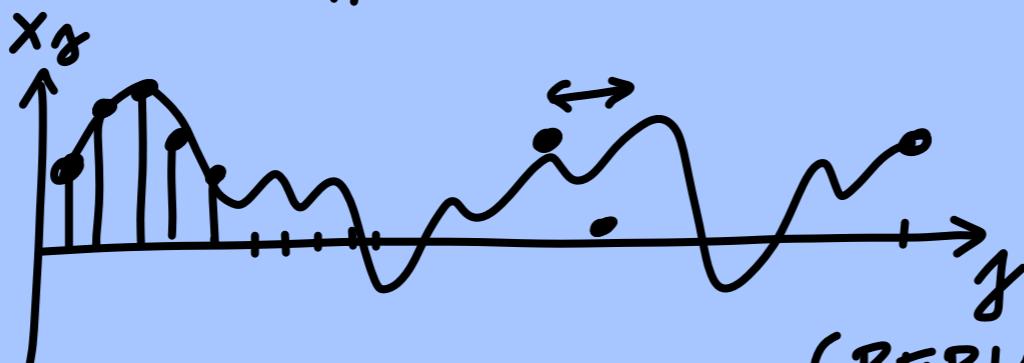
$$= \underbrace{\sum_k \lambda_k}_{\text{"LARGE"}} + \underbrace{\sum_k \lambda_k}_{\text{"SMALL"}}$$

"EXPLAINED VARIANCE"  
WHEN KEEPING  
ONLY  $z_k$  WITH  
LARGE  $\lambda_k$

$$= \frac{\sum_k \lambda_k}{\sum_k \lambda_k + \sum_k \lambda_k}$$

} FRACTION

EXAMPLE: TRANSLATIONALLY  
 (STATISTICALLY)  
 INVARIANT FIELDS



(PERIODIC  
 BOUNDARY CONDITIONS)

$$C_{kj} = C(k - j)$$

DIAGONALIZED BY PLANE WAVES

$$V_k^{(k)} = \frac{1}{\sqrt{N}} e^{ikx}$$

$$\text{NOW } \langle u | v \rangle = \sum_j u_j^* v_j$$

$$x \in \mathbb{R}^d \Rightarrow z^{(-k)} = z^{(k)*}$$

EIGENVALUES:

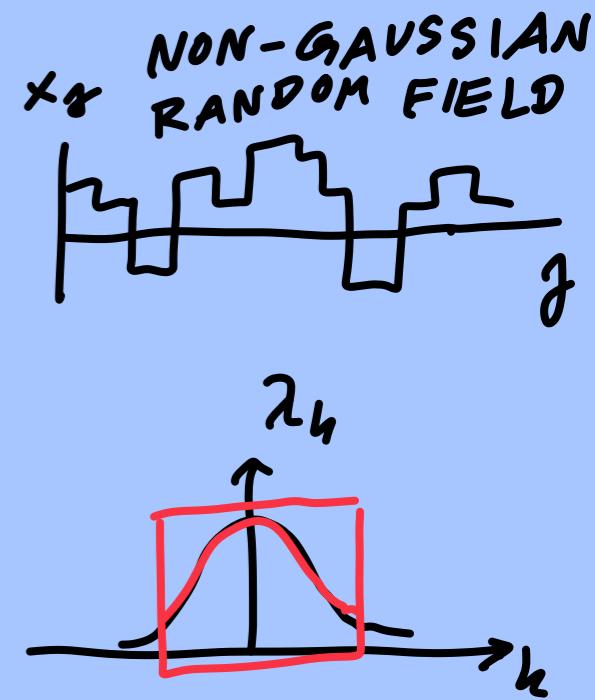
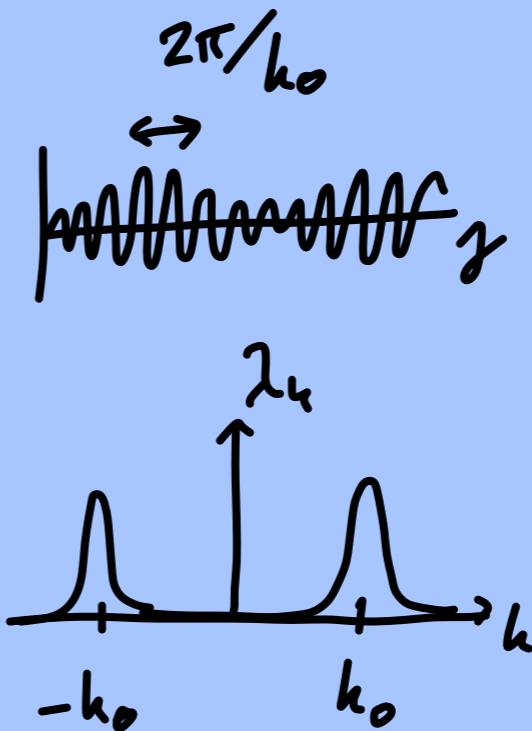
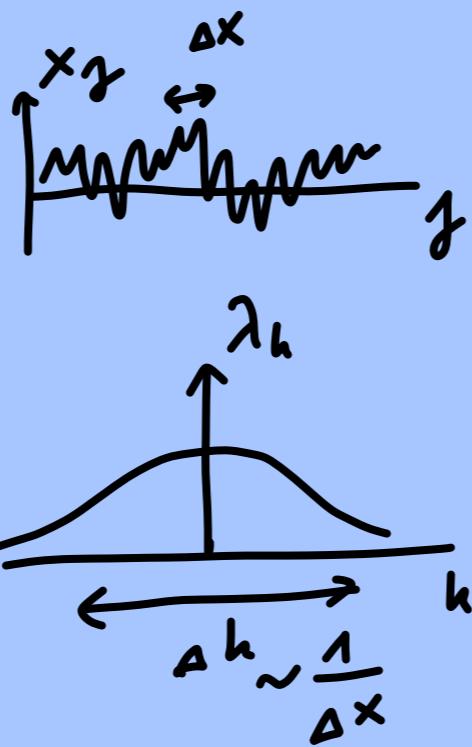
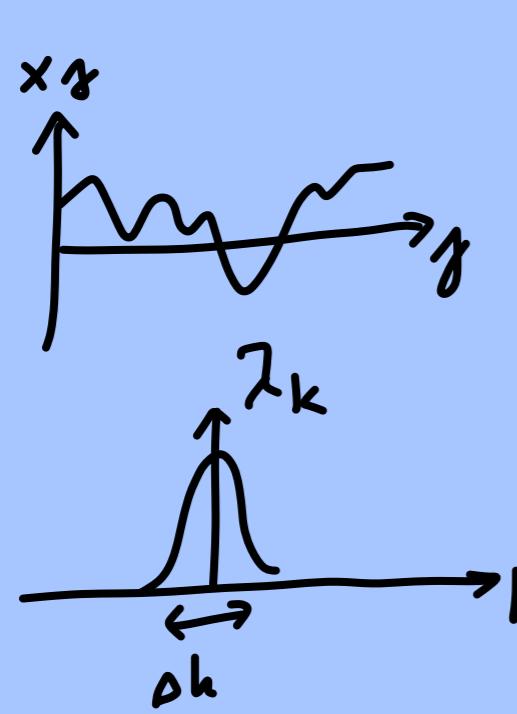
$$C_V^{(k)} = \lambda_k v^{(k)}$$

$$\Rightarrow \lambda_k = \langle v^{(k)} | C_V^{(k)} \rangle$$

$$= \frac{1}{N} \sum_{\ell, j} e^{-ik\ell} C_{\boxed{\ell-j}} e^{ikj}$$

$$\lambda_k = \sum_{\ell} e^{-ik\ell} C(\ell)$$

"POWER SPECTRUM"



$$\sum_k A_k e^{iky}$$

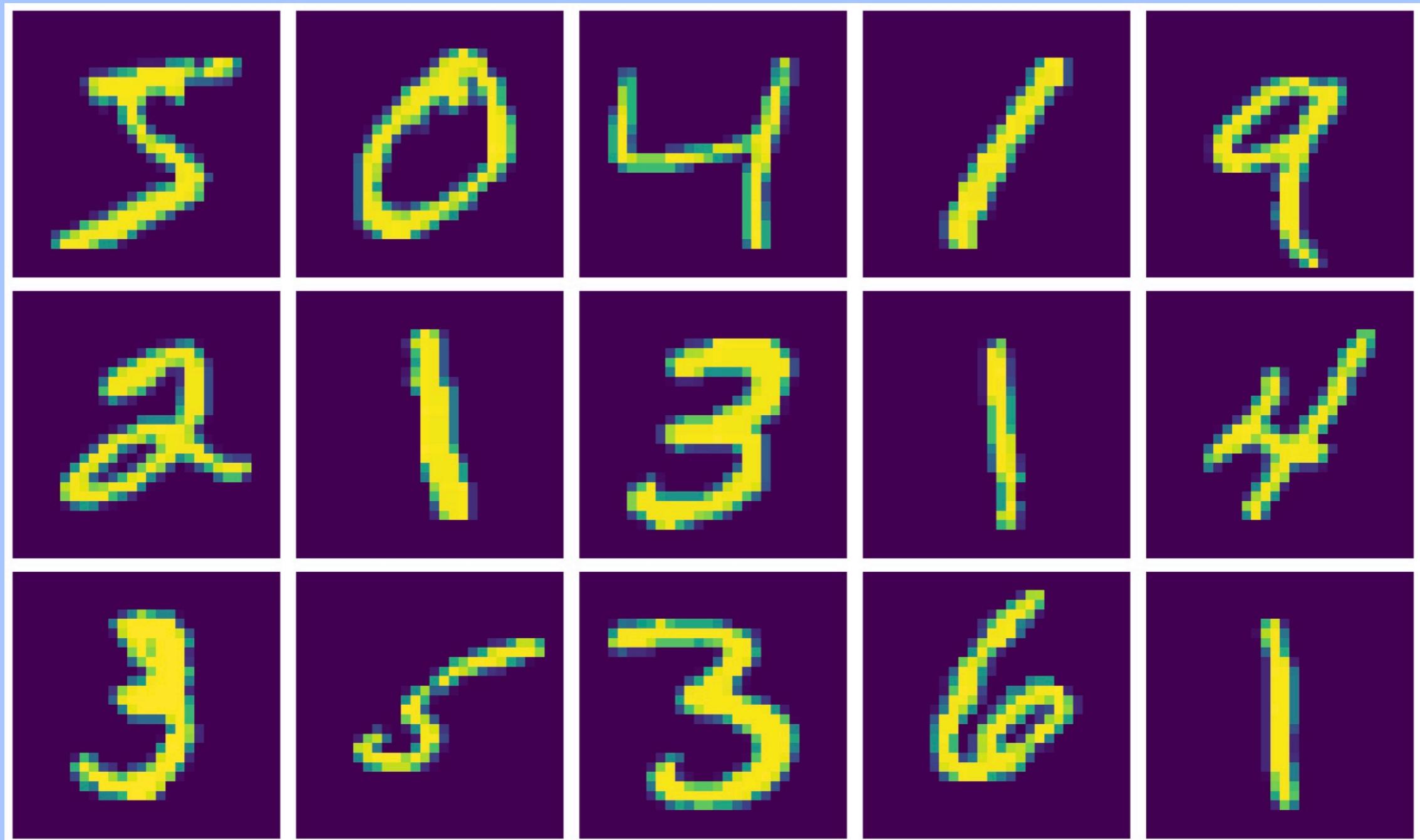
RANDOM GAUSSIAN

GAUSSIAN RANDOM FIELD

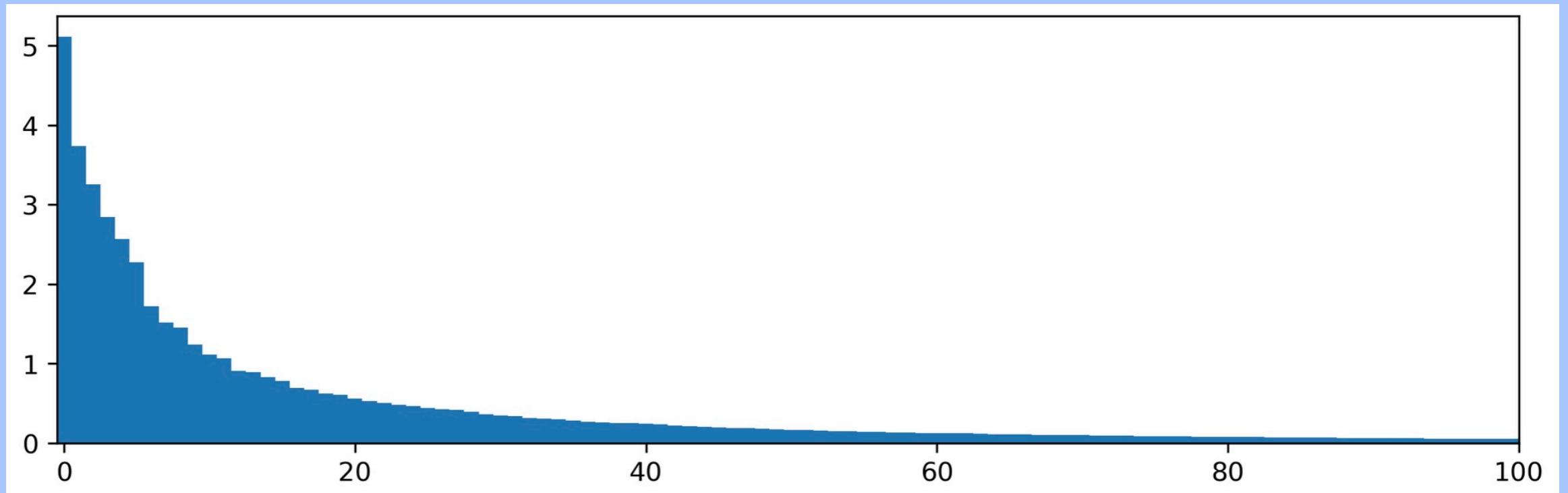
# PCA APPLIED TO MNIST $\langle s_{x_j} s_{x_\ell} \rangle = C_{j\ell}$

MNIST = DATA SET OF HANDWRITTEN DIGITS  
(60 000)

$$x \in \mathbb{R}^{748}$$

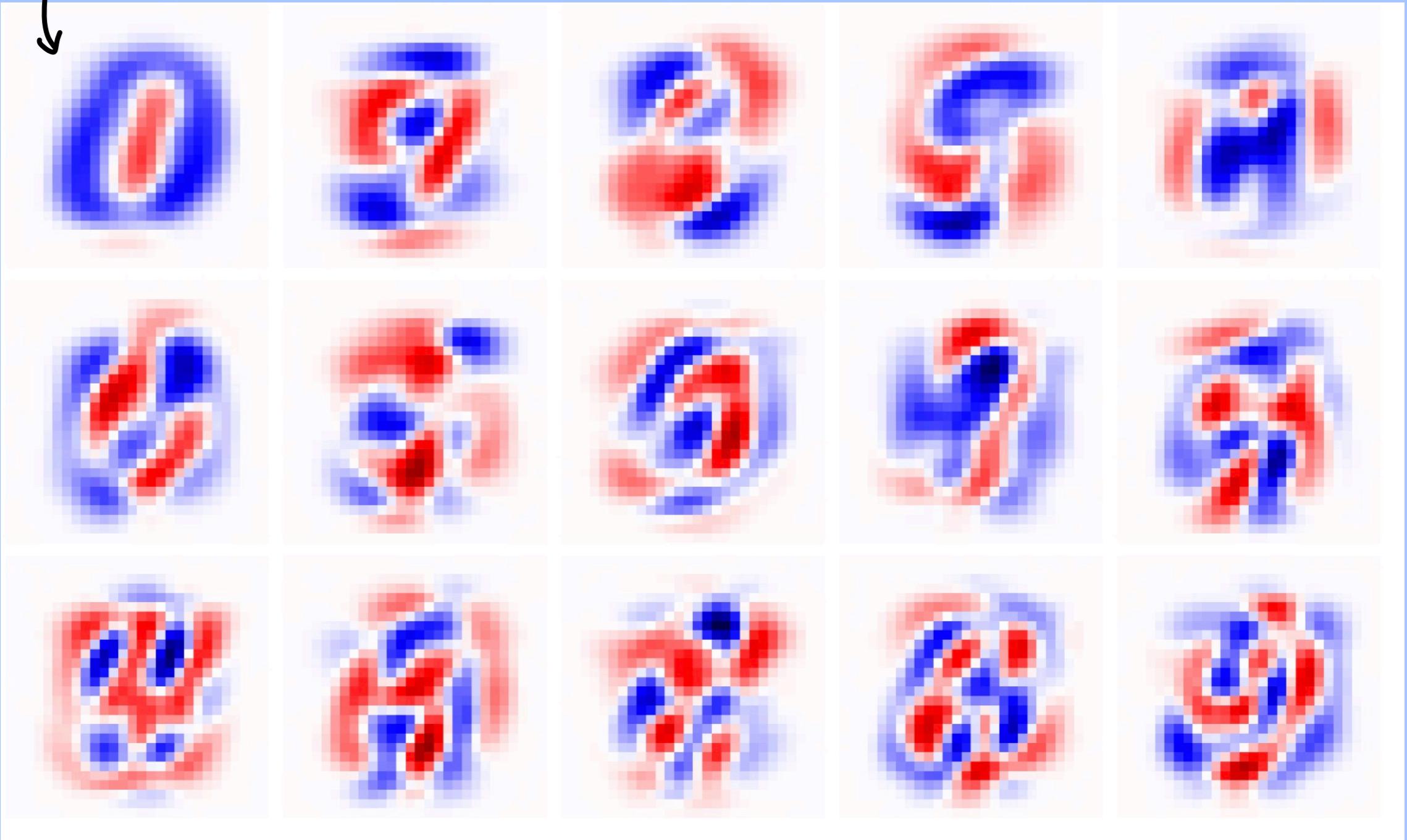


# EIGENVALUES OF COVARIANCE MATRIX



LARGEST EIGENVALUE

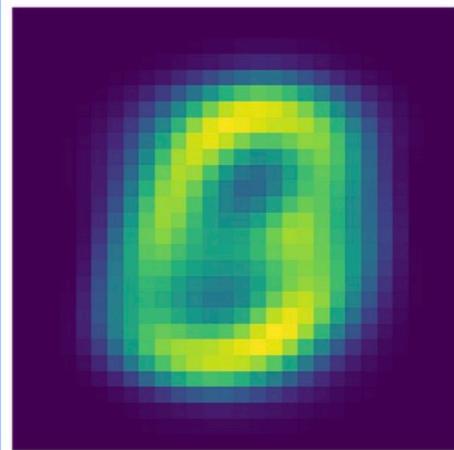
# EIGENVECTORS



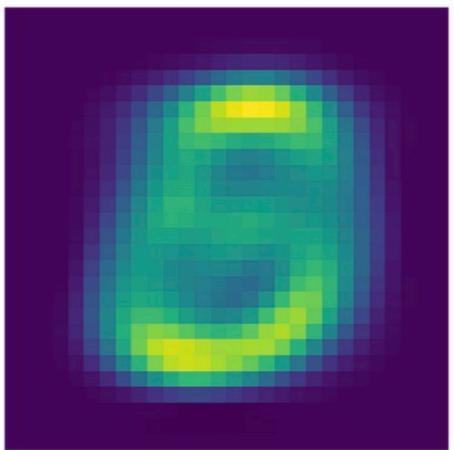
# SUCCESSIVE RECONSTRUCTION

$$x = \bar{x} + \sum_{k=1}^M v^{(k)} \langle v^{(k)} | x - \bar{x} \rangle$$

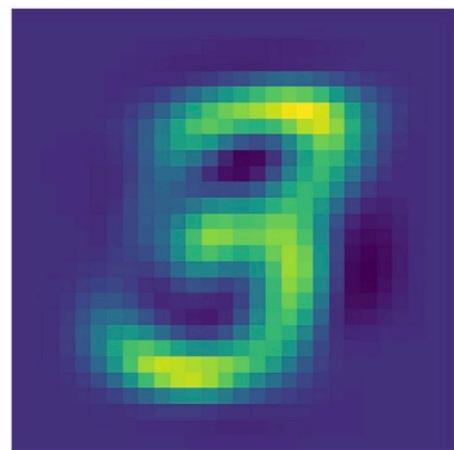
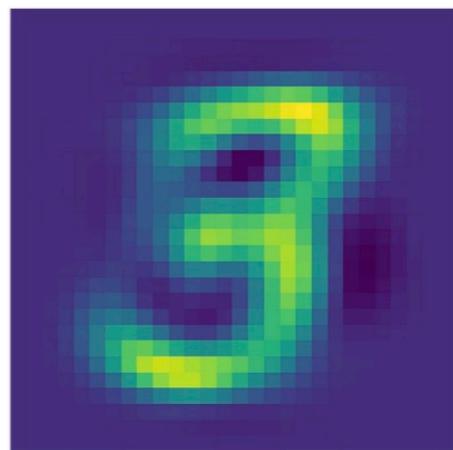
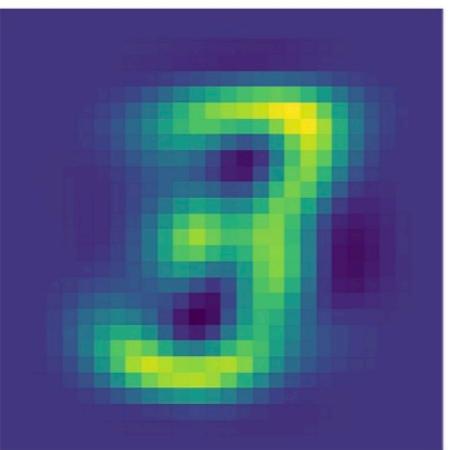
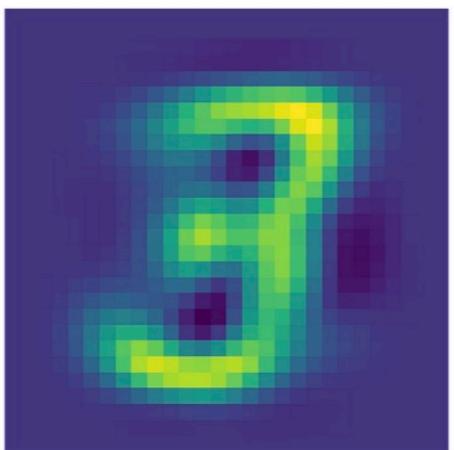
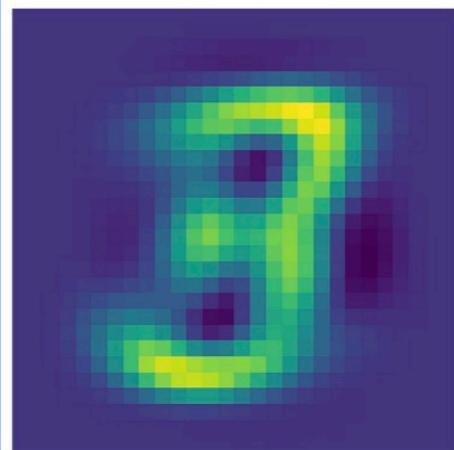
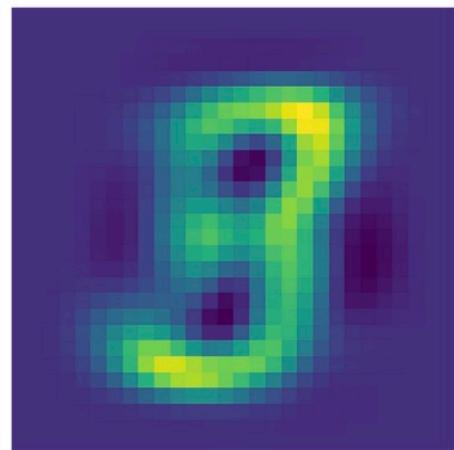
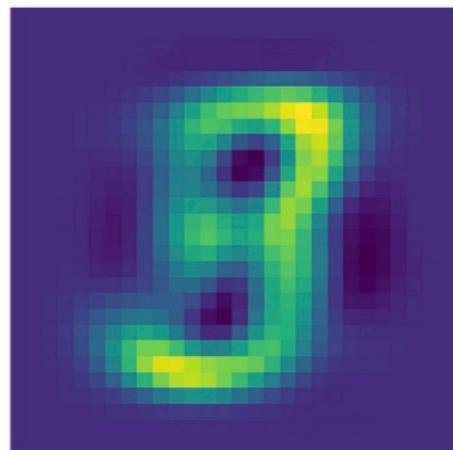
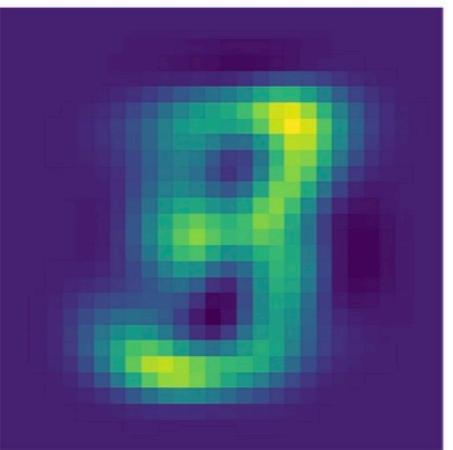
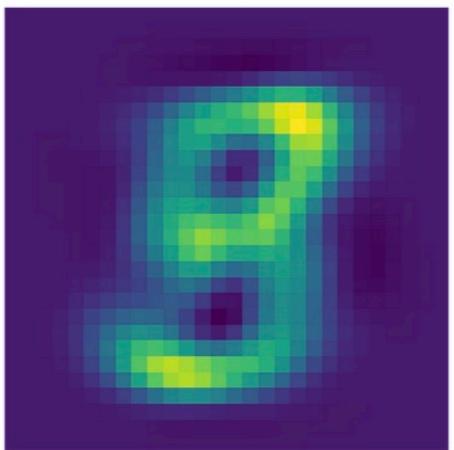
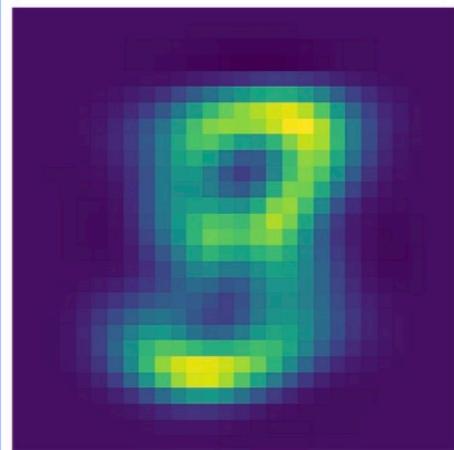
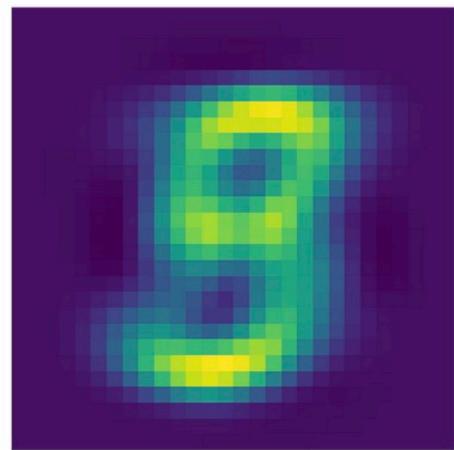
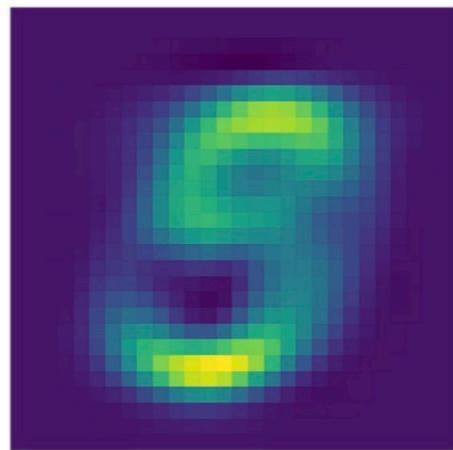
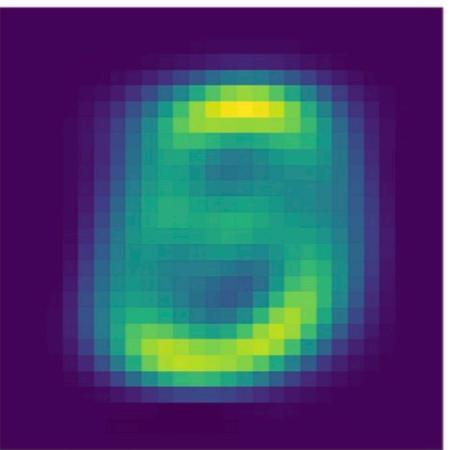
$M=1$

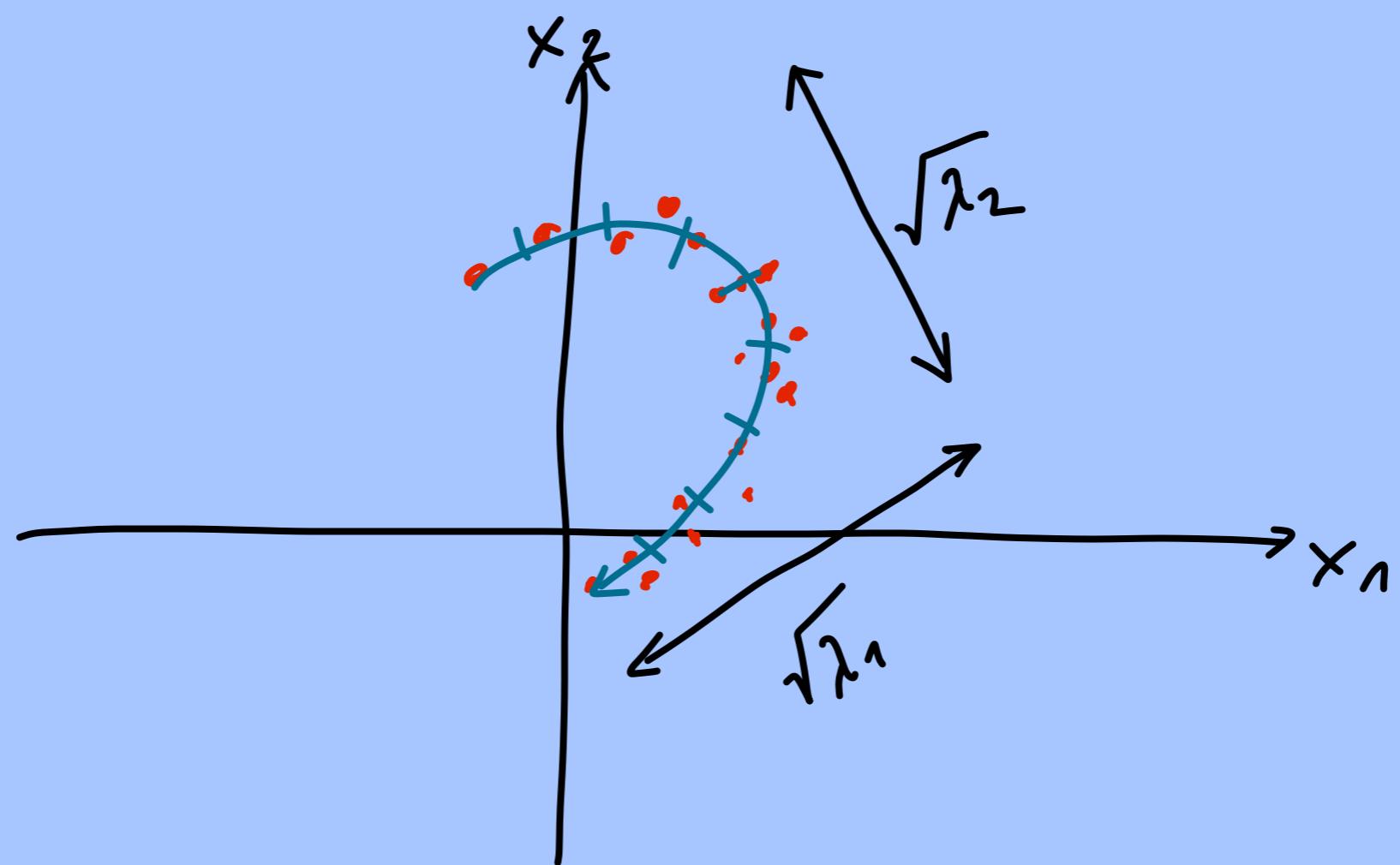


$M=2$



$M=3$





3.3

# AUTOENCODERS

3.3.1

# BASIC AUTOENCODER

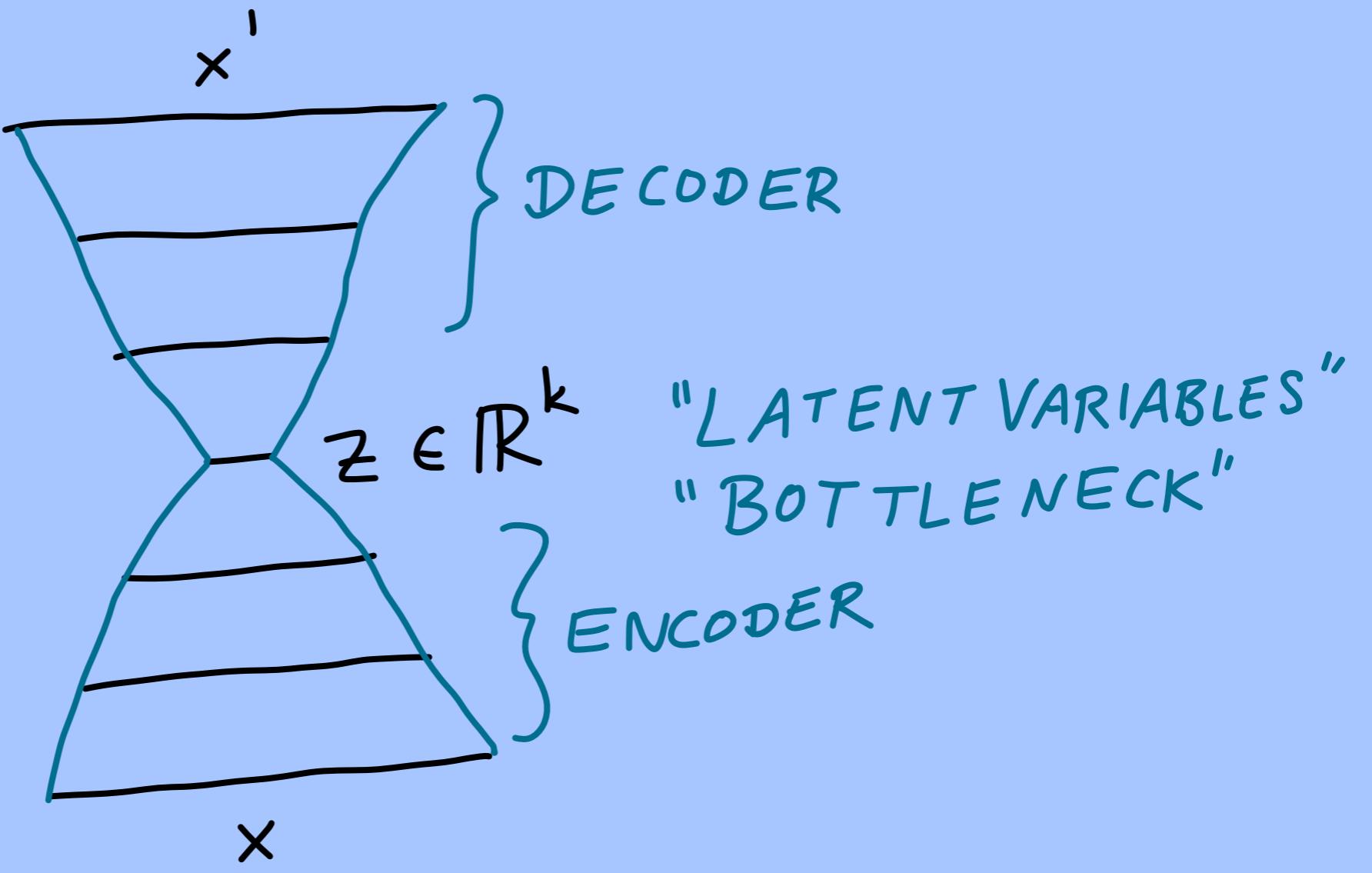
(~1986ff)

IDEA:  $x \xrightarrow{E} z \xrightarrow{\mathcal{D}} x' \approx x$  REPRODUCE INPUT

$\mathbb{R}^d$        $\mathbb{R}^k$        $\mathbb{R}^d$

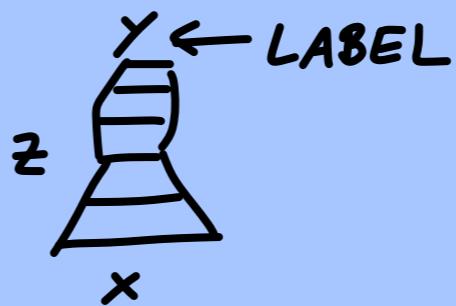
$x' = F_{\theta}(x) = \mathcal{D}_{\theta_D}(E_{\theta_E}(x))$

DECODER      ENCODER



GOALS:

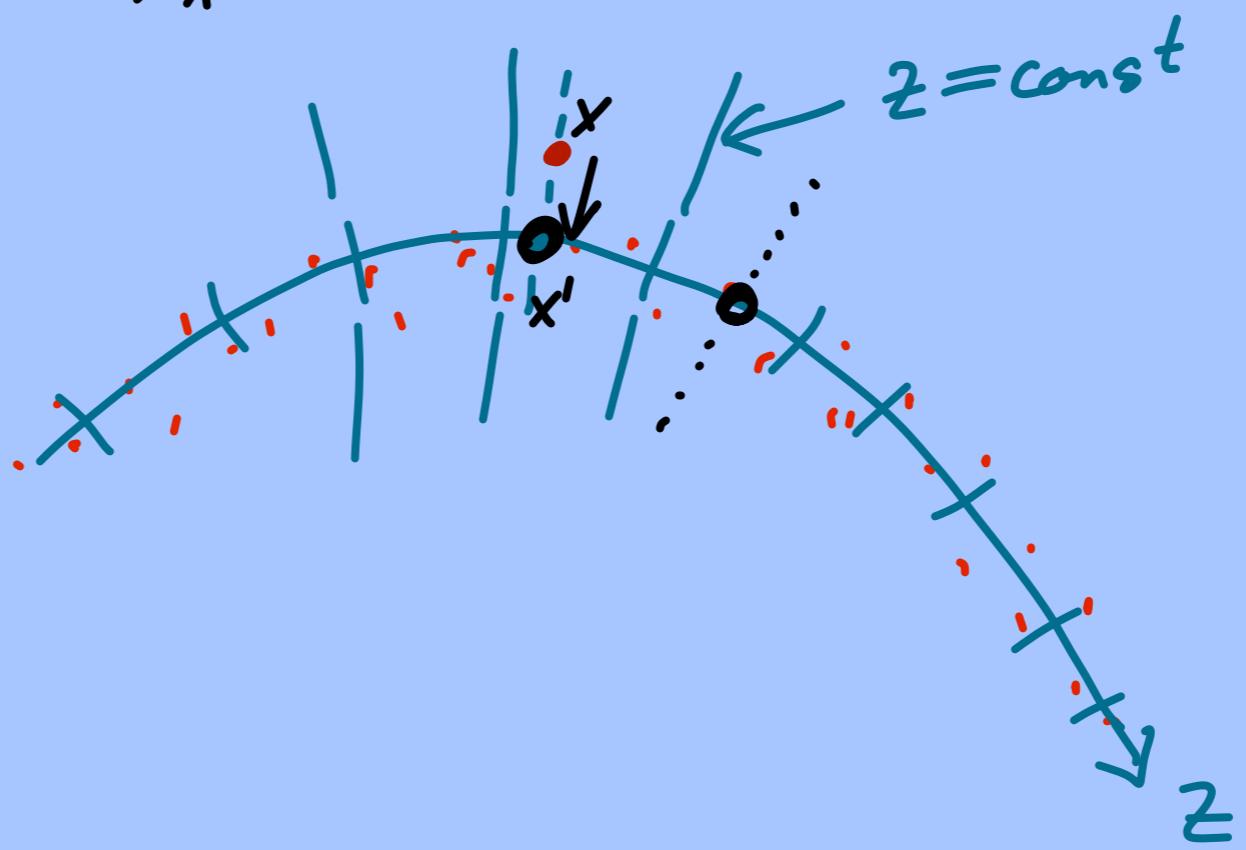
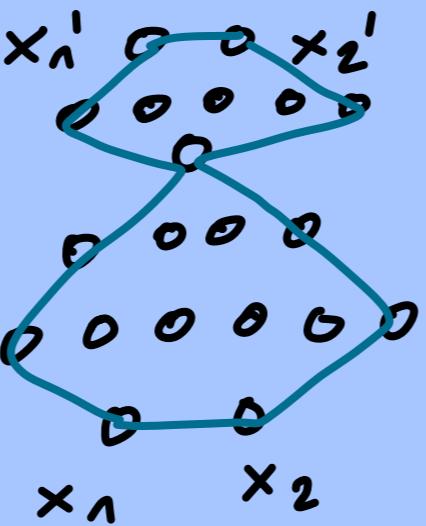
- DIMENSIONALITY REDUCTION
- USE  $z$  TO TRAIN CLASSIFIER

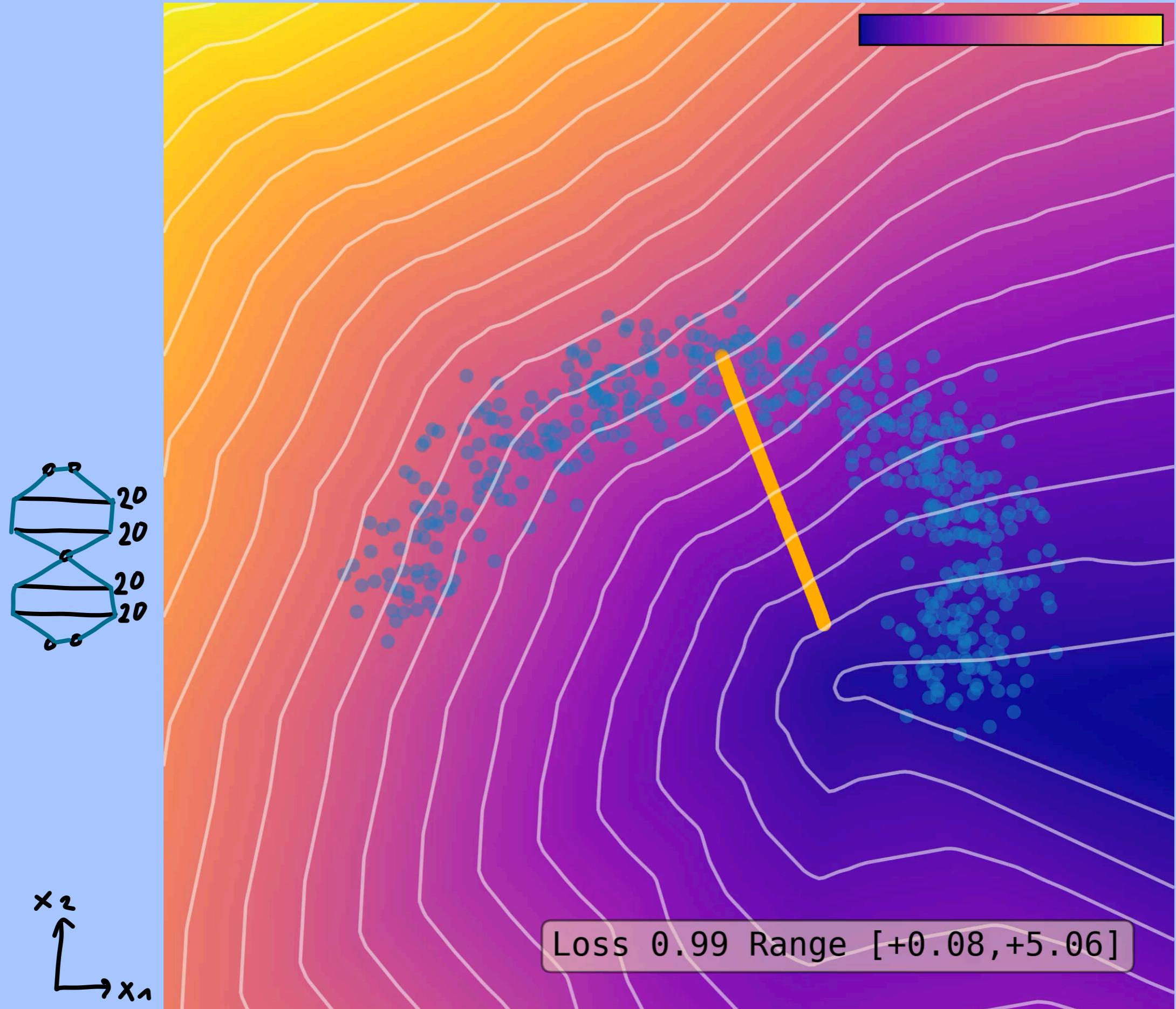


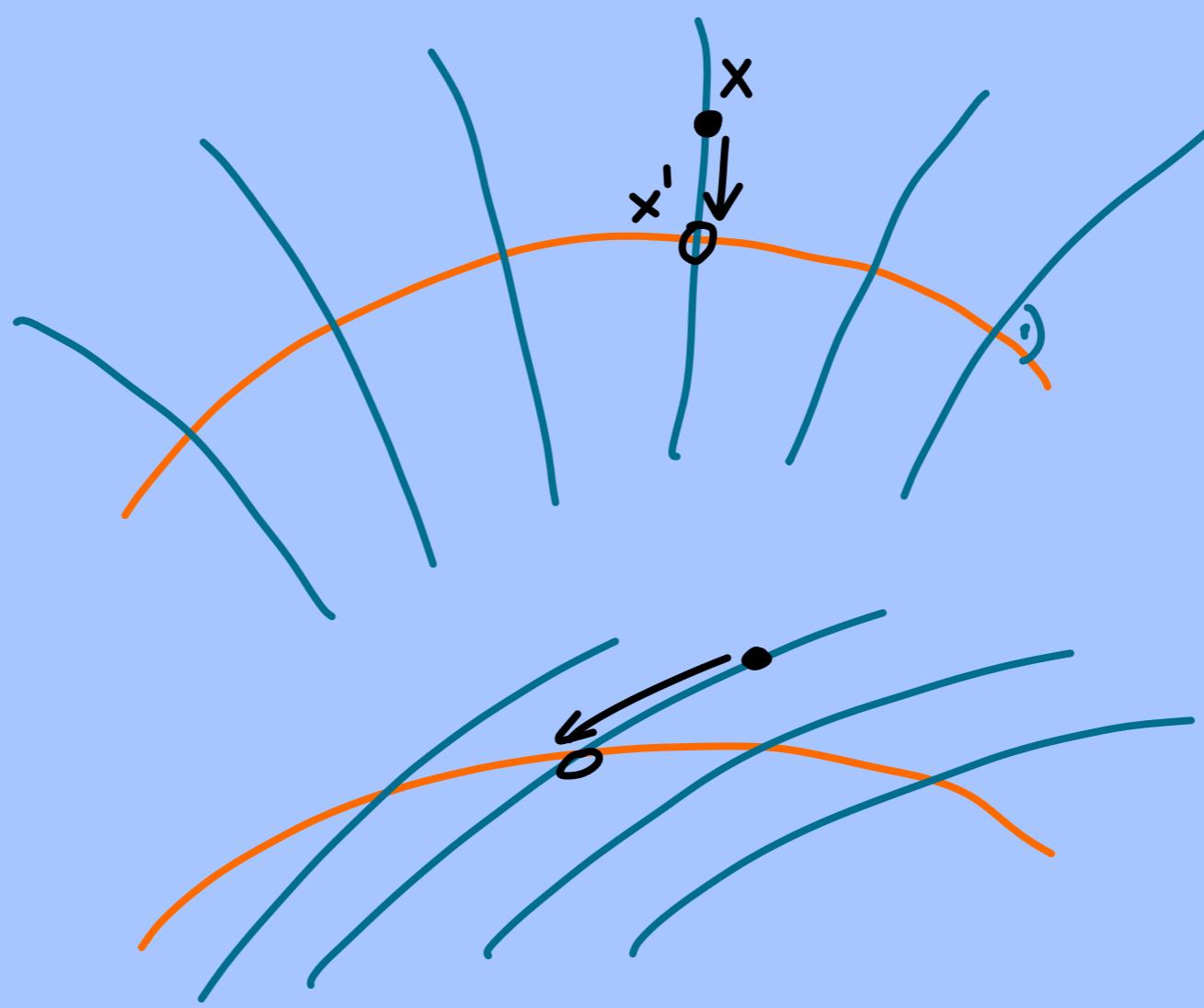
KEEP "MOST IMPORTANT FEATURES"  
(HERE: MOST IMPORTANT  
FOR GOOD RECONSTRUCTION)

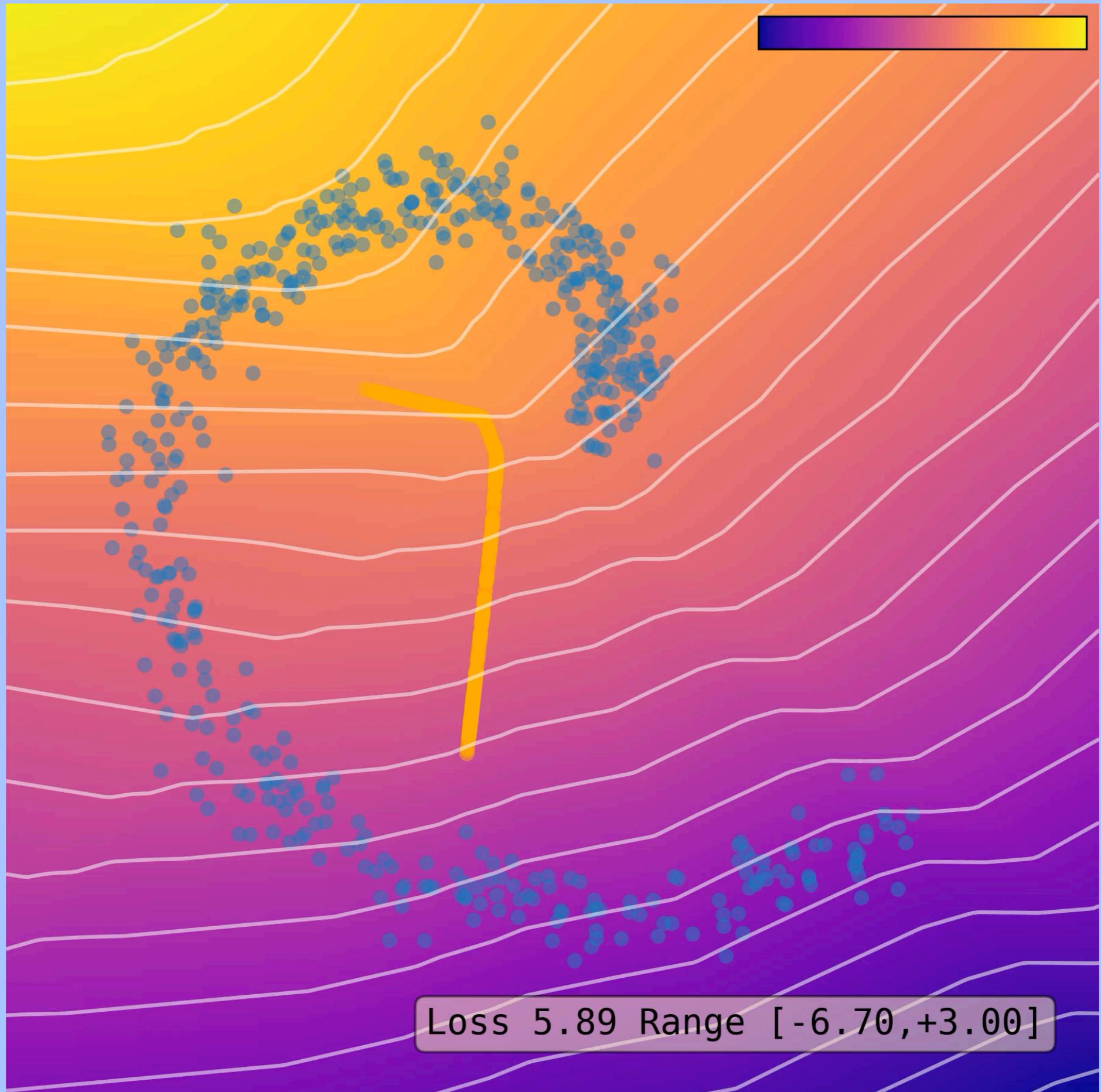
$$\begin{aligned}\mathcal{L}(x, \theta) &= \|x' - x\|^2 \\ &= \|F_\theta(x) - x\|^2 \\ &= \sum_j (x'_j - x_j)^2\end{aligned}$$

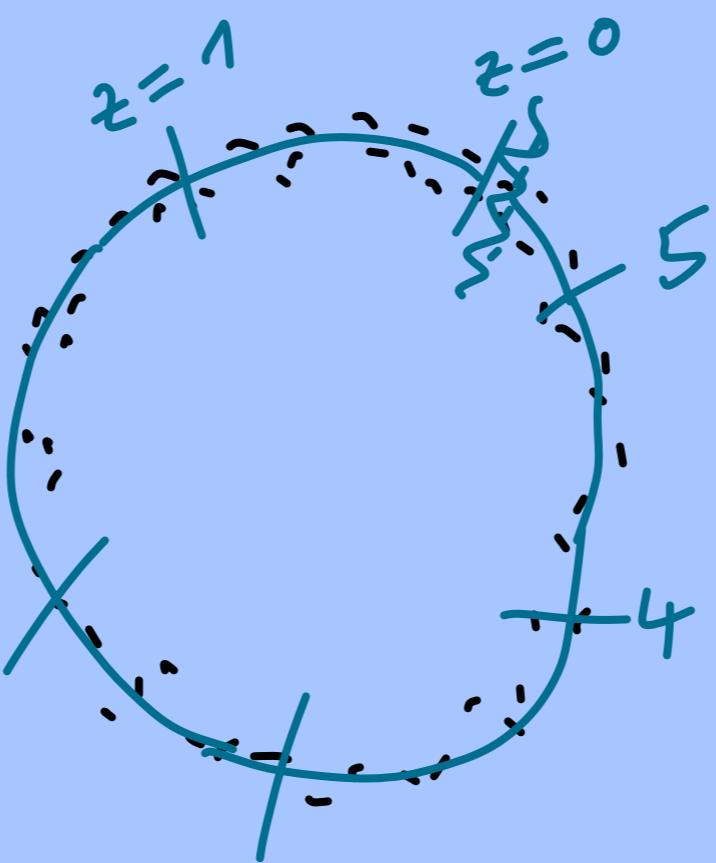
$x \in \mathbb{R}^2$   
 $z \in \mathbb{R}^1$







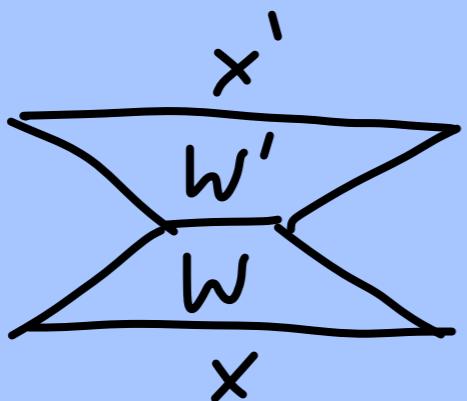




### 3.3.2

## CONNECTION TO PCA

### LINEAR AUTOENCODER



LET  $\langle x \rangle = 0$   
SET BIASES = 0

$$x' = \underbrace{W' \circ W}_{\mathbb{R}^{d \times h} \cap \mathbb{R}^{k \times d}} x \in \mathbb{R}^{d \times d}$$

HAS RANK  
AT MOST k  
dim (SUBSPACE  
NOT MAPPED  
TO ZERO)

(COULD CHOOSE  $W' = W^T$   
 → SYMMETRIC SITUATION  
 → "WEIGHTS ARE TIED")

FIND  $W'W$  SUCH THAT  
 $\langle \|x' - x\|^2 \rangle_x = \text{MIN}$

USE PCA AND EXPAND

$$x_j = \sum_{\ell} z_{\ell} v_j^{(\ell)}$$

$$\Rightarrow x' = \sum_{\ell} z_{\ell} (\underbrace{w' w}_{\perp}) v^{(\ell)}$$

$$\langle \|x' - x\|^2 \rangle = \left\langle \left\| \sum_{\ell} z_{\ell} (w' w - \perp) v^{(\ell)} \right\|^2 \right\rangle_x$$

$$\bar{z} \sum_{\ell} \underbrace{\langle z_{\ell}^2 \rangle}_{\lambda_{\ell} \text{ [PCA]}} \underbrace{\|(w' w - \perp) v^{(\ell)}\|^2}_{! \text{ MIN}}$$

$$\bar{z} = 0$$

$$z_{\ell} = \sum z_{\ell}$$

$$\langle \sum z_{\ell} \sum z_n \rangle = 0 \text{ FOR } \ell \neq n$$

IDEALLY  $W'W = I_{d \times d} \rightarrow$  IMPOSSIBLE

BEST CHOICE

$$W'W = \begin{pmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 0 & & & \\ & & & & 0 & & \\ & & & & & 0 & \\ & & & & & & \ddots \\ & & & & & & 0 \end{pmatrix} \xrightarrow{k}$$

(ORDERING  
BASIS  
ACCORDING  
TO  $\lambda_i$ )

$$\langle \|x' - x\|^2 \rangle = \sum_{l>h} \langle z_l^2 \rangle = \text{UNEXPLAINED VARIANCE}$$

$$\sum_{\ell} \lambda_{\ell} \| \underset{\text{RANK } k}{\uparrow} (M - 1) v^{(\ell)} \|^2 \stackrel{!}{=} \text{MIN}$$

USE  $v^{(\ell)}$ -BASIS:  $v^{(\ell)} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \end{pmatrix} \leftarrow \ell$

$$\sum_m (M_{nm} - S_{nm}) S_{\ell m} = M_{n\ell} - S_{n\ell}$$

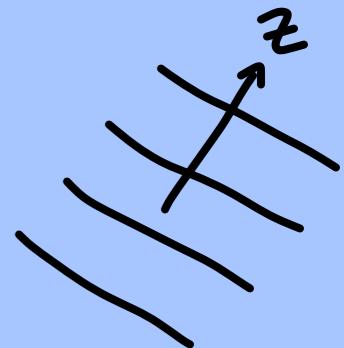
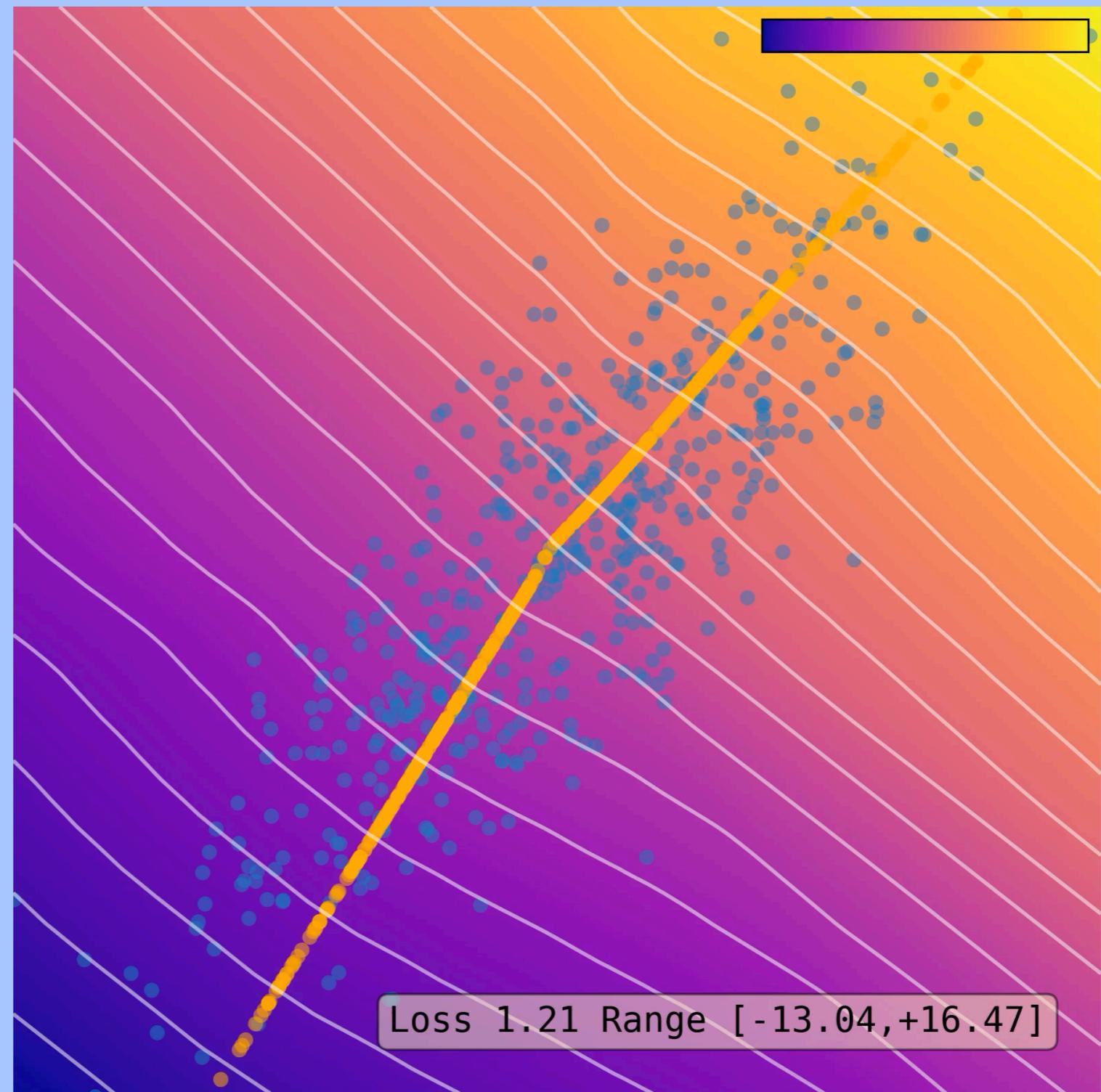
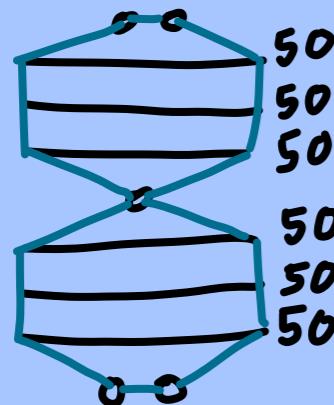
$$\sum_{\ell} \lambda_{\ell} \sum_n (M_{n\ell} - S_{n\ell})^2 \stackrel{!}{=} \text{MIN}$$

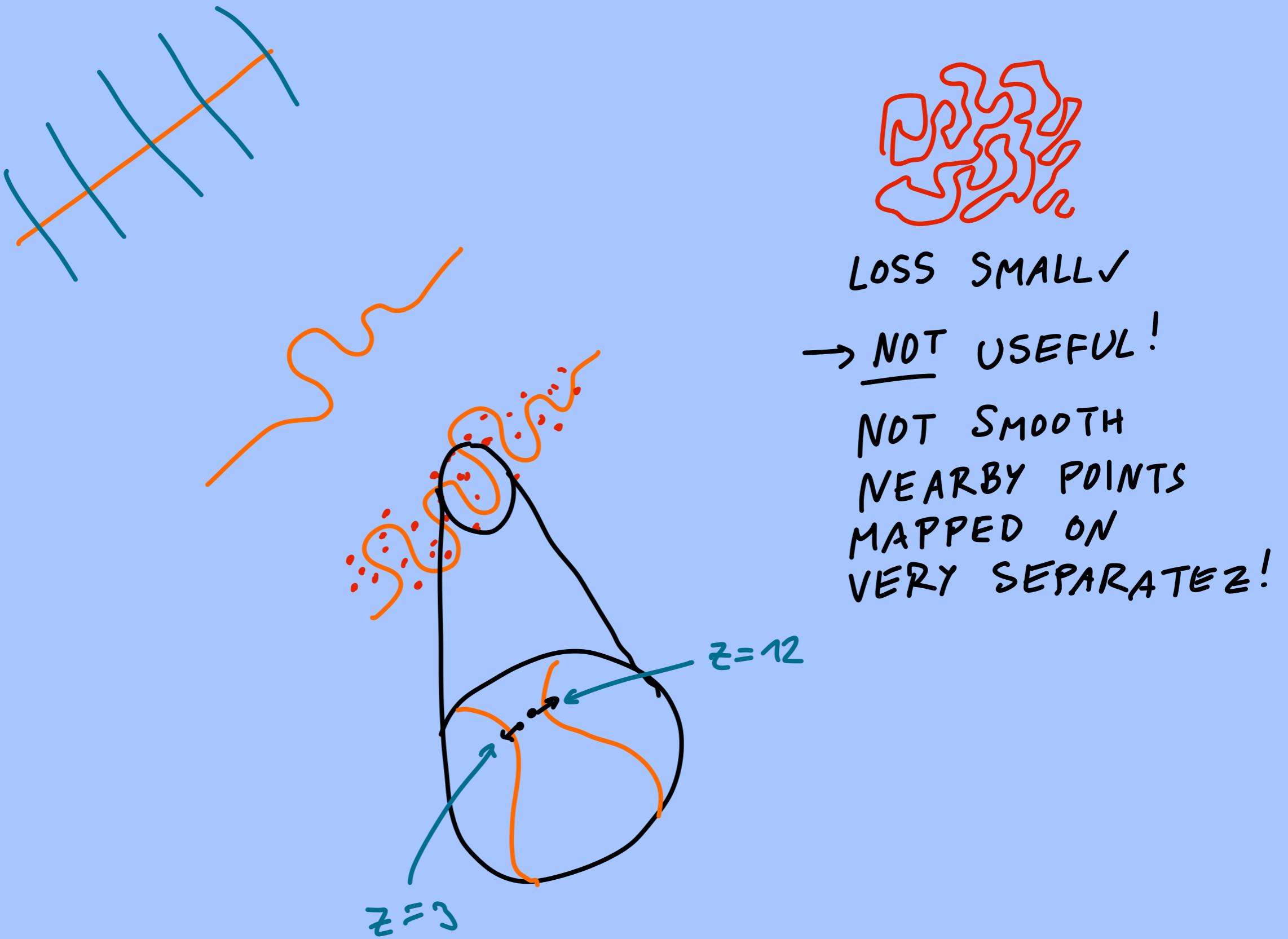
$$M_{n\ell} = 0 \text{ FOR } n \neq \ell$$

IDEALLY  $M_{\ell\ell} = 1$   
 BUT RANK  $\leq k \Rightarrow$  ONLY LARGEST  $\lambda_{\ell}$   
 CAN HAVE THIS

3.3.3

# A PROBLEM

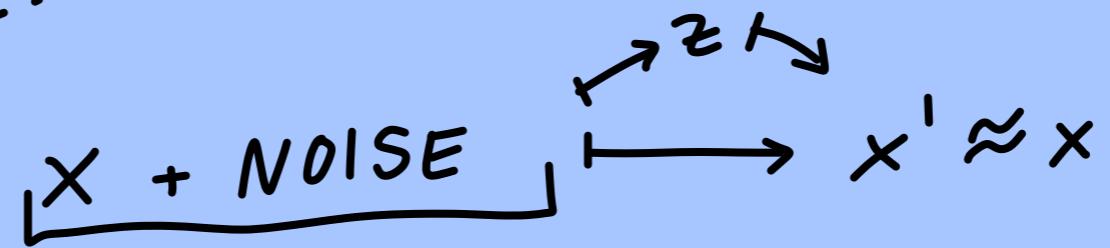




### 3.3.4

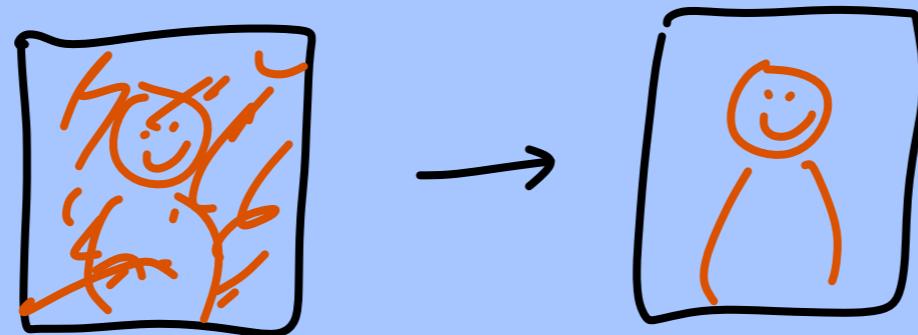
## DENOISING AUTOENCODER

GOAL:



REMOVE NOISE!

$\Rightarrow$  FOCUS ON ESSENTIAL FEATURES



(GAUSSIAN NOISE  
OR SWITCH OFF  
RANDOM PIXELS)

ANALYZE: VERY WEAK GAUSSIAN  
NOISE

$\xi \in \mathbb{R}^d$  NORMAL GAUSSIAN

$$\langle \xi \rangle_{\xi} = 0$$

$$\langle \xi_i \xi_j \rangle_{\xi} = S_{ij}$$

$$\mathcal{L}(\theta) = \left\langle \left\| F_{\theta}(\underline{x} + \varepsilon \underline{\xi}) - \underline{x} \right\|^2 \right\rangle_{x, \xi} \stackrel{!}{=} \text{MIN}$$

$$F_j(x + \varepsilon \xi) \approx F_j(x) + \varepsilon \sum_{\ell} \underbrace{\frac{\partial F_j(x)}{\partial x_\ell}}_{\text{JACOBIAN MATRIX OF } F(x)} \xi_\ell + \dots$$

"JACOBIAN  
MATRIX OF  $F(x)$ "

$$\begin{aligned} & \left\langle \|F(x + \varepsilon \xi) - x\|^2 \right\rangle_{x, \xi} \\ &= \left\langle \|F(x) - x\|^2 \right\rangle_x + \varepsilon^2 \left\langle \sum_j \left( \sum_{\ell} \xi_\ell \frac{\partial F_j}{\partial x_\ell} \right)^2 \right\rangle_{x, \xi} \\ & \quad \uparrow \\ & \langle \xi \rangle = 0 \qquad \qquad \left\langle \xi_\ell \xi_{\ell'} \right\rangle = \delta_{\ell \ell'} \\ &= \dots + \varepsilon^2 \left\langle \sum_j \sum_{\ell} \left( \frac{\partial F_j}{\partial x_\ell} \right)^2 \right\rangle_x \\ & \qquad \qquad \qquad \underbrace{\left\langle \sum_j \left( \frac{\partial F_j}{\partial x_\ell} \right)^2 \right\rangle_x}_{\|J\|_F^2} \end{aligned}$$

$$\|\mathcal{J}\|_F^2 = \sum_{l,h} |\mathcal{J}_{lh}|^2$$

$\Rightarrow$  THIS LOSS PUNISHES  
LARGE DERIVATIVES  
IN  $F!$

## 3.3.5

CONTRACTIVE  
AUTOENCODER

(RIFAI ET AL. 2011)

IDEA: PUNISH GRADIENTS  
OF BOTTLENECK  $z$   
VS. INPUT  $x$ !

$$\left\| \frac{\partial z}{\partial x} \right\| \rightarrow \text{SMALL}$$

COMPETITION BETWEEN TWO GOALS:

$$\left\| \frac{\partial z}{\partial x} \right\| \approx 0$$

$$\downarrow \\ z = \text{const.}?$$

$$\text{BUT THEN} \\ x' = \text{const.}?$$

$$x' \approx x$$

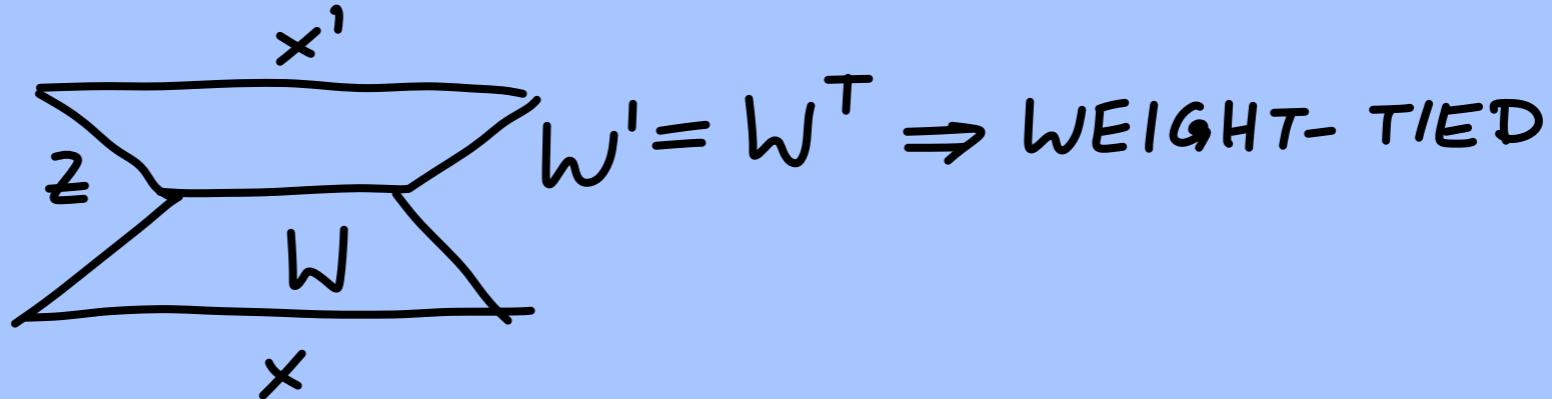
(SOLVED BY  $x' = x$   
e.g.  $z = x$   
THEN  $\frac{\partial z}{\partial x} = 1$ )

$$\mathcal{L}_{CAE}(\theta) = \underbrace{\left\langle \|F_{\theta}(x) - x\|^2 \right\rangle_x}_{MSE} + \lambda \underbrace{\left\langle \|\mathbb{J}_E\|_F^2 \right\rangle_x}_{\text{WEIGHTING FACTOR}} \rightarrow \text{JACOBIAN OF ENCODER}$$

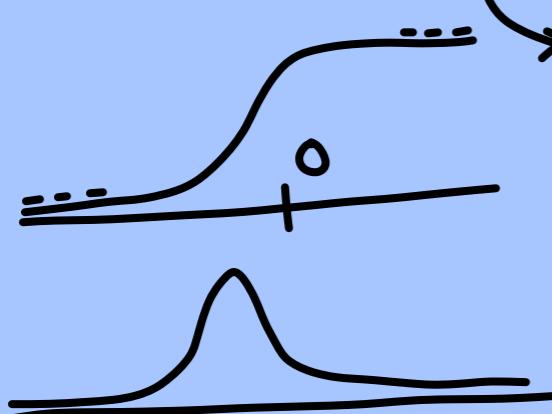
$$(\mathbb{J}_E)_{\ell j} = \frac{\partial z_{\ell}}{\partial x_j} = \frac{\partial E_{\ell}(x)}{\partial x_j}$$

$$\|\mathbb{J}_E\|_F^2 = \sum_{\ell, j} \left| \frac{\partial z_{\ell}}{\partial x_j} \right|^2$$

# ORIGINAL CAE EXAMPLE:



$$\frac{\partial z_e}{\partial x_g} = Z'([Wx + b]_e) \cdot W_{ej}$$



$$Z'(y) = Z(y)(1 - Z(y))$$

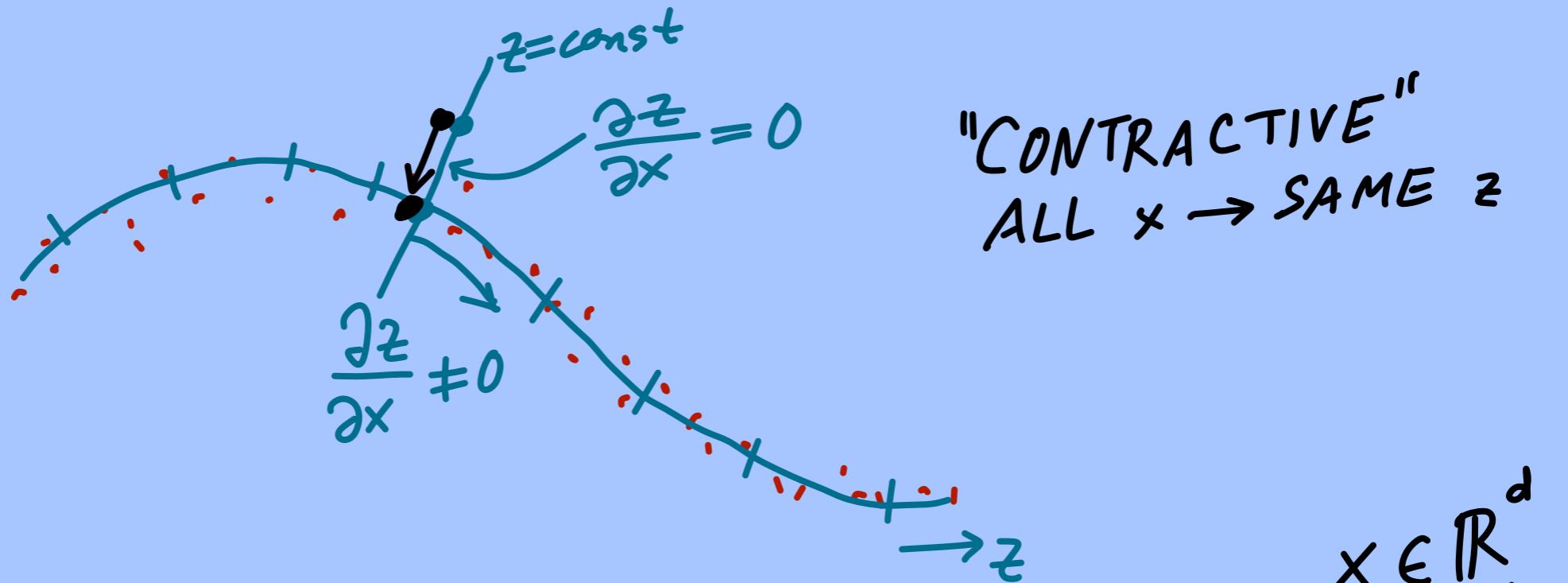
$\Rightarrow$  BIGGEST CONTRIBUTIONS  
FROM NON-SATURATED NEURONS!

$\Rightarrow$  MAKE MOST NEURONS SATURATE!

0 0 0.5 1 0 1

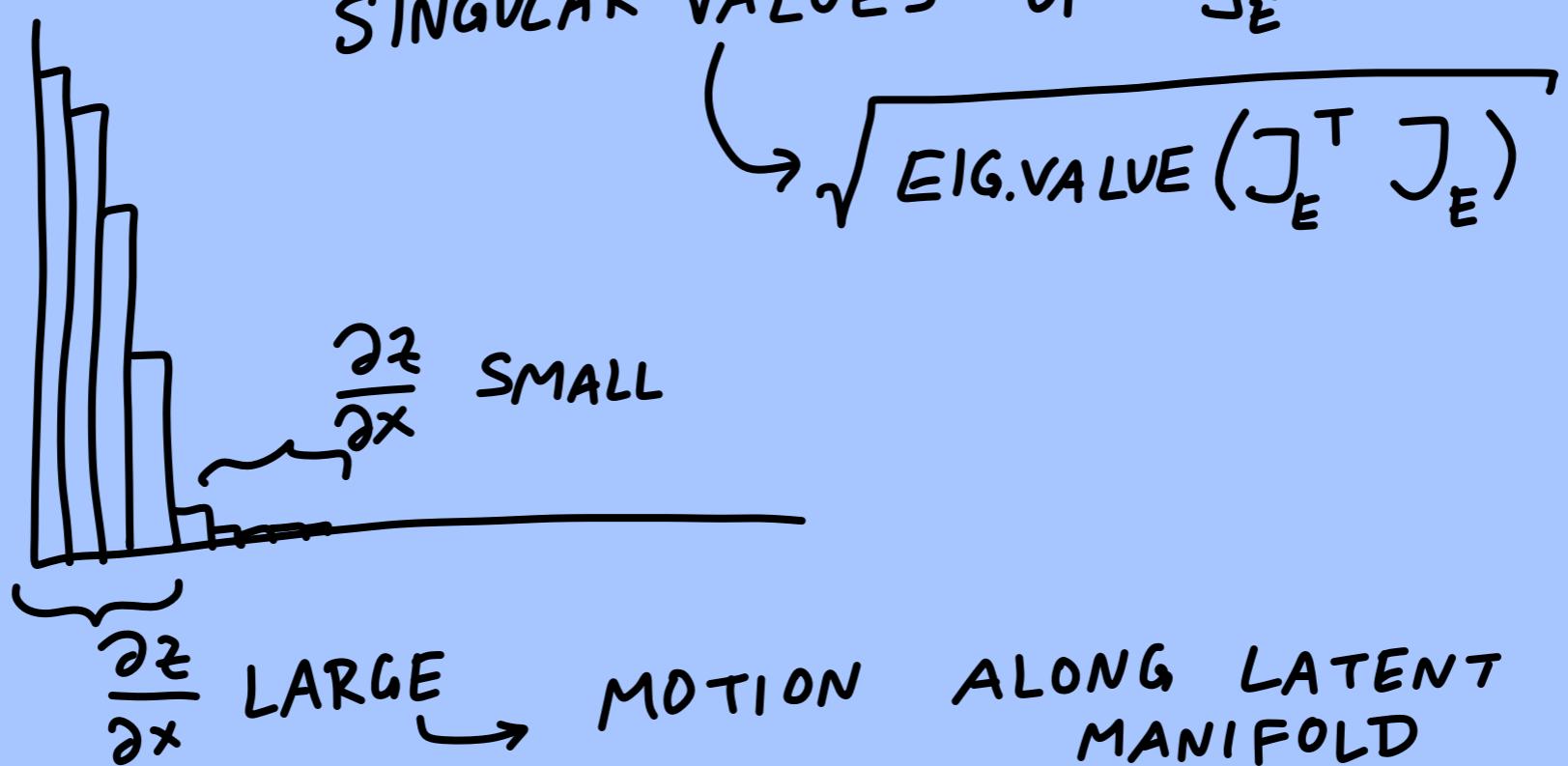
PRODUCES

"SPARSE"  
REPRESENTATION



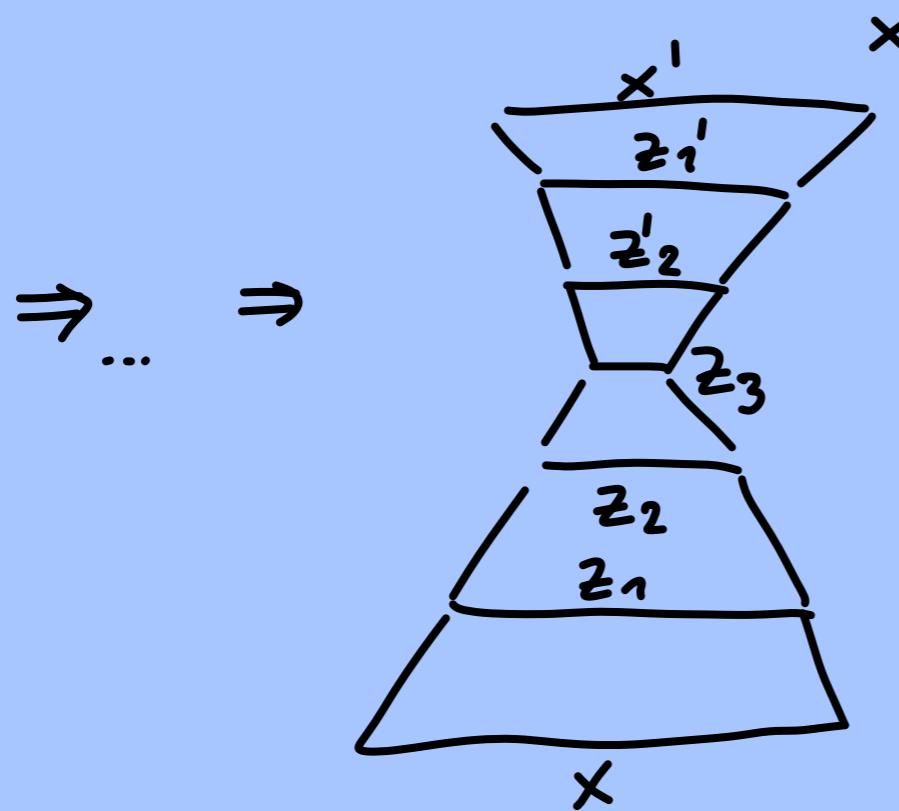
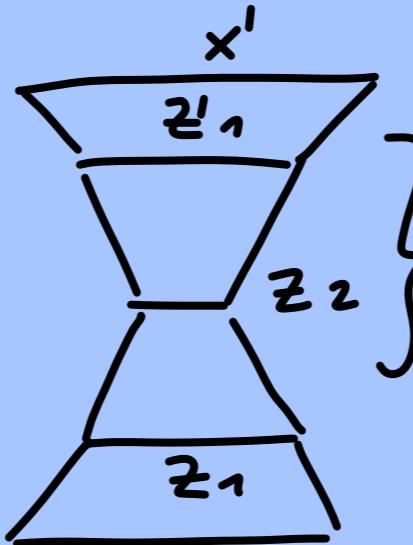
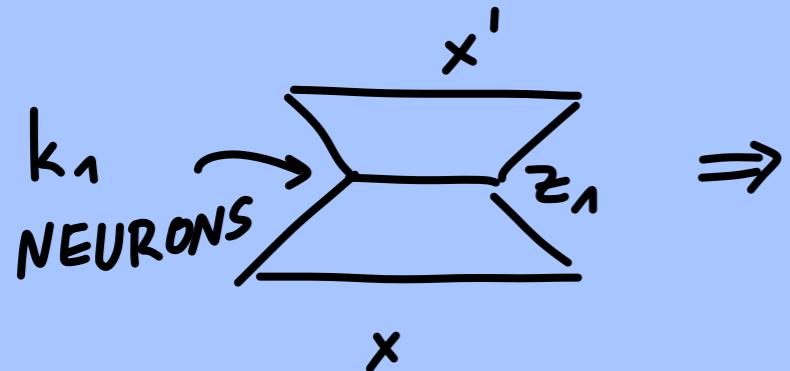
"HOW DOES  $J_E$  LOOK LIKE?"

NUMERICAL EXPERIMENTS (RIFAI ET AL)  
SINGULAR VALUES OF  $J_E$



MORE LAYERS:

"STACKING"



DEEP AUTOENCODERS  
VIA STACKING

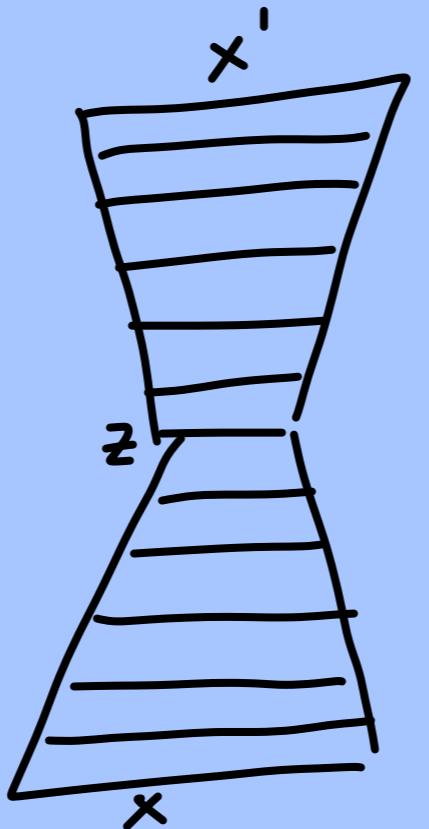
Q : WHY DOES THE CAE  
NOT 'CHEAT' BY SCALING DOWN  $z$ ?

$$\begin{aligned} w \text{ SMALL} &\Rightarrow z \text{ SMALL} \\ &\Rightarrow \frac{\partial z}{\partial x} \sim w \text{ SMALL} \end{aligned}$$

A : WEIGHT-TYING  $w' = w^T$  PREVENTS THIS!

$w \text{ SMALL} \Rightarrow$  WOULD NEED  $w'$  LARGE $\downarrow$

DIRECT DEEP TRAINING OF CAE<sup>2</sup>



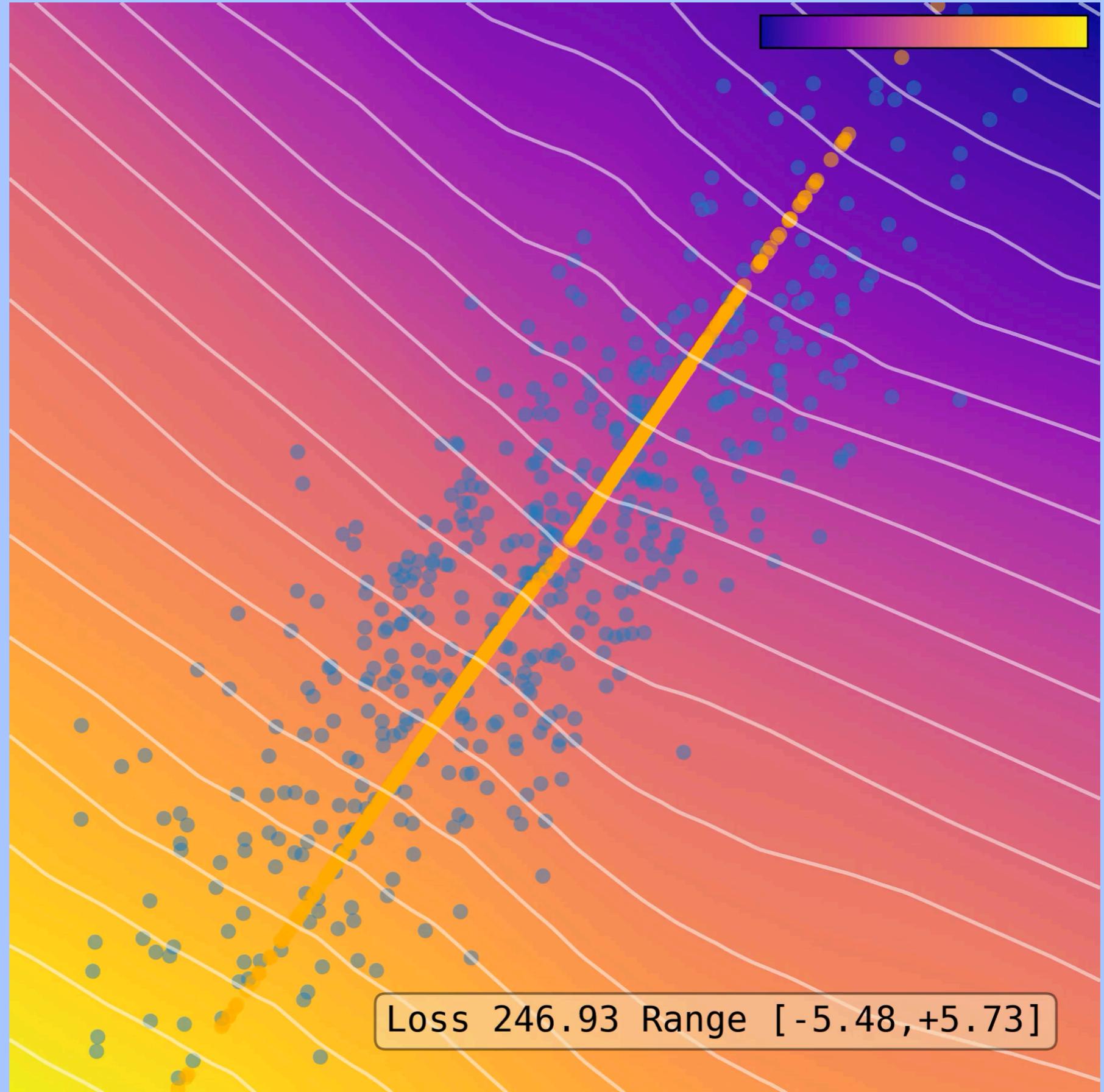
$$\|\mathcal{J}_E\|_F^2$$

⇒ HOW TO PREVENT CHEATING?

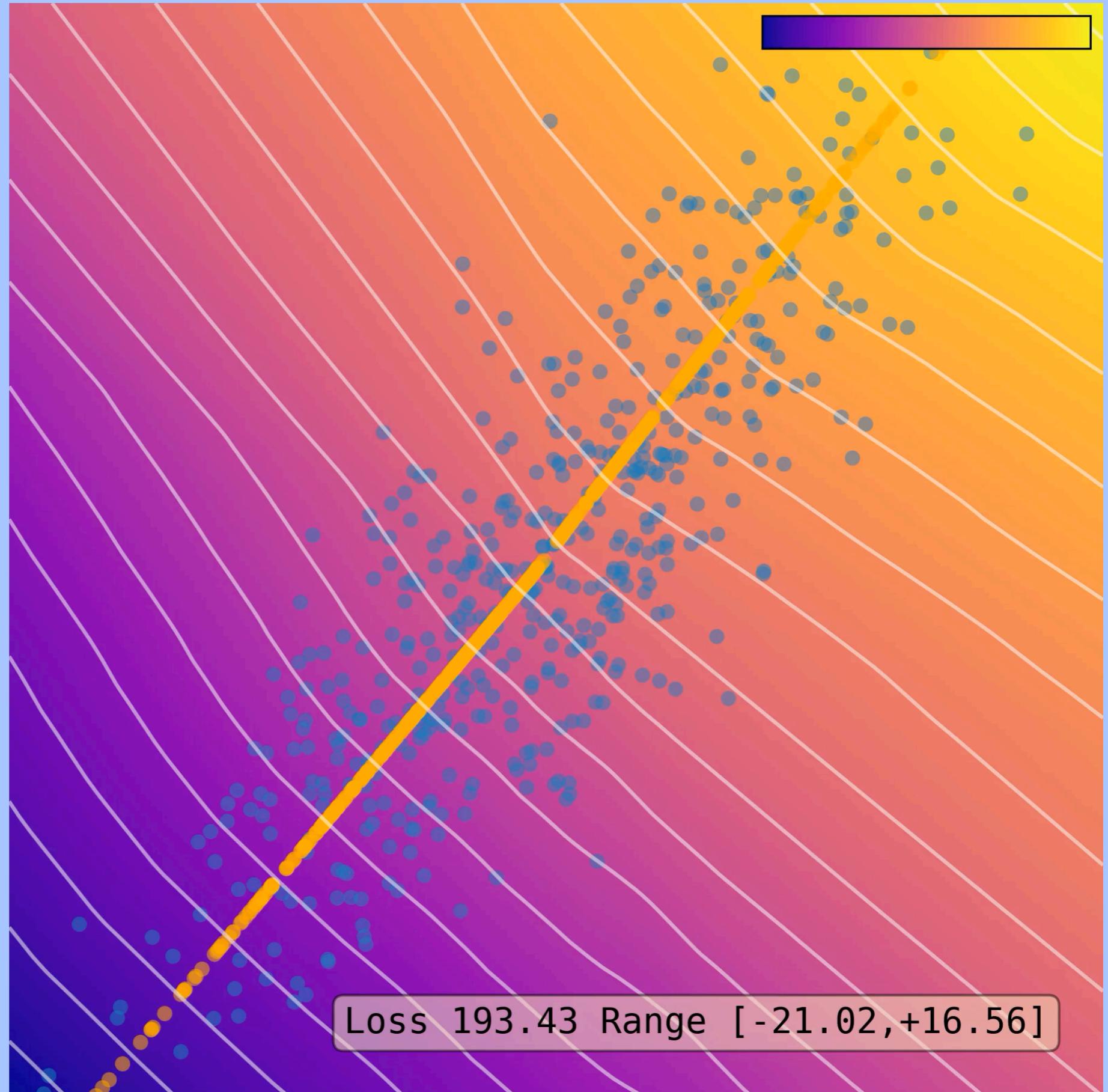
SUGGESTION:

$$\begin{aligned} \mathcal{L} = & \langle \|x - x'\|^2 \rangle_x + \langle \|\mathcal{J}_E\|_F^2 \rangle \lambda_E \\ & + \langle \|\mathcal{J}_D\|_F^2 \rangle \lambda_D \end{aligned}$$

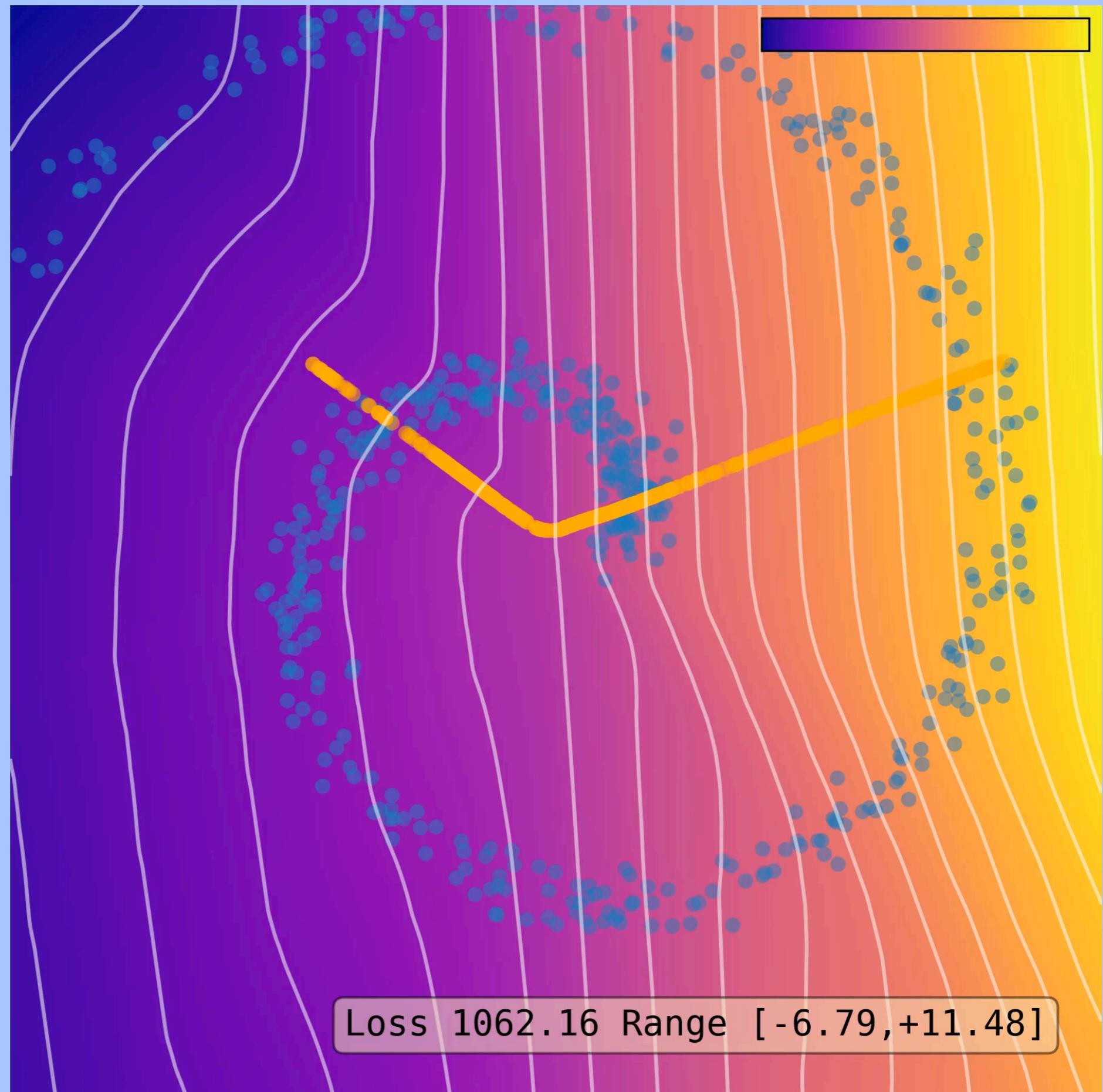
DEEP  
CAE:  
ONLY  
 $\|\mathcal{J}_E\|_F^2$



"JANUS"  
DEEP  
CAE:  
 $\|\mathbb{J}_{E/F}\|^2 + \|\mathbb{J}_{O/F}\|^2$   
(20x SMALLER  
WEIGHT  
FOR JACOBIAN  
THAN BEFORE!)



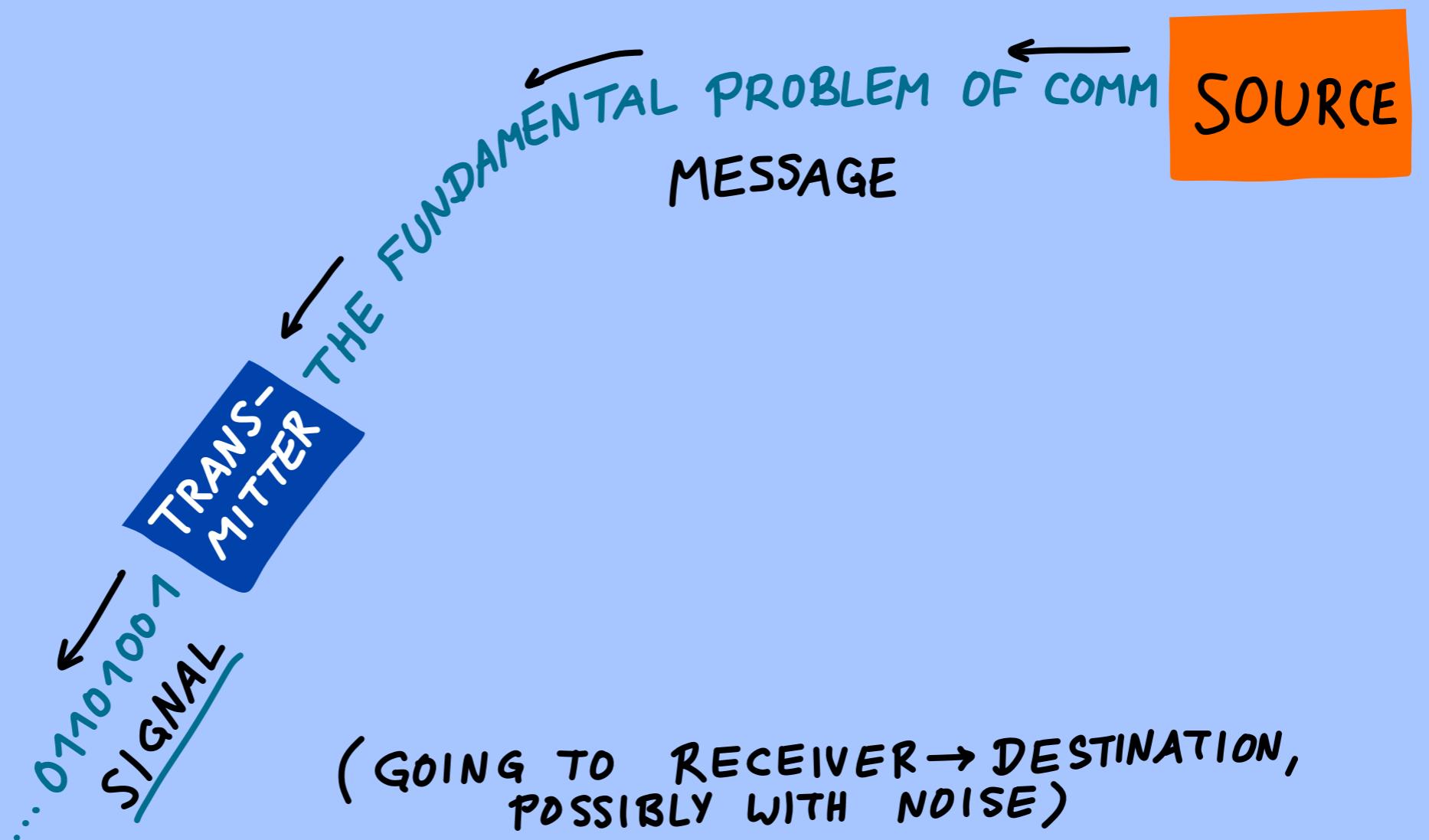
"JANUS"  
DEEP  
CAE:  
 $\|\mathcal{J}_E\|_F^2 + \|\mathcal{J}_{\partial}\|_F^2$



4.

# COMPRESSION, INFORMATION, AND ENTROPY

SHANNON 1948



ALWAYS: CCCCCC....  
→ NO INFORMATION

3141592...  
→ NO INFORMATION

BLUE BLUE BLUE....  
RED RED RED.... } 1 BIT OF INFORMATION  
("WHICH OF THE 2 MSG ?")

101011010010110....  
↑↑↑  
RANDOM, INDEPENDENT  
CANNOT COMPRESS!  
⇒ MAXIMUM INFORMATION!

⇒ STATISTICS OF SOURCE IMPORTANT!

(FROM SHANNON, 1948)

Zero-order approximation

XFOML RXKHRJFFJUJ ALPWXFWJXYJ FFJEYVJCQSGHYD  
QPAAMKBZAACIBZLKJQD

First-order approximation

OCRO HLO RGWR NMIELWIS EU LL NBNSEBYA TH EEI ALHENHTTPA  
OOBTTVA NAH BRL

Second-order approximation

ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE  
TUCOOWE AT TEASONARE FUSO TIZIN ANDY TOBE SEACE CTISBE

Third-order approximation

IN NO IST LAT WHEY CRATICT FROURE BIRS GROCID PONDENOME OF  
DEMONSTURES OF THE REPTAGIN IS REGOACTIONA OF CRE

First-order word approximation

REPRESENTING AND SPEEDILY IS AN GOOD APT OR COME CAN DIFFERENT  
NATURAL HERE HE THE A IN CAME THE TO OF TO EXPERT GRAY COME TO  
FURNISHES THE LINE MESSAGE HAD BE THESE

Second-order word approximation

THE HEAD AND IN FRONTAL ATTACK ON AN ENGLISH WRITER THAT THE  
CHARACTER OF THIS POINT IS THEREFORE ANOTHER METHOD FOR THE  
LETTERS THAT THE TIME OF WHO EVER TOLD THE PROBLEM FOR AN  
UNEXPECTED

CAN WE COMPRESS

11101111111011011111111

RANDOM, INDEPENDENT

80% '1'  
20% '0'

IDEA: CONSIDER LARGER SEGMENTS  
AS 'SYMBOLS'

NOT 1 vs 0

BUT  $\begin{matrix} 111011 \\ \text{vs} \\ 101011 \end{matrix}$  } DIFFERENT PROBS

## 4.1

## COMPRESSION

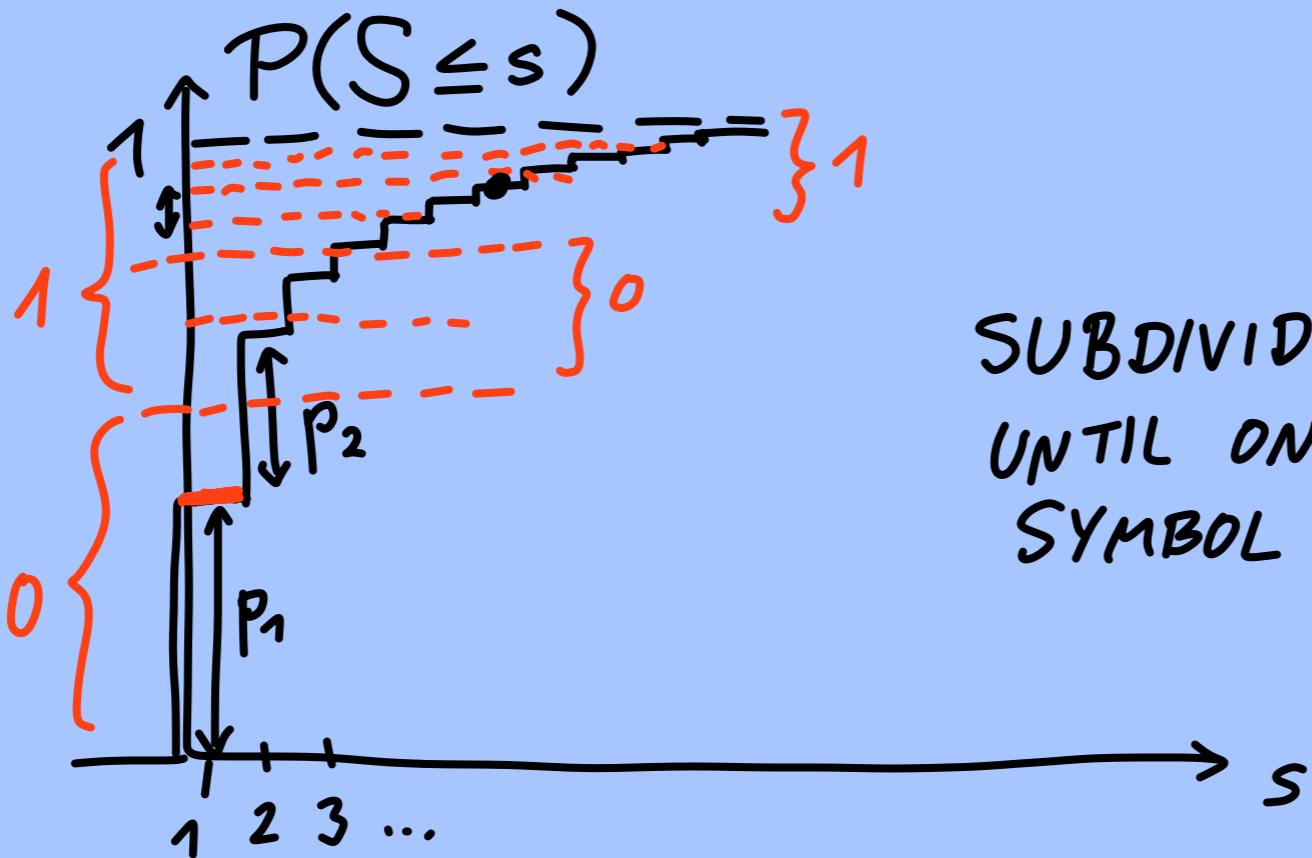
→ CONSIDER SYMBOLS  $s$

→ SOURCE PICKS THEM RANDOMLY & INDEPENDENTLY  
TO FORM SEQUENCES

FOR  $P_s \sim \frac{1}{2^n}$  ⇒ EXPECT CODE LENGTH  
OF  $n$  BITS

⇒ THERE CAN  
BE  $2^n$  SYMBOLS  $n = -\log_2 P_s$

DIFFERENT  $P_s$  ⇒ SHORT CODE  
WORDS FOR  
LARGE  $P_s$

$P_1, P_2, P_3, \dots$  $P_1 \geq P_2 \geq P_3 \dots$ 

RESOLUTION NEEDED  
FOR  $P_s$

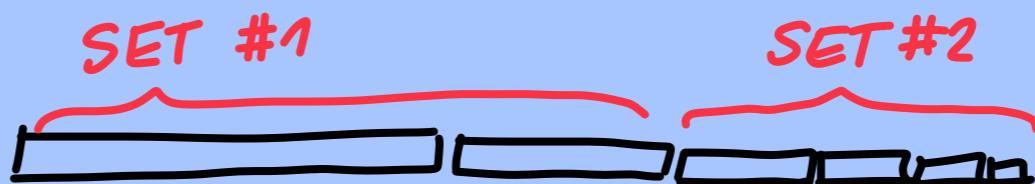
$$\frac{1}{2^n} \lesssim P_s$$

$$\Rightarrow n \sim -\log_2 P_s$$

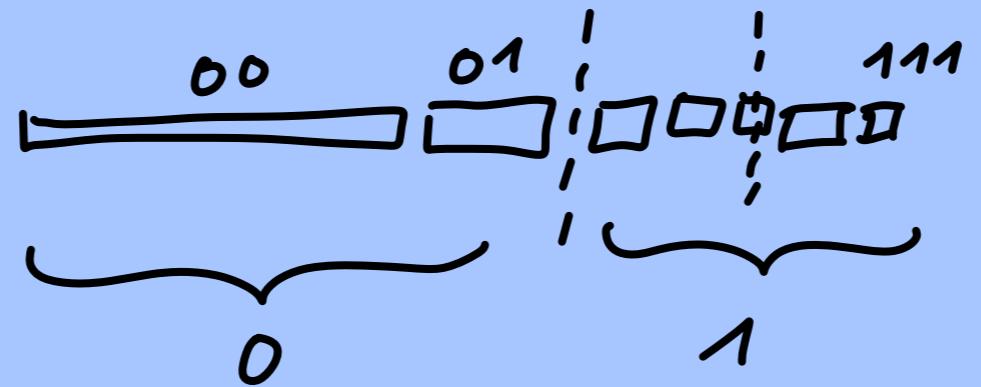
## EXPLICIT CONSTRUCTION (FANO):

- ORDER  $p_s$  BY SIZE
- DIVIDE INTO 2 SETS WITH  
(ALMOST) EQUAL PROBABILITY

$$\left( \sum_{\text{SET } \#1} p_s \approx \sum_{\text{SET } \#2} p_s \right)$$



- ASSIGN '0' TO SET #1  
    '1' TO #2
- SPLIT AGAIN ETC.,  
UNTIL ONLY ONE  
SYMBOL LEFT IN SET



"PREFIX CODE": NO NEED FOR  
STOP SYMBOL  
(NO CODE WORD  
IS BEGINNING OF  
LONGER CODE WORD)

NUMBER OF SPLITTINGS NEEDED?  
= CODE LENGTH  $\sim -\log_2 P_s$

ALTERNATIVE: SHANNON METHOD

CUMULATIVE PROBABILITIES

$$C_1 = 0 \quad C_2 = P_1 \quad C_3 = P_1 + P_2 \quad \dots$$

CODE FOR S:

USE CODE WORD

OF LENGTH  $L_s = \lceil -\log_2 P_s \rceil$

ROUND  
UP

NAMELY: BINARY

EXPANSION OF  $C_s$

(e.g.  $0.\underline{1011010} \dots$ )

KEEP  $L_s$  BINARY  
DIGITS

4.2

## SHANNON'S CODING THEOREM &amp; ENTROPY

AVERAGE CODE LENGTH?

$$\langle L \rangle = \sum_s p_s \underbrace{L_s}$$

$$= \lceil -\log_2 p_s \rceil \quad \text{FOR SHANNON COMPRESSION}$$

SHANNON'S CODING THEOREM:

$$\langle L \rangle \geq H$$

$$H = \text{ENTROPY} = - \sum_s p_s \log_2 p_s$$

OF SOURCE

$$-\log_2 p_s = \text{"INFORMATION"} \rightarrow \begin{array}{l} \text{LARGE} \\ \text{FOR } \underline{\text{RARE}} \\ \text{EVENTS} \end{array}$$

$$-\log_2 p_s = \text{"SURPRISE"}$$

FANO &amp; SHANNON COME WITHIN ONE BIT OF OPTIMAL

## → HUFFMAN CODING

ENGLISH LANGUAGE:

MAX. POSSIBLE ENTROPY  
FOR 26 LETTERS:  $-\log_2 \frac{1}{26} \approx \frac{4.7 \text{ BITS}}{\text{LETTER}}$

ACTUAL ENTROPY:  $\frac{2.6 \text{ BITS}}{\text{LETTER}}$

~ 50% REDUNDANCY

## EXAMPLE

1110111101110111111

80% '1'

20% '0'

PROB.

00000       $0.2^5$

:

00010       $0.2^4 \cdot 0.8$

:

11111       $0.8^5$

$$-\log_2(0.8^5) = 1.6$$

CODE:

11111  $\mapsto$  00

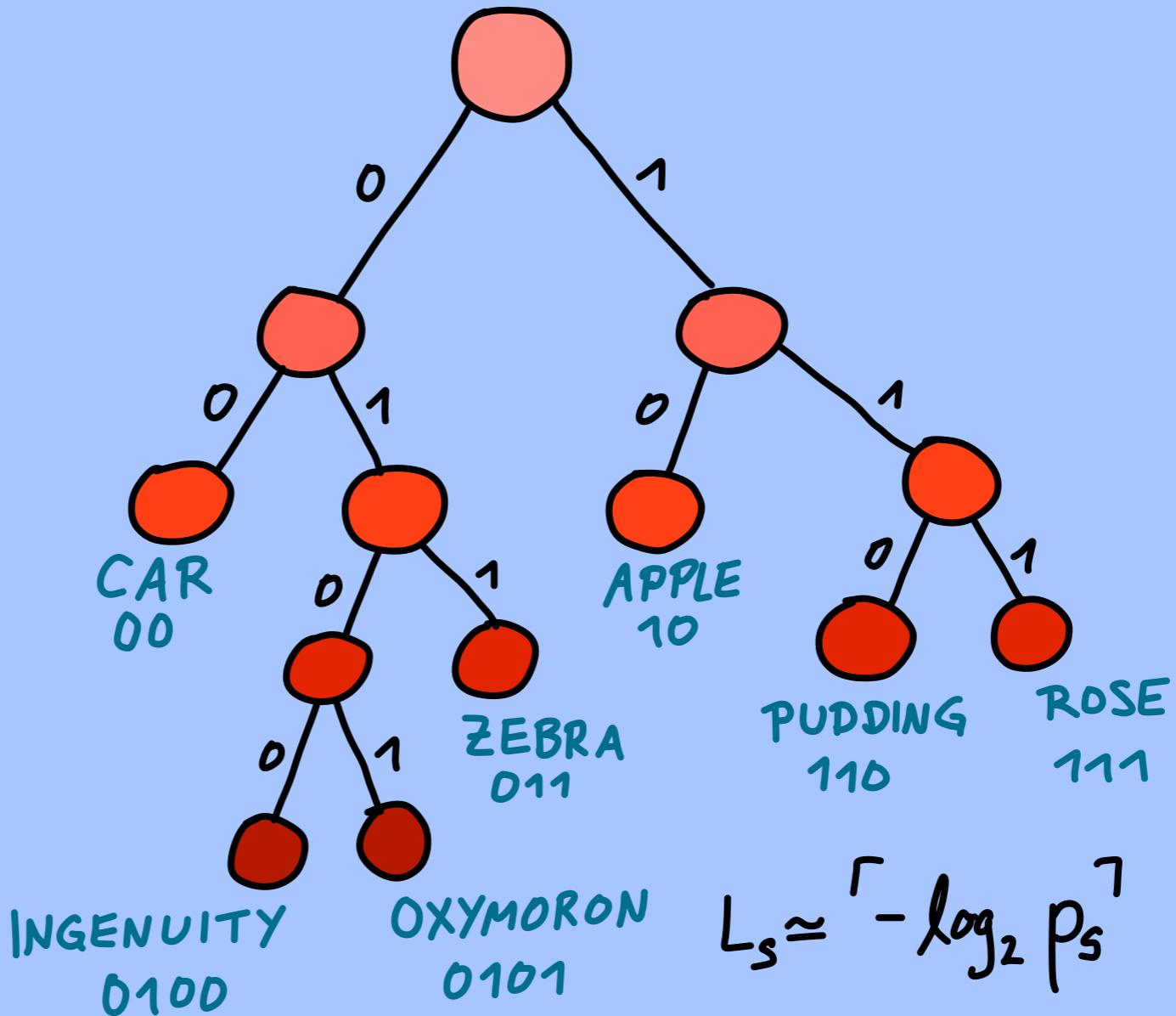
:

00000  $\mapsto$  1111111111...

12 BITS

$\Rightarrow$  H TELLS US HOW  
MUCH WE CAN COMPRESS.

## EXAMPLE: ENCODING WORDS



"DECISION TREE"

$$H = - \sum_s p_s \log_2 p_s$$

$$\langle L \rangle = \sum_s p_s L_s \geq H$$

4.3

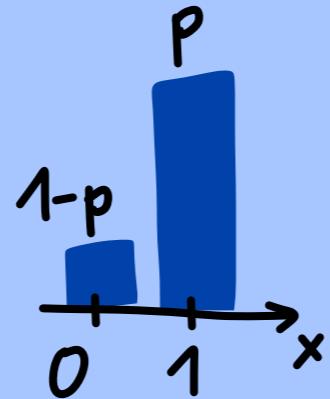
## BASIC PROPERTIES OF THE ENTROPY

$$H(X) = - \sum_j P(X=x_j) \log_b P(X=x_j)$$

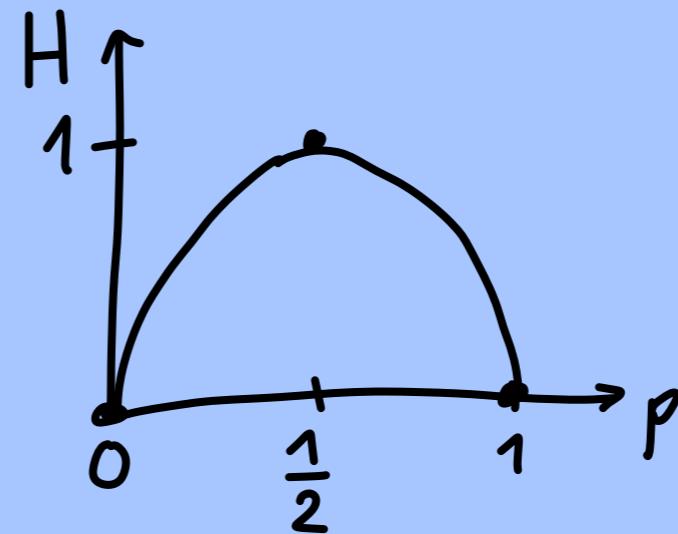
↑  
RANDOM VARIABLE

BASE b  
(b=2, b=e, ...)

EXAMPLE:



$$H = -p \log_2 p - (1-p) \log_2 (1-p)$$



$H \geq 0$  (SINCE  $P \leq 1$ )

$H \leq \log_b N$  ( $N = \text{DIFFERENT VALUES}$ )



$X, Y$  INDEPENDENT  $\Leftrightarrow P(X=x, Y=y)$   
 $= P_x(X=x) \cdot P_y(Y=y)$

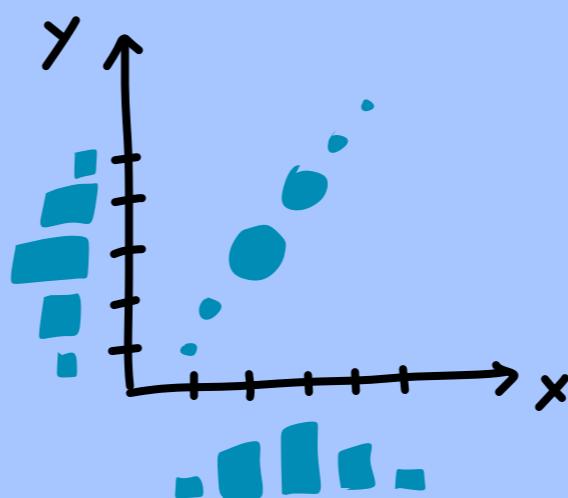
$$H(X, Y) = - \sum_{x, y} P(X=x, Y=y) \log_2 P(X=x, Y=y)$$

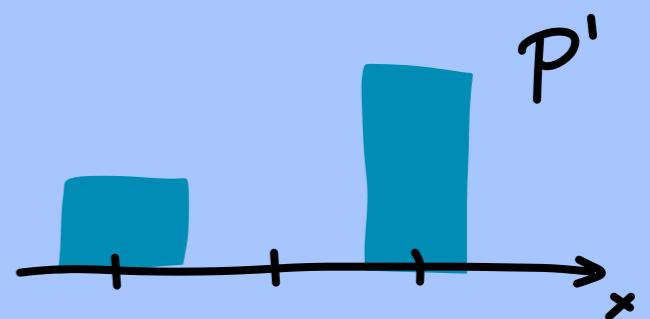
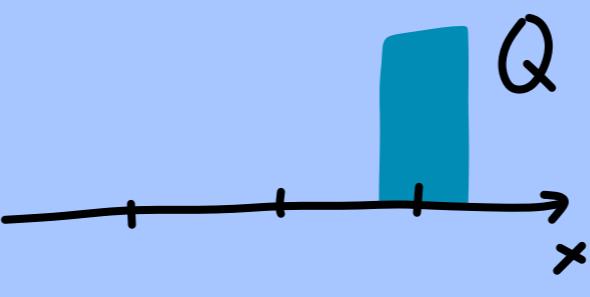
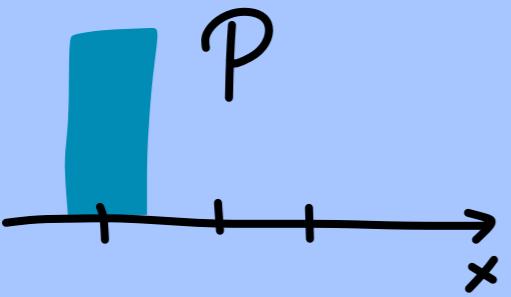
$\stackrel{=} \dots = H(X) + H(Y)$

$$\begin{aligned} & \log P_x \cdot P_y \\ &= \log P_x + \log P_y \end{aligned}$$

IN GENERAL:

$$H(X, Y) \leq H(X) + H(Y)$$





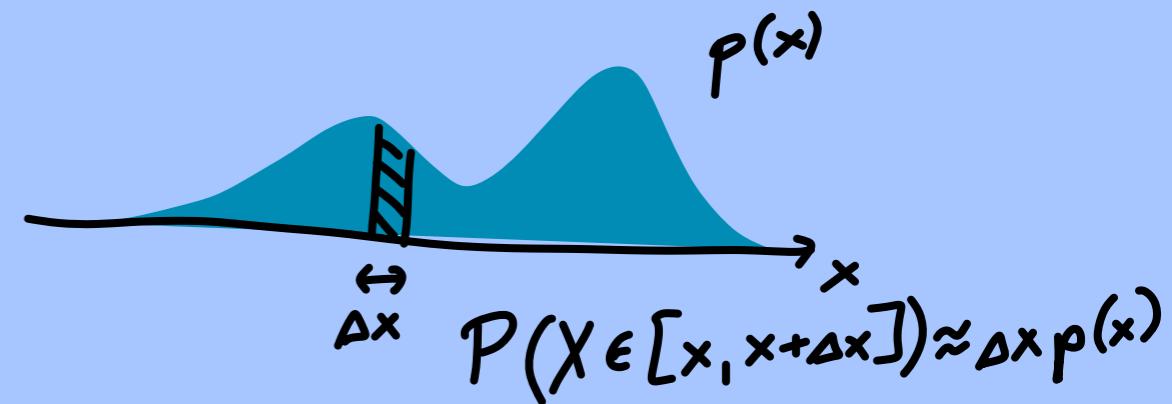
$$P' = \lambda P + (1-\lambda) Q$$

$$H(P') \geq \lambda H(P) + (1-\lambda) H(Q)$$

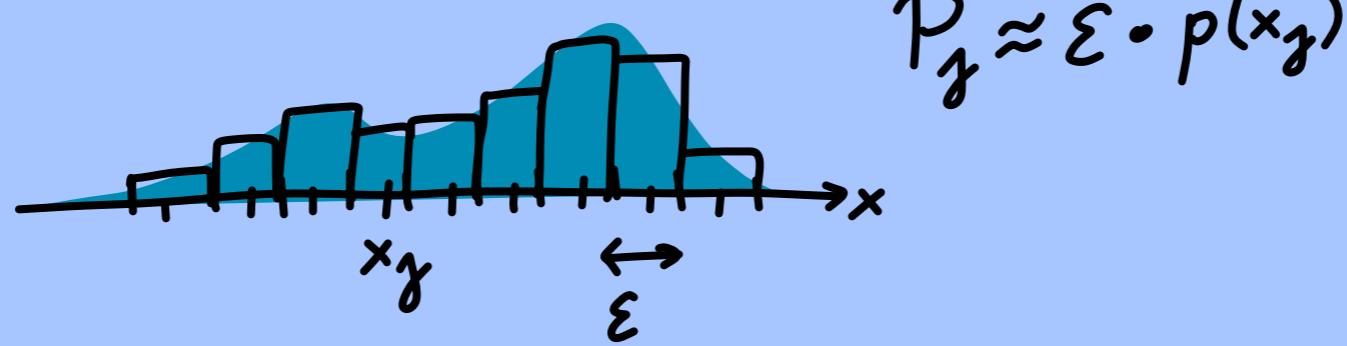
"CONCAVITY"

# CONTINUOUS RANDOM VARIABLES

→ PROBABILITY DENSITY



→ DISCRETIZE WITH RESOLUTION  $\varepsilon$

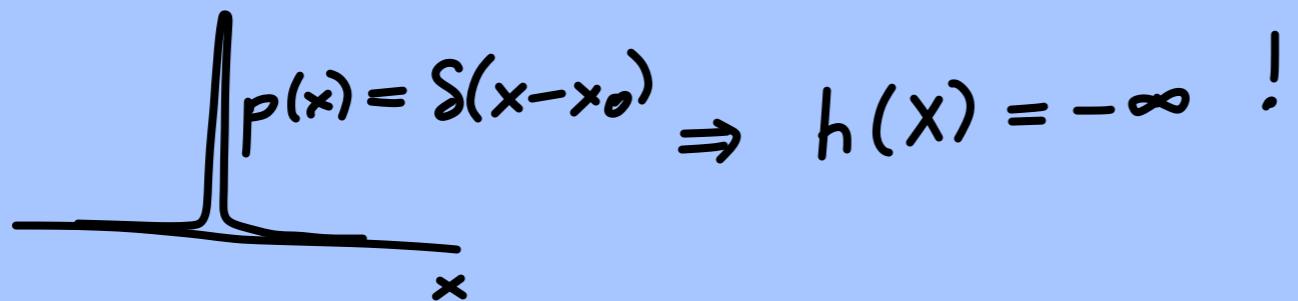


$$\begin{aligned}
 H_{\text{DISCRETE}}(X_{\text{DISCRETE}}) &= - \sum_j P_j \log P_j \\
 &= - \sum_j \varepsilon p(x_j) \log [\varepsilon \cdot p(x_j)] \\
 &\approx - \int dx p(x) \underbrace{\log(\varepsilon p(x))}_{\log \varepsilon + \log p(x)} \\
 &= - \log \varepsilon - \int dx p(x) \log p(x)
 \end{aligned}$$

$$\varepsilon \approx dx$$

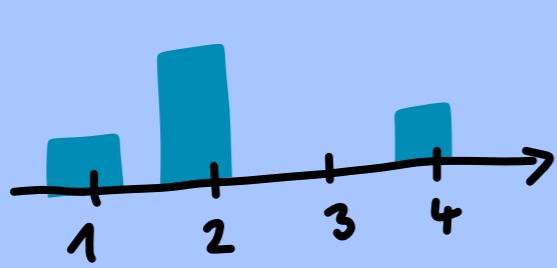
DEFINE DIFFERENTIAL ENTROPY

$$h(X) = - \int p(x) \log p(x) dx$$



NO UPPER BOUND

STILL:  $X, Y$  INDEP.  $\rightarrow h(X, Y) = h(X) + h(Y)$

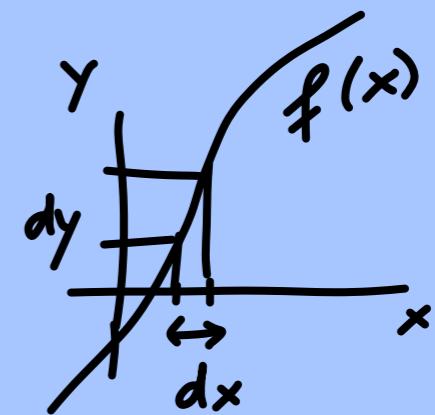


H DOES  
NOT CHANGE  
UNDER PERMUTATION

CONTINUOUS CASE :

BIGEACTIVE  
MAPPING

$$y = f(x)$$



$$|p_y(y)dy| = p_x(x(y))dx$$

$$\frac{dy}{dx}dx = f'(x)dx$$

$$p_y(y) = \frac{p_x(x(y))}{|f'(x(y))|}$$

$$\Rightarrow h(Y) = h(X) + \int \log |f'(x)| p(x) dx$$

HIGHER DIMENSIONS:

$$|f'(x)| \mapsto \left| \det \underbrace{\left( \frac{\partial f^e}{\partial x_j} \right)_{ej}}_{\text{JACOBIAN}} \right|$$

5.

# BAYES

5.1

## MOTIVATION

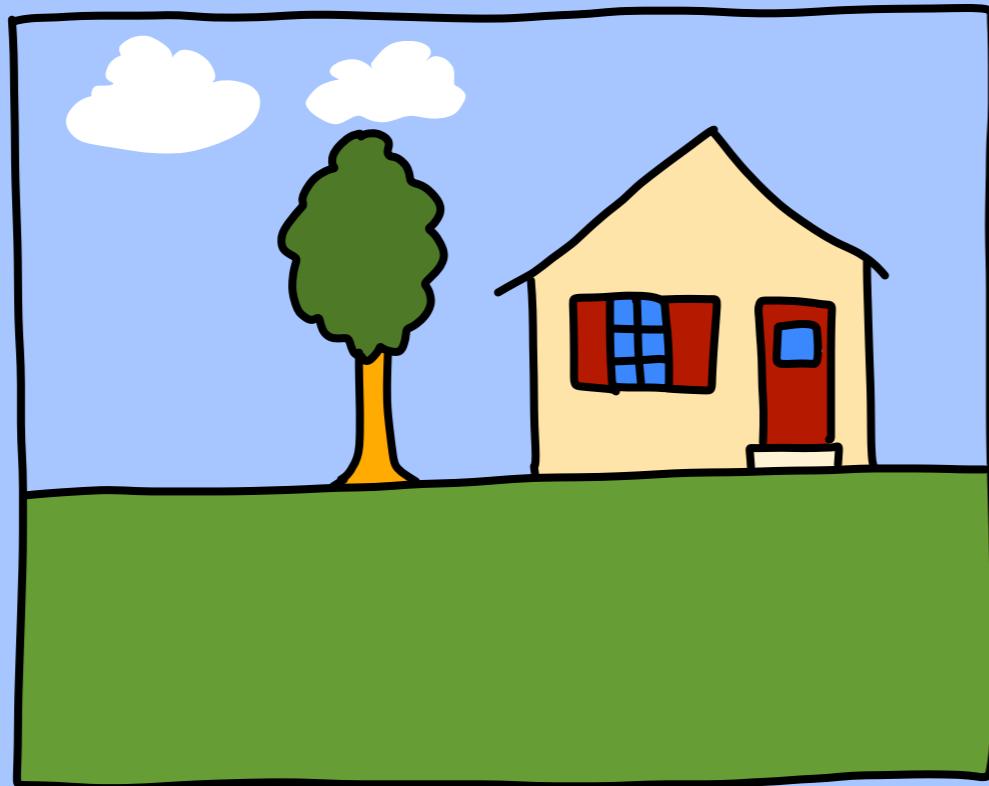
GIVEN AN OBSERVATION:

"WHAT IS THE MOST PLAUSIBLE  
EXPLANATION FOR THIS OBSERVATION?"

"HOW SHOULD WE UPDATE OUR  
KNOWLEDGE ABOUT THE WORLD?"

"HOW MUCH DO WE LEARN ABOUT  
THE WORLD FROM THIS OBSERVATION?"  
→ REDUCE UNCERTAINTY

## OBSERVATION

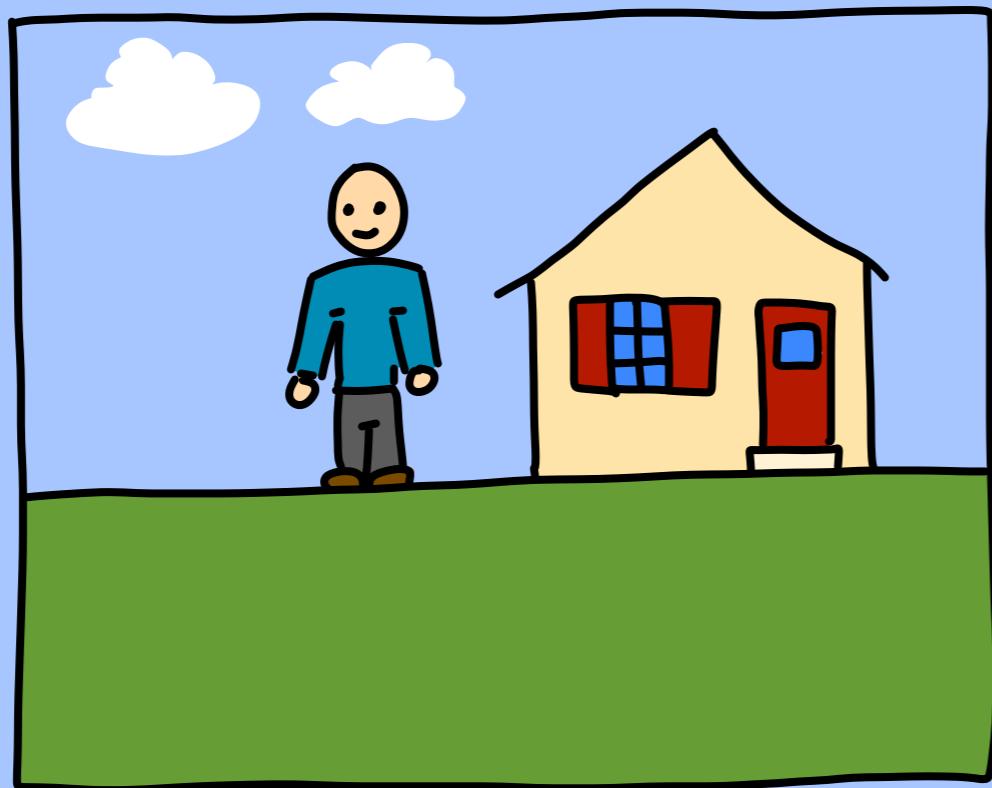


## POSSIBLE EXPLANATION

HOUSE AND TREE  
OF TYPICAL SIZE,  
NEXT TO EACH OTHER



## OBSERVATION



## POSSIBLE EXPLANATION

~~HOUSE AND HUMAN  
OF TYPICAL SIZE,  
NEXT TO EACH OTHER~~

(GIANT?)

TOY HOUSE?

PERSPECTIVE?

## 5.2

## THE BAYES FORMULA

CONDITIONAL PROBABILITIES

$$P(A|B) = \frac{P(A, B)}{P(B)}$$

"BOTH A AND B"

$$P(B|A) = \frac{P(A, B)}{P(A)} = \frac{P(A|B) P(B)}{P(A)}$$

BAYES  
FORMULA

$$P(A) = \sum_{B'} P(A|B') P(B') = \text{NORMALIZATION}$$

$$P(\text{MODEL} | \text{OBSERVATION}) \underset{\text{EXPLANATION}}{=} \frac{P(\text{OBS.} | \text{MODEL}) \cdot P(\text{MODEL})}{P(\text{OBS.})}$$

↳ UNDER ANY MODEL

$\Rightarrow$  "UPDATE KNOWLEDGE"

OFTEN:  $\lambda$  = PARAMETER OF MODEL

$y$  = OBSERVATION

$P(\lambda)$  = PRIOR DISTRIBUTION

$P(\lambda|y)$  = POSTERIOR DISTR.

$P(y|\lambda)$  = LIKELIHOOD

SEQUENCE OF OBSERVATIONS:

$$P(\lambda|\{y\}) = P(\lambda|y_1, \dots, y_N) = \frac{P(\{y\}|\lambda)P(\lambda)}{P(\{y\})}$$

$$P(y_1, \dots, y_N|\lambda) = \prod_{j=1}^N P(y_j|\lambda) \quad (\text{FOR INDEP. OBSERV. !})$$

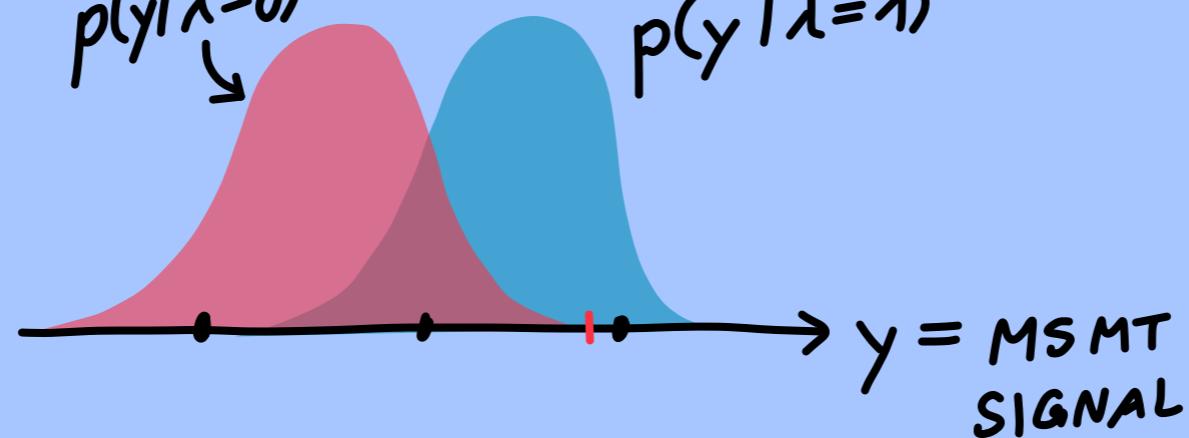
$$\log p(\lambda|\{y\}) = \underbrace{\sum_j \log P(y_j|\lambda)}_{\text{LOG-LIKELIHOOD}} + \log P(\lambda) - \underbrace{\log P(\{y\})}_{\text{INDEP. OF } \lambda}$$

5.3

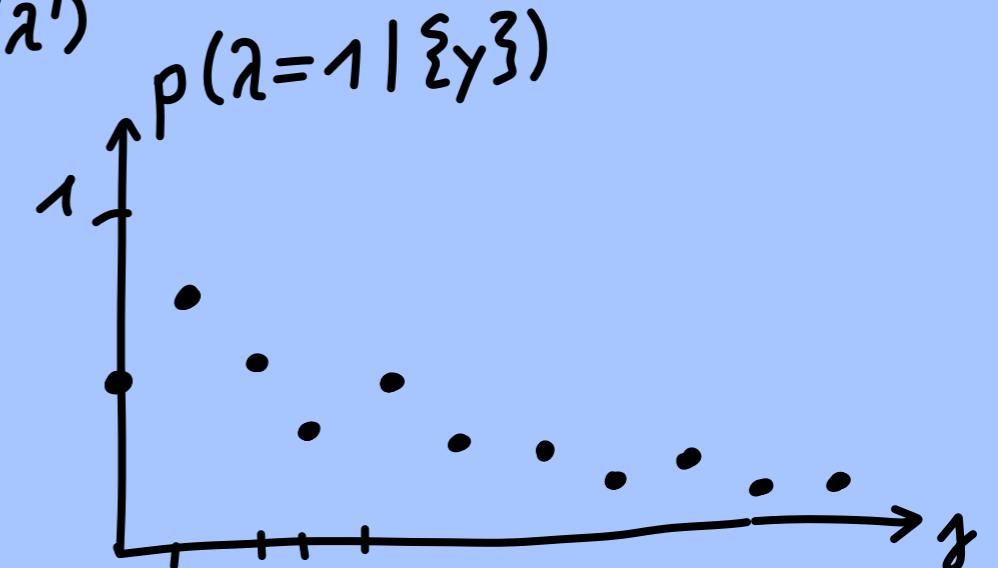
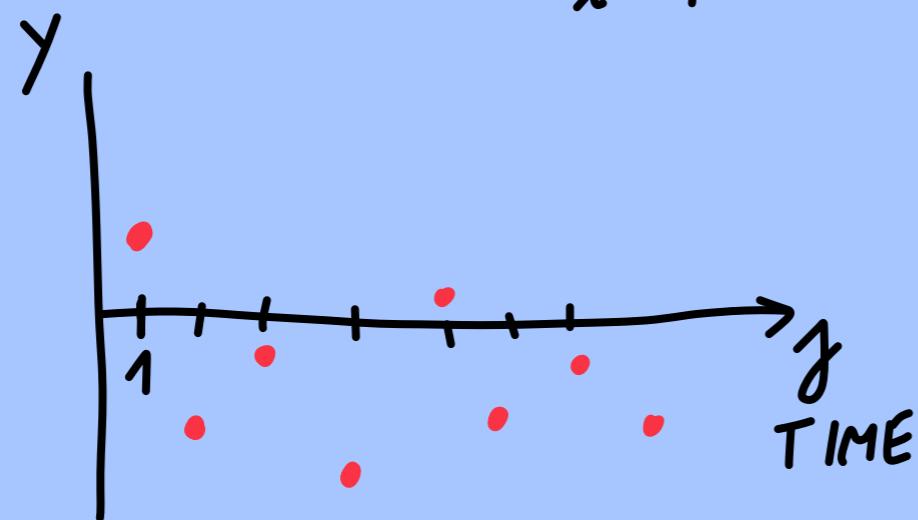
## TWO PHYSICS EXAMPLES

QUBIT MEASUREMENT

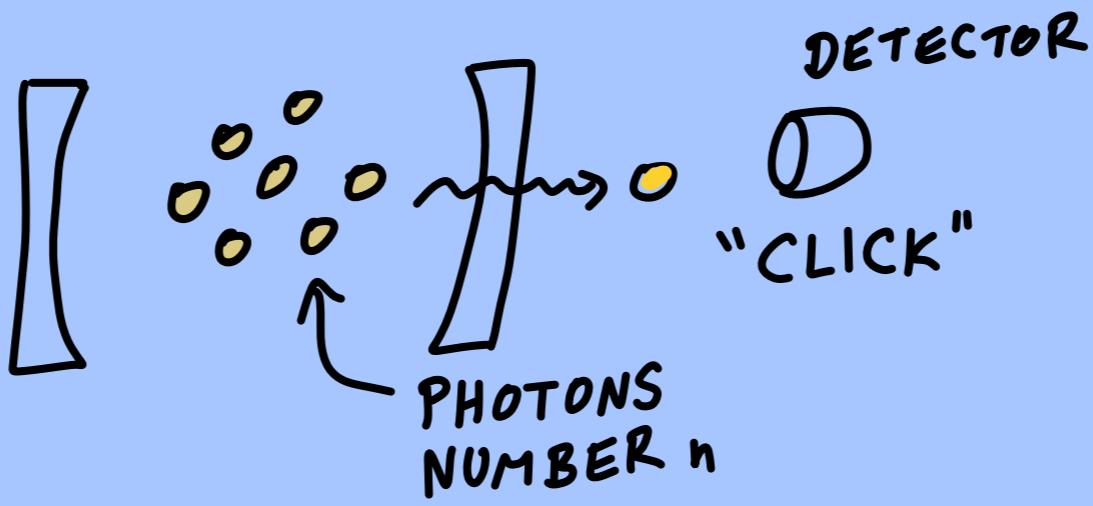
$$p(y|\lambda=0) \quad p(y|\lambda=1)$$



$$p(\lambda=1|y) = \frac{p(y|\lambda=1) \cdot p(\lambda=1)}{\sum_{\lambda'=0,1} p(y|\lambda') p(\lambda')}$$



# CAVITY DECAY



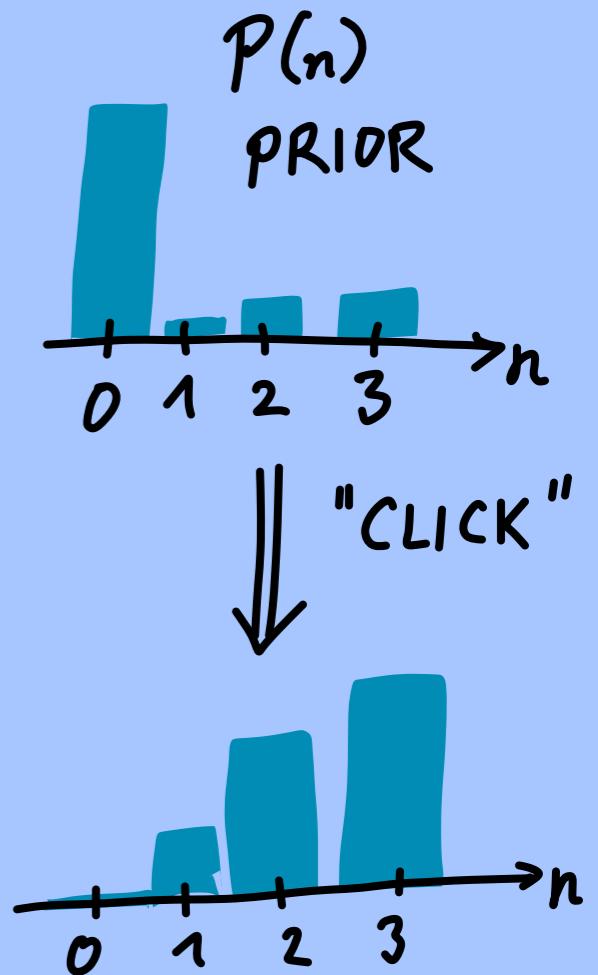
LIKELIHOOD

$$P(\text{CLICK} | n) \approx (\kappa\tau) \cdot n$$

↓  
IN A SHORT TIME INTERVAL  $\tau$

DECAY RATE

$$P(n | \text{CLICK}) = \frac{P(\text{CLICK}|n) P(n)}{P(\text{CLICK})}$$



PHYSICS  $n_{\text{FINAL}} = n - 1$

STILL CAN HAVE  $\langle n_{\text{FINAL}} \rangle_{\text{POSTERIOR}} > \langle n \rangle_{\text{PRIOR}}$

"DECAY INCREASES PHOTON NUMBER"  
 ~ UPDATE OF KNOWLEDGE!

## 5.4

## CONDITIONAL ENTROPY

$$\begin{aligned}
 H(X|Y) &= \left\langle - \sum_x P(X=x|Y=y) \log P(X=x|Y=y) \right\rangle_y \\
 &= - \sum_{x,y} \underbrace{P(x|y)P(y)}_{P(x,y)} \log P(x|y) \\
 &= - \sum_{x,y} P(X=x, Y=y) \log \underbrace{\frac{P(X=x|Y=y)}{P(x,y)}}_{\frac{P(x,y)}{P(y)}}
 \end{aligned}$$

ALWAYS:  $\underbrace{H(X|Y)}_{\text{REDUCED ENTROPY}} \leq H(X)$

BECAUSE OF ADDITIONAL INFO!

$$H(X|Y) = H(X, Y) - H(Y) \leq H(X) \quad \checkmark$$

NOTE:  $H(X|Y) \leq H(X)$  ONLY BECAUSE WE AVERAGE OVER  $Y$

$$H(X|Y=y) = - \sum_x P(X=x|Y=y) \log P(X=x|Y=y)$$

CAN GO UP OR DOWN!

(EXPECTED) "INFORMATION GAIN"

$$H(X) - H(X|Y) \geq 0$$

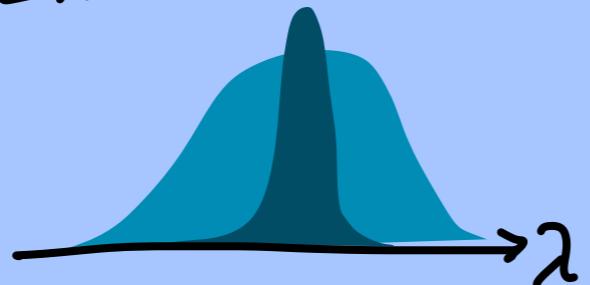
## 5.5

# UNDERSTANDING THE WORLD VIA BAYES

$$\lambda = \text{ALL POSSIBLE MODELS} = \left\{ \begin{array}{l} \text{PTOLEMY WITH PARAMS} \\ \text{KEPLER} \\ \text{NEWTON} \end{array} \right. \quad F \sim \frac{1}{r^\alpha}$$

$P(\lambda)$  PRIOR OVER MODELS ( $\rightarrow ?!$ )

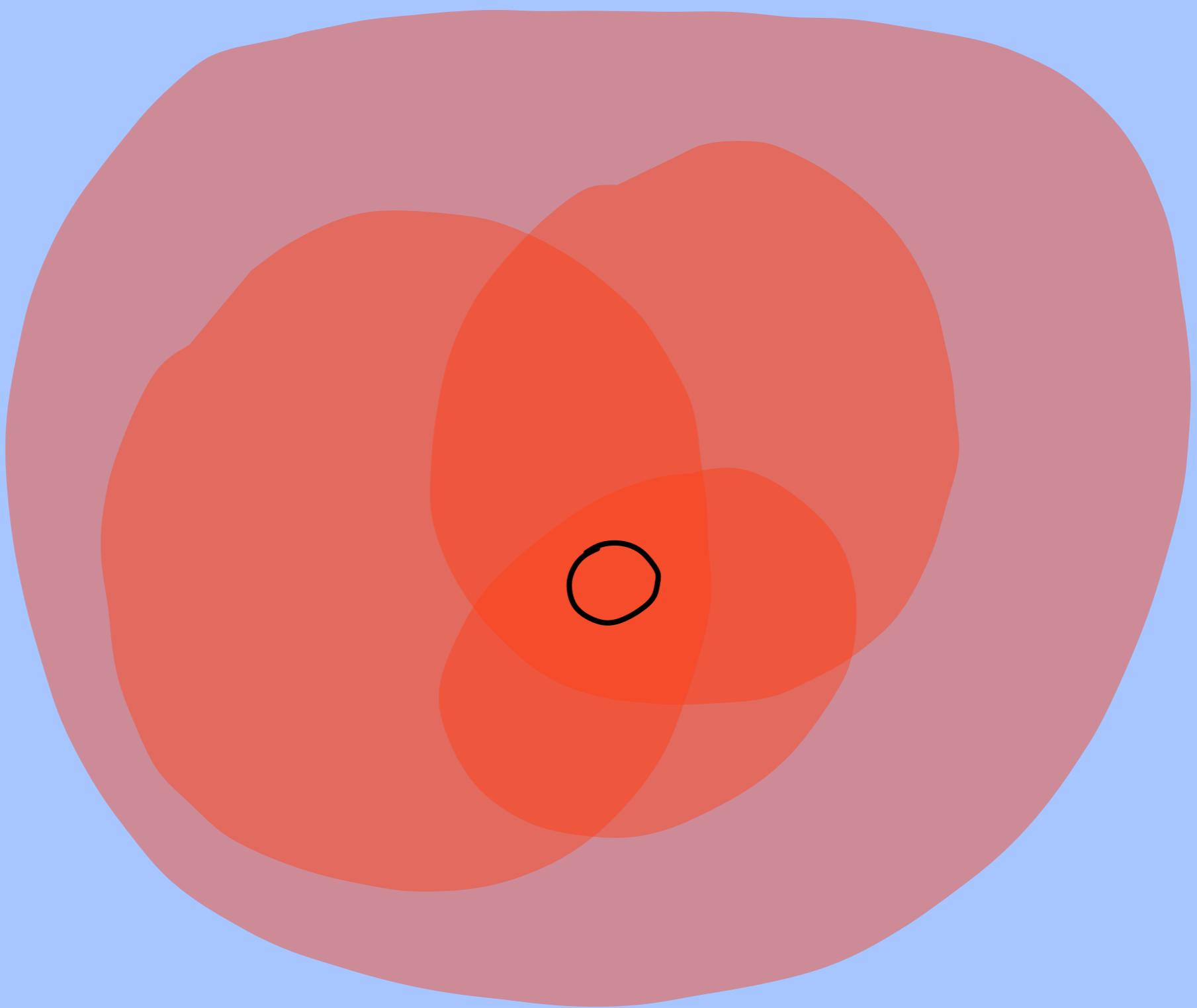
$\rightarrow$  EXPERIMENTS PINPOINT  $\lambda$



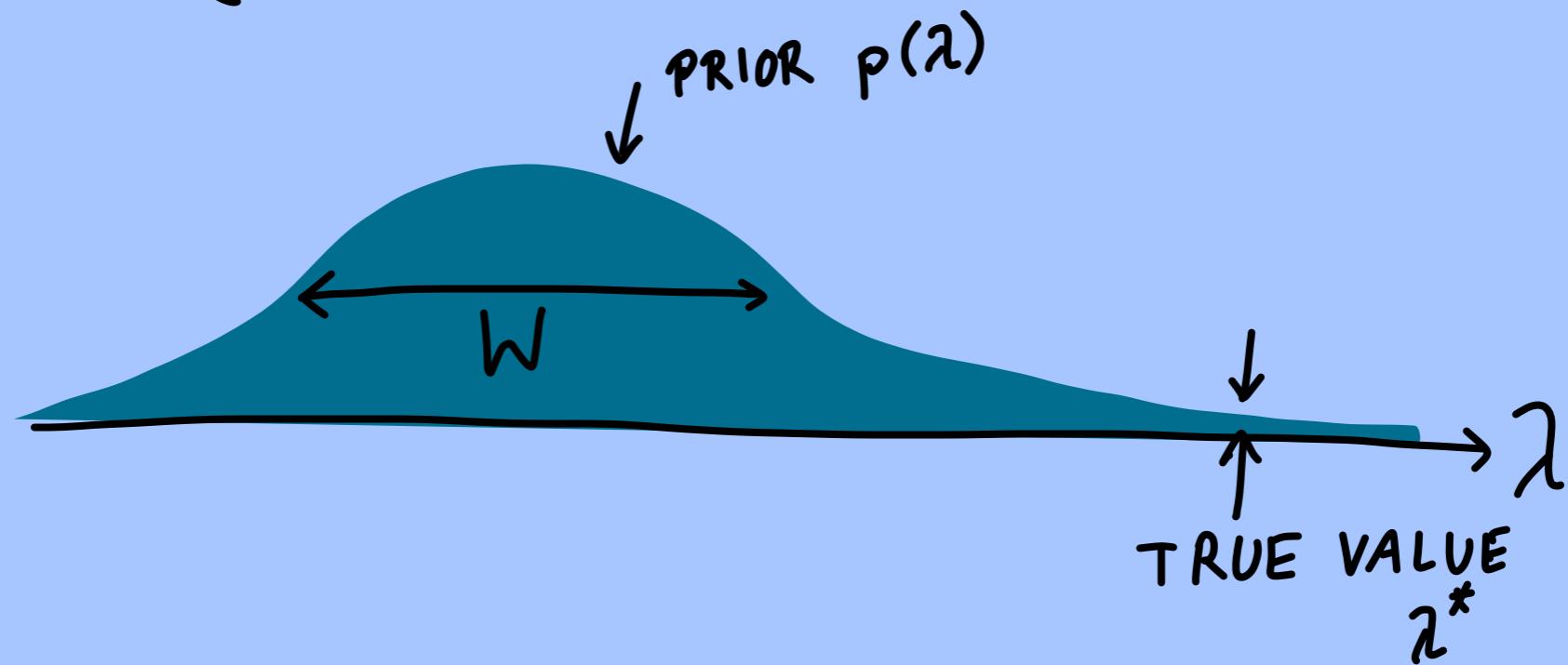
RATHER PINPOINT

$P(Z|\lambda)$

FOR ALL OBSERVABLES  $Z$   
"OF PRACTICAL INTEREST"



## 5.6

INFLUENCE OF THE PRIOR  
& CONSISTENCY OF BAYES

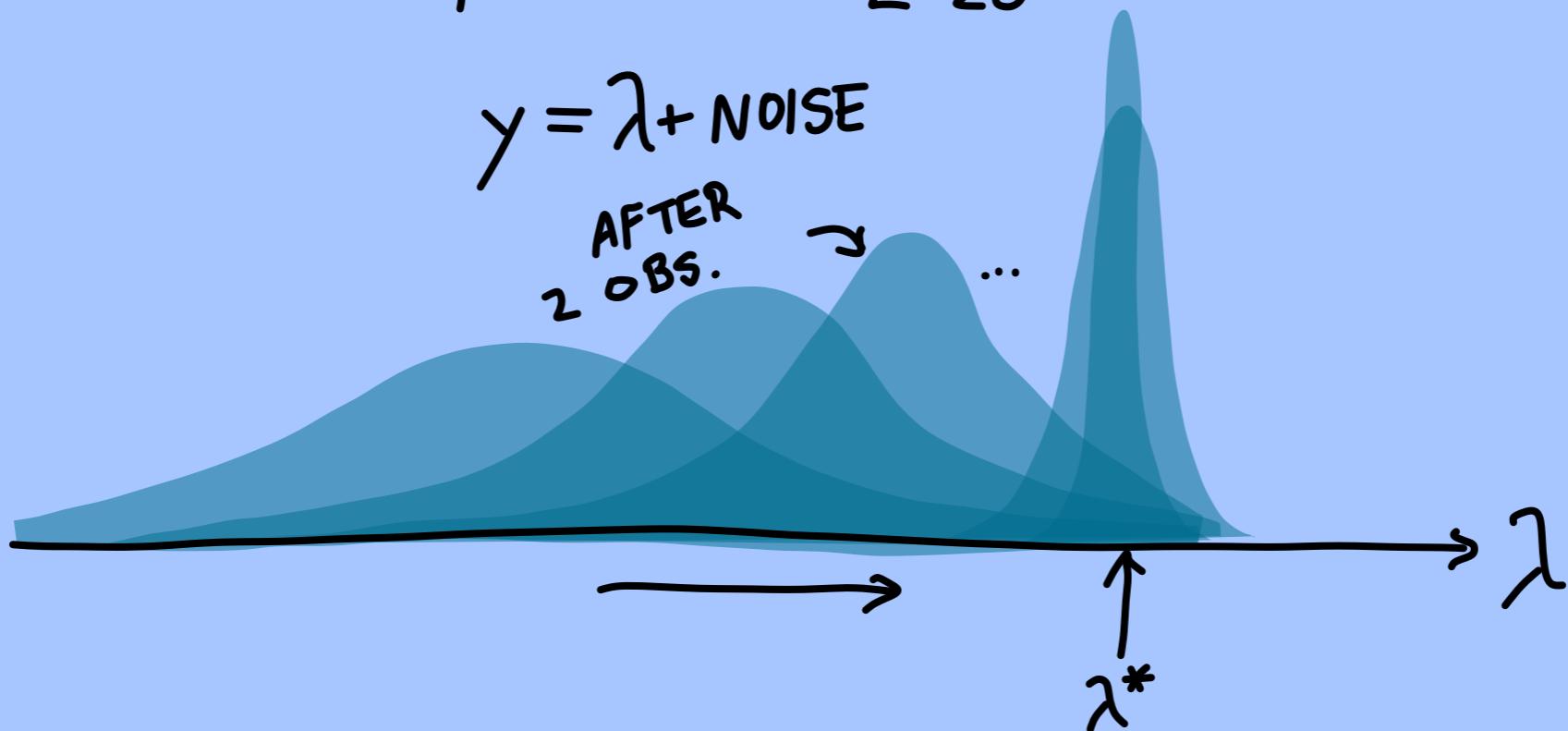
- ⇒ DO WE CONVERGE (AFTER  
MANY MSMTS) TOWARDS  $\lambda^*$ ?
- ⇒ HOW LONG DOES IT TAKE?

TOY MODEL:

GAUSSIAN

$$\text{PRIOR } p(\lambda) \sim \exp\left[-\frac{1}{2} \frac{\lambda^2}{w^2}\right]$$

$$\text{LIKELIHOOD } p(y|\lambda) \sim \exp\left[-\frac{1}{2\sigma^2} (y-\lambda)^2\right]$$



$$p(\lambda | y_1, \dots, y_N) \sim p(y_1|\lambda) \cdot \dots \cdot p(y_N|\lambda) p(\lambda)$$

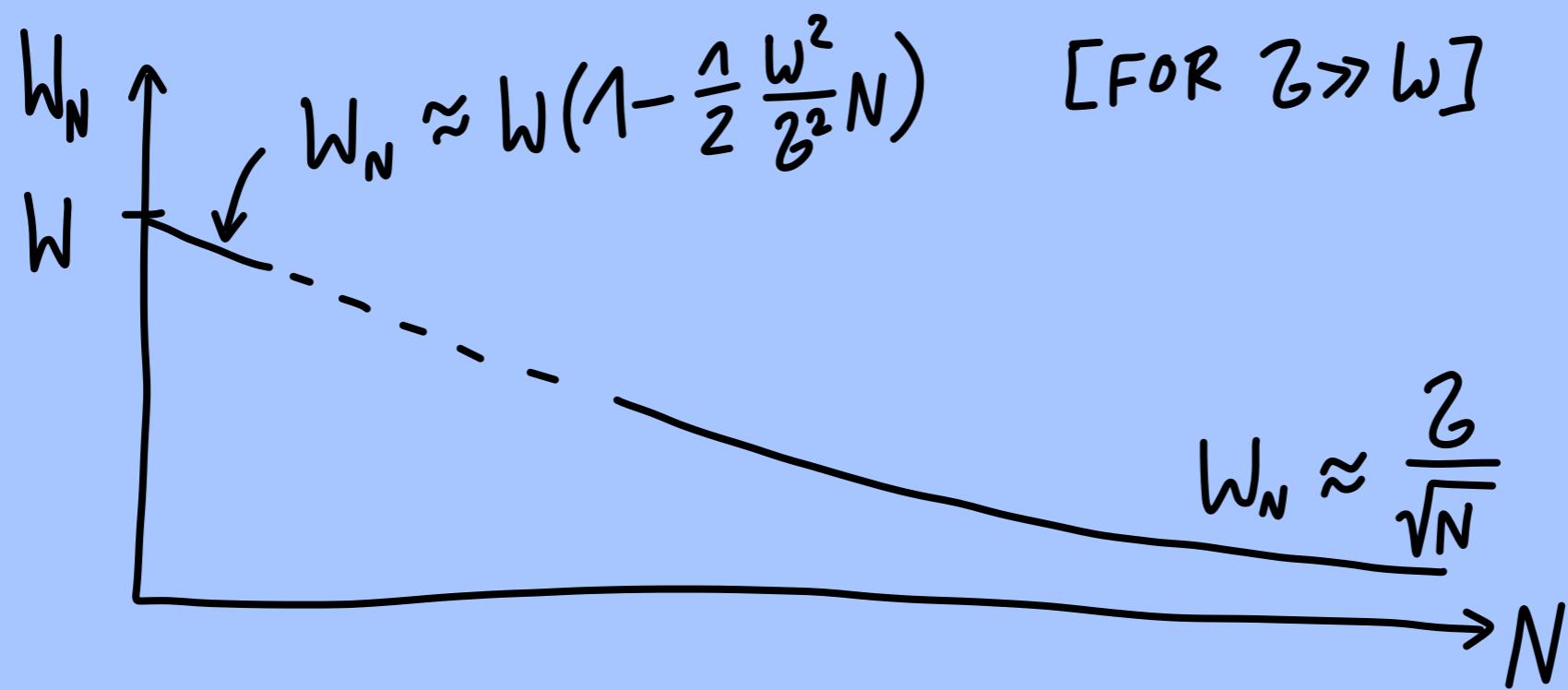
$$\ln p(\lambda | \{y\}) = -\frac{1}{2} \frac{\lambda^2}{W^2} - \frac{1}{2\sigma^2} \sum_{j=1}^N (y_j - \lambda)^2 + \underbrace{\text{const}}_{\text{INDEP. OF } \lambda}$$

$$= -\frac{1}{2} \frac{1}{W_N^2} (\lambda - \bar{\lambda}_N)^2 + \text{const}'$$

$$\frac{1}{W_N^2} = \frac{1}{W^2} + \frac{N}{\sigma^2} \Rightarrow W_N \text{ DECREASES WITH } N$$

$$W_N \xrightarrow[N \rightarrow \infty]{} \frac{\sigma}{\sqrt{N}}$$

$$\bar{\lambda}_N^{(\{y\})} = \frac{W_N^2}{\sigma^2} \sum_{j=1}^N y_j \quad \text{STOCHASTIC!}$$



NOW:  $\bar{\lambda}_N(\{y\})$

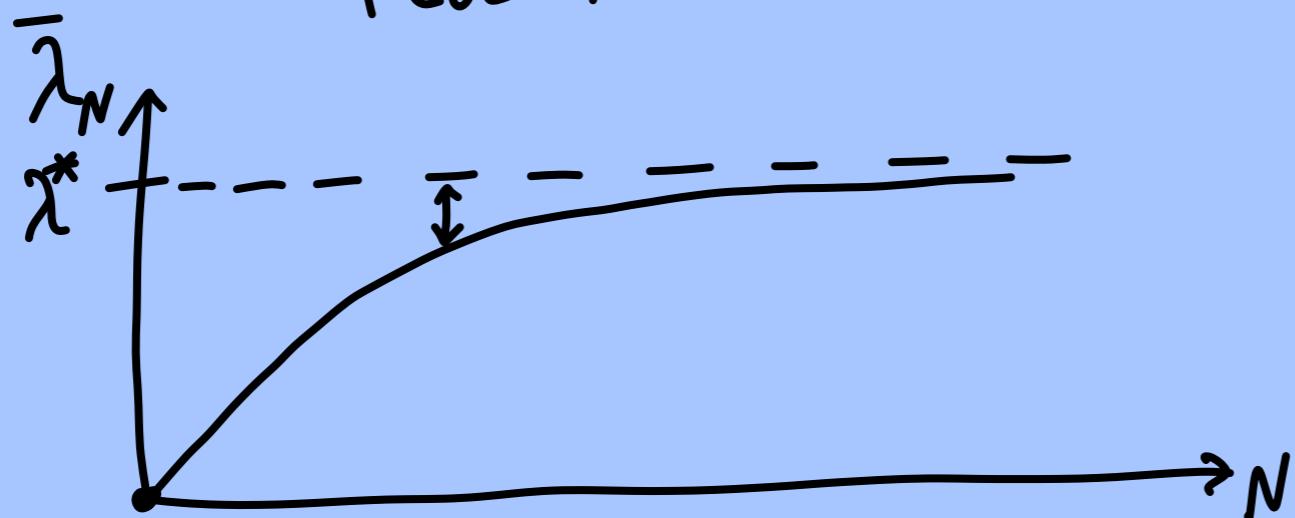
IMPORTANT:  $y_k$  ARE DRAWN FROM  $p(y|\lambda^*)$

TRUE

$$\Rightarrow \left\langle \sum_{k=1}^N y_k \right\rangle = N \cdot \lambda^*$$

$$\bar{\lambda}_N(\{y\}) \approx \frac{W_N^2}{Z^2} N \lambda^* \approx \begin{cases} \text{INITIALLY } (Z^2 \gg W^2 N) \\ \frac{W^2}{Z^2} N \lambda^* \Rightarrow \text{PULL TOWARDS } \lambda^* \\ \lambda^* \text{ FOR } N \rightarrow \infty \end{cases}$$

UP TO  
FLUCTUATIONS



1. WE CONVERGE  $\lambda \rightarrow \lambda^*$   
(NO MATTER HOW WRONG THE PRIOR)

2. STEPS NEEDED  
TO COME CLOSE  $|\lambda - \lambda^*| < \epsilon$

$$N \sim \frac{\lambda^*}{W} \frac{G^2}{\epsilon^2}$$

NOTE:  $\frac{\lambda^*}{W} \sim \sqrt{-\ln p(\lambda^*)}$

e.g.  $p(\lambda^*) \sim 10^{-10}$

STILL NEED ONLY SMALL NR OF MSMTS

NOTE: IF  $p(y|\lambda)$  HAS FAT TAILS, e.g.

$$p(y|\lambda) \sim \frac{1}{(y-\lambda)^2 + \sigma^2}$$

THIS WOULD BE  
MUCH WORSE ( $\Leftarrow$  SLOWER)

THEOREMS OF  
DOOB & SCHWARTZ  
1949 1965

$\Rightarrow$  CONSISTENCY OF BAYES

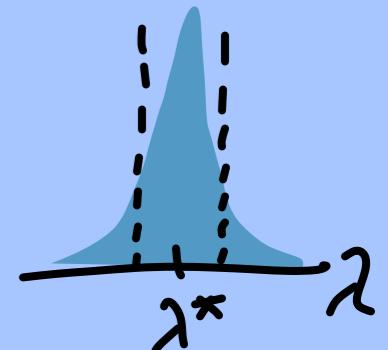
ESSENTIALLY : IF  $p(\lambda^*) \neq 0$

THEN "WITH PROB. 1" :

$p(\lambda | y_1, \dots, y_n)$   
CONCENTRATES

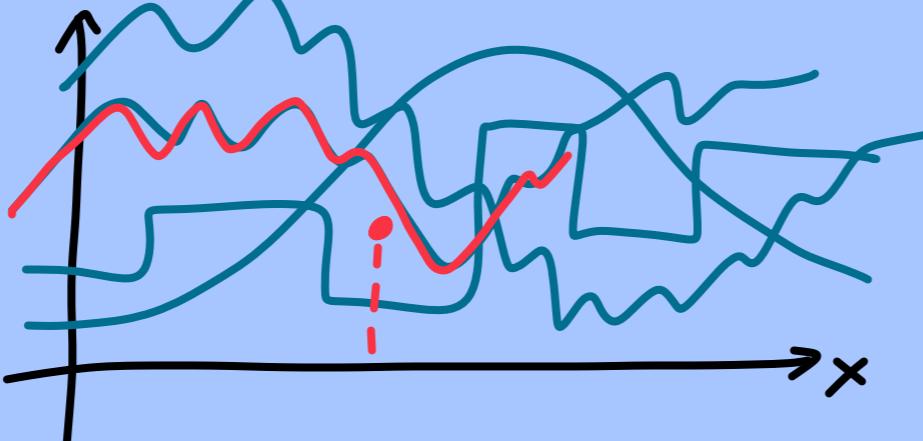
AROUND  $\lambda^*$  FOR  $N \rightarrow \infty$

&  $\langle g(\lambda) \rangle_{\lambda \sim p(\lambda | \{y_i\})} \xrightarrow{N \rightarrow \infty} g(\lambda^*)$



'MARTINGALES'

## 5.7

BAYES FOR GAUSSIAN  
RANDOM PROCESSES $\lambda = \text{FUNCTION } \phi(x)$ 

RANDOM PROCESSES / FIELDS

RANDOM FUNCTION  $\phi(t) \quad t \in \mathbb{R}$   
 $\phi(x) \quad x \in \mathbb{R}^d$

NOTATION

$$\{\phi(t)\}_{t \in \mathbb{R}} \geq \text{THE WHOLE FCT}$$

OR  $\phi(\cdot)$

SPECIFY JOINT PROB. DENSITY

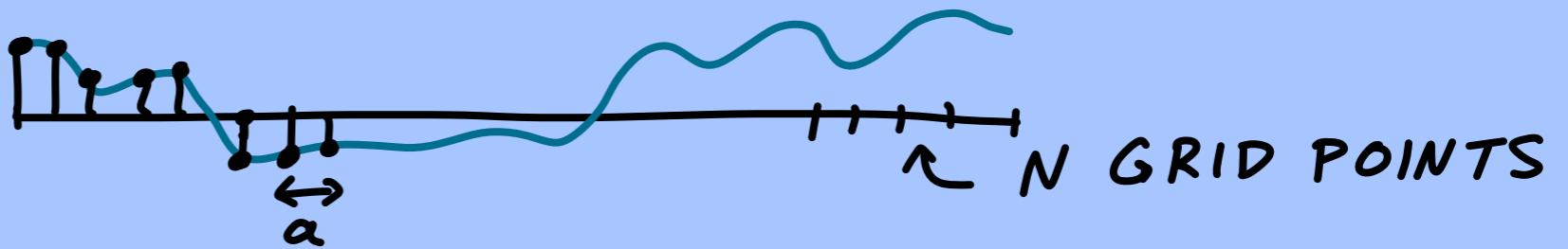
$p(\phi(t_1), \phi(t_2), \dots, \phi(t_n))$  FOR ANY  
SET  $\{t_1, \dots, t_n\}$

$$p(\phi(t_1), \phi(t_2)) = \int d\phi(t_3) p(\phi(t_1), \phi(t_2), \phi(t_3))$$

---

GAUSSIAN RANDOM FIELD  
 $p(\dots)$  IS GAUSSIAN

ON A FINITE DISCRETE LATTICE

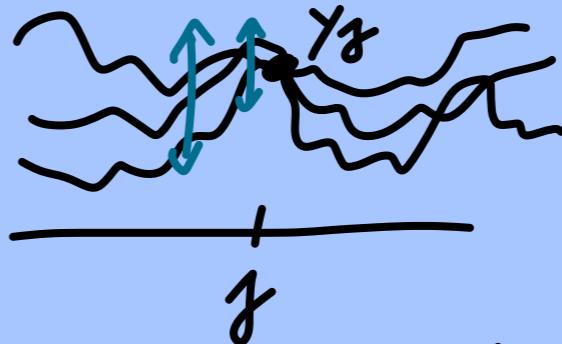


$$p(\phi_1, \phi_2, \dots, \phi_N) = \frac{1}{\sqrt{(2\pi)^N |\det C|}} \exp \left[ -\frac{1}{2} (\phi - \bar{\phi})^t C^{-1} (\phi - \bar{\phi}) \right]$$

$$\bar{\phi}_j = \langle \phi_j \rangle$$

$$C_{kj} \stackrel{\text{COVARIANCE}}{=} \langle (\phi - \bar{\phi})_k (\underbrace{\phi - \bar{\phi}}_{S\phi})_j \rangle_{\phi \sim p} = \langle S\phi_k S\phi_j \rangle$$

MSMT AT  
POSITION  $\gamma$   
WITH OUTCOME  $y_\gamma$



$$p(y_\gamma | \phi) = \frac{1}{\sqrt{2\pi} Z} \exp \left[ -\frac{1}{2} \frac{(y_\gamma - \phi_\gamma)^2}{Z^2} \right]$$

UNCERTAINTY

⇒ BAYES:

$$p(\phi | y_\gamma) = \frac{p(y_\gamma | \phi) p(\phi)}{\text{NORM. } \sim p(y_\gamma)}$$

NUMERATOR:

$$\begin{aligned} \ln p(\phi | y_\gamma) &= \text{const} - \frac{1}{2} \frac{(y_\gamma - \phi_\gamma)^2}{Z^2} - \frac{1}{2} \underbrace{(\phi - \bar{\phi})^T}_{\delta\phi^T} C^{-1} \underbrace{(\phi - \bar{\phi})}_{\delta\phi} \\ &= \text{const} - \frac{1}{2} \frac{(y_\gamma - \bar{\phi}_\gamma - \delta\phi_\gamma)^2}{Z^2} - \frac{1}{2} \delta\phi^T \boxed{C^{-1}} \delta\phi \\ S\phi &= \phi - \bar{\phi} \\ &= \text{const}' - \frac{1}{2} \delta\phi^T \boxed{C^{1/2}} \delta\phi + (y_\gamma - \bar{\phi}_\gamma) \frac{\delta\phi_\gamma}{Z^2} = \textcircled{*} \end{aligned}$$

FROM  $\frac{\delta\phi_j^2}{Z^2}$  TERM :

$$\left( C^{1-1} \right)_{nm} = \left( C^{-1} \right)_{nm} + \frac{1}{Z^2} S_{nj} S_{mj}$$

$$\textcircled{*} = \text{const}'' - \frac{1}{2} (\delta\phi - \bar{\delta\phi})^t C^{1-1} (\delta\phi - \bar{\delta\phi})$$

$$\bar{\delta\phi} = C'b$$

$$b_e = S_{ej} \frac{y_j - \bar{\phi}_j}{Z^2}$$

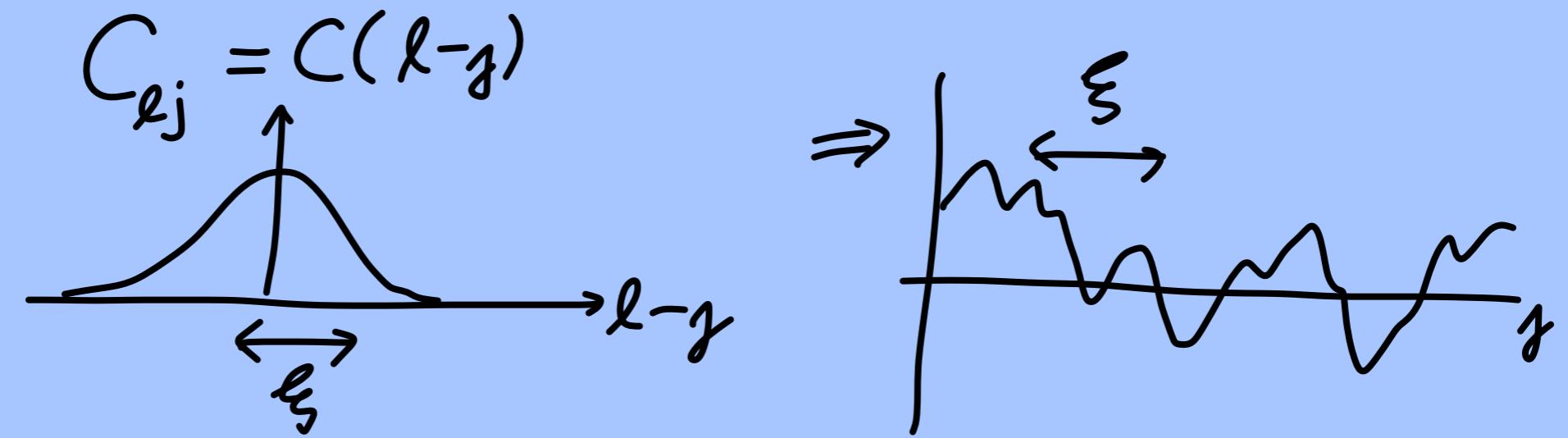
OVERALL:  $p(\phi|y)$  = NEW GAUSSIAN  
WITH

$$C'^{-1} = C^{-1} + \frac{1}{Z^2} |y\rangle\langle y|$$
$$(|y\rangle)_e = S_{ye} \quad (\langle y|)_e = S_{ye}$$

$$\bar{\phi}' = \bar{\phi} + C' |y\rangle \frac{y - \bar{\phi}}{Z^2}$$

$\Rightarrow$  CAN REPEAT  
THIS MULTIPLE TIMES,  
FOR EACH NEW MSMT!

# NUMERICS



# ANALYTICALLY TRACTABLE LIMITS

$$p(\phi) \sim \exp \left[ -\frac{k}{2} \underbrace{\int \left( \frac{\partial \phi}{\partial t} \right)^2 dt}_{\text{}} \right]$$

ON A LATTICE:  $\approx \sum_j \left( \frac{\phi_j - \phi_{j-1}}{\Delta t} \right)^2 \Delta t$

$$= \frac{1}{\Delta t} \phi^t \underbrace{\begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & \ddots & & \\ & & 0 & \ddots & \ddots & \\ & & & 0 & \ddots & \\ & & & & & 2 \end{bmatrix}}_{\sim C^{-1}} \phi$$

$$\int \left( \frac{\partial \phi}{\partial t} \right)^2 dt \stackrel{\uparrow}{=} - \int \phi \frac{\partial^2 \phi}{\partial t^2} dt$$

INTEGRATE  
BY PARTS

AFTER MULTIPLE MSMTS:

$$p(\phi | Y(t_0) = y_0, Y(t_n) = y_1, \dots)$$

$$\sim \exp \left[ -\frac{1}{2} K \int \left( \frac{\partial \phi}{\partial t} \right)^2 dt - \frac{1}{2\sigma^2} \sum_{\ell} (\phi(t_{\ell}) - y_{\ell})^2 \right]$$

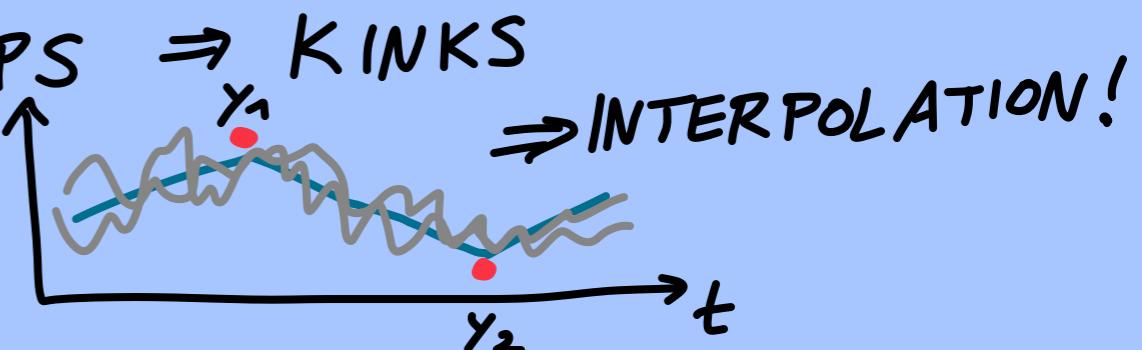
FIND  $\bar{\phi}(t)$  FOR POSTERIOR BY  
FINDING EXTREMUM OF EXPONENT.<sup>!</sup>

$$\frac{\delta}{\delta \phi(t)} \text{EXPONENT} = 0$$

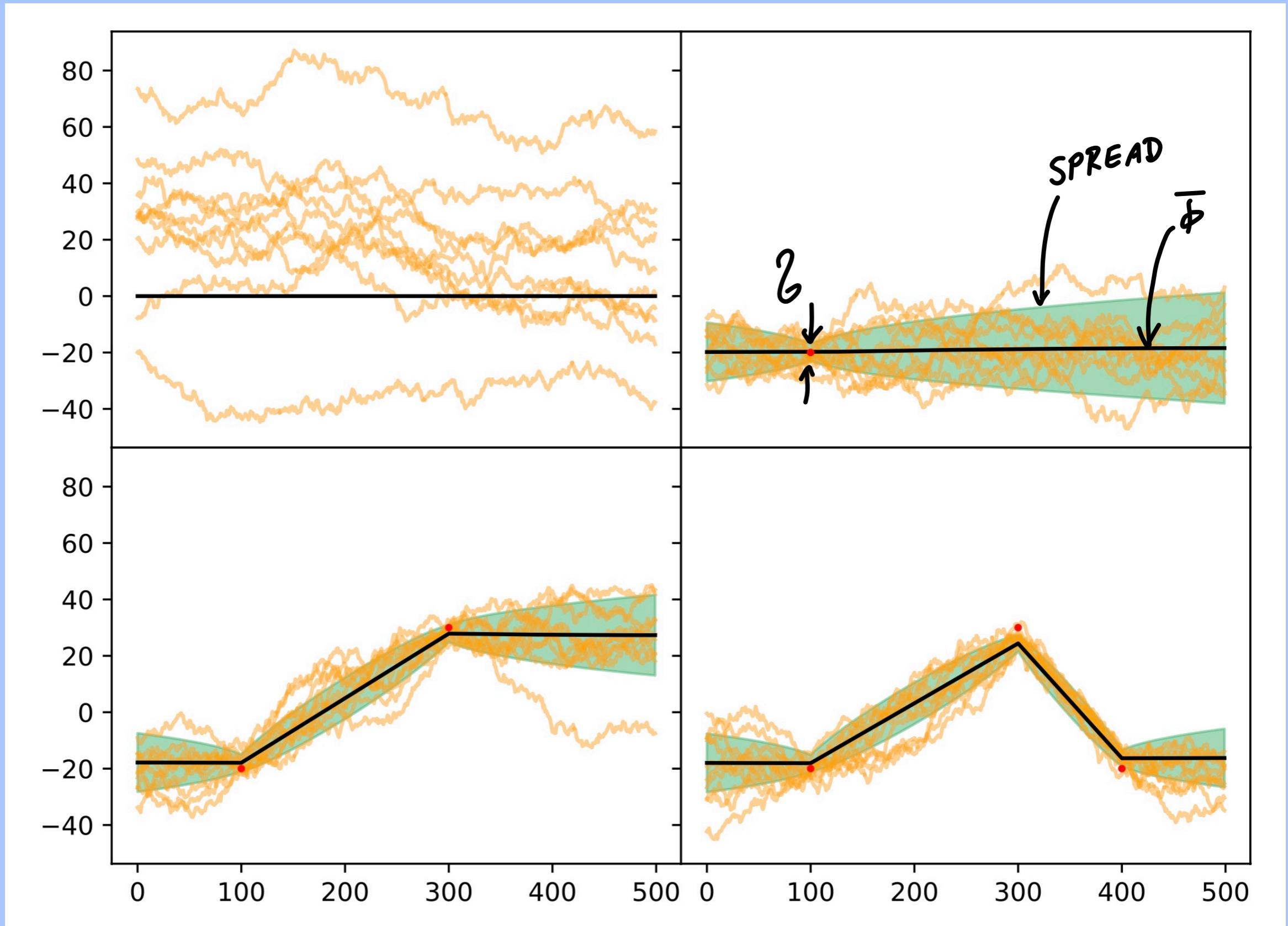
$$K \partial_t^2 \phi(t) - \frac{1}{\sigma^2} \sum_{\ell} \delta(t - t_{\ell}) (\phi(t_{\ell}) - y_{\ell}) = 0$$

$t \neq t_{\ell}$ :  $\partial_t^2 \phi = 0 \Rightarrow$  STRAIGHT LINES!

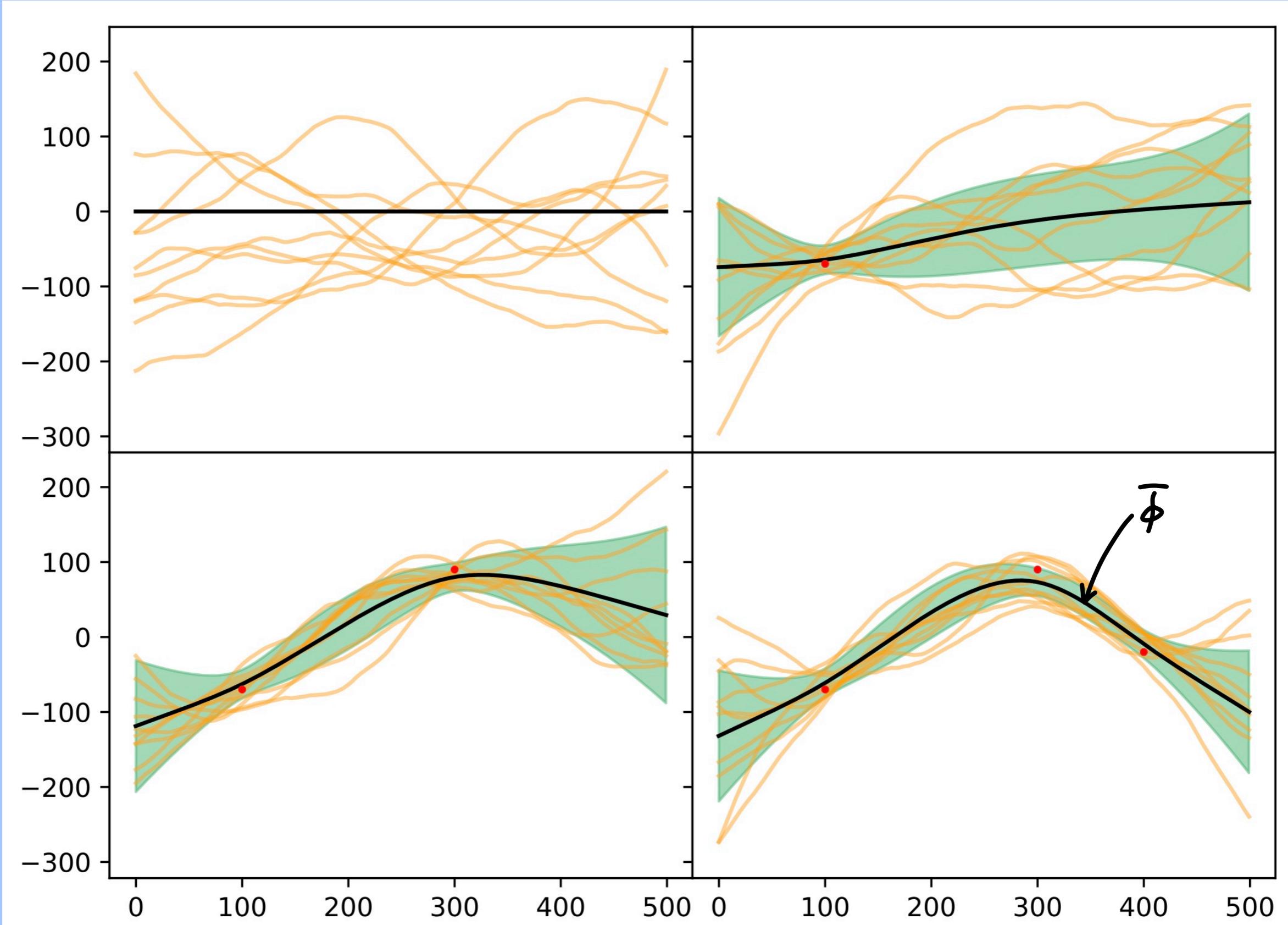
$t = t_{\ell}$ :  $\partial_t \phi$  JUMPS  $\Rightarrow$  KINKS



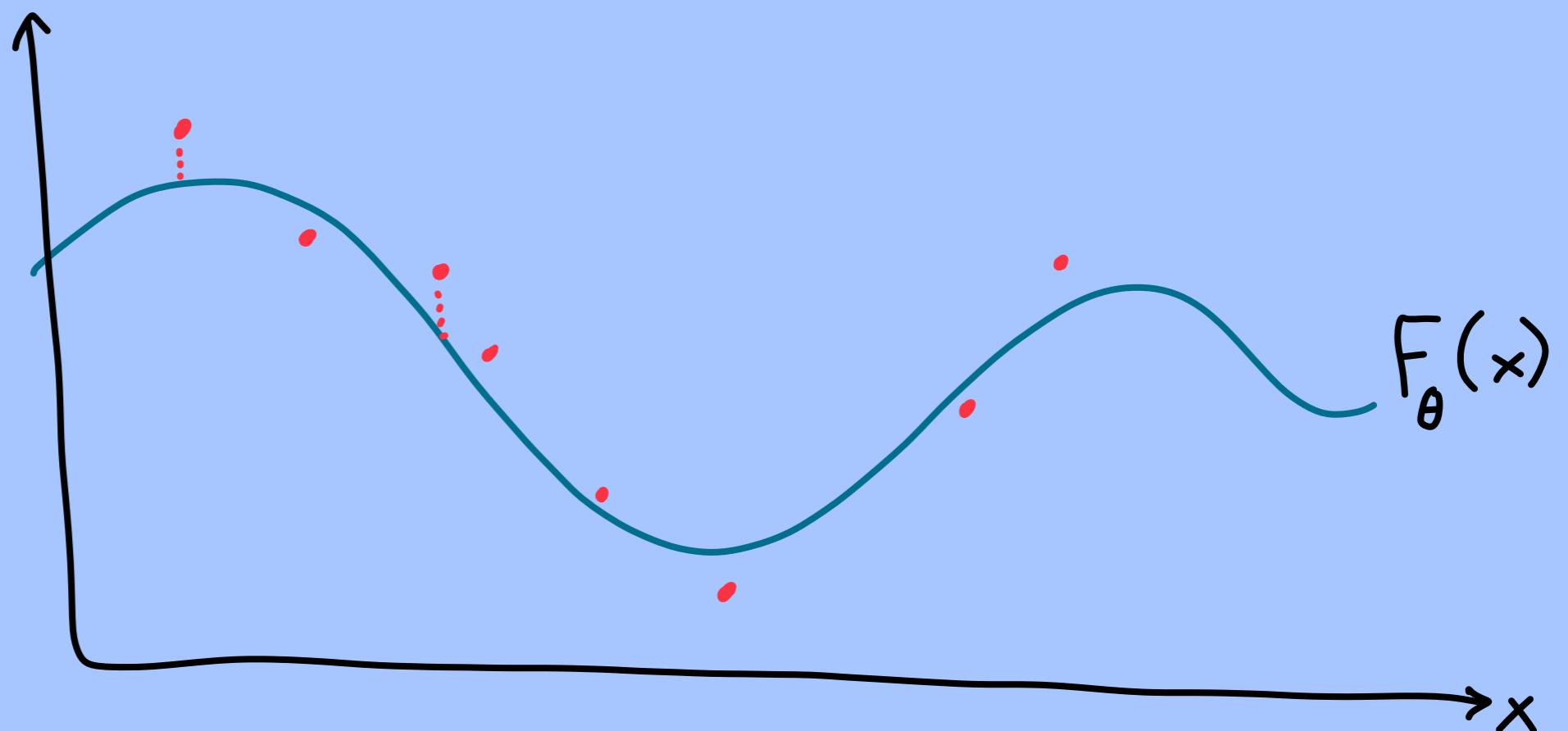
$$\ln p(\phi) \sim - \int \left( \frac{\partial \phi}{\partial t} \right)^2 dt \rightarrow \text{RANDOM WALK, LINEAR INTERPOLATION}$$



$$\ln p(\phi) \sim - \int \left( \frac{\partial^2 \phi}{\partial t^2} \right)^2 dt \rightarrow \text{SPLINE-TYPE INTERPOLATION}$$



## 5.8

BAYES AND THE LOSS FUNCTION  
IN NEURAL-NETWORK TRAINING

$$y^{(j)} = F_\theta(x^{(j)}) + \text{NOISE } \delta y^{(j)}$$

SAMPLES

ASSUME:  $y^{(g)}$  IDENTICALLY  
INDEPENDENTLY  
DISTRIBUTED  
GAUSSIAN MEAN ZERO  
VARIANCE  $\sigma^2$

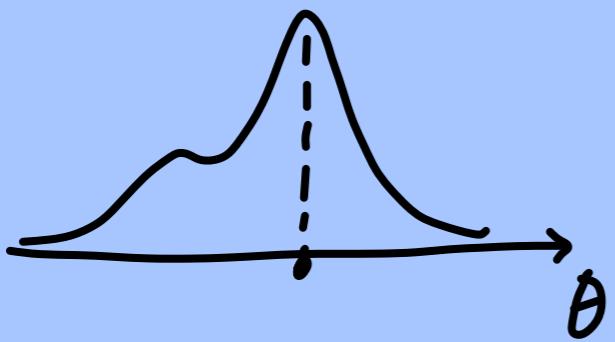
$\Rightarrow$  LIKELIHOOD

$$\ln p(y^{(g)} | \theta) = \text{const} - \frac{1}{2\sigma^2} \|y^{(g)} - F_\theta(x^{(g)})\|^2$$

$\Rightarrow$  AFTER OBSERVING ALL  $y^{(g)}$   $g=1\dots N$   
USE BAYES  $\Rightarrow$

$$\underbrace{\ln p(\theta | \{y\})}_{\text{POSTERIOR}} = \sum_{g=1}^N \ln p(y^{(g)} | \theta) + \underbrace{\ln p(\theta)}_{\text{PRIOR}} + \text{const}$$

$$= -\frac{1}{2\sigma^2} \sum_{g=1}^N \|y^{(g)} - F_\theta(x^{(g)})\|^2 + \ln p(\theta) + \text{const}'$$



MAXIMIZE POSTERIOR



MINIMIZE  $\sim \text{MSE}$

$$\frac{1}{2\sigma^2} \sum_{j=1}^N \|y^{(j)} - \dots\|^2 - \ln p(\theta)$$

INTERPRET AS LOSS FUNCTION  $\mathcal{L}(\theta)$

WITH  $-\ln p(\theta) = \text{REGULARIZATION TERMS}$

(e.g. GAUSSIAN  $\ln p(\theta) \sim \theta(\theta^2)$   
 $\Rightarrow L_2 - \text{REGULAR.}$ )

$p(\mathcal{S}_Y)$  DIFFERENT  $\Rightarrow$  OTHER LOSS FCT ✓

BEWARE: MODEL "F + NOISE" IS TYPICALLY WRONG

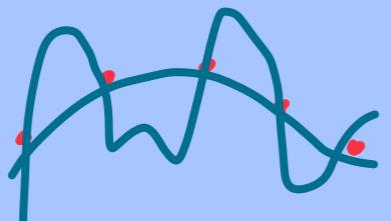


## 5.9

## INDUCTIVE BIAS &amp; BAYES

DEDUCTION : LOGICAL CONCLUSIONS  
 $(A \rightarrow B, A \text{ TRUE}) \Rightarrow B \text{ TRUE}$

INDUCTION :  $A \rightarrow B$   
 $A' \rightarrow B'$   
 $\xrightarrow{?} X \rightarrow Y$



GUESS A RULE  
 FROM EXAMPLES

PREFER SOME  
 HYPOTHESES OVER  
 OTHERS

= "INDUCTIVE BIAS"

= CHOICE OF PRIOR  
 IN BAYES

EXAMPLES :

- SMOOTH GAUSSIAN PROCESSES
- NN WITH CERTAIN STRUCTURE &  
LOSS FCT

⇒ ALLOWS TO GENERALIZE  
TO UNSEEN SITUATIONS

BUT: NO GUARANTEE!

✓ SHANNON,  
CODING,  
ENTROPY  
COMPRESSION

✓ BAYES

UPDATING PROBABILITIES  
CONDITIONAL ENTROPY  
GAUSSIAN RANDOM PROCESSES  
CONNECTION TO NN TRAINING

✓ INFORMATION THEORY II

FISHER INFORMATION  
KULLBACK-LEIBLER DIVERGENCE  
MUTUAL INFORMATION

| STOCHASTIC COMPONENTS IN  
NEURAL NETWORKS

| LEARNING PROBABILITY  
DISTRIBUTIONS  
| GENERATIVE  
APPROACHES

6.

## INFORMATION THEORY II

"HOW SENSITIVE IS  
A PROBABILITY DISTRIBUTION  
TO CHANGES IN SOME PARAMETER?"

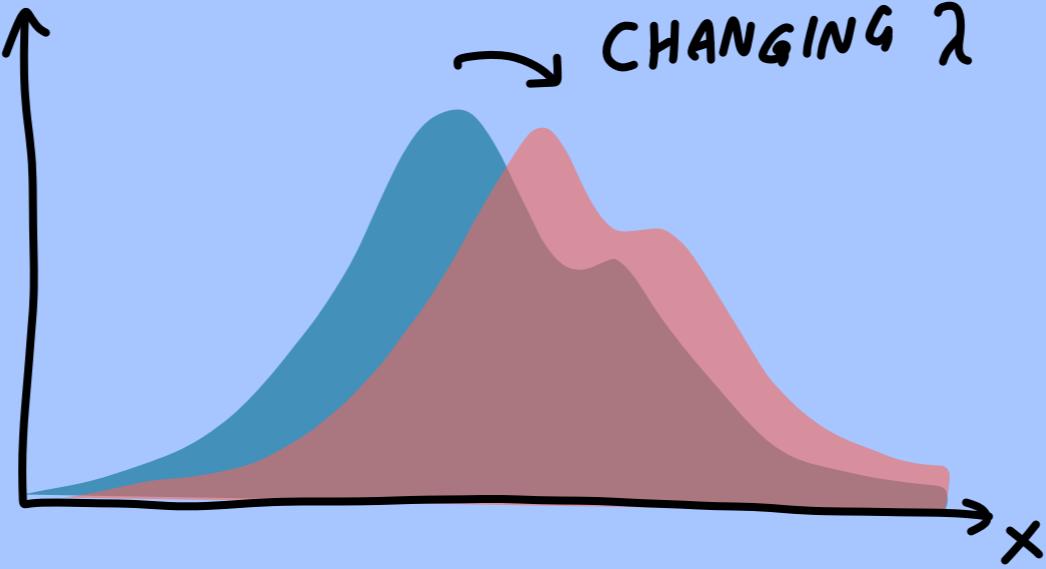
"HOW CAN WE COMPARE TWO  
PROBABILITY DISTRIBUTIONS?"

"HOW STRONG IS THE DEPENDENCE  
OF ONE RANDOM VARIABLE ON ANOTHER?"

## 6.1

## FISHER INFORMATION

$$p(x|\lambda)$$



GOAL: OBSERVE  $x \rightarrow$  GUESS  $\lambda$

WANT:  $\frac{\partial p(x|\lambda)}{\partial \lambda}$  LARGE !

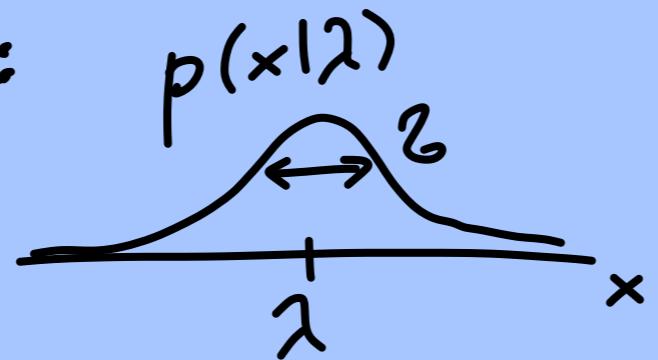
"FISHER INFORMATION":

$$I(\lambda) \equiv \left\langle \left( \frac{\partial \ln p(x|\lambda)}{\partial \lambda} \right)^2 \right\rangle_{x \sim p(x|\lambda)}$$

$$= - \left\langle \frac{\partial^2}{\partial \lambda^2} \ln p(x|\lambda) \right\rangle_{x \sim p(x|\lambda)}$$

$I$  LARGE  $\Rightarrow$   
EASY TO  
TELL  $\lambda$  BY  
OBSERVING  
 $x$  !

EXAMPLE:



$$\ln p(x|\lambda) = \text{const} - \frac{1}{2\lambda^2}(x-\lambda)^2$$

$$\frac{\partial}{\partial \lambda} \ln p = \frac{1}{\lambda^2}(x-\lambda)$$

$$I = \left\langle \left[ \frac{1}{\lambda^2}(x-\lambda) \right]^2 \right\rangle_{x \sim p(x|\lambda)}$$

$$= \frac{1}{\lambda^4} \cdot \lambda^2 = \frac{1}{\lambda^2}$$

## MEANING OF $\hat{\lambda}$ :

"UNBIASED ESTIMATOR"  $\hat{\lambda}(x)$

$$\langle \hat{\lambda}(x) \rangle_{x \sim p(x|\lambda)} = \lambda$$

REMEMBER:  $(\vec{a} \cdot \vec{b})^2 \leq |\vec{a}|^2 \cdot |\vec{b}|^2$

CAUCHY-SCHWARTZ INEQUALITY

$$\int (\hat{\lambda}(x) - \lambda) p(x|\lambda) dx \stackrel{!}{=} 0 \quad \forall \lambda$$

$$\frac{\partial}{\partial \lambda} \Rightarrow \int (\hat{\lambda}(x) - \lambda) \underbrace{\frac{\partial p(x|\lambda)}{\partial \lambda}}_{P} dx - 1 = 0$$

$$\int [(\hat{\lambda} - \lambda) \sqrt{P}] [\sqrt{P} \frac{\partial}{\partial \lambda} \ln P] dx = 1 = 1^2$$

$$\leq \left( \int [(\hat{\lambda} - \lambda) \sqrt{P}]^2 dx \right) \cdot \left( \int [\sqrt{P} \frac{\partial}{\partial \lambda} \ln P]^2 dx \right)$$

EXAMPLE:  
 $\hat{\lambda}(x) = \frac{x_1 + \dots + x_N}{N}$

FOR MEAN

$$1 \leq \left( \int (\hat{\lambda}(x) - \lambda)^2 p(x|\lambda) dx \right) \cdot \left( \int \left( \frac{\partial \ln p}{\partial \lambda} \right)^2 p(x|\lambda) dx \right)$$

$\text{Var } \hat{\lambda}(x)$        $I(\lambda)$

$$\text{Var } \hat{\lambda}(x) \geq \frac{1}{I(\lambda)}$$

"CRAMÉR-RAO  
BOUND"

PRECISION LIMITED BY  $\frac{1}{I}$

EXAMPLE:  $x \in \mathbb{R}^N$  INDEP. OBS., MEAN  $\lambda$ , VAR.  $\sigma^2$

$$\text{Var } \hat{\lambda}(x) \geq \frac{\sigma^2}{N}$$

## CONNECTION TO BAYES

LET  $\hat{\lambda}_N(x) = \int \lambda \underbrace{p(\lambda | x)}_{\text{POSTERIOR}} d\lambda$   
 (MIN. MSE BAYES ESTIMATOR)

SUPPOSE  $x_1, x_2, \dots, x_N$  ARE INDEP. SAMPLES  
 $p(x_g | \lambda^*)$  ↴ TRUE VALUE

$$\Rightarrow \hat{\lambda}_N(x_1, \dots, x_N) - \lambda^* \xrightarrow{N \rightarrow \infty} \begin{array}{l} \text{GAUSSIAN} \\ \text{MEAN}^0 \\ \text{VARIANCE } \frac{1}{N I(\lambda^*)} \end{array}$$

BAYES ESTIMATOR IS ASYMPTOTICALLY EFFICIENT!

AND: INDEP. OF PRIOR

# MATRIX FISHER INFORMATION

$$I_{\text{fish}}(\lambda) = \begin{Bmatrix} \frac{\partial \ln p(x|\lambda)}{\partial \lambda_g} & \frac{\partial \ln p(x|\lambda)}{\partial \lambda_e} \\ x \sim p(x|\lambda) \end{Bmatrix} \geq 0 \quad (\text{POS. SEMIDEF.: } u^T I u \geq 0)$$

$$= - \left\langle \frac{\partial^2}{\partial \lambda_g \partial \lambda_e} \ln p(x|\lambda) \right\rangle_x$$

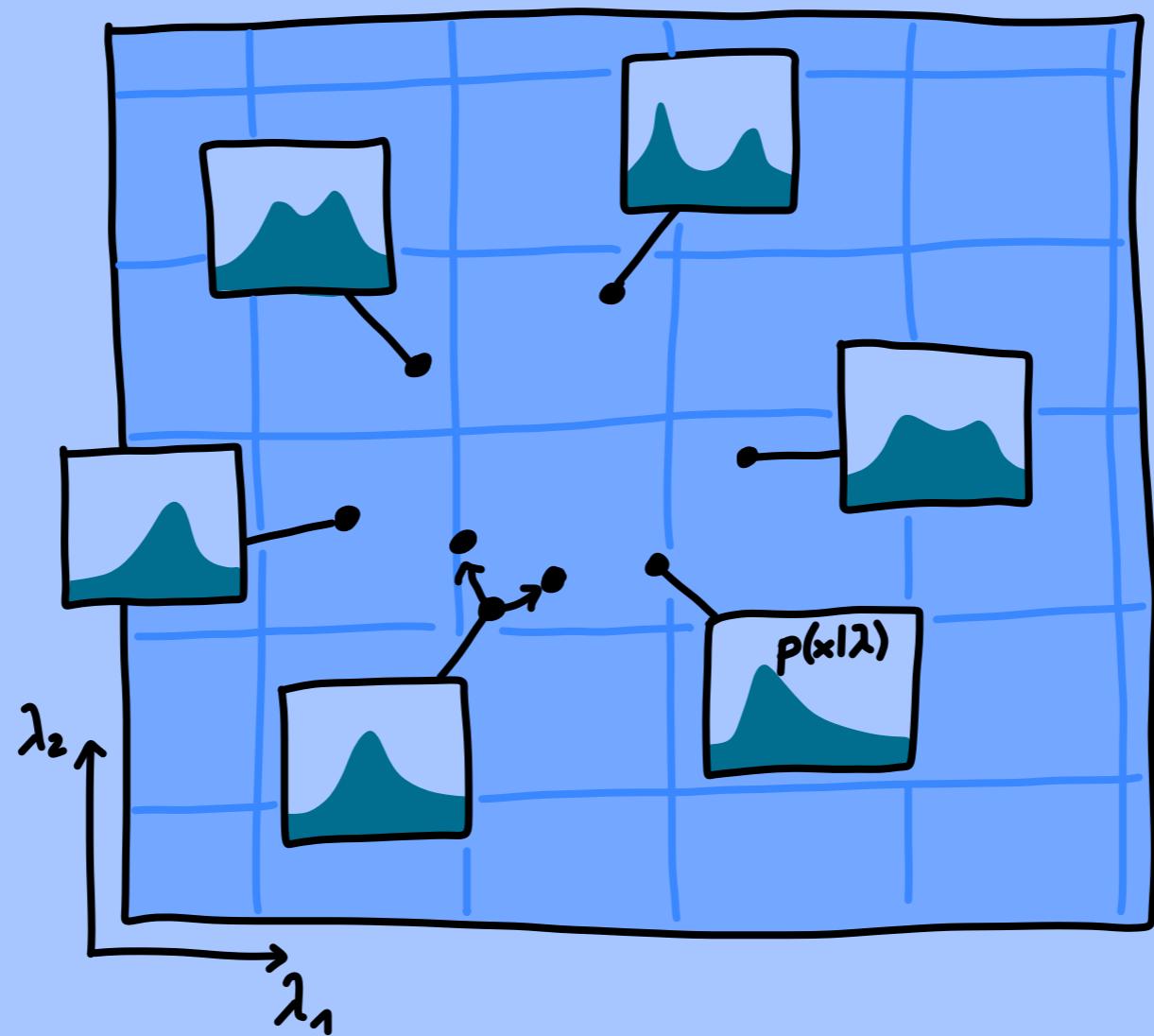
C.R. BOUND:

$$\text{Cov} \hat{\lambda}(x) \geq I(\lambda)^{-1}$$

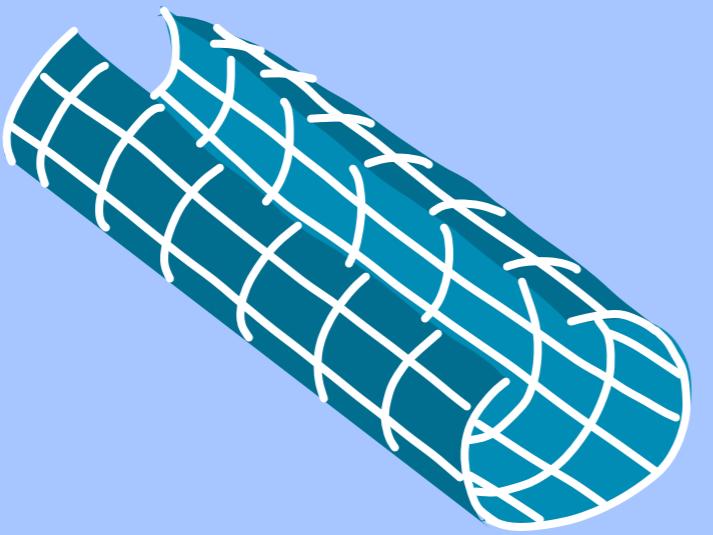
$$\begin{aligned} "A \geq B \Leftrightarrow A - B \geq 0" \\ \Leftrightarrow A - B \text{ POS. SEMIDEF.} \end{aligned}$$

$I(\lambda) \geq 0$  CAN BE UNDERSTOOD  
AS A METRIC!

$$S\lambda^t I S\lambda = \text{DISTANCE}^2$$



⇒ "INFORMATION  
GEOMETRY"



INTRINSIC  
CURVATURE  $\equiv 0$



INTRINSIC  
CURVATURE  $\neq 0$

# CHANGING THE COORDINATE SYSTEM "REPARAMETRIZATION"

$$\lambda = \varphi(\tilde{\lambda})$$

↗ BIJECTIVE

$$J_{\epsilon j} = \frac{\partial \lambda_\epsilon}{\partial \tilde{\lambda}_j}$$

$$I_{\tilde{\lambda}}(\tilde{\lambda}) = J^t \quad I_\lambda(\lambda) \quad J$$

COMPARE : CHANGE OF PROB. DENSITY

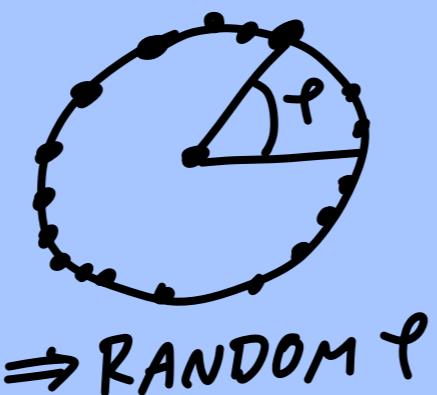
[REMEMBER 1D:  $p_{\tilde{\lambda}}(\tilde{\lambda})d\tilde{\lambda} = p_\lambda(\lambda)d\lambda$   
 $d\lambda = \frac{\partial \lambda}{\partial \tilde{\lambda}} d\tilde{\lambda}$ ]

$$p_{\tilde{\lambda}}(\tilde{\lambda}) = p_\lambda(\lambda) \left| \det \frac{\partial \lambda}{\partial \tilde{\lambda}} \right|$$

↗  $\lambda = \lambda(\tilde{\lambda})$

# JEFFREY'S PRIOR

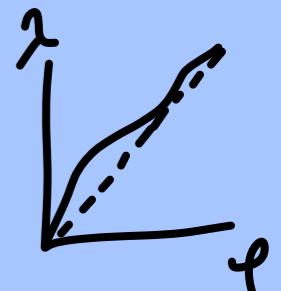
EXAMPLE:



$\Rightarrow$  RANDOM  $\theta$

$\theta =$  ANGLE  
(MEANING)  
SYMMETRY  
(ISOTROPIC)

}  $\Rightarrow$  ASSUME  
 $p(\theta) = \text{const}$



BUT: IF MEANING OF  $\lambda$  UNKNOWN  
 $\Rightarrow$  PRIOR = ??

IDEA IN BAYES: EXPLOIT  $p(x|\lambda)$   
TO UNDERSTAND MEANING &  
SYMMETRY OF  $\lambda$

CHOOSE

$$p(\lambda) \sim \sqrt{\det I(\lambda)}$$

"JEFFREY PRIOR"

$\sim$  GENERALIZATION OF UNIFORM PRIOR

PROOF:

$$\parallel P_{\tilde{\lambda}}(\tilde{\lambda}) = p_{\lambda}(\lambda) \left| \det J \right| \parallel \frac{\partial \lambda}{\partial \tilde{\lambda}}$$

Q: IS THAT FULFILLED BY  $p_{\lambda} \sim \sqrt{\det I_{\lambda}}$  &  $p_{\tilde{\lambda}} \sim \sqrt{\det I_{\tilde{\lambda}}}$  ?

$$\begin{aligned} \parallel \sqrt{\det I_{\tilde{\lambda}}(\tilde{\lambda})} &= \sqrt{\det J^t I_{\lambda} J} \\ &= \sqrt{(\det J J^t) (\det I_{\lambda})} \\ &= \left| \det J \right| \sqrt{\det I_{\lambda}} \quad \text{YES ✓} \end{aligned}$$

$\Rightarrow$  JEFFREY PRIOR REPRESENTS  
A COOR.-SYS.-INDEP. PROB. DISTR.

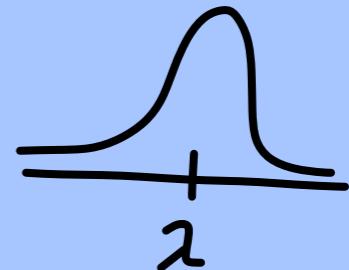


"NON-INFORMATIVE  
PRIOR"

EXAMPLES:

$$-\frac{(x-\lambda)^2}{2\sigma^2}$$

$$p(x|\lambda) \sim e$$



JEFFR.:  $p(\lambda) = \underline{\text{const}}$

$$p(x|\lambda) \sim \frac{e^{-\frac{x^2}{2\lambda^2}}}{\lambda}$$

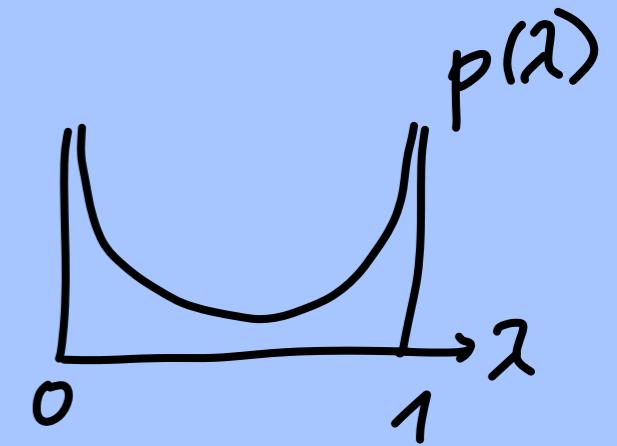
∴  $p(\lambda) \sim \underline{\frac{1}{\lambda}}$

$$\tilde{\lambda} = \ln \lambda \Rightarrow p_{\tilde{\lambda}}(\tilde{\lambda}) = \underline{\text{const}}$$

BERNOULLI

$$p(y \mid \lambda) = \begin{cases} \lambda & y=1 \\ 1-\lambda & y=0 \end{cases}$$

$$\Rightarrow \dots \Rightarrow p_{\lambda}(\lambda) \sim \frac{1}{\sqrt{2(1-\lambda)}}$$



$$\lambda = \sin^2(\tilde{\lambda})$$

$$\sim p_{\lambda}(\tilde{\lambda}) = \text{UNIFORM ON } [-\frac{\pi}{2}, \frac{\pi}{2}]$$

## 6.2

## NATURAL GRADIENT

$$\delta\theta = -\eta \nabla_{\theta} \mathcal{L}_{\theta}(\theta)$$

TD:  $\delta\theta = -\eta \frac{\partial \mathcal{L}_{\theta}(\theta)}{\partial \theta}$

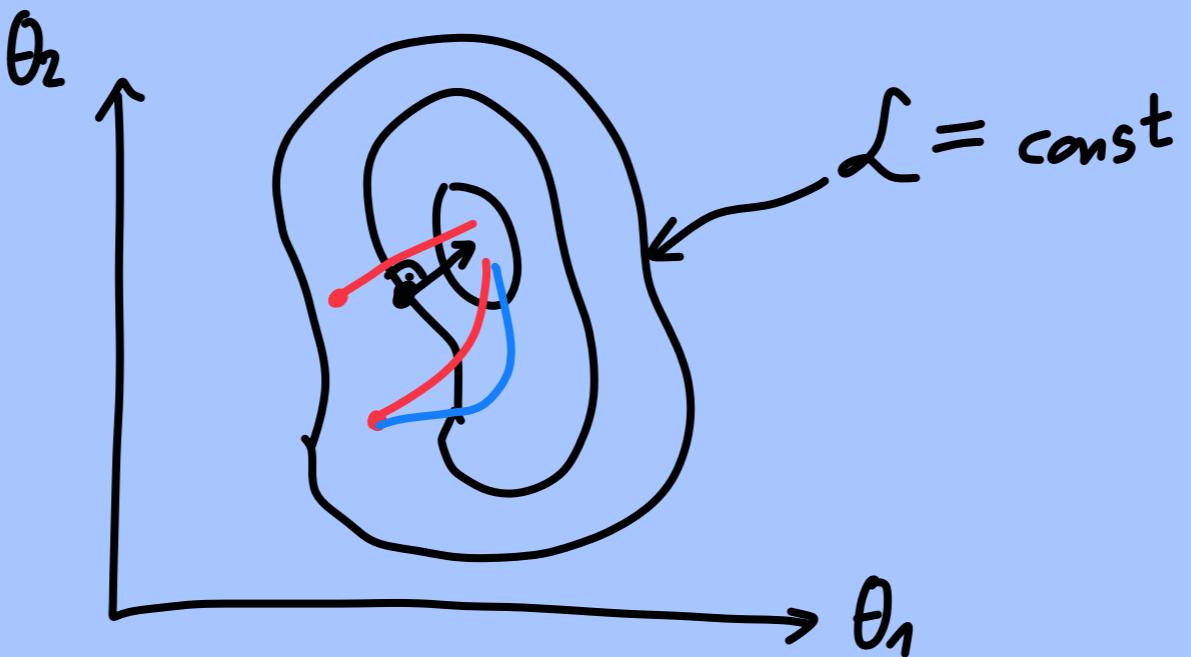
$$\boxed{\theta = \mu \tilde{\theta}}$$

$$\mu \delta\tilde{\theta} = -\eta \frac{1}{\mu} \frac{\partial \mathcal{L}_{\tilde{\theta}}(\tilde{\theta})}{\partial \tilde{\theta}}$$

$$\mathcal{L}_{\theta}(\theta) = \mathcal{L}_{\tilde{\theta}}(\tilde{\theta}(\theta))$$

$$\delta\tilde{\theta} = -\eta \frac{1}{\mu^2} \frac{\partial \mathcal{L}_{\tilde{\theta}}}{\partial \tilde{\theta}}$$

IS NOT THE  
SAME AS SOMEONE  
WORKING DIRECTLY IN  
THE  $\tilde{\theta}$  SYSTEM WOULD  
ASSUME:  $\delta\tilde{\theta} = -\eta \frac{\partial \mathcal{L}_{\tilde{\theta}}}{\partial \tilde{\theta}}$



$$\delta\theta = -\vec{\nabla}_{\theta} \mathcal{L}_{\theta}(\theta)$$

$$\& \quad \delta\tilde{\theta} = -\vec{\nabla}_{\tilde{\theta}} \mathcal{L}_{\tilde{\theta}}(\tilde{\theta})$$

ARE NOT RELATED

BY THE  $\theta \leftrightarrow \tilde{\theta}$  COORD. TRANSF.

i.e. WE DO NOT HAVE

$$\delta\tilde{\theta} = \underbrace{\frac{\partial \tilde{\theta}}{\partial \theta}}_{\text{?}} \delta\theta$$

$\Rightarrow ?$

IDEA : USE A METRIC  $g_{ij}(\theta)$ :

$$\boxed{\delta\theta_i = \left(g^{-1}\right)_{il} \frac{\partial \mathcal{L}_\theta}{\partial \theta_l}}$$

IS REPARAMETERIZATION-INVARIANT

$$\delta\theta^t g \delta\theta = \delta\tilde{\theta}^t \tilde{g} \delta\tilde{\theta}$$

↪

$$g(\theta) = \left( \frac{\partial \tilde{\theta}}{\partial \theta} \right)^t \tilde{g}(\tilde{\theta}) \frac{\partial \tilde{\theta}}{\partial \theta}$$

$$\left( \frac{\partial \tilde{\theta}}{\partial \theta} \right)_{ij} = \frac{\partial \tilde{\theta}_i}{\partial \theta_j}$$

FIND, FROM  $\tilde{S}\tilde{\theta} = \tilde{g}^{-1} \frac{\partial \tilde{L}_{\tilde{\theta}}}{\partial \tilde{\theta}}$  :

$$S\theta = \frac{\partial \theta}{\partial \tilde{\theta}} \quad \tilde{S}\tilde{\theta} \quad \checkmark$$

EXPECTED FOR SWITCHING  
COORD. SYS.!

WHICH METRIC SHOULD WE USE?

LIKE BEFORE: " $y = F_\theta(x) + \text{NOISE}$ "

$$p(x, y | \theta) = p(y|x, \theta) \cdot \underbrace{p_x(x)}_{\text{ALL SAMPLES}}$$

$$\frac{\partial}{\partial \theta} \ln p(x, y | \theta) = \underbrace{\frac{\partial}{\partial \theta} \ln p(y|x, \theta)}_{-\mathcal{L}(y, F_\theta(x))}$$

SAMPLE-SPECIFIC  
LOSS FCT.

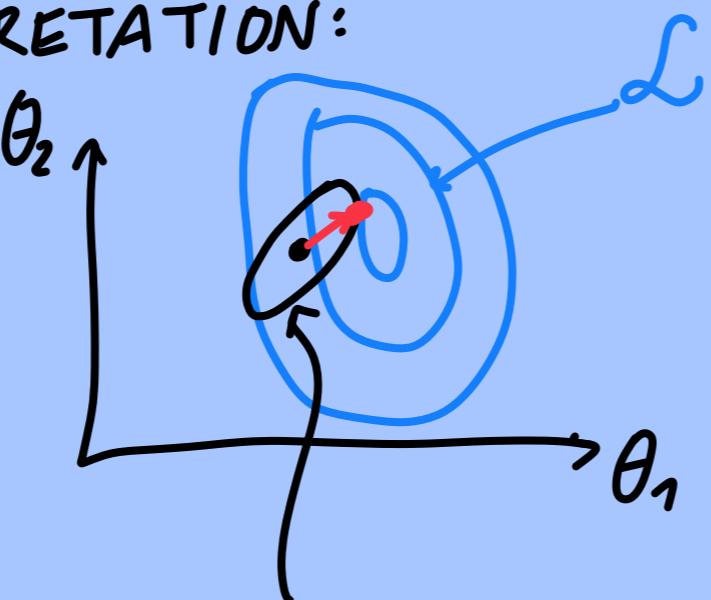
FISHER MATRIX

$$I_{\theta_i \theta_j} = \left\langle \frac{\partial \mathcal{L}}{\partial \theta_i}, \frac{\partial \mathcal{L}}{\partial \theta_j} \right\rangle_{\substack{x \sim p_x(x) \\ y \sim p(y|x, \theta)}}$$

$\Rightarrow$  "NATURAL GRADIENT"

$$I^{-1} \frac{\partial L}{\partial \theta}$$

INTERPRETATION:



ALL POINTS WITH

$$\nabla \theta^T I \nabla \theta = \text{const (small)}$$

$\Rightarrow$  CHOOSE POINT  
WITH LARGEST  
CHANGE OF  $L$  !

$$\text{FOR } \mathcal{L}(y, F_{\theta}(x)) = \frac{1}{2G^2} \|y - F_{\theta}(x)\|^2$$

WE HAVE:

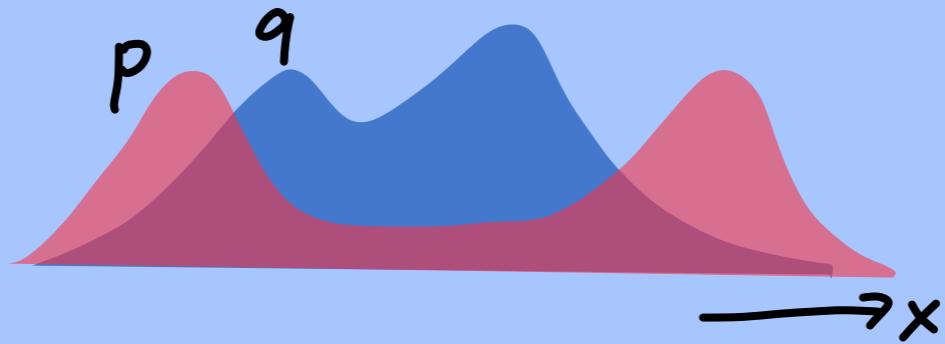
$$\frac{\partial \mathcal{L}}{\partial \theta_j} = -\frac{1}{G^2} \underbrace{(y - F_{\theta}(x))}_{\delta y} \frac{\partial F_{\theta}(x)}{\partial \theta_j}$$

$$\begin{aligned} I_{\theta j} &= \left\langle \frac{\partial \mathcal{L}}{\partial \theta_\ell}, \frac{\partial \mathcal{L}}{\partial \theta_j} \right\rangle \\ &= \frac{1}{G^4} \sum_{n,n'} \underbrace{\left\langle \delta y_n \frac{\partial (F_{\theta})_n}{\partial \theta_j}, \delta y_{n'} \frac{\partial (F_{\theta})_{n'}}{\partial \theta_\ell} \right\rangle}_{\langle \delta y_n \delta y_{n'} \rangle \langle \dots \rangle} \\ &\quad G^2 \delta_{n,n'} J_{nj} = \frac{\partial (F_{\theta})_n}{\partial \theta_j} \end{aligned}$$

$$I_{ej} = \frac{1}{6^2} \left( J^t J \right)_{j\ell}$$

6.3

## KULLBACK-LEIBLER DIVERGENCE (RELATIVE ENTROPY)



$$\mathcal{D}_{KL}(P \parallel Q) = \sum_y P_y \log \frac{P_y}{Q_y}$$

"DIVERGENCE  
OF P FROM Q"

OR:

$$\mathcal{D}_{KL}(P \parallel Q) = \int p(x) \log \frac{p(x)}{q(x)} dx$$

# PROPERTIES:

- $D_{KL}(P||Q) \geq 0$  ("=" FOR  $P=Q$ )

- MINIMIZATION:  $\min_Q D_{KL} = ?$

$$Q = \begin{pmatrix} Q_1 \\ \vdots \\ Q_N \end{pmatrix} \text{ WITH } \sum_j Q_j = 1$$

LAGRANGE MULTIPLIER

$$0 = \frac{\partial [D_{KL} + \lambda \sum_j Q_j]}{\partial Q_j} = -\frac{P_j}{Q_j} + \lambda = 0$$

$$Q_j = \frac{1}{\lambda} P_j$$

→ CHOOSE  
 $\lambda = 1$

$$D_{KL} = \sum_j P_j \log \frac{P_j}{Q_j}$$

$$-\frac{\partial}{\partial Q_j} \log Q_j = -\frac{1}{Q_j}$$

- FOR DENSITIES  $p(x)$ ,  $q(x)$ :

$$x = f(y)$$

BIJECTIVE

$D_{KL}$  IS INVARIANT UNDER THIS!

$$\left| \det \frac{\partial y}{\partial x} \right| \text{ CANCELS INSIDE}$$
$$\log \frac{p_x(x)}{q_x(x)}$$
$$= \log \frac{p_y(y)}{q_y(y)}$$

- GAUSSIAN EXAMPLE:

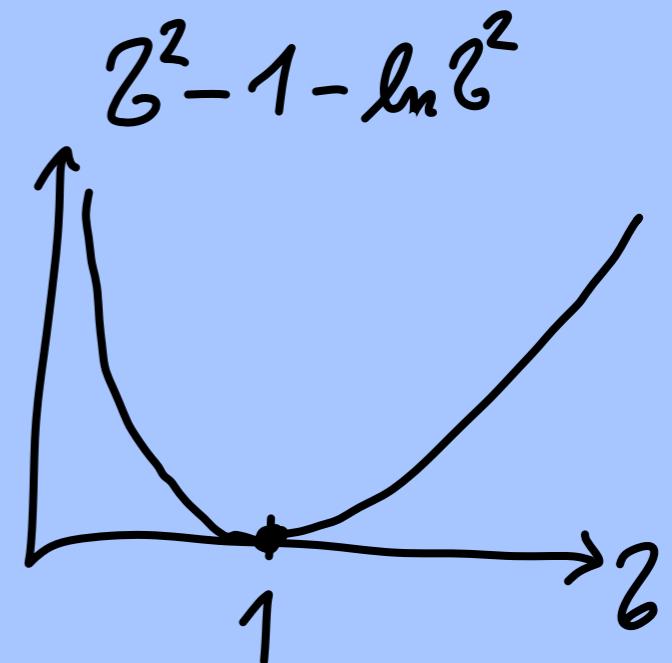
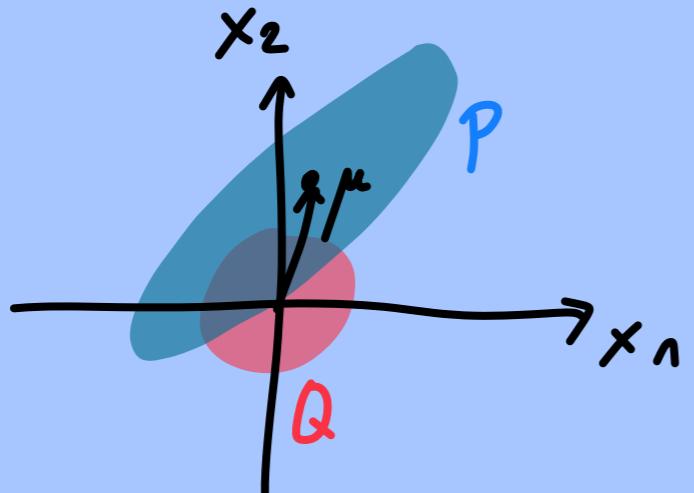
$$D_{KL}(P \parallel Q) = \frac{1}{\ln(b)} \sum_{j=1}^d \frac{1}{2} \left( Z_j^2 + \mu_j^2 - 1 - \ln Z_j^2 \right)$$

MULTIVAR.  
GAUSSIAN

NORMAL  
GAUSS

$\mu = 0, C = 1$

EIGENVALUES OF Cov FOR P



- INFORMATION THEORY / CODING:

-  $-\log_2 Q_j$  = OPTIMAL NR. OF  
BITS IN CODE FOR  
SYMBOL  $j$ , GIVEN  $Q$   
AS DISTR.

$D_{KL}(P||Q)$  = EXPECTED EXTRA  
AVG. CODE LENGTH  
IF TRUE DISTR. IS  $P$

- BAYES:  $P = P(\lambda|y)$  POSTERIOR

$Q = P(\lambda)$  PRIOR

$D_{KL}(P(\lambda|y) \parallel P(\lambda)) \equiv$  "INFORMATION GAIN"  
FOR PARTICULAR  $y$

$$\langle D_{KL}(P(\lambda|y) \parallel P(\lambda)) \rangle_y = \int P(y) P(\lambda|y) \log \frac{P(\lambda|y)}{P(\lambda)} dy$$

$$= H(\lambda) - H(\lambda|y)$$

= EXPECTED INFORMATION GAIN

- PRODUCT DISTR.:

$$\begin{matrix} & \times & y \\ p_x(x) & & p_y(y) \end{matrix}$$

$$\begin{aligned} \mathcal{D}_{KL}(p_x(x)p_y(y) \parallel q_x(x)q_y(y)) \\ = \mathcal{D}_{KL}(p_x(x) \parallel q_x(x)) + \mathcal{D}_{KL}(p_y(y) \parallel q_y(y)) \end{aligned}$$

-  $\mathcal{D}_{KL}$  IS NOT A DISTANCE MEASURE

( NOT SYMMETRIC, NO TRIANGLE INEQ. )

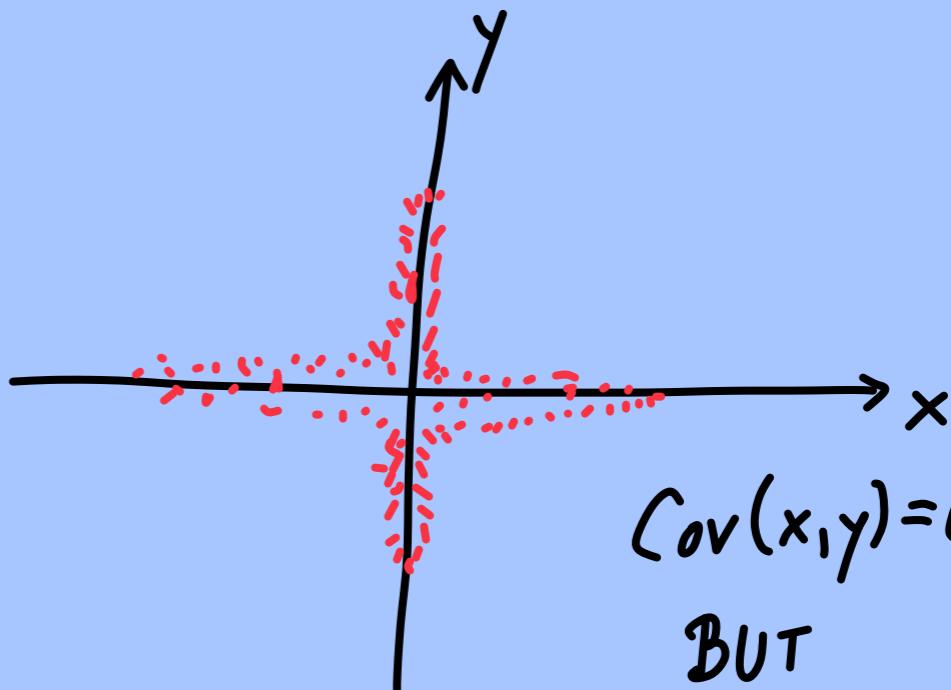
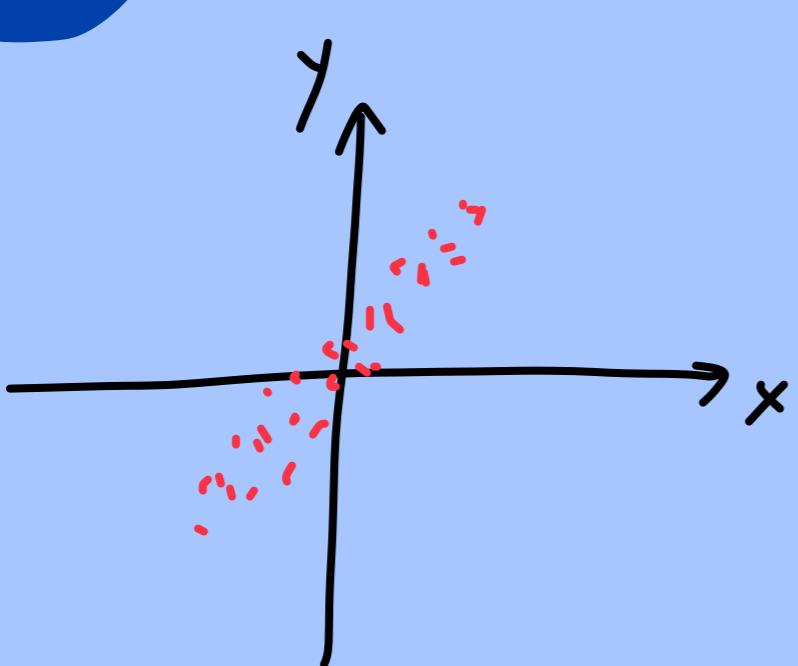
CONNECTION TO FISHER INFO:

$$\mathcal{D}_{KL}(p(x|\lambda) \parallel p(x|\lambda_0)) = \frac{1}{2} S\lambda^+ \mathbb{I} S\lambda + O(S\lambda^3)$$

$$S\lambda = \lambda - \lambda_0$$

6.4

## MUTUAL INFORMATION



$\text{Cov}(x, y) = 0$   
BUT  
 $x, y$  ARE  
DEPENDENT!

ANALYZE VIA  
 $\text{Cov}(x, y) = \langle (x - \bar{x})(y - \bar{y}) \rangle$

IDEA: COMPARE  $P(X=x, Y=y)$   
AGAINST  $P_x(X=x) \cdot P_y(Y=y)$

HOW TO  
QUANTIFY?

WHERE MARGINAL DISTRIBUTION:

$$P_x(X=x) = \sum_y P(X=x, Y=y)$$

$$MI(X,Y) \equiv D_{KL}(P(X,Y) \parallel P_x(X) \cdot P_y(Y))$$

$$= \sum_{x,y} P(X=x, Y=y) \log \frac{P(X=x, Y=y)}{P_x(X=x) \cdot P_y(Y=y)}$$

$$= H(X) + H(Y) - H(X,Y) \geq 0$$

REMEMBER  $H(X,Y) \leq H(X) + H(Y)$

$$\left[ \frac{P(X=x, Y=y)}{P_y(Y=y)} = P(X=x|Y=y) \right]$$

$$= \sum_{x,y} P(x,y) \log \frac{P(x|y)}{P_x(x)}$$

$$= H(X) - H(X|Y)$$

$$= H(Y) - H(Y|X)$$

= EXPECTED INFORMATION GAIN

$$= \langle D_{KL}(P(X|Y=y) \parallel P_x(X)) \rangle_Y$$

$$MI \geq 0$$

MI ALSO WORKS FOR CONTINUOUS  $x, y$ :

$$MI = \int dx dy P(x, y) \log \frac{P(x, y)}{P_x(x) \cdot P_y(y)}$$

IS REPARAMETERIZATION-INVARIANT

$$\tilde{x} = f(x) \quad \tilde{y} = g(y) \Rightarrow MI \text{ UNCHANGED}$$

BIJECTIVE

IF WE CAN CHOOSE WHICH  $y_f$  TO MEASURE:

$$\text{CHOOSE } MI(x, y_f) \stackrel{!}{=} \max_f$$



6.5

## EXCURSION: RENORMALIZED MUTUAL INFORMATION

GOAL : FEATURE EXTRACTION/ REPR. LEARNING

$$Y = F(X)$$

|  
HIGH-DIM.  
LOW-DIM.

DETERMINISTIC!

CHOOSE  $F$  TO MAXIMIZE  $MI(X, Y)$

PROBLEM FOR CONTINUOUS  $X, Y$ :

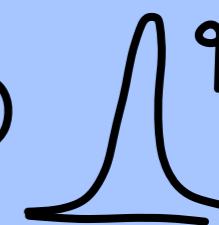
$$MI(X, Y) = \infty !$$

$$P(Y=y | X=x) = \delta(y - F(x))$$

$$\int \underbrace{\delta(y - F(x))}_{\sim \delta(0) = \infty} \log \underbrace{\delta(y - F(x))}_{\sim \delta(0) = \infty} dy$$

OLD SOLUTION  
("INFOMAX"  
BELL & SEJNOWSKI  
'95)

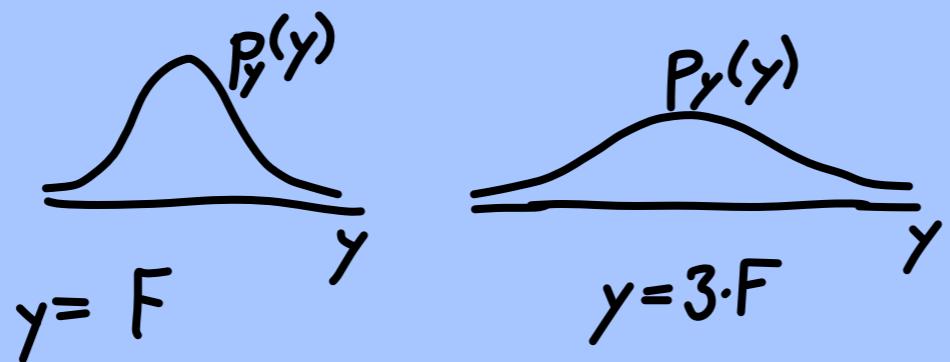
IMAGINE

$$y = F(x) + \text{NOISE}$$
$$\mathcal{S}(y - F(x)) \mapsto q(y - F(x))$$


$\Rightarrow$  MI FINITE (LATER: NOISE  $\rightarrow 0$ )

$$\text{MI} = \underbrace{\text{FINITE}}_{H(y)} + \text{NOISE REL.}$$

$\Rightarrow$  MAXIMIZE  $H(y)$ !



$\Rightarrow$  NOT REPARAMETERIZATION -  
INVARIANT!

CAN WE

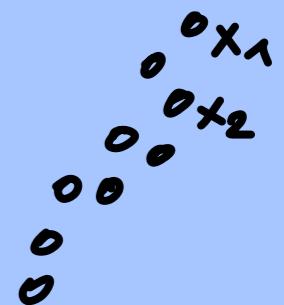
- AVOID  $\infty$
- KEEP REPARAMETERIZATION INVARIANCE FOR  $y^2$ ?

SOLUTION: RENORMALIZED MI ("RMI")  
(SARRA, AIELLO, F.M. 2021)

- IMAGINE

$$y = F(x + \text{NOISE})$$

RESOLUTION/  
UNCERTAINTY



- CALCULATE MI( $x, y$ )
- SUBTRACT PART THAT ONLY DEPENDS ON NOISE
- LIMIT NOISE  $\rightarrow 0$   
(ASSUME e.g. i.i.d. NOISE)

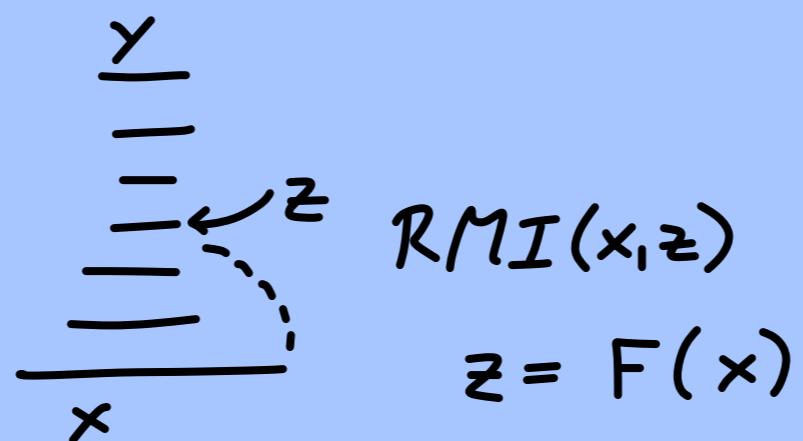
$$RMI(x,y) = H(y) - \int dx P_x(x) \ln \sqrt{\det \left( \frac{\partial F}{\partial x} \right) \left( \frac{\partial F}{\partial x} \right)^t}$$

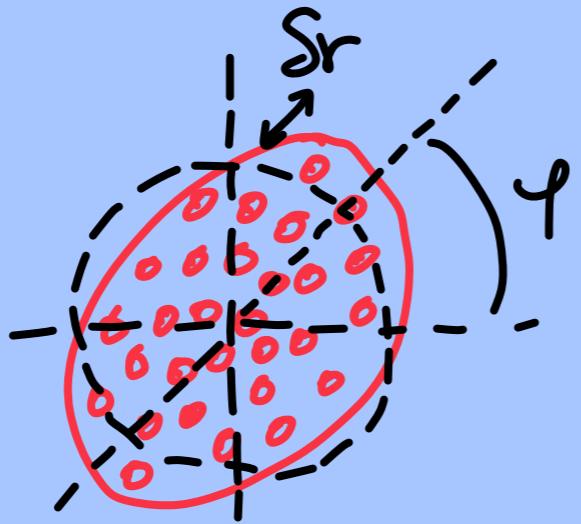
$$\left[ \left( \frac{\partial F}{\partial x} \right) \left( \frac{\partial F}{\partial x} \right)^t \right]_{\mu\nu} = \sum_j \frac{\partial F_{\mu}}{\partial x_j} \frac{\partial F_{\nu}}{\partial x_j}$$

$\Rightarrow$  REPARAM. INV. WITH RESPECT TO  $y$

$\rightarrow$  FEATURE EXTR.  
(COLLECTIVE COORDS)

$\rightarrow$  ANALYZE NN INFO FLOW





CHALLENGE:  $H(y)$  ESTIMATE  
FOR HIGH-DIM.  $y$ !

7.

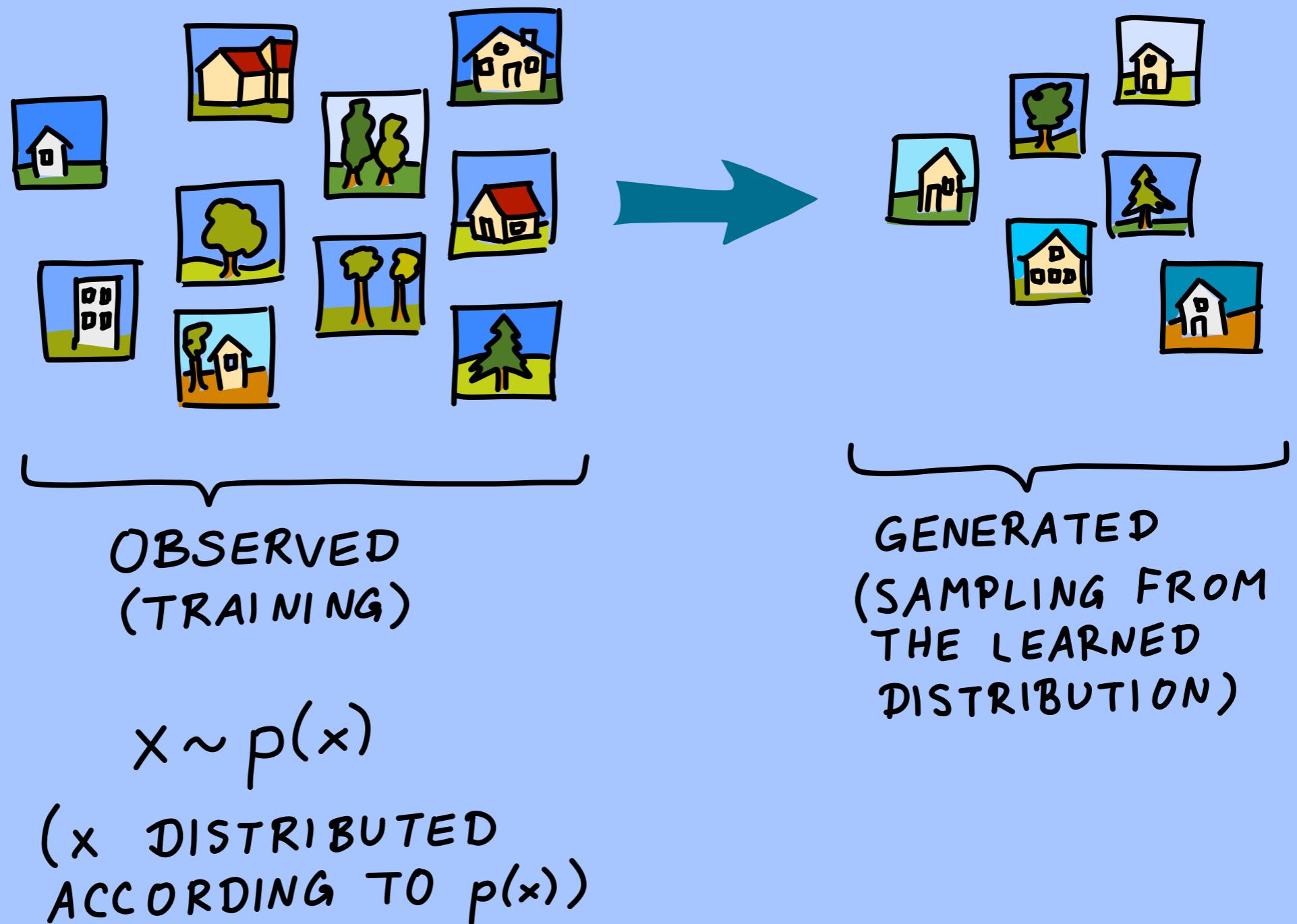
# LEARNING PROBABILITY DISTRIBUTIONS

"LEARNING FROM SAMPLES"

"LEARNING TO SAMPLE"

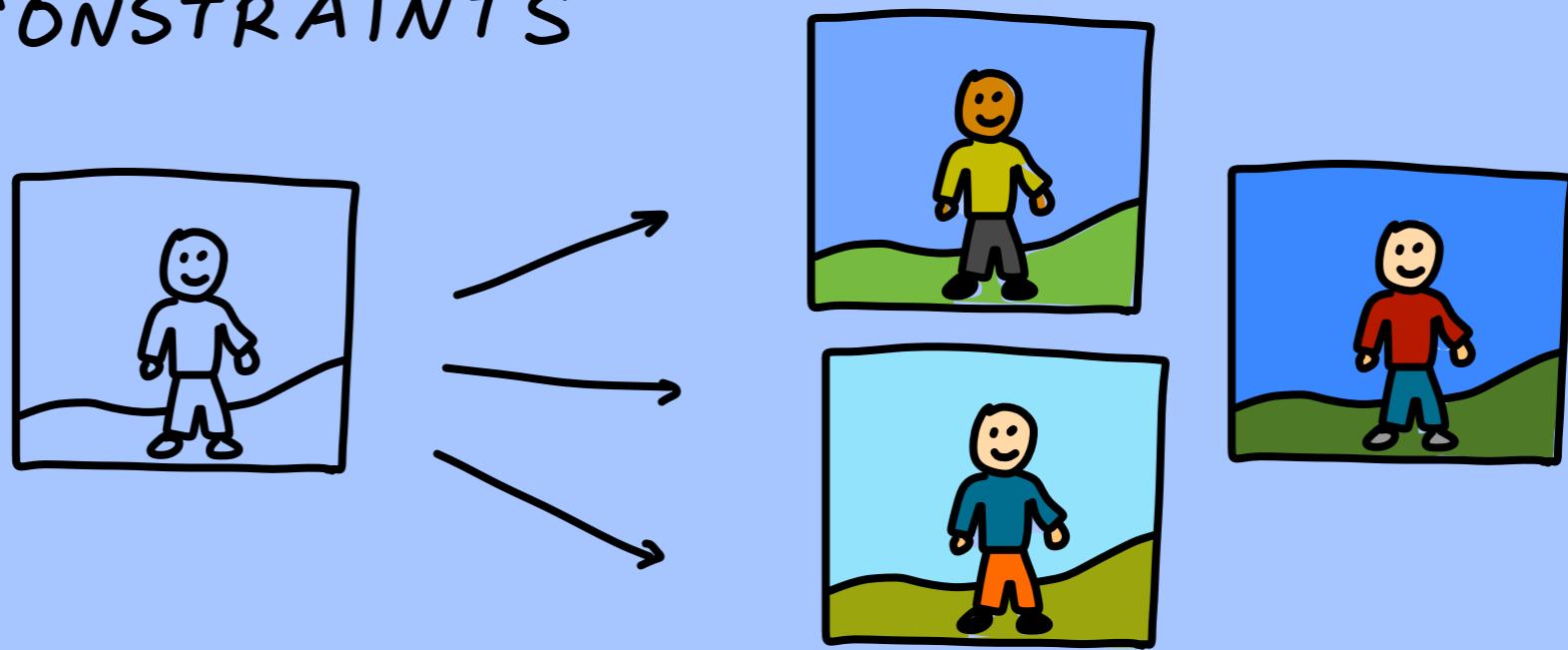
7.1

# MOTIVATION & GOALS

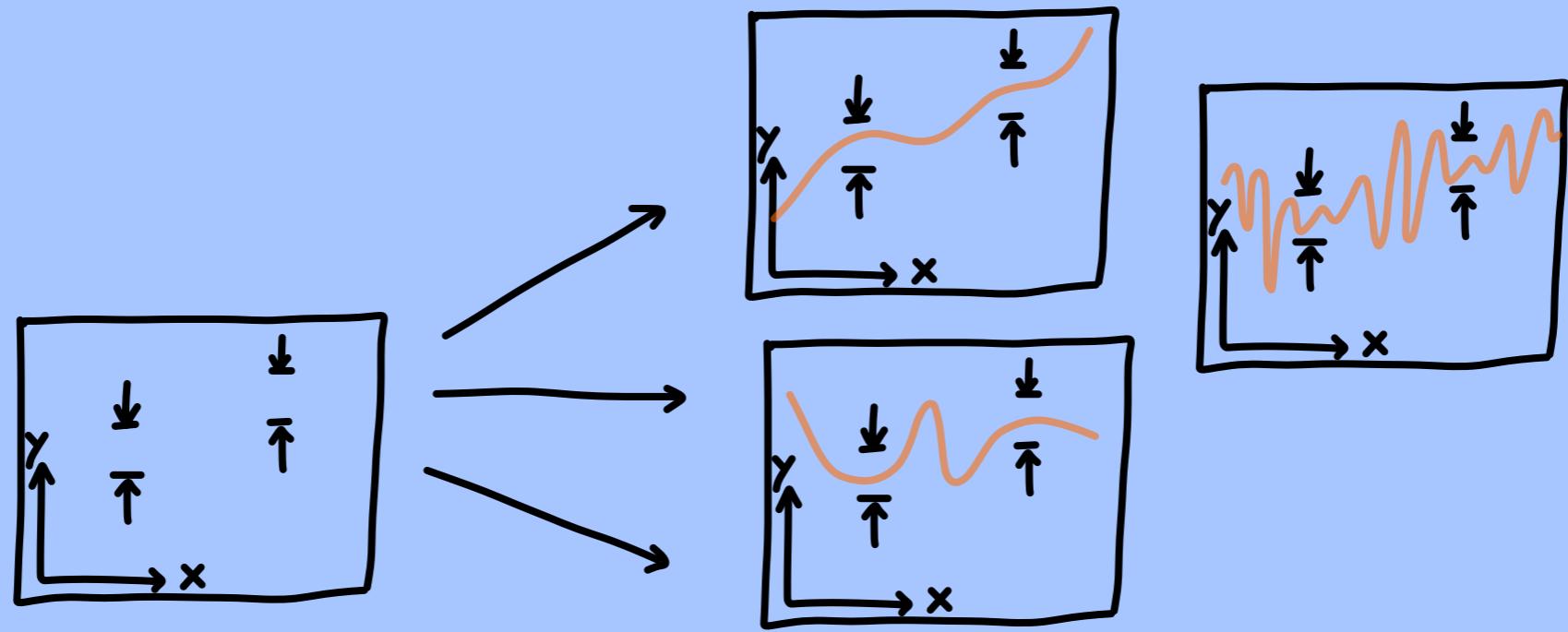


# APPLICATIONS

- PRODUCE SAMPLES COMPATIBLE WITH CONSTRAINTS

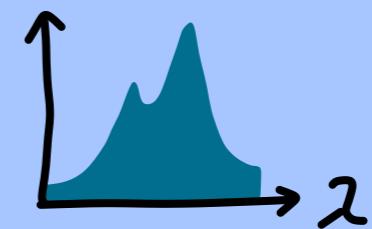


~ "ASSOCIATIVE MEMORY"

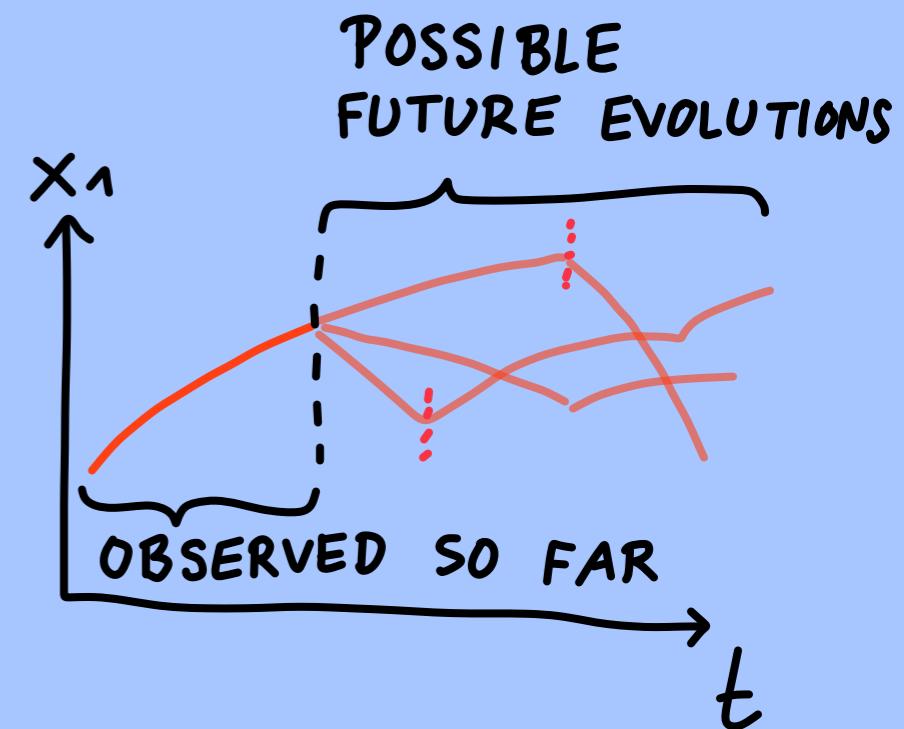
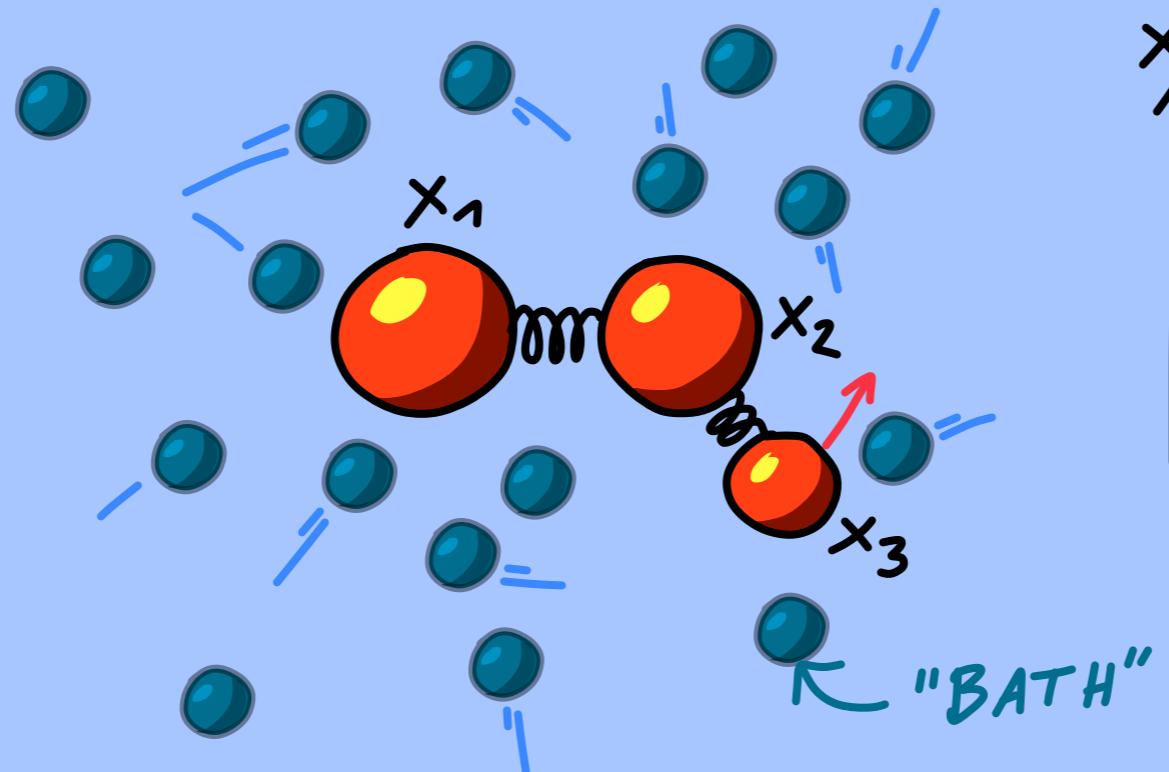


■ BAYES  $p(\lambda|y)$

(→ OPTIMAL BAYESIAN EXPERIMENTAL DESIGN,  
ACTIVE LEARNING)



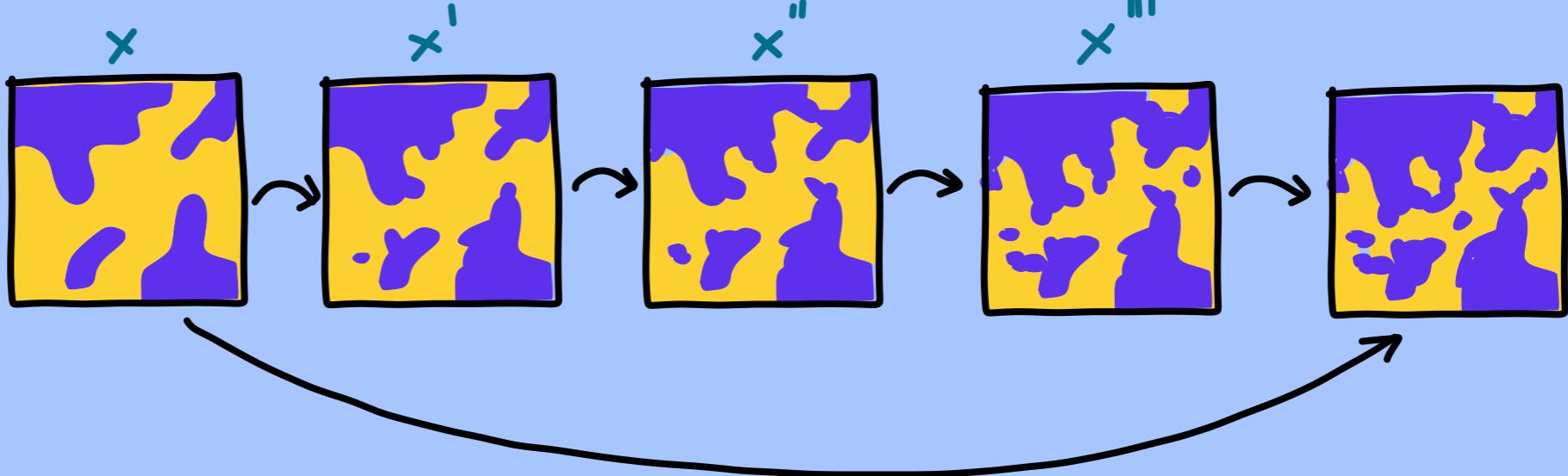
## ■ LEARN STOCHASTIC EQUATIONS OF MOTION



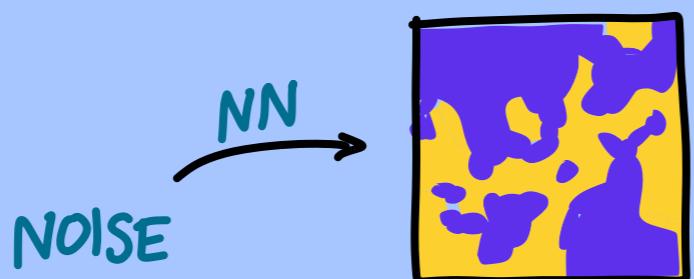
FULL SYSTEM : DETERMINISTIC

ONLY RED PARTICLES: NONLINEAR  
STOCHASTIC  
NON-MARKOVIAN (MEMORY)  
DYNAMICS

## ■ ACCELERATE MONTE CARLO

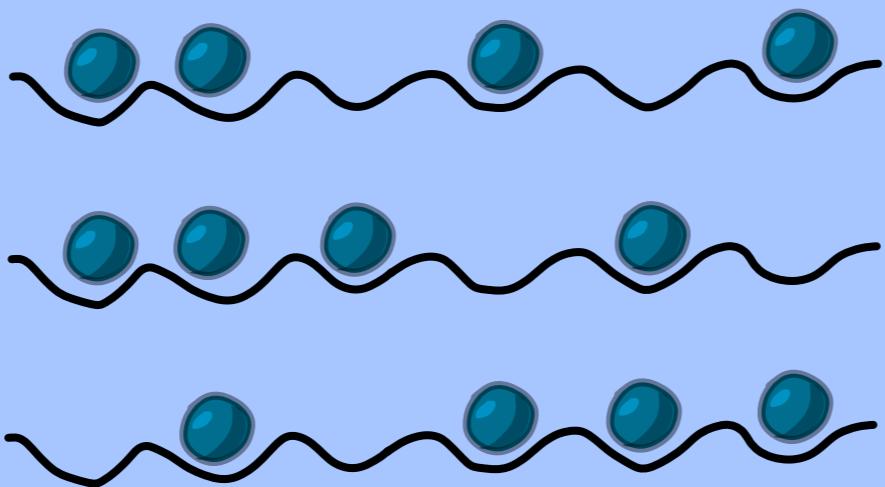


## ■ REPLACE MONTE CARLO



LEARN TO SAMPLE  
FOR AN EXPLICITLY  
GIVEN  
FORM OF  $p(x)$

## ■ LEARN QUANTUM STATE FROM OBSERVATIONS



DIFFERENT SNAPSHOTS  
OF A QUANTUM MANY-BODY  
SYSTEM IN A GIVEN STATE  $|\psi\rangle$

$$P(\hat{A}=\alpha | \psi) = \underbrace{|\langle \alpha | \psi \rangle|^2}_{\text{MEASUREMENT BASIS}}$$

EASY:  $P_{j \sim}$   $\xrightarrow{\text{FEW DISCRETE EVENTS}}$

$$P(j|z) = [F_\theta(z)]_j$$

$j = 1 \ 2 \ 3 \ 4$   
  
 $NN$

- E.G.
- LABELS
  - PROB. OF NEXT ITEM IN SEQUENCE

THE PH?

HARD:

$P_j \xrightarrow{\text{MANY DISCR.}}$

COMBINATORIALLY LARGE

EXAMPLE:  
B/W IMAGES  
SPIN CONFIGUR.

OR

$p(x) \xrightarrow{\text{CONTINUOUS}}$

(CHALLENGE EVEN FOR  $d \geq 3$ )

→ ALWAYS RELY ON SAMPLING

GOALS:

GIVEN SAMPLES, LEARN

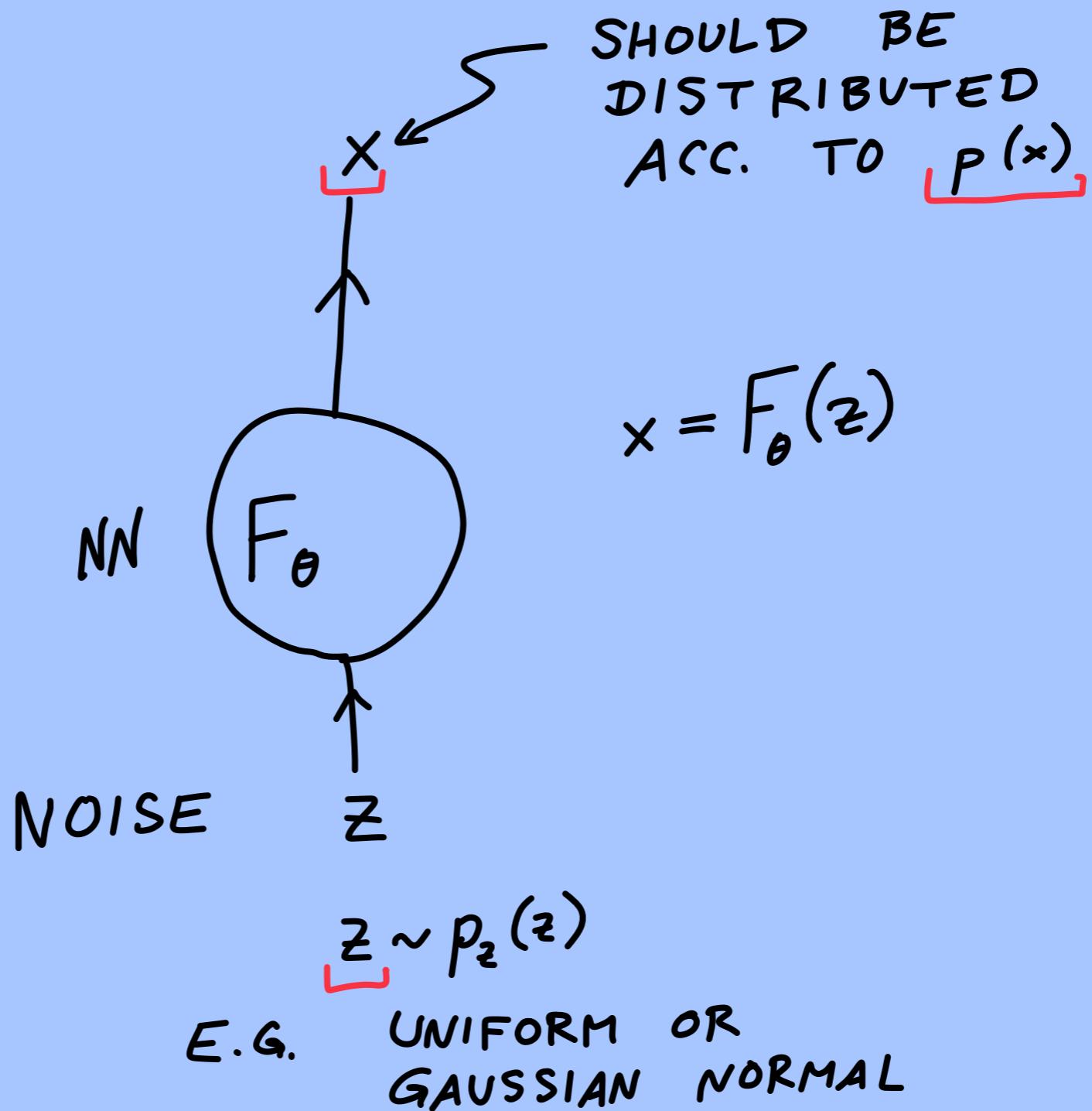
TO SAMPLE FROM  $p$

TO PRODUCE  
REPRESENTATION  
OF  $p(x)$

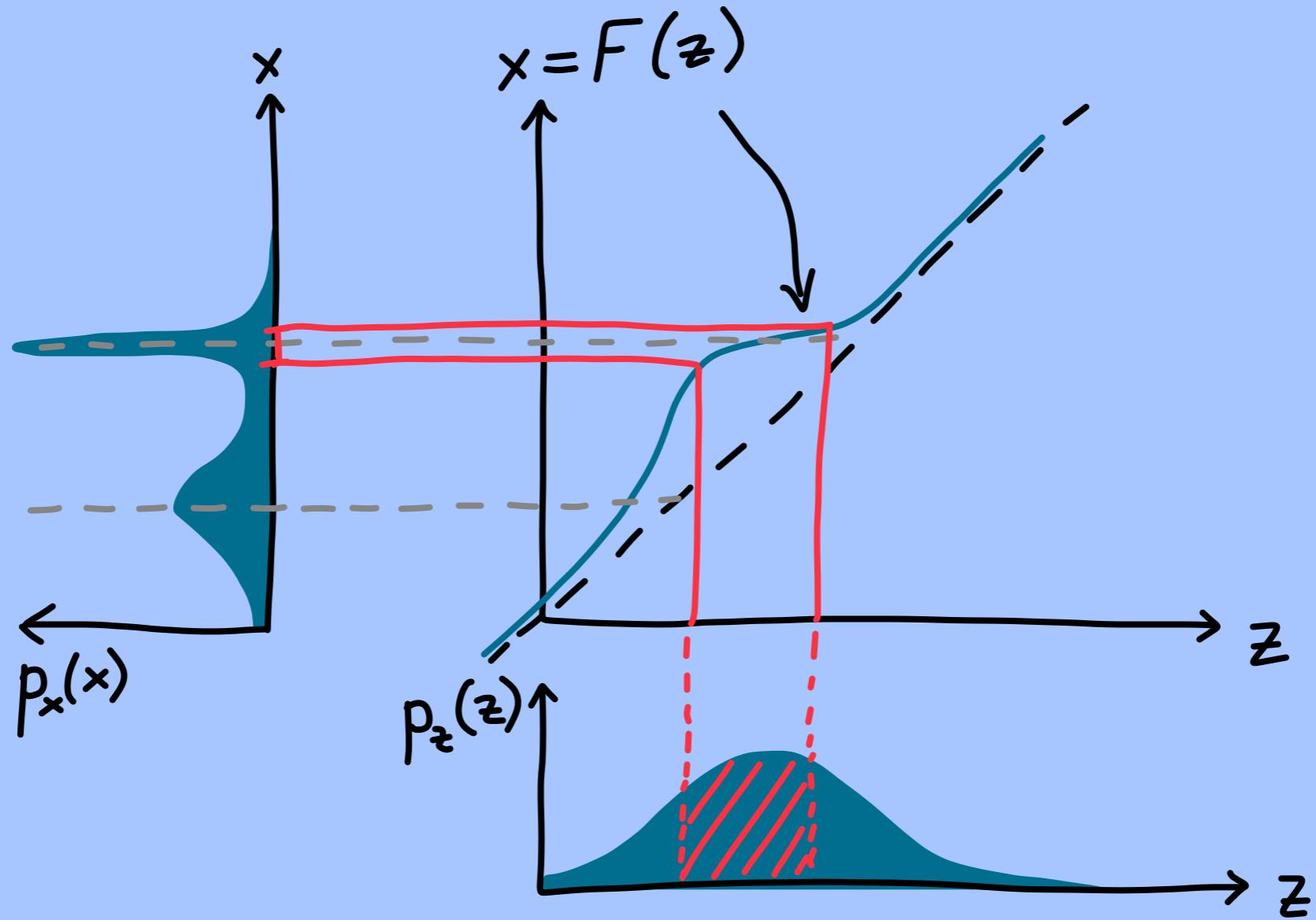
GIVEN  $p(x)$ , LEARN TO SAMPLE  
EFFICIENTLY

7.2

## NORMALIZING FLOWS: INVERTIBLE NEURAL NETWORKS



# REMINDER: TRANSFORMING PROBABILITY DENSITIES



$$1D: |p_x(x) dx| = |p_z(z) dz|$$

$$p_x(x) = p_z(z(x)) \left| \frac{\partial z(x)}{\partial x} \right| = \frac{p_z(z(x))}{\left| \frac{\partial x}{\partial z}(z(x)) \right|}$$

$$z(x) = F^{-1}(x)$$

VECTORS

$$x, z \in \mathbb{R}^D$$

$$\Rightarrow p_x(x) = p_z(z) \cdot \left| \det \underbrace{\frac{\partial z}{\partial x}}_{\text{JACOBIAN}} \right|$$
$$= \frac{p_z(z)}{\left| \det \frac{\partial z}{\partial x} \right|}$$

$\Rightarrow$  NEED TO:

- INVERT  $F$ :  $z = F^{-1}(x)$
- CALCULATE  $\left| \det \frac{\partial z}{\partial x} \right|$

$$z \xrightarrow{\quad} x = F_{\theta}(z)$$

$(z \sim p_z(z))$

$x \sim q_{\theta}(x)$

WANT  $q_{\theta}^{(x)} \approx p(x)$  "TARGET  
DENSITY"

$\Rightarrow$

$$\begin{aligned} \text{MINIMIZE } D_{KL}(p \parallel q_{\theta}) &= \int p(x) \log \frac{p(x)}{q_{\theta}(x)} dx \\ &= \left\langle \log \frac{p(x)}{q_{\theta}(x)} \right\rangle_{x \sim p(x)} \end{aligned}$$

EASY

OR: MAXIMIZE

$$\left\langle \log q_{\theta}(x) \right\rangle_{x \sim p(x)}$$

$\Rightarrow$  MAX. AVERAGE LOG-LIKELIHOOD OF  $x$  WITH REGARD TO  $q_{\theta}$

$$q_{\theta}(x) = P_z(z_{\theta}(x)) \left| \det \frac{\partial z_{\theta}(x)}{\partial x} \right|$$

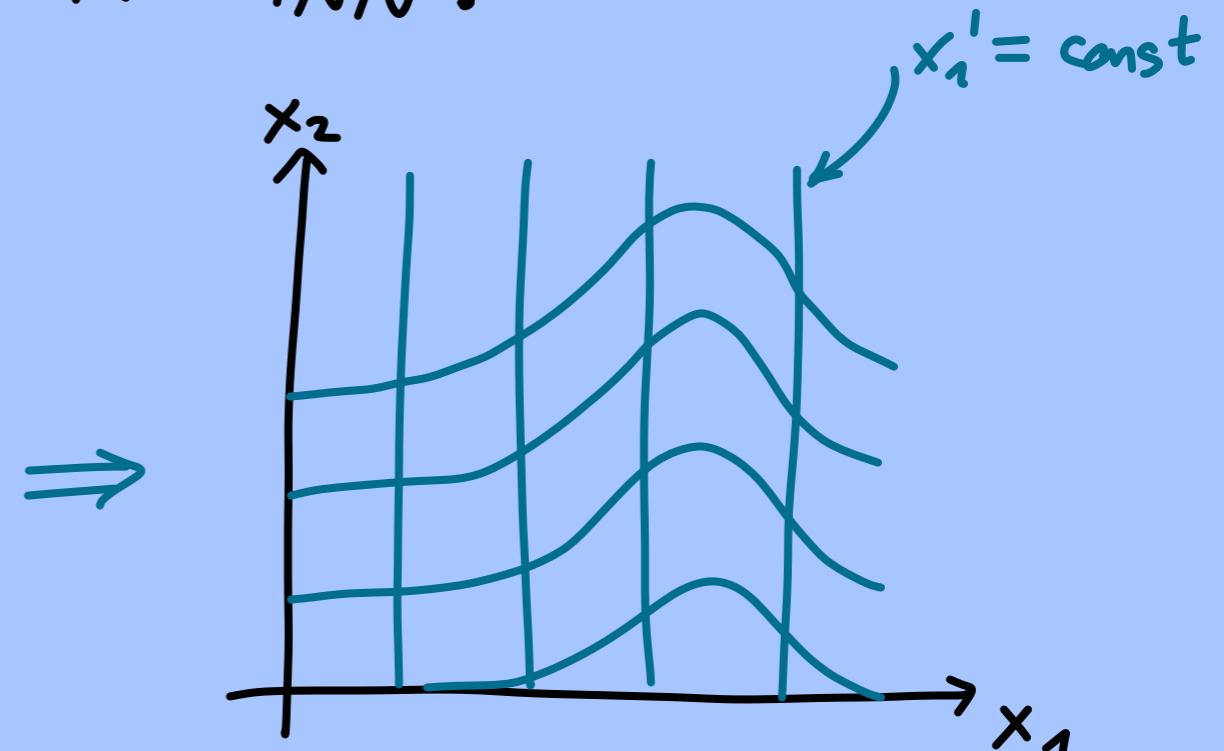
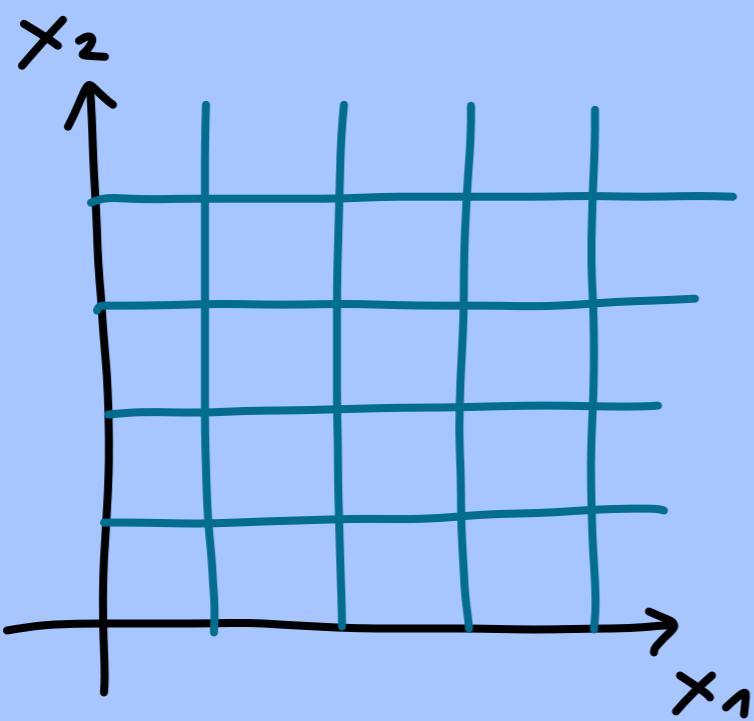
$$z_{\theta}(x) = F_{\theta}^{-1}(x)$$

WORKS IF  $z = z(x)$  IS INVERTIBLE!  
 $x = x(z)$

## ⇒ INVERTIBLE NEURAL NETWORKS

- NOTE:
- ARBITRARY NN NOT INVERTIBLE  
(DIFF'T NEURON #)
  - FOR SAME NEURON #: COULD BE INVERTIBLE

# BASIC IDEA FOR INN:

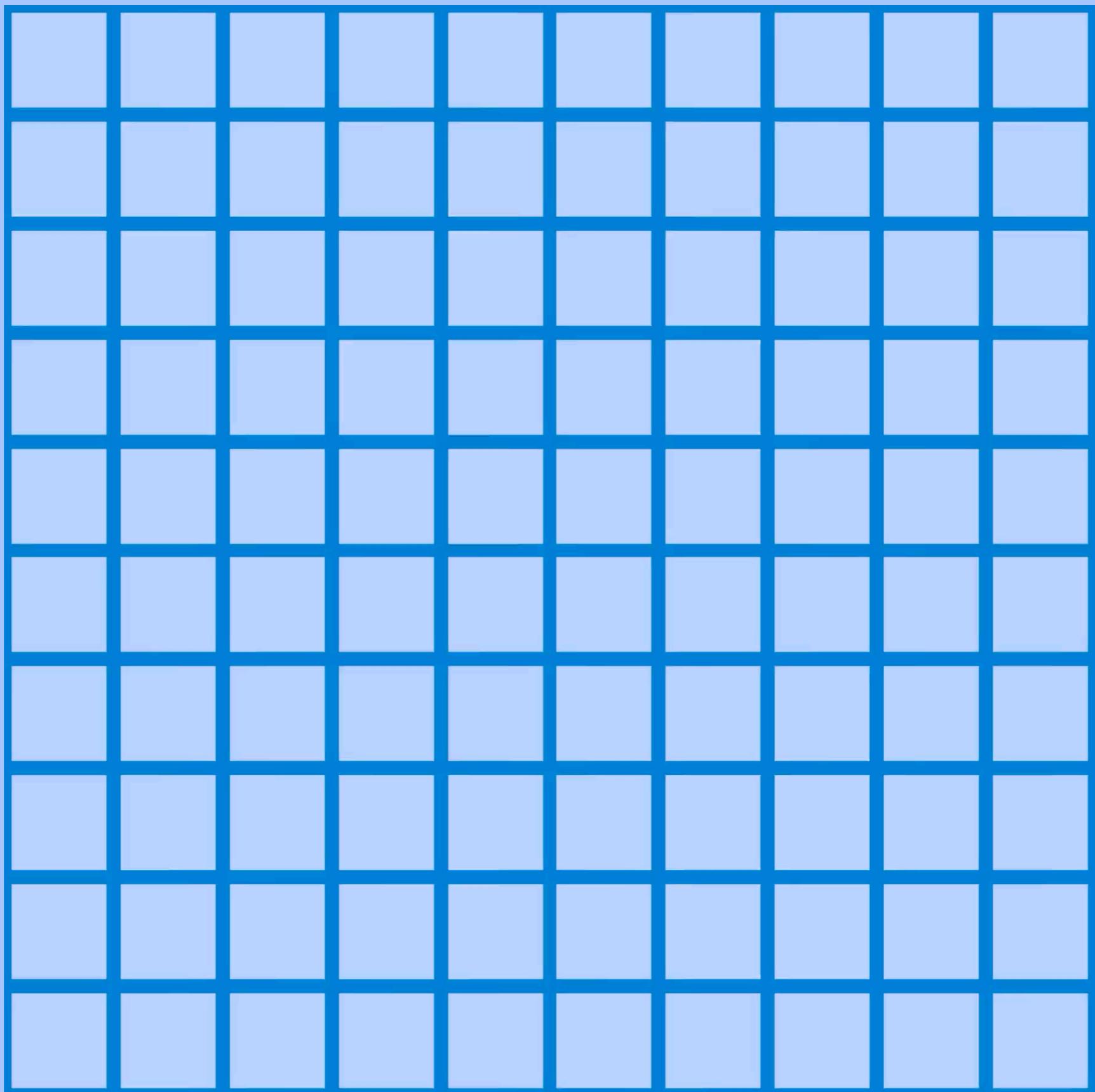


FORWARD

$$\left[ \begin{array}{l} x_1' = x_1 \\ x_2' = x_2 + \underline{\underline{m(x_1)}} \end{array} \right] \quad \left\{ \text{"COUPLING LAYER"} \right.$$

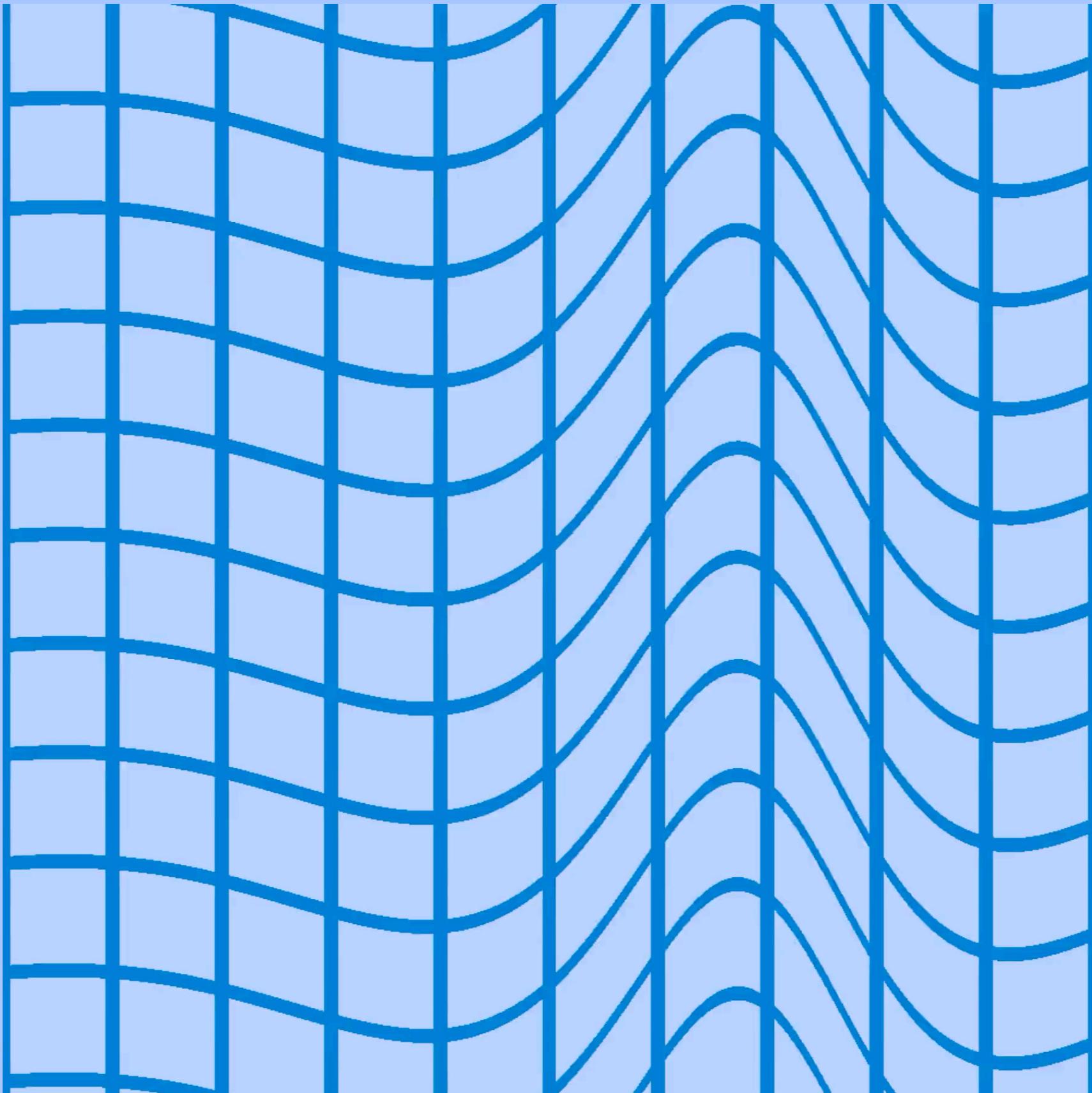
BACKWARD

$$\left[ \begin{array}{l} x_1 = x_1' \\ x_2 = x_2' - m(x_1') \end{array} \right]$$



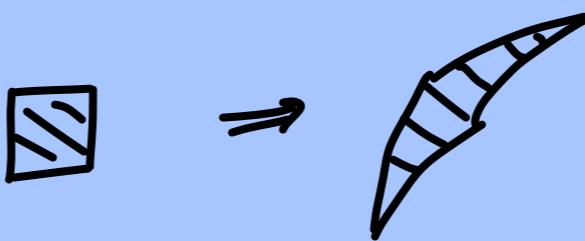
$$x_1'' = x_1' + \tilde{m}(x_2')$$

$$x_2'' = x_2'$$



$$\frac{\partial \mathbf{x}'}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial x'_1}{\partial x_1} \\ \frac{\partial x'_2}{\partial x_1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \frac{\partial m(x_1)}{\partial x_1} & 1 \end{bmatrix}$$

$$\det \frac{\partial \mathbf{x}'}{\partial \mathbf{x}} = 1$$



ALSO INVERTIBLE FOR

$$x_2' = g(x_2, m(x_1))$$

PROVIDED  $g(a, b)$  CAN  
BE INVERTED GIVEN  $b$   
IN  $a$

EXAMPLE:

ELEMENTWISE PRODUCT ELEMENTWISE

$$x_2' = \underbrace{\exp(s(x_1))}_{>0 \Rightarrow \text{INVERTIBLE}} \odot x_2 + t(x_1)$$
$$[\alpha \odot b]_j = \alpha_j \cdot b_j$$

"AFFINE COUPLING LAYER"

$$x_1' = x_1$$

ALSO POSSIBLE FOR

$$x_1 \in \mathbb{R}^{d_1}$$
$$x_2 \in \mathbb{R}^{d_2}$$

$$\frac{\partial \mathbf{x}'}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial x_1'}{\partial x_1} & | & \frac{\partial x_n'}{\partial x_2} \\ \hline \vdots & | & \vdots \\ \frac{\partial x_2'}{\partial x_n} & | & \frac{\partial x_n'}{\partial x_2} \end{bmatrix} \text{ MATRIX}$$

$$= \begin{bmatrix} 1 & | & 0 \\ \hline \vdots & | & \vdots \\ \frac{\partial t}{\partial x_1} & | & \begin{bmatrix} e^s \\ \vdots \\ e^s \end{bmatrix} \\ \hline 0 & | & \begin{bmatrix} e^s \\ \vdots \\ e^s \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 \\ \hline \vdots \\ \hline \end{bmatrix}$$

$d_2 \times d_2$  - MATRIX

$$\det \begin{bmatrix} \hline \vdots \\ \hline \end{bmatrix} = \prod_j [e^{s(x_j)}]_j = e^{\sum_j [s(x_j)]_j}$$

INVERSION:

$$x_2 = e^{-s(x_n)} \odot (x_1' - t(x_n'))$$

$$x_1 = x_1'$$

FOR NN:

$$\left. \begin{array}{l} s(x_n) \\ t_\theta(x_n') \end{array} \right\} \begin{array}{l} \text{DIFFERENT} \\ \text{s,t NETWORKS} \\ \text{FOR EACH} \\ \text{COUPLING LAYER} \end{array}$$

GENERAL SITUATION:

$x \in \mathbb{R}^D \rightarrow$  HOW TO APPLY  
COUPLING LAYER?

TWO OPTIONS:

1. 'SPLITTING':

SELECT SUBSETS OF  
COORD's IN  $x$  TO  
FORM  $x_1, x_2$

$$x = (\overbrace{\quad}^{\equiv}, \overbrace{\quad}^{\equiv}, \overbrace{\quad}^{\equiv}, \overbrace{\quad}^{\equiv}, \overbrace{\quad}^{\equiv}, \overbrace{\quad}^{\equiv}, \overbrace{\quad}^{\equiv})$$
$$x_1$$
$$x_2$$

2. RANDOM INVERTIBLE LINEAR TRAFO:

$$\tilde{x} = Sx$$

$\hookrightarrow \det S \neq 0$  (e.g. ROTATION MATRIX  
 $\det S = 1$ )

$x_1, x_2$  = (FIXED) COMPONENTS  
OF  $\tilde{x}$

# IMPLEMENTATION

$z \rightarrow x$   
 FORWARD PASS  
 = SEQUENCE OF  
 COUPLING LAYERS

$x \rightarrow z$   
 BACKWARD PASS

TRACK JACOBIAN:

$$\ln \left| \det \frac{\partial z}{\partial x} \right|$$

$$= \ln \left| \det \frac{\partial z}{\partial y_N} \cdot \frac{\partial y_N}{\partial y_{N-1}} \cdot \dots \cdot \frac{\partial y_1}{\partial x} \right|$$

$$= \ln \left| \prod_j \det \frac{\partial y_j}{\partial y_{j-1}} \right|$$

$$= \sum_j \underbrace{\ln \left| \det \frac{\partial y_j}{\partial y_{j-1}} \right|}_{\text{ONE TERM FOR EACH COUPLING}}$$

$$\begin{aligned} y_0 &= x \\ y_{N+1} &= z \end{aligned}$$



# ALGORITHM

① SAMPLE  $x$  FROM  $p(x)$  (BATCH)

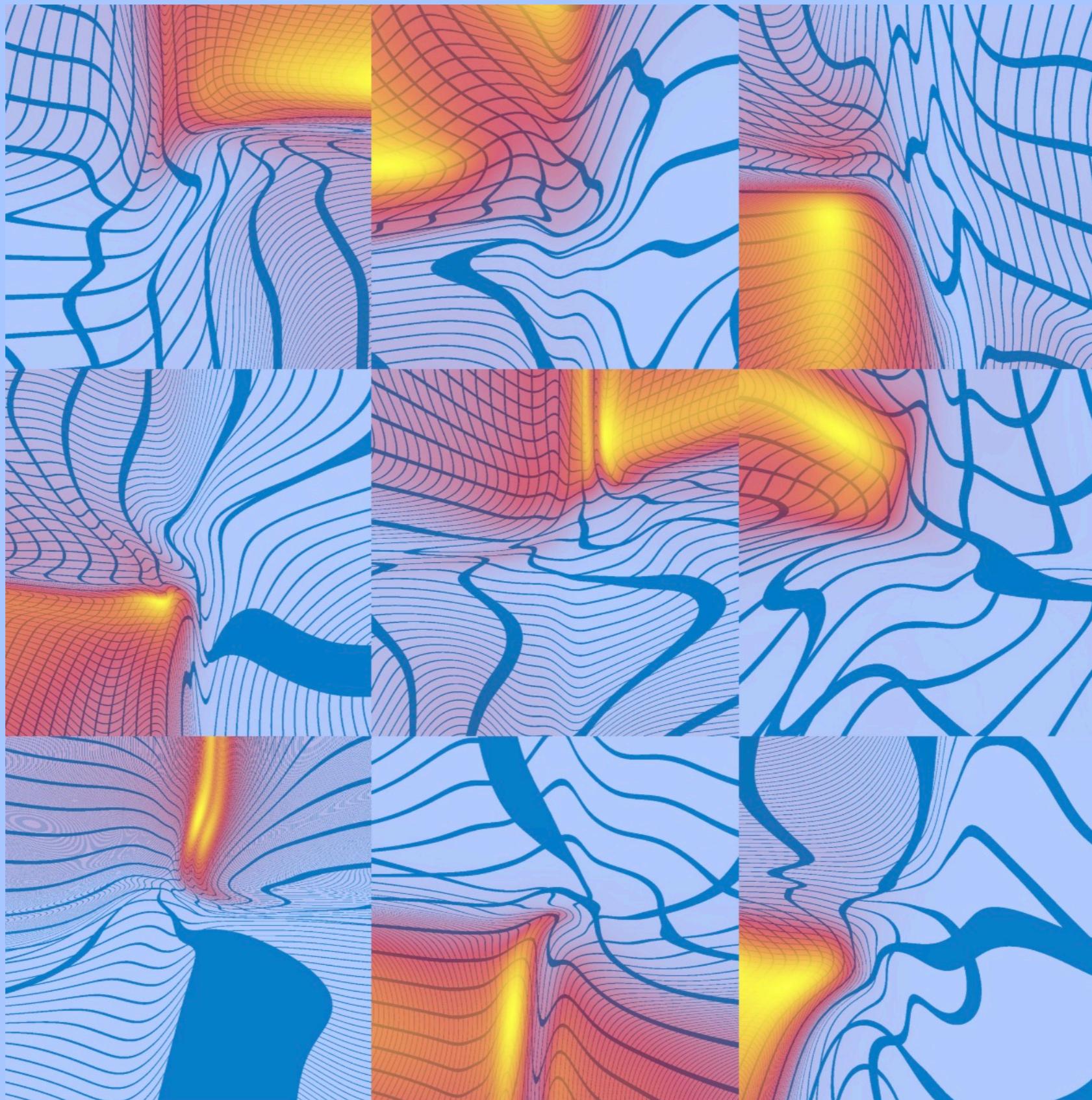
② CALCULATE  $z(x)$   
&  $\ln \left| \det \frac{\partial z(x)}{\partial x} \right|$

③ MAXIMIZE  $\underline{\underline{\langle \ln q_{\theta}(x) \rangle_{x \sim p(x)}}}$

BY:

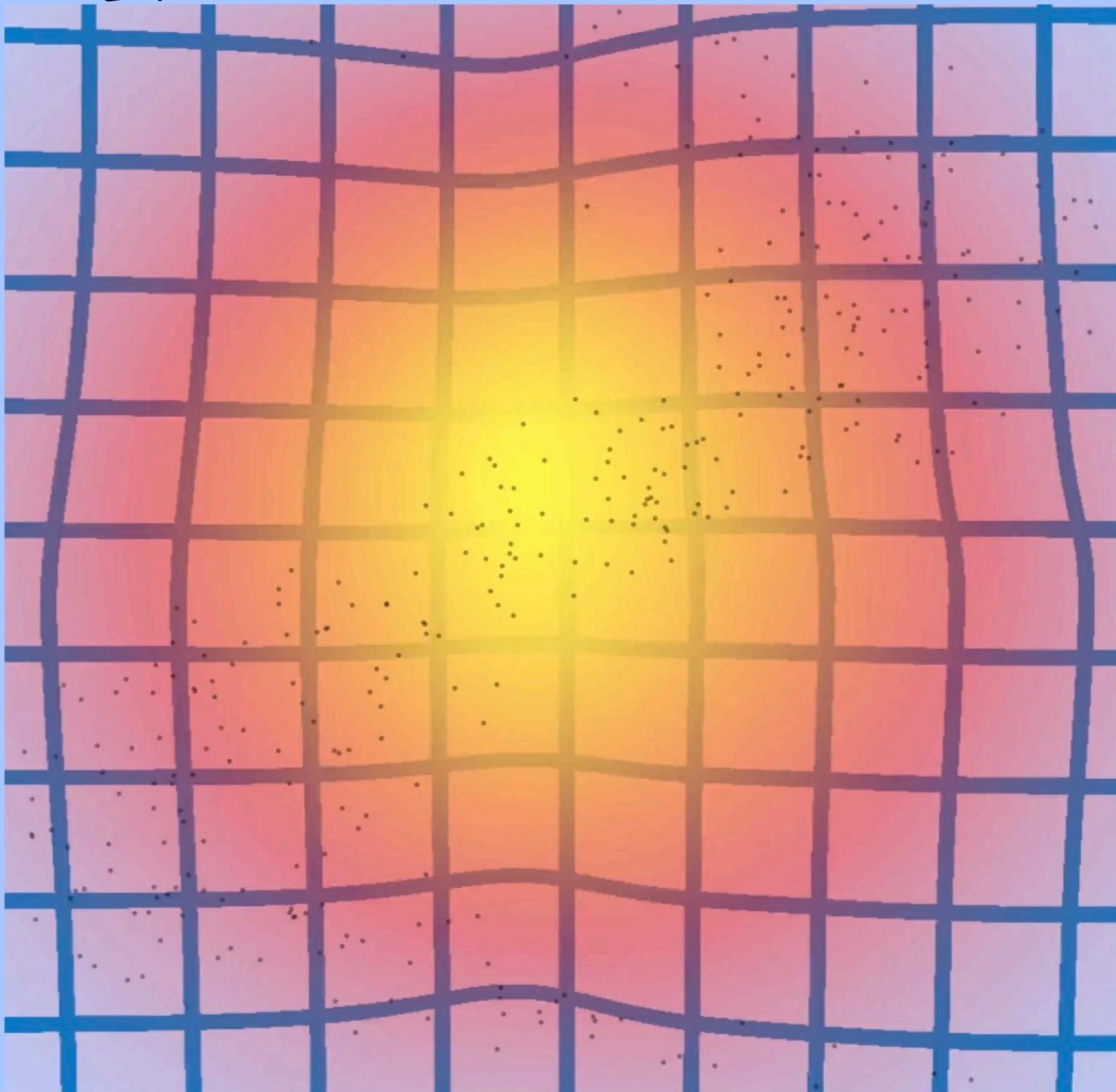
$$S\theta = \eta \frac{\partial}{\partial \theta} \left\{ \ln P_z(z_{\theta}(x)) + \ln \left| \det \frac{\partial z_{\theta}(x)}{\partial x} \right| \right\}$$

# RANDOMLY INITIALIZED INVERTIBLE NEURAL NETWORKS

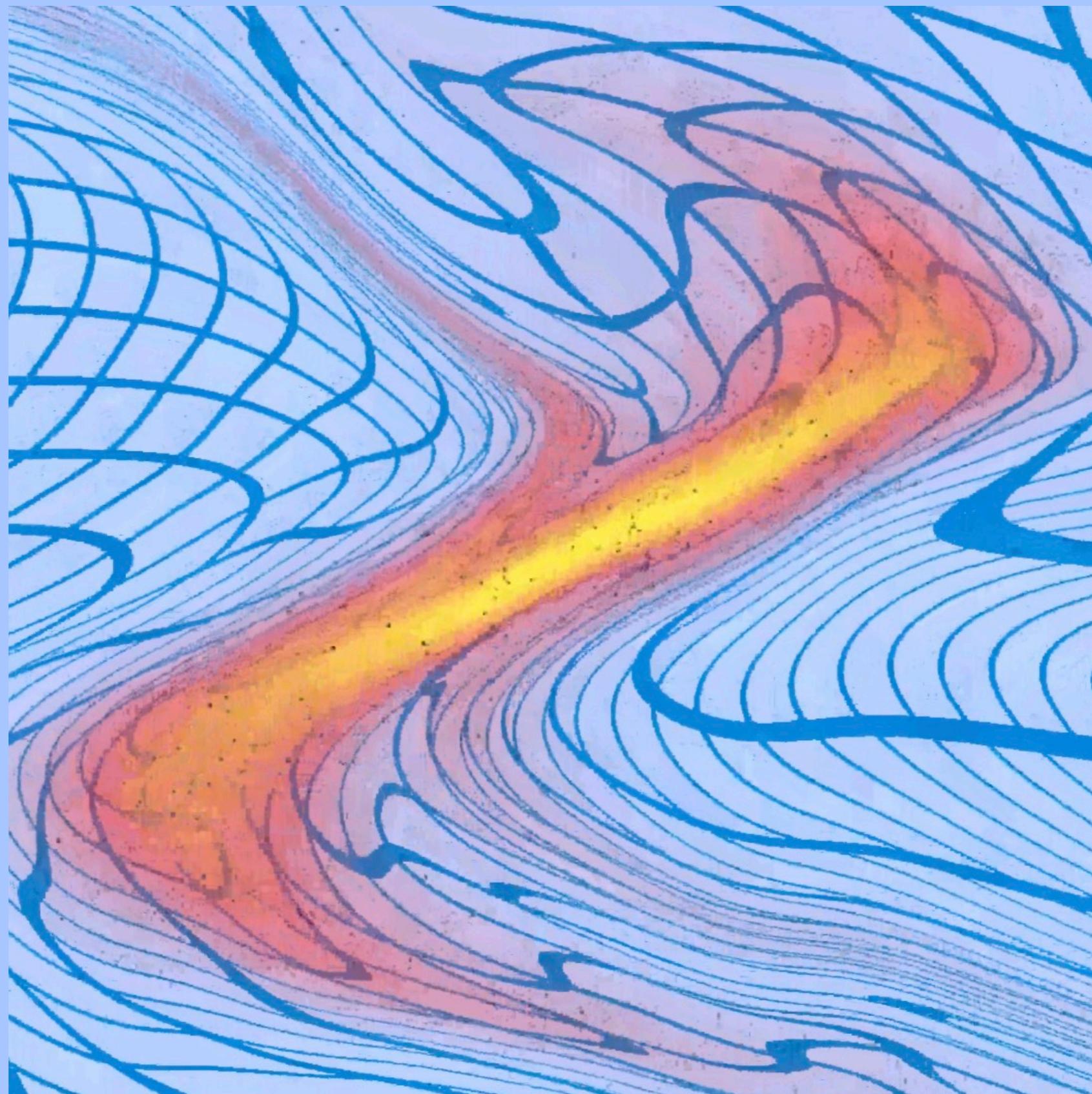


COLOR :  $\ln|\det \frac{\partial z}{\partial x}|$  (YELLOW = HIGH)

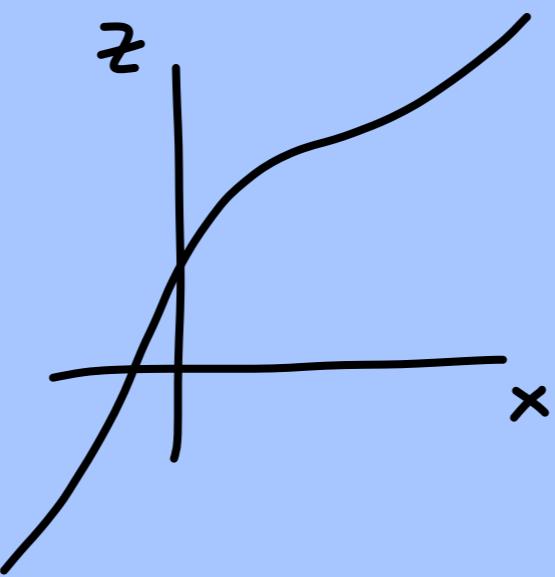
# LEARNING FROM SAMPLES



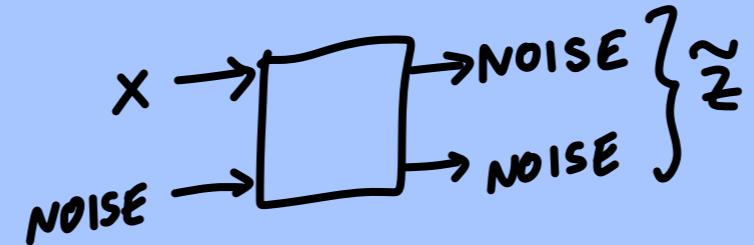
POINTS: TARGET DISTRIBUTION  $p(x)$  COLOR: NN  $q_\theta(x)$



1D CASE?



EXTEND:  $\tilde{x} = \begin{pmatrix} x \\ \text{NOISE} \end{pmatrix} \in \mathbb{R}^2$   
 $\mathbb{R}^1 \quad \mathbb{R}^1$  (e.g. GAUSSIAN)



# CONVOLUTIONAL LAYERS

$x_1 \rightarrow$  ONE OR SEVERAL CHANNELS OF AN IMAGE

$[x_1]_{j,c} \rightarrow$  CHANNEL  
LOCATION

$x_2 \rightarrow$  OTHER CHANNELS

$$x = (x_1, x_2)$$

$$x_2' = \exp(\underbrace{s(x_1)}_{\text{NONLOCAL (IN SPACE & CHANNEL)}}) \odot x_2 + \underbrace{t(x_1)}_{\text{NONLOCAL IN SPACE & CHANNEL}} \circ$$

NONLOCAL  
(IN SPACE & CHANNEL)

LOCAL (ELEMENTWISE)  
IN SPACE & CHANNEL

EXAMPLE:

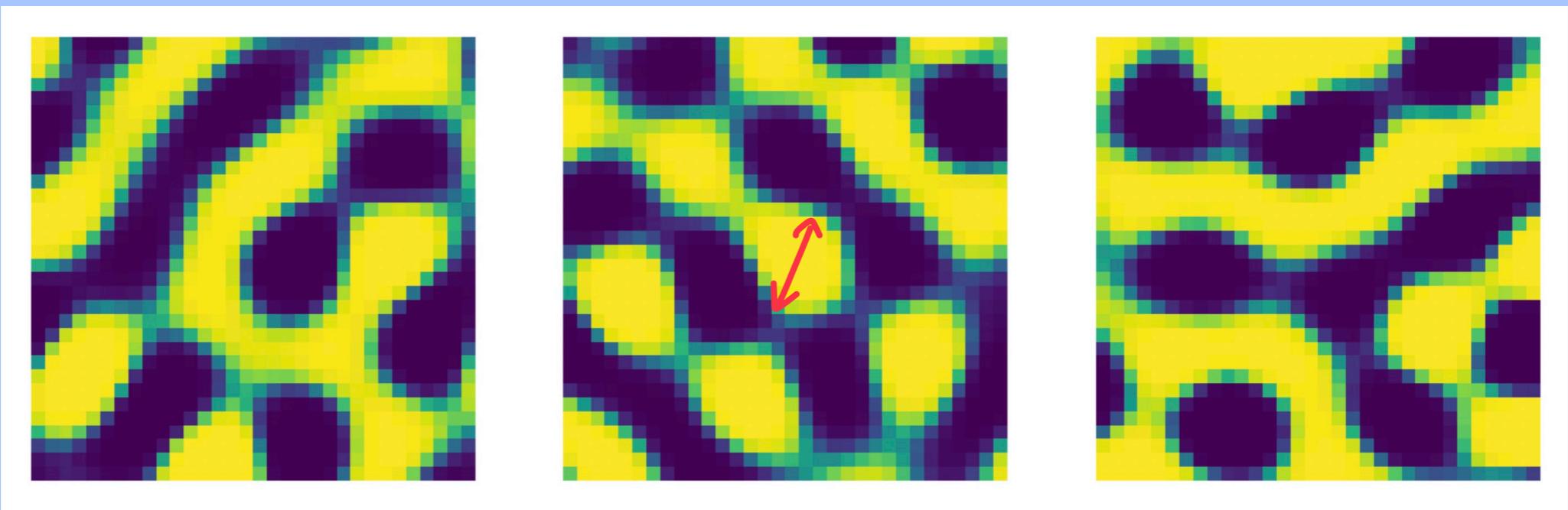
$$[s(x_1)]_{j,c} = f \left( \sum_{j',c'} w_{cc'}(j-j') [x_1]_{j'c'} + b_c \right)$$

"KERNEL"

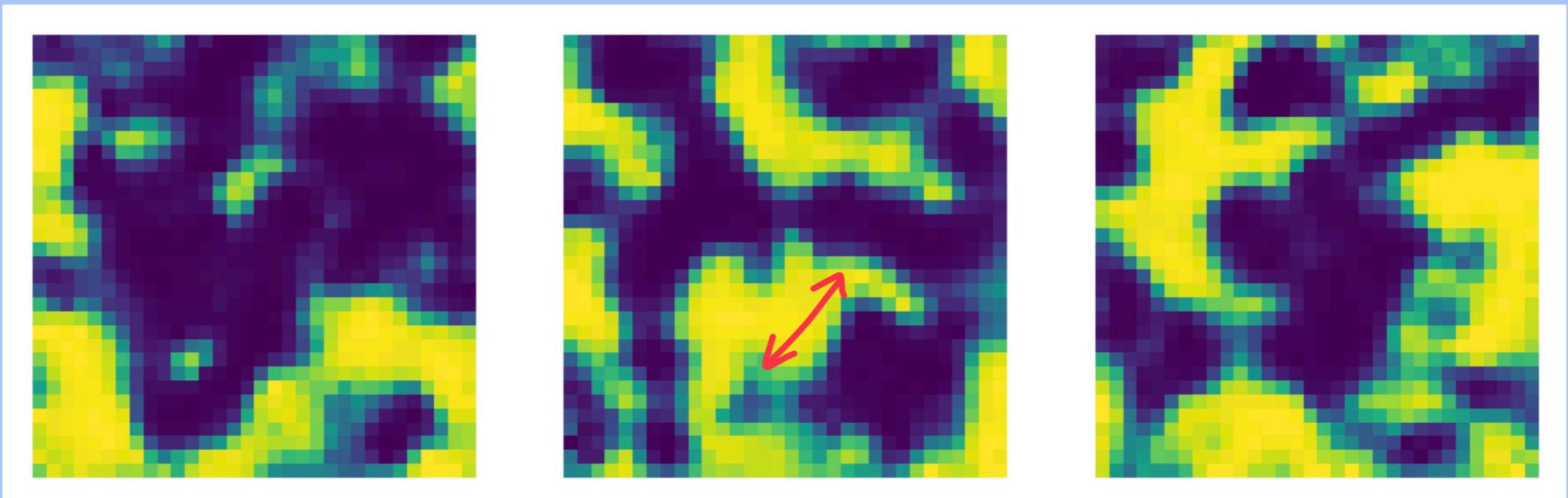
$$[x_2']_{j,c} = e^{[s(x_1)]_{j,c} \cdot [x_2]_{j,c} + [t(x_1)]_{j,c}}$$

# OF CHANNELS STAYS CONSTANT

TRUE SAMPLES

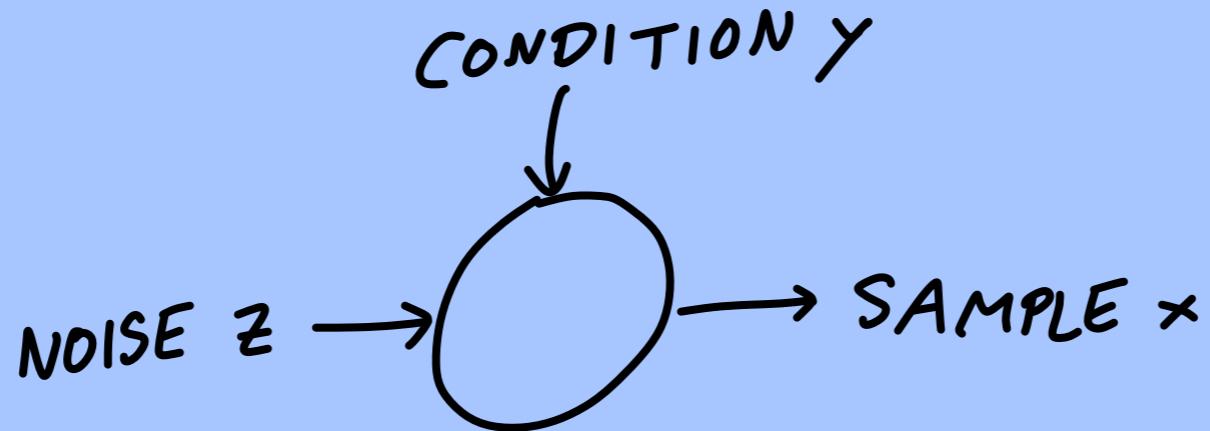


SAMPLES FROM NN



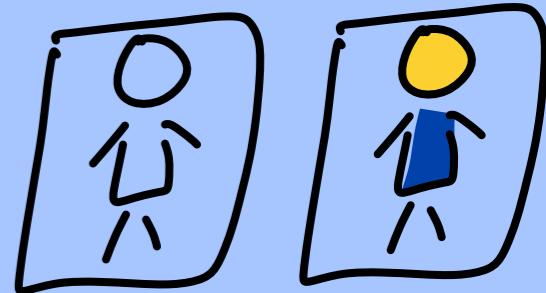
AFTER TRAINING ON  $\sim 2 \cdot 10^4$  BATCHES (32 SAMPLES/BATCH)

# CONDITIONAL INVERTIBLE NN



EXAMPLES:

- COND = B&W IMAGE  
SAMPLE = COLOR IMAGE



- $\text{COND } y = \text{TEMP. } T$   
SAMPLE = FIELD CONFIG. AT  $T$
- $\text{COND } y = \text{OUTCOME OF MSMT}$   
SAMPLE FROM  $P(\lambda|y)$  [BAYES]

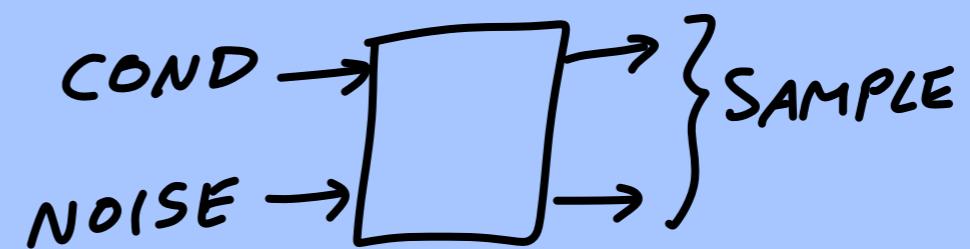
IDEA:

$$s_{\theta}(x_1, y) \quad \& \quad t_{\theta}(x_1, y)$$

$\uparrow$   
COND

$\uparrow$   
COND

ALTERNATIVE:



USE  $\underbrace{q_\theta(x)}$  TO CALCULATE  $H_p$   
 FROM NN

$H_p \approx$  ENTROPY OF  $q_\theta$

$$H_{q_\theta} = - \left\langle \log \underbrace{q_\theta(x)}_{\text{SEE ABOVE}} \right\rangle_{x \sim q_\theta(x)}$$

$$= - \left\langle \log p_z(z(x)) \right\rangle_{x \sim q_\theta(x)} - \left\langle \log \left| \det \frac{\partial z}{\partial x} \right| \right\rangle_{x \sim q_\theta(x)}$$

$$= - \underbrace{\left\langle \log p_z(z) \right\rangle}_{z \sim p_z(z)} - \underbrace{\left\langle \log \left| \det \frac{\partial z}{\partial x} \right| \right\rangle}_{z \sim p_z(z)}$$

$H_{p_z}$

ANALYTICALLY

# LEARNING TO SAMPLE FROM AN EXPLICITLY KNOWN $p(x)$

EXAMPLE:  $p(x) = \frac{\exp\left[-\frac{E(x)}{k_B T}\right]}{Z \sim \text{NORM.}}$

$$E(x) = \sum_{l < j} U(|x_l - x_j|)$$

IDEA: NOW MINIMIZE  $E$

$$\begin{aligned} D_{KL}(q_\theta \| p) &= \int d\mathbf{x} q_\theta(\mathbf{x}) \log \frac{q_\theta(\mathbf{x})}{p(\mathbf{x})} \\ &= \left\langle \log \frac{q_\theta(\mathbf{x})}{p(\mathbf{x})} \right\rangle_{\mathbf{x} \sim q_\theta(\mathbf{x})} \\ &= -H_{q_\theta} - \left\langle \log p(\mathbf{x}) \right\rangle_{\mathbf{x} \sim q_\theta(\mathbf{x})} \\ &= -H_{q_\theta} + \ln Z + \left\langle \frac{E(\mathbf{x})}{k_B T} \right\rangle_{\mathbf{x} \sim q_\theta(\mathbf{x})} \end{aligned}$$

$\log = \ln$

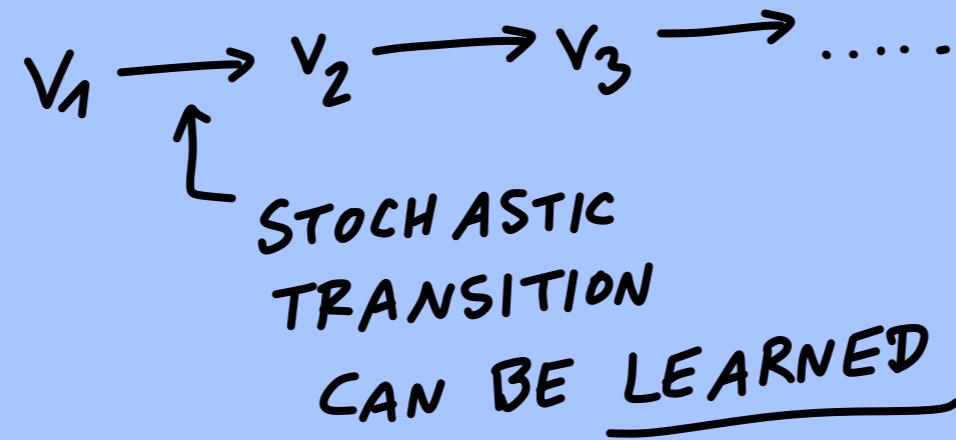
$$= \frac{\text{"FREE ENERGY"} }{k_B T} \text{ IN PHYSICS}$$

$$F = E - TS$$

7.3

## LEARNING A DISCRETE PROBABILITY DISTRIBUTION: RESTRICTED BOLTZMANN MACHINE

IDEA :

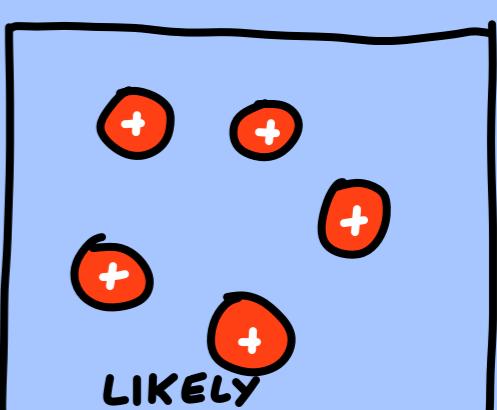
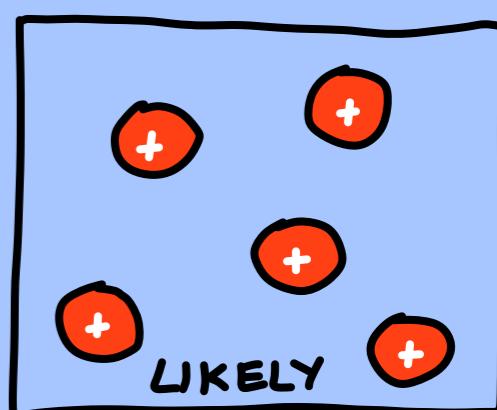
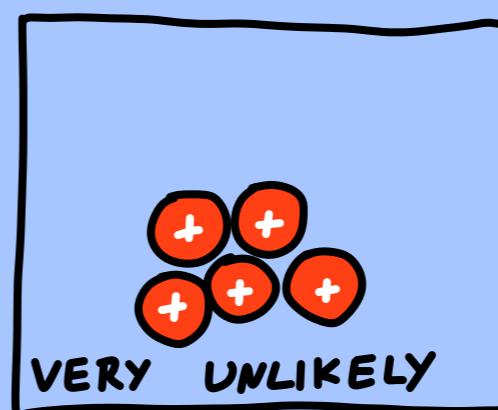
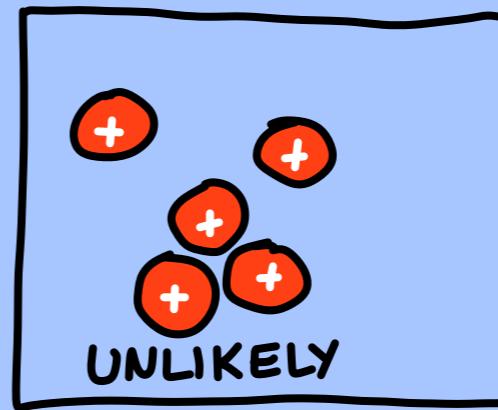
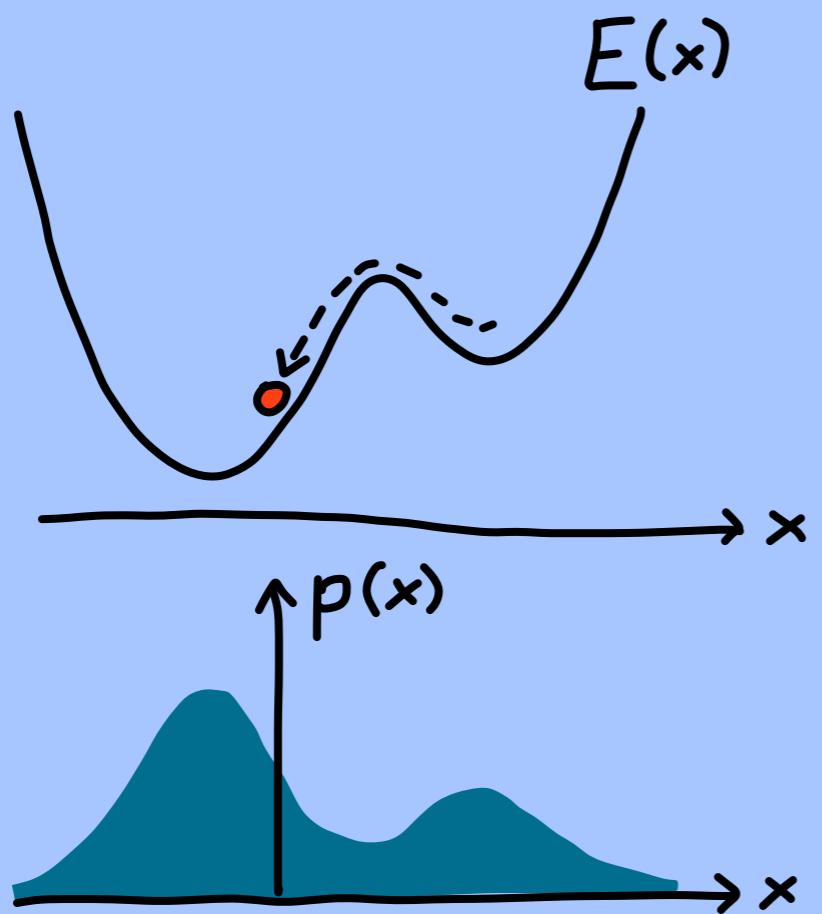


GOAL  
 $\underbrace{P_\theta(v)}_{\text{FROM THIS "CHAIN"}}$   $\approx \underbrace{P(v)}_{\text{TARGET}}$

# BOLTZMANN MACHINES (IN GENERAL, NOT RESTRICTED)

THERMAL EQUILIBRIUM :  
CONFIGURATIONS WITH LOWER  
ENERGIES ARE MORE LIKELY

$$P(x) \sim e^{-\frac{E(x)}{k_B T}}$$



# BOLTZMANN MACHINES

-  $E_\theta(s)$

$$P_\theta(s) = \frac{e^{-E_\theta(s)}}{Z_\theta}$$

$$\beta = \frac{1}{k_B T} = 1$$

$$Z_\theta = \sum_s e^{-E_\theta(s)}$$

$$D_{KL}(P || P_\theta) = \sum_s P(s) \ln \frac{P(s)}{P_\theta(s)} \stackrel{!}{=} \underset{\theta}{\text{MIN}}$$

$$\Leftrightarrow \sum_s P(s) \ln P_\theta(s) \stackrel{!}{=} \underset{\theta}{\text{MAX}}$$

$$\Leftrightarrow \left\langle \ln P_\theta(s) \right\rangle_{s \sim P(s)} \stackrel{!}{=} \text{MAX}$$

LOG-LIKELIHOOD

$$\frac{\partial}{\partial \theta} \langle \ln P_\theta(s) \rangle = - \left\langle \frac{\partial}{\partial \theta} E_\theta(s) \right\rangle - \frac{\partial}{\partial \theta} \ln Z_\theta$$

IF  $E_\theta(s) = \sum_j \theta_j \underbrace{\varepsilon_j(s)}_{\text{LINEAR IN } \theta}$   $\Rightarrow P_\theta$  : "EXPONENTIAL FAMILY" OF PROB. DISTR.

EXAMPLE  
 $s = (s_1, s_2, s_3, \dots)$

 $\varepsilon_j(s) = s_j \cdot s_{j+1}$

THEN  $\left\langle \frac{\partial}{\partial \theta_j} E_\theta(s) \right\rangle = \left\langle \varepsilon_j(s) \right\rangle_{s \sim P(s)}$  "DATA"

$$\begin{aligned} \frac{\partial}{\partial \theta_j} \ln Z_\theta &= \frac{1}{Z_\theta} \sum_s (-\varepsilon_j(s)) e^{-E_\theta(s)} \\ &= - \left\langle \varepsilon_j(s) \right\rangle_{s \sim P_\theta(s)} \text{"MODEL"} \end{aligned}$$

$\Rightarrow$

$$\left| \frac{\partial}{\partial \theta_j} \left\langle \ln P_{\theta}(s) \right\rangle_{s \sim P(s)} \right. = \underbrace{\left\langle \varepsilon_j(s) \right\rangle_{s \sim P_{\theta}(s)}}_{\text{MODEL}} - \underbrace{\left\langle \varepsilon_j(s) \right\rangle_{s \sim P(s)}}_{\text{DATA}}$$

$\rightsquigarrow$  GENERAL UPDATE RULE  
FOR BOLTZMANN MACHINES

$$\left\langle \varepsilon_j \right\rangle_{\text{MODEL}} > \left\langle \varepsilon_j \right\rangle_{\text{DATA}} \\ \Rightarrow S\theta_j > 0$$

$$\Rightarrow \left\langle \varepsilon_j \right\rangle_{\text{MODEL}} \downarrow$$

NOTE : FOR GAUSSIAN

$$E(s) = \theta_2 s^2 + \theta_1 s$$
$$\& s \in \mathbb{R}$$

ALSO: BINOMIAL

$$P_\theta(s) \sim p^s (1-p)^{N-s}$$
$$= \exp[s \ln p + (N-s) \ln (1-p)]$$
$$= \exp \left[ s \underbrace{\ln \frac{p}{1-p}}_{\theta} + \text{const} \right]$$

MOST IMPORTANT CASE:

$$S = (s_1, s_2, \dots)$$

$s_j \in \{0, 1\}$  'BITS'

$\in \{-1, +1\}$  'SPINS'

&  $E_\ell(s) =$  LINEAR  
OR  
QUADRATIC IN S

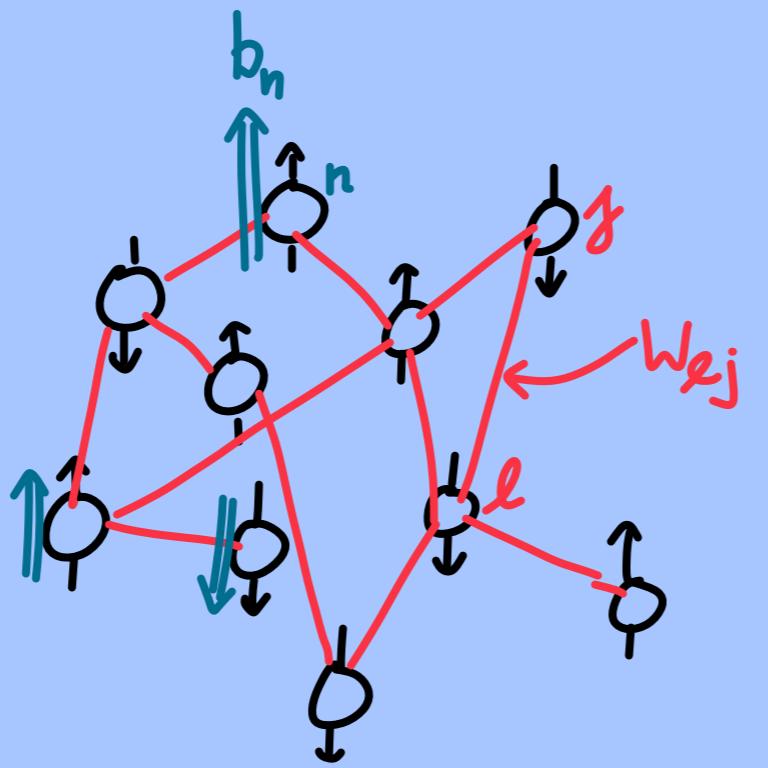
⇒ MOST GENERAL:

$$E_\theta(s) = - \sum_{\ell, j} s_\ell W_{\ell j} s_j - \sum_\ell b_\ell s_\ell$$

$\downarrow$   
COUPLING                            "MAGNETIC FIELD"

$W_{\ell j} > 0 \Rightarrow$   
TENDENCY  $s_\ell s_j > 0$

$b_\ell > 0 \Rightarrow s_\ell > 0$   
(TENDENCY)



"SPIN GLASS"

# UPDATE

$$\frac{\partial}{\partial w_{ej}} \langle \ln P_\theta(s) \rangle_{s \sim P(s)} = \underbrace{\langle S_e S_j \rangle}_{\text{DATA}} - \underbrace{\langle S_e S_j \rangle}_{\text{MODEL}}_{s \sim P_\theta(s)}$$

$$\theta = (\begin{matrix} \text{ALL } w_i \\ \text{ALL } b \end{matrix})$$

$$\text{r.h.s.} > 0 \Rightarrow w_{ej} \uparrow \Rightarrow \underbrace{\langle S_e S_j \rangle}_{\text{MODEL}} \uparrow$$

$$\frac{\partial}{\partial b_e} \langle \ln P_\theta(s) \rangle_{s \sim P(s)} = \underbrace{\langle S_e \rangle}_{\text{DATA}} - \underbrace{\langle S_e \rangle}_{\text{MODEL}}_{s \sim P_\theta(s)}$$

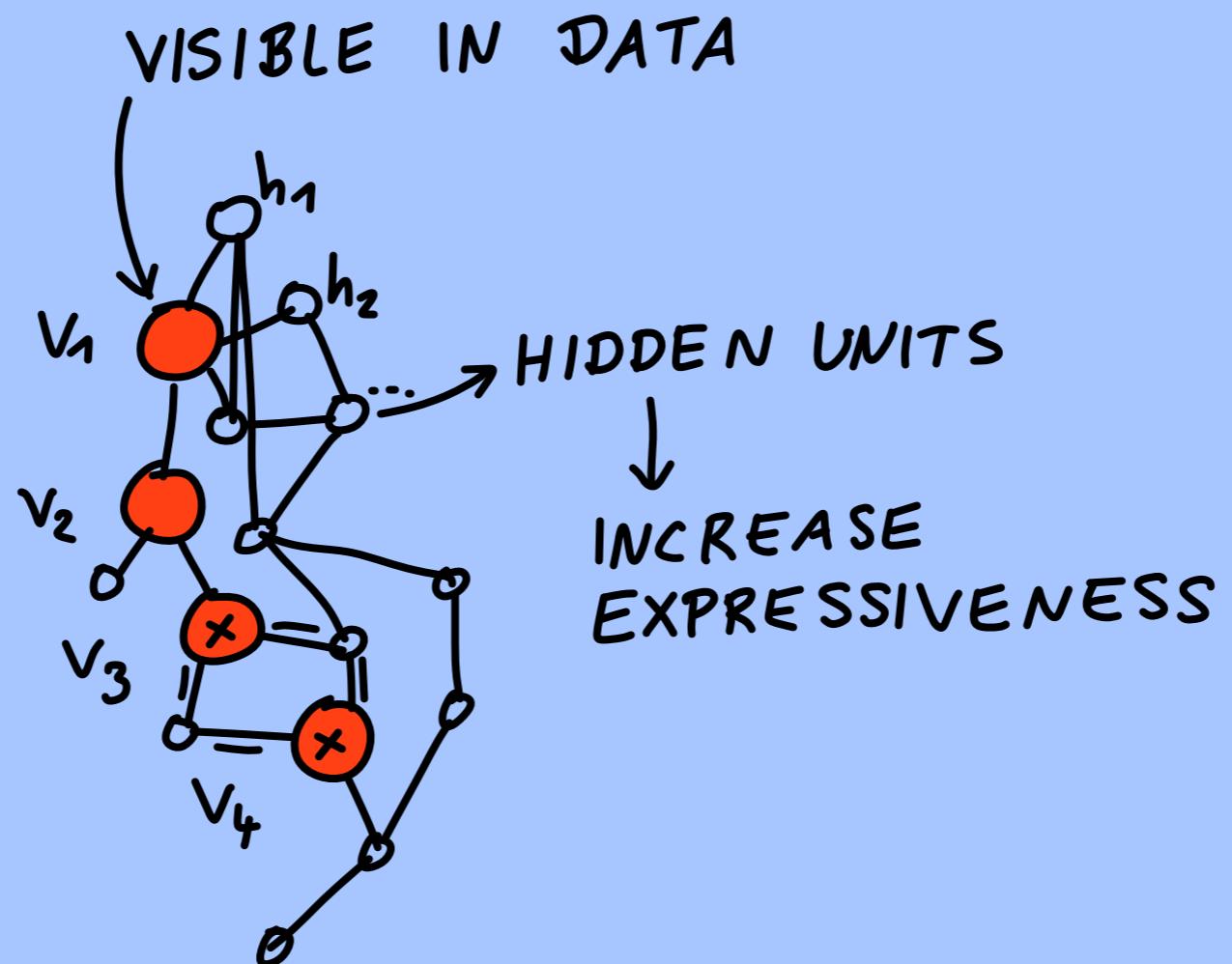
$\Rightarrow$  COULD LEARN MODEL FOR SYSTEM  
FROM DATA

BUT MAYBE REAL  $E = \dots + S_3 \cdot S_5 \cdot S_9 + \dots$   
 $\Rightarrow ?$

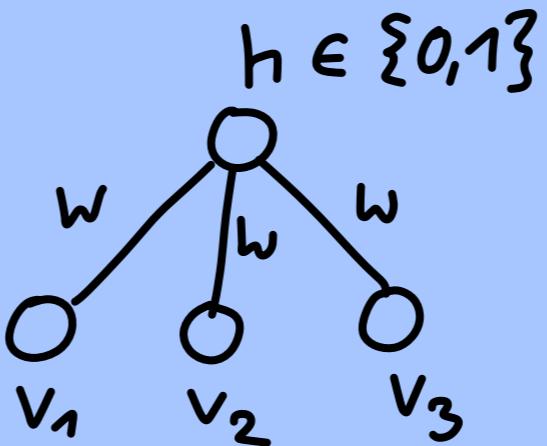
IDEA: EXTEND DEGREES OF FREEDOM  
"HIDDEN VARIABLES"  $h = (h_1, h_2, \dots)$   
vs.

"VISIBLE VARIABLES"  $v = (v_1, v_2, \dots)$

$s = (v, h)$



EXAMPLE



$$P(v, h) \sim e^{wh(v_1 + v_2 + v_3)}$$

$$P(v) = \sum_{h=0,1} P(v, h) \sim 1 + e^{-E(v)} \sim e^{w(v_1 + v_2 + v_3)}$$

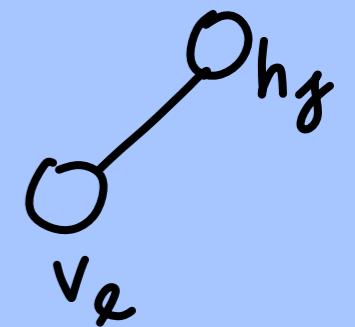
$$E(v) = \text{const} - \underbrace{\ln(1 + e^{w(v_1 + v_2 + v_3)})}_{\text{TAYLOR}}$$

TAYLOR

$$\approx \dots O(v^1) + \dots O(v^2) + w^3 (v_1 + v_2 + v_3)^3 \cdot \text{const} + \dots$$

v<sub>1</sub> v<sub>2</sub> v<sub>3</sub>

$$\underbrace{\langle v_e h_g \rangle}_{\text{DATA}}_{S \sim P(s)} = ?$$



BUT DATA ONLY SUPPLIES  
STATISTICS OF  $v \Rightarrow ?$

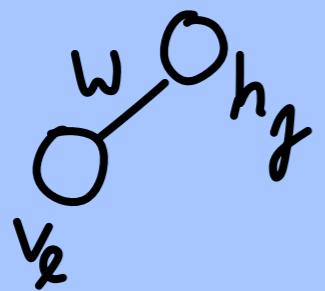
IDEA:

$$P(v, h) := \underbrace{P_\theta(h|v)}_{\text{"DATA"} \atop \text{FROM MODEL, GIVEN } v} \cdot P(v)$$

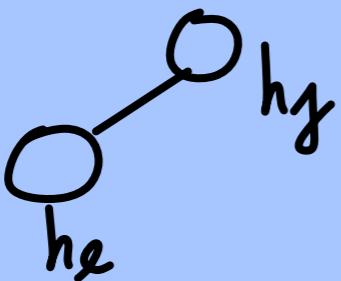
$$\text{"DATA"} \quad s \sim P(s) \xrightarrow{\text{NOW}} v \sim P(v), \quad h \sim P_\theta(h|v)$$

ALTERNATIVE:

$$\mathcal{D}_{KL}(P(v) \parallel \underbrace{P_\theta(v)}_{= \sum_h P_\theta(v, h)}) \stackrel{!}{=} \text{MIN}$$



$$\frac{\partial}{\partial w_{ej}} \left\langle \ln P_\theta(s) \right\rangle_{s \sim P(s)} = .. = \underbrace{\left\langle v_e h_g \right\rangle_{v \sim P(v)}_{h \sim P_\theta(h|v)}}_{\text{"DATA"}} - \underbrace{\left\langle v_e h_g \right\rangle_{v, h \sim P_\theta(v, h)}}_{\text{MODEL}}$$



$$.... = \underbrace{\left\langle h_e h_g \right\rangle_{v \sim P(v)}_{h \sim P_\theta(h|v)}}_{\text{MODEL}} - \underbrace{\left\langle h_e h_g \right\rangle_{v, h \sim P_\theta(v, h)}}_{\text{MODEL}}$$

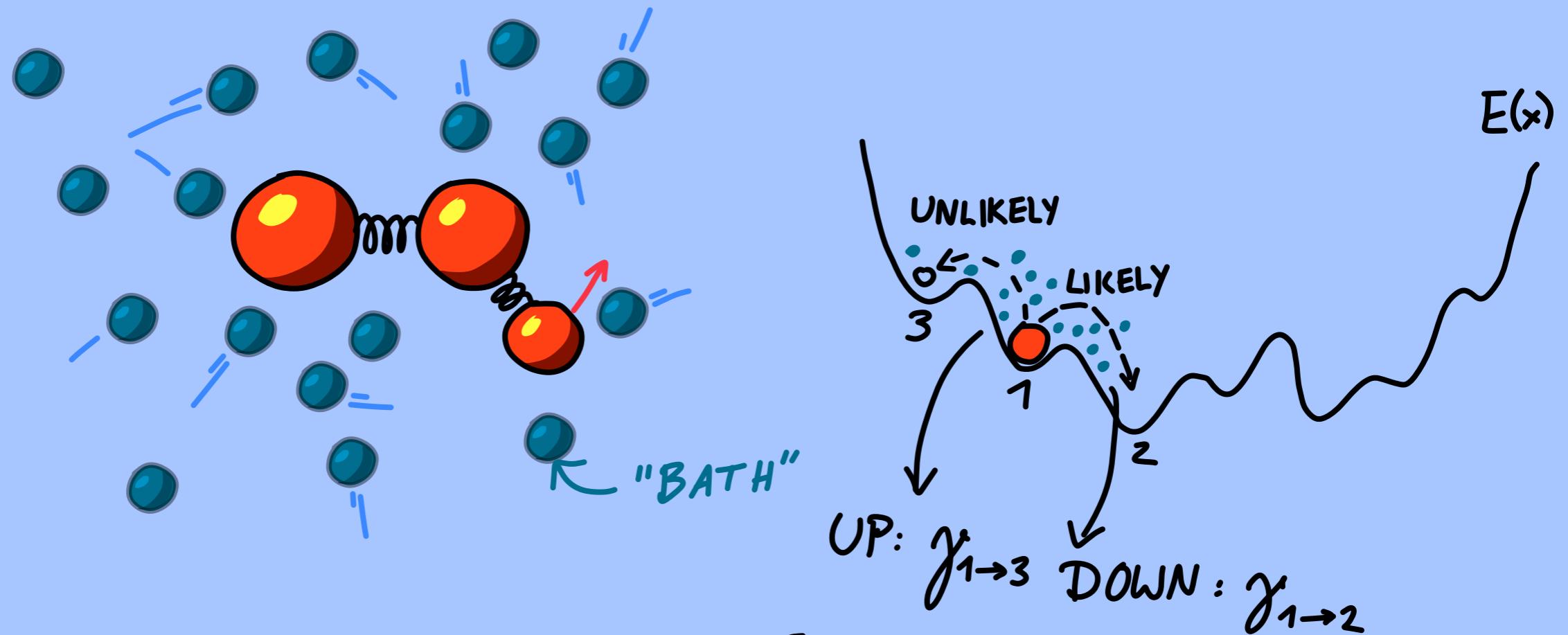
MODEL, "CLAMPED":  
FIX  $v$  TO DATA

HOW TO SAMPLE FROM  $s = (v, h)$   
ACCORDING TO THE  
MODEL

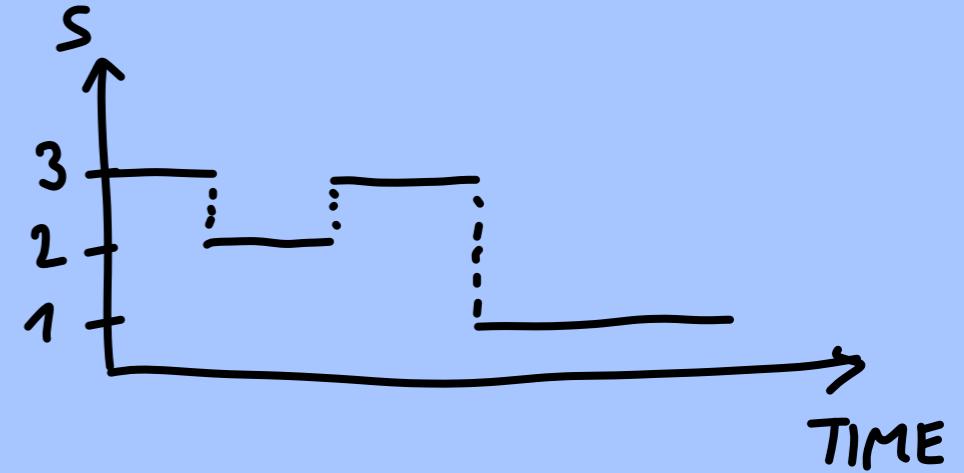
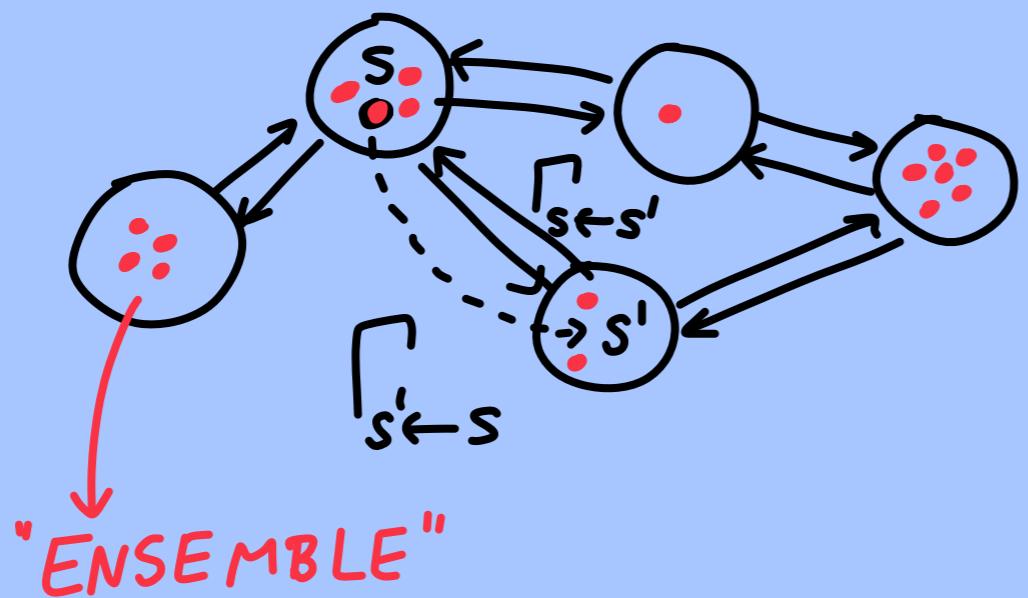
$$P_\theta(s) \sim e^{-E_\theta(s)}$$

$\Rightarrow$  MONTE CARLO !

# REMINDER: MARKOV CHAIN MONTE CARLO



$$\frac{\gamma_{3 \rightarrow 1}}{\gamma_{1 \rightarrow 3}} = e^{\frac{E_3 - E_1}{k_B T}} \quad \gamma_{3 \rightarrow 1}$$



$P(s, t) =$  FRACTION OF PARTICLES  
IN  $s$  AT TIME  $t$

$$\frac{d}{dt} P(s, t) = \underbrace{\sum_{s'} \sum_{s \leftarrow s'} P(s', t)}_{\text{IN}} - \underbrace{\sum_{s'} \sum_{s' \leftarrow s} P(s', t)}_{\text{OUT}}$$

STEADY STATE?

$$\frac{d}{dt} P(s, t) = 0 \quad \forall s$$

IF

$$\frac{\int_{s \leftarrow s'}^{} ds'}{\int_{s' \leftarrow s}^{} ds} = \frac{f(s)}{f(s')} \quad \forall s, s'$$

"DETAILED BALANCE"

THEN

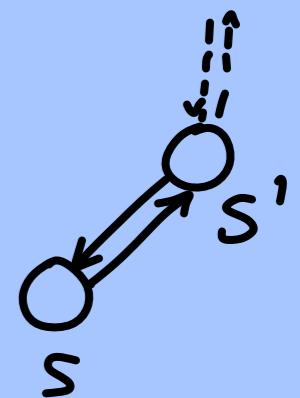
$\frac{dP}{dt} = 0$  IS SOLVED BY:

$$P(s) = \frac{f(s)}{\text{Normaliz.}}$$

⇒ PROOF: CHECK

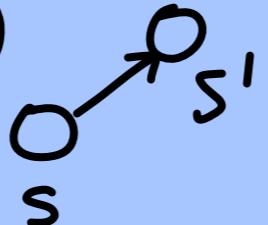
PHYSICS :  $f(s) = e^{-\frac{E(s)}{k_B T}}$

$$P(s) = \frac{1}{Z} e^{-\frac{E(s)}{k_B T}}$$

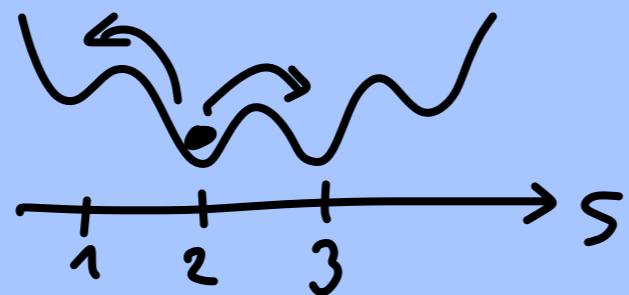


$\Rightarrow$  CAN SAMPLE FROM  $P(s)$   
BY CHOOSING  $\Gamma$  IN THIS WAY

METROPOLIS  
 $E(s) > E(s')$



$$\begin{aligned}\Gamma_{s' \leftarrow s} &= \gamma \\ \Gamma_{s \leftarrow s'} &= \gamma e^{-\frac{E(s) - E(s')}{kT}}\end{aligned}$$



$s = \text{CONFIGURATIONS}$

$$\begin{bmatrix} \uparrow & \downarrow & \uparrow \\ \downarrow & \uparrow & \uparrow \end{bmatrix} \text{ OR } \begin{bmatrix} \downarrow & \downarrow & \downarrow \\ \downarrow & \uparrow & \uparrow \end{bmatrix} \text{ OR...}$$



NOTES:

- WE CAN FIX SOME VISIBLE UNITS

$$v = (v_{\text{FREE}}, v_{\text{FIXED}})$$

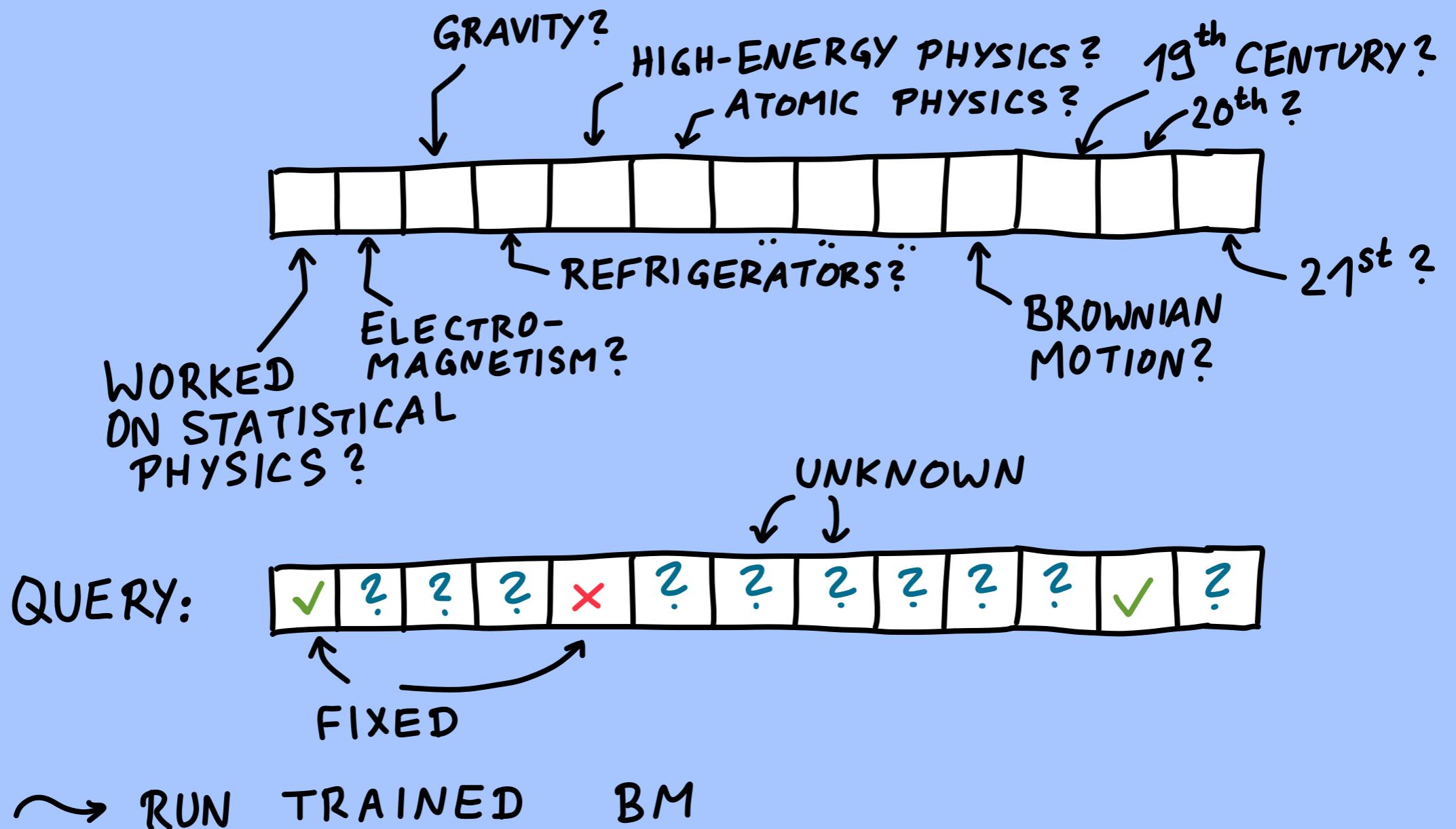
⇒ THEN WE SAMPLE FROM

$$P_\theta(v_{\text{FREE}} \mid v_{\text{FIXED}})$$

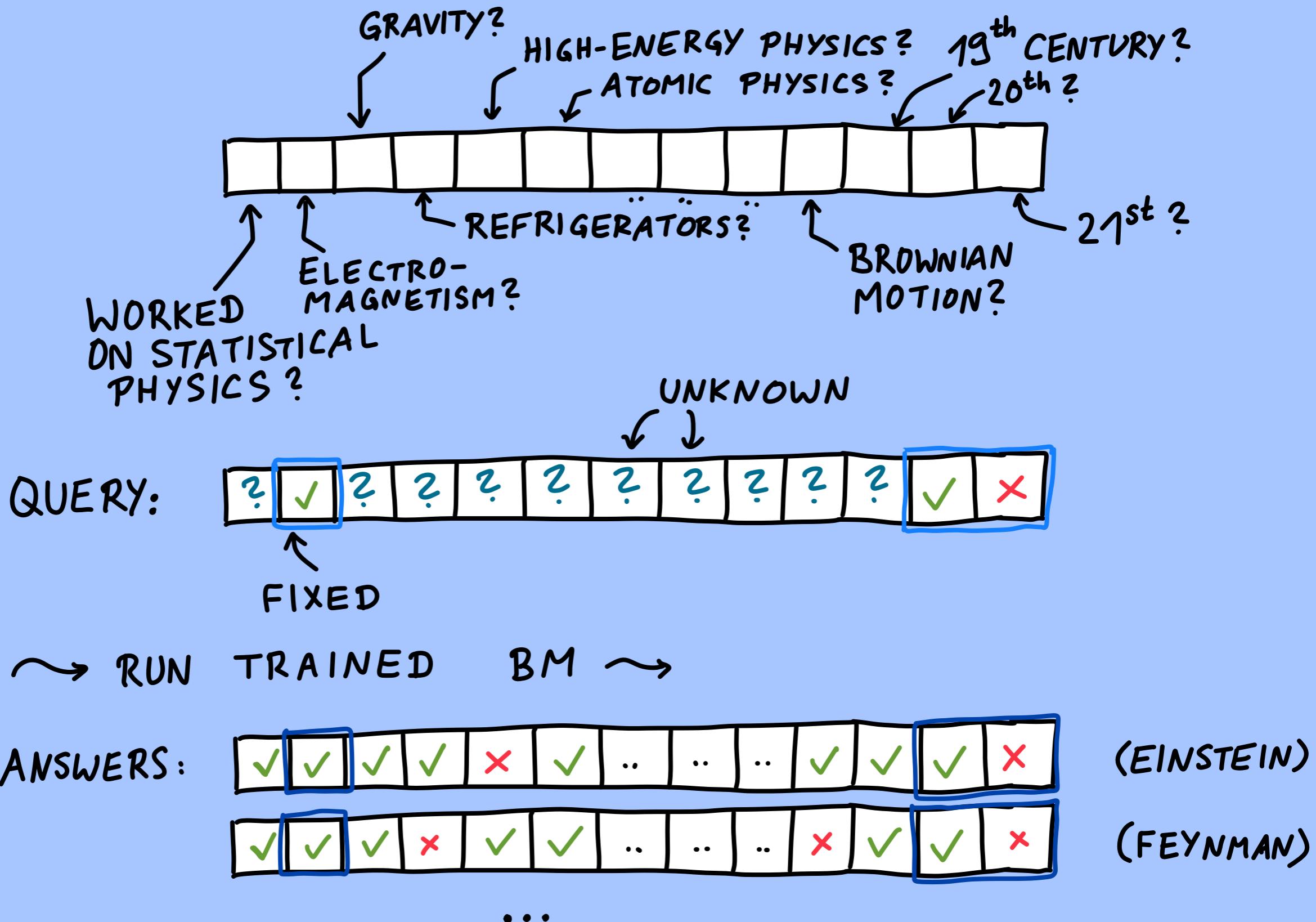
⇒ USEFUL FOR BAYES!

⇒ B.M. BECOMES  
PROBABILISTIC  
ASSOCIATIVE MEMORY

# “GUESS A PHYSICIST”

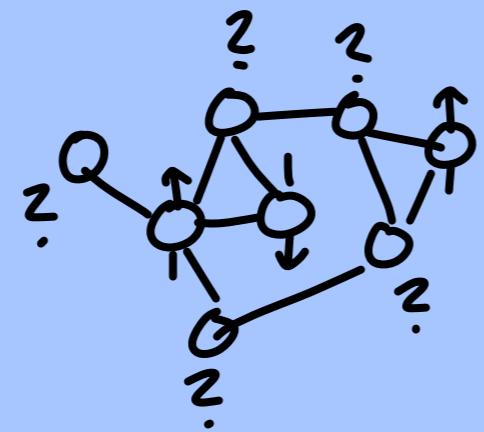


# "GUESS A PHYSICIST"



NOTE:

## HOPFIELD MODEL



LIKE  $T=0$  B.M.

NOTE:

- COULD IMPOSE EXTRA CONSTRAINTS, E.G. LOGIC:



$$E = \dots - B \underbrace{v_1 \cdot v_3 \cdot v_4}_{\text{BIG NUMBER}}$$

## RESTRICTED

## BOLTZMANN MACHINE

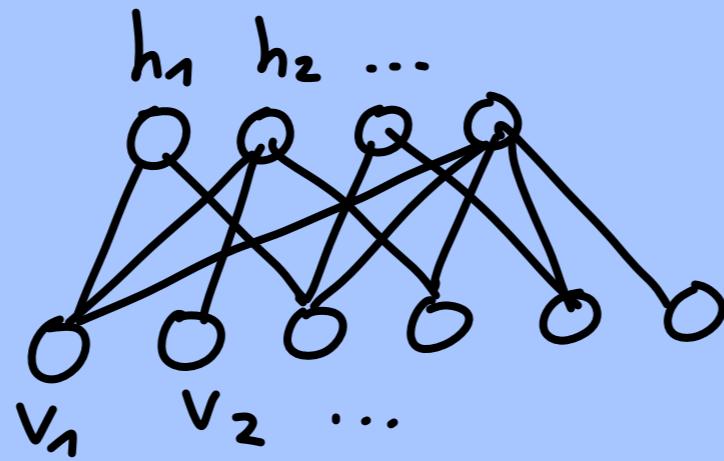
GOAL:

SIMPLE

$$P_\theta(h|v)$$

$$\& P_\theta(v|h)$$

IDEA:



NO v-v OR h-h

$$E = -\sum_j a_j v_j - \sum_j b_j h_j - \sum_{i,j} v_i w_{ij} h_j$$

WHAT IS

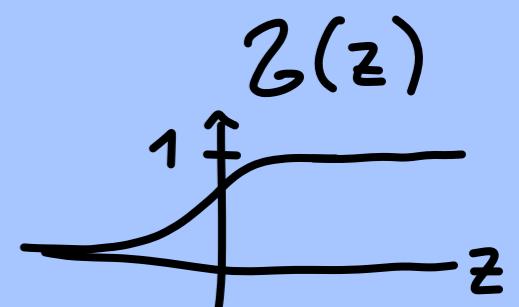
$$P_\theta(h|v) \quad \& \quad P_\theta(v) = \sum_h P_\theta(v, h)$$

$$P_\theta(v) = \sum_h P_\theta(v, h) = \frac{1}{Z_\theta} e^{\alpha^t v} \prod_j (1 + e^{z_j})$$

$$P_\theta(v, h) = \frac{e^{-E_\theta(v, h)}}{Z_\theta}$$

$$\sum_{h_j=0,1} e^{\underbrace{(b_j + \sum_i v_i w_{ij})}_{z_j} h_j} = 1 + e^{z_j}$$

$$\sum_h \dots = \sum_{h_1 \in \{0,1\}} \sum_{h_2 \in \{0,1\}} \dots$$



$$P_\theta(h_j=1 | v) = \sum_{\substack{\{h\} \\ h_j=1}} \frac{P_\theta(v, h)}{P_\theta(v)} = \frac{e^{z_j}}{1 + e^{z_j}} = \frac{1}{e^{-z_j} + 1}$$

$$= Z(z_j)$$

$$= Z(b_j + \sum_i v_i w_{ij})$$

$$h_j \text{ INDEPENDENT} \\ P_\theta(h|v) = \prod_j P_\theta(h_j | v)$$

LIKewise

$$P_\theta(v|h) = \prod_\ell P_\theta(v_\ell | h)$$

$$P_\theta(v_\ell=1|h) = Z(\alpha_\ell + \sum_j w_{\ell j} h_j)$$

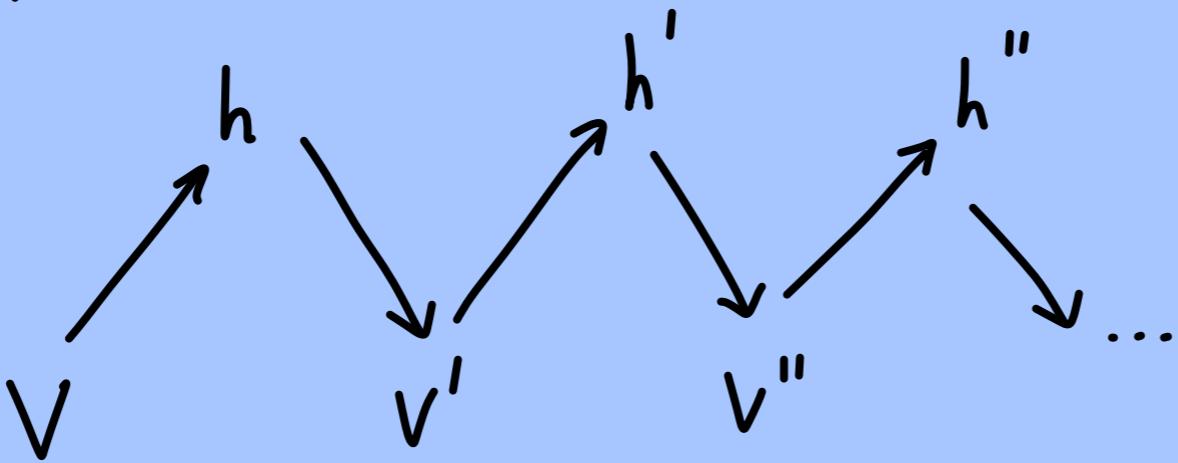
DETAILED BALANCE:

$$\frac{P_\theta(v|h)}{P_\theta(h|v)} = \frac{P_\theta(v)}{P_\theta(h)}$$

$\Rightarrow$  MARKOV CHAIN  
CONVERGES TO  
BOLTZM. DISTR.

MARKOV

CHAIN:

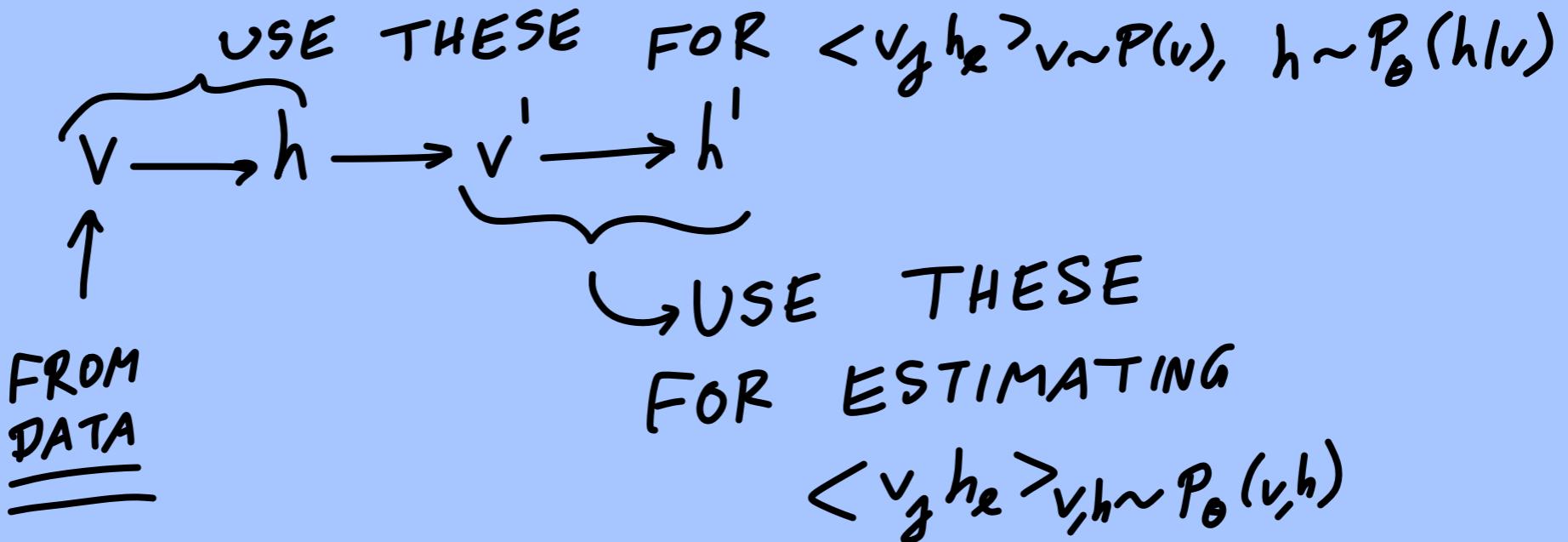


$\langle v_e h_y \rangle_{\text{MODEL}}$

$\langle v_e h_y \rangle^{\text{"DATA"}}$   
 $P(v)$   
 $P_\theta(h|v)$

FOR  $P_\theta(h, v)$

REPLACE FULL MARKOV CHAIN BY



WORKS WELL, SINCE TRAINING  
MAKES  $P_\theta(v) \rightarrow P(v)$

⇒ SAMPLING FROM DATA  $P(v)$   
IS ALMOST AS GOOD AS  
SAMPLING FROM MODEL  $P_\theta(v)$

"CONTRASTIVE DIVERGENCE"

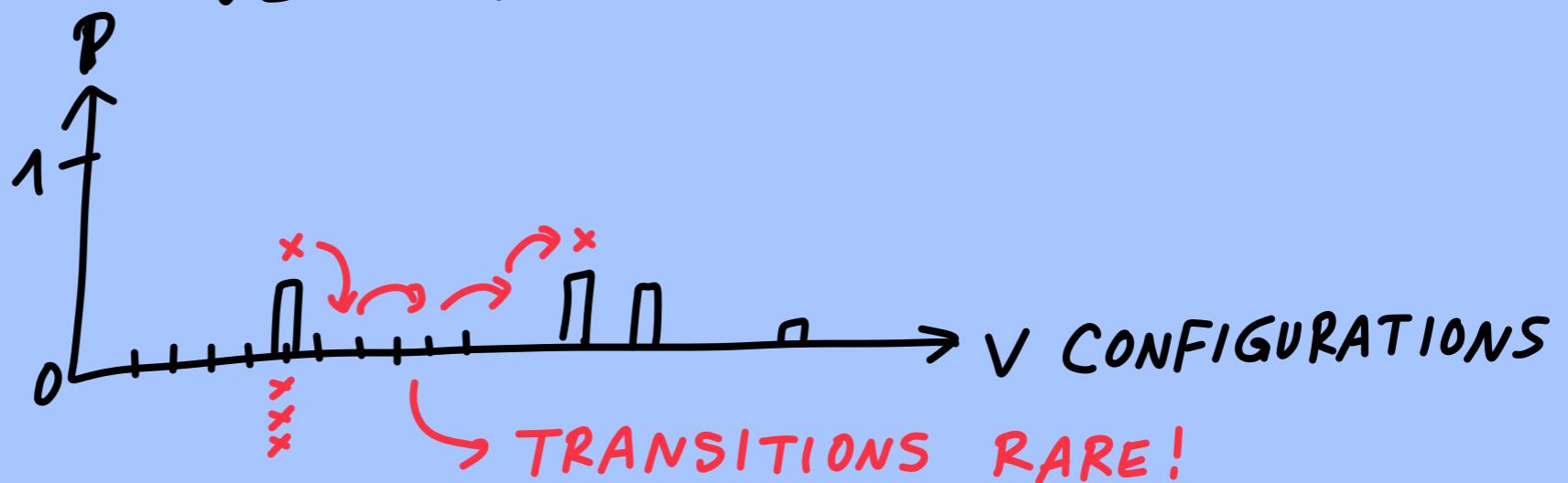
NOTE:

EXTRA TRICK

$$\langle v_y | h_e \rangle = \langle P(h_e=1|v) v_y \rangle$$

$\Rightarrow$  LESS FLUCTUATIONS!

NOTE : RBM DOES NOT WORK WELL ON VERY RESTRICTED DATASET



$\Rightarrow$  B.M. WORK BETTER IF  $P(v) > 0 \forall v$

ALSO: CAN USE 'HIGHER TEMPERATURE'

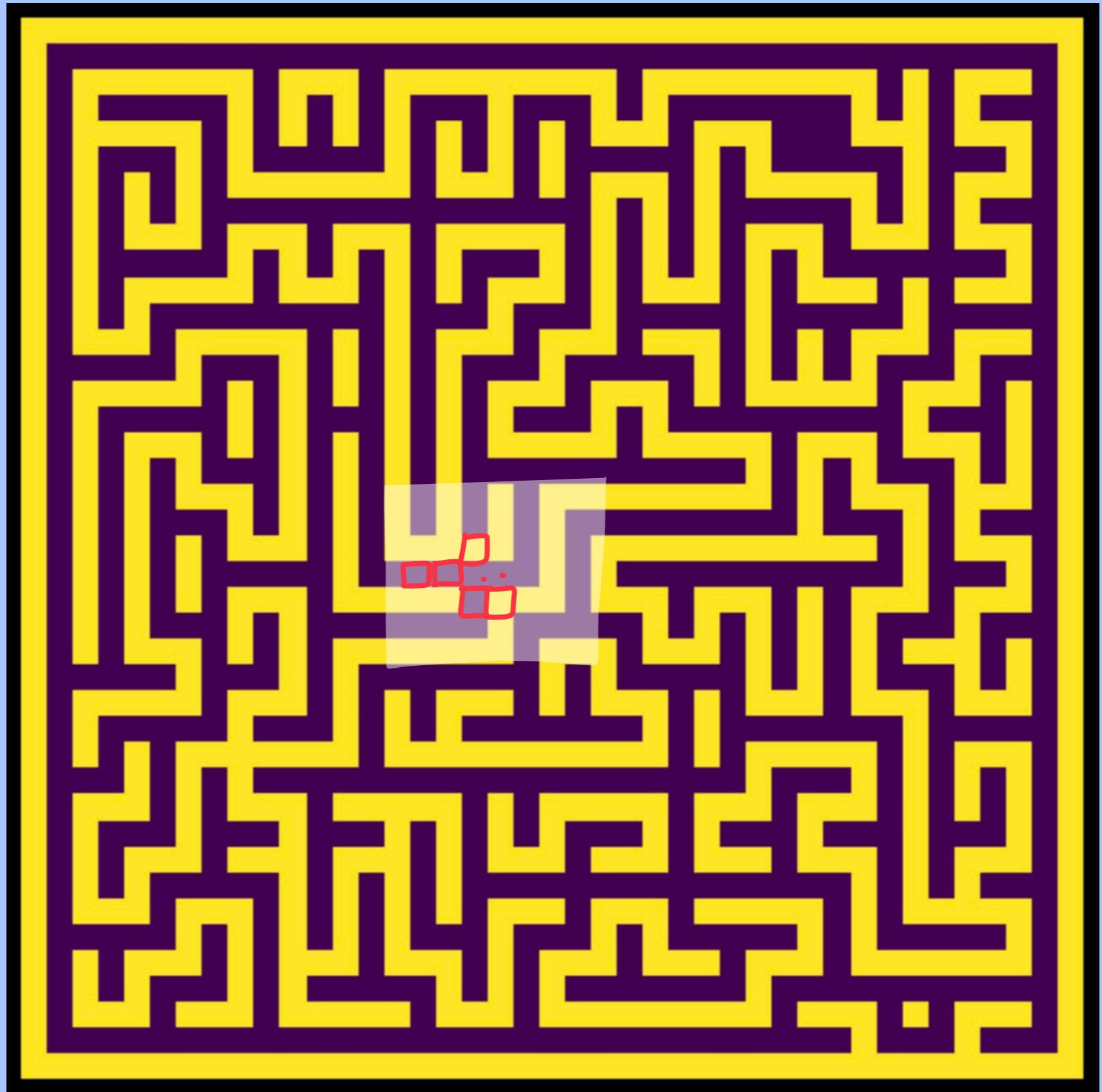
$$E \mapsto \beta E$$

INVERSE TEMP.

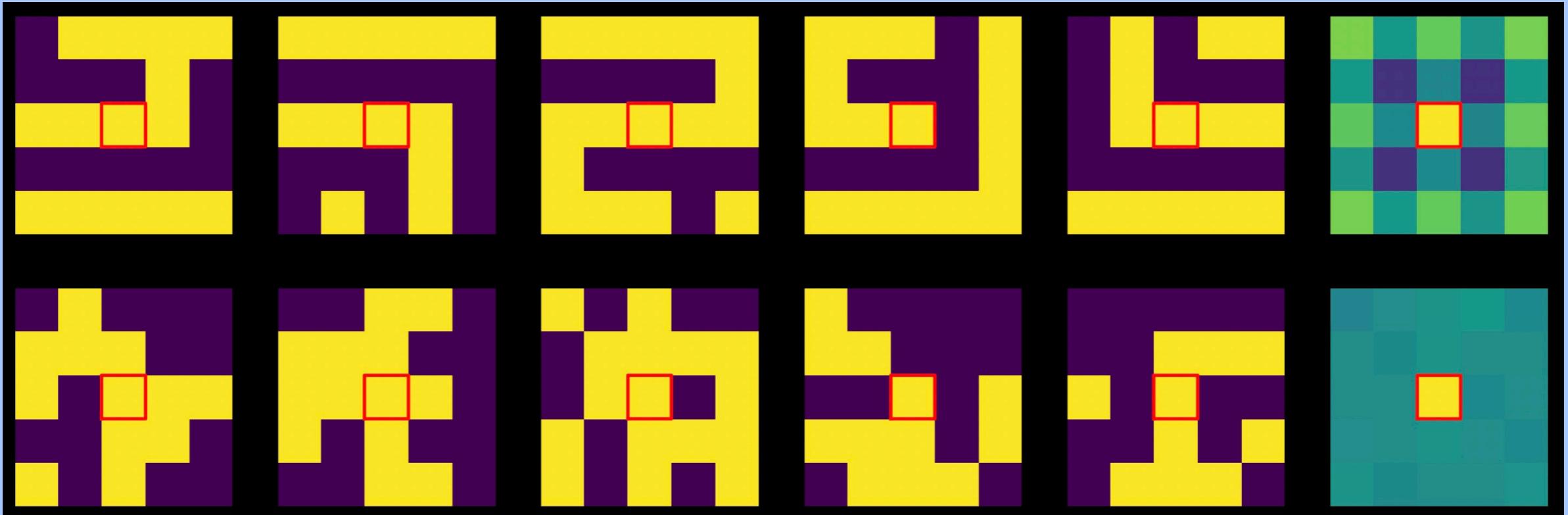
CHOOSE  $\beta$  SMALL  
& THEN  $\beta \rightarrow 1$

"ANNEALING"

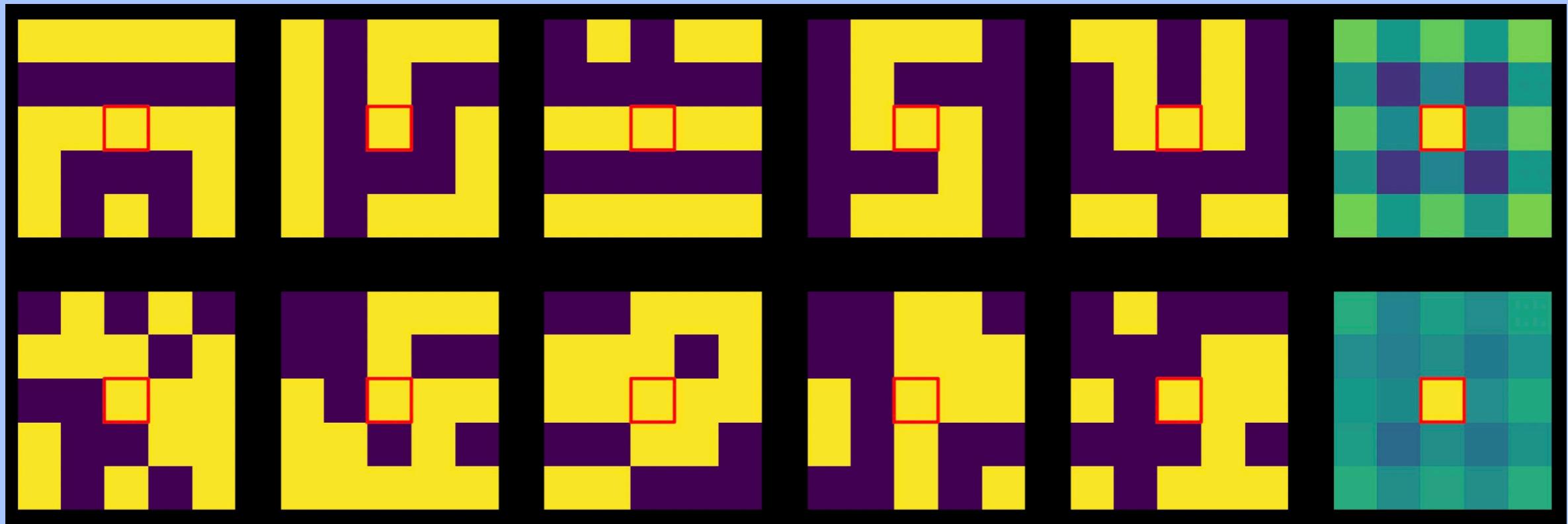
LATER



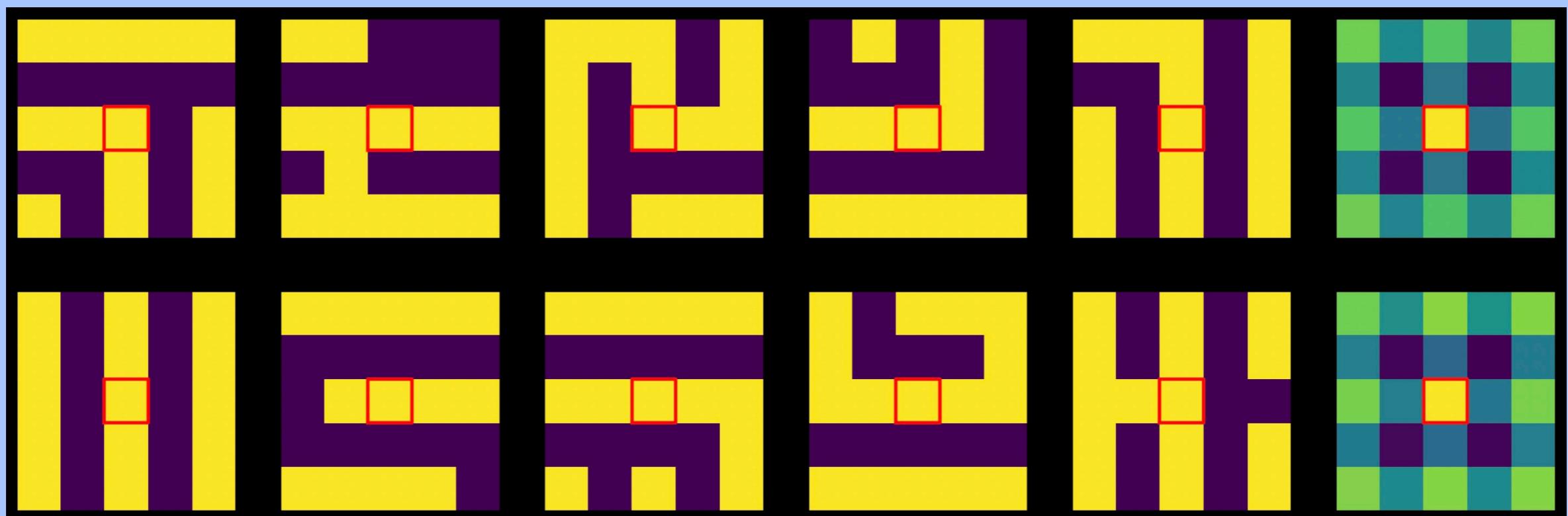
TRUE SAMPLES, WITH CONSTRAINT



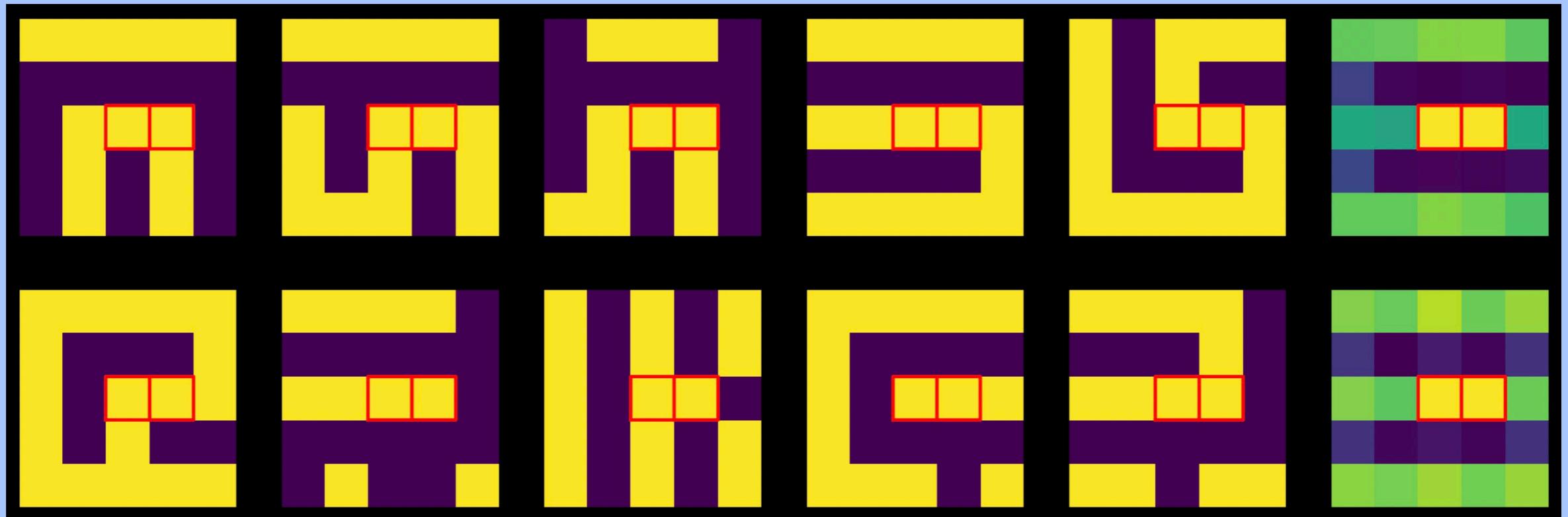
SAMPLES FROM RBM (BEFORE TRAINING)  
ALSO WITH CONSTRAINT



AFTER 1000 TRAINING BATCHES  
(BATCHSIZE 32)  
(LEARNING RATE  $\eta = 0.01$ )

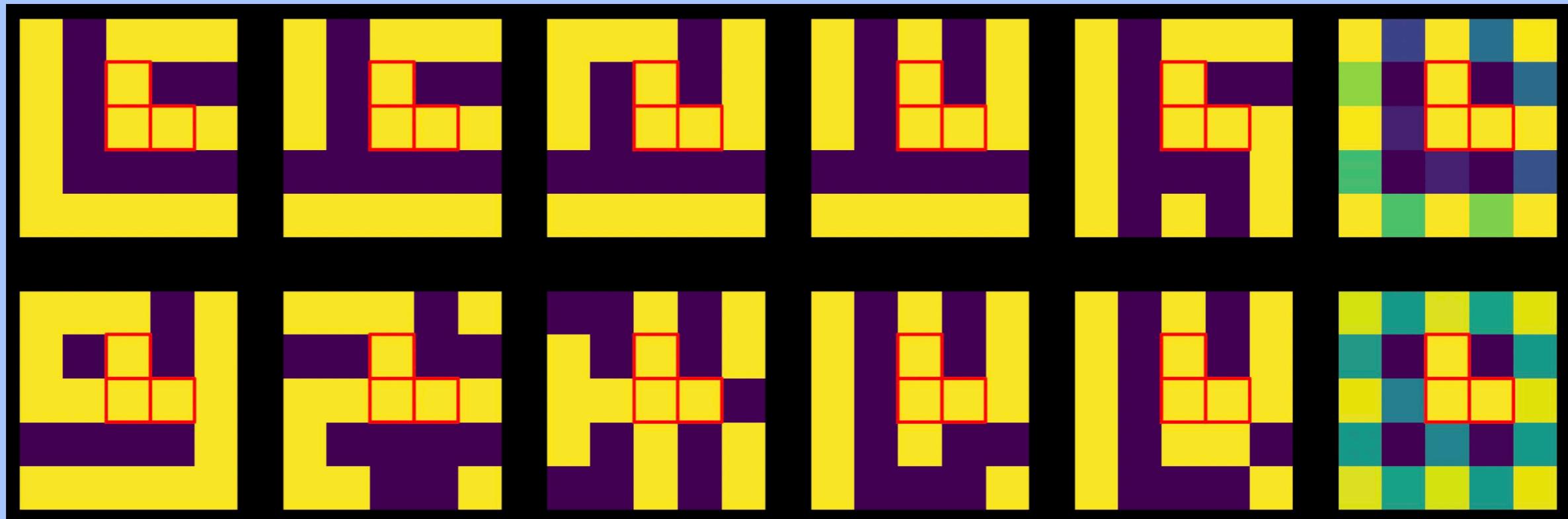


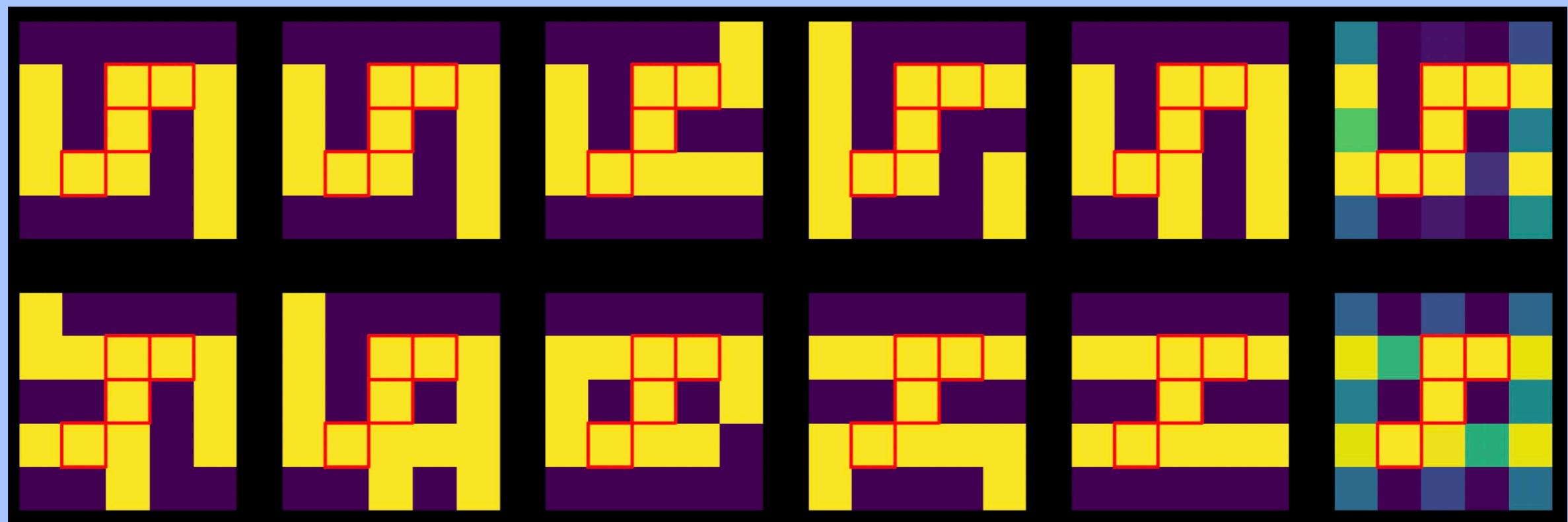
AFTER  $\sim 10^4$  BATCHES



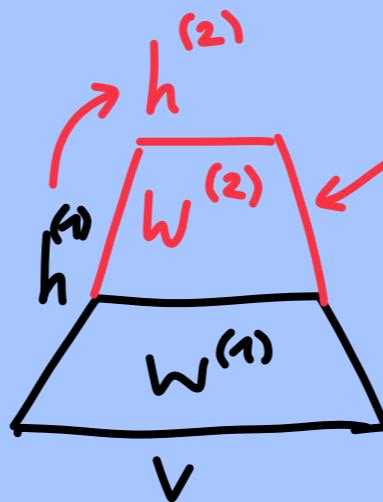
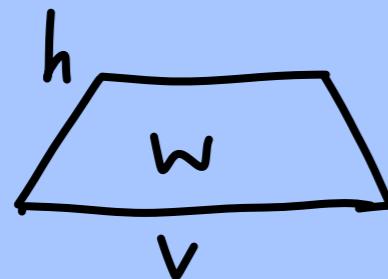
LONG-RANGE  
DEPENDENCIES  
STILL NOT CAPTURED  
ENTIRELY!

MORE CONSTRAINED!

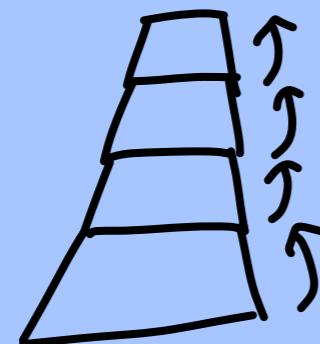




# DEEP RBM : "STACKING"



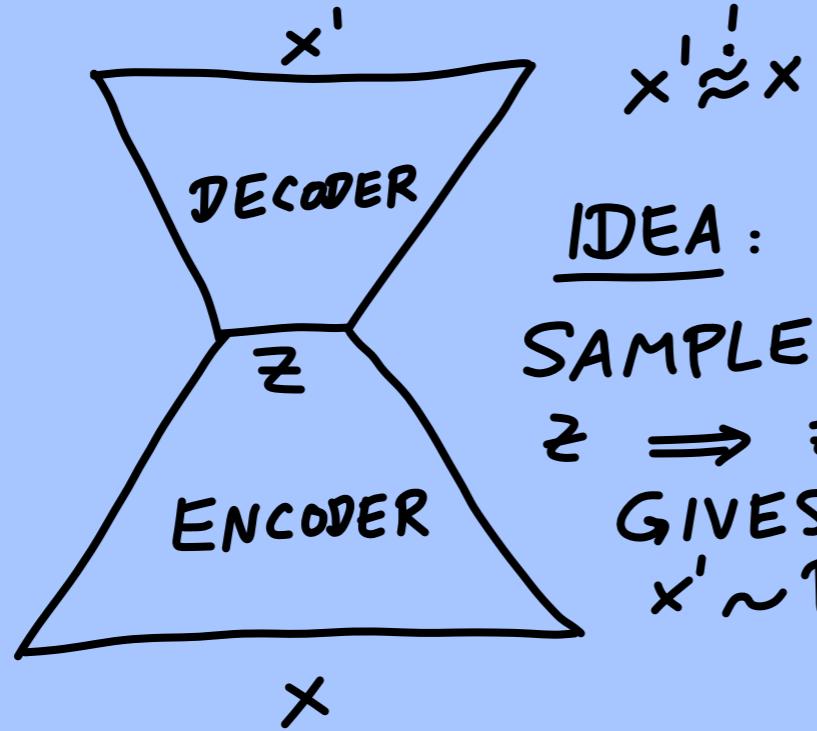
ONLY TRAIN  
THIS RBM  
WITH  $h^{(1)}$  (SAMPLED  
VIA  $v \rightarrow h^{(1)}$ ) TREATED  
AS 'VISIBLE' INPUT



7.4

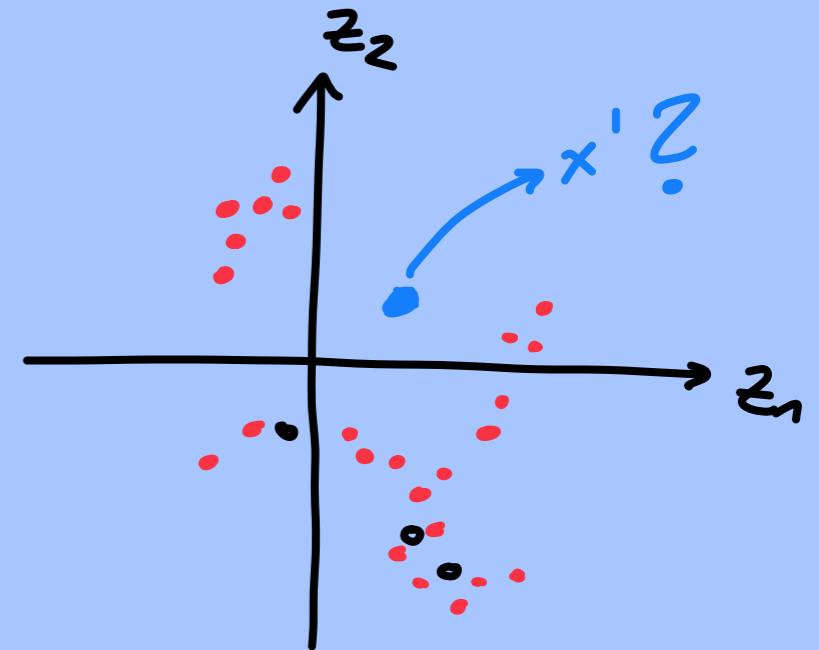
## VARIATIONAL AUTOENCODER

USUAL AUTOENCODER



IDEA:  
SAMPLE FROM  
 $z \Rightarrow z \rightarrow x'$   
GIVES SAMPLES  
 $x' \sim P(x) ??$

'LATENT SPACE'



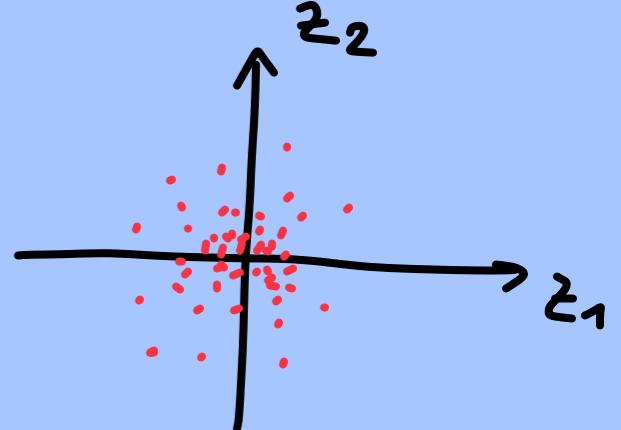
IMPOSSIBLE TO USE  
AS A GENERATIVE  
METHOD (TO SAMPLE  $x$ ),  
SINCE  $P_\theta(z)$   
IS UNKNOWN

IDEA: TRY TO ENFORCE (BESIDES  $x' \approx x$ ):

$$P_{\theta}(z) \approx P(z)$$

ENCODED  
FROM DATA  $x$

SIMPLE, FIXED  
(e.g. GAUSSIAN)

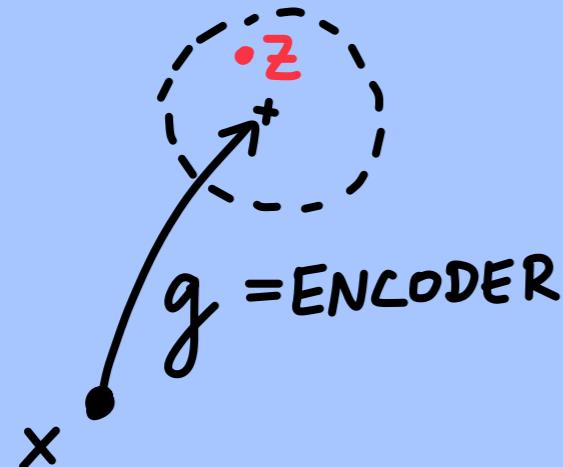


THUS ALSO

$$P_{\theta}(x') \approx \underbrace{P(x')}_{\text{DATASET}}$$

WHEN  
 $z$  SAMPLED  
ACCORDING TO  $P(z)$

IDEA : STOCHASTIC ENCODER & DECODER



$$z = g_{\phi}(x, \varepsilon) \Rightarrow z \sim q_{\phi}(z|x)$$

$\downarrow$

$\xrightarrow{\text{NOISE}}$

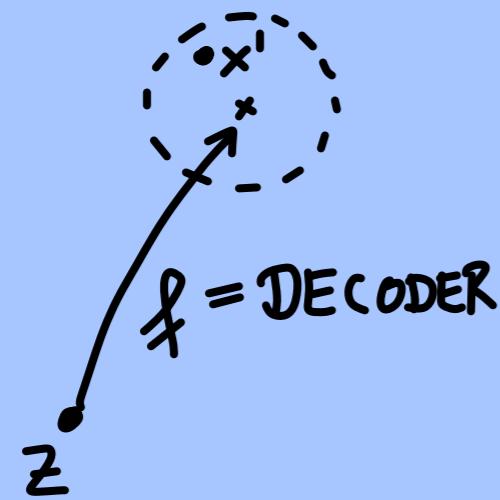
$\xrightarrow{\text{NN PARAMETERS}}$

$$= g_{\phi}^{(\mu)}(x) + g_{\phi}^{(\sigma)}(x) \odot \varepsilon$$

$\xleftarrow{\text{ELEMENTWISE}}$

"REPARAMETERIZATION TRICK"

$\xleftarrow{\text{NOISE (NORMAL GAUSSIAN)}}$



$$x' = f_{\theta}(z, \xi) \Rightarrow x' \sim p_{\theta}(x'|z)$$

$\downarrow$

$\xrightarrow{\text{NOISE}}$

$$= f_{\theta}(z) + \xi$$

$\xleftarrow{\text{NORMAL GAUSSIAN}}$

OPTIMIZE LOG-LIKELIHOOD OF  
MODEL UNDER DATA:

$$\left\langle \log P_{\theta}(x) \right\rangle_{\substack{x \sim p(x) \\ \text{DATA}}} \stackrel{!}{=} \max \quad \left. \begin{array}{l} \text{MEANS} \\ \mathcal{D}_{KL}(p \parallel p_{\theta}) \stackrel{!}{=} \min \end{array} \right\}$$

WITH CONSTRAINED  $p(z)$

$$\log p_{\theta}(x) = \underbrace{\left[ \int dz q_{\phi}(z|x) \right]}_1 \log p_{\theta}(x)$$

DATA

$$= \int dz q_{\phi}(z|x) \log \left\{ \frac{p_{\theta}(x,z)}{p_{\theta}(x,z)} p_{\theta}(x) \right\}$$

$\underbrace{\frac{p_{\theta}(z)p_{\theta}(x|z)}{p_{\theta}(z|x)}}$

$$= \left\langle \log \frac{p_{\theta}(z)}{p_{\theta}(z|x)} \right\rangle_{z \sim q_{\phi}(z|x)} + \left\langle \log p_{\theta}(x|z) \right\rangle_{z \sim q_{\phi}}$$

$D_{KL} (q_{\phi}(z|x) \| p_{\theta}(z|x)) \geq 0$  & WILL BECOME SMALL AFTER TRAINING

$$\begin{aligned} \text{MAX. } \mathcal{L} &\equiv \left\{ -D_{KL} (q_{\phi}(z|x) \| \underbrace{p_{\theta}(z)}_{=p(z)}) \right\} \text{ ANALYTICAL, SEE BELOW} \\ &\quad \left\{ + \left\langle \log p_{\theta}(x|z) \right\rangle_{z \sim q_{\phi}} \right\} \text{ FROM SAMPLING} \\ &\quad x \rightarrow z \rightarrow x' \\ &\quad \log p_{\theta}(x|z) \end{aligned}$$

$\mathcal{L}$  = "VARIATIONAL LOWER BOUND  
FOR  $\log p_{\theta}(x)$ "

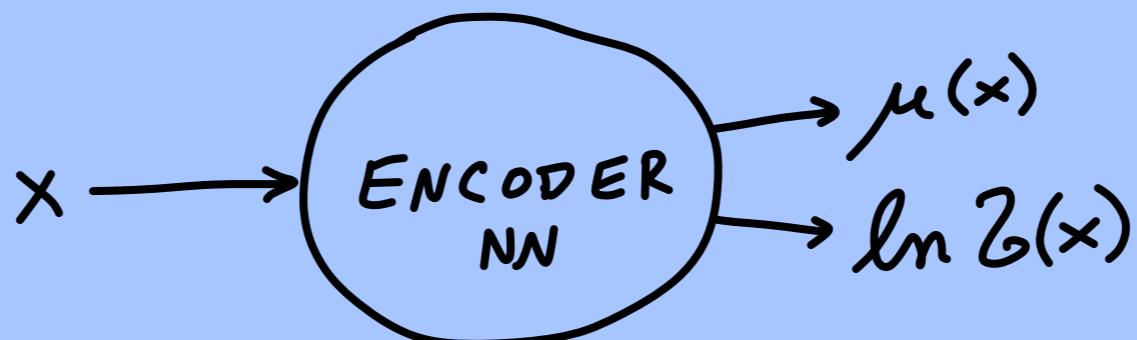
$\Rightarrow$  MAXIMIZE  $\mathcal{L}$

$$\ln p_{\theta}(x|z) = \text{const} - \frac{1}{2} \|x - \underbrace{f_{\theta}(z)}_{\text{DECODER NETWORK}}\|^2$$

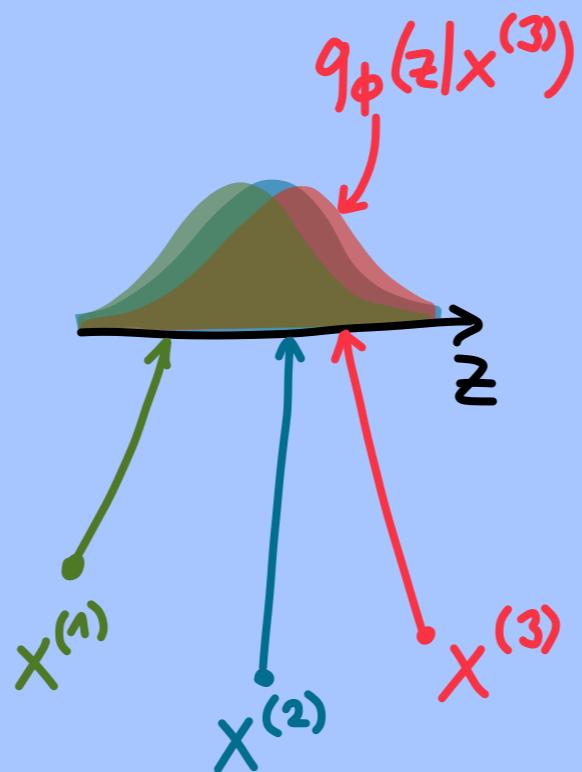
AND:

$$\begin{aligned} & D_{KL}(q_{\phi}(z|x) || p(z)) \\ &= -\frac{1}{2} \sum_{j \in \text{LATENT SPACE}} \left\{ 1 + \ln \tilde{\sigma}_j^2(x) - \mu_j^2(x) - \tilde{\sigma}_j^2(x) \right\} \end{aligned}$$

$\uparrow \quad \uparrow \quad \uparrow$   
 $p(z) = \text{NORMAL GAUSSIAN}$        $\text{PRODUCED BY ENCODER}$

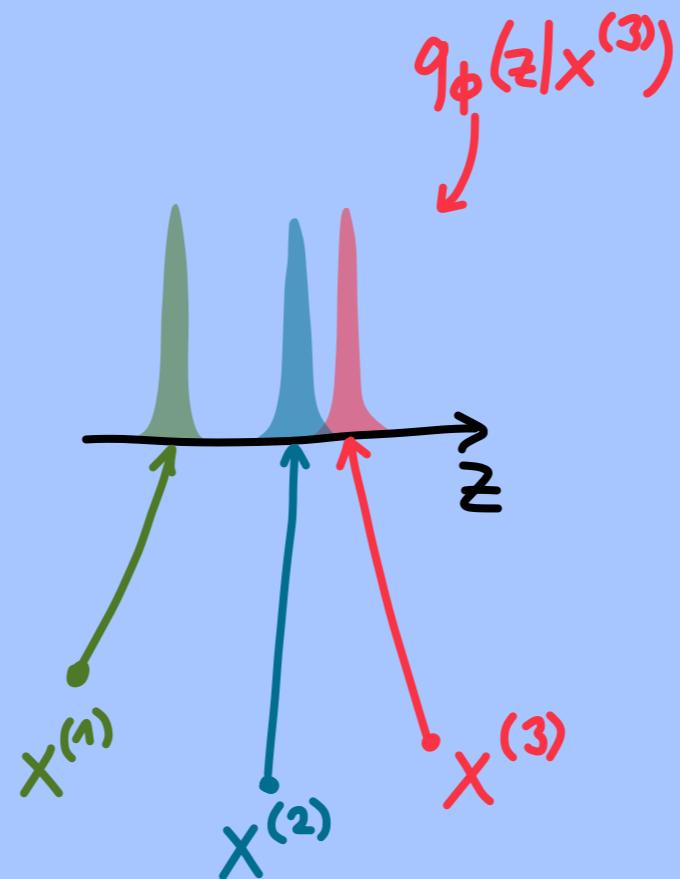


# VAE HEURISTICS



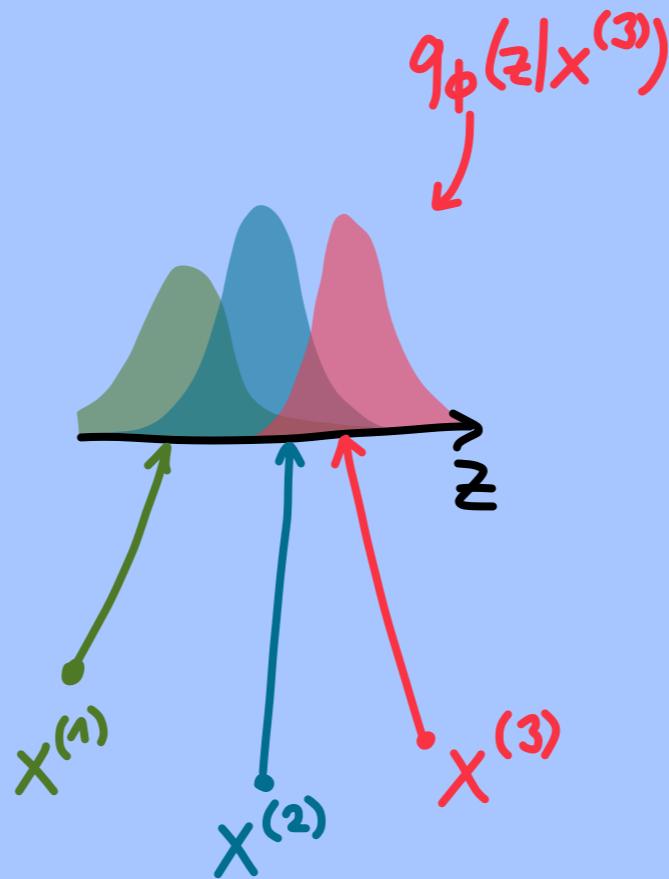
$q_\phi(z|x^{(\ell)}) \approx p(z) = \text{NORMAL GAUSSIAN}$   
BUT  $x^i$  &  $x$  ARE  
UNCORRELATED

# VAE HEURISTICS



$x' \approx x$   
BUT  $q_\phi(z|x)$  VERY  
DIFFERENT FROM  
 $p(z)$

# VAE HEURISTICS

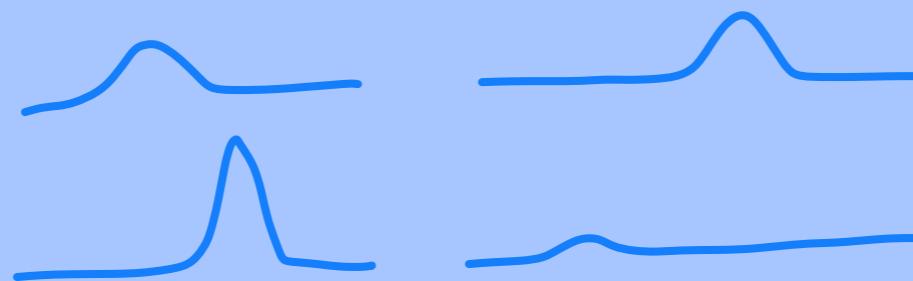
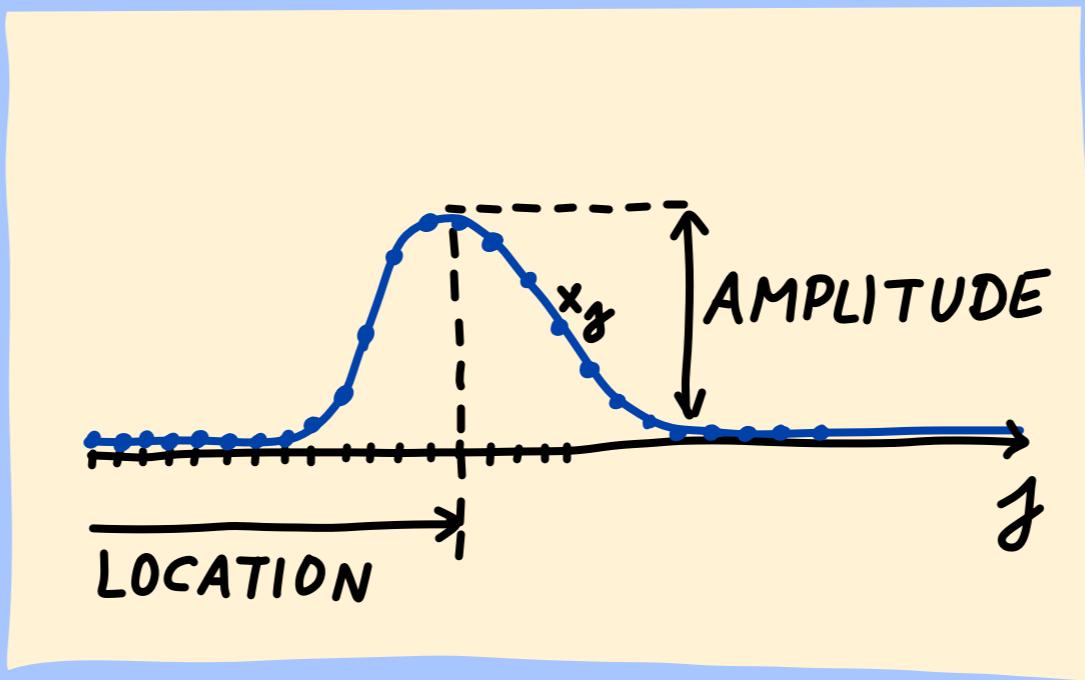


GOOD COMPROMISE :

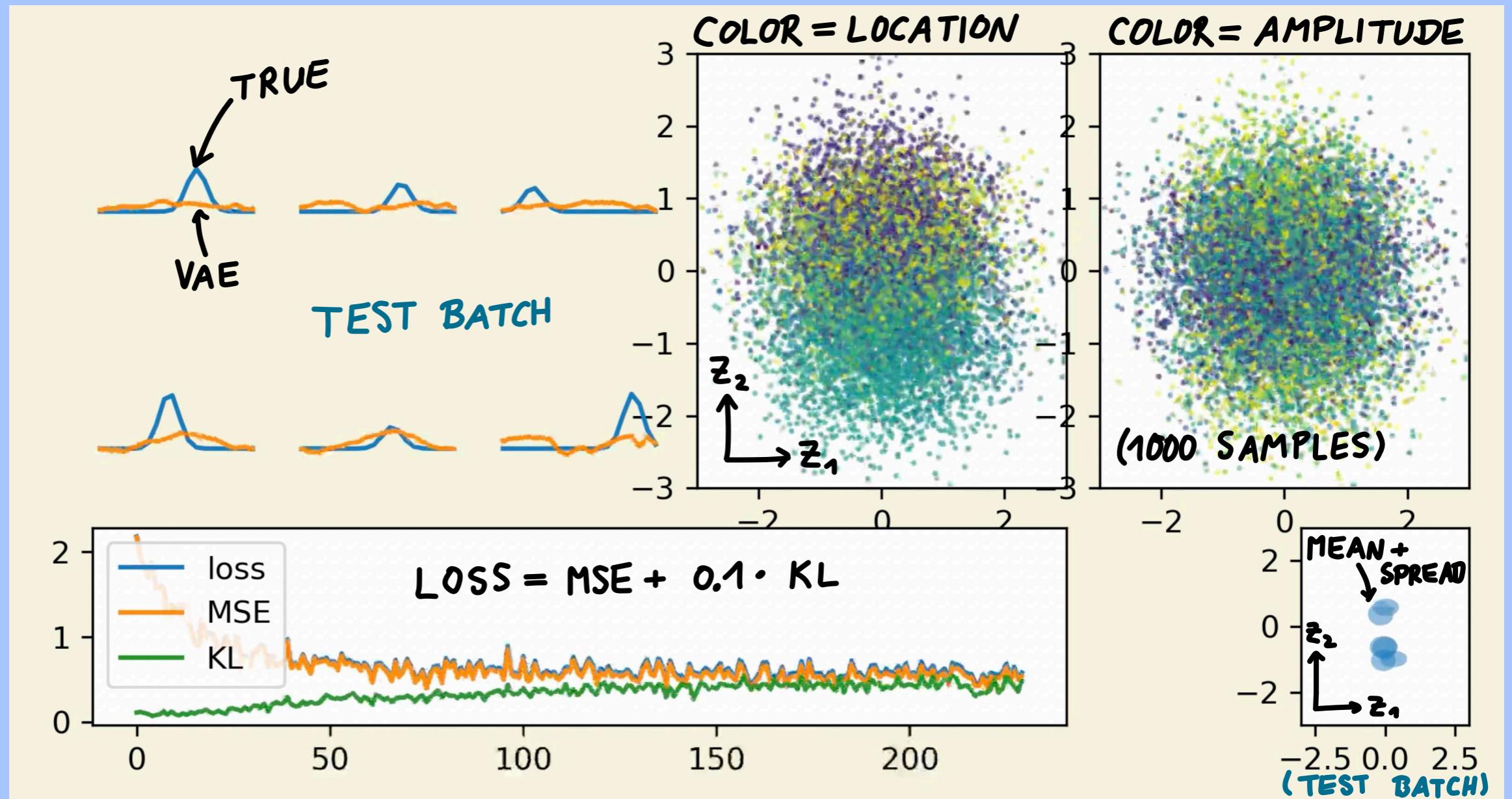
$$x' \approx x$$

AND  $q_\phi(z|x)$  'CLOSE' TO  $p(z)$

# VARIATIONAL AUTOENCODER TEST CASE: ENCODING A WAVE PACKET

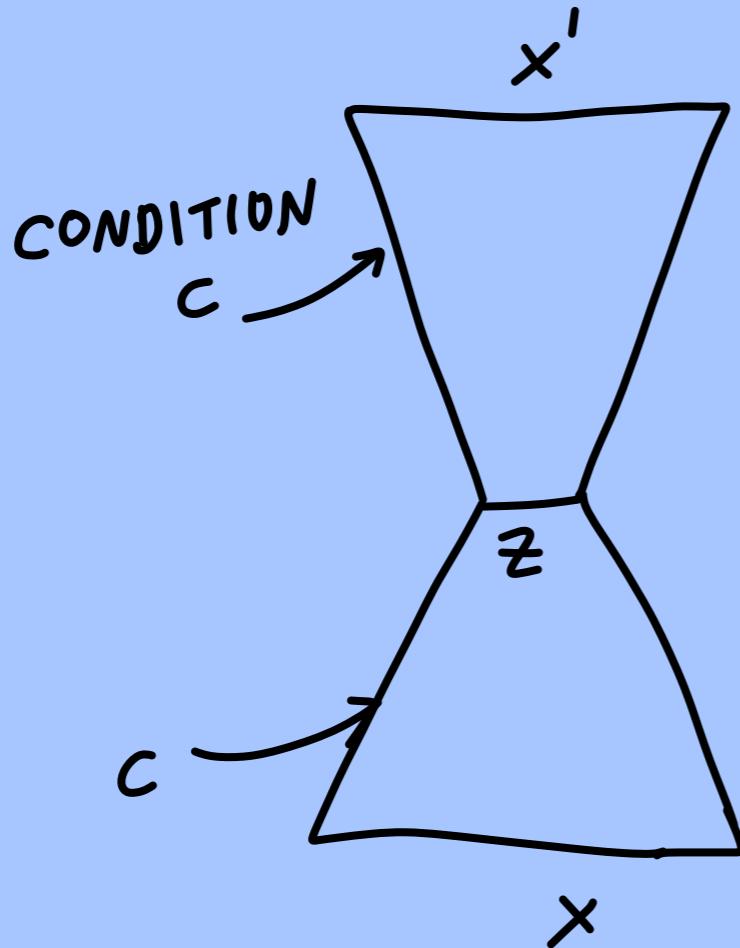


# VARIATIONAL AUTOENCODER TEST CASE: ENCODING A WAVE PACKET



CONDITIONAL

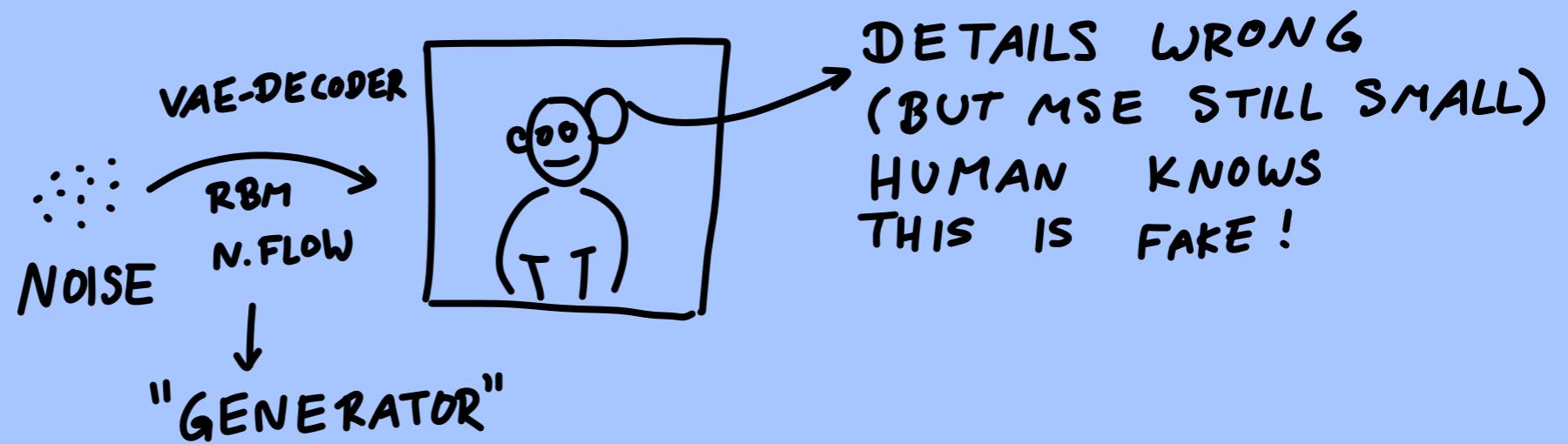
VAE :



$$z = g_{\phi}(x, \varepsilon; c)$$
$$x' = f_{\theta}(z; c) + \xi$$

⇒ MORE EFFICIENT  
THAN SEPARATE VAE  
FOR EACH  $c$

## 7.5

GENERATIVE ADVERSARIAL  
NETWORKS

IDEA: SECOND NETWORK, "DISCRIMINATOR"  
TRAINED TO TELL US WHETHER  
SAMPLE  $x$  IS REAL ( $\Rightarrow D(x) = 1$ )  
OR FAKE ( $D(x) = 0$ )

$$G(x) \leftrightarrow D(x)$$

GAN SETUP:

$z$  →  $x' = G(z)$   
NOISE FAKE SAMPLE

WANT:

$$D(x') \approx 0$$

$x$  REAL (FROM  
DATASET)

$$D(x) \approx 1$$

$$D(x) \in [0,1]$$

LOSS FUNCTION FOR DISCRIMINATOR:

$$\mathcal{L}_D = - \left\langle \ln D(x) \right\rangle_{\substack{x \sim P(x) \\ \text{DATA}}} - \left\langle \ln (1 - D(G(z))) \right\rangle_{\substack{z = \text{NOISE} \\ z \sim P(z)}} \quad x' \text{ FAKE}$$

LOSS FCT FOR GENERATOR:

$$\mathcal{L}_G = + \left\langle \ln (1 - D(G(z))) \right\rangle_{z = \text{NOISE}} \rightarrow \text{SMALL IF } D \text{ IS FOOLED!}$$

IMAGINE  $\mathcal{D}$  CONVERGES AT FIXED  $G$ :

LET  $p_G(x') = \text{DENSITY OF } x' = G(z)$   
 $= \int \delta(x' - G(z)) p_z(z) dz$

&  $p(x) = \text{DENSITY OF DATA } x$

$$\Rightarrow \mathcal{L}_D = - \int dx \left\{ p(x) \ln D(x) + p_G(x) \ln (1-D(x)) \right\}$$

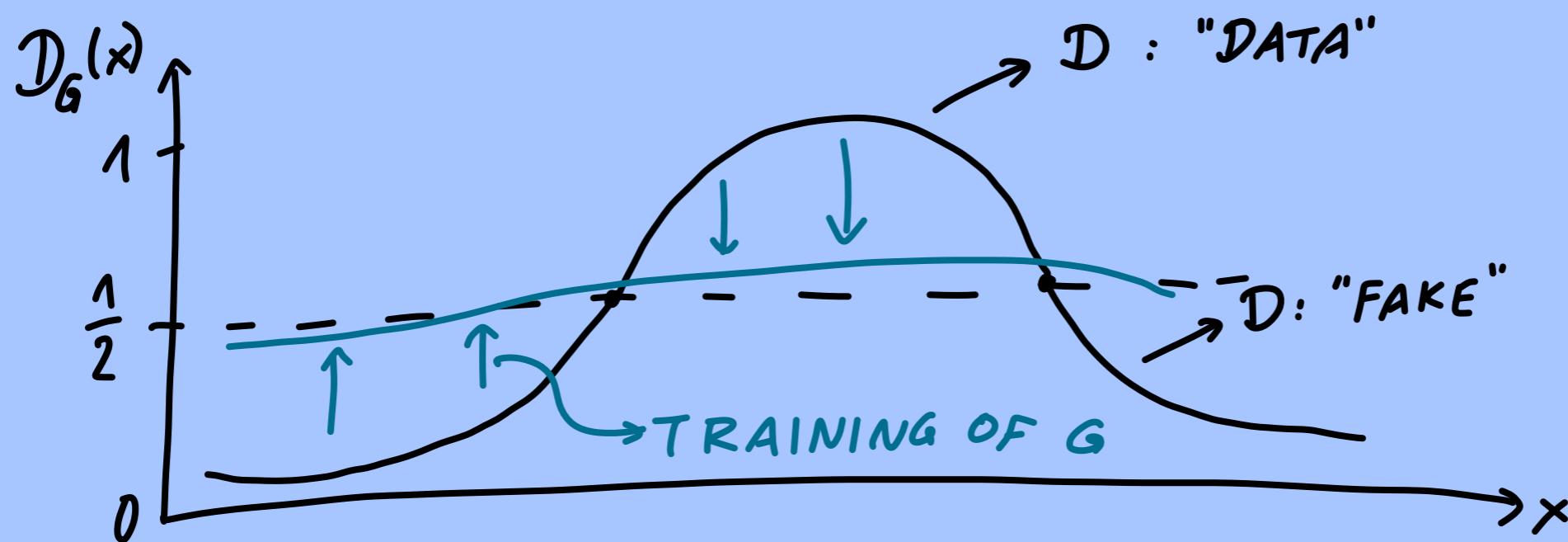
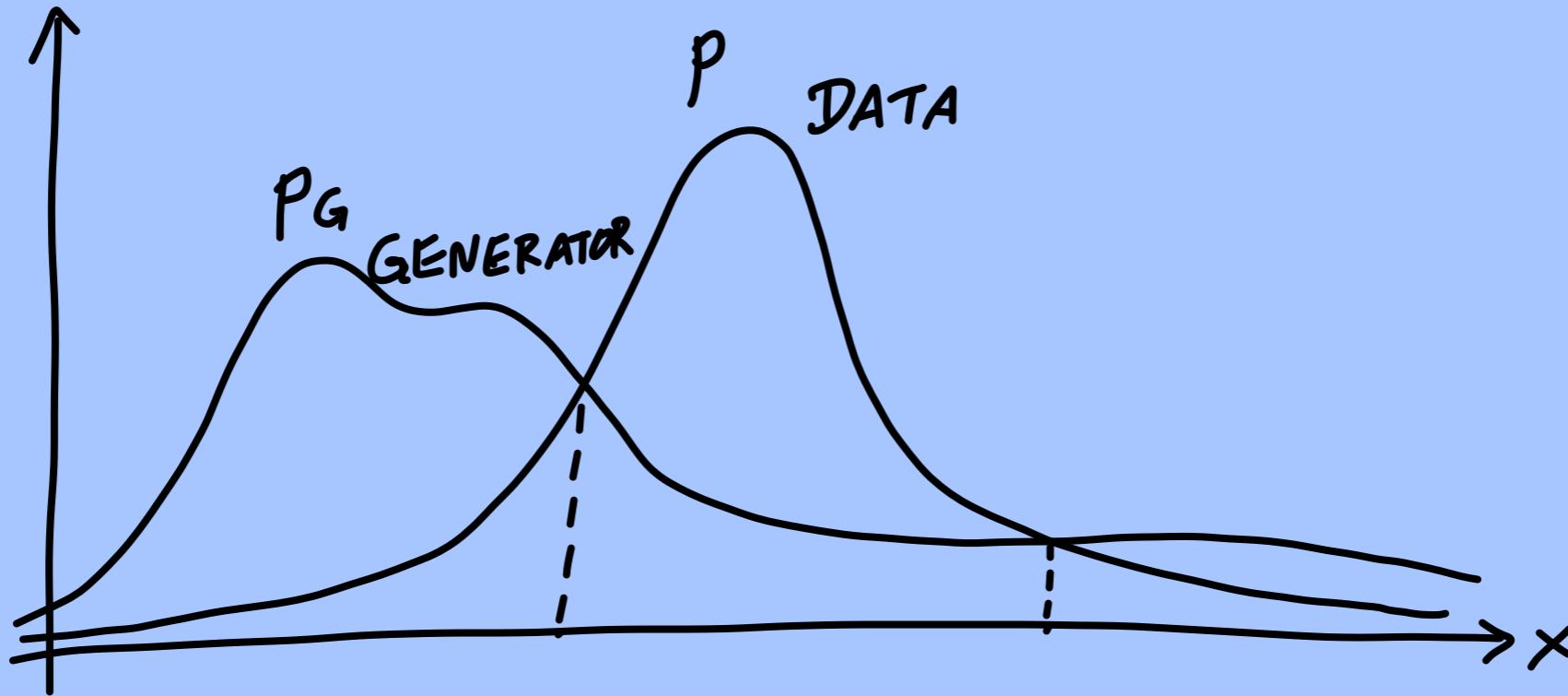
$\doteq \underset{\substack{! \\ \text{MAX}}}{} \text{ (vs. } D(x) \text{)}$

$$\frac{\partial}{\partial D(x)} \text{INTEGRAND} = 0$$

$$\frac{p(x)}{D(x)} - \frac{p_G(x)}{1-D(x)} = 0$$

$$\Leftrightarrow D(x) = \frac{p(x)}{p(x) + p_G(x)} \equiv D_G(x)$$

$\xrightarrow[\substack{\text{AT} \\ \text{FIXED } G}]{} \quad$



NOW: MINIMIZE  $\mathcal{L}_G$ ,  
 ASSUMING ALWAYS  $D \equiv D_G$

$$\mathcal{L}_G \left|_{D=D_G} \right. = \int dx p_G(x) \ln \left( \frac{p_G(x)}{p_G(x) + p(x)} \right) \stackrel{!}{=} \text{MIN}$$

→ FIND  $p_G(x)$  THAT MINIMIZES THIS!  
 WITH CONSTRAINT  $\int dx p_G(x) = 1$

$$\int dx p_G(x) \ln ( ) - \lambda \int dx p_G(x) \stackrel{!}{=} \text{MIN}$$

$\uparrow$   
LAGR. MULTIPLIER

$$\frac{\delta}{\delta p_G(x)} ( " ) \stackrel{!}{=} 0 \quad \frac{d}{dp_G(x)} \text{ INTEGRAND} = 0$$

$$\Rightarrow \dots \Rightarrow \text{SOLUTION: } \frac{p_g(x)}{p(x)} = \text{const}(\lambda)$$

& NORM.  $\Rightarrow \text{const} = 1 \Rightarrow p_g(x) = p(x)$

✓

FROM GENERATOR      DATA

---

IF  $D \approx D_g$  THEN  $G$  CONVERGES  
SUCH THAT

$$p_a(x) = p(x)$$

$$D(x) = \frac{1}{2}$$

IN PRACTICE:

[  
SAMPLE  $z \Rightarrow x' = G(z)$   
SAMPLE  $x$  FROM DATA  
GRAD. DESCENT FOR  $D$  ON  $\mathcal{L}_D$   
  
[  
SAMPLE  $z \Rightarrow x' = G(z)$   
GRAD. DESC. FOR  $G$  ON  $\mathcal{L}_G$



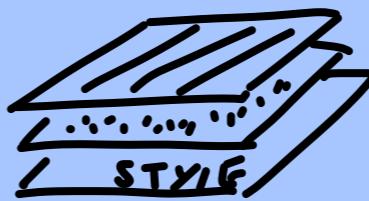
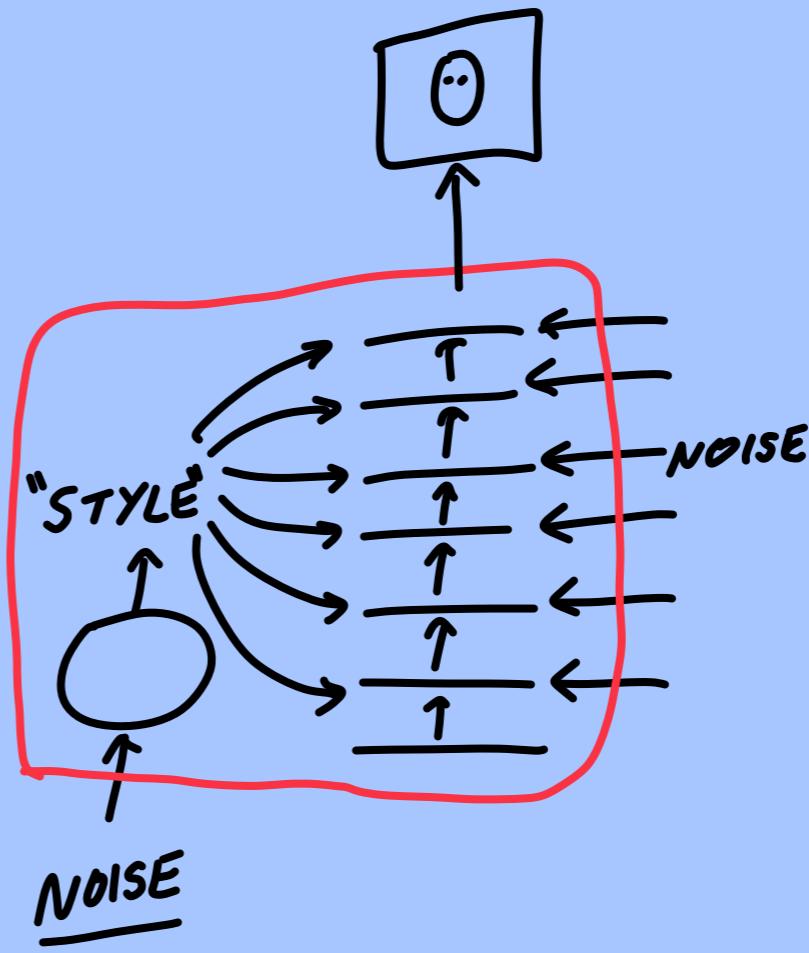
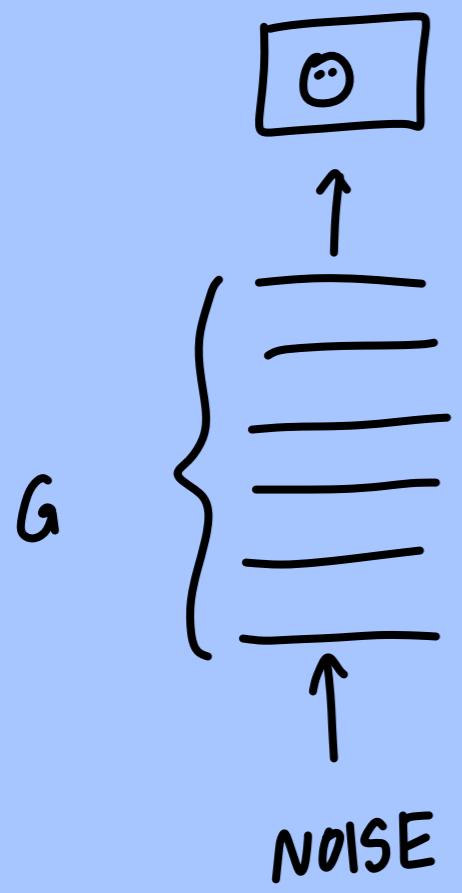
FROM: GOODFELLOW ET AL,  
"GENERATIVE ADVERSARIAL  
NETWORKS"  
arXiv : 1406.2661  
NeurIPS PROCEEDINGS 2014



FROM: GOODFELLOW ET AL,  
"GENERATIVE ADVERSARIAL  
NETWORKS"  
arXiv: 1406.2661  
NeurIPS PROCEEDINGS 2014



FROM: KARRAS ET AL,  
"A STYLE-BASED GENERATOR  
ARCHITECTURE FOR GENERATIVE  
ADVERSARIAL NETWORKS"  
arXiv: 1812.04948  
CVPR 2019



CONDITIONAL GAN

$$G(z; c)$$

CONDITION

LABEL

BLACK&WHITE

$D(z; c) \approx 1 \hat{=} \text{REAL SAMPLE}$   
 $\text{FOR THIS CONDITION}$

$$\left[ \begin{array}{l} \dots \\ \text{loss\_D} = -\ln D \dots \end{array} \right]$$

grads=tape.gradients(loss\_D, D.trainable\_variables)

$$\left[ \begin{array}{l} \dots \\ \text{loss\_G} = \dots \\ \dots \\ \text{loss_G, G.trainable\_variables} \end{array} \right]$$

## 7.6

# COMPARISON: LEARNING PROBABILITY DISTRIBUTIONS

### BOLTZMANN MACHINE

- DISCRETE (TYPICALLY)
- NEED MARKOV CHAIN
- DEEP VIA STACKING
- SIMPLE TO CONDITION WITHOUT EXTRA TRAINING
- NICE PHYSICS CONNECTION
- ACCESS TO  $p(x)$  ONLY UP TO NORM.

### NORMALIZING FLOWS WITH INVERTIBLE NN

- CONTINUOUS
- BOTH SAMPLING & ACCESS TO  $p(x)$
- $\Rightarrow$  ESTIMATES FOR KL-DIV. ETC.
- DEEP ✓
- CONDITION WITH TRAINING

### VARIATIONAL AUTOENCODER

- CONNECTION TO AUTOENCODERS
- LOWER-DIM. LATENT Z
- DEEP ✓
- ACCESS TO  $p(x)$  NOT AVAILABLE
- CONDITION WITH TRAINING

### GENERATIVE ADVERSARIAL NETWORK

- FOCUS ON GETTING DETAILS RIGHT
- ACCESS TO  $p(x)$  NOT AVAILABLE
- CONDITION WITH TRAINING
- DEEP ✓
- FLEXIBLE

# 8.

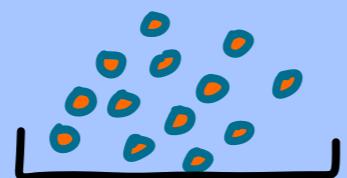
## ADVANCED NETWORK ARCHITECTURES

→ HANDLING/EXPLOITING  
SPECIAL STRUCTURE IN THE DATA

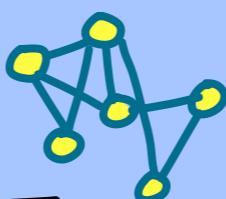
THIS IS A LONG TEXT. WE ...



LONG-RANGE DEPENDENCIES  
IN SEQUENCES



VARIABLE-SIZE INPUT  
PERMUTATION INVARIANCE  
...



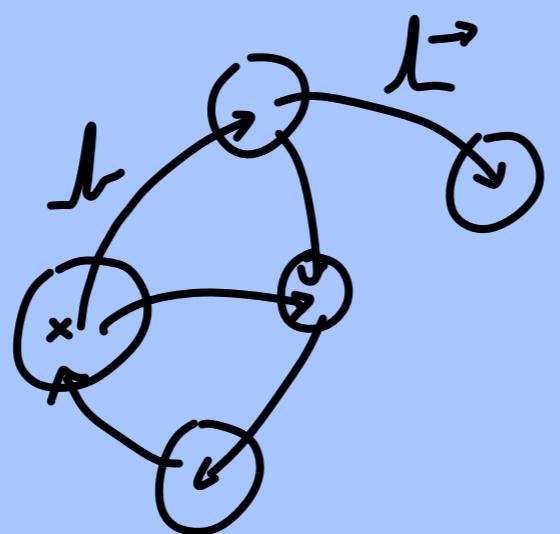
8.1

# RECURRENT NEURAL NETWORKS

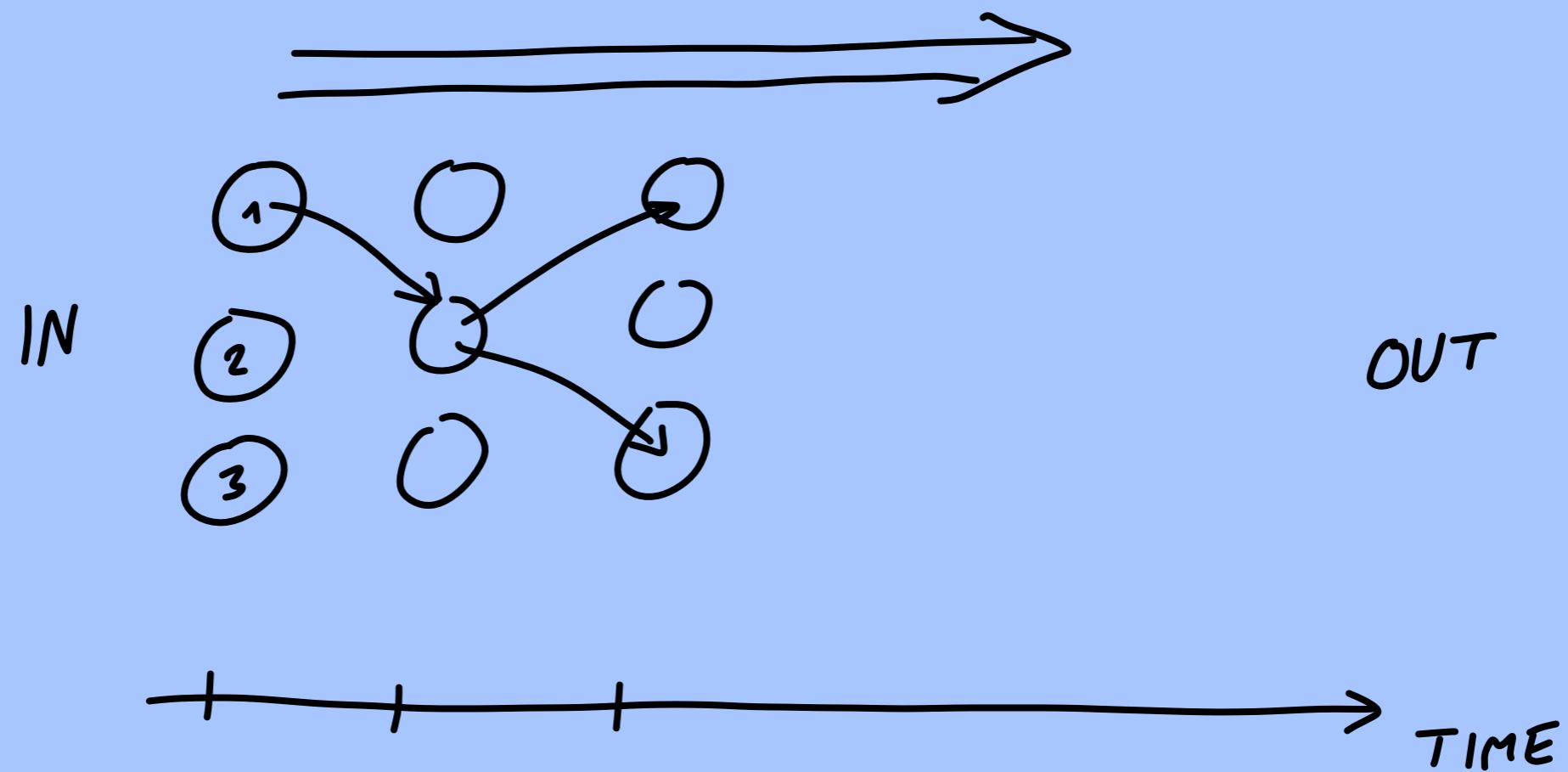
(NETWORKS WITH MEMORY)

GOAL: - PREDICT/INTERPRET  
SEQUENCES  
WITH TRANSLATIONAL INV. IN TIME  
& CAUSALITY ( $t \rightarrow t'$  INFLUENCE  
ONLY FOR  $t' > t$ )

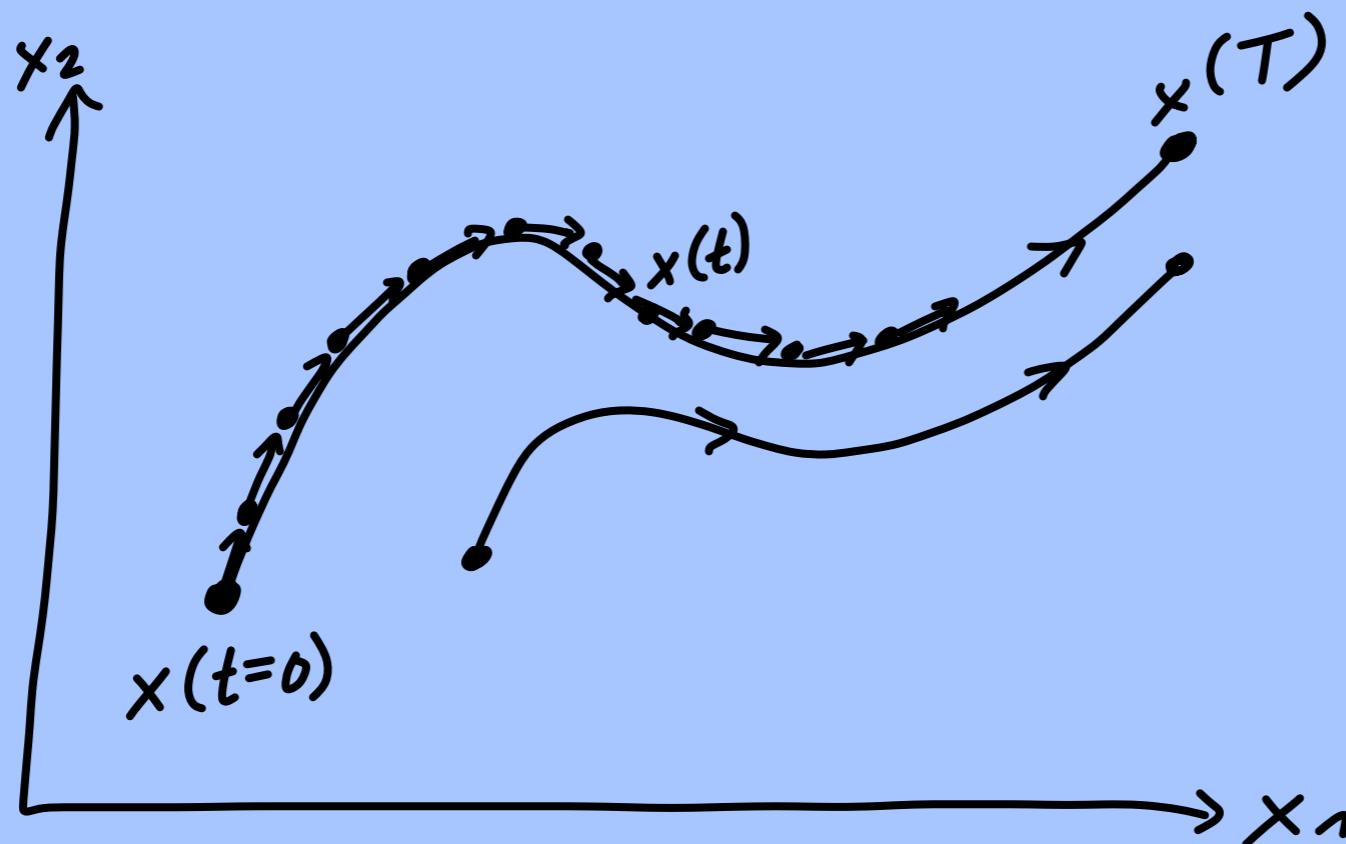
SPEECH (AUDIO),  
SENTENCES, DNA SEQUENCES,  
PHYSICAL SIGNAL,...



$$\frac{\text{OUT}}{1} \quad \frac{1}{\text{IN}}$$



INTRODUCTION VIA PHYSICS:  
TIME DYNAMICS  
MODELING SOLUTION  
OF DIFF. Eqs. VIA NN?



$$\frac{dx(t)}{dt} = f(x)$$

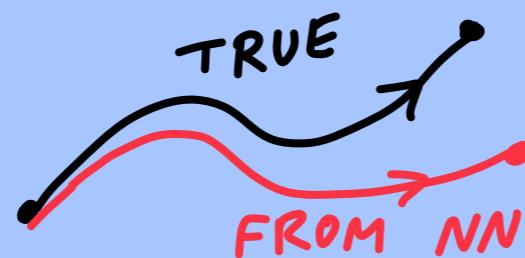
EULER:  $x(t+\Delta t) \approx x(t) + \Delta t f(x(t))$

# MANY OPTIONS FOR NN APPLICATION

① LEARN R.H.S.

$$f_{\theta}(x) \xrightarrow{\text{NN}} f(x) \approx \overbrace{f(x)}^{\text{R.H.S.}}$$

BUT:  $\dot{x} = f_{\theta}(x)$  MAY LEAD TO  
LARGE DEVIATIONS  
AT LONG TIMES!



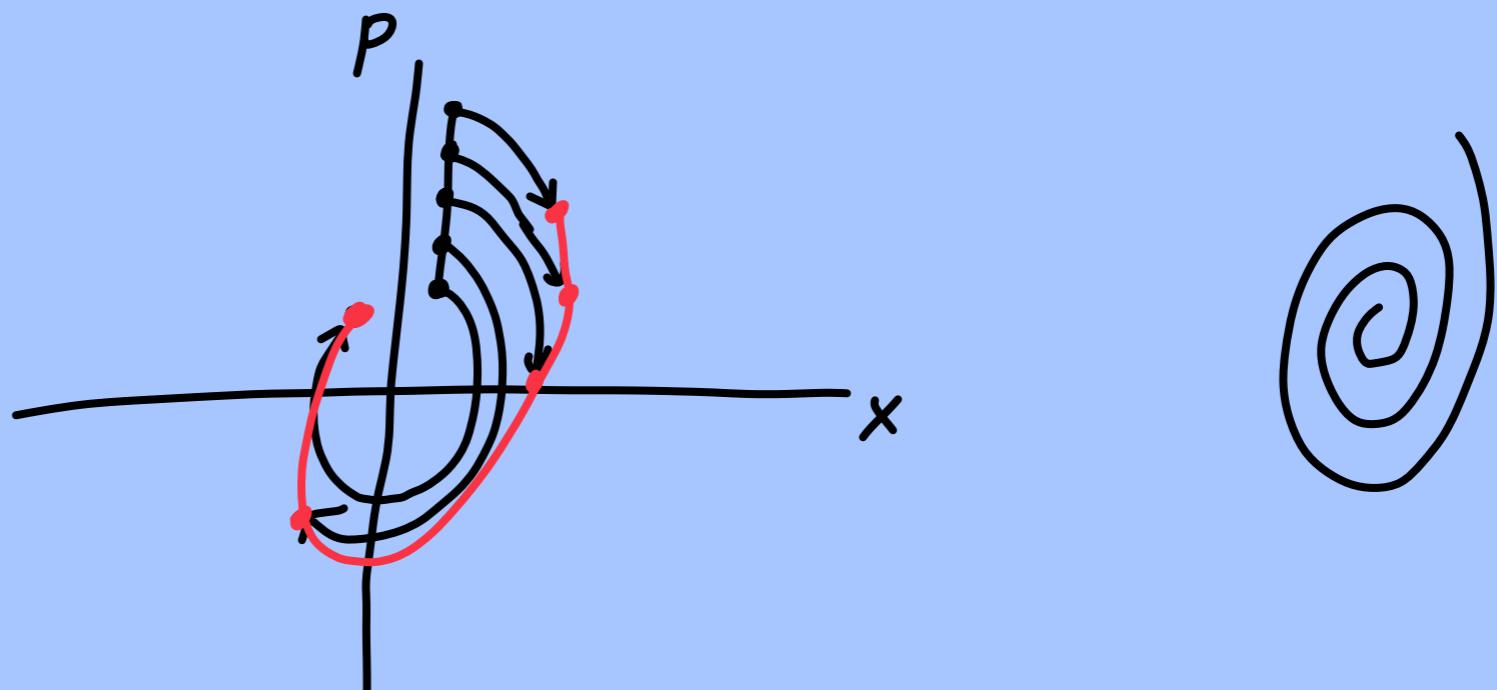
②

LEARN DIRECT MAPPING

$$x(0) \rightarrow x(T)$$

$$F_{\theta}(x(0)) \stackrel{!}{\approx} \underbrace{x(T)}_{\text{FROM DIFF.Eq FOR } x(0)}$$

BUT: CAN BE VERY COMPLICATED!



③ LEARN APPROX. EVOLUTION AS  
SEQUENCE

$$N \text{ STEPS}$$

$$x^{(n+1)} = x^{(n)} + \underbrace{f_{\theta}(x^{(n)})}_{\text{SAME NET FOR EACH STEP}}$$

$$n=0, 1, \dots, N-1$$

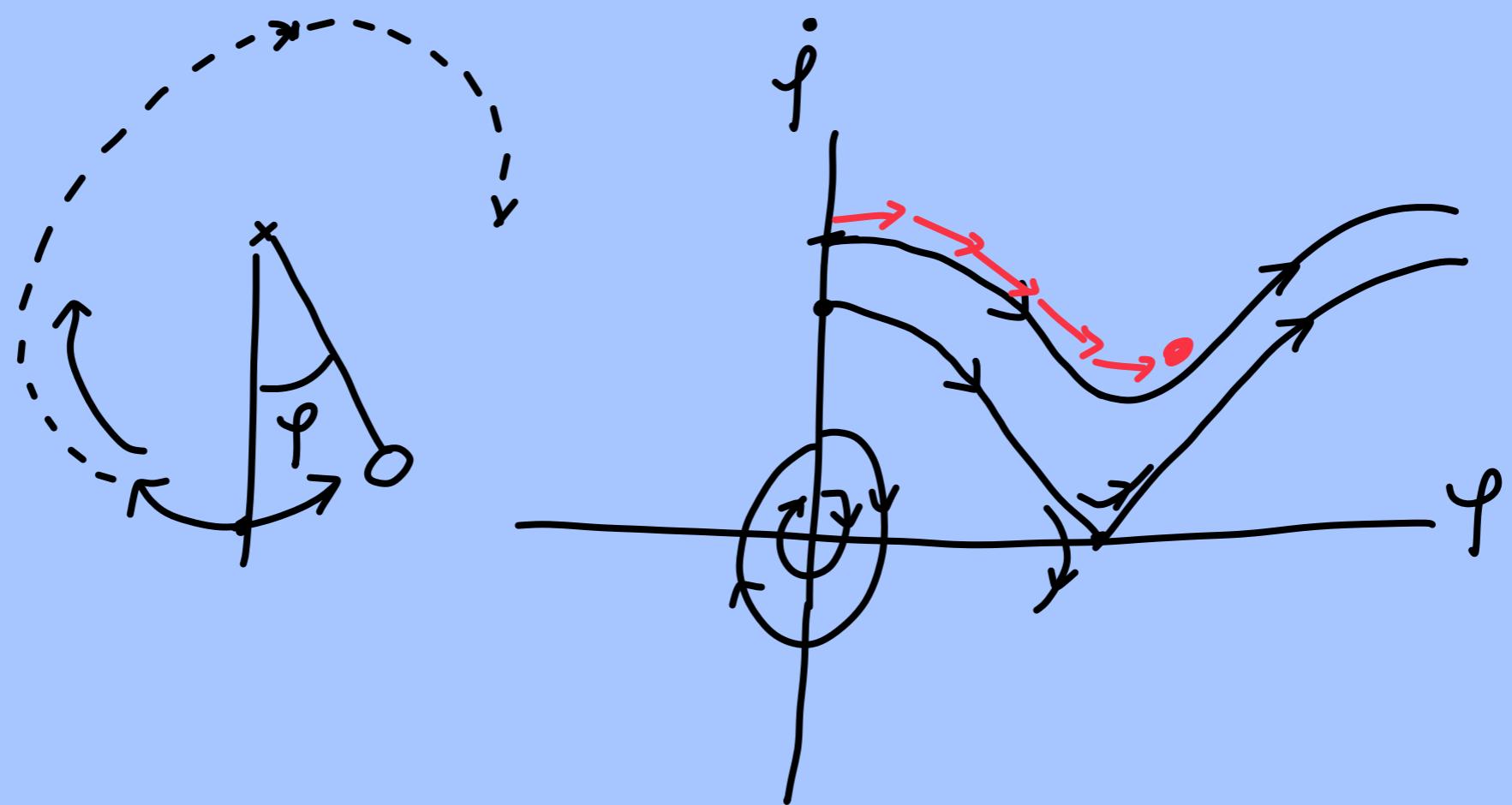
→ CAN BE MANY  
FEWER STEPS

THAN NEEDED FOR DIFF. EQU. EULER

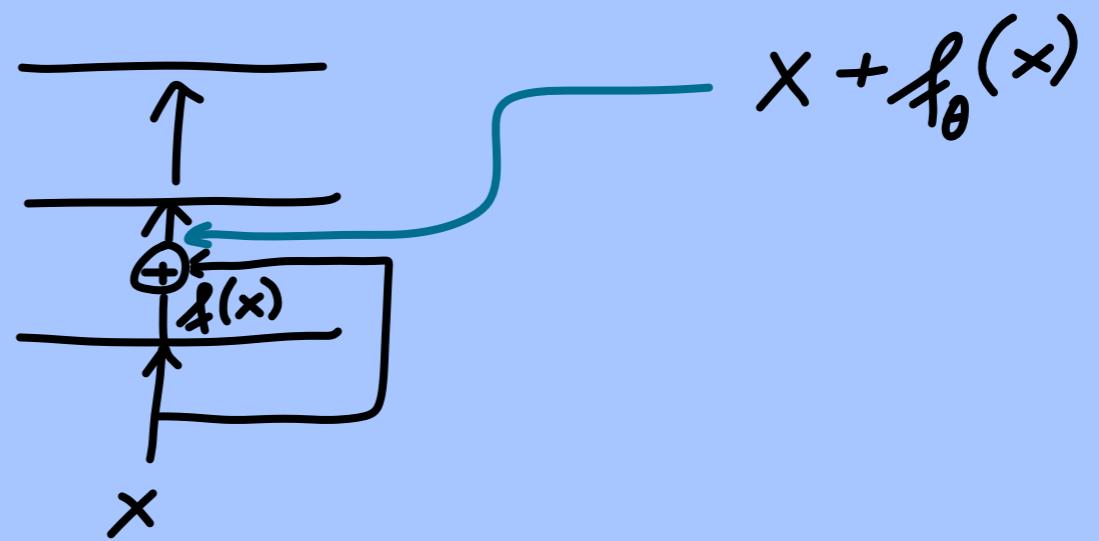
LOSS:

$$\mathcal{L} = \left\langle \left\| \underbrace{x^{(N)}}_{\text{FROM NN}} - \underbrace{x(T)}_{\text{FROM DIFF. Eq.}} \right\|^2 \right\rangle_{x(0) \sim p(x(0))}$$

ONLY COMPARE AT END

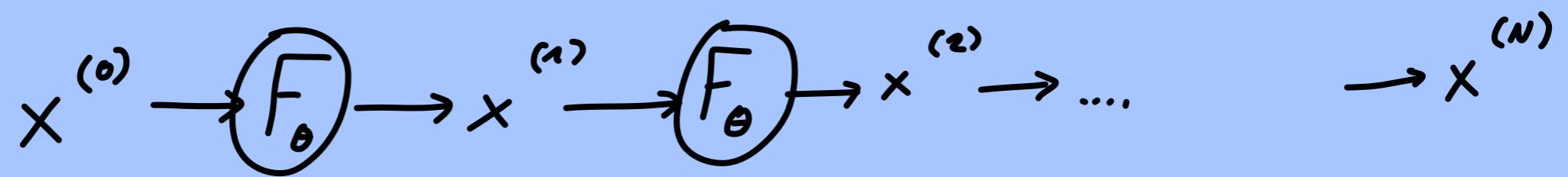


"RESIDUAL"  
NETWORK



$\Rightarrow$  GENERAL STRUCTURE:

$$x^{(t+1)} = F_{\theta}(x^{(t)}) \quad t=0, 1, 2, \dots$$

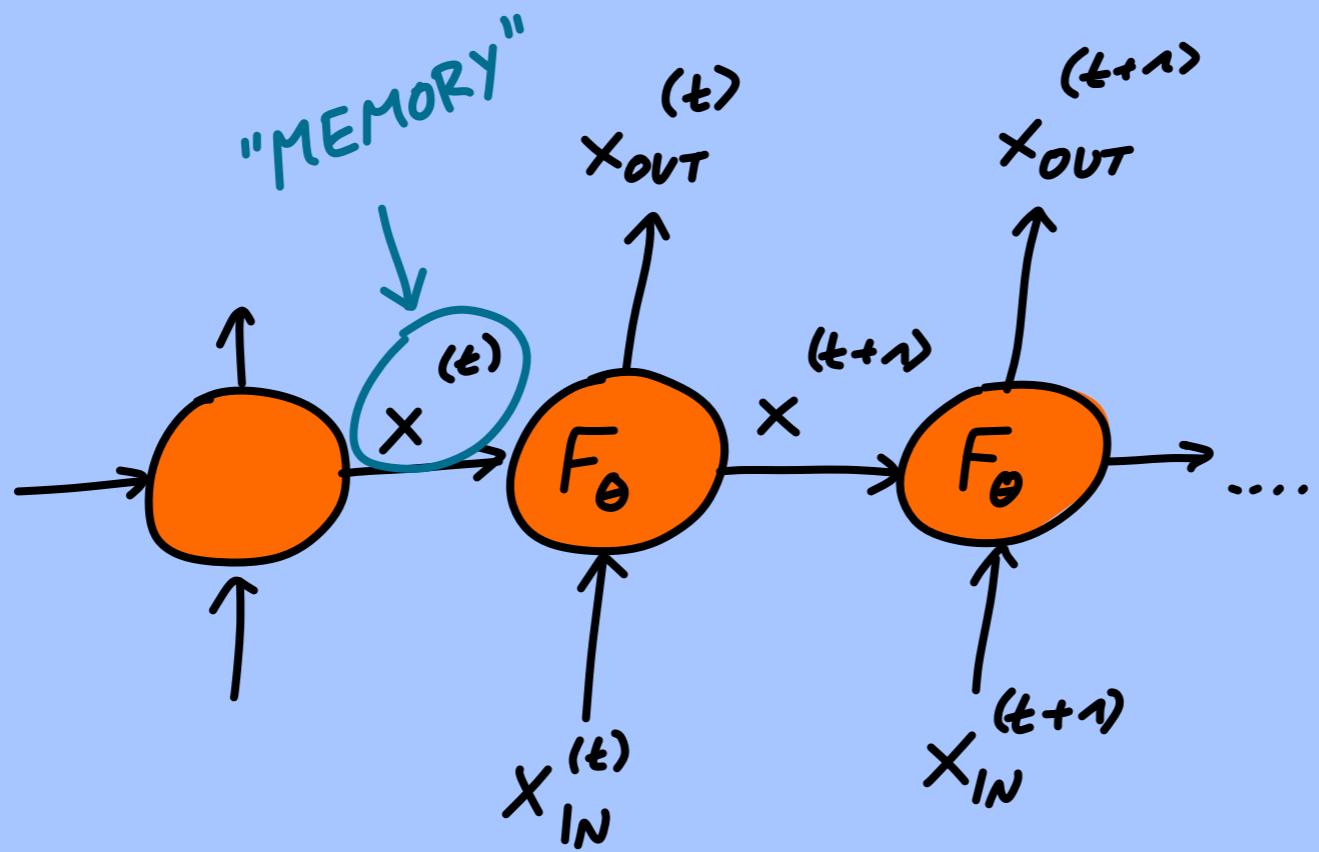


$$x^{(N)} = F_{\theta}(x^{(N-1)}) = F_{\theta}(F_{\theta}(x^{(N-2)})) = \dots$$

MORE GENERAL:

- WITH INPUT AT  $t \in x_{IN}^{(t)}$   
(PHYSICS: DRIVING,  
LANGUAGE: LETTER/WORD)
- WITH OUTPUT AT  $t \in x_{OUT}^{(t)}$

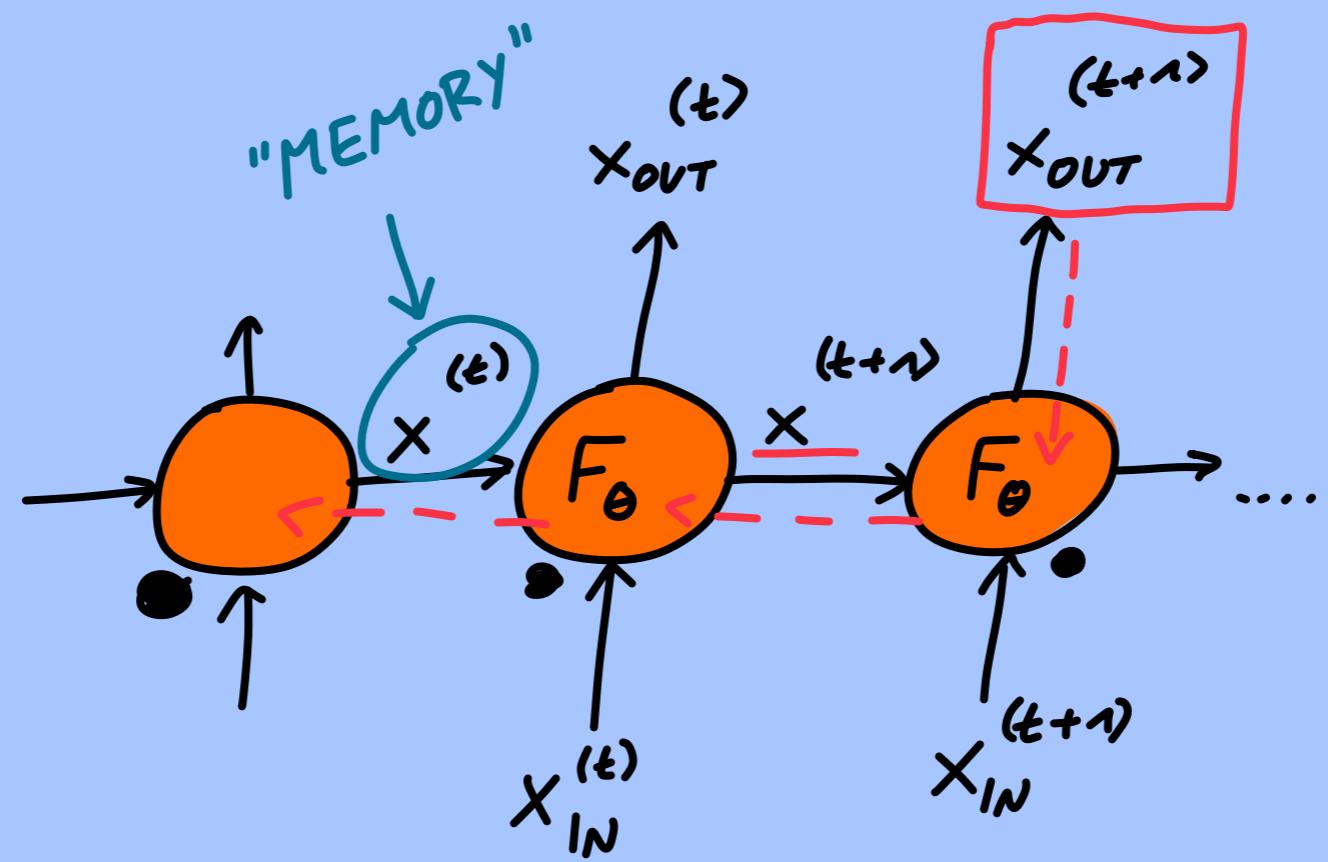
$$(x^{(t+1)}, x_{\text{OUT}}^{(t)}) = F_{\theta}(x^{(t)}, x_{\text{IN}}^{(t)})$$



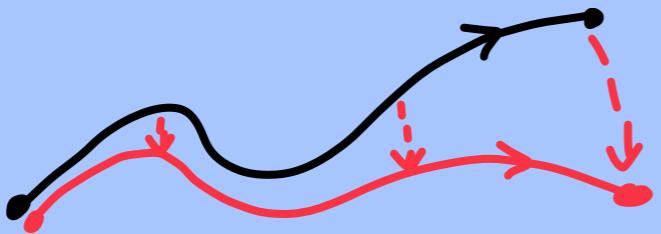
GENERAL  
RECURRENT  
NEURAL  
NETWORK

LOSS CAN DEPEND ON  $x_{out}^{(t)}$  AT EACH TIME  $t$

$$\mathcal{L} = \left\langle \sum_t \| x_{out}^{(t)} - x_{TARGET}^{(t)} \|^2 \right\rangle_{x_{in} \in \text{DATA}}$$



# CHALLENGE: "EXPLODING / VANISHING GRADIENTS" SENSITIVITY



$$\left. \frac{\partial F_\theta}{\partial \theta} \right|_{x=x^{(t)}} \equiv J_t^{(\theta)} \begin{matrix} \\ (\text{MATRIX}) \\ \dim(x) \cdot \dim(\theta) \end{matrix}$$

$$\left. \frac{\partial F_\theta}{\partial x} \right|_{x=x^{(t)}} \equiv J_t \begin{matrix} \\ \dim(x) \cdot \dim(x) \end{matrix}$$

$$\begin{aligned} \frac{\partial x^{(T)}(x^{(0)})}{\partial \theta} &= J_{T-1}^{(\theta)} + J_{T-1} \cdot J_{T-2}^{(\theta)} + \dots + \\ &\quad + J_{T-1} \cdot J_{T-2} \cdot \dots \cdot J_1 \cdot J_0^{(\theta)} \end{aligned}$$

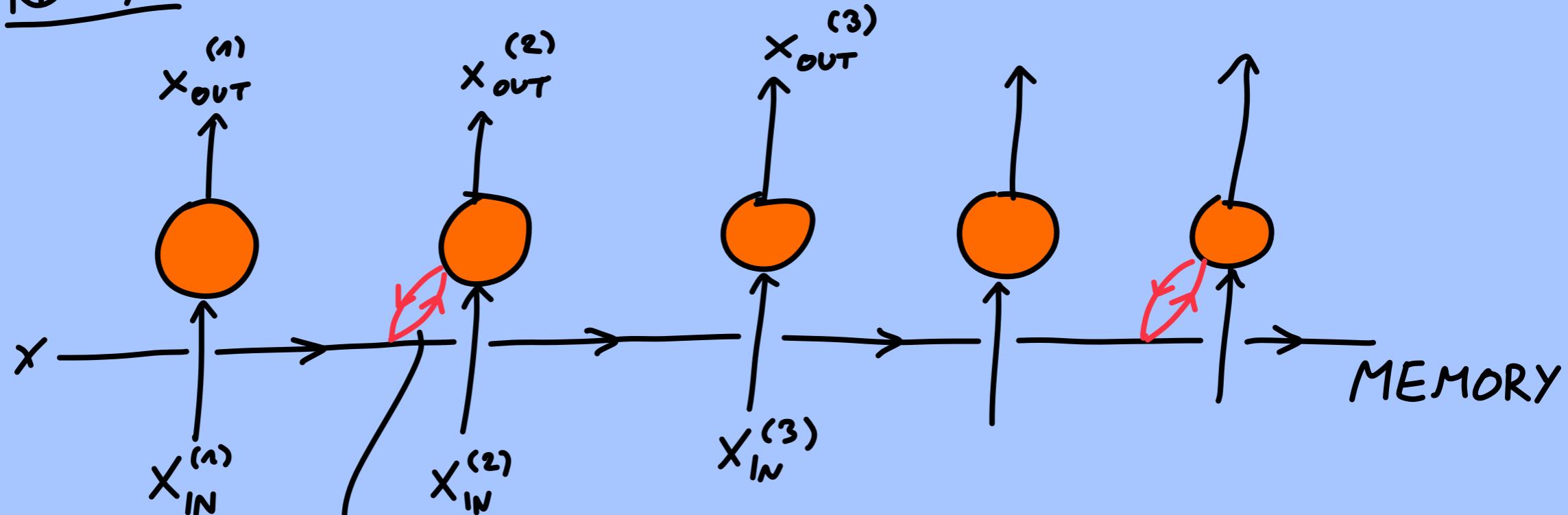
... EXPLODE WITH  $T \uparrow$

$\Rightarrow$  TINY CHANGES  
IN  $\theta$  HAVE LARGE EFFECT ON  $x^{(T)}$   
 $x_{\text{OUT}}^{(T)}$

VANISH WITH  $T \uparrow$

$\Rightarrow$  NO "LEARNING SIGNAL"  
FROM EARLY IN TIME SEQUENCE

IDEA TO CIRCUMVENT THIS PROBLEM:



RARE MEMORY ACCESS!

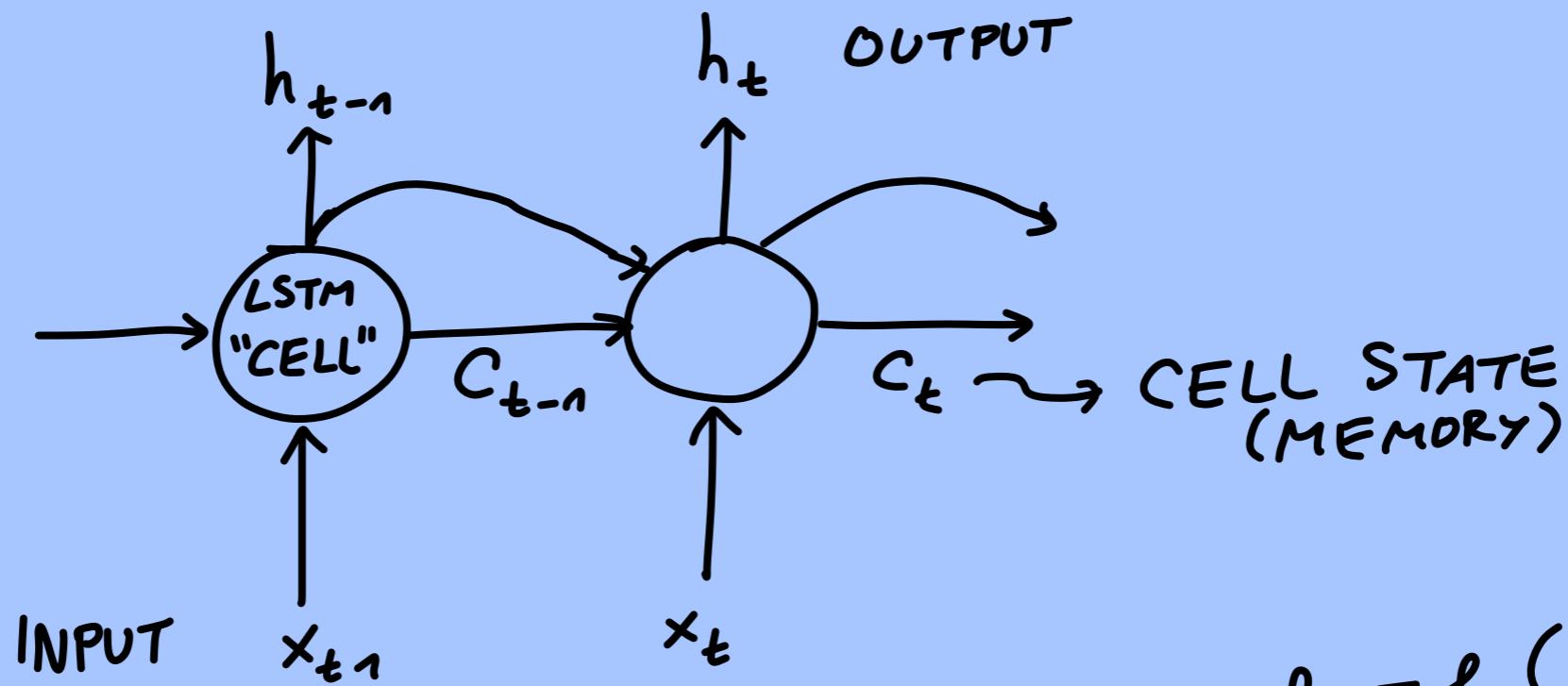
FOR OTHER

TIME STEPS

$$\frac{\partial x^{(t+1)}}{\partial x^{(t)}} = 1$$
$$x^{(t+1)} = x$$

# FIRST EXAMPLE: "LONG SHORT-TERM MEMORY" (LSTM)

USING STANDARD LSTM NOTATION:



$$C_t = \underbrace{f_t \odot C_{t-1}}_{\text{KEEP OLD IF } f_t \equiv 1} + \underbrace{i_t \odot \tilde{C}_t}_{\text{WRITE IN}}$$

$$h_t = O_t \odot \tilde{C}_t \xrightarrow{\tanh}$$

$$O_t = o_t(x_t, h_{t-1}) \quad \text{"OUTPUT GATE"}$$

$$\begin{aligned} f_t &= f_t(x_t, h_{t-1}) \\ &\text{"FORGET GATE"} \\ &= \sigma(W_f x_t + V_f h_{t-1} + b_f) \end{aligned}$$

$$\begin{aligned} i_t &= i_t(x_t, h_{t-1}) \\ &\text{"INPUT GATE"} \end{aligned}$$

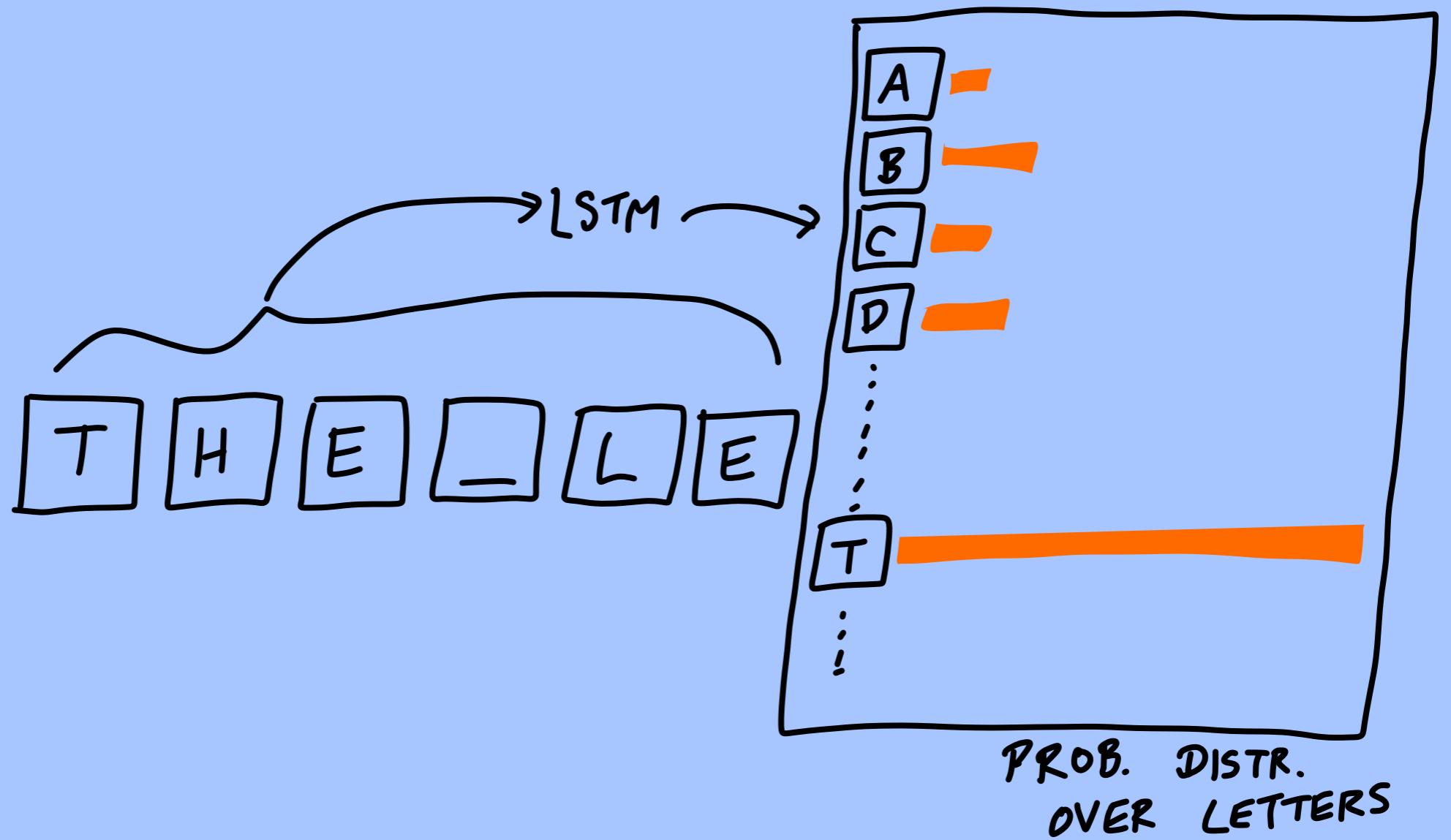
$$\begin{aligned} \tilde{C}_t &= c_t(x_t, h_{t-1}) \\ &\text{"POSSIBLE NEW VALUE"} \end{aligned}$$

## SLIGHTLY SIMPLIFIED:

# "GATED RECURRENT UNIT" (GRU)

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \cdot \tilde{h}_t$$

CONTROLS BOTH INPUT & FORGET



$$-\sum_j P_j \ln Q_j$$

↓  
OUTPUT  
OF NN  
  
1 FOR  
TRUE LETTER

(FROM A. KARPATHY, "THE UNREASONABLE EFFECTIVENESS OF RECURRENT NEURAL NETWORKS")

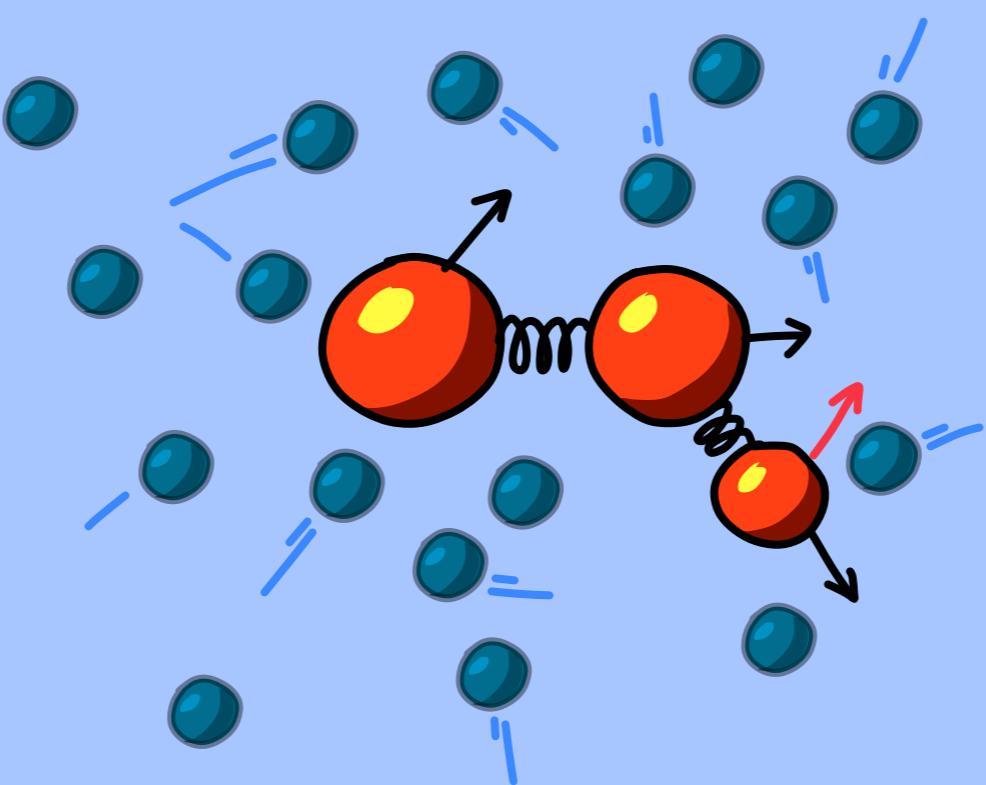
tyntd-iafhatawiaoahrdemot lytdws e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e plia tkldrgd t o idoe ns,smtt h ne etie h,hregtrs nigtike,aoaenns lng

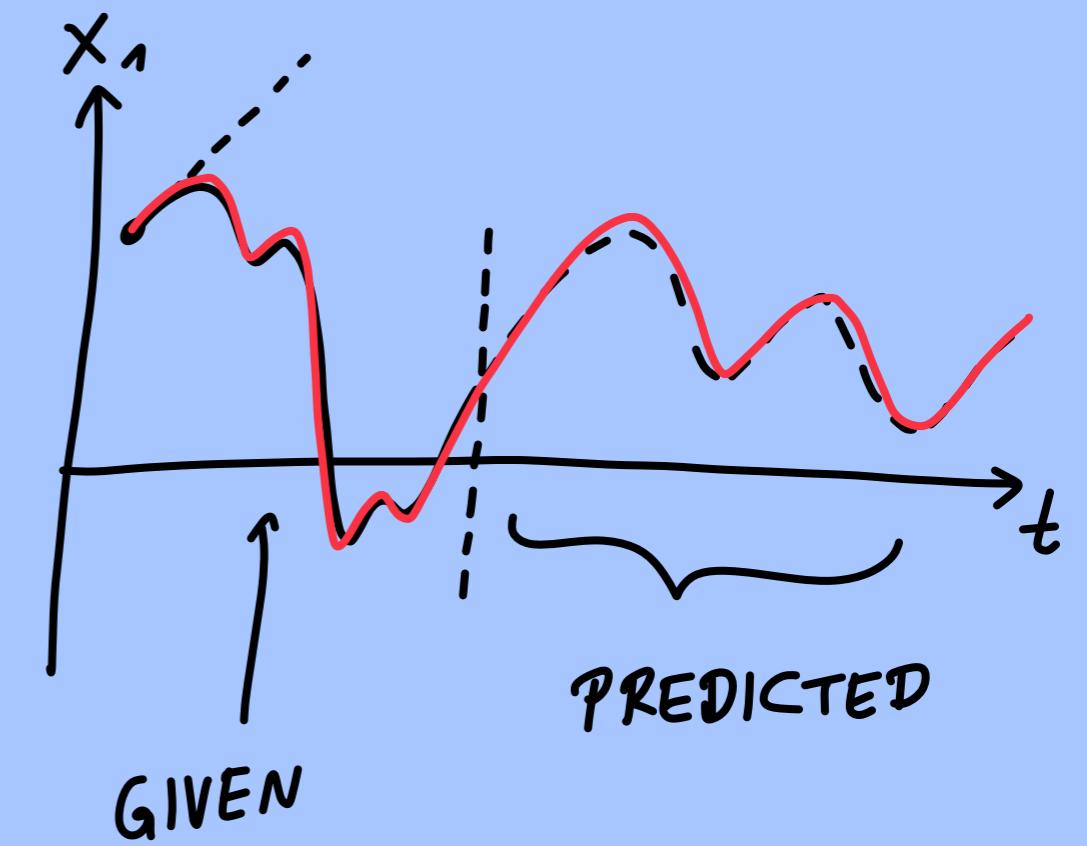
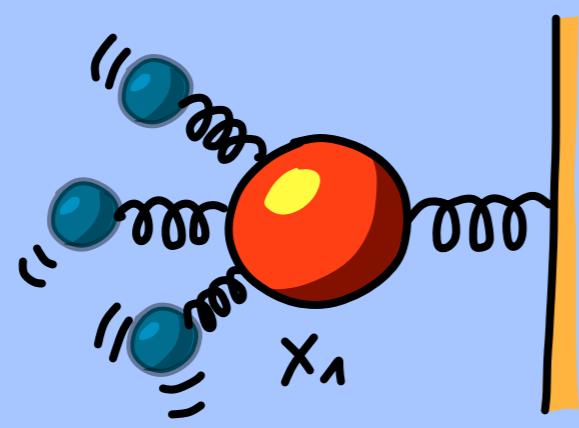
"Tmont thithey" fomesscerliund  
Keushey. Thom here  
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome  
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."

we counter. He stutn co des. His stanted out one ofler that concossions and was  
to gearang reay Jotrets and with fre colt otf paitt thin wall. Which das stimn

Aftair fall unsuch that the hall for Prince Velzonski's that me of  
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort  
how, and Gogition is so overelical and ofter.

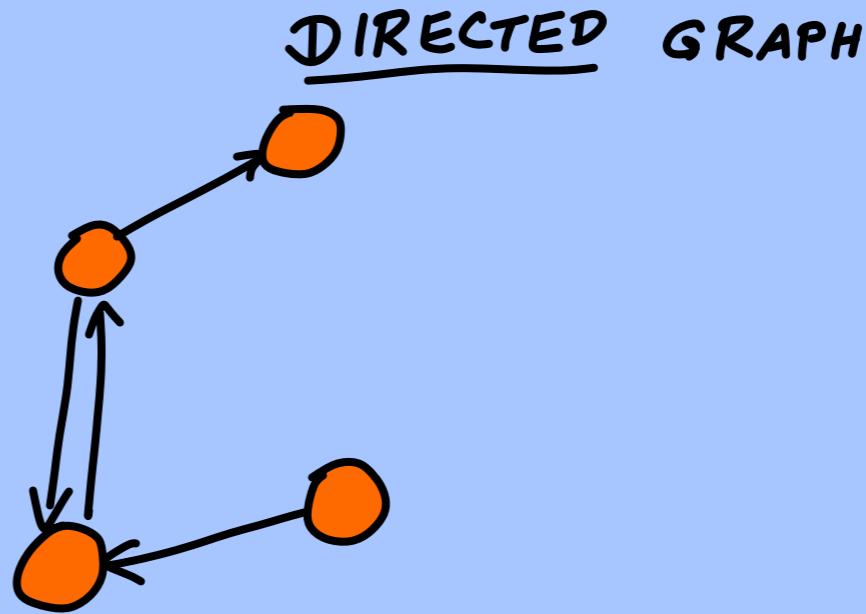
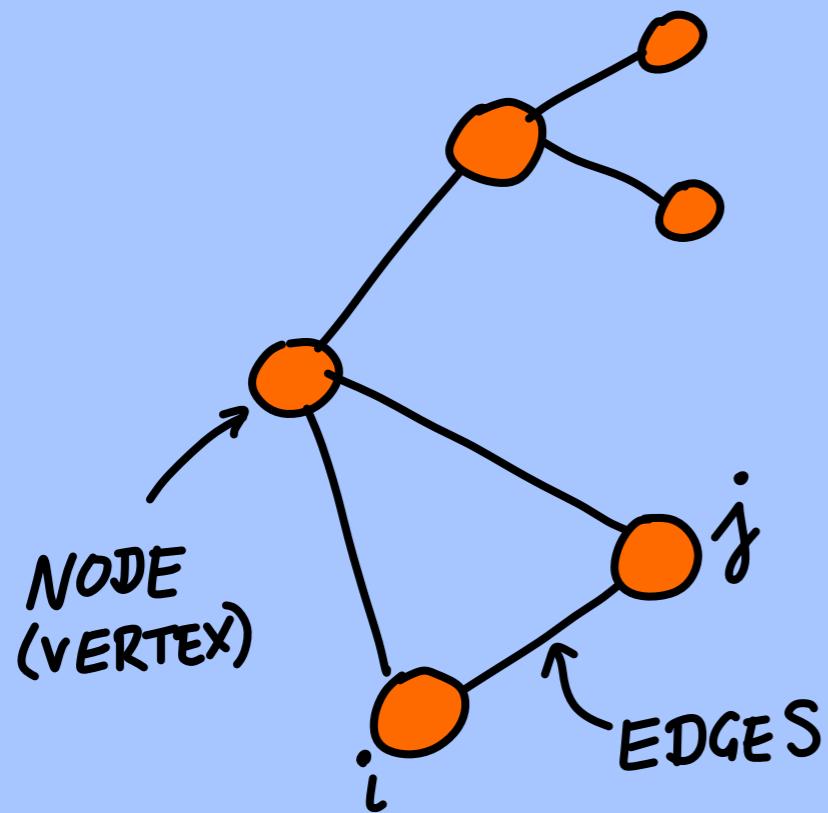
"Why do what that day," replied Natasha, and wishing to himself the fact the  
princess, Princess Mary was easier, fed in had oftened him.  
Pierre aking his soul came to the packs and drove up his father-in-law women.





## 8.2

## GRAPH NEURAL NETWORKS

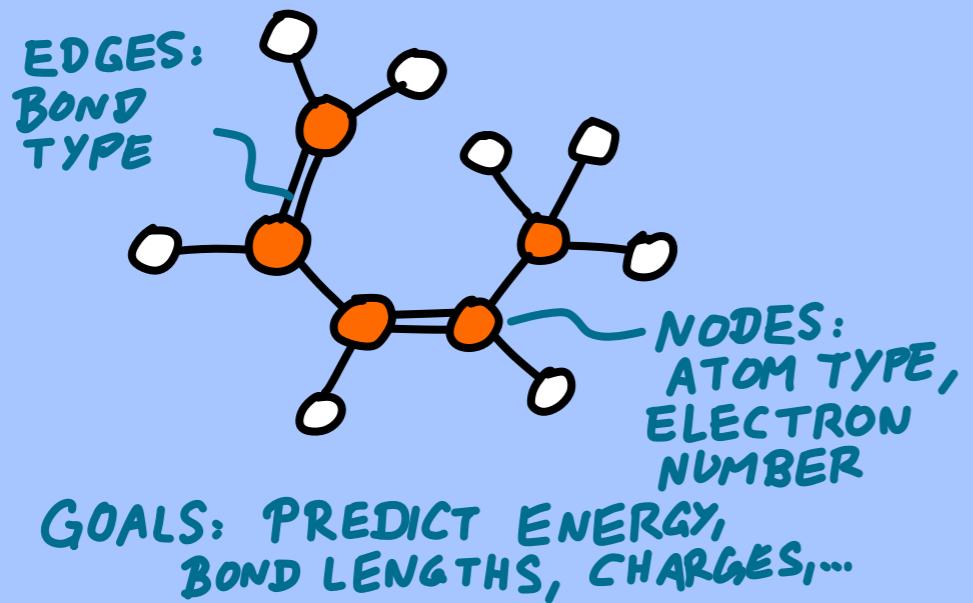


ADJACENCY MATRIX:

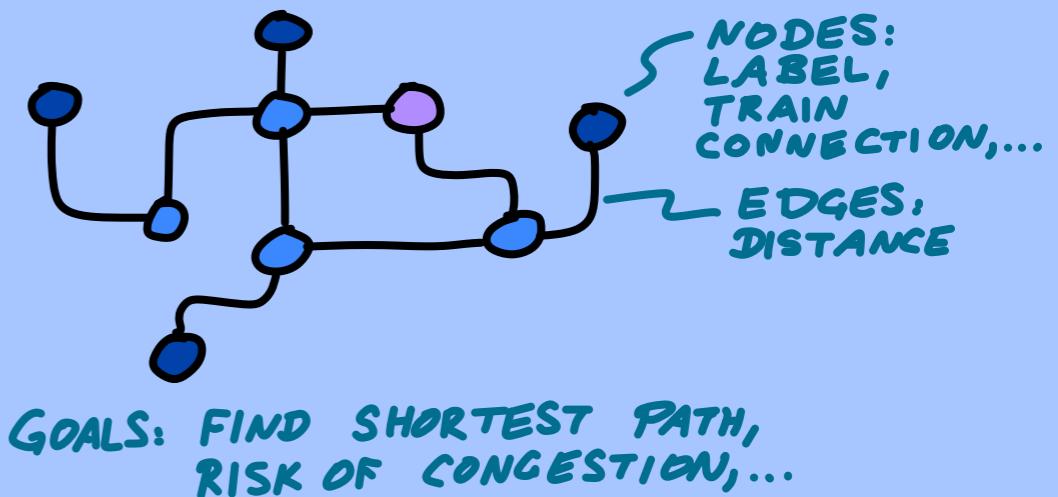
$$A_{ij} = \begin{cases} 1 & \text{EDGE } i \leftarrow j \\ 0 & \text{NO EDGE} \end{cases}$$

$(A^n)_{ij}$  : # OF WALKS  
OF LENGTH  $n$  FROM  $j \rightarrow i$

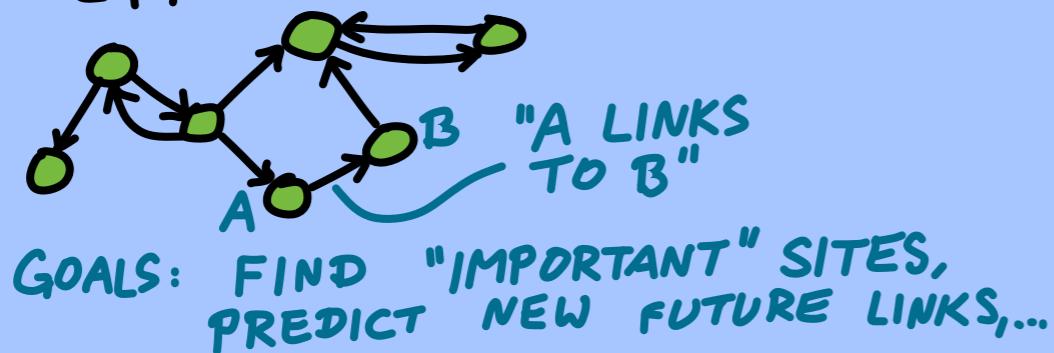
## MOLECULES



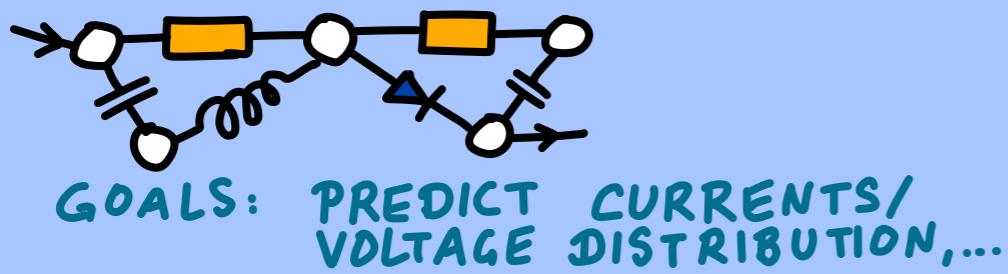
## TRAFFIC NETWORK / SOCIAL NETWORK



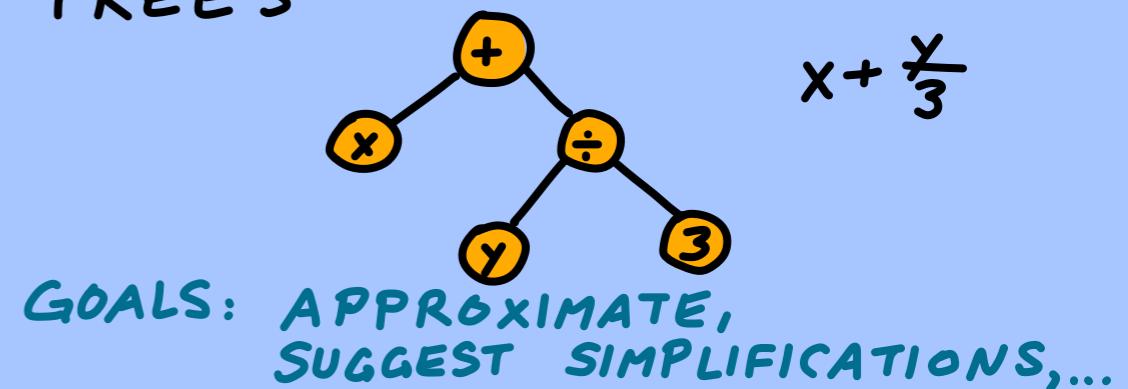
## WORLD WIDE WEB / CITATION NETWORK



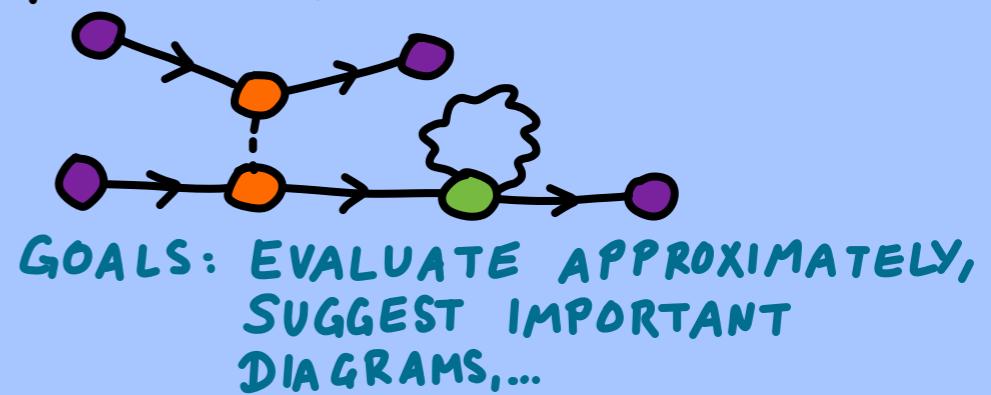
## ELECTRICAL CIRCUITS



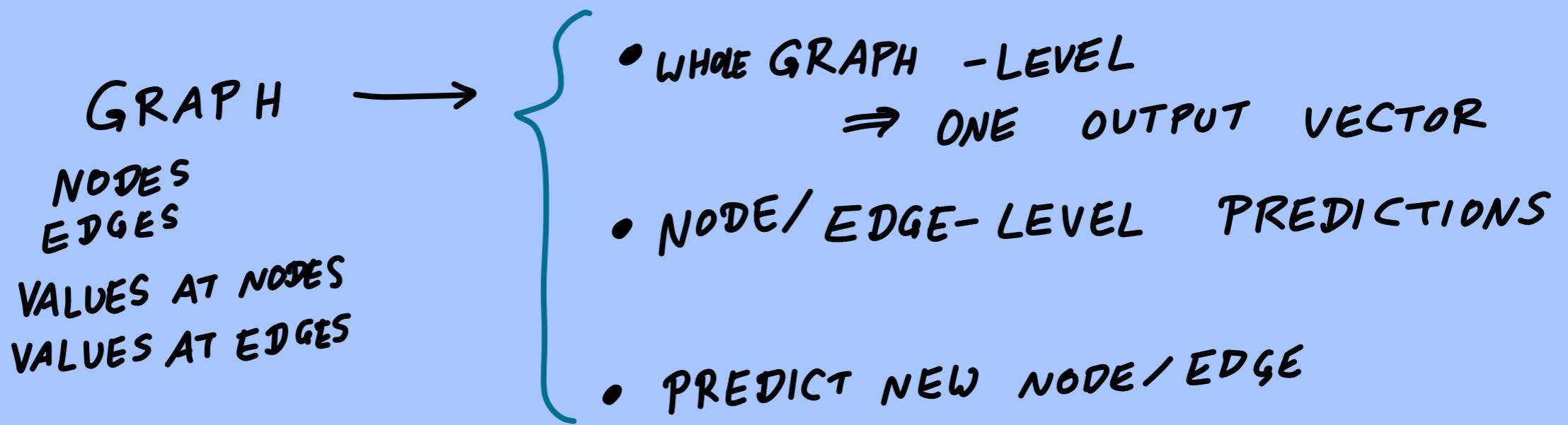
## SYMBOLIC EXPRESSION TREES



## FEYNMAN DIAGRAMS

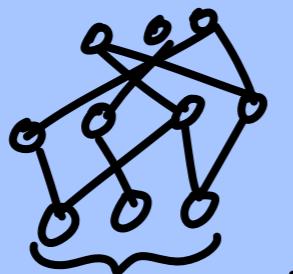


## GOALS FOR GRAPH NN



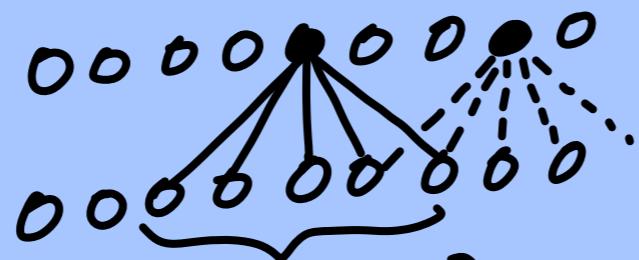
OTHER DATA → GRAPH

GRAPHS ARE VARIABLE-SIZE INPUT!



FIXED #  
OF NEURONS

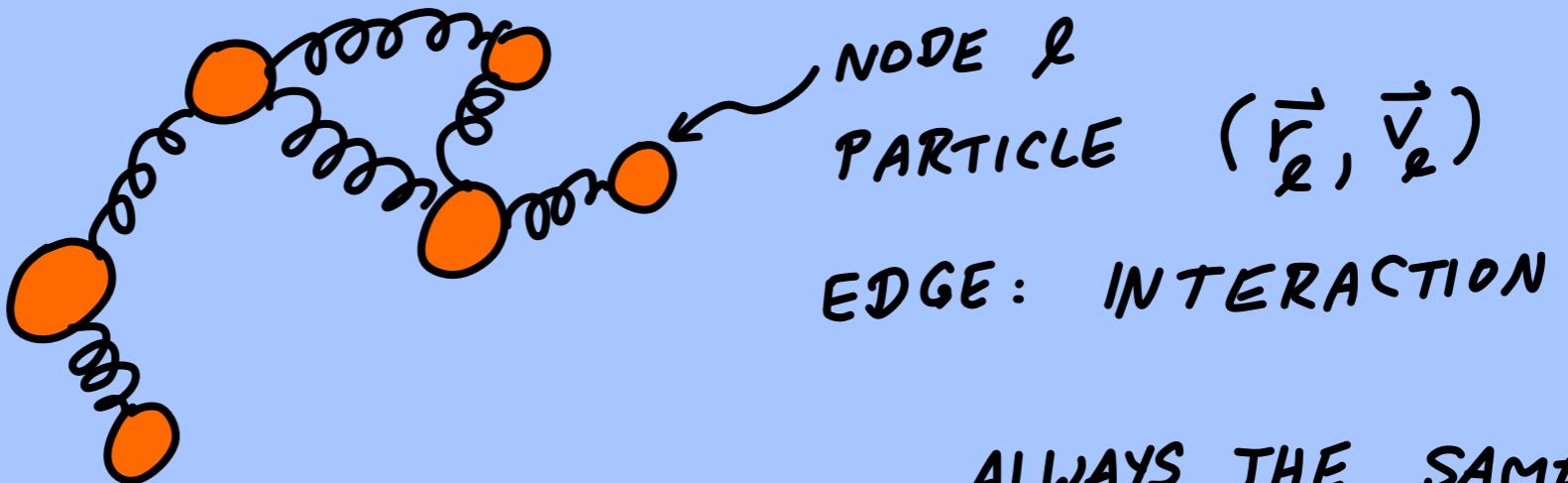
INSPIRATION FROM CNN:



RE-USE 'KERNEL'  
EVERYWHERE

→ INDEP. OF SIZE!

# INSPIRATION FROM PHYSICS



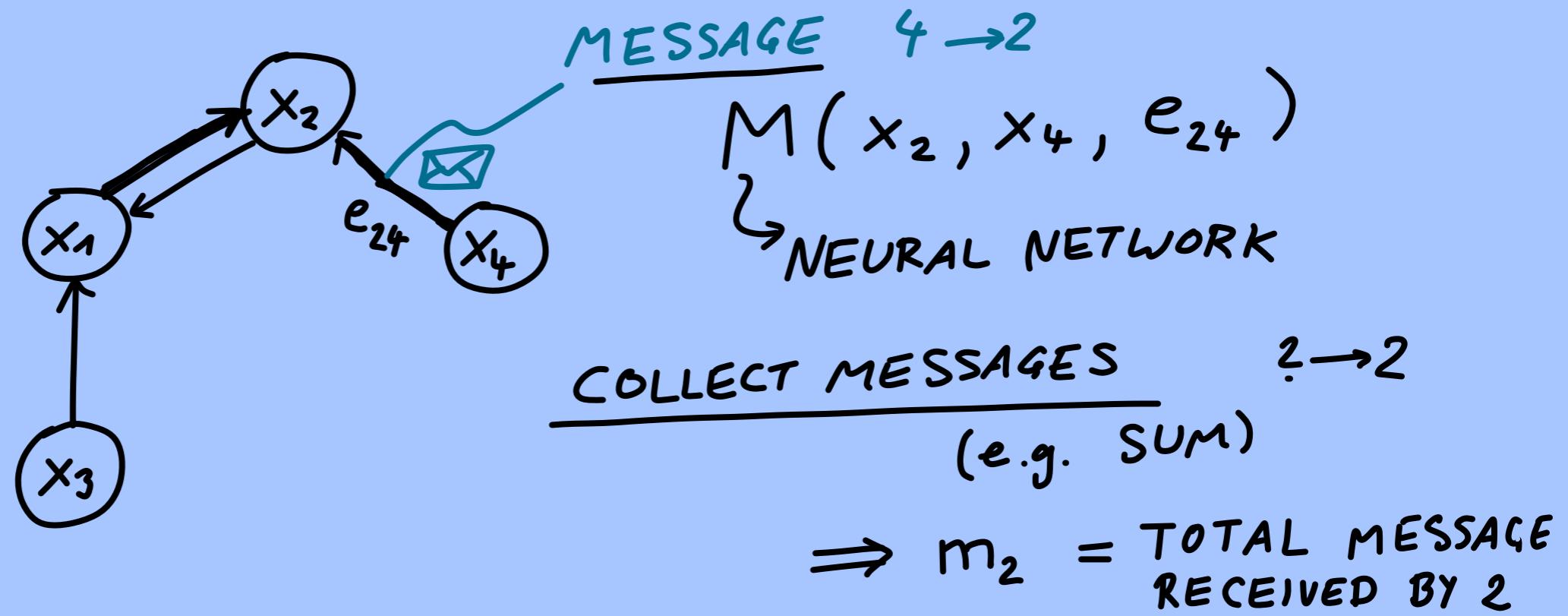
$$m \frac{d}{dt} \vec{v}_\ell = \sum_{j \in \text{NEIGHBOR}(\ell)} \vec{F}(\vec{r}_\ell, \vec{r}_j, D_{\ell j})$$

ALWAYS THE SAME

CHANGE  
OF NODE  
PROPERTY

The equation shows the change of node property (velocity) as a sum of forces from neighboring nodes. A brace under the summation term indicates that the force is always the same for all neighbors. Another brace under the entire equation indicates that it represents the change of node property.

# NEURAL MESSAGE PASSING NETWORK (GENERAL FRAMEWORK)



UPDATE

$$x_2^{\text{NEW}} = U(x_2, m_2)$$

$\curvearrowright$  N.N.

GENERAL PROCEDURE:

$$\forall v \quad m_v^{t+1} = \sum_{w \in \text{NEIGHBORS}(v) \text{ [WITH LINKS } v \leftarrow w]} M_t(x_v^t, x_w^t, e_{vw})$$

VECTOR  
 TARGET NODE  
 SOURCE NODE  
 FIXED EDGE PROPERTY

$w \in$   
 $\text{NEIGHBORS}(v)$   
 [WITH LINKS  
 $v \leftarrow w$ ]

$$\forall v \quad x_v^{t+1} = U_t(x_v^t, m_v^{t+1})$$

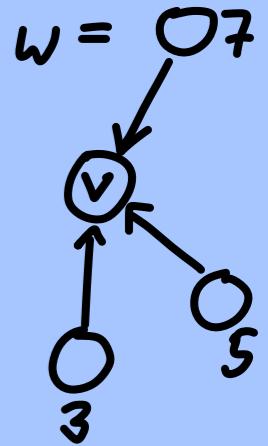
PLUS, POSSIBLY, A READOUT / POOLING:

$$y_{\text{out}} = R(\underbrace{\{x_v^\top \mid v \in \text{GRAPH}\}}_{\text{SET OF NODE VALUES}})$$

PERMUTATION-INVARIANT IN  
 NODE LABELS

$$R(\{\dots\}) = \sum_v r(x_v^\top) \quad \text{OR} \quad R(\{\dots\}) = \max_v r(x_v^\top)$$

MAYBE A  
 LITTLE N.N.



$$OR \quad R(\{\dots\}) = R\left(\sum_v r(x_v^\top), \max_v \tilde{r}(x_v^\top), \min_v \tilde{F}(x_v^\top)\right)$$

POSSIBLE EXTENSION:

$$e_{vw}^{t+1} = U_t^{(\text{EDGE})}(e_{vw}^t, x_v^t, x_w^t)$$

(& THEN MIGHT USE  
 $M_t(x_v^t, x_w^t, e_{vw}^t) = e_{vw}^t$ )

# IMPLEMENTATION

DIRECT IN TENSORFLOW ETC.

$$m_v^{t+1} = \sum_{w \in \text{NEIGHB}(v)} M_t(x_v^t, x_w^t, e_{vw})$$

EXPLOIT "CNN"

$x_1 \ x_2 \ x_3 \ \dots$

$\underbrace{(x_1, x_2)}_{\dots} \underbrace{(x_1, x_3)}_{\dots} \dots \underbrace{(x_5, x_9)}_{\dots} \dots$

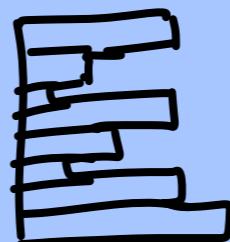
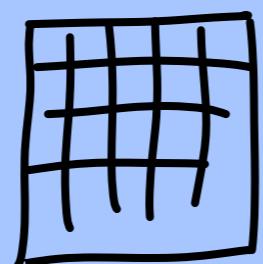
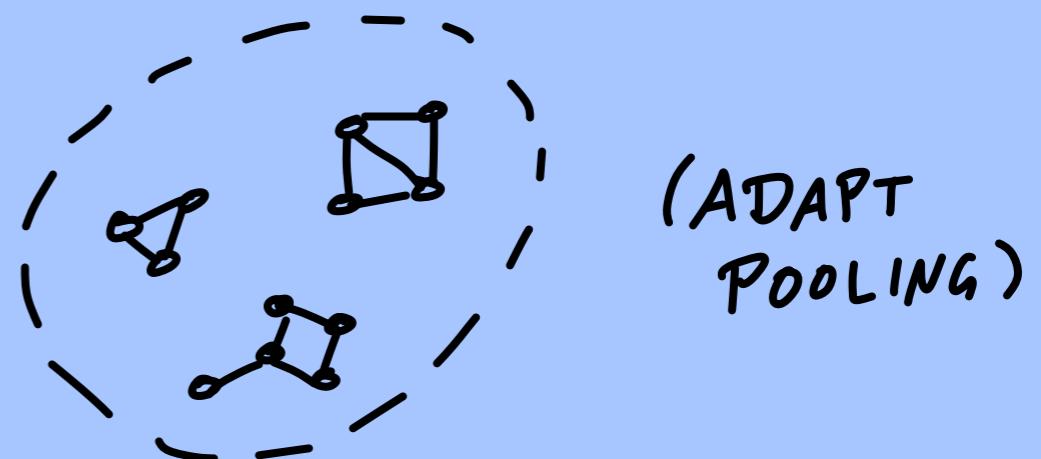
BATCHES?

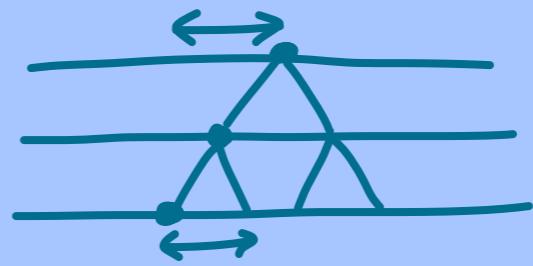
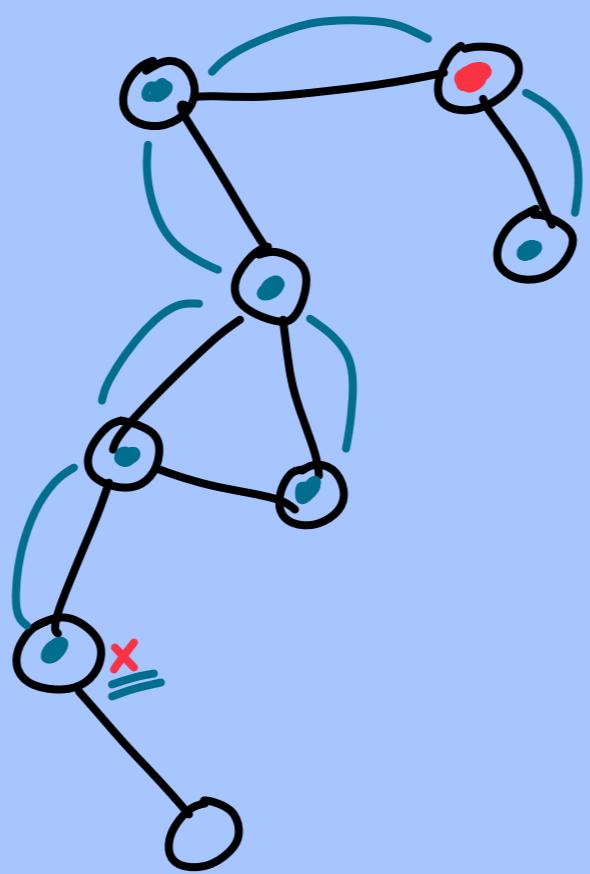
USUALLY

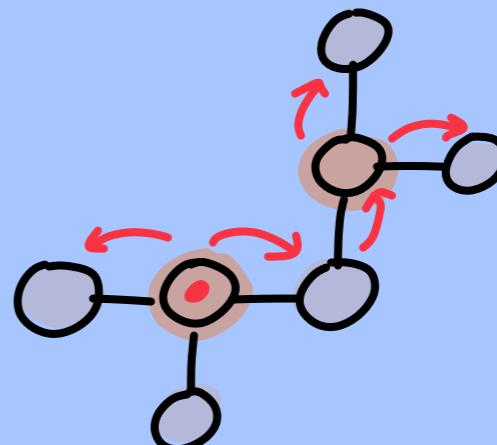
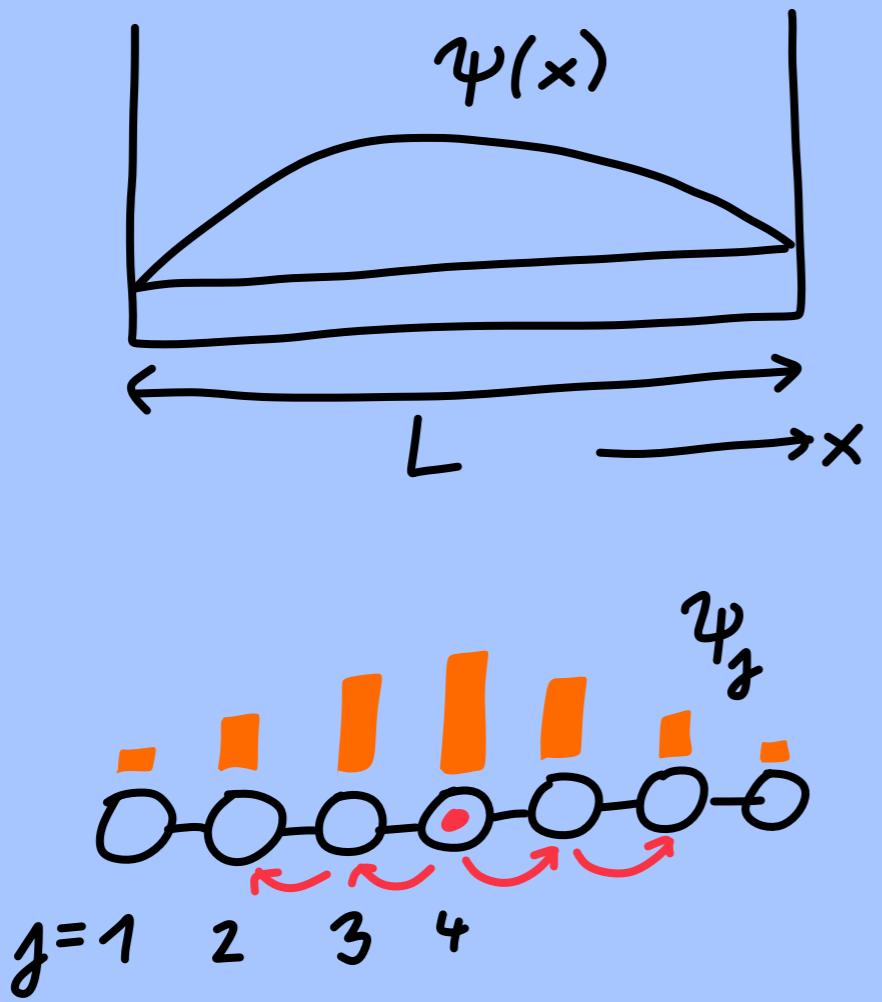
[BATCH-DIM., INPUT-NEUR.]

VARIABLE SIZE!

- ONE GRAPH AT A TIME
- "PADDING"
- ONE COMBINED GRAPH
- "RAGGED TENSORS"

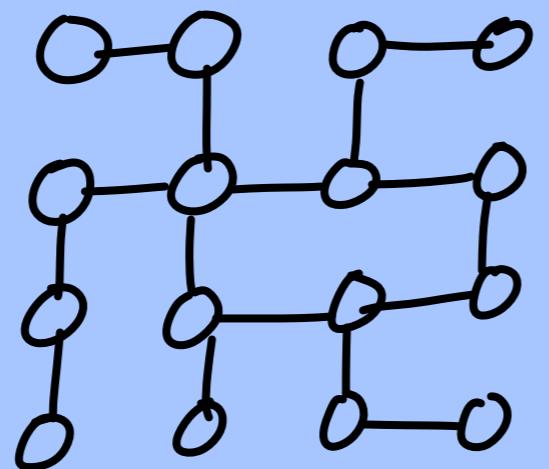






$$H_{ij} = \begin{cases} -K & i \leftrightarrow j \\ 0 & i \not\leftrightarrow j \end{cases}$$

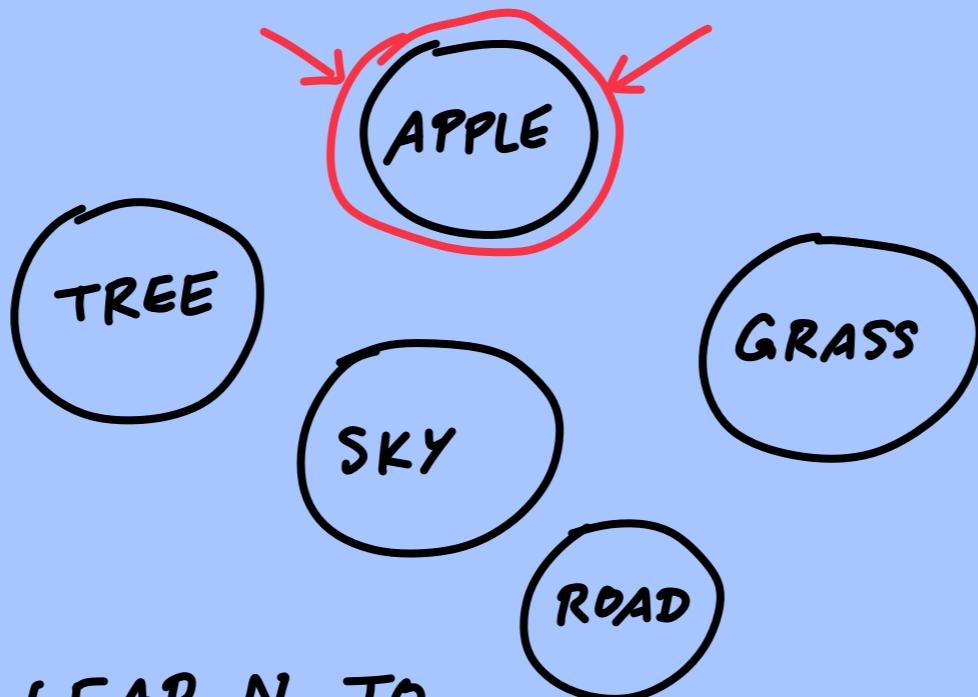
$= -K A_{ij}$



8.3

## ATTENTION

AGAIN, GOAL: VARIABLE-SIZE DATA



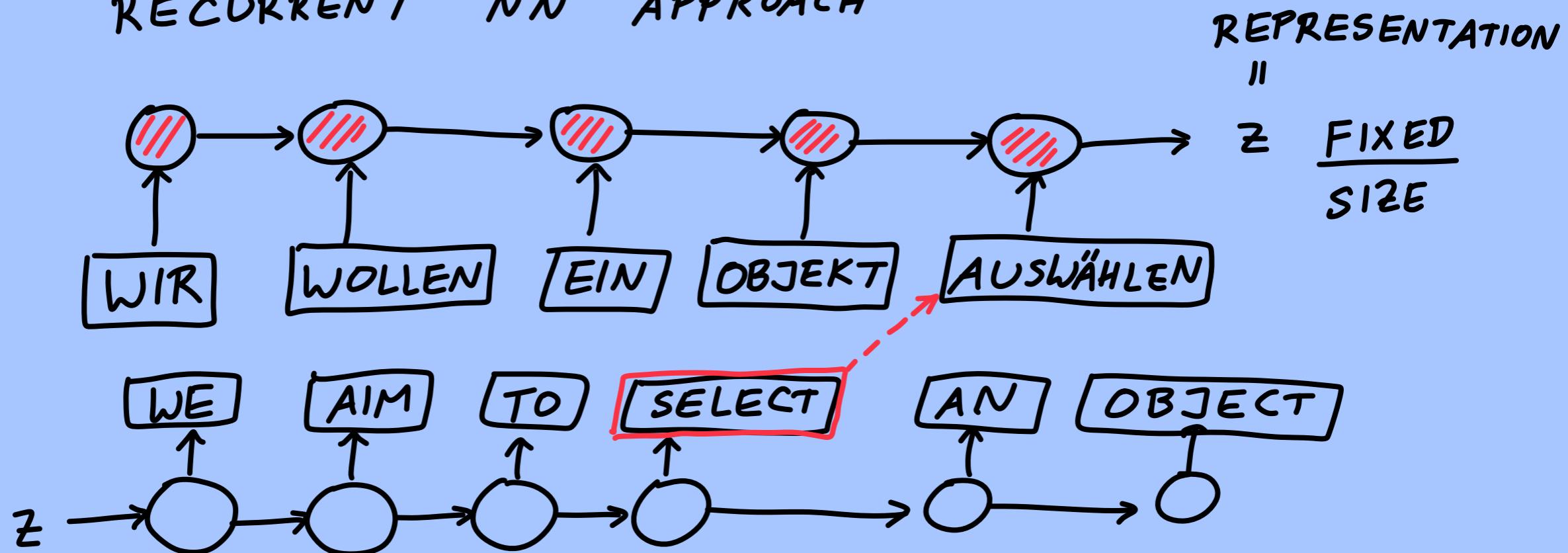
LEARN TO...

- SELECT FROM VAR.-SIZE SET OF OBJECTS (DEPENDING ON CURRENT & PREVIOUS INPUTS)
- LEARNABLE SEARCH

## MOTIVATION:

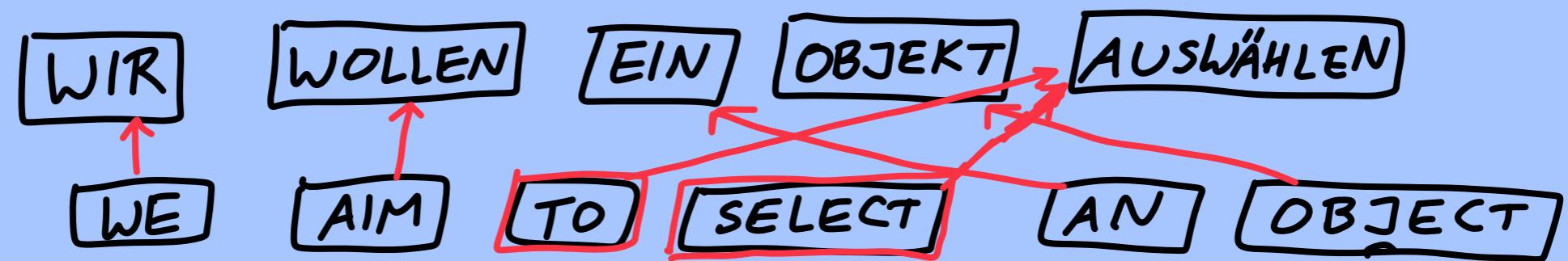
### 1. EXAMPLE: TRANSLATION

#### RECURRENT NN APPROACH



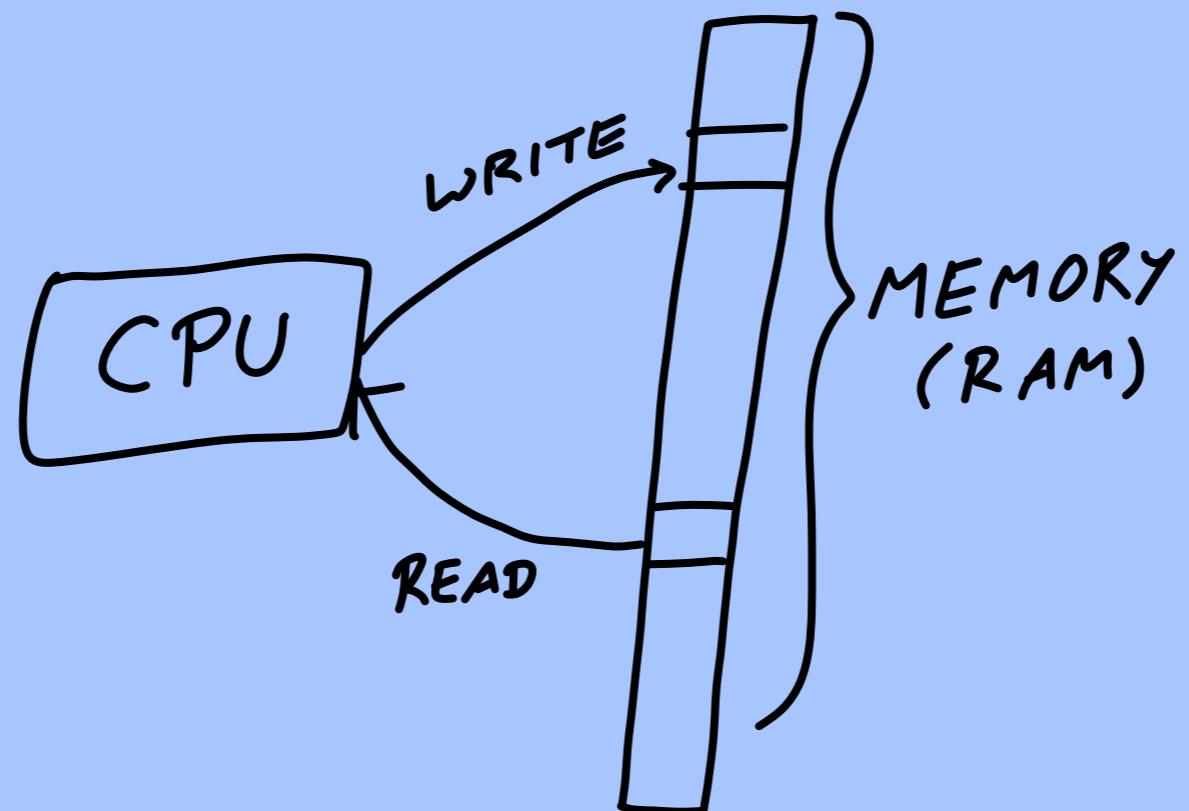
#### IDEA:

PAY ATTENTION TO SELECTED WORDS IN THE ORIGINAL SENTENCE, ACCORDING TO WHICH WORD WE ARE TRYING TO PRODUCE



## 2. EXAMPLE: DIFFERENTIABLE NEURAL COMPUTER

GOAL: USE VARIABLE-SIZE MEMORY,  
LEARN 'PROGRAM' (NN) THAT  
ACCESES MEMORY



# WHERE TO WRITE/READ?

- SIMPLE: LOOP SEQUENTIALLY



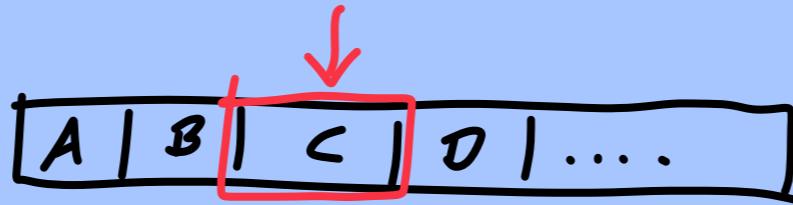
- CONTENT-ADDRESSABLE MEMORY

KEY,	VALUE,
" 2	" 2
3	3

LOOKUP: " QUERY "  $q \Rightarrow$   
FIND KEY  $k$  MATCHING  $q$ ,  
RETURN VALUE  $v$

$\triangleq$  DATABASE LOOKUP

## HARD ATTENTION:



→ REINFORCEMENT  
LEARNING

## SOFT ATTENTION:

### ATTENTION:



### ATTENTION WEIGHTS

⇒ CAN USE  
GRADIENT DESCENT

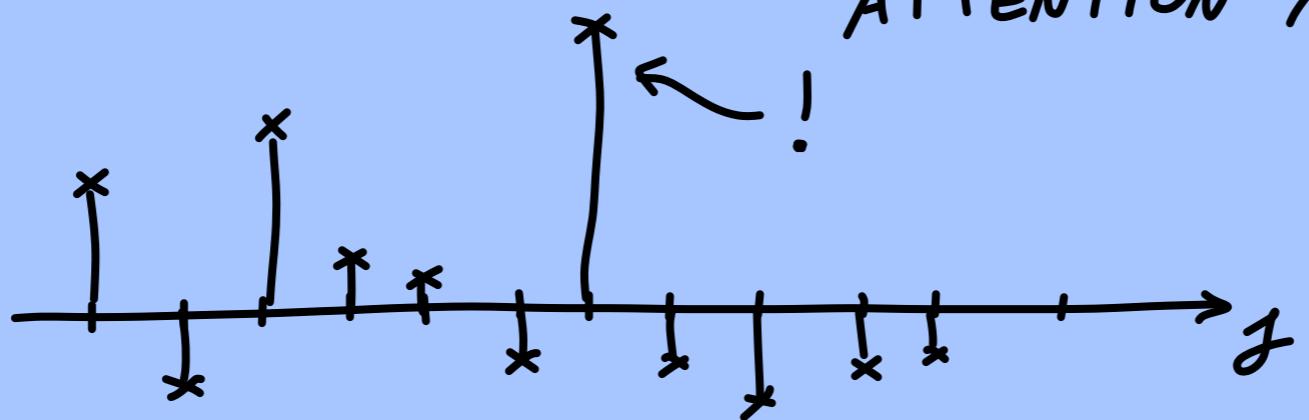
## IMPLEMENTATION:

CONSIDER OBJECTS WITH KEYS  $k_j \in \mathbb{R}^{d_k}$   
AND VALUES  $v_j \in \mathbb{R}^{d_v}$  ( $j = 1, 2, \dots, n$ )

1. GIVEN SOME QUERY  $q \in \mathbb{R}^{d_k}$   
 $\Rightarrow$  CALCULATE "SCORES"

$$e_j = a(q, k_j)$$

"ATTENTION" / "ALIGNMENT"

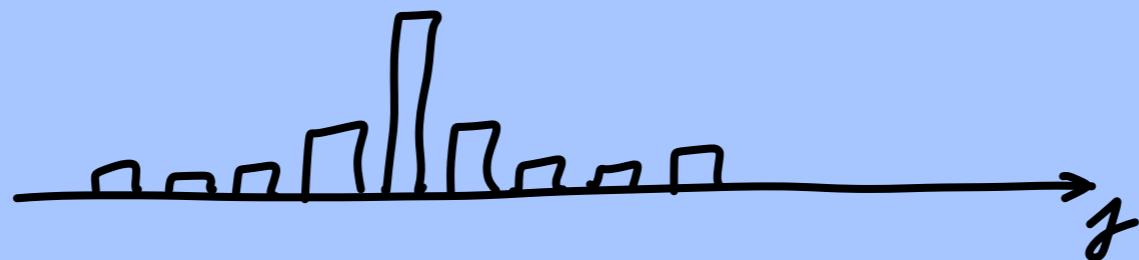


2. CONVERT TO PROB. DISTRIBUTION  
USING SOFTMAX:

$$\underline{w_j} = \frac{\exp(\beta e_j)}{\sum_{j'} \exp(\beta e_{j'})}$$

$\beta$  LARGE  $\Rightarrow$  "HARD ATTENTION"

(MAYBE NOT NEEDED  
IF  $\alpha$  LEARNED)



3. RETURN WEIGHTED SUM:

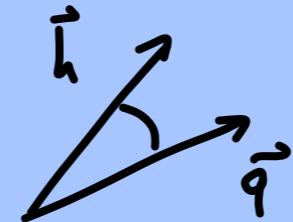
$$\underbrace{v^{\text{out}}}_{\substack{\text{OUTPUT} \\ \text{OF THE} \\ \text{ATTENTION} \\ \text{MECHANISM}}} = \sum_j w_j v_j$$

## ATTENTION EXPRESSIONS:

$$\alpha(\vec{q}, \vec{h}) = NN$$

$$\alpha(\vec{q}, \vec{h}) = \vec{q} \cdot \vec{h}$$

"DOT PRODUCT"



$$\alpha(\vec{q}, \vec{h}) = \frac{\vec{q} \cdot \vec{h}}{|\vec{q}| \cdot |\vec{h}|} = \cos \hat{x}(\vec{h}, \vec{q})$$

"COSINE SIMILARITY"

$$\alpha(\vec{q}, \vec{h}) = \frac{\vec{q} \cdot \vec{h}}{\sqrt{\dim \vec{h}}}$$

"SCALED DOT PRODUCT"

MOTIVATION: IF  $\vec{q}_2$  RANDOM &  
 $\vec{h}_2$  VARIANCE=1

$$\Rightarrow \vec{q} \cdot \vec{h} \sim \mathcal{O}(\sqrt{\dim \vec{h}})$$

MEAN=0

$$\Rightarrow \alpha \sim \mathcal{O}(1) \Rightarrow \text{GOOD}$$

EXAMPLE:

$$\vec{h} = \boxed{0 \ 1 \ 0 \ 0 \ 0 \ 0} \xrightarrow{\text{RED}} \text{GREEN} \xrightarrow{\text{BLUE}} \rightarrow \text{'ONE-HOT ENCODING'}$$
$$= (0, 1, 0, 0, 0, 0)$$

$\Rightarrow$  FOR  $\vec{q} = (0, 1, 0, 0, \dots)$  :

$$\vec{q} \cdot \vec{h} = 1$$
$$\vec{q} \cdot \vec{h}' = 0 \quad \text{FOR } \vec{h}' \neq \vec{h}$$

$\Rightarrow$  ALSO FOR MORE PROPERTIES

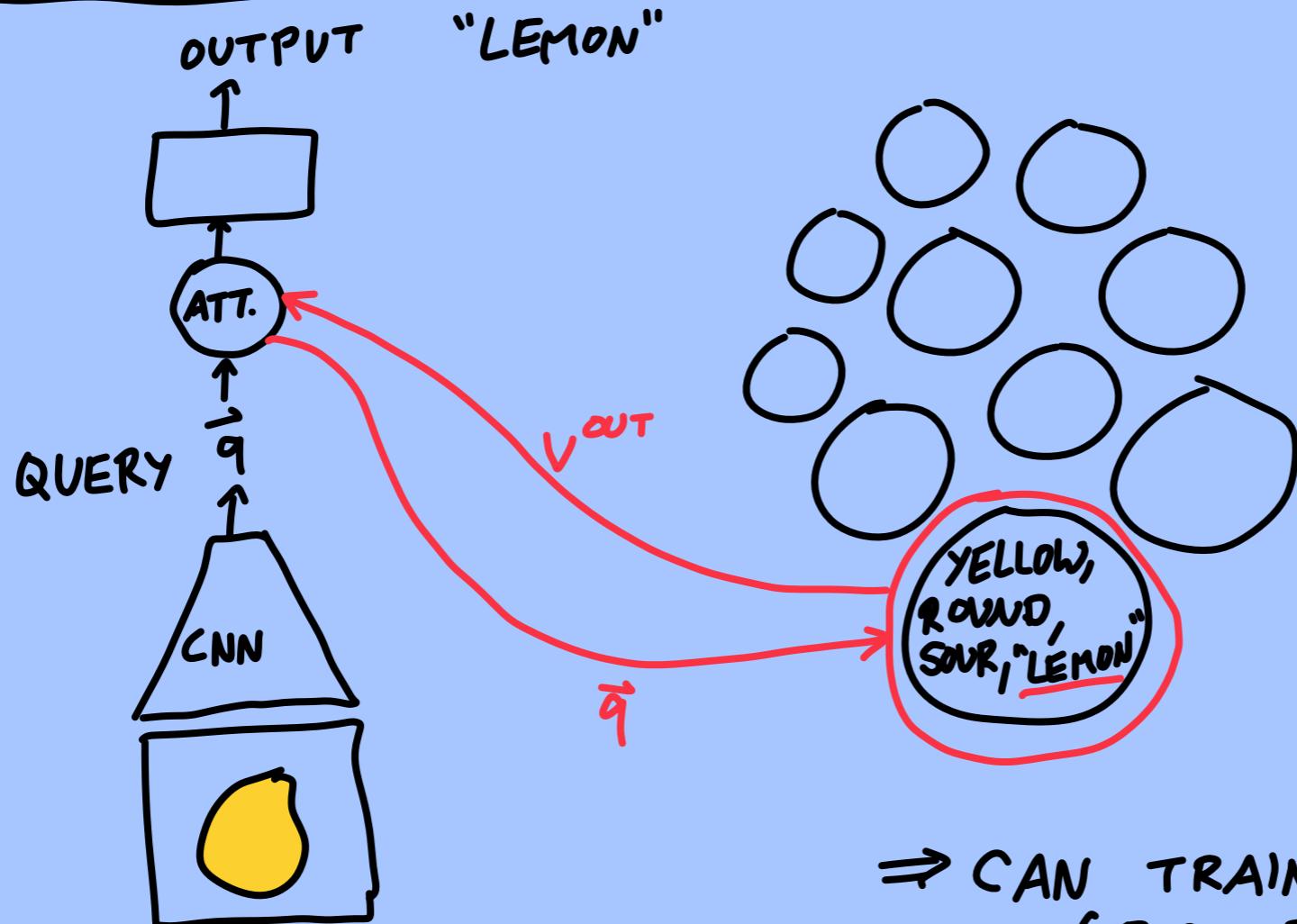
$$\vec{h} = (\text{COLOR}, \text{SHAPE}, \text{TASTE})$$

$$= \boxed{\text{---} \text{---} \text{---} \text{---} \text{---} \text{---}}$$

$$\vec{q} = 000000 \ 00\underline{1}000 \ 000000$$
$$\underline{0010\underline{1}00}$$

NOTE: QUERY & KEY FORMAT ARE LEARNED!  
(CAN USE GRADIENT DESCENT!)

### TOY MODEL



⇒ CAN TRAIN ONCE  
(PICTURE → QUERY),  
USE ON DIFF'T DATABASES

CAN WE ALSO CHANGE THE SET OF OBJECTS?

EXAMPLE:

→ NEURAL TURING MACHINE /  
DIFFERENTIABLE NEURAL COMPUTER (DNC)

ADD WRITE OPERATION TO DNC:

ATTENTION WEIGHTS  $w_j$

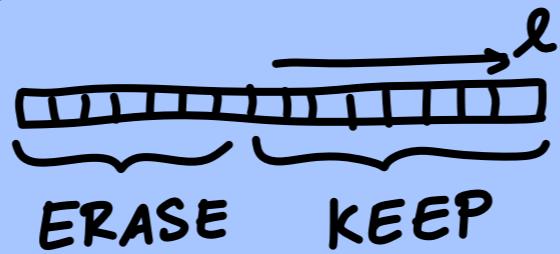
$M_{j\ell}$   
"MEMORY"      MEMORY  
COMPONENT  $\ell$  OF  
OBJECT / VECTOR  $M_j$ .

$$M_{j\ell} = w_j \cdot M_e^{\text{NEW}} + (1 - w_j) \cdot M_{j\ell}^{\text{OLD}}$$

↑ WRITE WHERE  $w_j = \text{LARGE}$

OLD CONTENT

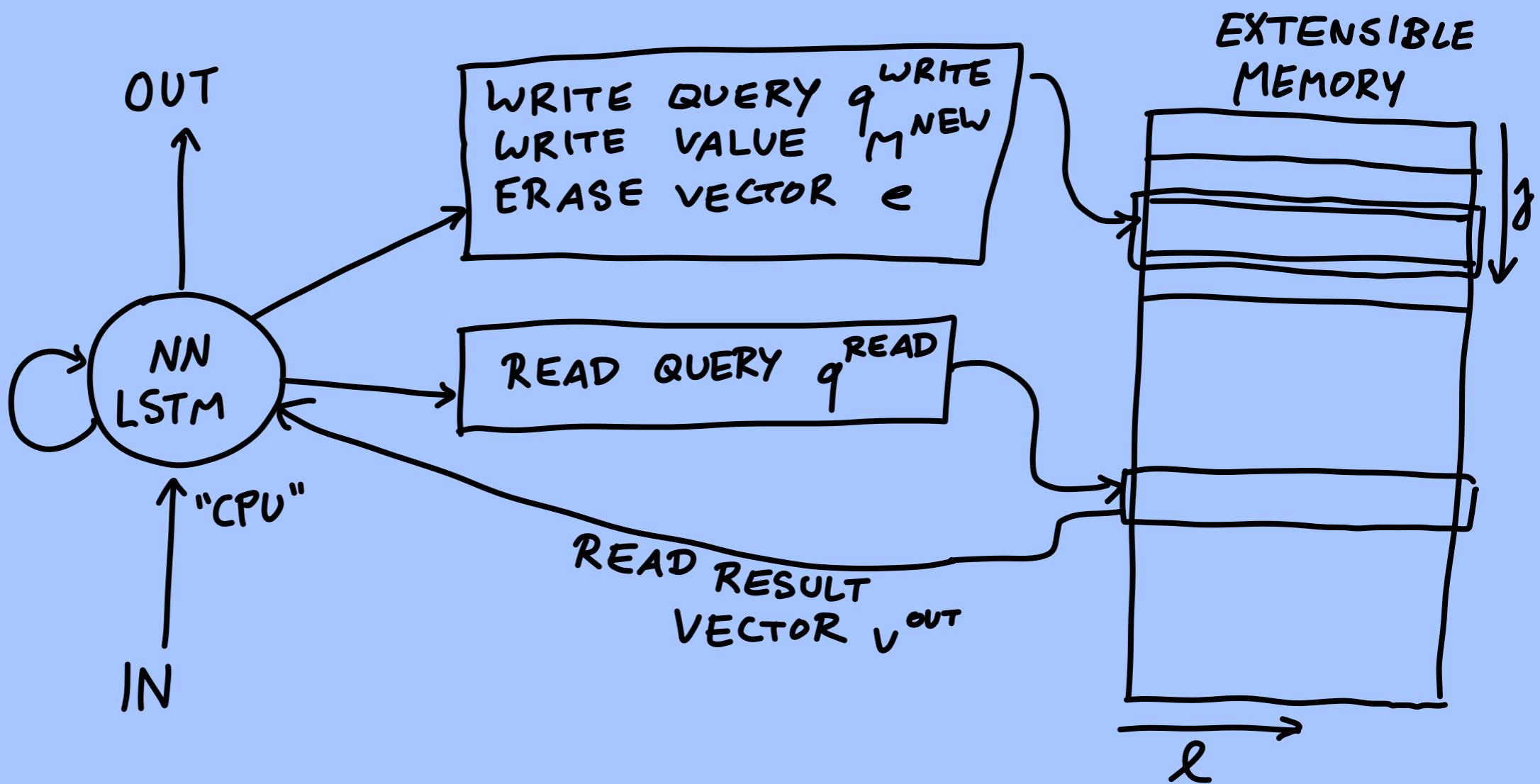
PARTIAL ERASURE

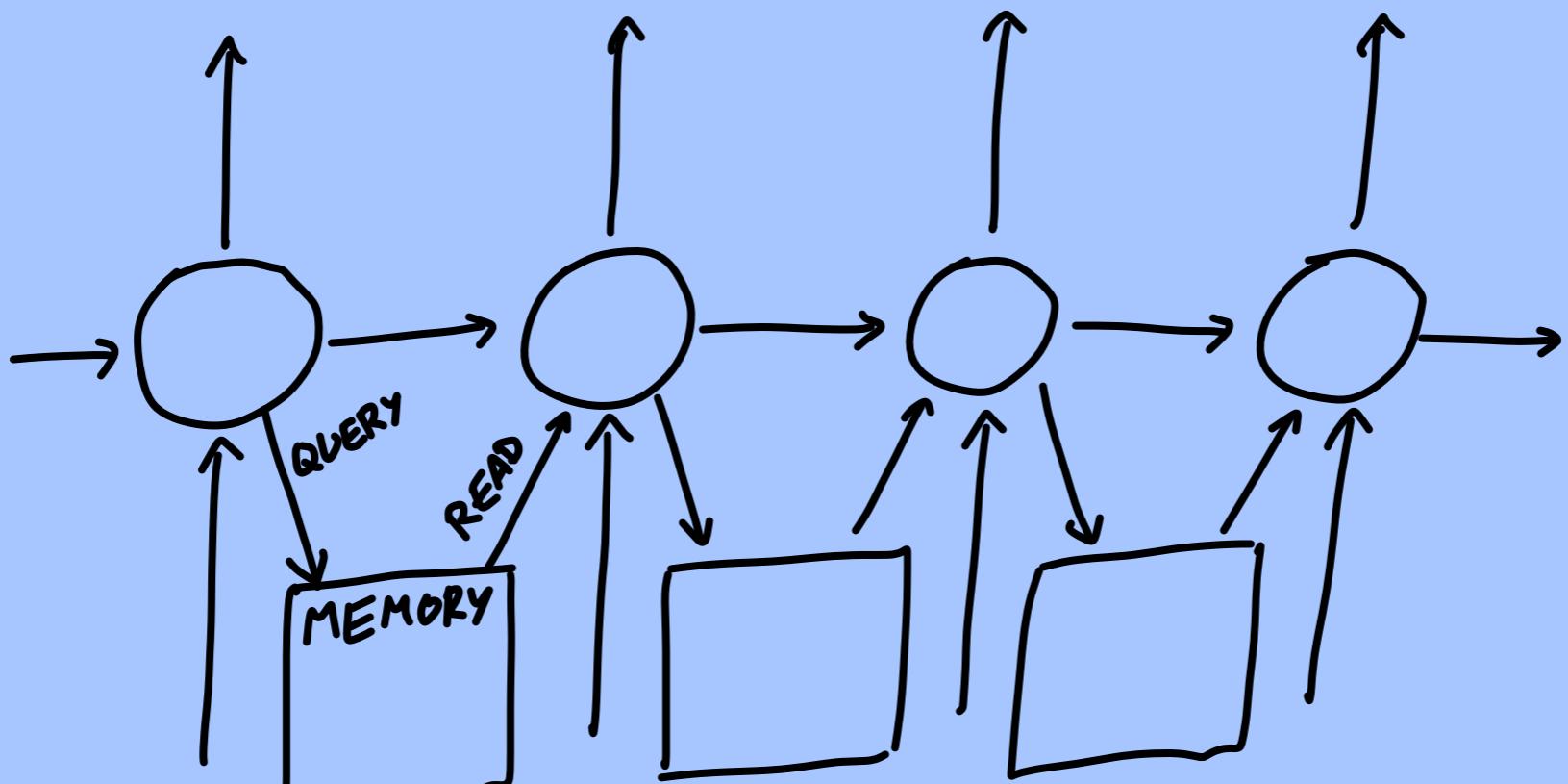


$$(1 - w_j) M_{je} \Rightarrow (1 - w_j e_e) M_{je}$$

$\uparrow$   
"ERASE VECTOR"

# DNC LAYOUT





### ADDITIONAL FEATURES:

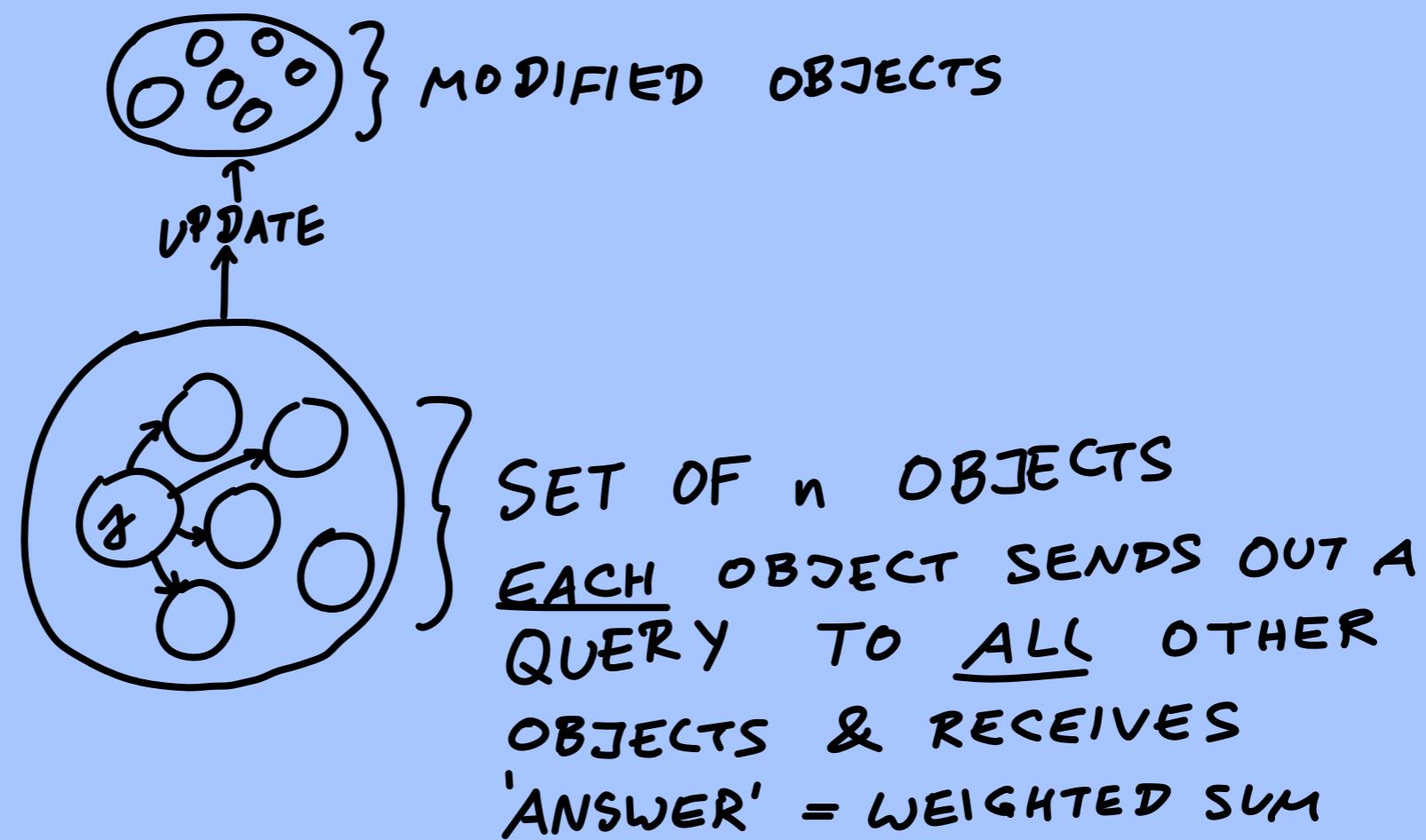
- MULTIPLE READ/WRITE "HEADS" TIME
- TRACK USAGE & WRITE INTO "FREE" LOCATIONS
- TRACK SEQUENCE OF WRITE LOCATIONS → "LINK MATRIX"
- READ IN ORDER OF WRITING

8.4

## TRANSFORMERS

NO RECURRENT NET

ONLY ATTENTION



SIMPLEST VERSION:

TRACK  $j$  = INDEX OF OBJECT  
THAT SENDS OUT QUERY

$j'$  = INDEX OF 'OTHER' OBJECT

$\ell$  = VECTOR COMPONENT

$D_{j\ell}$  = DATA IN OBJECT  $j$

$D \in \mathbb{R}^{n \times d_{\text{model}}}$

$Q_{j\ell'} = \sum_k D_{jk} W_{k\ell'}^Q$   $\Rightarrow$  QUERY  
BY OBJECT  $j$   
LEARNABLE  
MATRIX

$Q \in \mathbb{R}^{n \times d_k}$

MATRIX NOTATION:  $Q = DW^Q$

LIKewise:  $K = DW^K \in \mathbb{R}^{n \times d_h}$

$V = DW^V \in \mathbb{R}^{n \times d_v}$

NOW: NEW VALUES VIA ATTENTION:

$$V' = \text{Attention}(Q, K, V)$$

$$V'_{j^l} = \sum_{j'} w_{j \leftarrow j'}^{\text{Att}} \cdot V_{j'^l}$$

$w_{j \leftarrow j'}^{\text{Att}}$  = ATT. WEIGHT  
FOR QUERY  $Q_j$ .  
AND KEY  $K_{j'}$ .

$$= \underset{(\text{OVER } j')}{\text{softmax}} \left( \frac{\sum_l Q_{j^l} K_{j'^l}}{\sqrt{d_k}} \right)$$

$$V' = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) \cdot V$$

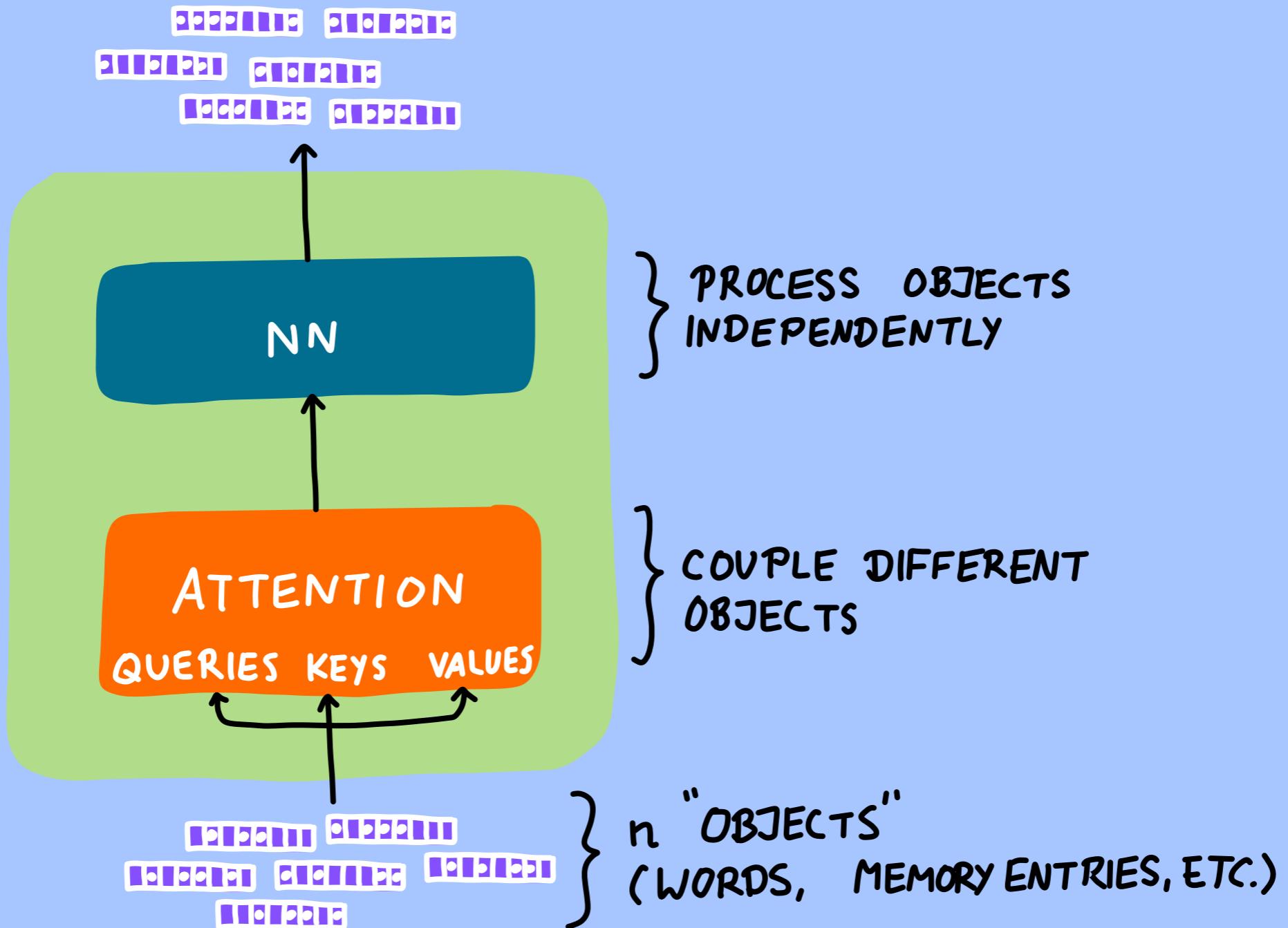
THEN:

$$\mathcal{D}'_{j\cdot} = \text{NN}(V'_{j\cdot})$$

SAME NN  
FOR EACH OBJECT

UPDATE  $\mathcal{D} \rightarrow \mathcal{D}'$

$\Rightarrow$  REPEAT MULTIPLE TIMES!  
(LIKE MANY LAYERS)



HARD TO TRAIN MULTILAYER NN:  
TRICKS:

RESIDUAL CONNECTIONS

$$x' = f(x) \quad \xrightarrow{\text{~~~~~}} \quad x' = x + f(x)$$

LAYER NORMALIZATION

$$[\text{LayerNorm}(x)]_l = \frac{x_l - \bar{x}}{\sqrt{d}}$$

$$\bar{x} = \frac{1}{d} \sum_{l=1}^d x_l$$

$$\sigma^2 = \frac{1}{d} \sum_l (x_l - \bar{x})^2$$

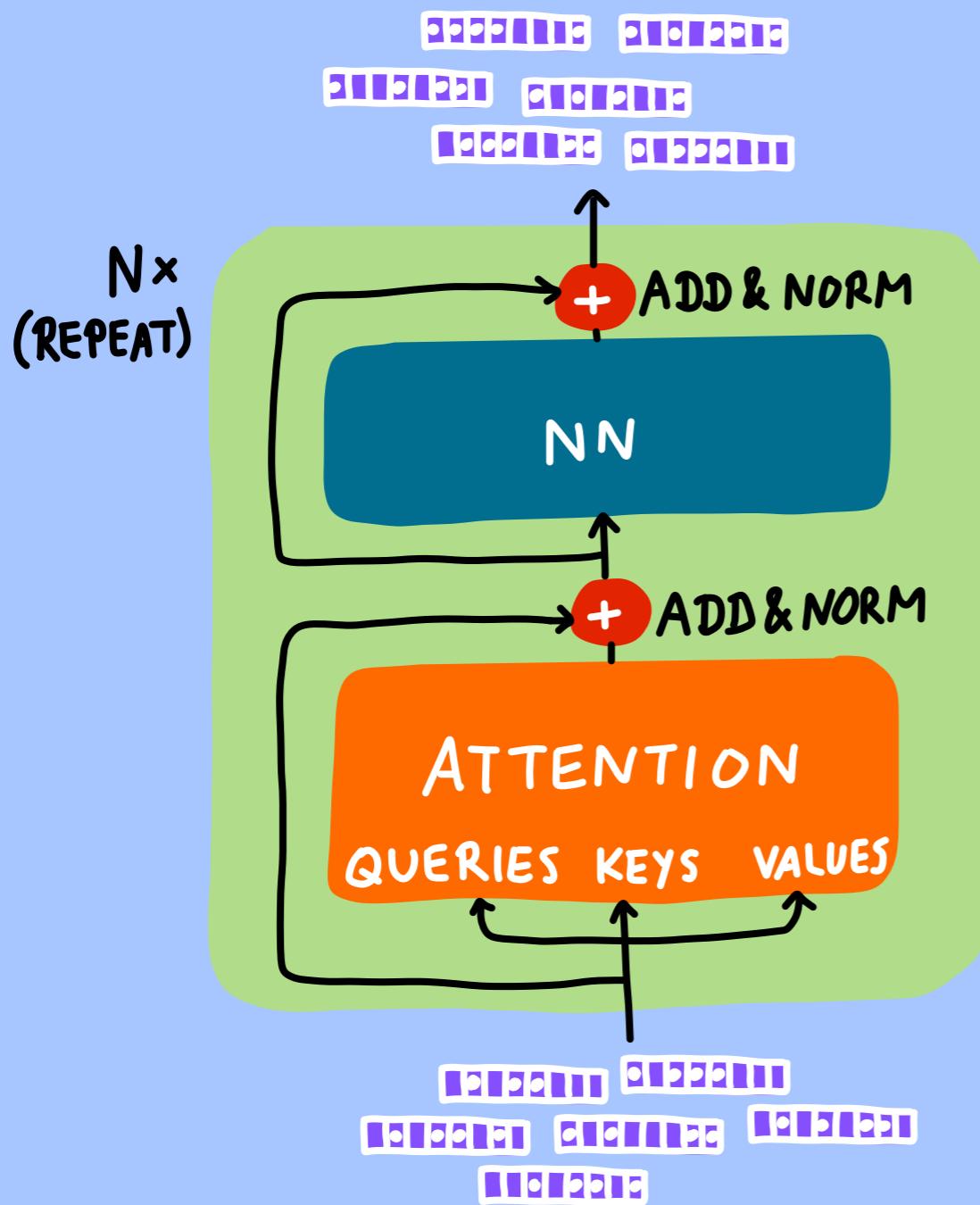
HERE :

$$V' = \text{LayerNorm} \left( \text{⊗} + \text{Attention}(Q, K, V) \right)$$

INDEP. FOR EACH OBJ.

WORKS IF  $d_{\text{model}} = d_V$

$$D' = \text{LayerNorm} (V' + \text{NN}(V'))$$



MULTIPLE "HEADS":

$$Q_{j\ell'}^{(i)} \overset{\text{HEAD}}{\leftarrow} = \sum_{\ell} D_{j\ell} W_{\ell\ell'}^{Q(i)}$$

$$i=1, \dots, h$$

ALSO FOR K, V

$$\Rightarrow V^{(i)} = \text{Attention}(Q^{(i)}, K^{(i)}, V^{(i)})$$

$$V' = \text{Multi Head}(Q, K, V) = \underbrace{\text{Concat}(V^{(1)}, \dots, V^{(h)})}_{\text{ALONG VECTOR INDEX}} \overset{h \cdot d_v \times d_{\text{model}}}{\overbrace{W^O}}$$

ORIG. TRANSF.:

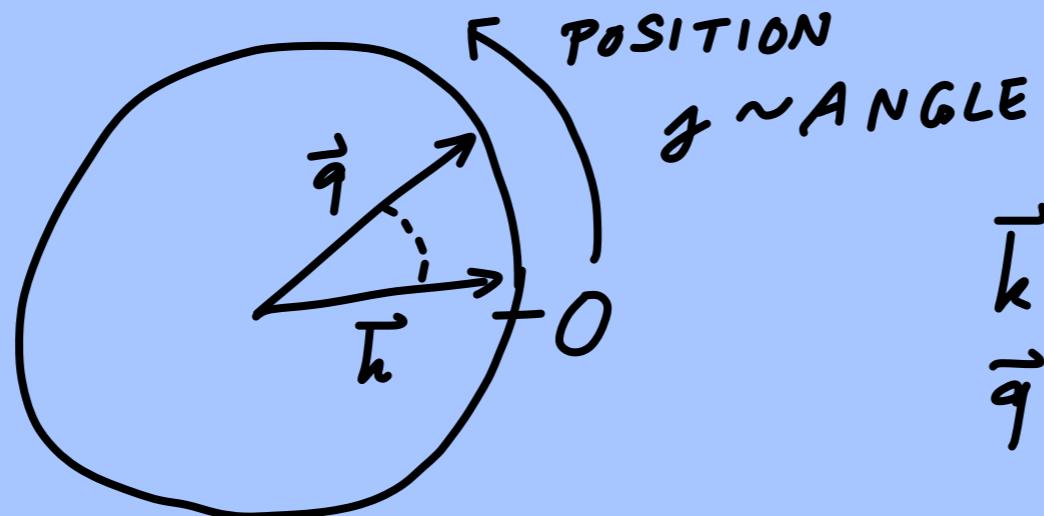
$$h = 8 \text{ HEADS}$$

$$d_{\text{model}} = 512$$

$$d_h = d_v = \frac{d_{\text{model}}}{h} = 64$$

ENCODING POSITION  
COULD BE LEARNED

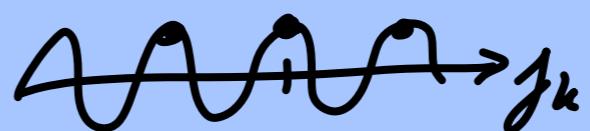
$h = NN(g)$   $\downarrow$  POSITION



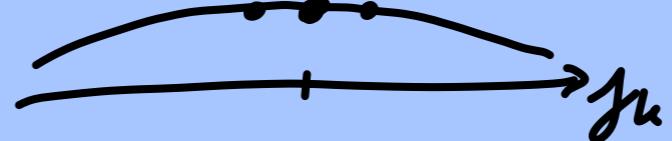
$$\begin{aligned}\vec{q} \cdot \vec{h} &= \cos \angle(\vec{q}, \vec{h}) \\ &= \cos(\omega(j_q - j_h))\end{aligned}$$

CHOICE OF  $\omega^2$

$\omega$  LARGE  $\Rightarrow$

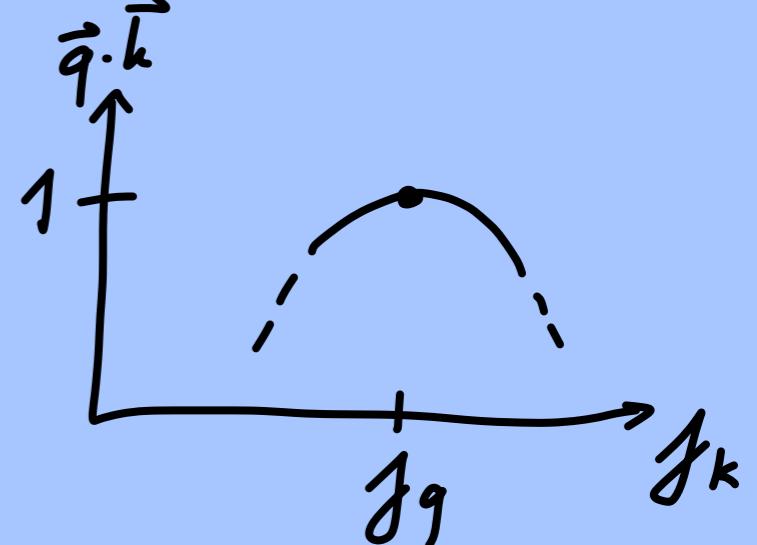


$\omega$  SMALL  $\Rightarrow$



"USE MANY  $\omega$ "

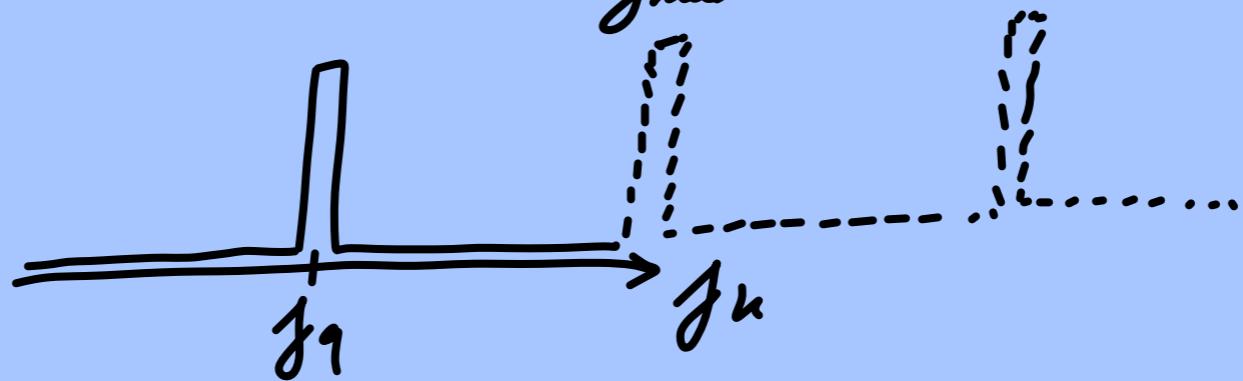
$$\begin{aligned}\vec{k} &= (\cos(\omega j_k), \sin(\omega j_k)) \\ \vec{q} &= \text{LIKEWISE}\end{aligned}$$



$$\left. \begin{array}{l} h_\ell^c = \cos(\omega_e f_k) \\ h_\ell^s = \sin(\omega_e f_k) \end{array} \right\} \text{LONG VECTOR ENCODING POSITION}$$

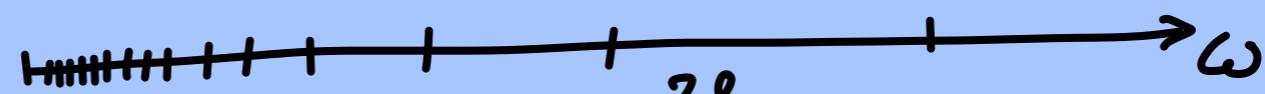
$$\vec{q} \cdot \vec{h} = \sum_{\omega_e} \cos(\omega_e (f_q - f_k))$$

COULD USE:  $\omega_e = \frac{2\pi}{f_{\max}} \cdot l$   $l = 0, 1, 2, \dots, f_{\max} - 1$

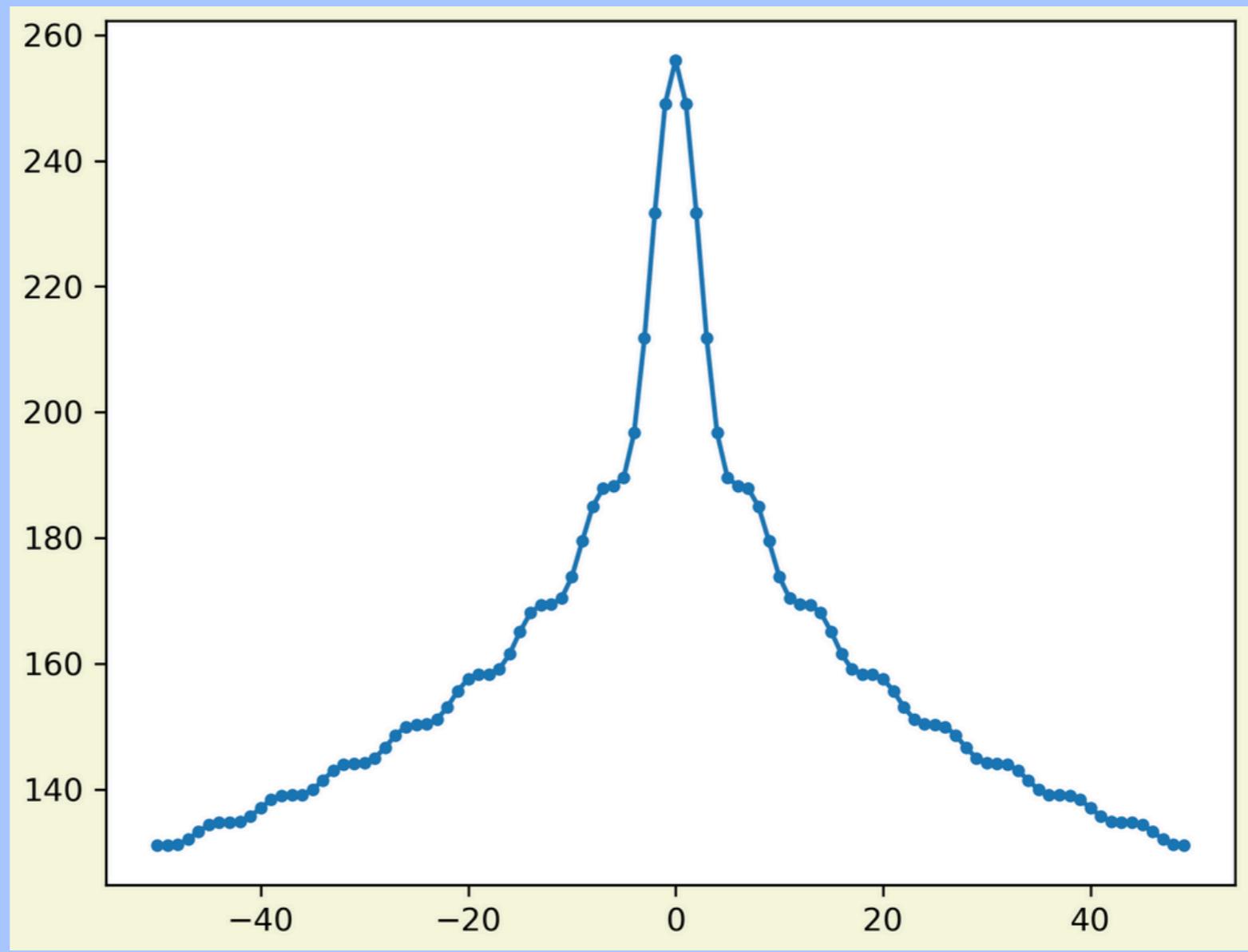


CHOICE OF TRANSFORMER.

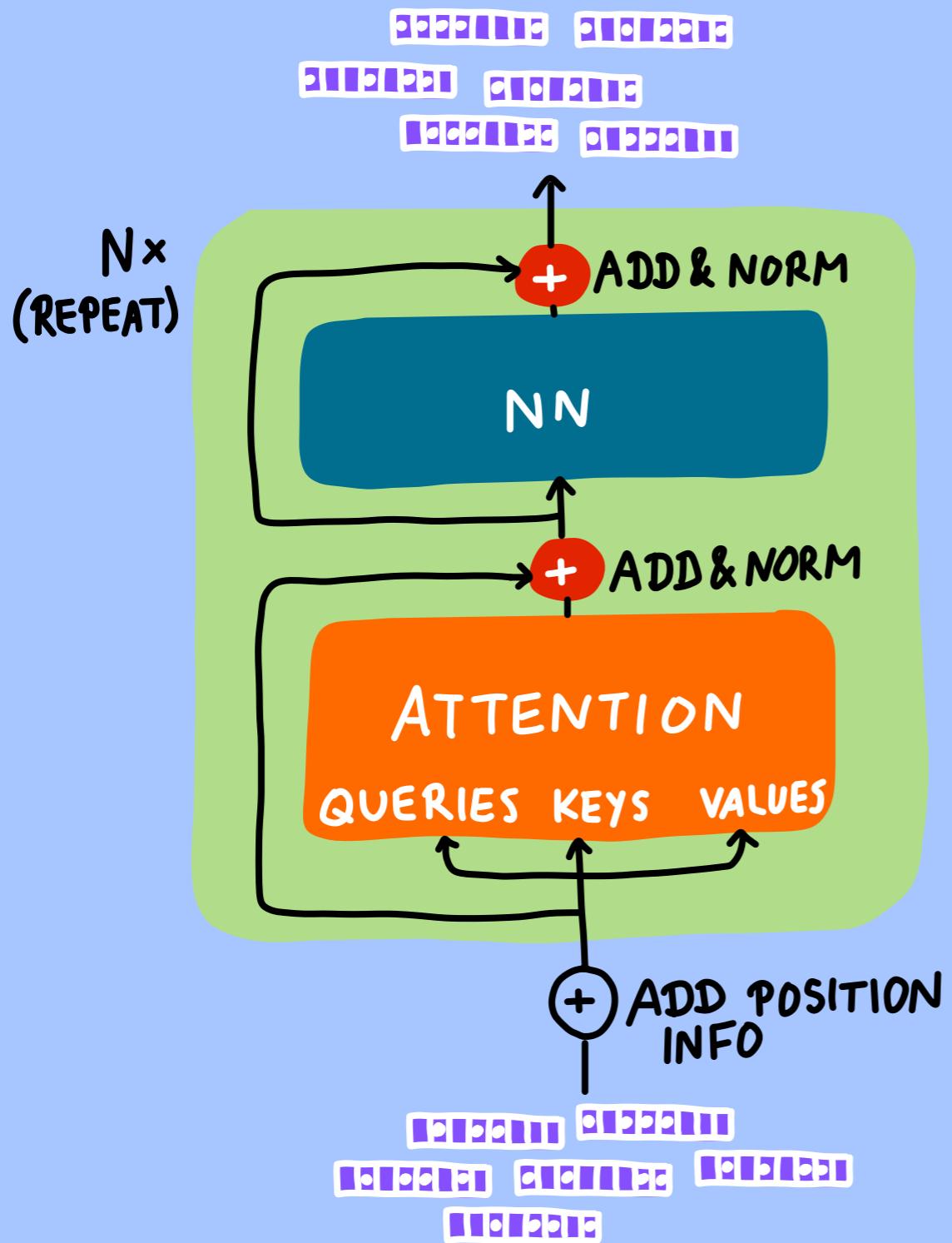
LOGARITHMIC SCALE

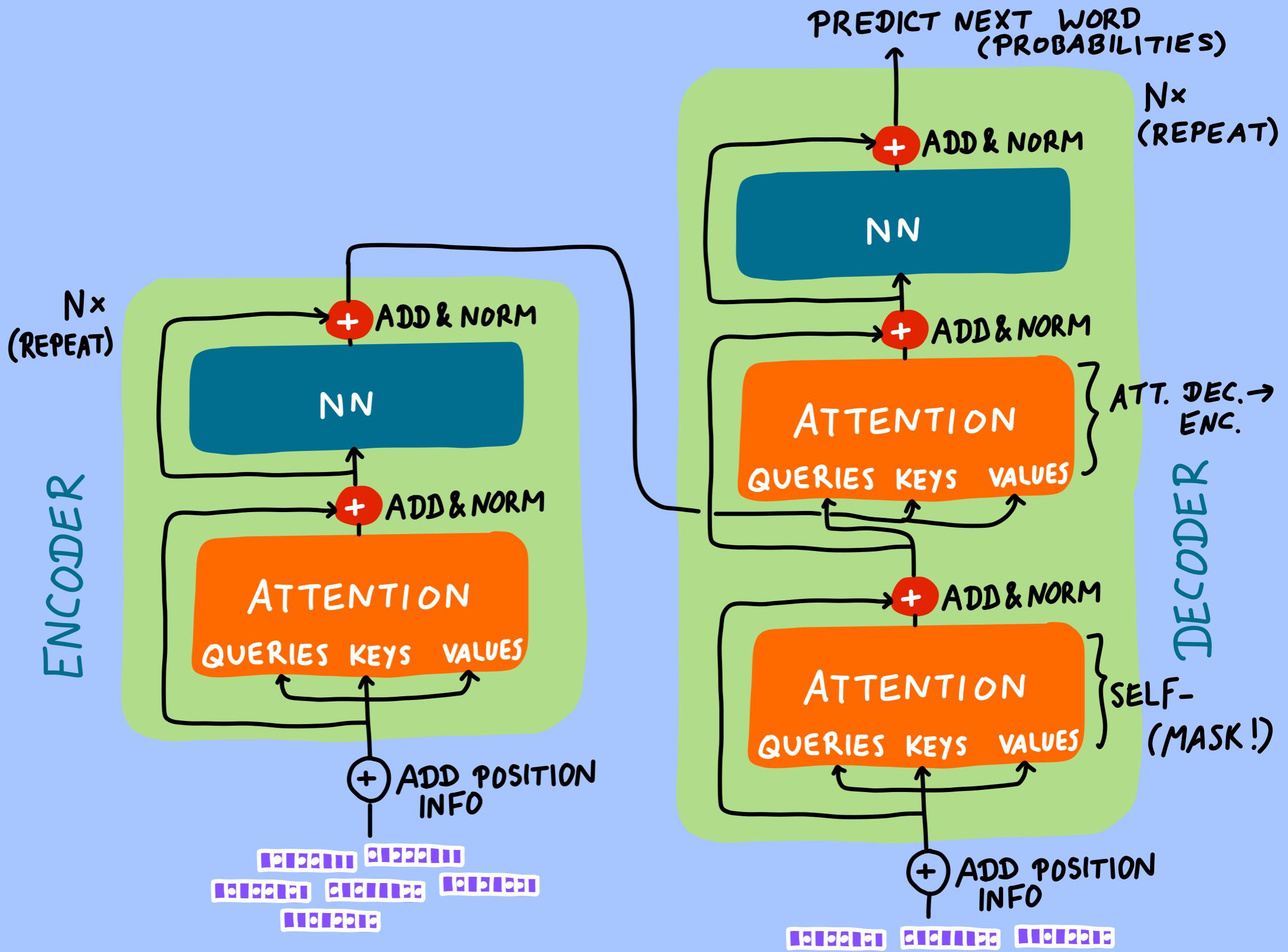


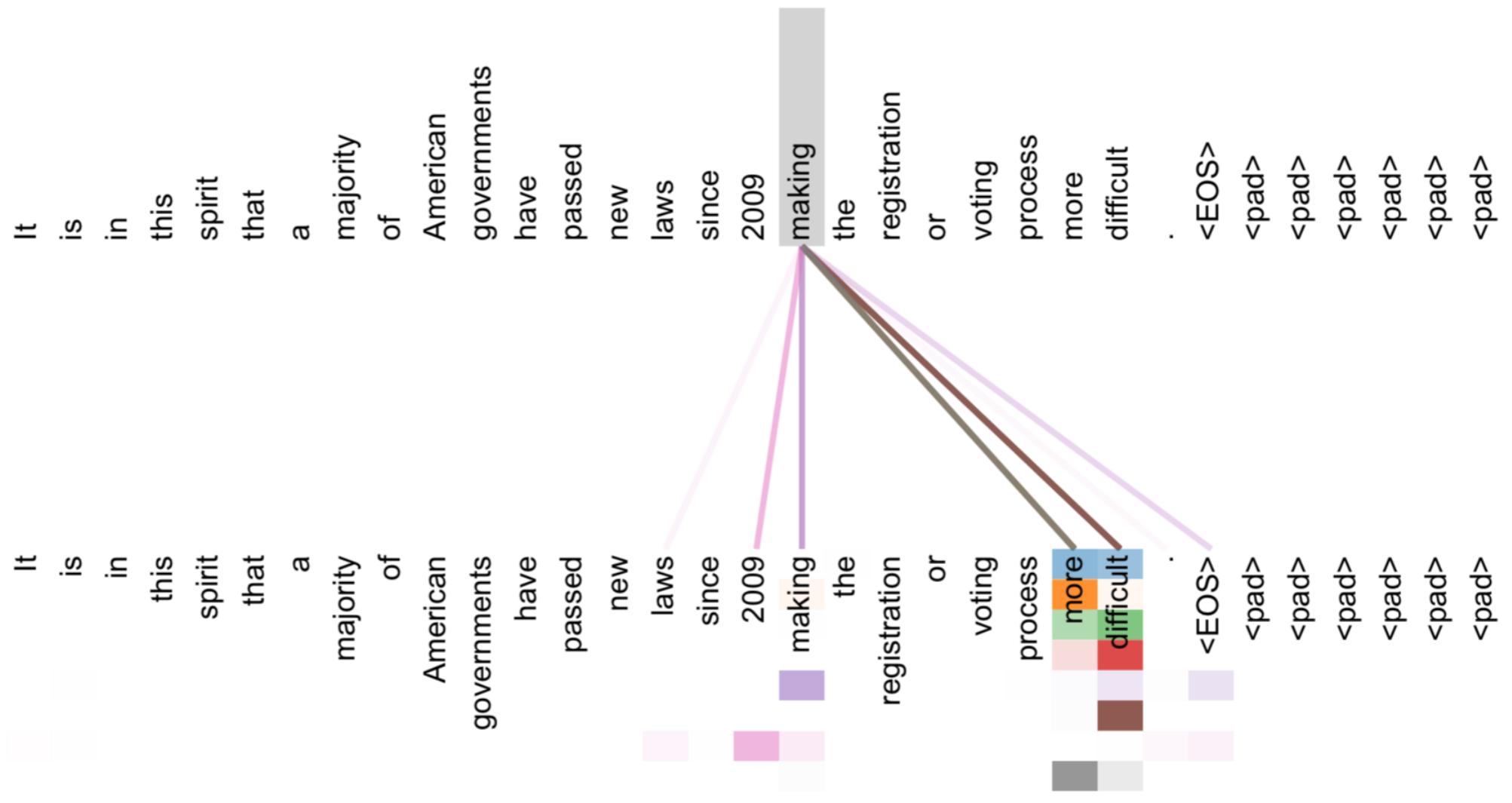
$$\omega_L = [\omega_{\min}]^{\frac{2L}{f_{\max}}} \quad 2L \leq f_{\max}$$

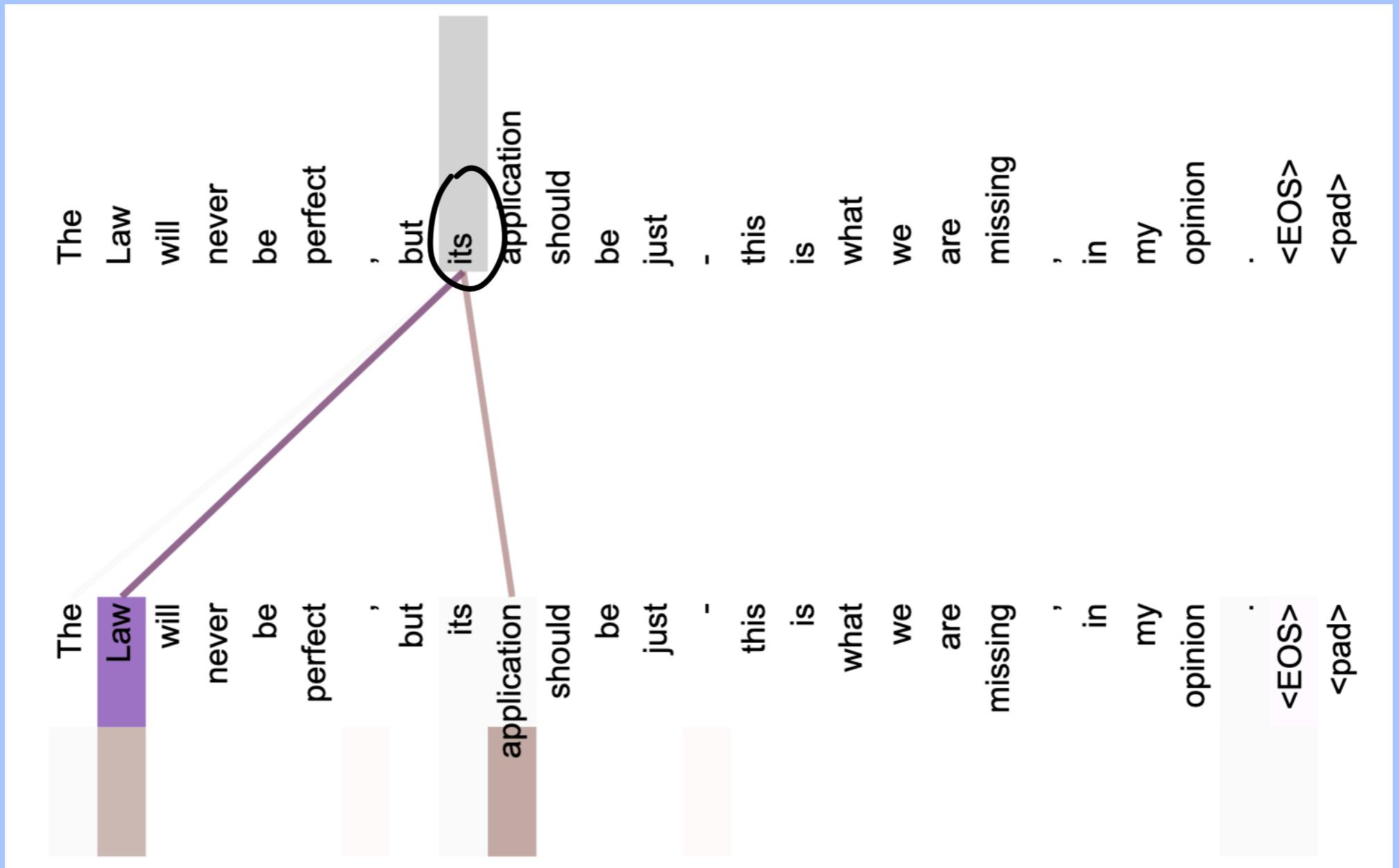


$\mathcal{J}_k - \mathcal{J}_q$









FROM:  
arXiv: 1706.03762

VASWANI ET AL.  
"ATTENTION IS ALL YOU NEED"

# TRAINING OF LANGUAGE TRANSFORMERS

LANGUAGE MODELLING:

PREDICT NEXT WORD (GIVEN PREVIOUS WORDS)  
PREDICT MASKED WORDS

NEXT SENTENCE PREDICTION:

"IS THIS SENTENCE LIKELY?"

ETC.

TASKS:

LANGUAGE UNDERSTANDING (STORY → QUESTIONS)  
QUESTION ANSWERING  
TEXT GENERATION  
TEXT CLASSIFICATION

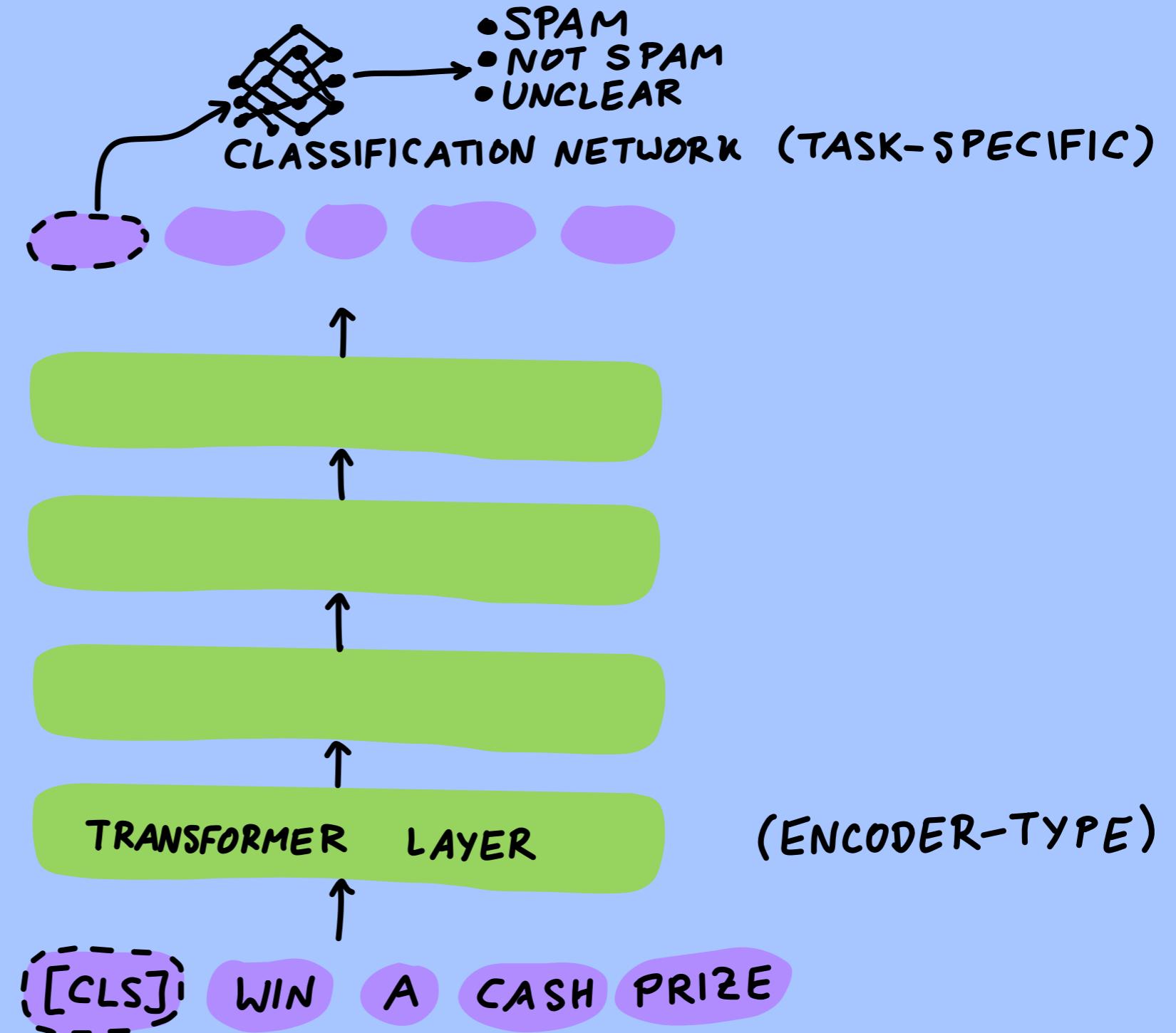
POSSIBILITY : TASK-SPECIFIC NETWORK  
ON TOP OF PRE-TRAINED  
TRANSFORMER MODEL

## EXAMPLE: TEXT CLASSIFICATION

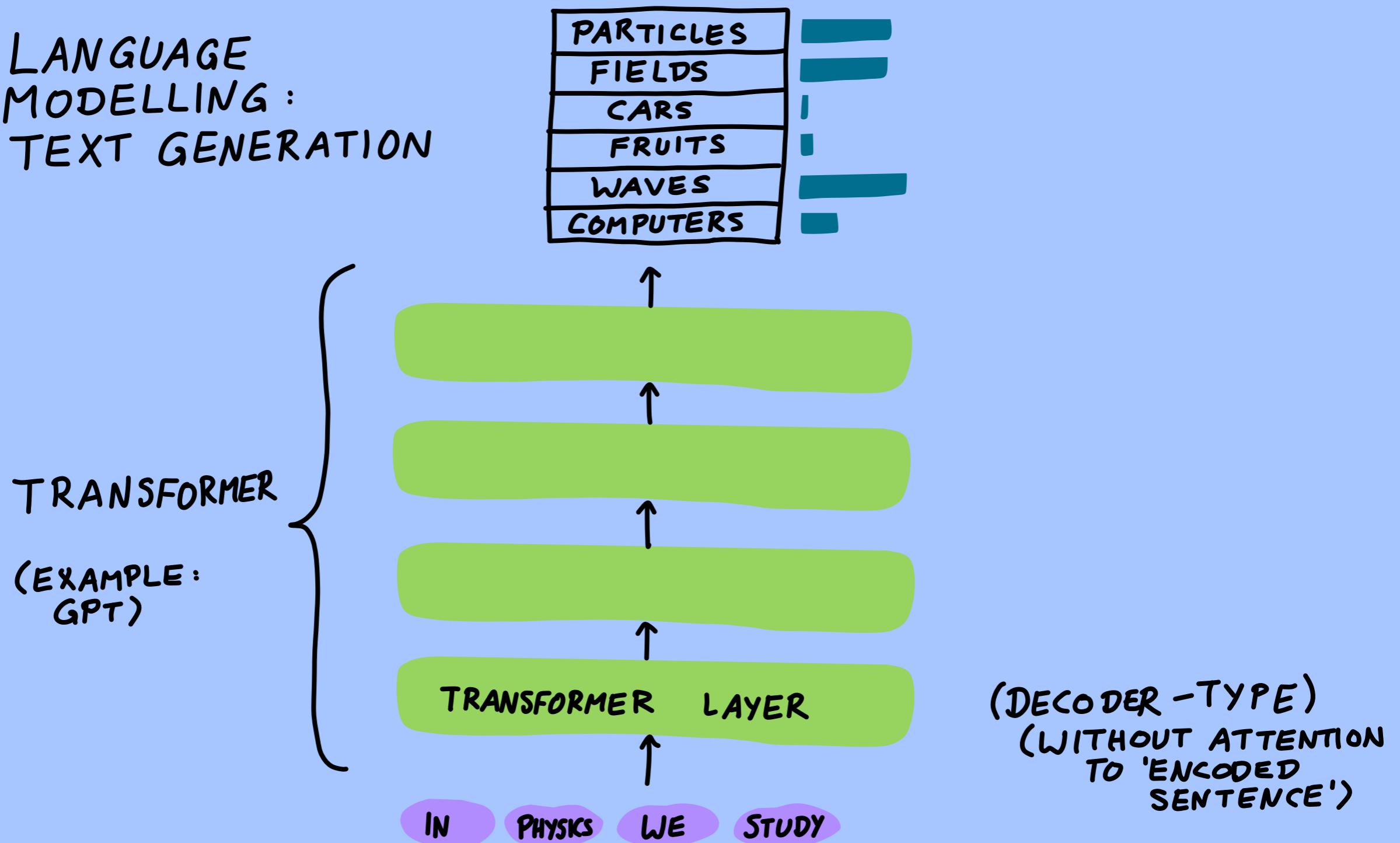
PRETRAINED MODEL +  
CLASSIFIER TRAINING +  
FINE-TUNING

TRANSFORMER

(EXAMPLE:  
BERT)



# LANGUAGE MODELLING : TEXT GENERATION



# DEFINE TASK VIA EXAMPLE! NO FINE-TUNING

A "whatpu" is a small, furry animal native to Tanzania. An example of a sentence that uses the word whatpu is:

We were traveling in Africa and we saw these very cute whatpus.

---

To do a "farduddle" means to jump up and down really fast. An example of a sentence that uses the word farduddle is:

One day when I was playing tag with my little sister, she got really excited and she started doing these crazy farduddles.

A "yalubalu" is a type of vegetable that looks like a big pumpkin. An example of a sentence that uses the word yalubalu is:

I was on a trip to Africa and I tried this yalubalu vegetable that was grown in a garden there. It was delicious.

---

A "Burringo" is a car with very fast acceleration. An example of a sentence that uses the word Burringo is:

In our garage we have a Burringo that my father drives to work every day.

---

A "Gigamuru" is a type of Japanese musical instrument. An example of a sentence that uses the word Gigamuru is:

I have a Gigamuru that my uncle gave me as a gift. I love to play it at home.

---

To "screeg" something is to swing a sword at it. An example of a sentence that uses the word screeg is:

We screeghed at each other for several minutes and then we went outside and ate ice cream.

---

FROM : GPT-3 PAPER / OPEN-AI

arXiv: 2005.14165

BROWN ET AL. "LANGUAGE MODELS  
ARE FEW-SHOT LEARNERS"

---

Context →

The City  
BY C. P. CAVAFY  
TRANSLATED BY EDMUND KEELEY  
[Poem text omitted]

SOME TREES  
John Ashbery  
[Poem text omitted]

Shadows on the Way  
Wallace Stevens

---

----- Generated Poem 1 -----

I must have shadows on the way  
If I am to walk I must have  
Each step taken slowly and alone  
To have it ready made

And I must think in lines of grey  
To have dim thoughts to be my guide  
Must look on blue and green  
And never let my eye forget  
That color is my friend  
And purple must surround me too

The yellow of the sun is no more  
Intrusive than the bluish snow  
That falls on all of us. I must have  
Grey thoughts and blue thoughts walk with me

----- Generated Poem 3 -----

The sun was all we had. Now, in the shade  
All is changed. The mind must dwell on those  
White fields, that to its eyes were always old;  
Those ancient gleams, convoluting  
The way they lay among the huge roots,  
The great dark tomes of reverie,  
The plumed labyrinth of the sea.

When our own breath gonme faint and fade  
Up the long stairway of the dead  
We shall remember well  
The blindness of the light, the whiteness  
Of that white land. The footsteps, and  
The grasses in that light, and in the shade  
The menacing life.

FROM : GPT-3 PAPER / OPEN-AI

arXiv: 2005.14165  
BROWN ET AL. "LANGUAGE MODELS  
ARE FEW-SHOT LEARNERS"

## WRITE PROGRAMS BASED ON TASK DESCRIPTION

```
1 import datetime
2
3 def parse_expenses(expenses_string):
4     """Parse the list of expenses and return the list of triples (date, value, currency).
5     Ignore lines starting with #.
6     Parse the date using datetime.
7     Example expenses_string:
8         2016-01-02 -34.01 USD
9         2016-01-03 2.59 DKK
10        2016-01-03 -2.72 EUR
11    """
12     expenses = []
13     for line in expenses_string.splitlines():
14         if line.startswith("#"):
15             continue
16         date, value, currency = line.split(" ")
17         expenses.append((datetime.datetime.strptime(date, "%Y-%m-%d"),
18                           float(value),
19                           currency))
```

GITHUB COPILOT  
[copilot.github.com](https://copilot.github.com)

BERT / GOOGLE

TRAINED ON  
WIKIPEDIA  $2.5 \cdot 10^9$  WORDS  
& BOOKS  $800 \cdot 10^6$  WORDS

345 MILLION PARAMETERS  
(BERT-LARGE)

GPT-3 / OPEN-AI

TRAINED ON  
 $\sim 500 \cdot 10^9$  WORDS (TOKENS)

175 BILLION PARAMETERS

# IMAGE GENERATION FROM TEXT PROMPTS

## TEXT PROMPT

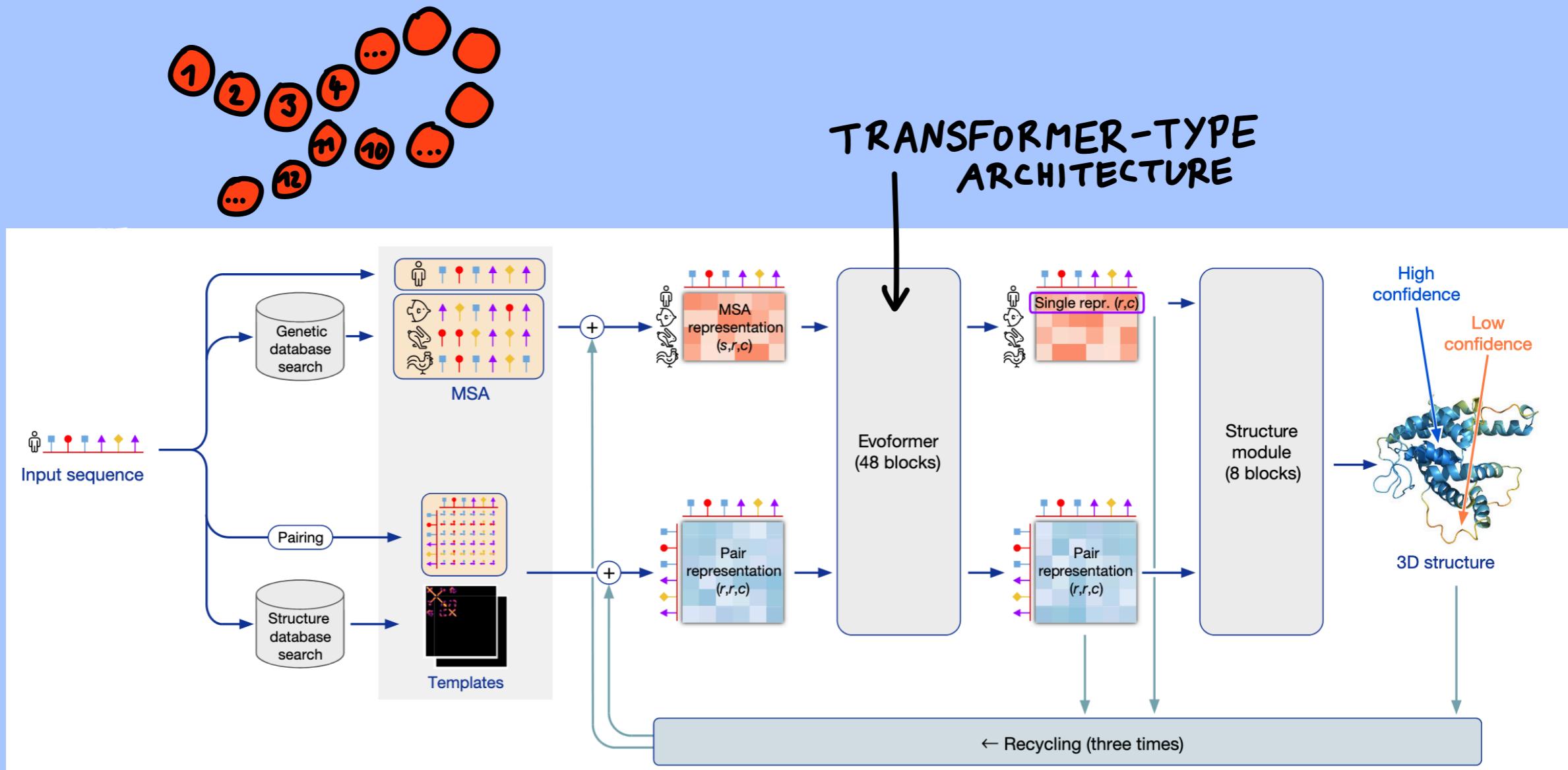
a penguin made of cabbage. a penguin with the texture of a cabbage.

## AI-GENERATED IMAGES



FROM: OpenAI DALL-E WEBSITE  
[openai.com/blog/dall-e](https://openai.com/blog/dall-e)

# PROTEIN STRUCTURE PREDICTION



ALPHA-FOLD 2 (DeepMind)

PICTURE FROM: "HIGHLY ACCURATE PROTEIN STRUCTURE PREDICTION USING ALPHA FOLD"  
JUMPER ET AL.  
NATURE 596, 583 (2021)

# SYMBOLIC MATHEMATICS

## (INTEGRALS, DIFF. EQUATIONS,...)

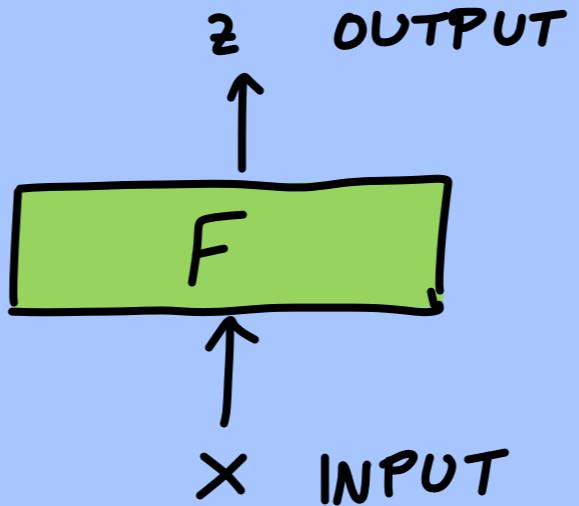
Equation	Solution
$y' = \frac{16x^3 - 42x^2 + 2x}{(-16x^8 + 112x^7 - 204x^6 + 28x^5 - x^4 + 1)^{1/2}}$	$y = \sin^{-1}(4x^4 - 14x^3 + x^2)$
$3xy \cos(x) - \sqrt{9x^2 \sin(x)^2 + 1}y' + 3y \sin(x) = 0$	$y = c \exp(\sinh^{-1}(3x \sin(x)))$
$4x^4yy'' - 8x^4y'^2 - 8x^3yy' - 3x^3y'' - 8x^2y^2 - 6x^2y' - 3x^2y'' - 9xy' - 3y = 0$	$y = \frac{c_1 + 3x + 3 \log(x)}{x(c_2 + 4x)}$

LAMPLE, CHARTON  
ICML 2020

"DEEP LEARNING FOR  
SYMBOLIC MATHEMATICS"

## 8.5

## DEEP IMPLICIT LAYERS

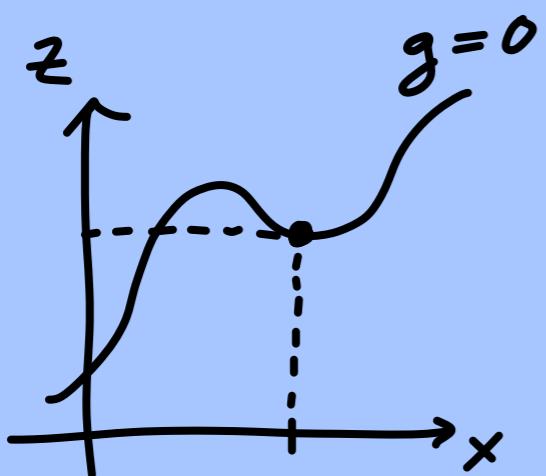


$$z = F_{\theta}(x)$$

BUT NOW:  $F$  = GENERAL OPERATION  
LIKE

- SOLVE EQ. THAT DEPENDS ON  $x$   
→ SOLUTION  $z$
- DIFF.EQ.
- OPTIM. PROBLEM

## EXAMPLE 1 : EQUATION SOLVING



$$g(z, x) \stackrel{!}{=} 0$$

FIND  $z$  AT FIXED  $x$   
⇒ SOLUTION

$z^*(x)$  = OUTPUT OF IMPLICIT LAYER

FOR GRADIENT DESC.:  
NEED DERIVATIVES

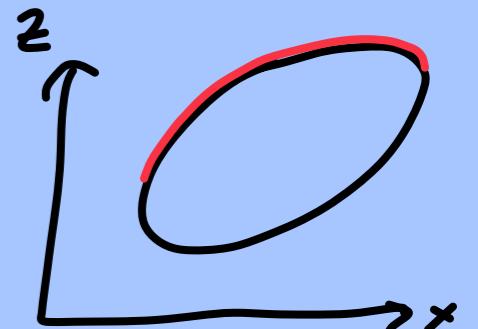
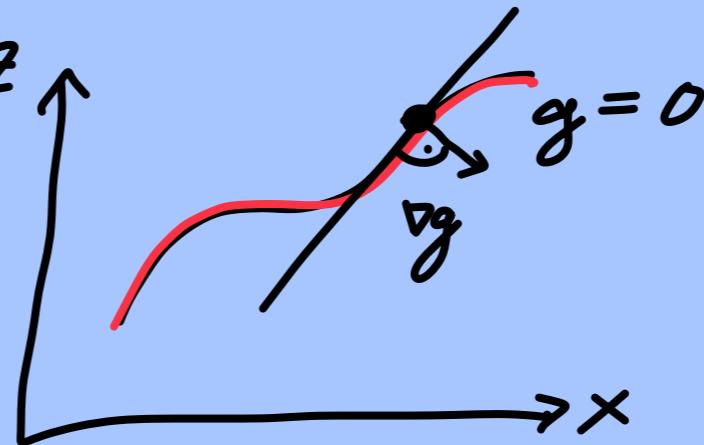
NAIVE WAY: WRITE SOLVER  
USING TF/ PYTORCH

→ SLOW / MEMORY - INTENSIVE

# IMPLICIT FUNCTION THEOREM

LOCALLY GET DIFFERENTIABLE

FCT  $z^*(x)$   $z$



$$\parallel g(z^*(x), x) = 0 \quad \forall x$$

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

$$\frac{\partial g}{\partial z} \frac{\partial z^*(x)}{\partial x} + \frac{\partial g}{\partial x} = 0$$

$$\underbrace{\frac{\partial g}{\partial z}}_{\text{MATRIX}} \frac{\partial z^*(x)}{\partial x} + \frac{\partial g}{\partial x} = 0$$

MATRIX

DERIVATIVE  
W. RESPECT TO FIRST  
ARG.

$$\frac{\partial z^*(x)}{\partial x} = - \left( \frac{\partial g}{\partial z} \right)^{-1} \left( \frac{\partial g}{\partial x} \right)$$

AVAILABLE!

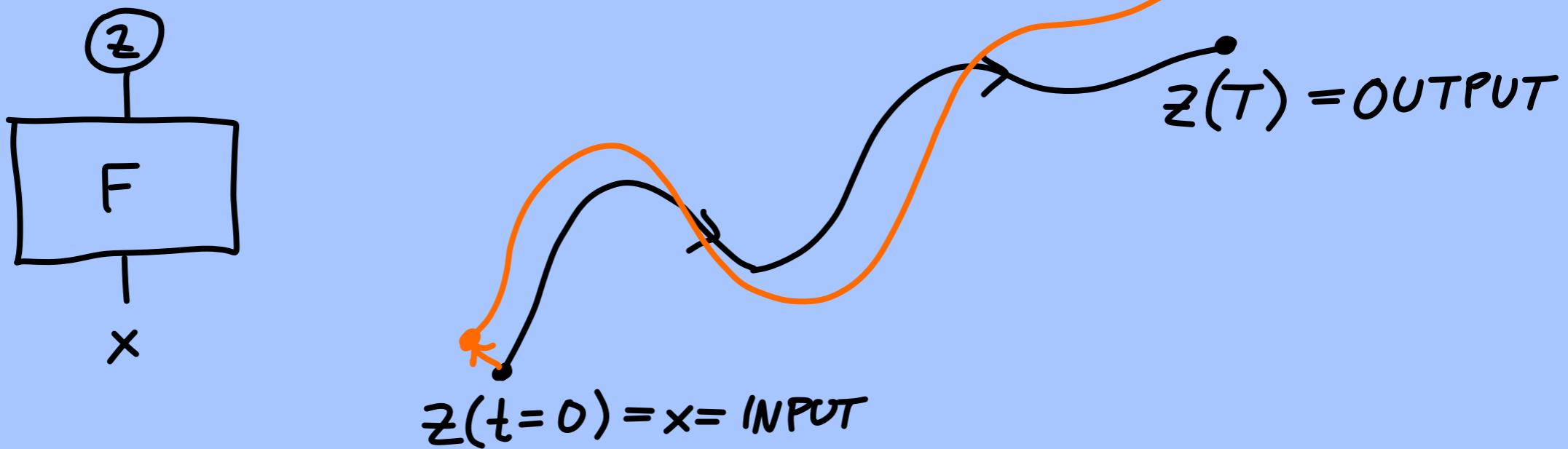
$\frac{\partial \text{OUTPUT}}{\partial \text{INPUT}}$  IS KNOWN!

$$g = NN = g_\theta(z, x)$$

RECIPE:

1. USE ANY ALG. TO FIND  $z^*(x)$
2.  $\frac{\partial z^*}{\partial x}$  = SEE ABOVE

## EXAMPLE 2: NEURAL ODEs



$$\frac{\partial z}{\partial t} = f_{\theta}(z, t)$$

$\xrightarrow{\text{NN}}$

COULD IMPLEMENT DIFF. EQ. SOLVER IN TF  
 $\rightarrow$  NOT EFFICIENT

1. USE ANY EFFIC. SOLVER !

2. GET  $\frac{\partial z(T)}{\partial x}$

$$z = z(x, t)$$

$$z(x, t=0) \equiv x$$

$$\frac{\partial}{\partial x} \frac{\partial z}{\partial t} = \frac{\partial f_0}{\partial x}(z(x, t), t)$$

$$\frac{\partial}{\partial t} \left( \frac{\partial z}{\partial x} \right) = \frac{\partial f_0}{\partial z} \left( \frac{\partial z}{\partial x} \right)$$

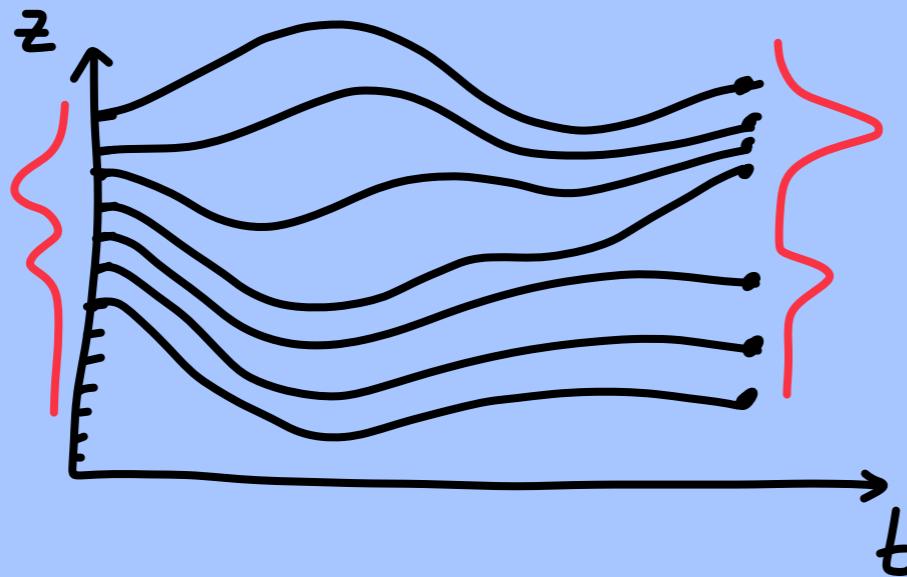
FUNCTION OF  $t$

LINEAR DIFF. EQ. FOR  $\frac{\partial z}{\partial x}$

$$\frac{\partial z}{\partial x}(x, t=0) = 1$$

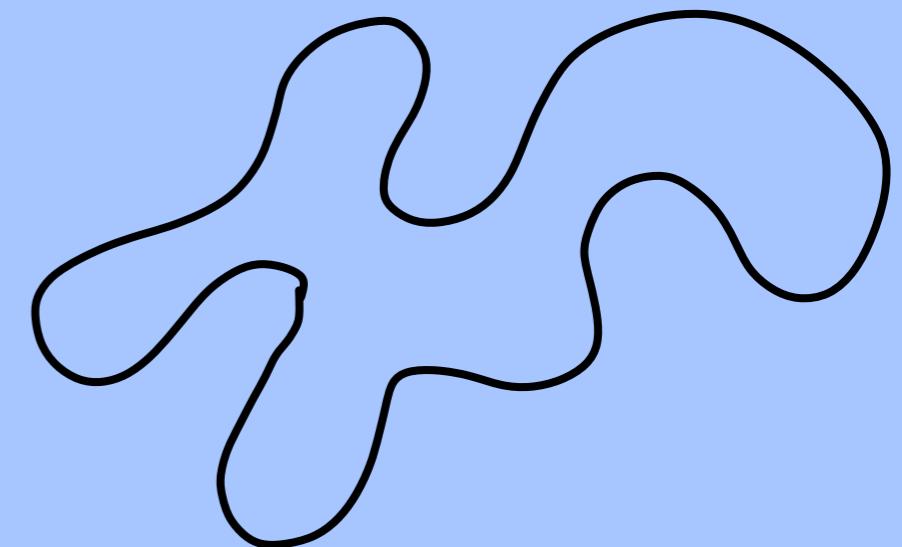
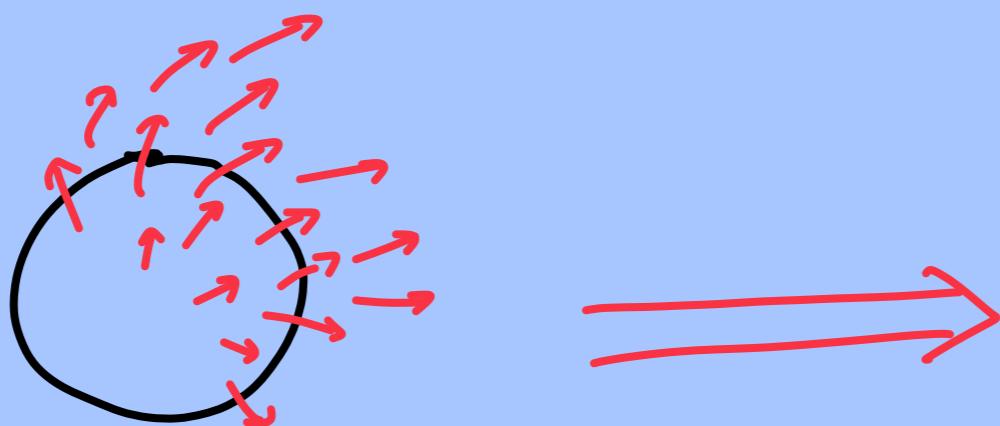
$\Rightarrow$  SOLVE NUMER.  
TO GET  $\frac{\partial z(x, t)}{\partial x}$

MODEL SMOOTH FLOW:



⇒ INVERTIBLE

→ NORMALIZING  
FLOWS!



$$(u, v) \rightarrow \vec{r}_\theta(u, v)$$

OPTIMIZATION

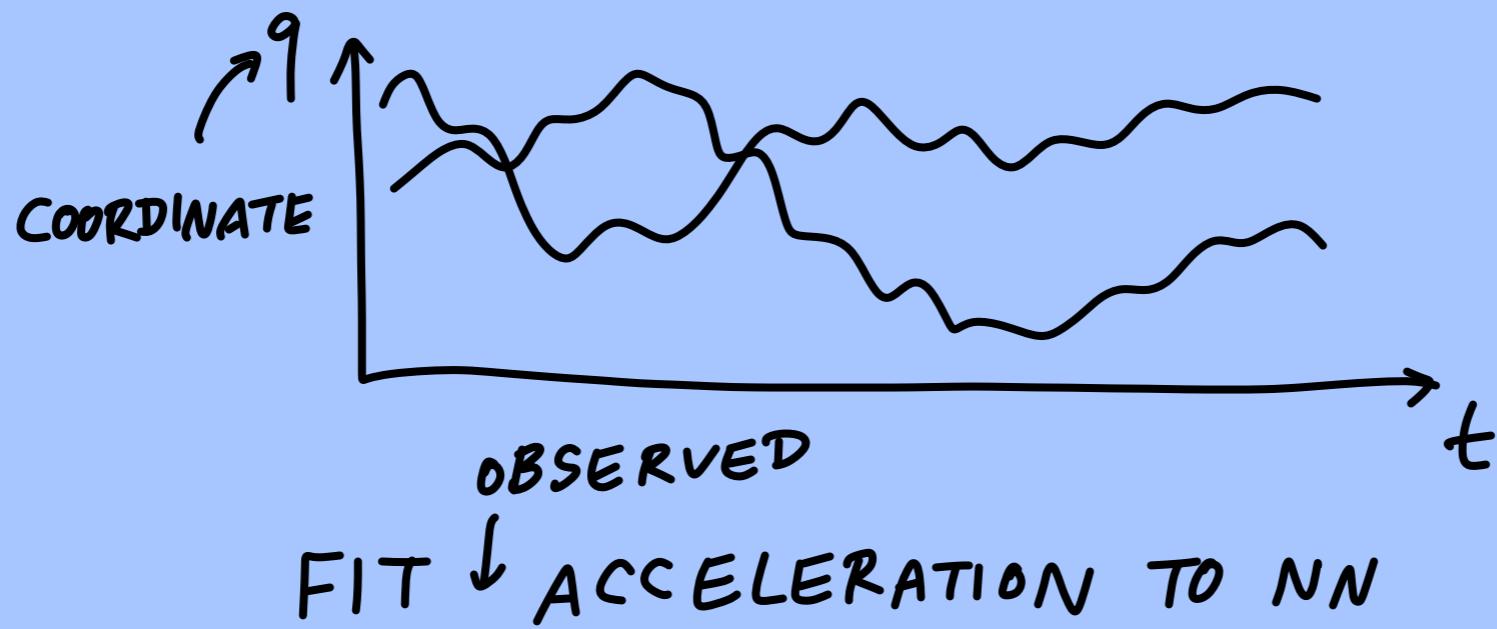
$$z^* = \underset{z}{\operatorname{argmin}} \quad f_{\theta}(x, z)$$

↑ SCALAR  
↓ INPUT

EIGENVALUES :

OUTPUT  $z$  = FIRST  $n$  EIGENVALUES  
OF  $M_{\theta}(x)$

## 8.6

HAMILTONIAN &  
LAGRANGIAN NETWORKS

$$\ddot{q}(t) \stackrel{!}{\approx} f_{\theta}(q(t), \dot{q}(t), t)$$

OBSERVED

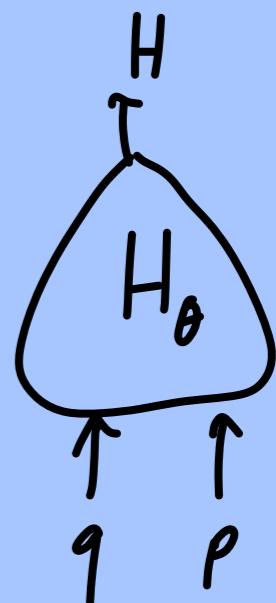
BUT : APPROX. MAY  
VIOLATE E.G. ENERGY CONSERVATION

# HAMILTONIAN NETWORKS

$$H_\theta(q, p) \xrightarrow{\text{MOMENTA}}$$

$$\dot{q}_{\text{OBS}} \stackrel{!}{\approx} \dot{q} = \frac{\partial H_\theta(q, p)}{\partial p}$$

$$\dot{p}_{\text{OBS}} \stackrel{!}{\approx} \dot{p} = - \frac{\partial H_\theta(q, p)}{\partial q}$$



MINIMIZE

$$\left( \dot{q}_{\text{OBS}} - \frac{\partial H_\theta}{\partial p} \right)^2 + \left( \dot{p}_{\text{OBS}} + \frac{\partial H_\theta}{\partial q} \right)^2$$

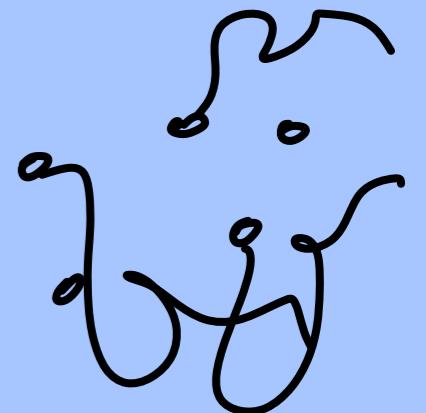
PROBLEM: SOMETIMES  $p$  UNKNOWN

- MAGNETIC FIELD
- RELATIVISTIC PARTICLE
- COMPLICATED COORDS

# LAGRANGIAN NETWORKS

$$L_\theta(q, \dot{q}, t)$$

$$\frac{d}{dt} \frac{\partial L_\theta}{\partial \dot{q}_j} = \frac{\partial L_\theta}{\partial q_j}$$



$$\frac{\partial^2 L_\theta}{\partial t \partial \dot{q}_j} + \sum_k \dot{q}_k \frac{\partial^2 L_\theta}{\partial q_k \partial \dot{q}_j} + \sum_k \ddot{q}_k \frac{\partial^2 L_\theta}{\partial \dot{q}_k \partial \dot{q}_j} = \frac{\partial L_\theta}{\partial q_j}$$

$$\ddot{q} = \left[ \frac{\partial^2 L}{\partial \dot{q} \partial \dot{q}} \right]^{-1} \left\{ \frac{\partial L}{\partial q} - \frac{\partial^2 L}{\partial t \partial \dot{q}} - \dot{q}^T \frac{\partial^2 L}{\partial q \partial \dot{q}} \right\}$$

↓

HESSIAN

⇒ AUTOMATIC DIFF.



# THIS LECTURE SERIES

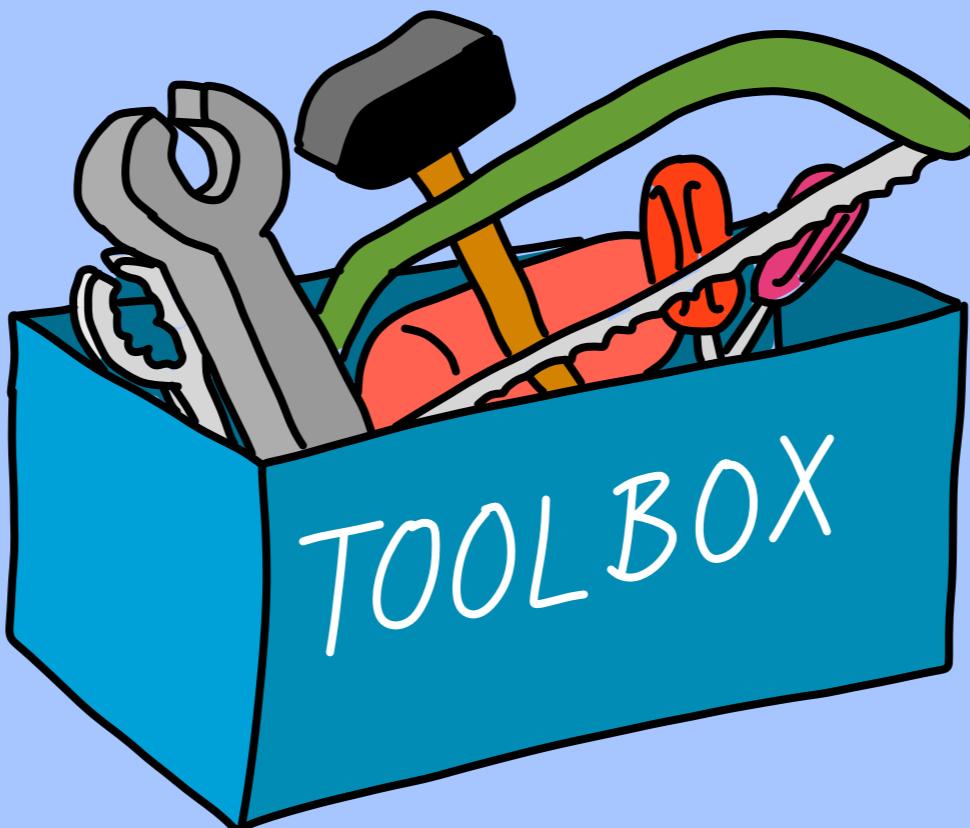
ARTIFICIAL  
NEURAL ✓  
NETWORKS

BAYES ✓

✓ INFORMATION  
THEORY

✓ REPRESENTATION  
LEARNING

✓ ADVANCED  
NN STRUCTURES



✓ LEARNING  
PROBABILITY  
DISTRIBUTIONS

DISCOVERING  
STRATEGIES ✓!

ADAPTIVE  
OBSERVATIONS

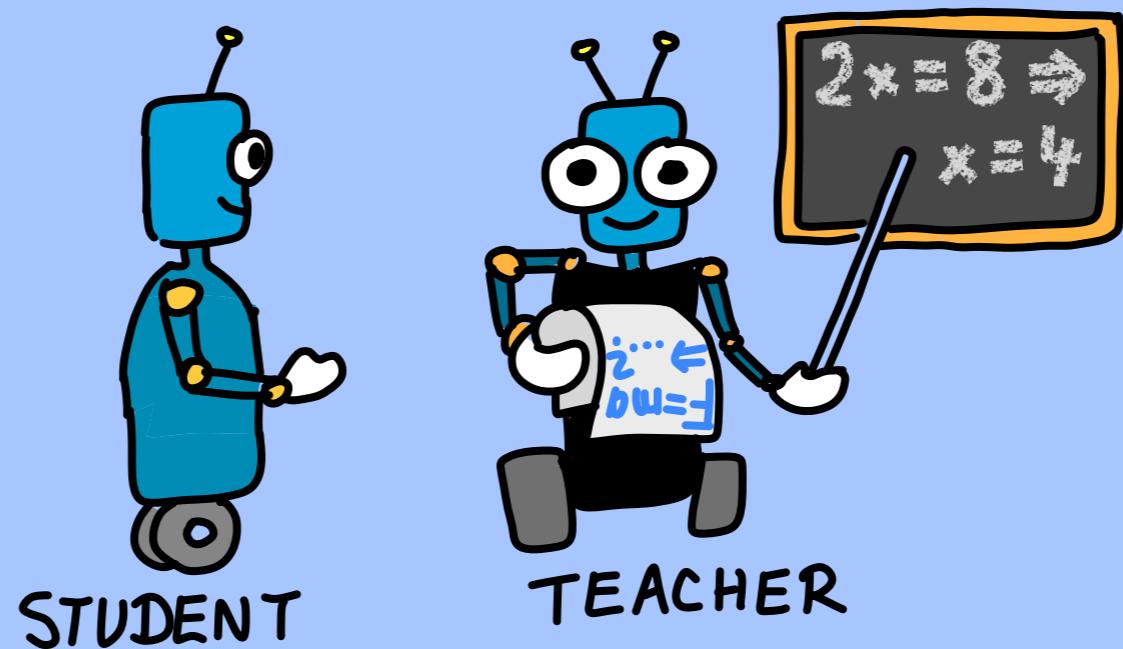
MEASURING  
COMPLEXITY

g.

# REINFORCEMENT LEARNING

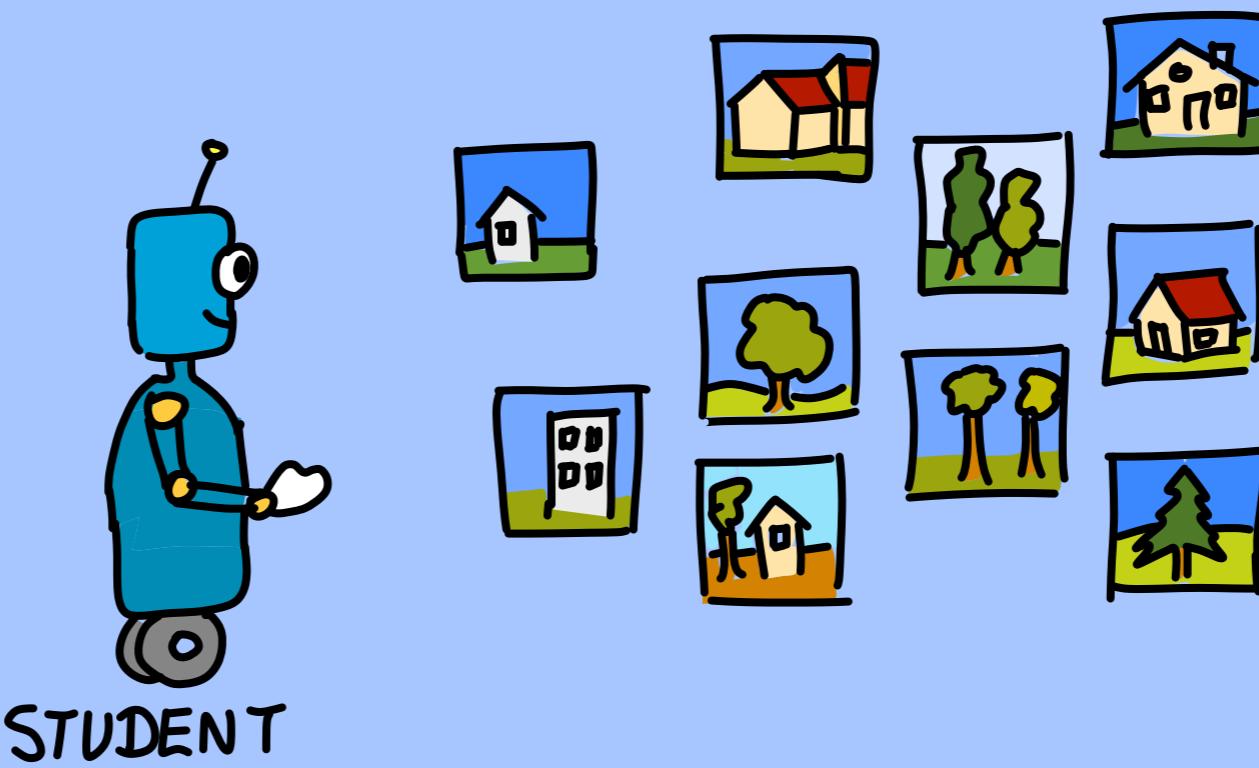
DISCOVERING  
STRATEGIES

# SUPERVISED LEARNING



CORRECT ANSWERS KNOWN

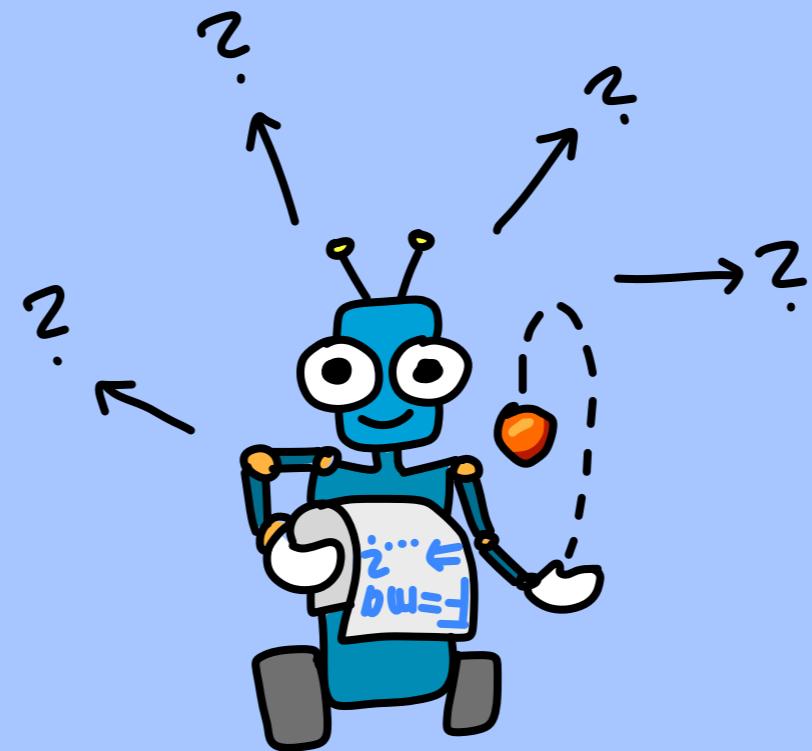
# UNSUPERVISED / SELF-SUPERVISED LEARNING



STUDENT

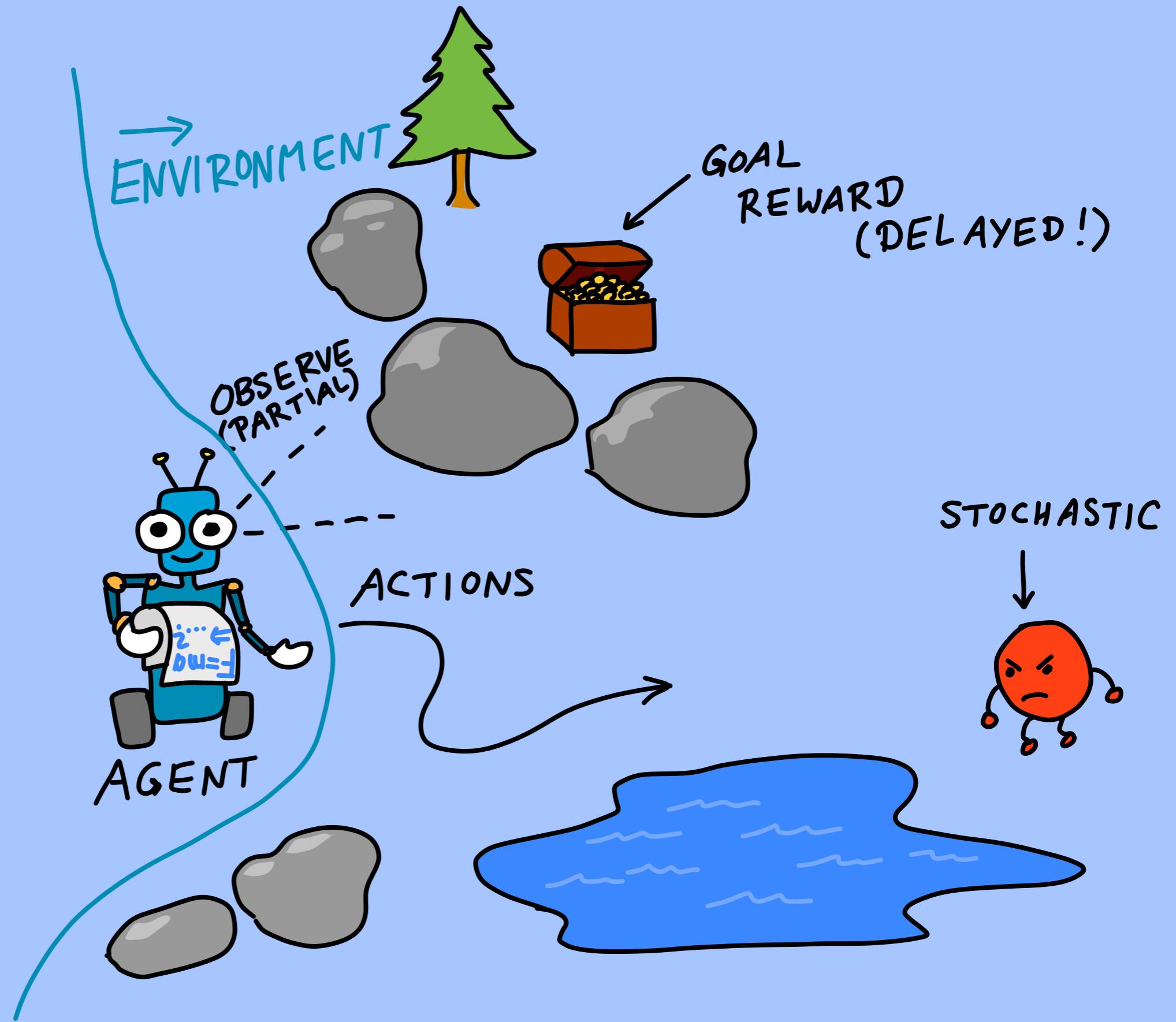
NO LABELS (ANSWERS)  
NEEDED  
RECOGNIZING & REPRODUCING  
PATTERNS

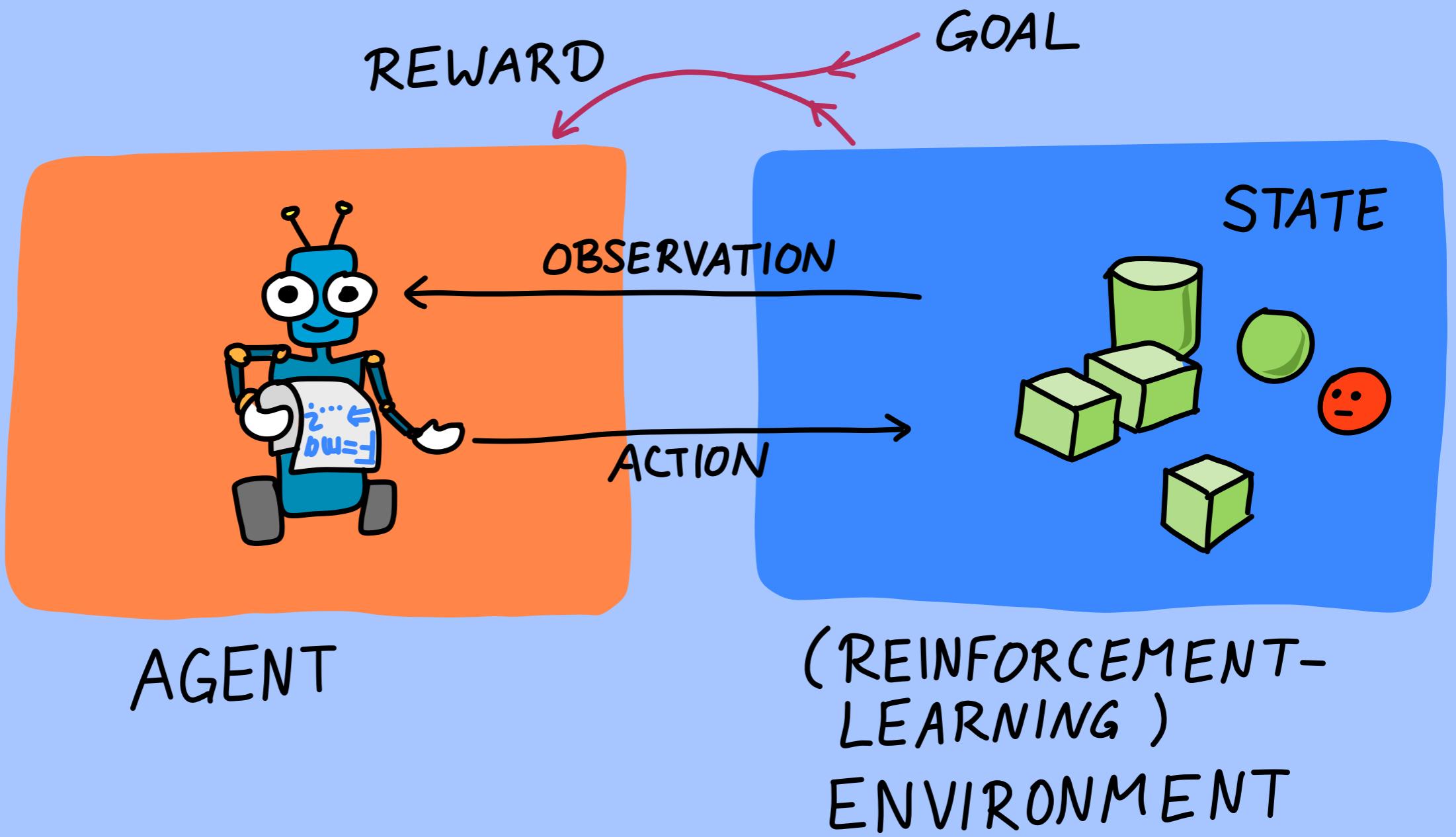
# REINFORCEMENT LEARNING

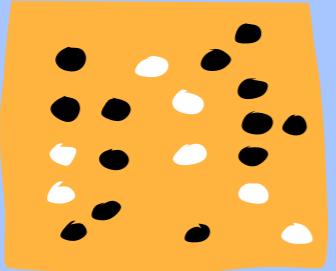


DISCOVER A STRATEGY FOR REACHING A GOAL

TRIAL & ERROR



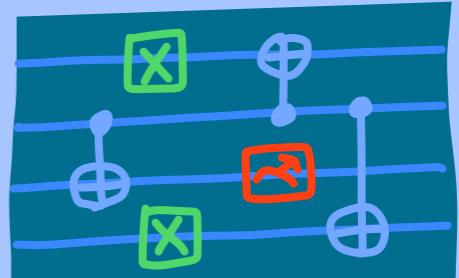




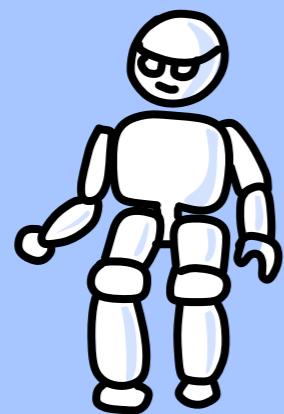
PLAYING GAMES



INVESTMENT STRATEGIES



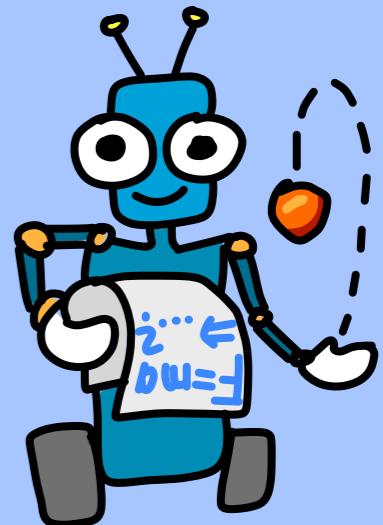
QUANTUM CONTROL STRATEGIES



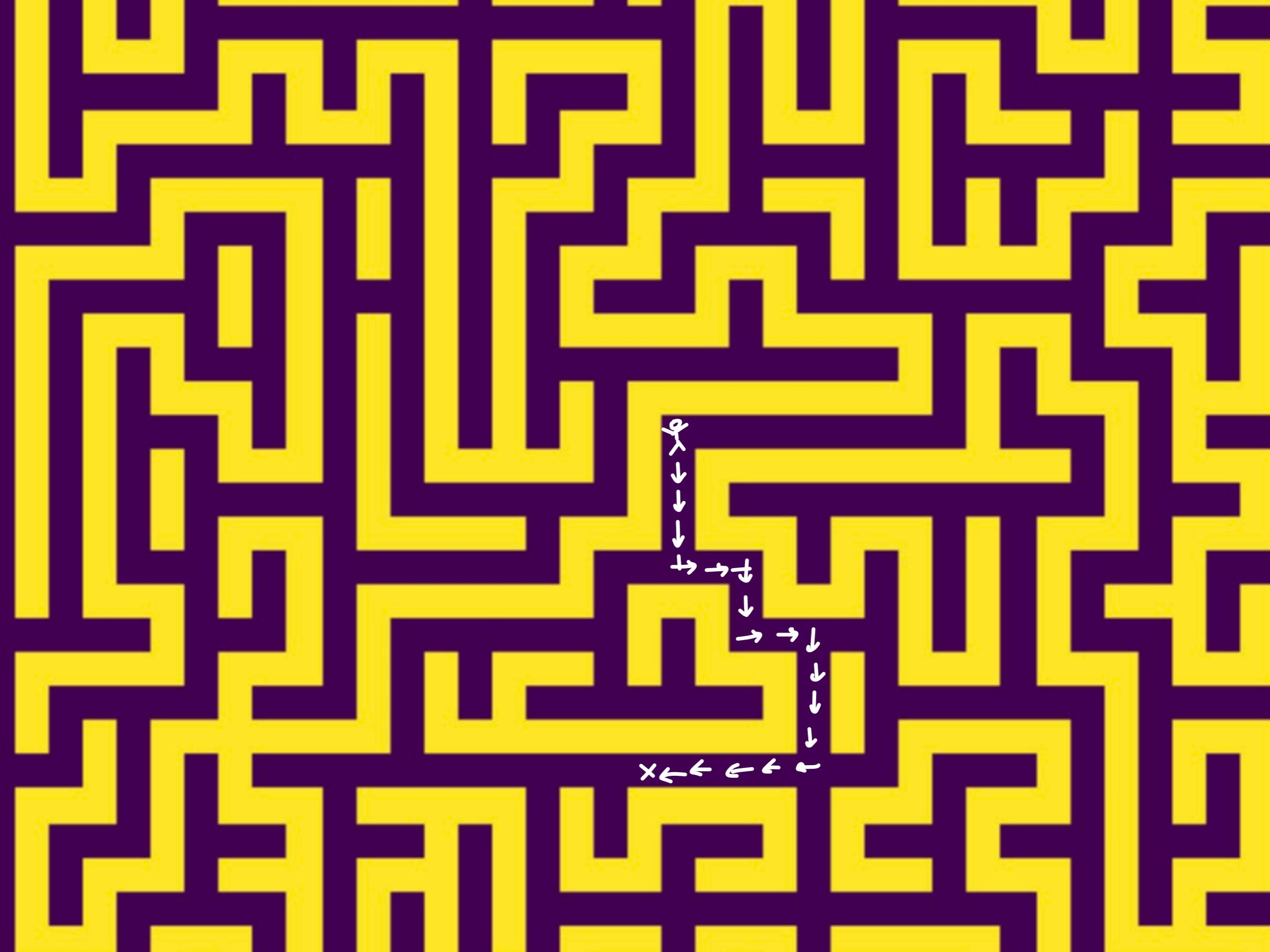
ROBOTIC MOTION

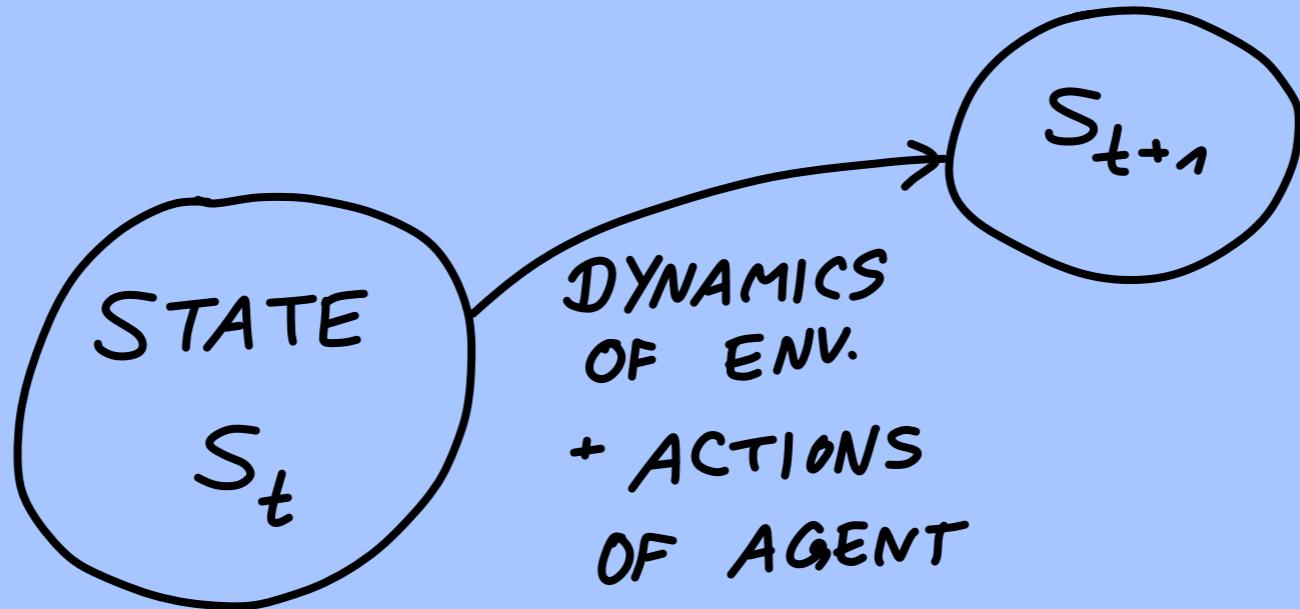


SELF-DRIVING CARS



SCIENTIFIC DISCOVERY STRATEGIES ?  
GENERAL AI ?





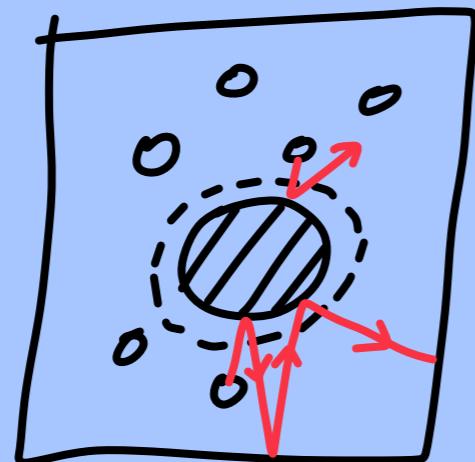
DYNAMICS OF ENVIRONMENT  
(STOCHASTIC)  
MARKOV!

$P(S_{t+1} | S_t, a_t)$

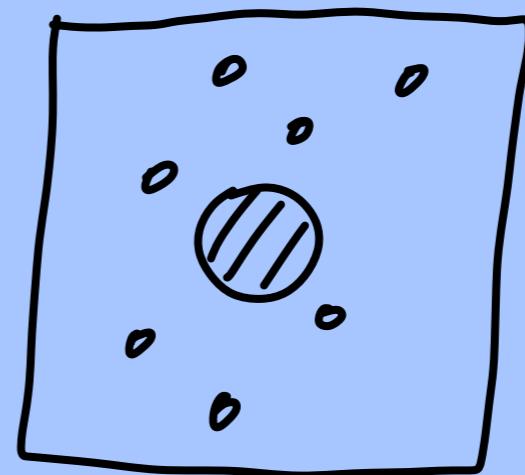
STATE TRANSITION PROBABILITIES

ACTION OF AGENT

IF NOT MARKOV:  
ENLARGE STATE SPACE  
(CONCEPTUALLY)



NON-MARKOVIAN



MARKOV

AGENT:  $a_t$  SELECTED  
 ACCORDING TO  
 STRATEGY/ POLICY (BASED ON  
 CURRENT KNOWL.)

$$\pi(a_t | s_t) = \begin{matrix} \text{PROBAB.} \\ \text{FOR } a_t \\ \text{GIVEN } s_t \end{matrix}$$

ACTION      STATE

↑  
POLICY

ABOUT 'STATE':

- PARTIAL OBSERVATION :

$s_t$  = FULL STATE OF ENV.

$$\pi(a_t | s_t) = \pi(a_t | \text{OBS}(s_t))$$

$\hookrightarrow$  OBSERVABLE  
BY AGENT

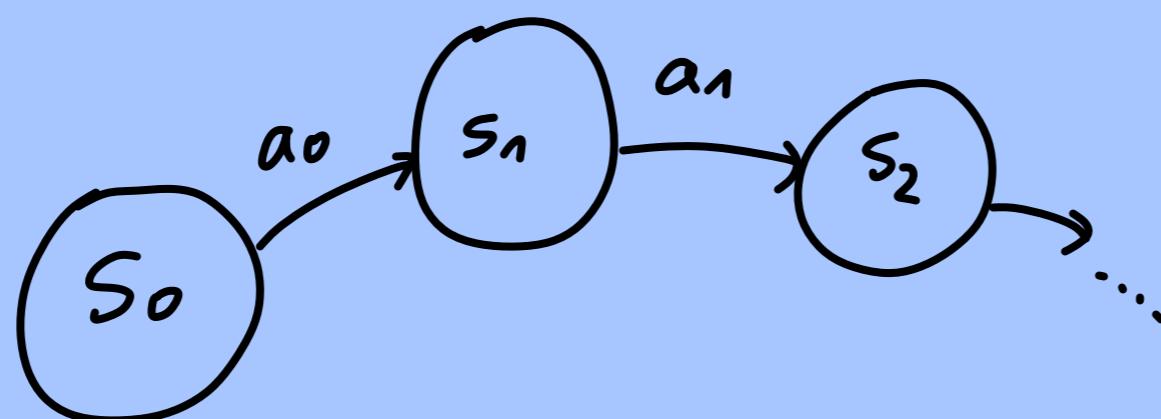
- AGENT MEMORY :

KEEP MEMORY / ALL THAT AGENT'S  
 SENSORS HAVE PERCEIVED AS PART  
 OF  $s_t$

→ COMPLETELY GENERAL

TRAJECTORY

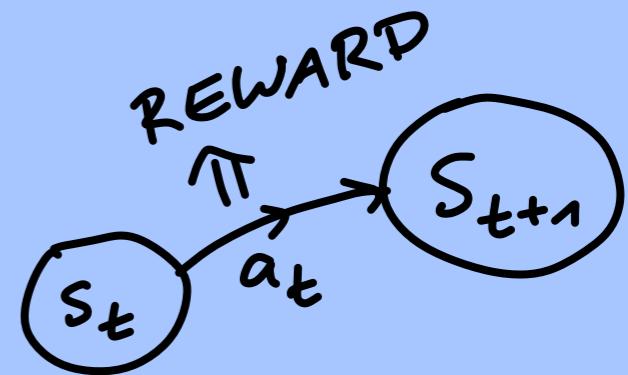
$$\tau = s_0, a_0, s_1, a_1, \dots, s_{T-1}, a_{T-1}, s_T$$



REWARD

$$r = r(s_t, a_t, s_{t+1})$$

$= r_{t+1}$  (DEF. LIKE  
SUTTON/BARTO)



RETURN

$$R = \text{CUMULATIVE REWARD} = \sum_{t=1}^T r_t$$

RL  $\equiv$  OPTIMIZE  
AVERAGE  
RETURN

$$R = \sum_{t=1}^T r_t$$

AT TIME  $t$ : CUMULATIVE FUTURE REWARD

$$R_{t+1} = r_{t+1} + r_{t+2} + \dots + r_T$$

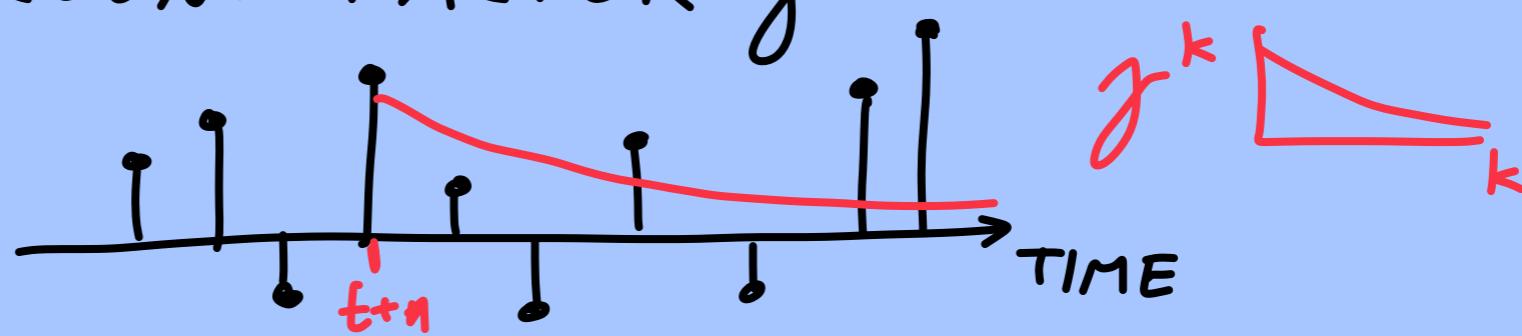
OPTIMIZE  $R = R_1$

FOR INFINITE SEQUENCES :  $R = \infty$  ?

⇒ DISCOUNTED RETURN

$$R_{t+1}^{(\gamma)} = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots$$

DISCOUNT FACTOR  $\gamma \leq 1$

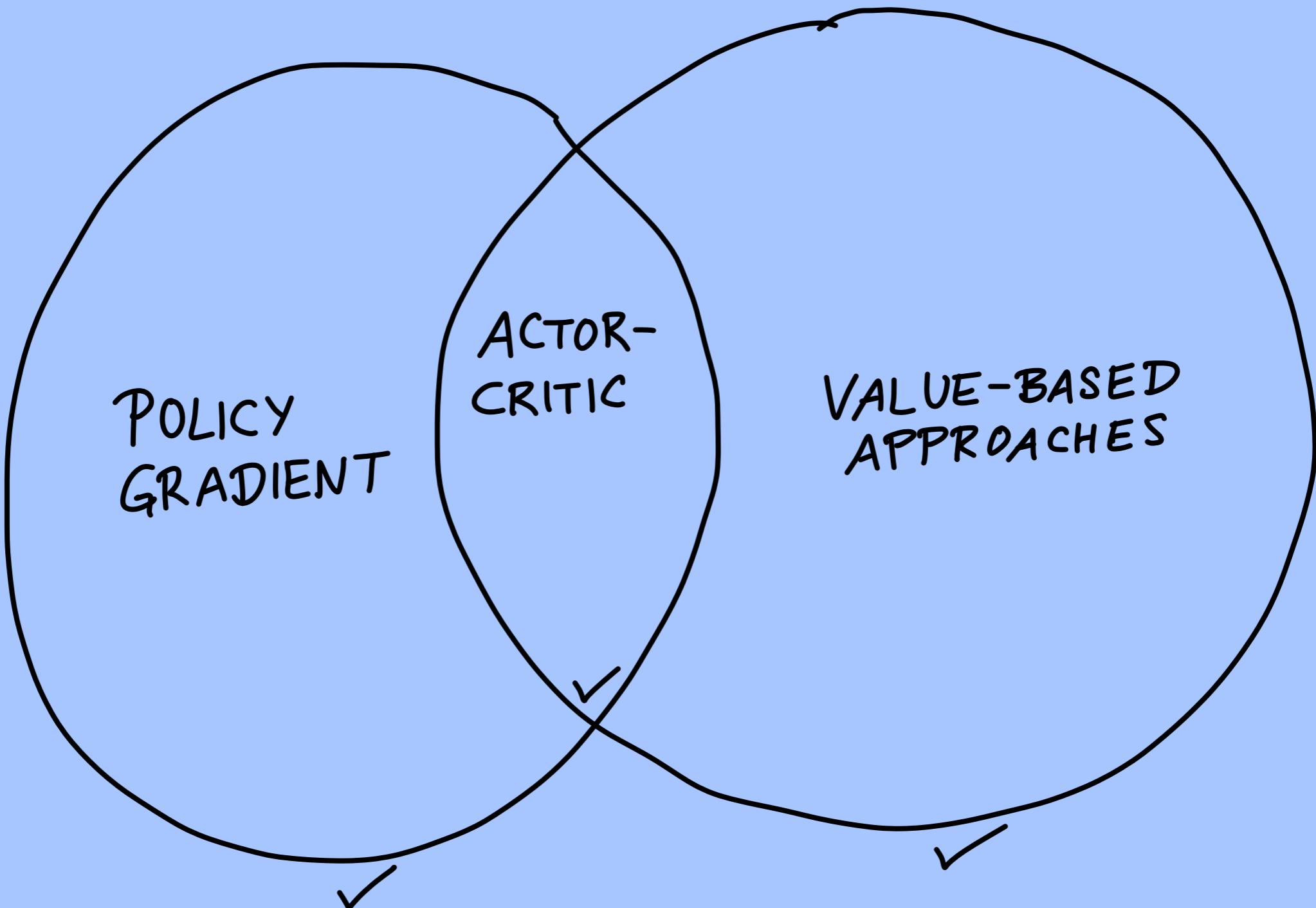


OPTIMIZING  $R_{t+1}^{(\gamma)}$

⇒ MORE EMPHASIS ON  
REWARDS THAT HAPPEN SOON

$\gamma=0$  : COMPLETELY GREEDY  
STRATEGY

# APPROACHES



9.2

## POLICY GRADIENT

$$\pi = \pi_{\theta}(a_t | s_t)$$

↗ NN

MAXIMIZE

$$E_{\pi}(R)$$

↗ DEPENDS  
ON POLICY!

$$E_{\pi}(R) = \sum_{\tau} R(\tau) P_{\pi}(\tau)$$

↗ TRAJECTORY

$$\tau = s_0, a_0, s_1, a_1, \dots, s_T$$

(& REWARDS)

$$R(\tau) = \sum_{t=1}^T r_t = \sum_t r(s_t, a_t, s_{t+1})$$

FROM  $\tau$



$$\begin{aligned}
 P_\pi(\tau) &= \dots \cdot \frac{P(s_1 | s_0, a_0)}{\pi(a_0 | s_0)} P(s_0) \\
 &= \left\{ \prod_{t=0}^{T-1} \frac{P(s_{t+1} | s_t, a_t)}{\pi(a_t | s_t)} \right\} P(s_0)
 \end{aligned}$$

↓ **FIXED**      ↓ **OPTIMIZE!**

GRADIENT

ASCENT

$$S\theta \sim \frac{\partial}{\partial \theta} E_{\pi_\theta}(R) = \sum_{\tau} R(\tau) \underbrace{\frac{\partial}{\partial \theta} P_{\pi_\theta}(\tau)}$$

$$P(\tau) \cdot \underbrace{\frac{\frac{\partial}{\partial \theta} P(\tau)}{P(\tau)}}$$

$$\frac{\partial}{\partial \theta} \ln P_{\pi_\theta}(\tau)$$

$$\frac{\partial}{\partial \theta} \ln P_{\pi_\theta}(\tau) = \sum_{t=0}^{T-1} \frac{\partial}{\partial \theta} \ln \pi_\theta(a_t | s_t)$$

$$S\theta = \gamma \frac{\partial}{\partial \theta} E_{\pi_\theta}(R) = \gamma E_\pi \left( R \cdot \sum_{t=0}^{T-1} \frac{\partial}{\partial \theta} \ln \pi_\theta(a_t | s_t) \right)$$

POLICY GRADIENT / "REINFORCE"

STEPS:

RUN MANY TRAJECTORIES

RECORD  $R$  &  $\tau = s_0, a_0, \dots$

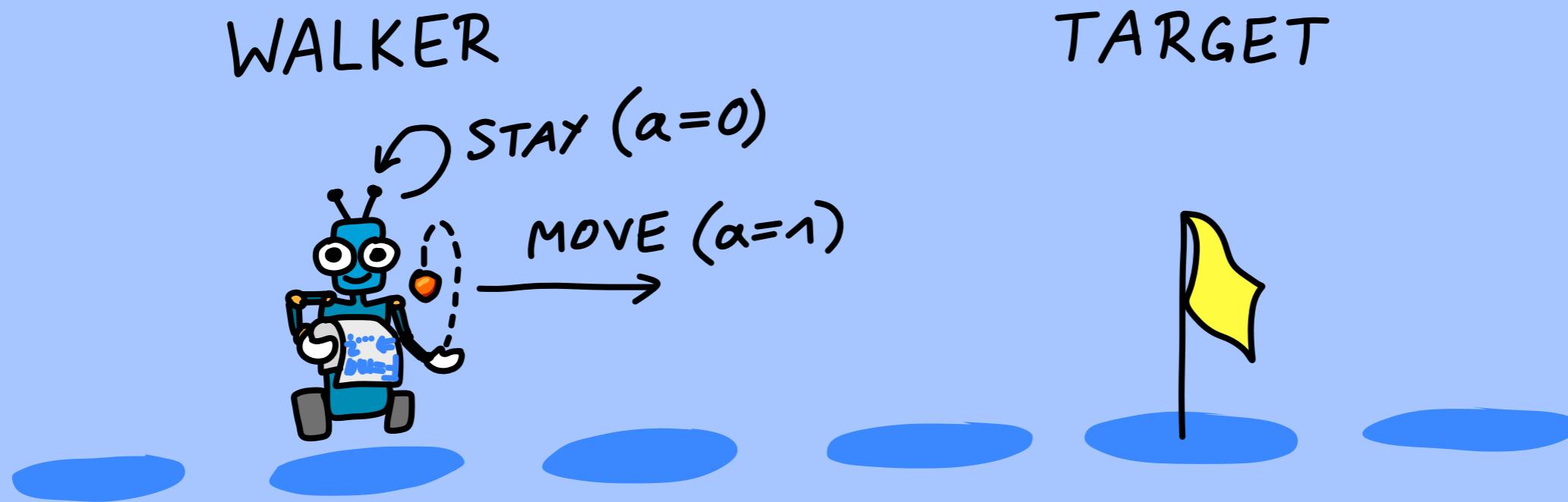
EVALUATE  $R \cdot \sum_t \partial_\theta \ln \pi(a_t | s_t)$

AVERAGE

UPDATE  $\theta$

REPEAT !

# SIMPLE EXAMPLE FOR POLICY GRADIENT

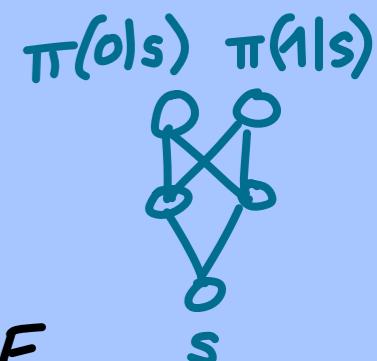


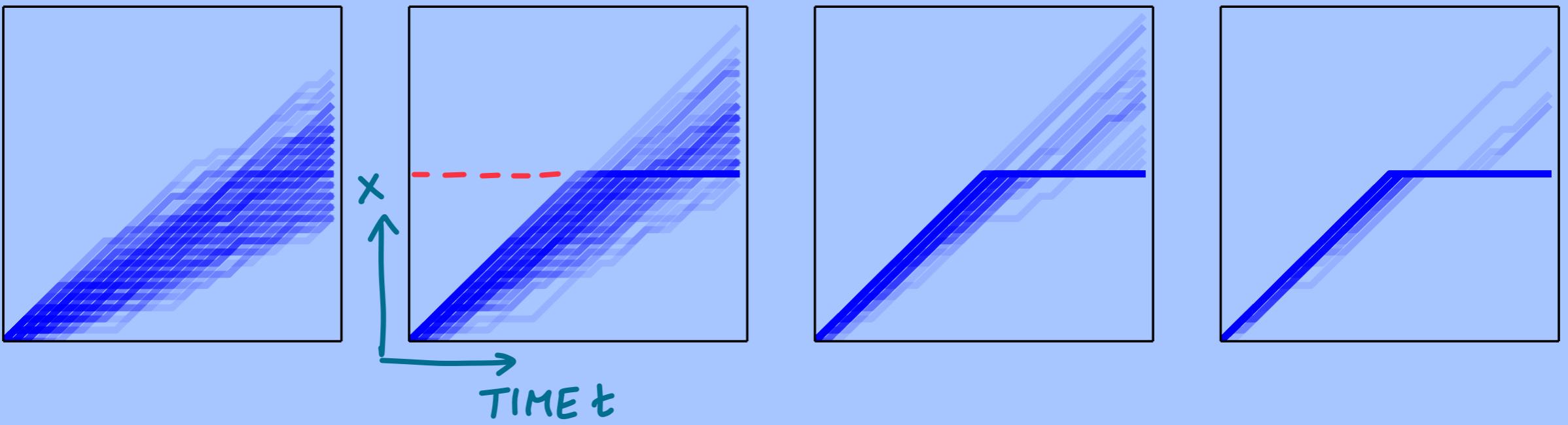
STATE = POSITION OF ROBOT & POS. OF TARGET (RANDOM)  
OF ENV.

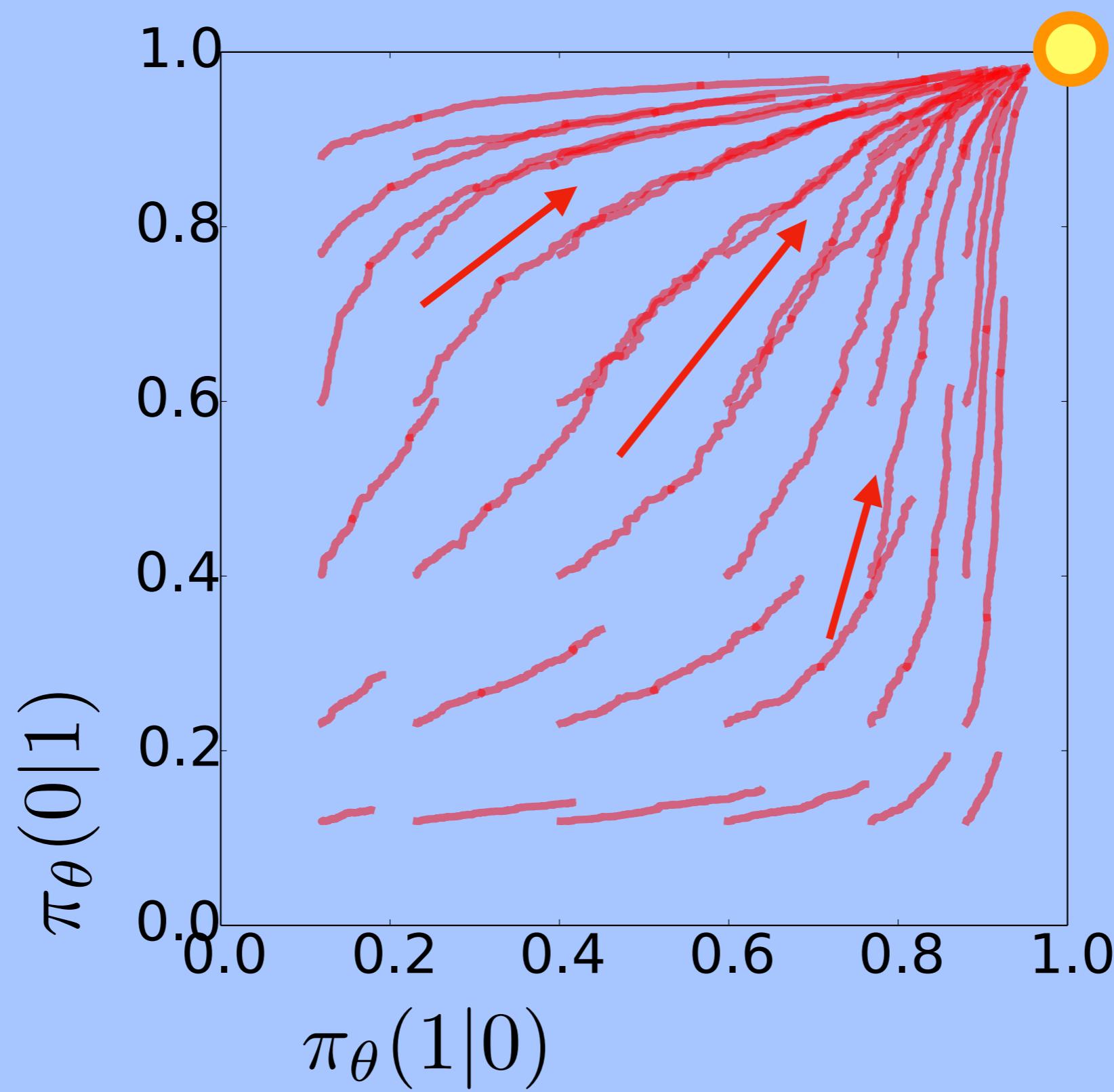
OBSERVED STATE =  $\begin{cases} 1 & \text{ON TARGET} \\ 0 & \text{OFF TARGET} \end{cases}$

POLICY  $\pi_{\theta}(a|s)$   
 $\pi_{\theta}(1|0)$   
 $= \text{PROB. TO } \underline{\text{MOVE}}$   
 $\text{IF OFF TARGET}$

RETURN  $R = \text{NUMBER OF TIME STEPS ON TARGET}$



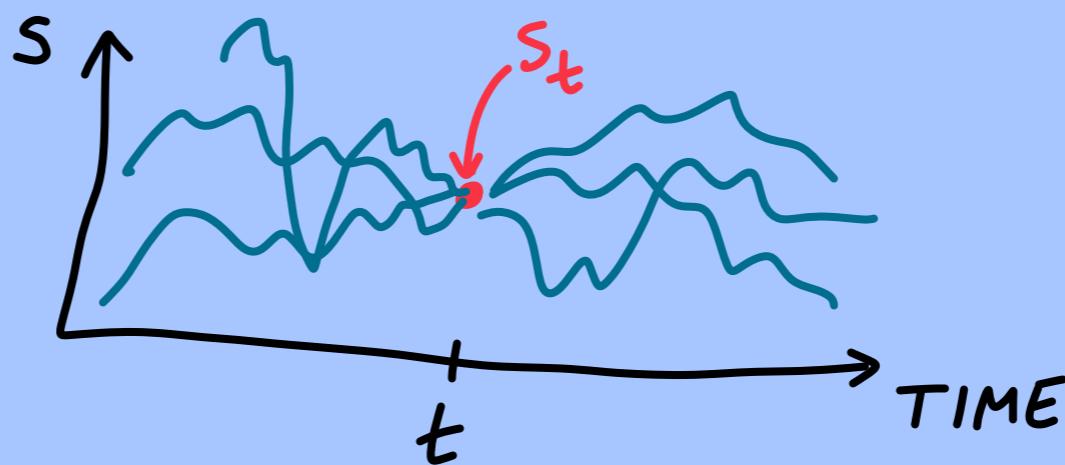




## MATHEMATICAL OBSERVATION:

$$\tau = s_0, a_0, s_1, a_1, \dots$$

CONSIDER RANDOM VARIABLE  $X$   
THAT ONLY DEPENDS ON  $s_0, \dots, s_t$



$$\begin{aligned}
 & E_{\pi} (g(a_t, s_t) X) \\
 &= \underbrace{\sum_s}_{\text{MARKOV}} E_{\pi} (g(a_t, s_t) | s_t = s) E_{\pi} (X | s_t = s) \cdot \underbrace{P(s_t = s)}_{(\text{MISSING IN LECTURES})} \\
 &= \sum_a \pi_\theta(a_t | s) g(a, s)
 \end{aligned}$$

NOW: FOR  $g = \partial_\theta \ln \pi_\theta(a_t | s_t)$ :

$$\sum_a \pi_\theta(a | s) \underbrace{g(s, a)}_{\frac{\partial_\theta \pi_\theta}{\pi_\theta}} = \sum_a \partial_\theta \pi_\theta = \partial_\theta \sum_a \pi_\theta = 0$$

$\Rightarrow$

$$E_\pi(\partial_\theta \ln \pi_\theta(a_t | s_t) X) = 0$$

$$\begin{aligned}
 & \xrightarrow{\Rightarrow} E_{\pi} ( R_1 \partial_{\theta} \ln \pi_{\theta}(a_t | s_t) ) \\
 &= E_{\pi} ( R_{t+1} \partial_{\theta} \ln \pi_{\theta}(a_t | s_t) ) \\
 R_{t+1} - R_1 &= r_1 + r_2 + \dots + r_t = X \\
 &\quad \text{WAS DROPPED!}
 \end{aligned}$$

$\Rightarrow$  ONLY FUTURE RETURN  
MATTERS FOR POLICY  
GRADIENT AT  $t$ !

GOOD:  $R_{t+1}$  MAY FLUCTUATE  
LESS THAN  $R_1$ !

NOW: CAN USE DISCOUNTED  
FUTURE RETURN

$$R_{t+1}^{(\gamma)}$$

AS APPROXIMATION!

$$\delta\theta = \gamma \sum_{t=0}^{T-1} E_\pi \left( R_{t+1}^{(\gamma)} \partial_\theta \ln \pi_\theta (\alpha_t | s_t) \right)$$

MAY SUBTRACT "BASELINE"  $B(s_t, t)$

$$E_{\pi} \left[ \underbrace{(R_{t+1}^{(\gamma)} - B(s_t, t))}_{\downarrow} \partial \ln \dots \right]$$

CAN REDUCE  
VARIANCE OF  $E_{\pi}[\dots]$

$$S\theta = \eta \frac{\partial}{\partial \theta} \dots$$

NATURAL POLICY GRADIENT:

USE FISHER INFORMATION MATRIX

$$I_{\ell j} = E_{\pi} \left[ \frac{\partial}{\partial \theta_\ell} \ln \pi_\theta(a|s) \quad \frac{\partial}{\partial \theta_j} \ln \pi_\theta(a|s) \right]$$

$$S\theta = \gamma I^{-1} \sum_t E_{\pi} [ R_{t+1}^{(g)} \partial_\theta \ln \pi_\theta ]$$

9.3

## Q LEARNING

DEFINE Q FUNCTION

$$Q_{\pi}(s_t, a_t) = E_{\pi}(R_{t+1} | s_t, a_t)$$

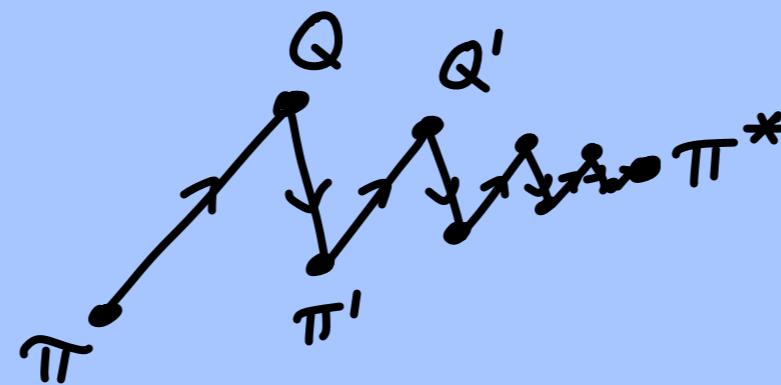
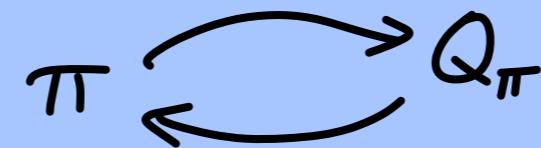
VALUE FUNCTION

$$V_{\pi}(s_t) = E_{\pi}(R_{t+1} | s_t)$$

OBVIOUS POLICY: PICK BEST ACTION  $a_t$   
ACCORDING TO  $Q_{\pi}$ 

$$a_t = \arg \max_a Q_{\pi}(s_t, a)$$

$\Rightarrow$  SELF-CONSISTENCY PROBLEM



COULD SAMPLE  $Q_\pi = E_\pi(\dots)$   
(MONTE CARLO)

# "BELLMAN EQUATION"

$$\underline{Q_{\pi}(s_t, a_t)} = E_{\pi} (R_{t+1}^{(\gamma)} | s_t, a_t)$$

$$= E_{\pi} (r_{t+1} + \underbrace{\gamma r_{t+2} + \gamma^2 r_{t+3} + \dots}_{\gamma R_{t+2}^{(\gamma)}} | s_t, a_t)$$

$$= E_{\pi} (r_{t+1} | s_t, a_t) + \gamma \underbrace{E_{\pi} (R_{t+2}^{(\gamma)} | s_t, a_t)}_{E_{\pi} (\underbrace{Q_{\pi}(s_{t+1}, a_{t+1})}_{\max_{a'} Q_{\pi}(s_{t+1}, a')} | s_t, a_t)}$$

$$= \sum_{s_{t+1}} (r_{t+1} + \gamma \underbrace{\max_{a'} Q_{\pi}(s_{t+1}, a')}_{\max_{a'} Q_{\pi}(s_{t+1}, a')} ) P(s_{t+1} | s_t, a_t)$$

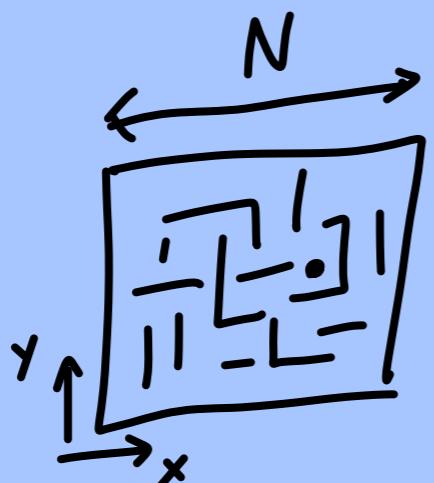
IDEA: UPDATE UNTIL FIXED POINT!

$$Q^{\text{NEW}}(s, a) = Q^{\text{OLD}}(s, a) + \alpha \left( \text{R.H.S.} \Big|_{Q^{\text{OLD}}} - Q^{\text{OLD}}(s, a) \right)$$

$$\alpha < 1$$

## TABLE-BASED APPROACH

$$Q(s, a) = \text{TABLE}$$



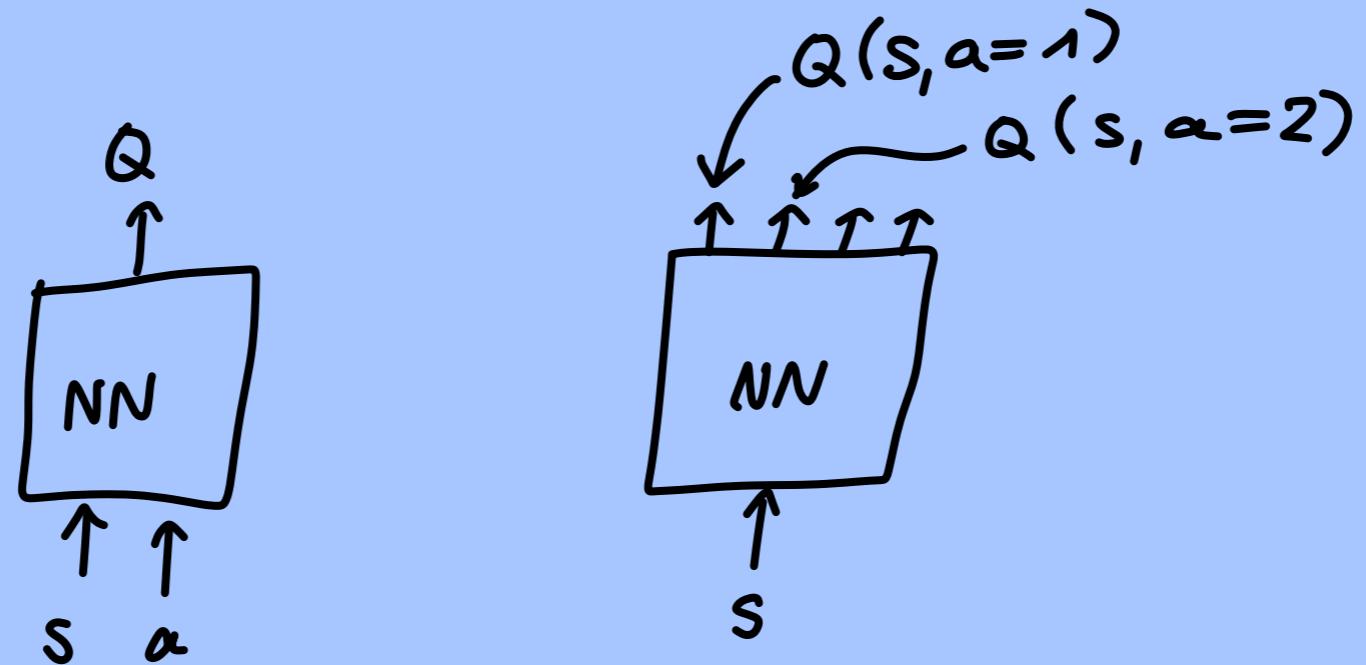
$s : N^2$  POSSIBILITIES  
 $a : 4$  POSS.

$Q : N^2 \cdot 4$  ENTRIES

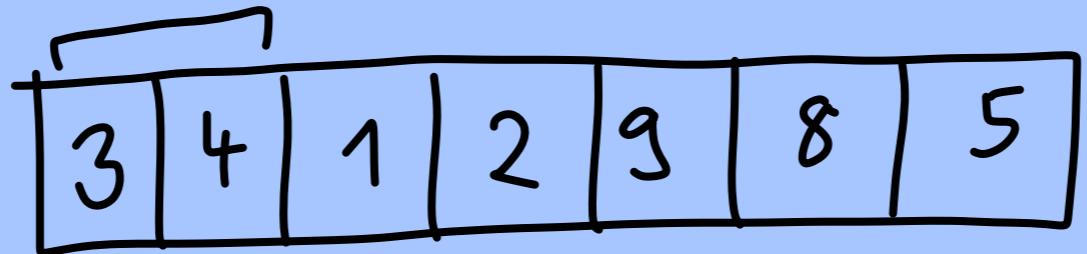
# NEURAL NETWORK

$Q_{\theta}(s, a)$   
→ NN PARAMETERS

$$\mathcal{L} = \langle \langle (Q_{\theta}(s, a) - \text{UPDATED})^2 \rangle \rangle_{s, a}$$



# BAD SCENARIO FOR RL



$10^7$

ONLY ONE

REST

$R = 1$

$R = 0$

EXPLORATION

vs.

EXPLORATION

USE POLICY

$$\arg \max_a Q(s, a)$$

TRY OUT NEW ACTIONS

EXAMPLE "ε-GREEDY"

WITH PROB.  $\epsilon$ : RANDOM a  
ELSE: POLICY

$$(\text{OR } p(a) \sim e^{\beta Q(s, a)})$$

Q LEARNING CAN RE-USE SAMPLES

STORE

$(s_t, a_t, s_{t+1}, r_{t+1})$   
"EXPERIENCE"

USE AGAIN LATER FOR  
UPDATING Q !

→ "EXPERIENCE REPLAY"

OLD SAMPLES

"OFF-POLICY" → BUT STILL CAN  
BE USED  
(CONTRAST TO  
POLICY GRADIENT)

9.4

## ADVANTAGE ACTOR-CRITIC

 $\rightarrow$  COMBINE POL. GRAD. & Q-LEARNING

$$E_{\pi} \left[ R_{t+1} + \gamma \frac{\partial}{\partial \theta} \ln \pi_{\theta}(a_t | s_t) \right]$$

$\underbrace{R_{t+1}}_{= r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots}$

$$E_{\pi} [R_{t+1} | s_t, a_t] = Q_{\pi}(s_t, a_t)$$

NOTE:

$$\begin{aligned} E[f(x, y) g(x)] \\ = E[E[f(x, y) | x] g(x)] \end{aligned}$$

$$= E_{\pi} \left[ Q_{\pi}(s_t, a_t) \frac{\partial}{\partial \theta} \ln \pi_{\theta}(a_t | s_t) \right]$$

$$E \left[ (Q(s_t, a_t) - B(s_t)) \partial_\theta \ln \pi \right] \\ = E \left[ Q \frac{\partial}{\partial \theta} \ln \pi_\theta \right]$$

BUT CHOOSE  $B$  TO REDUCE VARIANCE

IDEA: CHOOSE

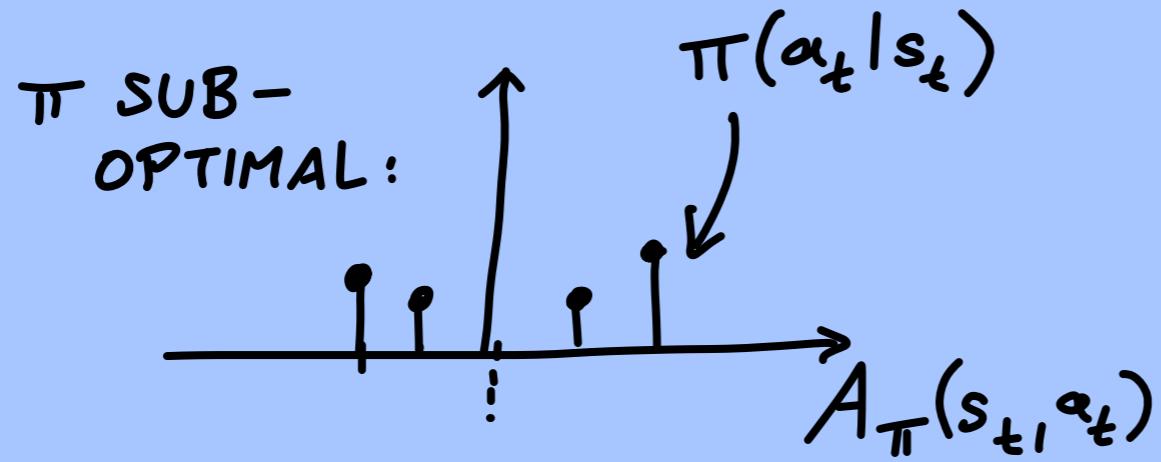
$$E_{\pi} [Q_{\pi}(s_t, a_t) | s_t = s] = V_{\pi}(s_t = s)$$

$$\dots = E_{\pi} \left[ \underbrace{\left( Q_{\pi}(s_t, a_t) - V_{\pi}(s_t) \right)}_{\text{"ADVANTAGE"}} \frac{\partial}{\partial \theta} \ln \pi_{\theta}(a_t | s_t) \right]$$

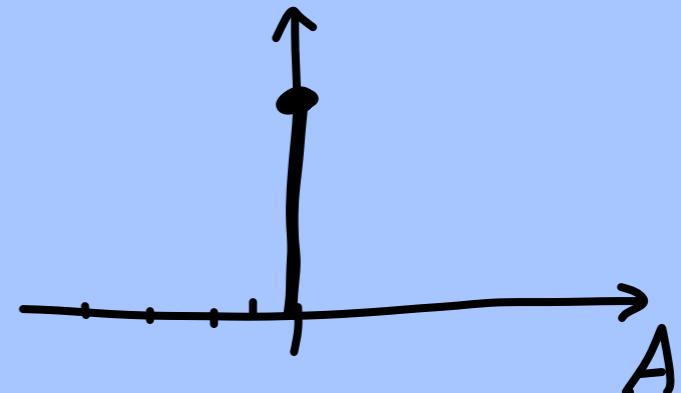
$A_{\pi}(s_t, a_t)$

= HOW MUCH BETTER  
DOES  $a_t$  PERFORM  
THAN  $\pi$  ON AVERAGE?

$$E_{\pi} [ A_{\pi}(s_t, a_t) | s_t = s ] = 0$$



$\pi$  OPTIMAL:



$$\Delta \theta = \gamma \sum_t E_\pi \left[ A_\pi(s_t, a_t) \frac{\partial}{\partial \theta} \ln \pi_\theta(a_t | s_t) \right]$$

ADV. ACTOR-CRITIC

APPROXIMATION FOR  $A$ :

$$A_\pi(s_t, a_t) = r_{t+1}(s_t, a_t, s_{t+1}) + \gamma V_\pi(s_{t+1}) - V_\pi(s_t)$$

FOR  $V_\pi$ :

$$V_\pi^{\text{NEW}}(s_t) = V_\pi^{\text{OLD}}(s_t) + \alpha \underbrace{(r_{t+1} + \gamma V_\pi^{\text{OLD}}(s_{t+1}) - V_\pi^{\text{OLD}}(s_t))}_{= \text{OUR } A\text{-APPROX.!}}$$

# NEURAL NETWORK APPROX.:

$$\mathcal{L}_V = \sum_t E_{\pi} \left( V_{\mu}^{NN}(s_t) - \underbrace{V_{\pi}^{NEW}(s_t)}_{\substack{\text{FROM} \\ \text{BELLMAN} \\ \text{UPDATE}}} \right)^2$$

$(V_{\pi}^{\text{OLD}} = V_{\mu}^{NN}) \rightarrow \text{CURRENT VALUE}$

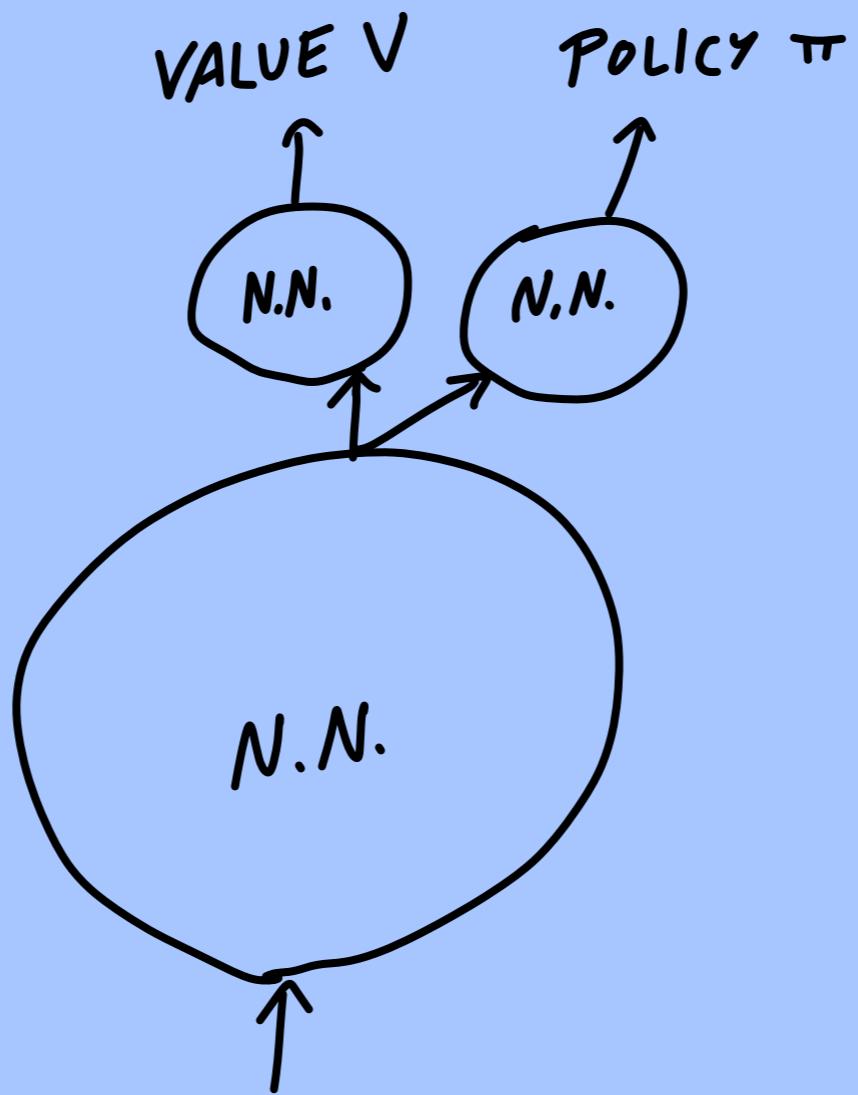
POLICY NETWORK

$$\pi_{\theta}(a_t | s_t)$$

VALUE NETWORK

$$V_{\mu}(s_t)$$

COMBINE!



"TWO HEADS"

$$\theta = \underline{\text{ALL PARAMETERS IN ALL PARTS}}$$
$$\mathcal{L} = \sum_t E_{\pi} \left[ \underbrace{A_{\pi}(s_t, a_t)}_{\theta \text{ FIXED HERE}} \ln \pi_{\theta}(a_t | s_t) \right] + \mathcal{L}_V$$

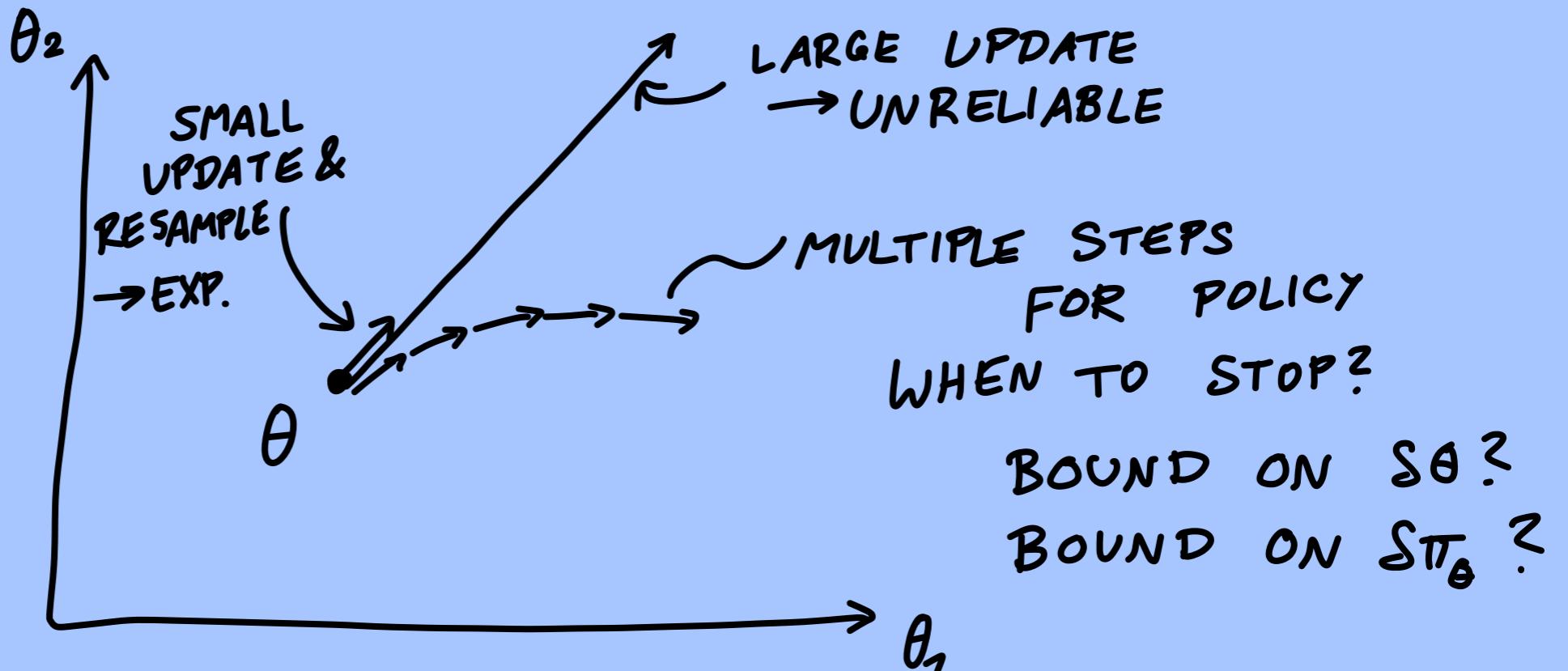
(WITH  $\pi \mapsto \theta$ )

9.5

## TRUST REGIONS

TRAJECTORIES MAYBE EXPENSIVE

⇒ LARGE POLICY UPDATES?

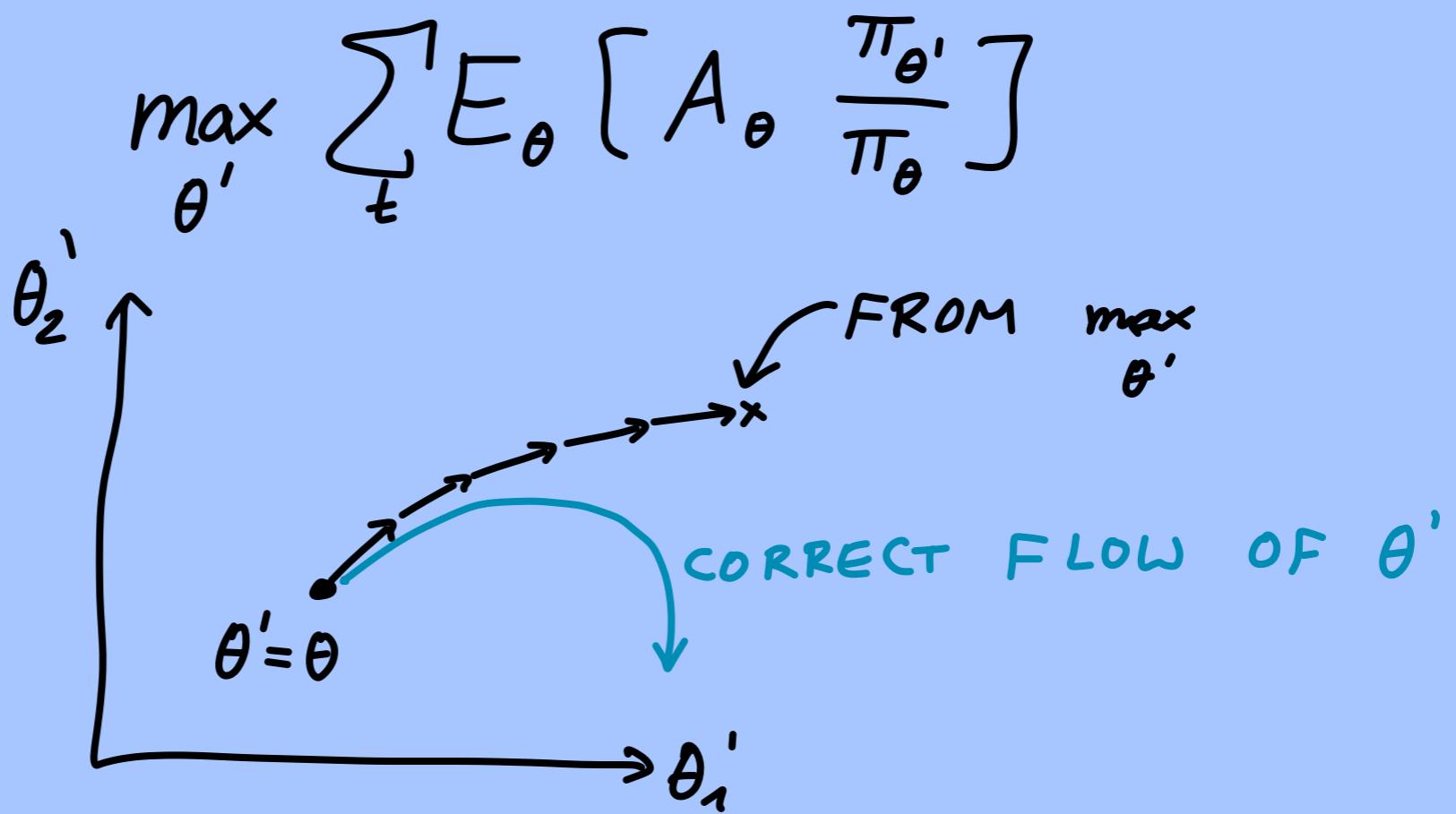


$$\sum_t E_\theta [A_\theta \frac{\partial}{\partial \theta} \ln \pi_\theta]$$

DEPEND ON  $s_t, a_t$

$$\frac{\partial}{\partial \theta} \ln \pi_\theta = \frac{\frac{\partial}{\partial \theta} \pi_\theta}{\pi_\theta}$$

WOULD APPEAR IN GRAD-MAX.OF :



IDEA : LIMIT CHANGE OF  $\pi$  !

$$\mathcal{D}_{\text{KL}}(\pi_{\theta} \parallel \pi_{\theta'}) \leq S$$

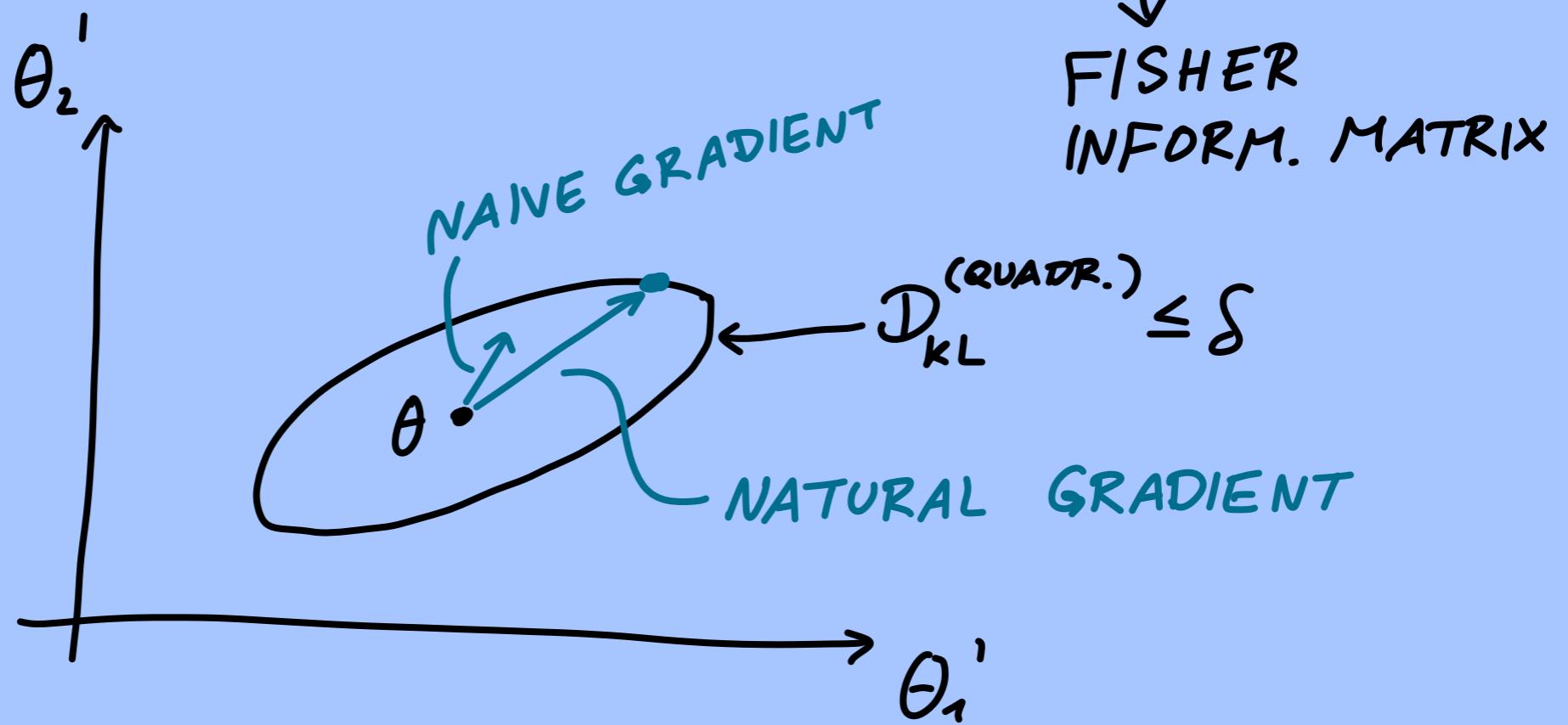
||

$$E_{\pi_{\theta}}(\ln \frac{\pi_{\theta}}{\pi_{\theta'}})$$

$\Rightarrow$  "TRUST REGION POLICY OPTIMIZATION"  
TRPO

QUADRATIC APPROXIMATION FOR  $D_{KL} \Rightarrow$

$$D_{KL}(\pi_\theta \| \pi_{\theta'}) \approx \frac{1}{2} (\theta' - \theta)^t \mathcal{I} (\theta' - \theta)$$



(NOTE I IS EXPENSIVE)

9.6

## PROXIMAL POLICY OPTIMIZATION (PPO)

LIKE TRPO, BUT WITHOUT  $\mathcal{I}$

OPTION 1:

$$\text{MAXIMIZE } \sum_t E_{\theta} \left[ A_{\theta} \frac{\pi_{\theta'}}{\pi_{\theta}} - \beta \mathcal{D}_{\text{KL}} (\pi_{\theta} \parallel \pi_{\theta'}) \right]$$

BUT  $\beta = ?$  WANT  $\mathcal{D}_{\text{KL}} \leq \mathcal{S}$

ADAPT  $\beta$ !

OPTION 2:

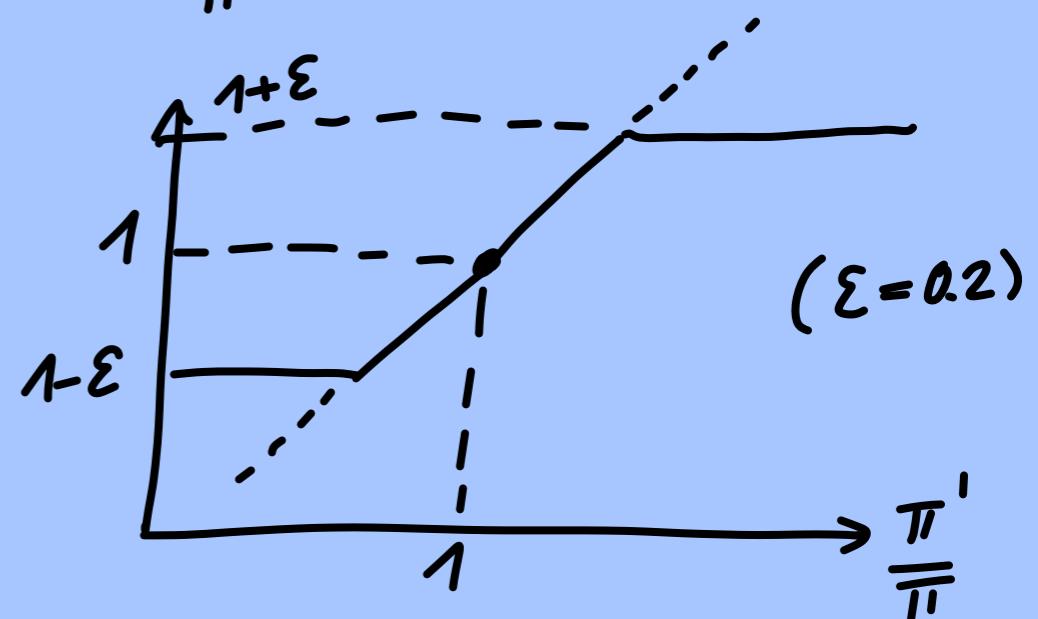
## "CLIPPED OBJECTIVE"

$$A \frac{\pi'}{\pi}$$

OBJECTIVE



$$\frac{\pi_{\theta'}}{\pi_{\theta}} \mapsto \text{CLIP}\left(\frac{\pi_{\theta'}}{\pi_{\theta}}\right)$$



$\Rightarrow$  NO INCENTIVE FOR  
LARGE  $\frac{\pi'}{\pi}$

$$A \frac{\pi'}{\pi} \mapsto \min \underbrace{\left( A \frac{\pi'}{\pi}, A \text{ CLIP}\left(\frac{\pi'}{\pi}\right) \right)}_{\text{PESSIMISTIC ESTIMATE}}$$

SO: REPLACE

$$E[A \ln \pi]$$

BY

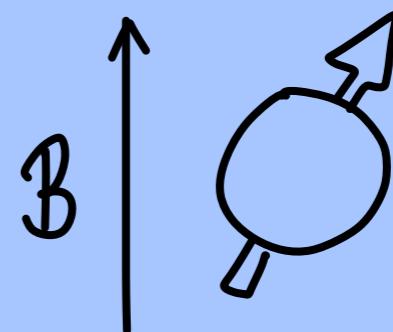
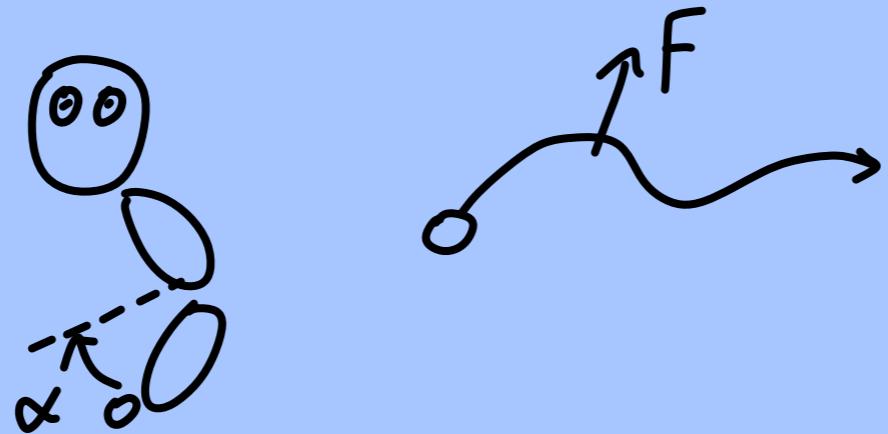
$$E \left[ \min \left( A \frac{\pi'}{\pi}, A \text{CLIP} \left( \frac{\pi'}{\pi} \right) \right) \right]$$

$\Rightarrow$  PPO

- NO KL NEEDED
- VERY SIMPLE
- WORKS WELL !

9.7

## CONTINUOUS ACTIONS IN REINFORCEMENT LEARNING

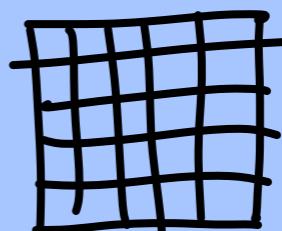


$$\pi_{\theta}(a_t | s_t) \in \mathbb{R}^d$$

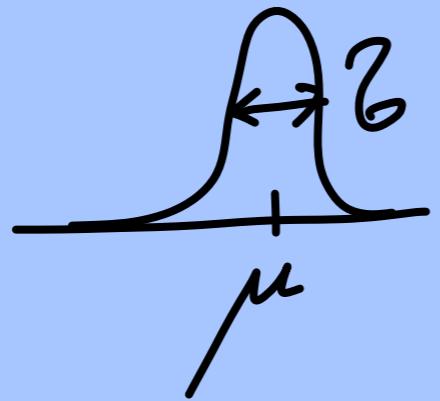
COULD DISCRETIZE



NOT EFFICIENT  
FOR HIGH-DIM.  
 $a \in \mathbb{R}^d$

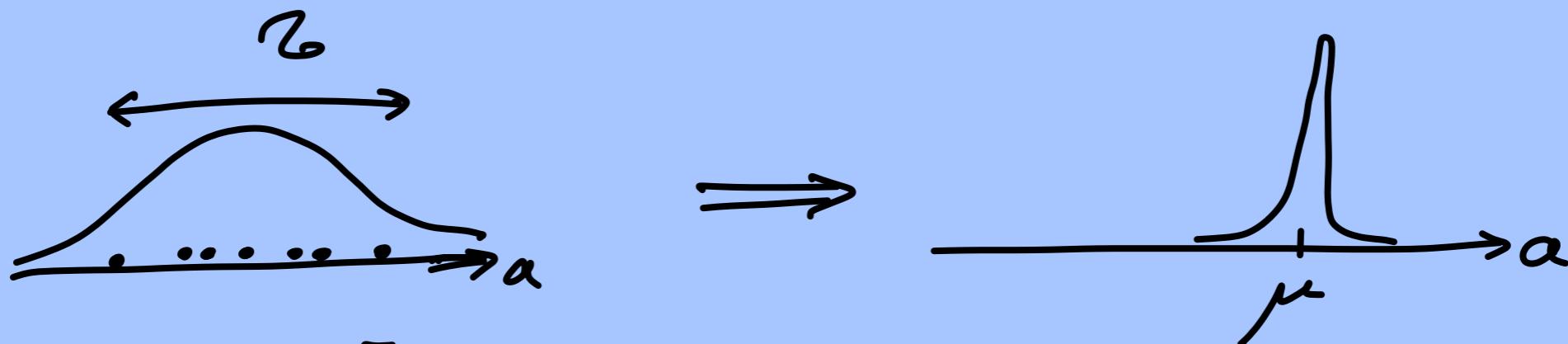


$$\pi_{\theta}(a|s) \sim \frac{1}{Z} \exp\left(-\frac{1}{2\sigma^2}(a-\mu)^2\right)$$



$$\begin{aligned}\mu &= \mu_{\theta}(s) = \text{NN} \\ Z &= Z_{\theta}(s) = \text{NN} \\ &\quad \text{SPREAD}\end{aligned}$$

$$\frac{\partial}{\partial \theta} \ln \pi_{\theta} = -\frac{\partial}{\partial \theta} \ln Z + \frac{\partial}{\partial \theta} \left[ -\frac{1}{2\sigma^2} (a - \mu_{\theta})^2 \right]$$



START OF  
TRAINING

9.8

## MODEL-BASED REINFORCEMENT LEARNING

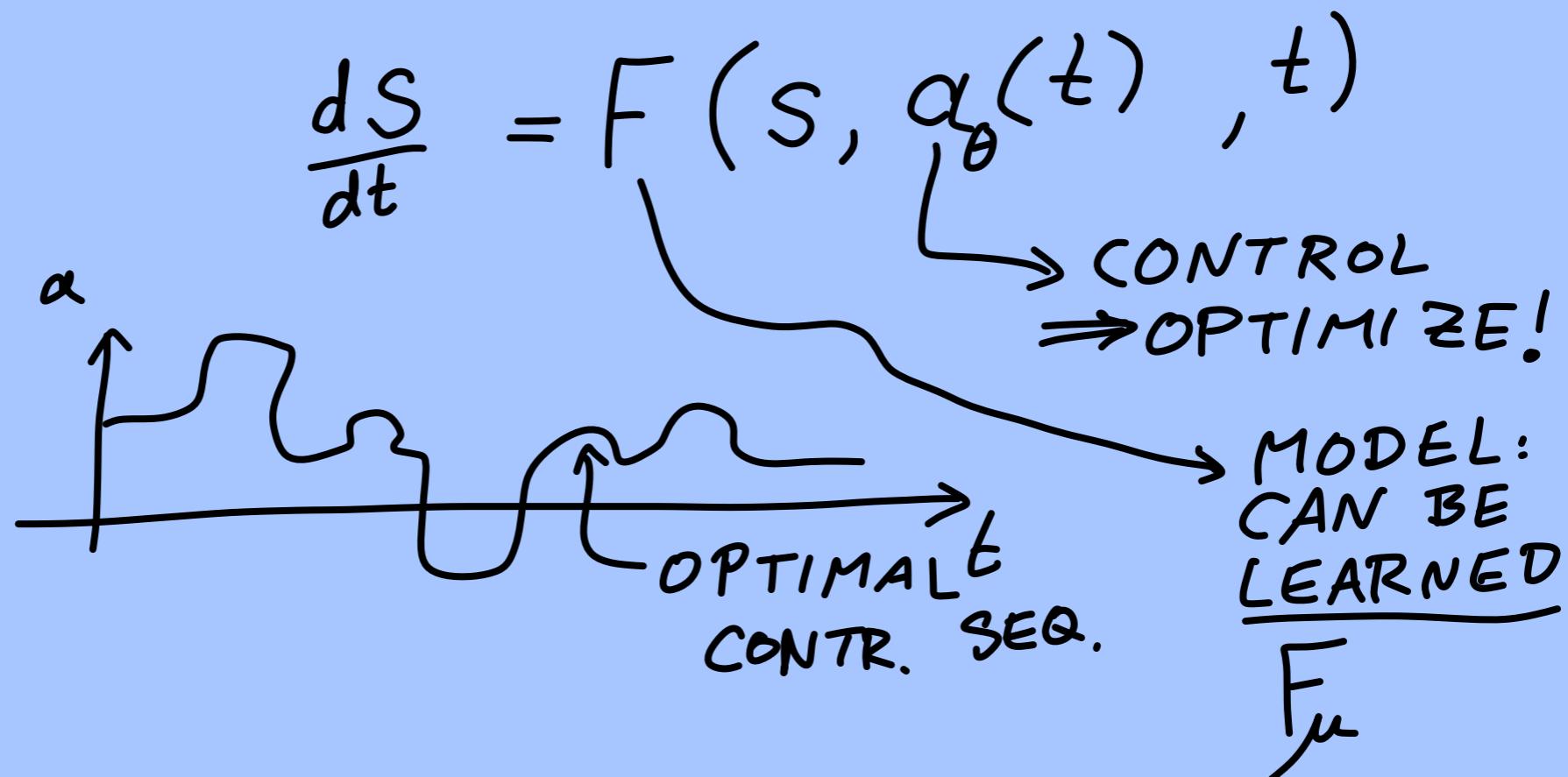
SO FAR: MODEL-FREE

- ⇒ NO NEED FOR MODEL
- ⇒ APPLY TO 'BLACK BOX'  
ENV. !

IF WE KNOW A MODEL:

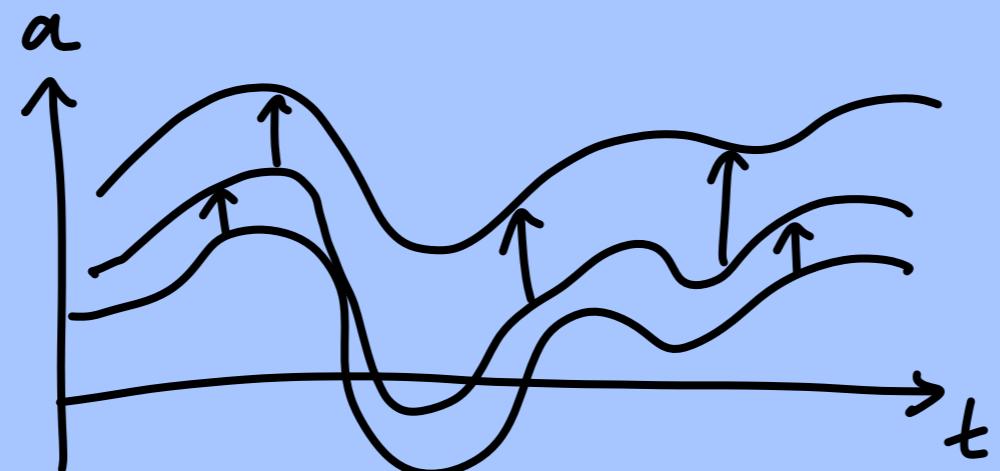
MODEL-BASED RL  
"PLANNING"

EXAMPLE: DIFFERENTIABLE  
DETERMINISTIC MODEL  
(EQS. OF MOTION)



$$R = R[\tau_\theta] \xrightarrow{\text{WHOLE TRAJECTORY}}$$

$$\Rightarrow S\theta = \gamma \frac{\partial R}{\partial \theta}$$



"OPTIMAL CONTROL"

QUANTUM:  $\hat{U}_\theta$  GRAPE

EXAMPLE : STOCHASTIC DISCRETE MODEL  
LEARN PROBAB. MODEL FROM OBSERV.!

$$P_{\mu}(s_{t+1} | s_t, a_t)$$

IF NOT FULLY OBSERV.:

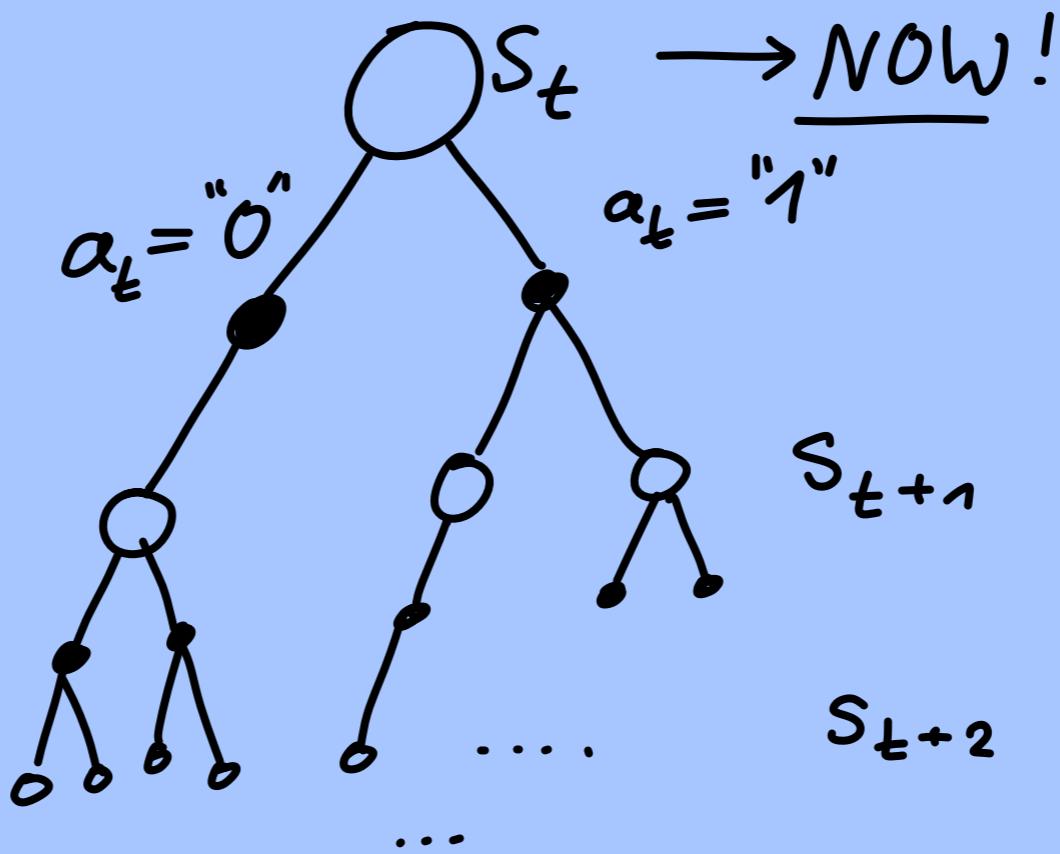
$s_t$  (HERE) INCLUDES  
ALL OBSERV. UP TO  $t$

SIMPLE STRATEGY :

- LEARN MODEL FROM EXPERIENCE
- SAMPLE FROM MODEL  
IN SIMULATIONS FOR MODEL-FREE  
RL !

$\Rightarrow$  USE THAT POLICY  
IN REAL  
WORLD

# SIMULATION-BASED SEARCH



SIMULATE MANY EPISODES  
FROM  $s_t$  (USING MODEL)

AVERAGE RETURN  
 $Q_\pi(s_t, a)$   
SIMULATION  
POLICY (FIXED)  
→ SELECT  $\arg\max_a Q_\pi(s_t, a)$

EVEN BETTER:

## MONTE-CARLO TREE SEARCH

EVALUATE  $Q(s, a)$  FOR ALL  $s, a$  VISITED  
IN OUR EXPLORATION

START FROM  $s_t$

"OFF-TREE" (FOR STATES THAT  
HAVE NOT BEEN EXPLORED)  
RANDOM ACTION (OR SOME  
FIXED POLICY)

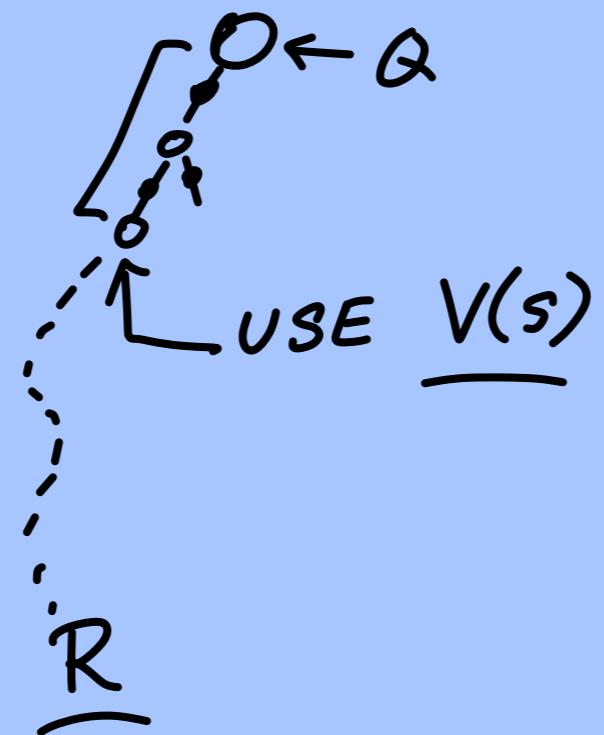
"ON-TREE" POLICY =  $\epsilon$ -GREEDY  
 $\arg \max_a Q(s, a)$   
→ IMPROVES

ON-TREE POLICY MAY PREFER LESS  
VISITED ACTIONS

# IMPROVEMENTS VIA DEEP LEARNING (→ ALPHA-GO)

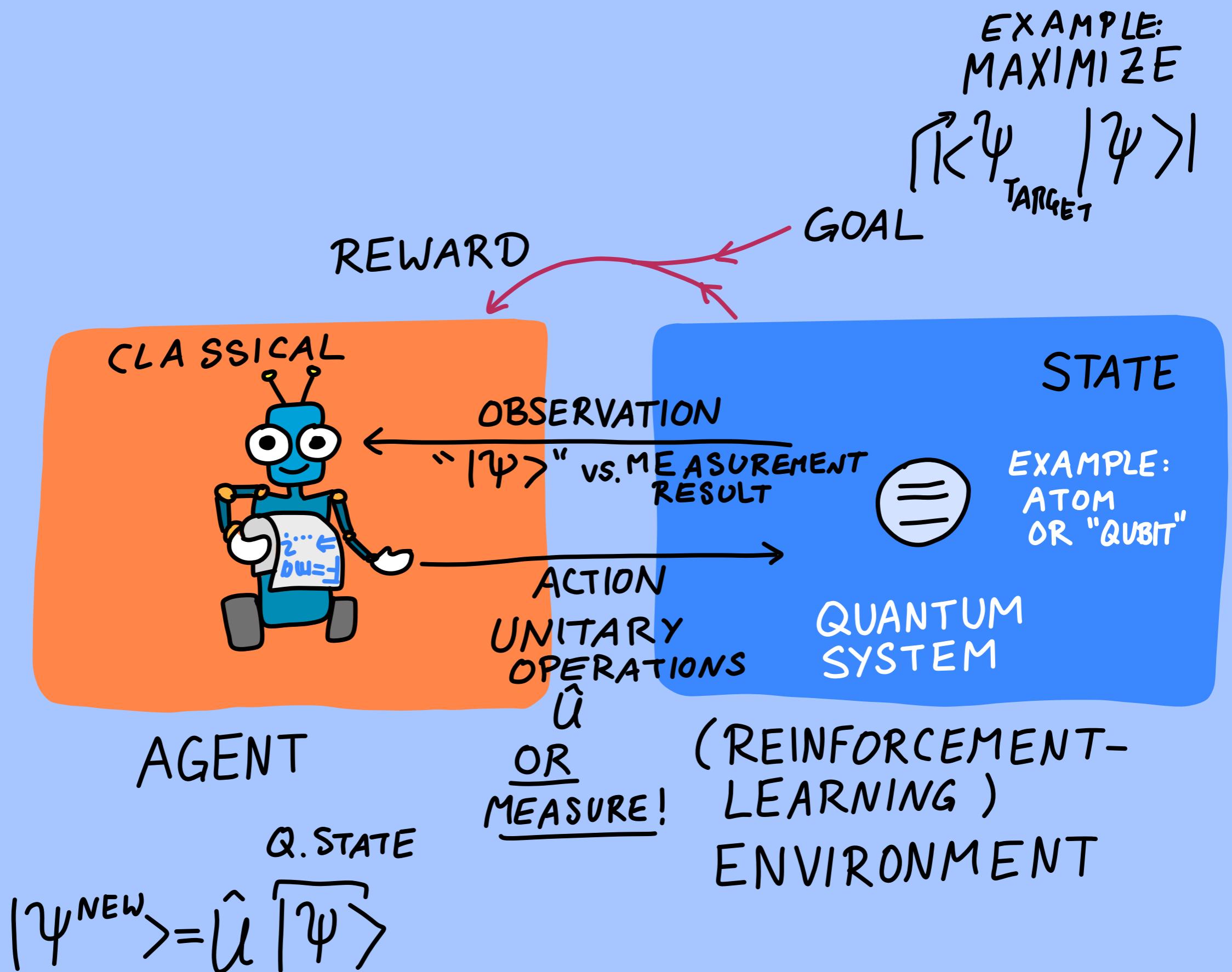
$$MCTS + Q + V + \pi_\theta \text{ (POLICY GRAD.)}$$

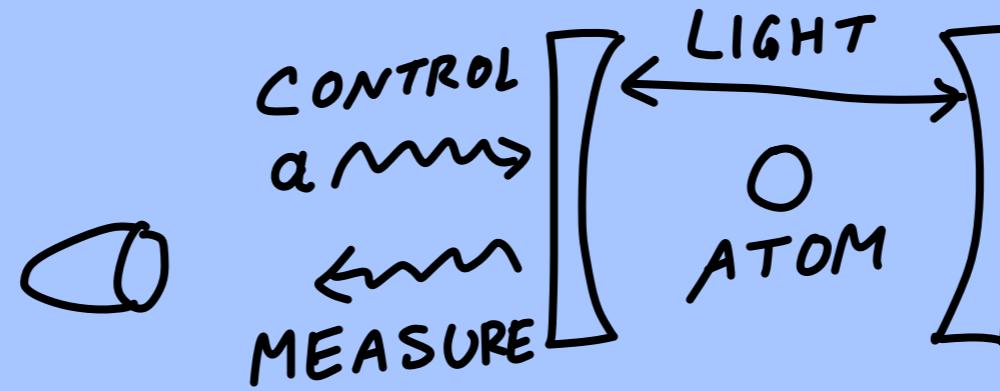
- OFF-TREE  $\pi_\theta$  TRAINED USING POLICY GRAD.
- ESTIMATE  $Q$  BY USING A VALUE FUNCTION  $V_\theta$  (ALSO TRAINED) AT "LEAF NODES" (NOT EXPLORED)



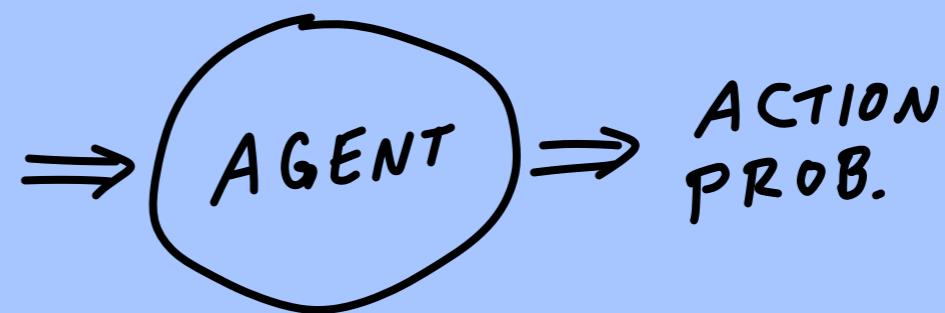
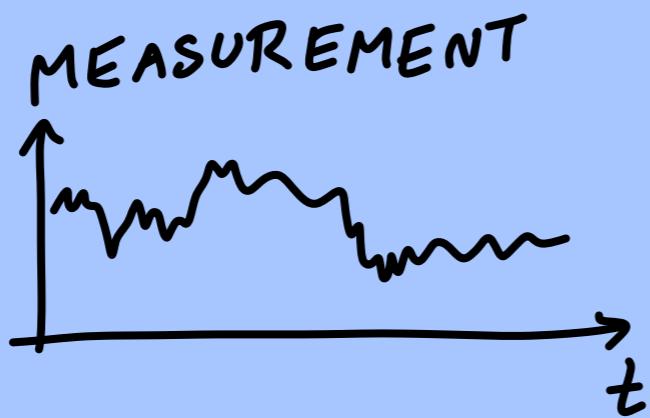
9.9

## CASE STUDY: REINFORCEMENT LEARNING FOR QUANTUM CONTROL & FEEDBACK

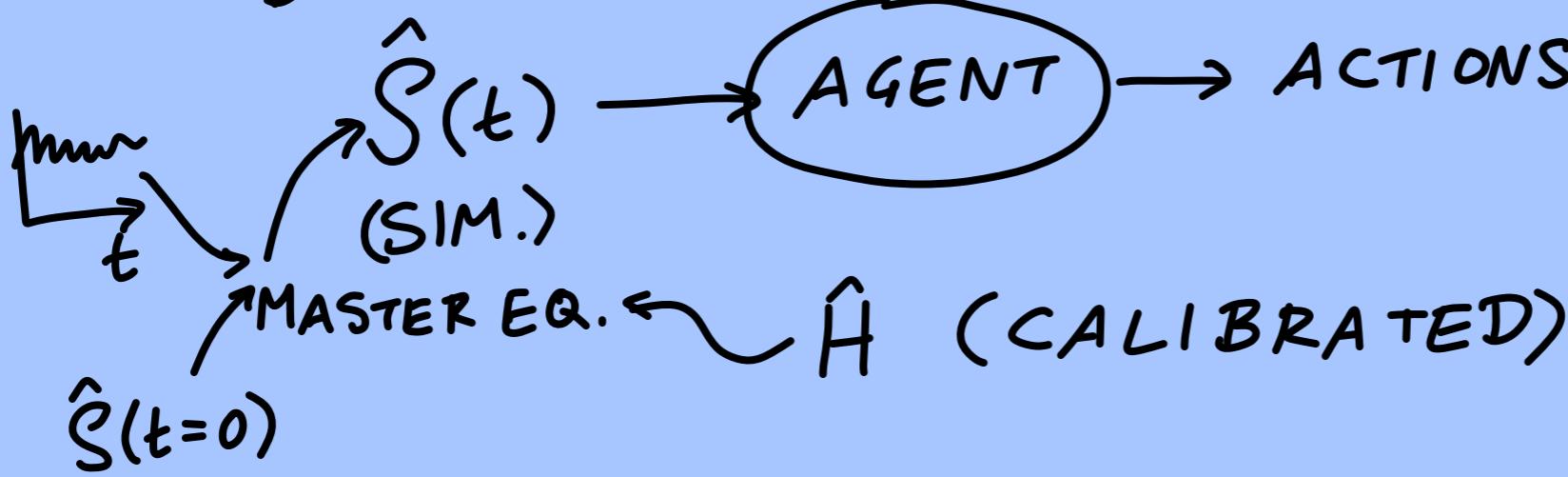


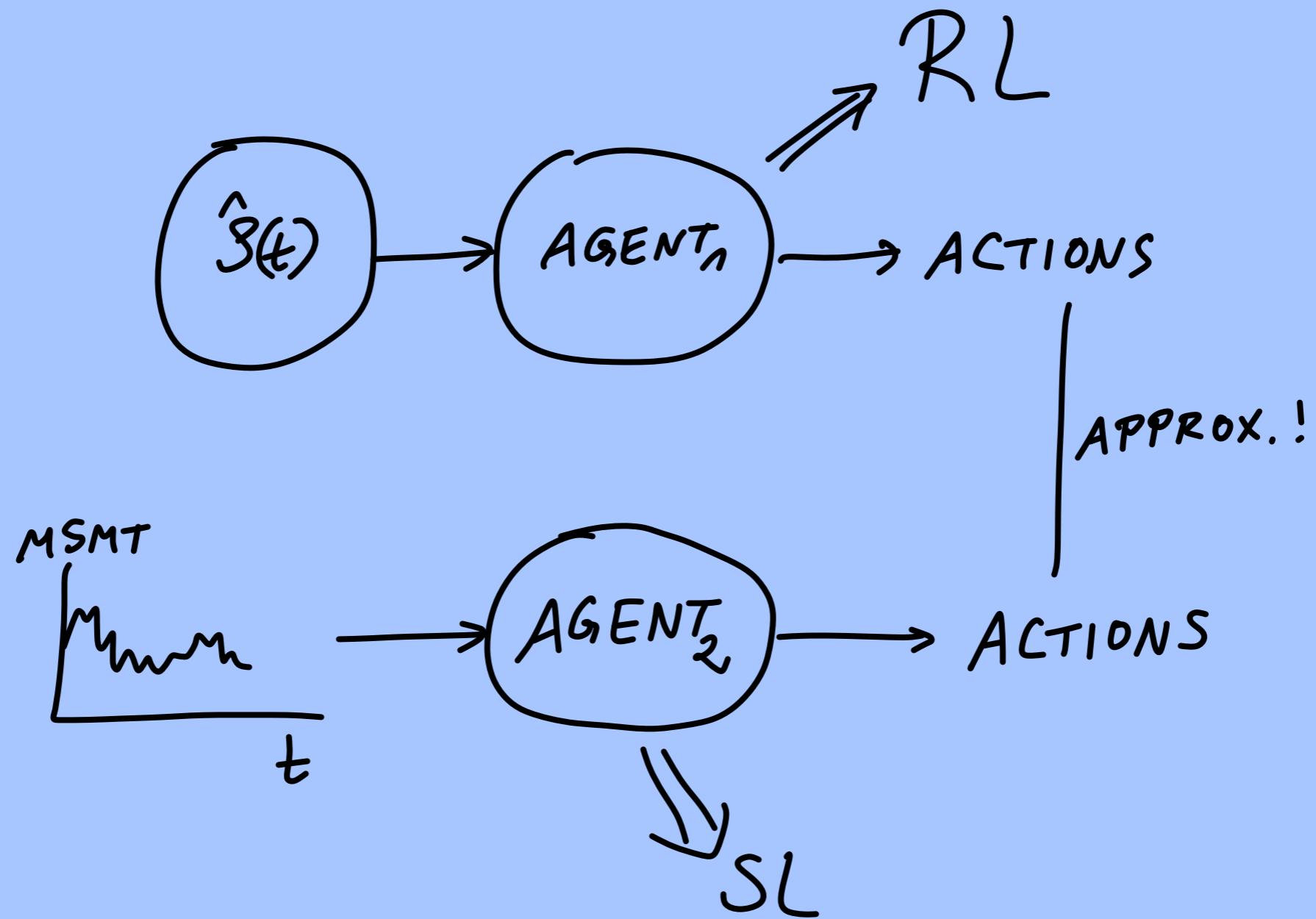


MODEL-FREE

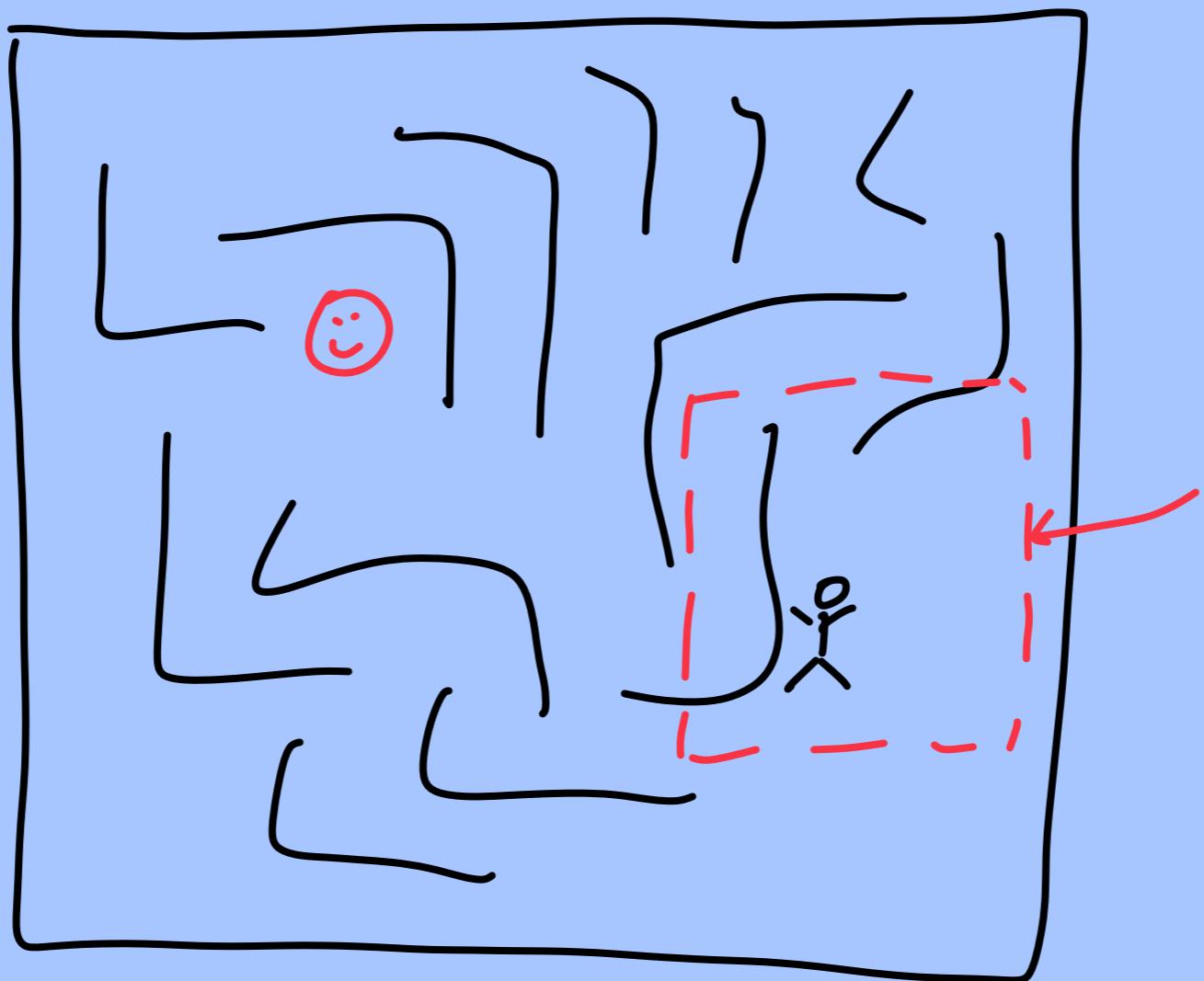


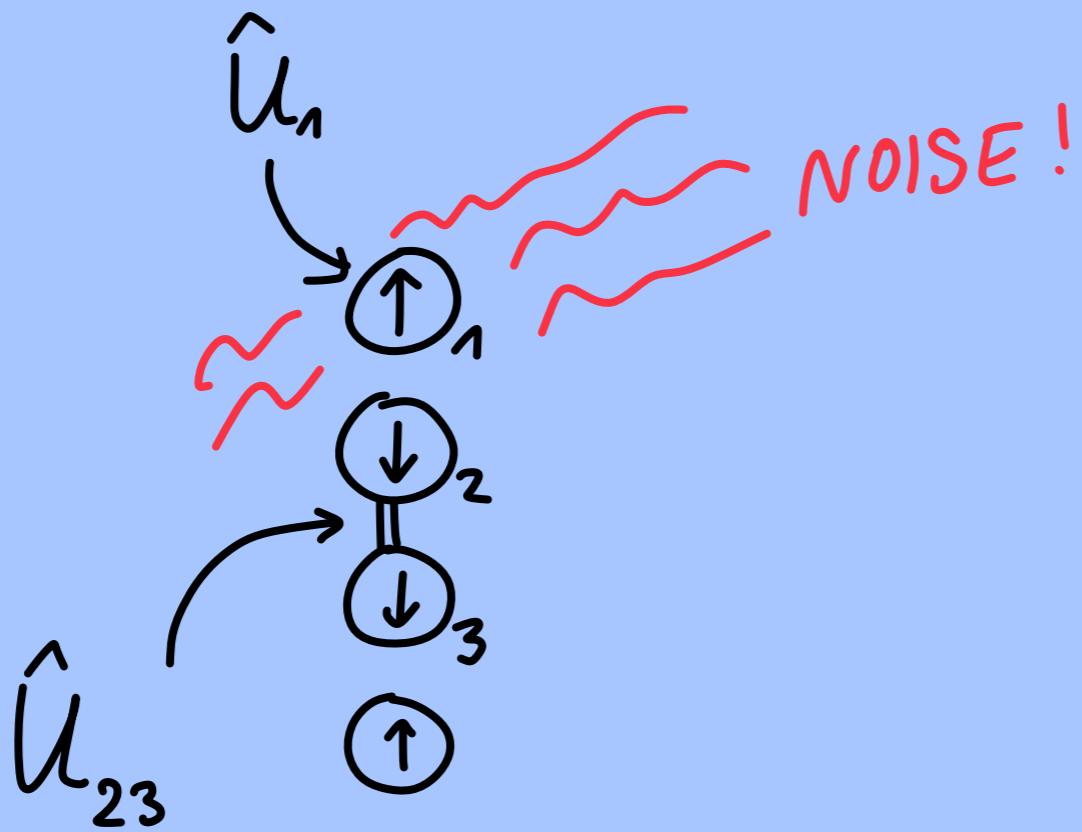
DENSITY MATRIX





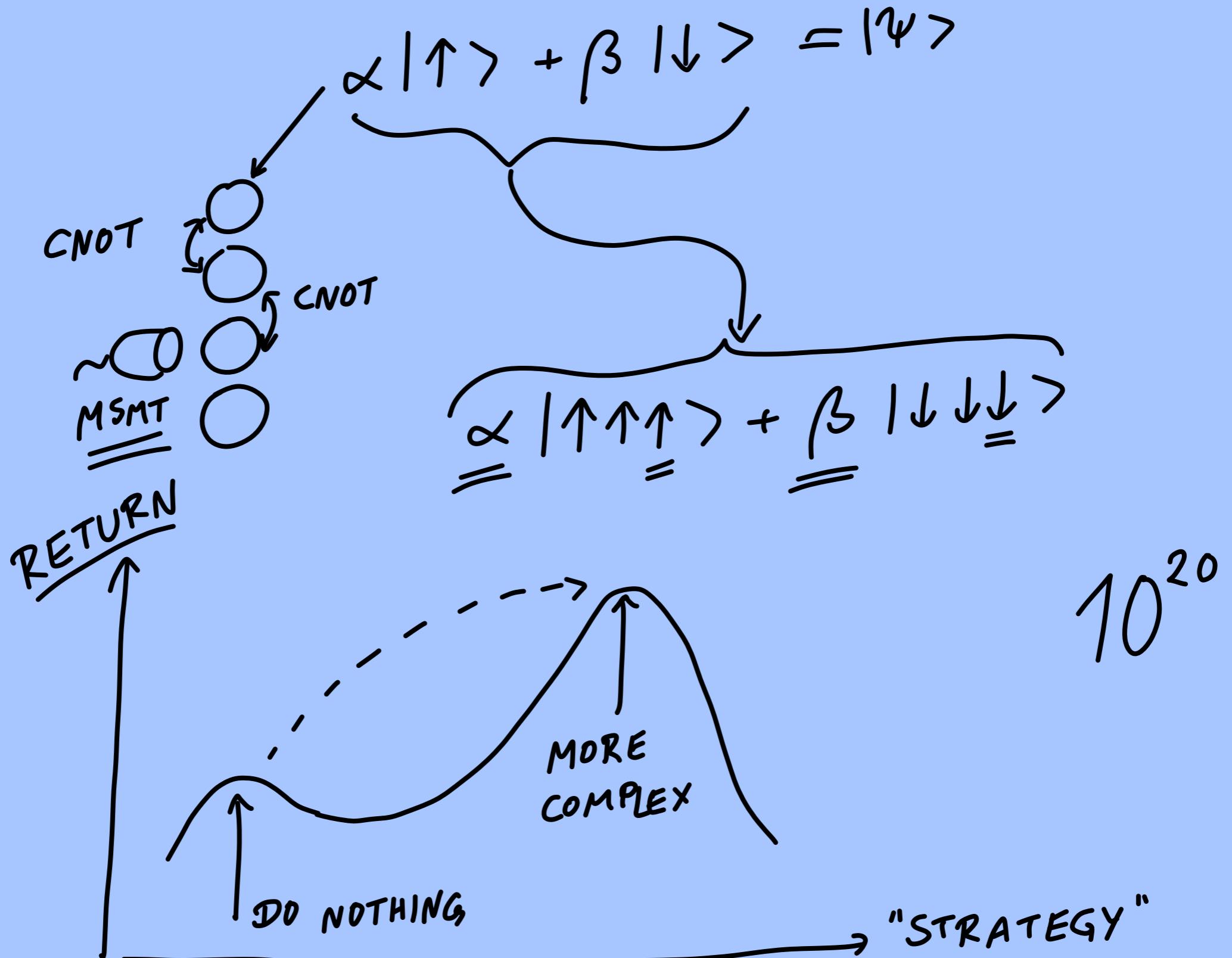
"TWO-STAGE LEARNING"

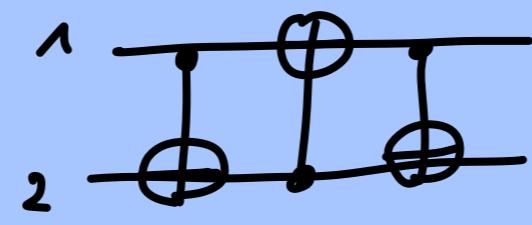


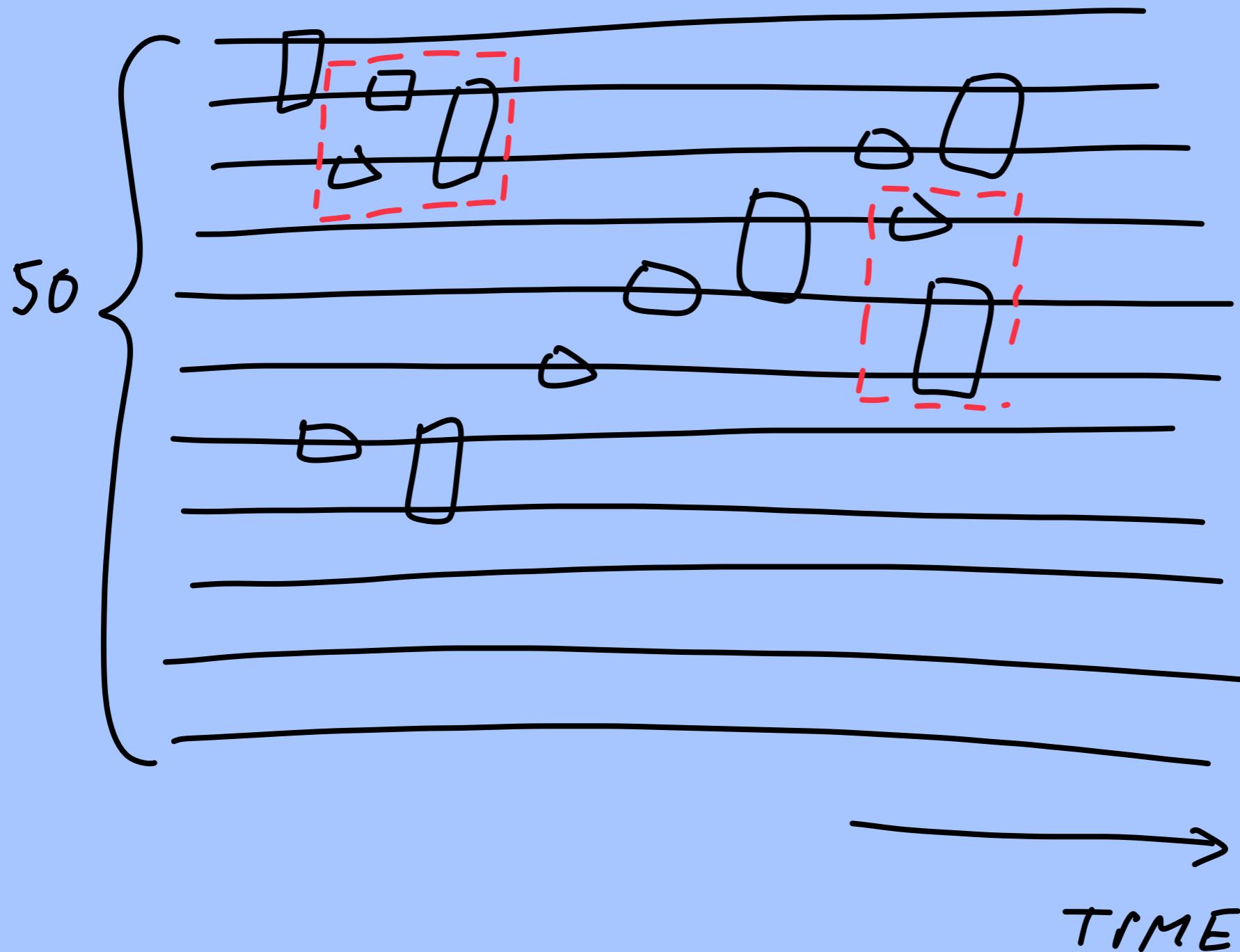


CNOT : FLIP SECOND QBIT<sub>2</sub>  
DEPENDING ON  
QBIT<sub>1</sub>

$$\uparrow \downarrow - \downarrow \uparrow$$





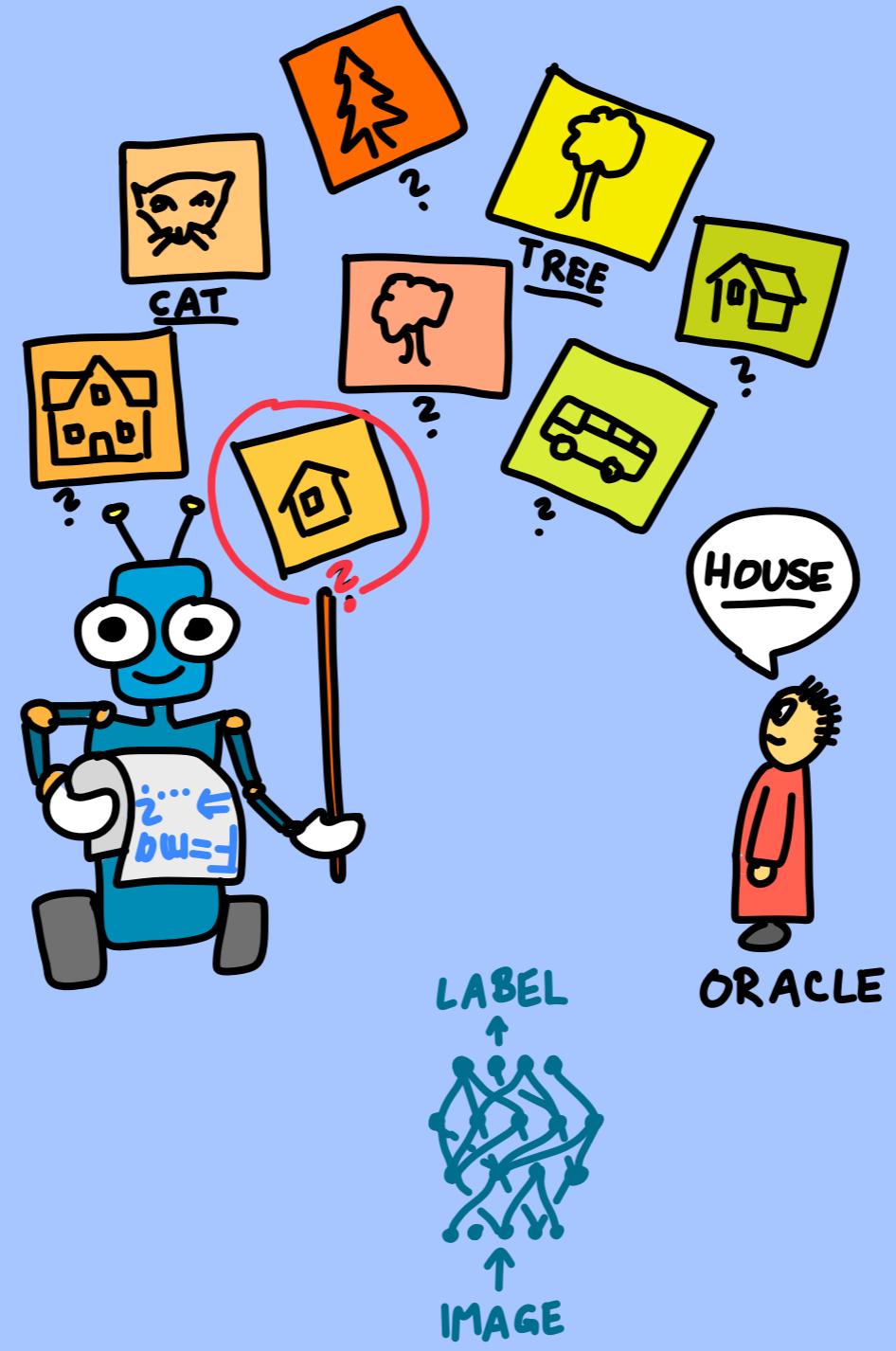


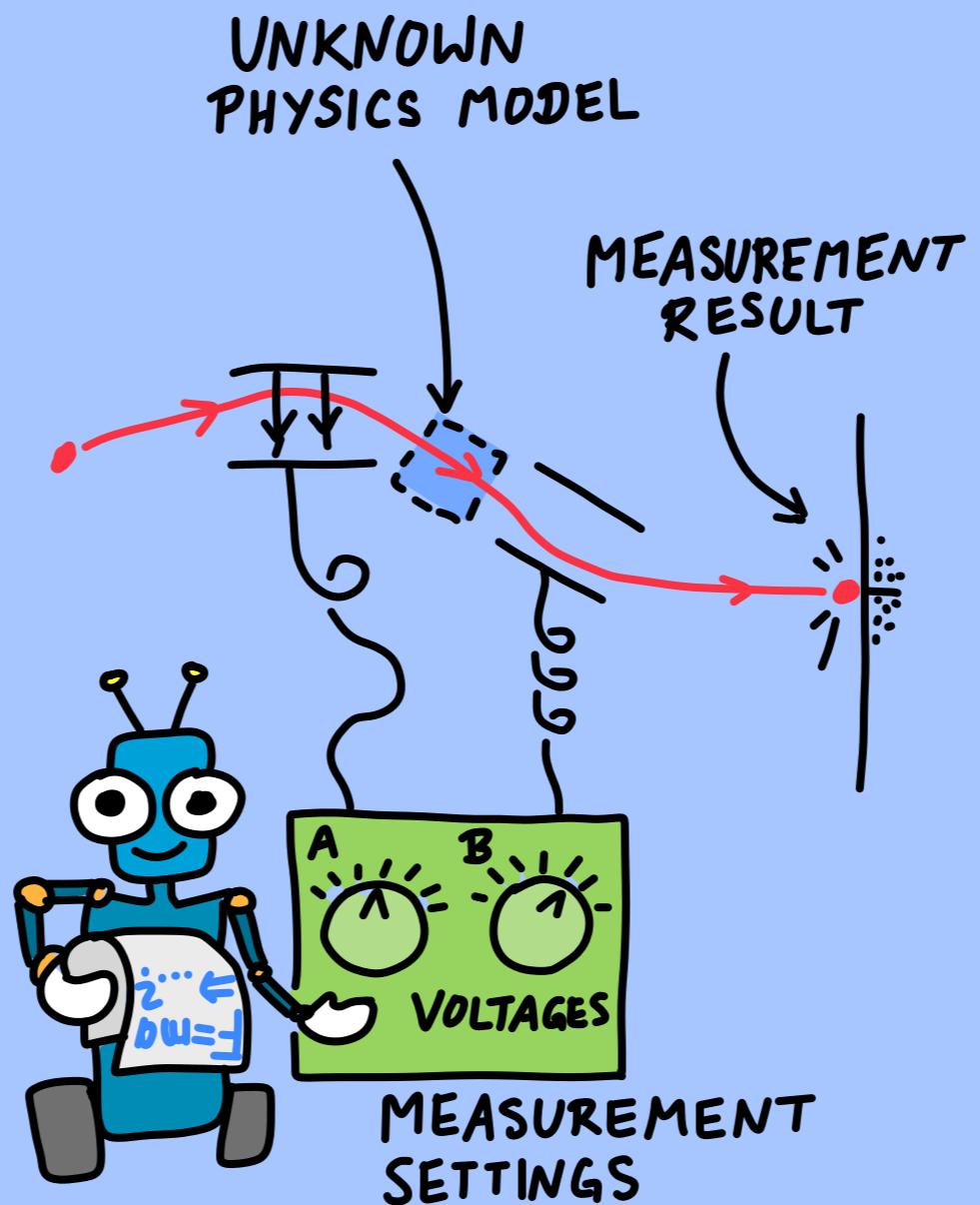
10.

## ACTIVE LEARNING / OPTIMAL EXPERIMENTAL DESIGN

CHOOSE TRAINING SAMPLES/  
MEASUREMENTS

⇒ BE MORE EFFICIENT  
IN LEARNING!





10.1

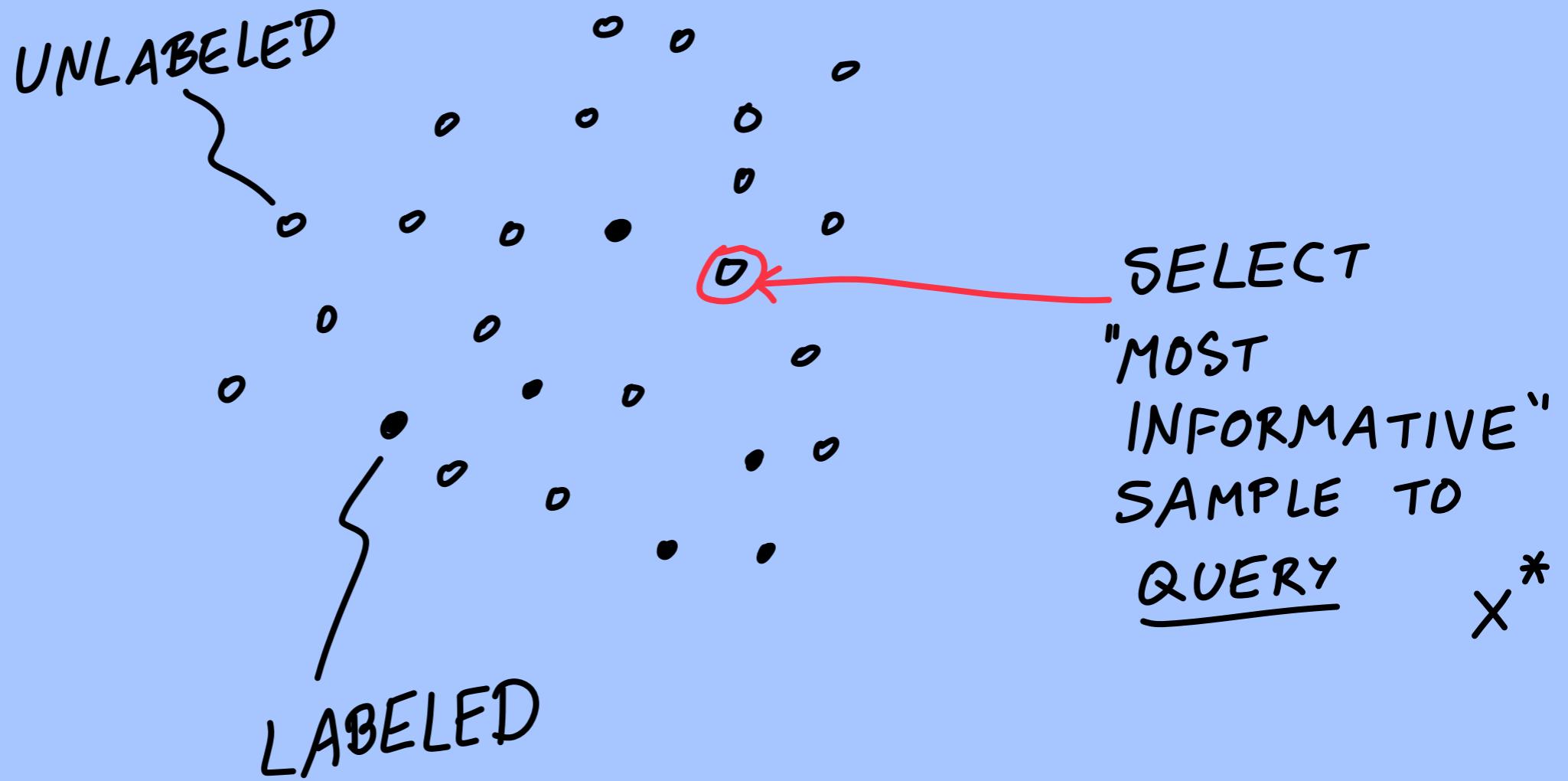
## ACTIVE LEARNING IN SUPERVISED TRAINING (NN OR OTHER MODELS)

MODEL:

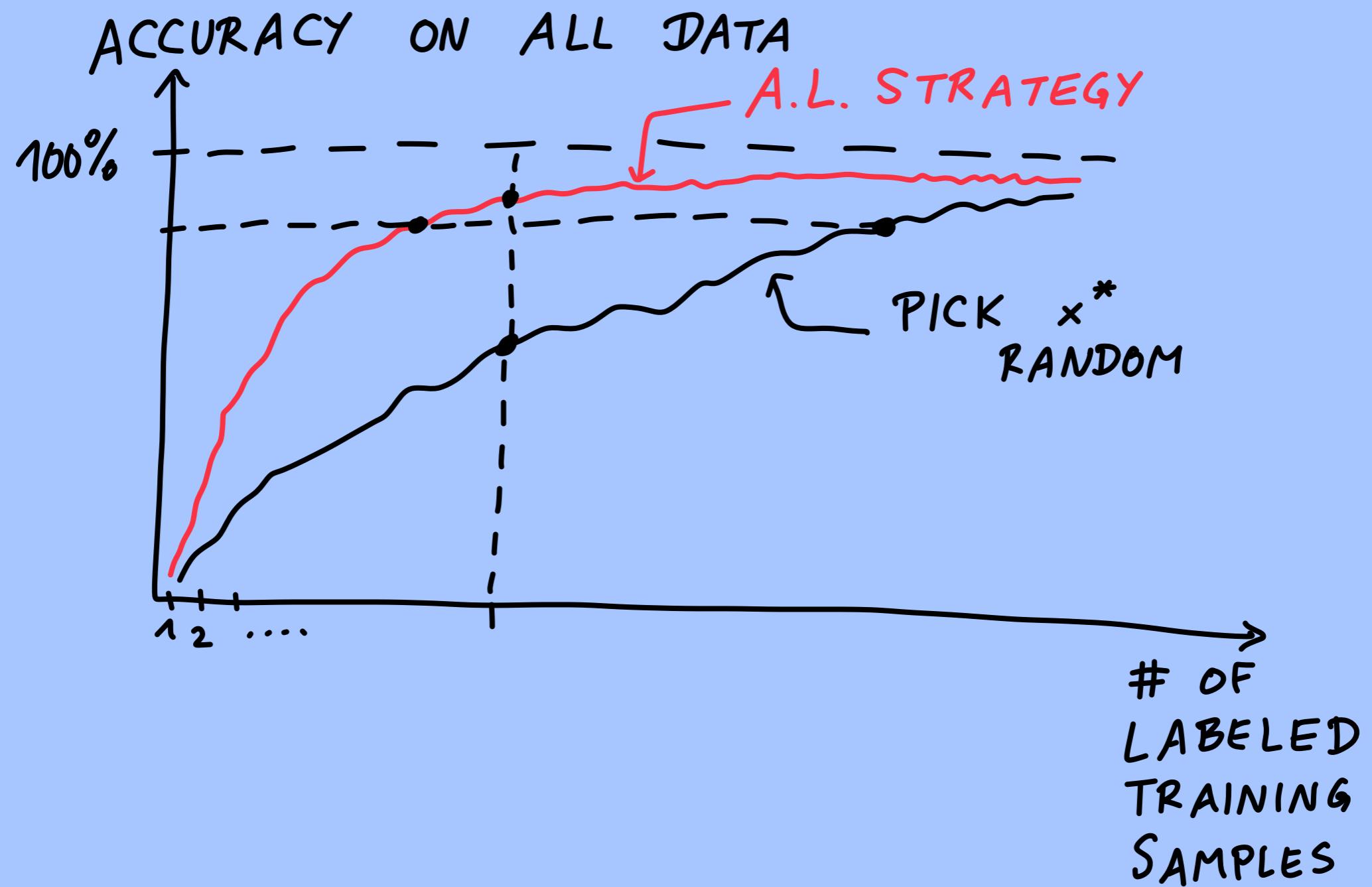
$$y^{\text{PRED}} = F_{\theta}(x)$$

PREDICTION      INPUT  
                        MODEL PARAMETERS

BASIC SITUATION:



GOAL : STRATEGY FOR PICKING 'BEST' X  
GIVEN MODEL &  
"ALL DATA TO OPTIMIZE  
"LEARNING SUCCESS"  
EXPENSIVE: LABELS                    INEXPENSIVE: MODEL  
    PREDICTIONS



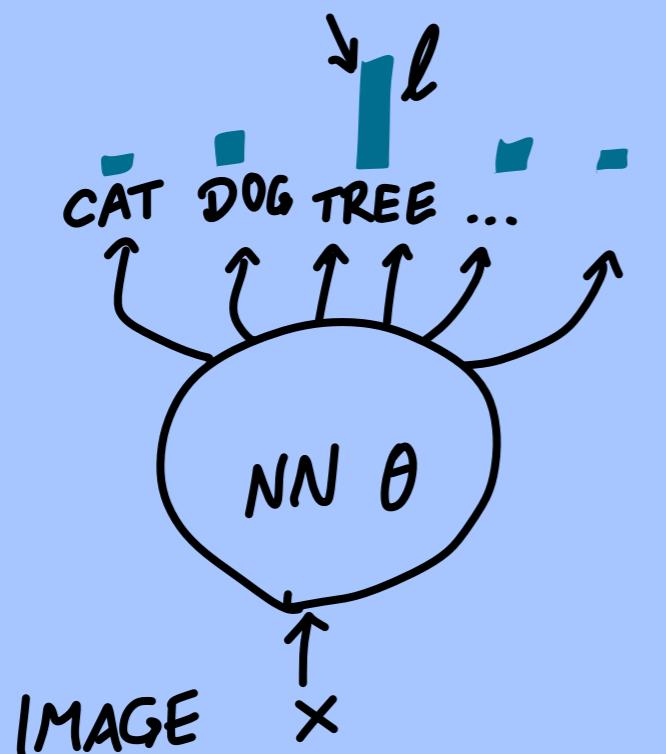
SIMPLE  
& EFFICIENT:

## UNCERTAINTY SAMPLING

PICK  $x$  WITH LARGEST UNCERTAINTY  
OF MODEL PREDICTION

HOW TO ESTIMATE UNCERTAINTY?

"EASY" FOR CLASSIFICATION TASK:



INTERPRET NN-OUTPUT  
AS PROB. DISTR.!

$$P_\theta(\text{LABEL } \ell | x) = [F_\theta(x)]_\ell$$

⇒ TRUE AFTER TRAINING  
WITH CATEG. CROSS-ENTROPY

$$-\left\langle \ln P_\theta(\text{TRUE LABEL } \ell | x) \right\rangle_{x \sim \text{DATA SET}}$$

PICK  $x$  WITH:

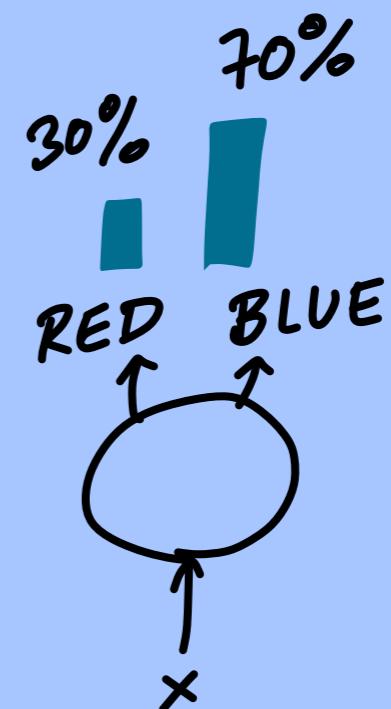
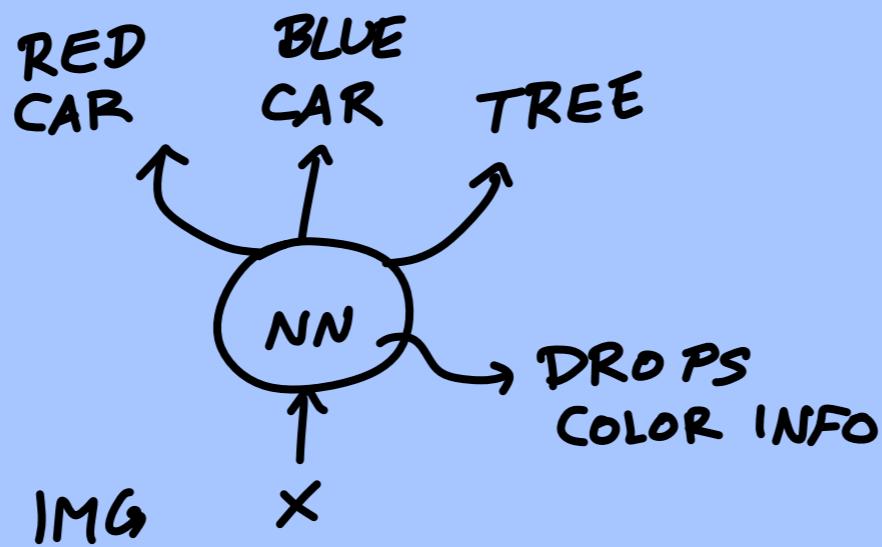
- MAXIMUM ENTROPY OF  $P_\theta(\text{LABEL}|x)$
- SMALLEST VALUE OF  
 $\max_l P_\theta(l|x)$
- SMALLEST DIFFERENCE  
BETWEEN TOP TWO P  
"MARGIN SAMPLING"

NOTE: • AT START OF TRAINING:  
OUTPUT  $P(\text{LABEL}_1 \times)$  RANDOM  
 $\Rightarrow$  NOT TRUE MEASURE OF  
UNCERTAINTY

-----

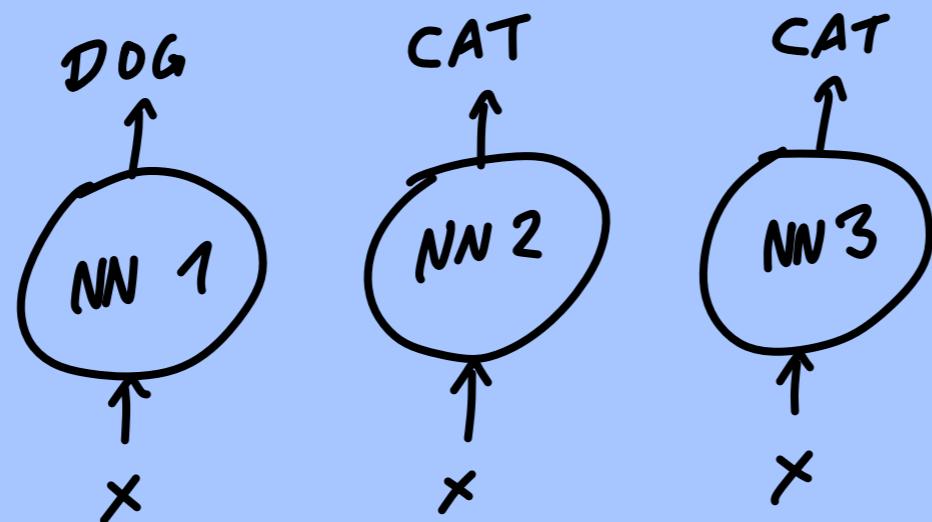
- LATER: OK

### EXAMPLE



NO DIRECT ESTIMATE OF UNCERTAINTY  
FROM MODEL  $\Rightarrow ?$

### COMMITTEE OF MODELS



TRAIN SEVERAL MODELS  
(DIFF. INITIAL CONDITIONS  $\theta$ ,  
MAYBE ALSO DIFF. TRAINING SAMPLES)

PICK  $x$  WITH MAXIMUM DISAGREEMENT!

REGRESSION:

$$\bar{F}(x) = \frac{1}{N} \sum_j F_{\theta_j}(x)$$

$$\text{MAXIMIZE}_{(\text{OVER } x)} : \frac{1}{N-1} \sum_j (F_{\theta_j}(x) - \bar{F}(x))^2$$

CLASSIFICATION:  $\bar{P}(x) = \frac{1}{N} \sum_j P_{\theta_j}(l|x)$

$$\text{MAXIMIZE} : \sum_j D_{KL}(\bar{P} || P_{\theta_j})$$

EXPECTED MODEL CHANGE

TRAINING SAMPLE  $(x, \overset{\text{TRUE LABEL}}{y}) \Rightarrow$

$$\delta\theta = -\gamma \frac{\partial \mathcal{L}_\theta(x, y)}{\partial \theta}$$

EXAMPLE:

$$\mathcal{L}_\theta(x, y) = (F_\theta(x) - y)^2$$

FOR  
MEAN-SQ.-ERROR

IDEA: PICK  $x$  WHICH MAXIMIZES  $\|\delta\theta\|$

BUT  $y = ?$

$$x^* = \arg \max_x \left\langle \left\| \frac{\partial \mathcal{L}_\theta(x, y)}{\partial \theta} \right\| \right\rangle_{y \sim P_\theta(y|x)}$$

PROBLEM:  $\|\theta\|$  IS NOT DIRECTLY RELATED TO PERFORMANCE IMPROVEMENT

→ IDEA: MAXIMIZE EXPECTED ERROR REDUCTION

WHICH ERROR?

OPTION: EXPECTED ERROR REDUCTION ON UNLABELED SET, ACCORDING TO MODEL

EXAMPLE: CLASSIFICATION

$$x^* = \underset{x}{\operatorname{argmin}} \left\langle \left\langle H \left( P_{\theta(x,y)}(\text{LABEL} | \tilde{x}) \right) \right\rangle \right\rangle_{\tilde{x} \in \text{UNLABELED SET}} y \sim P_{\theta}(y|x)$$

$\theta(x,y) = \theta$  AFTER TRAINING UPDATE FOR NEW SAMPLE  $(x,y)$

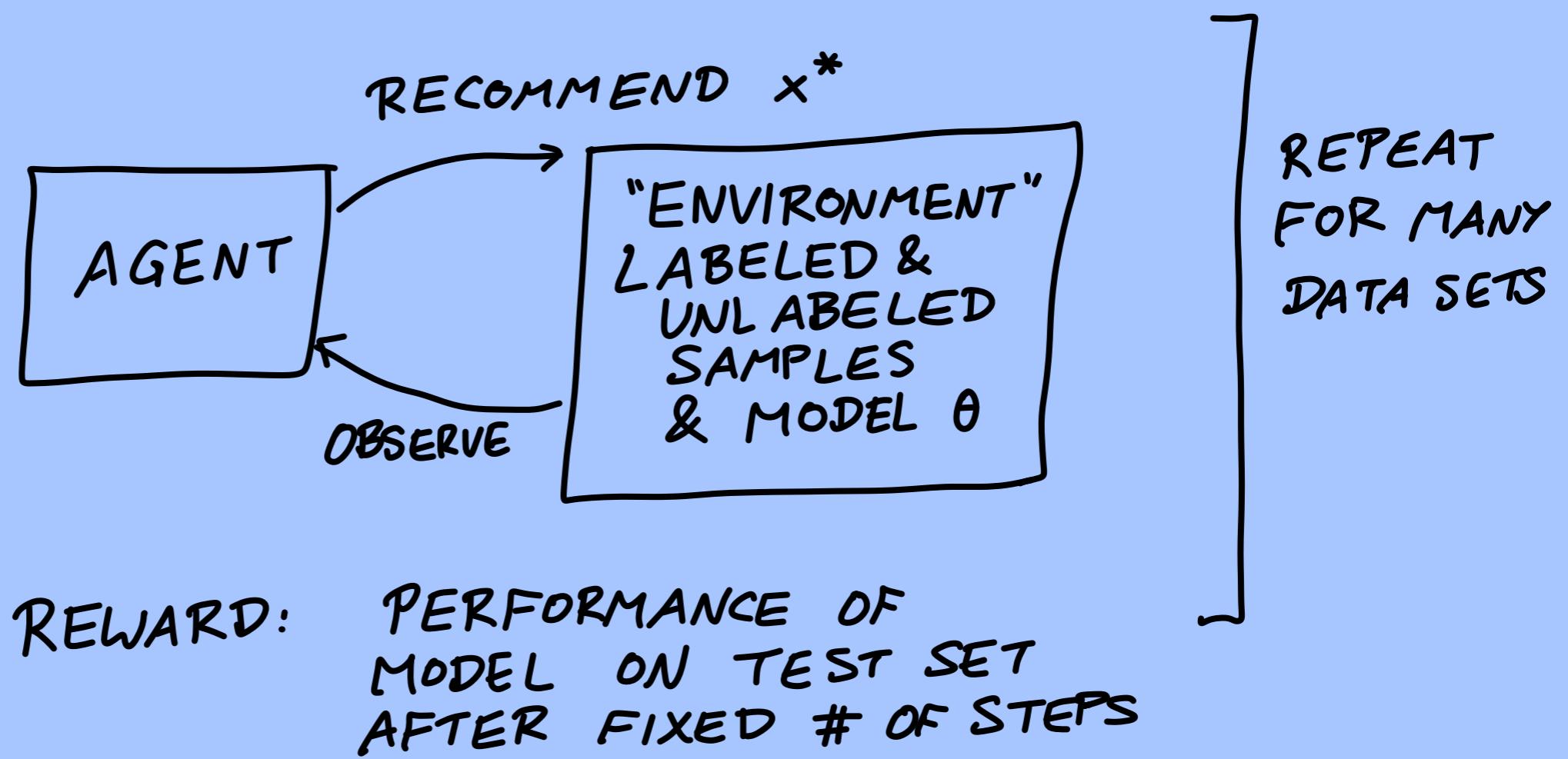
OPTION: EXPECTED  
GENERALIZATION ERROR  
ESTIMATED FROM VALID. SET

$$x^* = \underset{x}{\operatorname{argmin}} \left\langle \sum_{\theta(x,y)} (\tilde{x}, \tilde{y}) \right\rangle_{(x,y) \in \text{VALID. SET}} y \sim P_\theta(y|x)$$

⇒ THIS IS WHAT WE WANT  
... BUT VERY EXPENSIVE

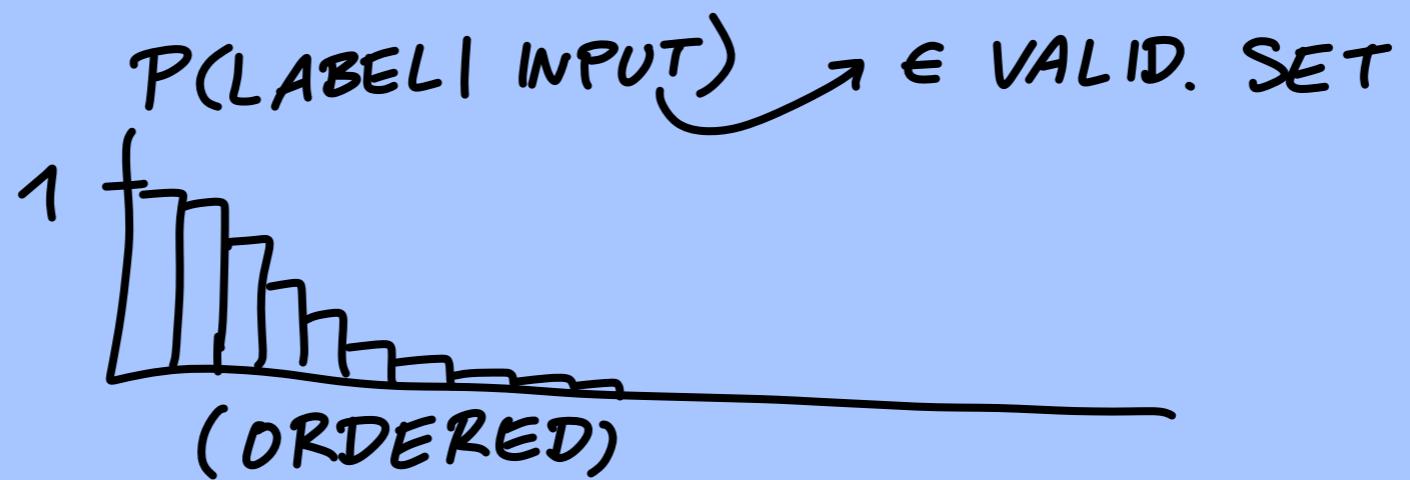
(ALL  $x$  AND ALL  $\tilde{x}$ )

# ACTIVE LEARNING FOR RL



GOAL : FIND STRATEGY THAT WORKS FOR NEW DATASET

IDEA :



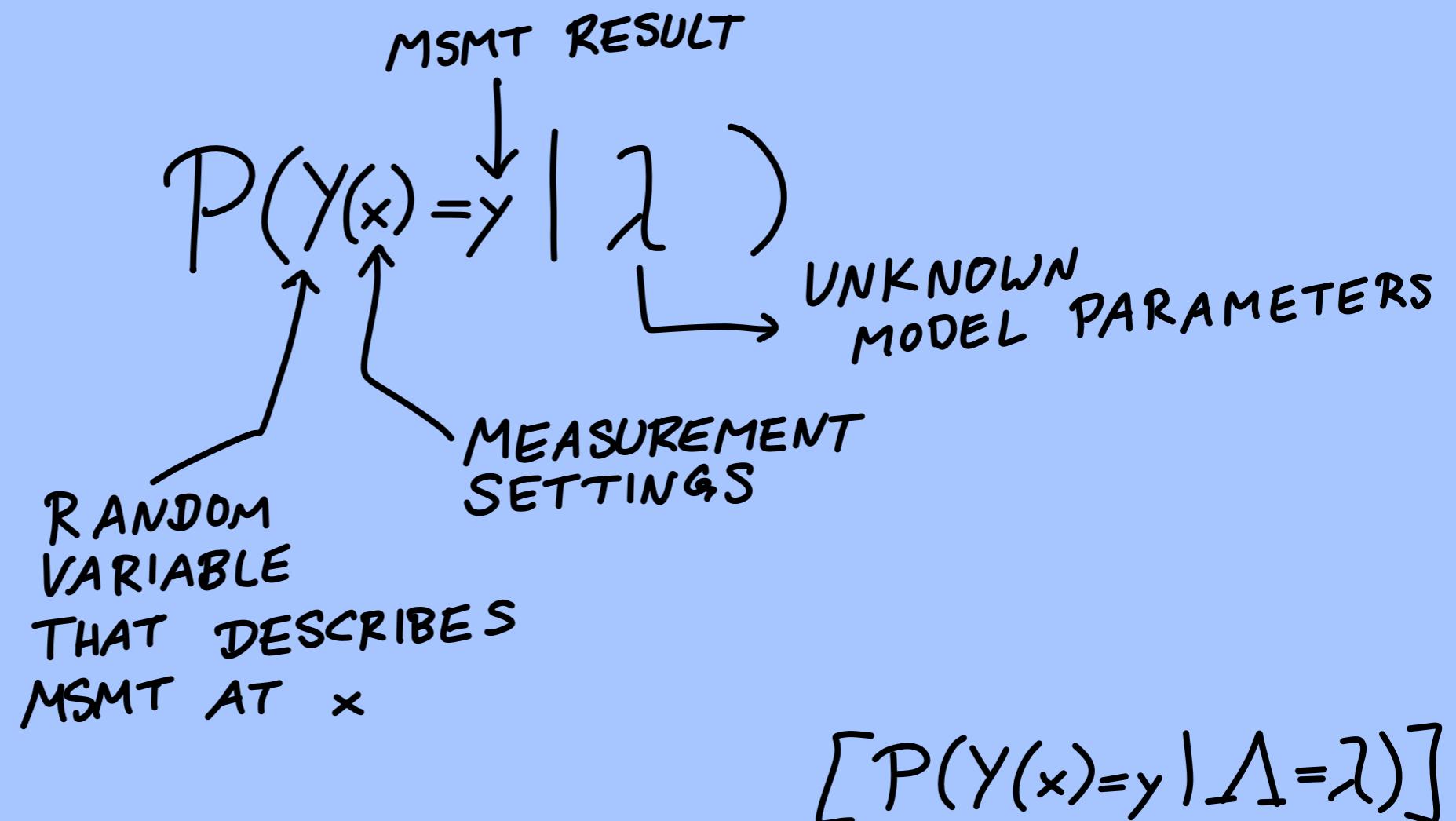
ACTIONS  $\alpha$ :

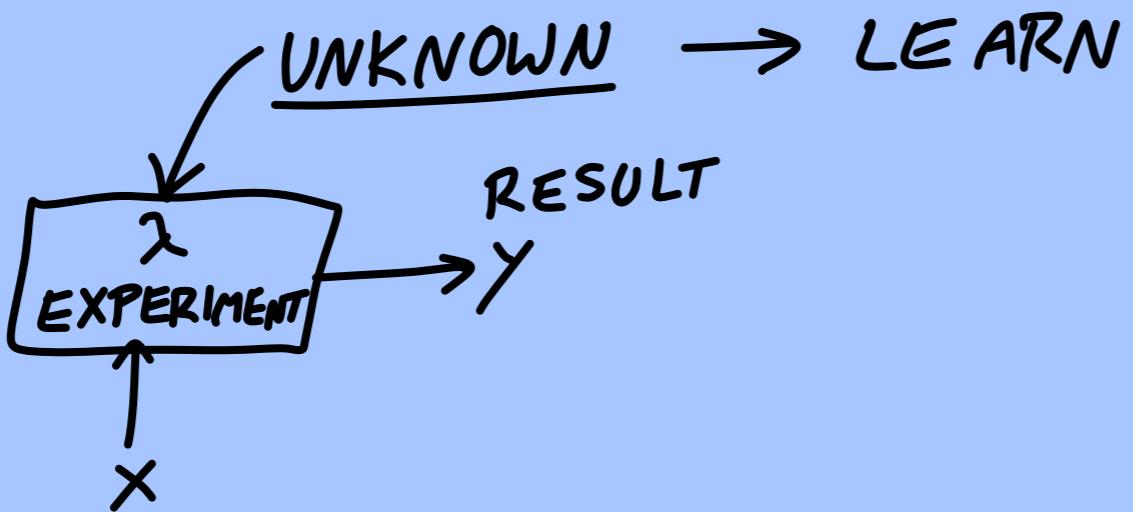
- FOR EACH UNLABELED  $x$ :
- $\alpha$  FROM PRED. OF MODEL  
(& DISTANCE TO NEAREST LABELED SAMPLES)

PICK  $x$  WITH HIGHEST  $\alpha$

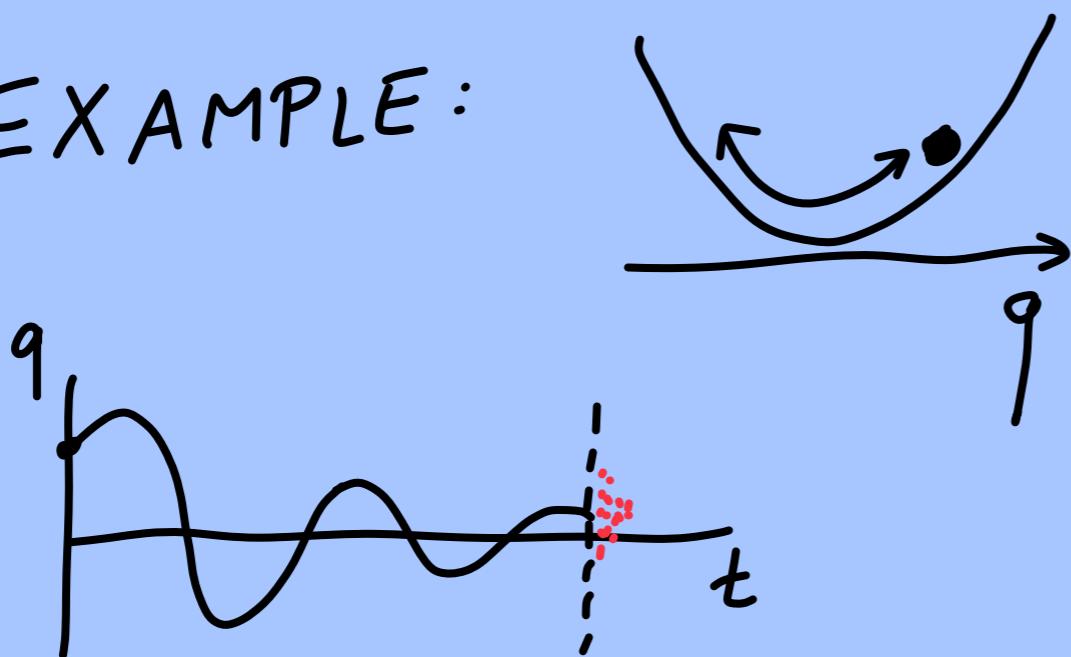
10.2

## BAYESIAN OPTIMAL EXPERIMENTAL DESIGN





EXAMPLE :

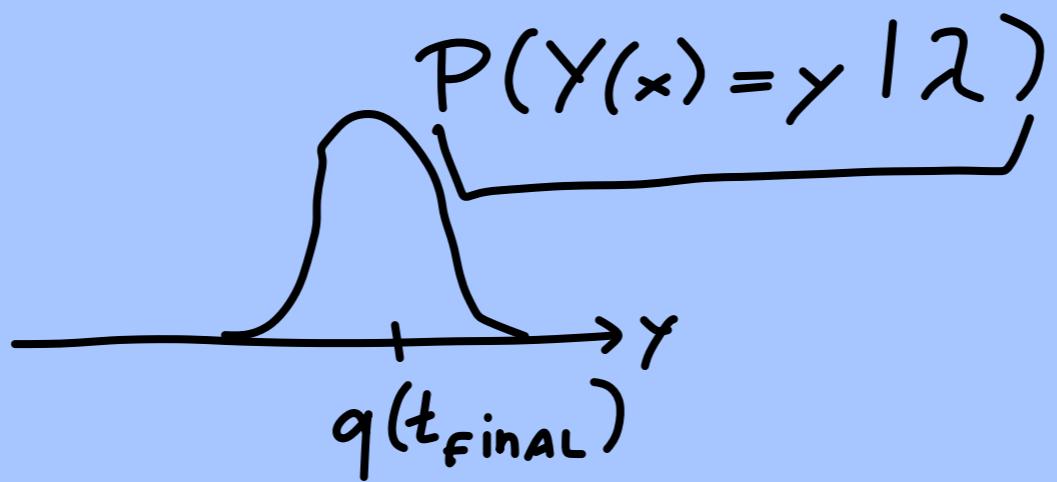


$$\ddot{q} = -\omega^2 q - \gamma \dot{q}$$

$$\lambda = (\omega, \gamma)$$

$$x = (q(t=0), \dot{q}(t=0))$$

$$y = q(t_{FINAL}) + \text{NOISE}$$

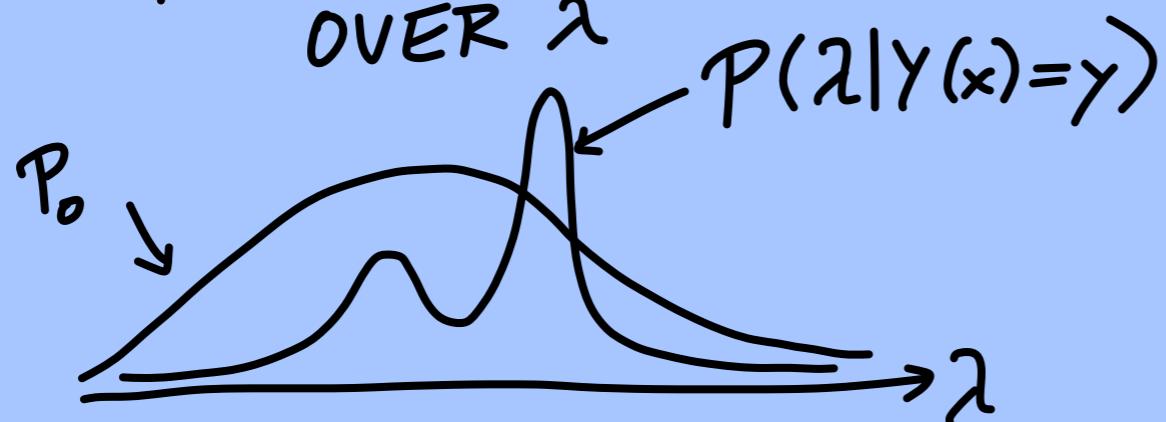


BAYES UPDATE :

$$P(\lambda | Y(x)=y) = \frac{P(Y(x)=y|\lambda) \cdot P_0(\lambda)}{P(Y(x)=y)}$$

$$P(Y(x)=y) = \int d\lambda P(Y(x)=y|\lambda) P_0(\lambda)$$

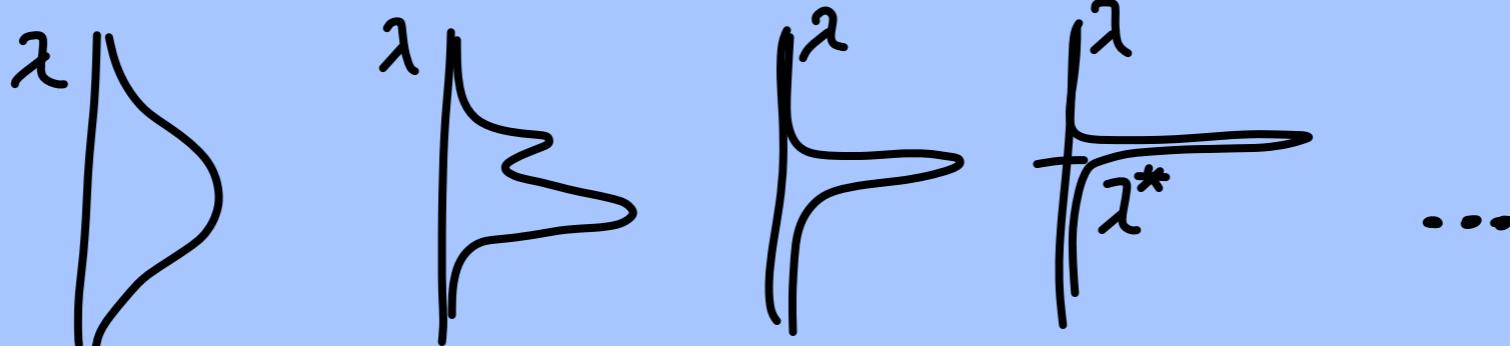
POSTERIOR  
OVER  $\lambda$



NOTE: THERE  
IS ONE TRUE  
VALUE  $\lambda^*$

$$y \sim P(Y(x)=y | \lambda^*)$$

POSTERIOR BECOMES PRIOR FOR NEXT MSHT



INFORMATION GAIN:

$$IG(Y(x)=y) = D_{KL}(P(\lambda|Y(x)=y) \parallel P_0(\lambda))$$

EXPECTED INF. GAIN: AVERAGE OVER Y

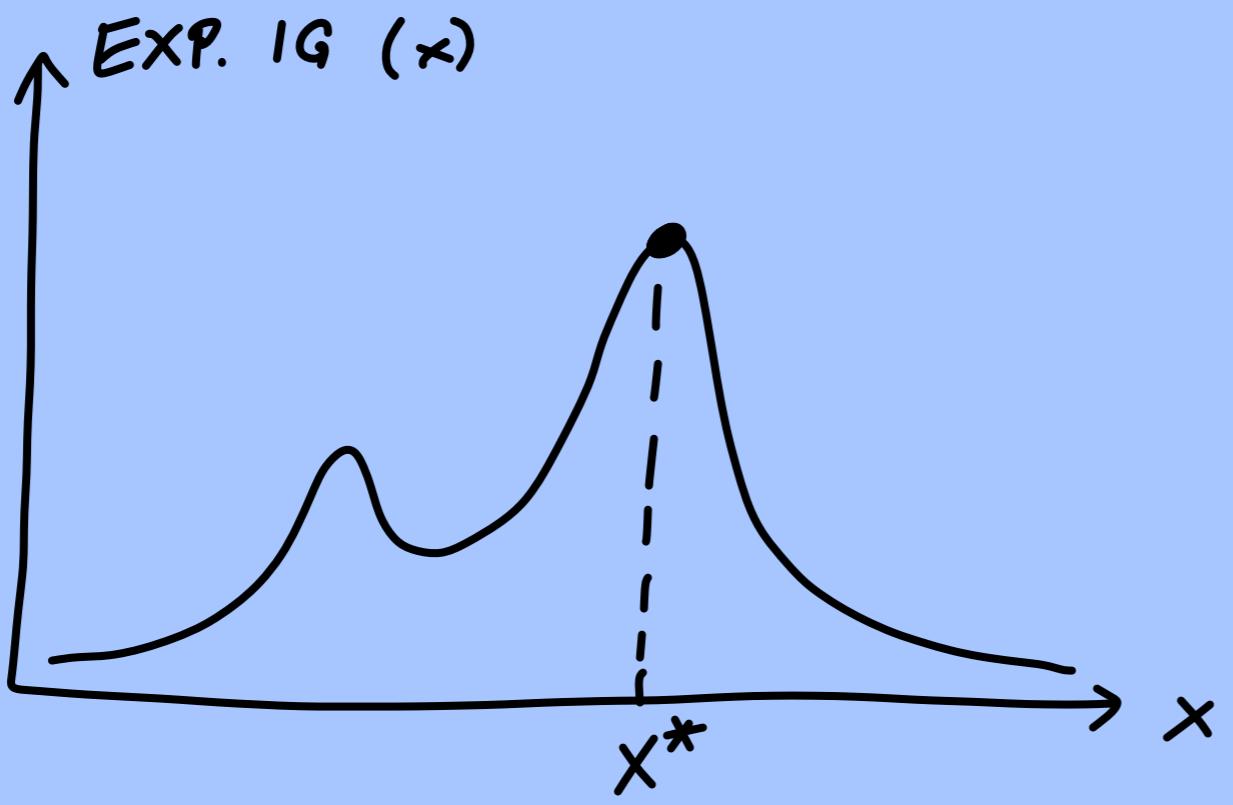
= EXPECTED REDUCTION OF ENTROPY

$$= \underbrace{H(\lambda)}_{\text{OF PRIOR}} - \underbrace{H(\lambda|Y(x))}_{\text{COND. ENTROPY (AVG. OVER Y)}}$$

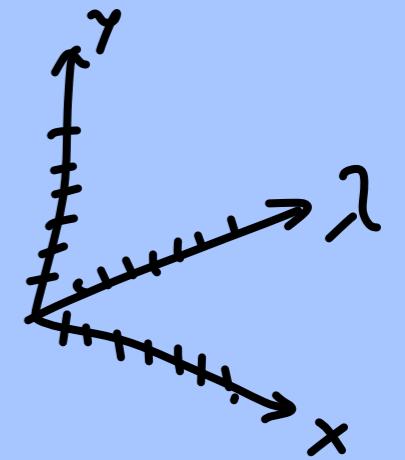
$$= - \int d\lambda P_0(\lambda) \ln P_0(\lambda)$$

$$+ \int dy P(y) \int d\lambda P(\lambda|Y(x)=y) \ln P(\lambda|Y(x)=y)$$

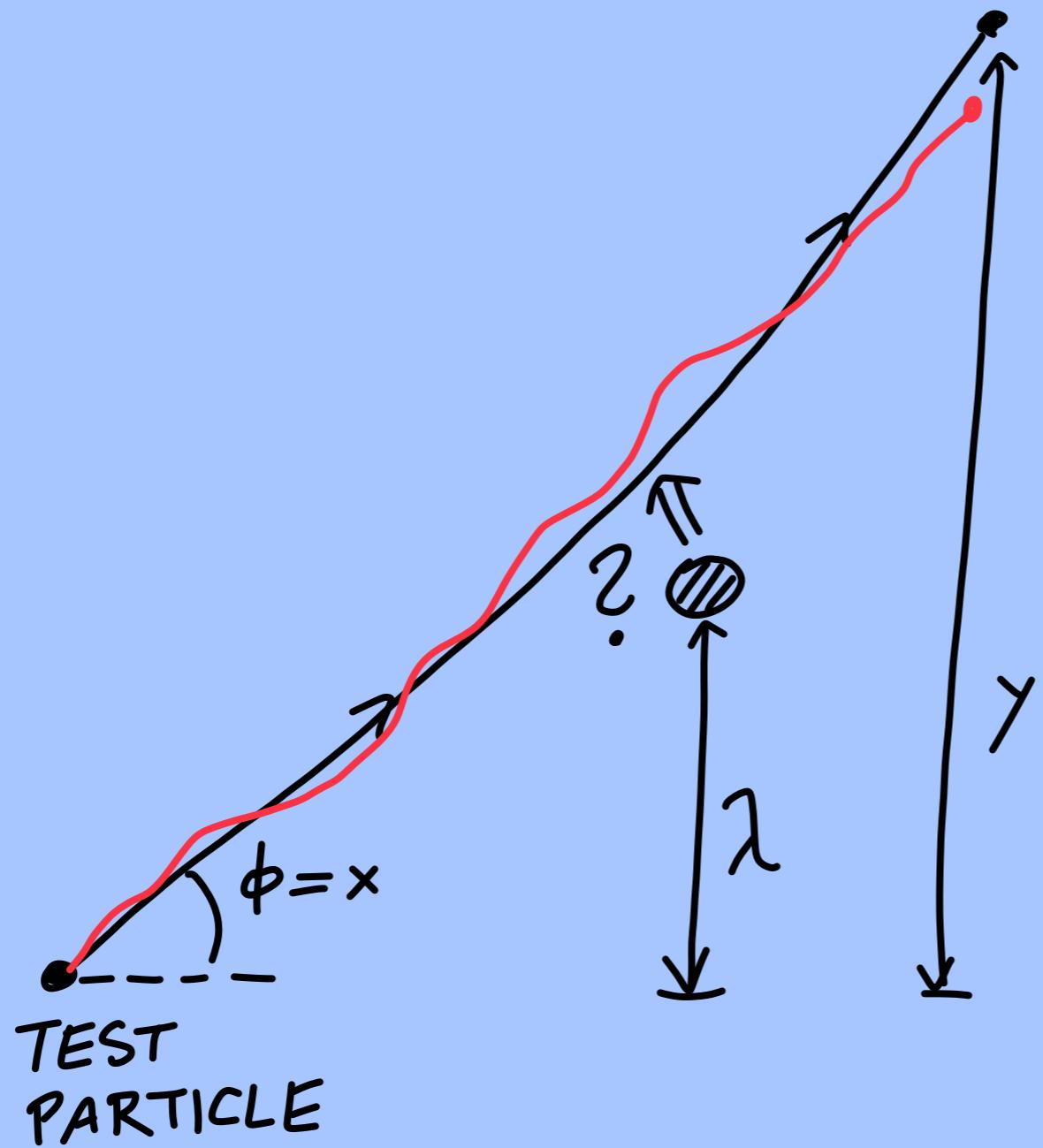
$$= MI(\lambda, Y(x))$$

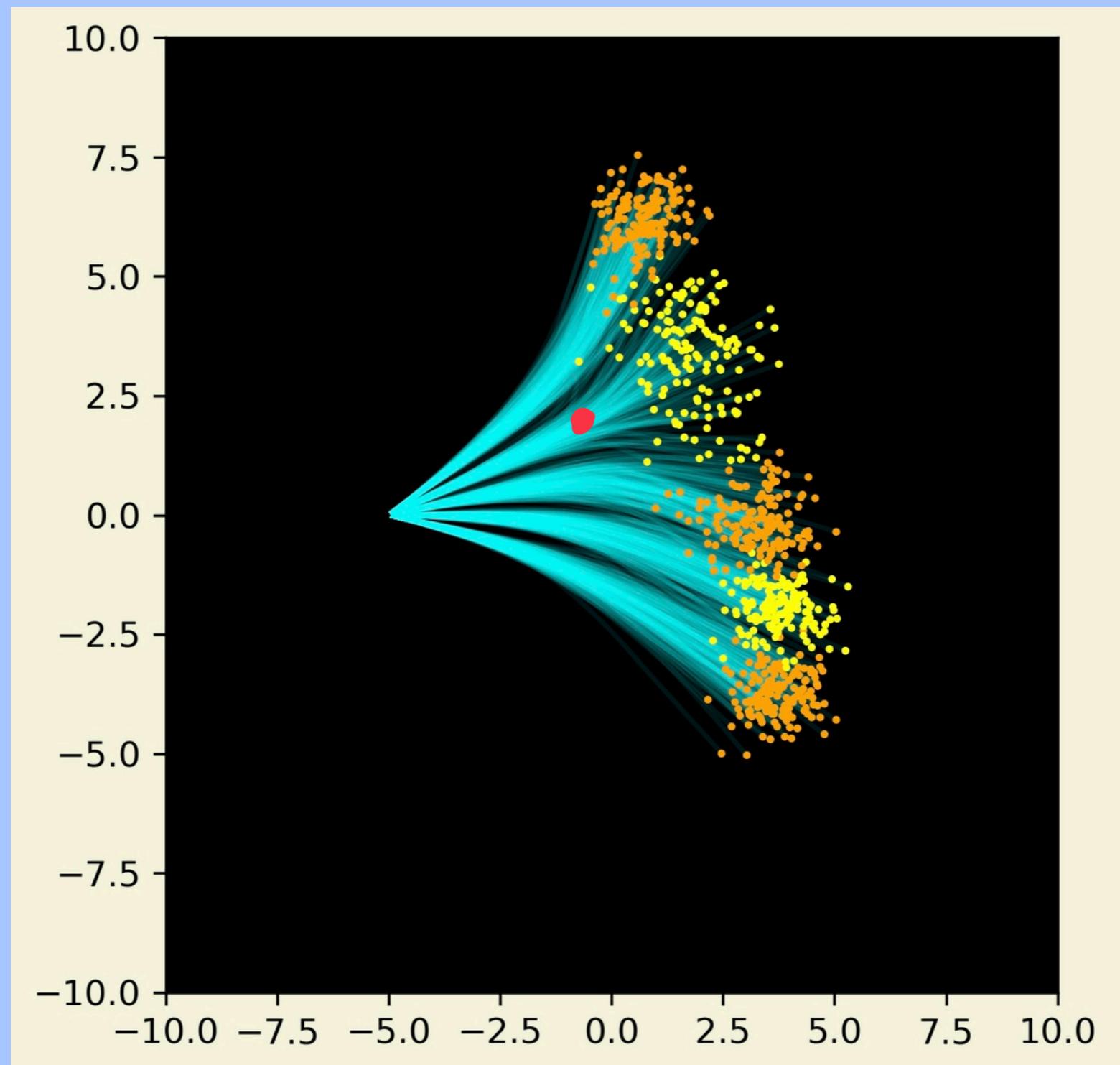


$\lambda, x, y$  LOW.-DIM.  $\rightarrow$  GRID-BASED

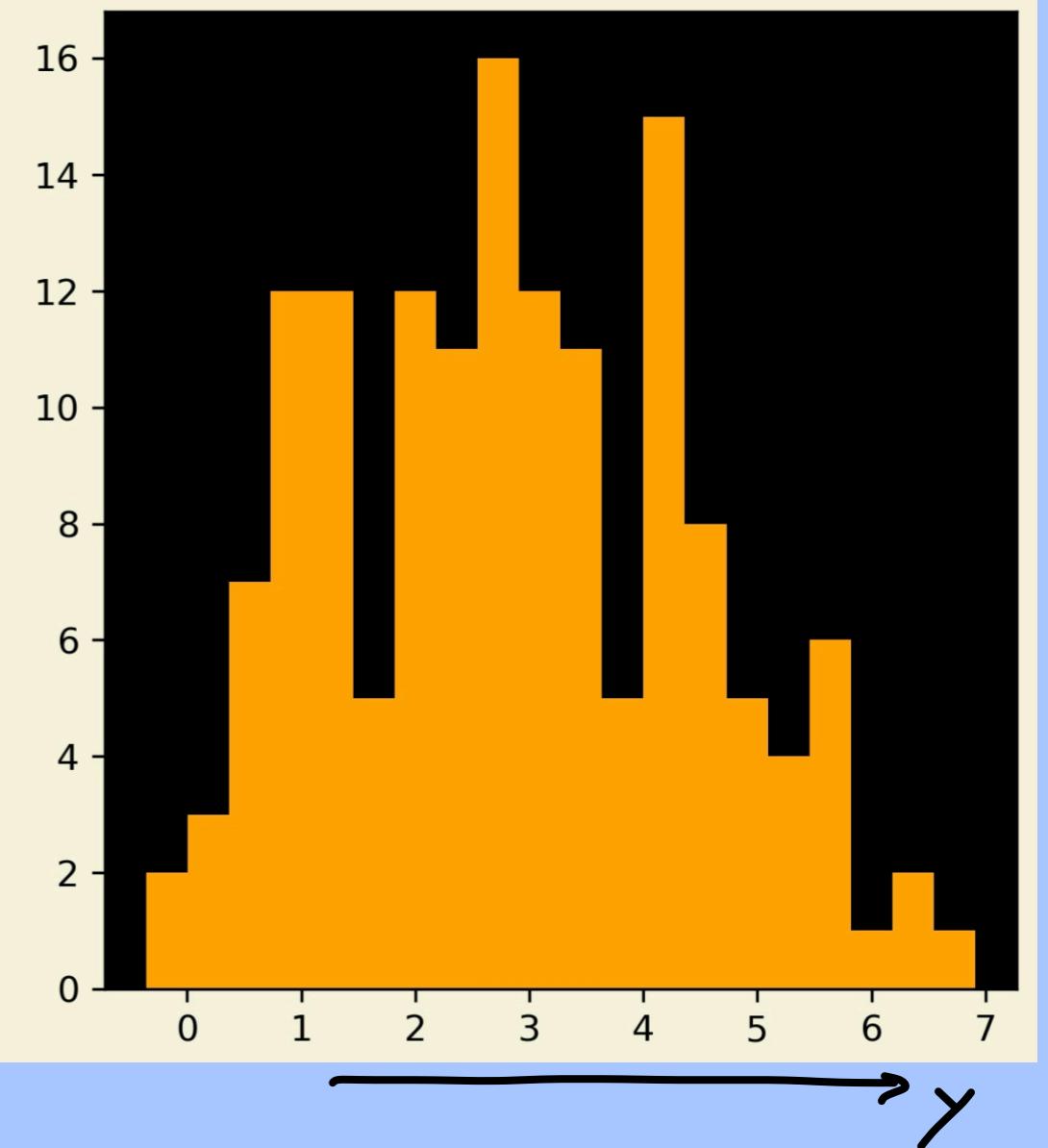
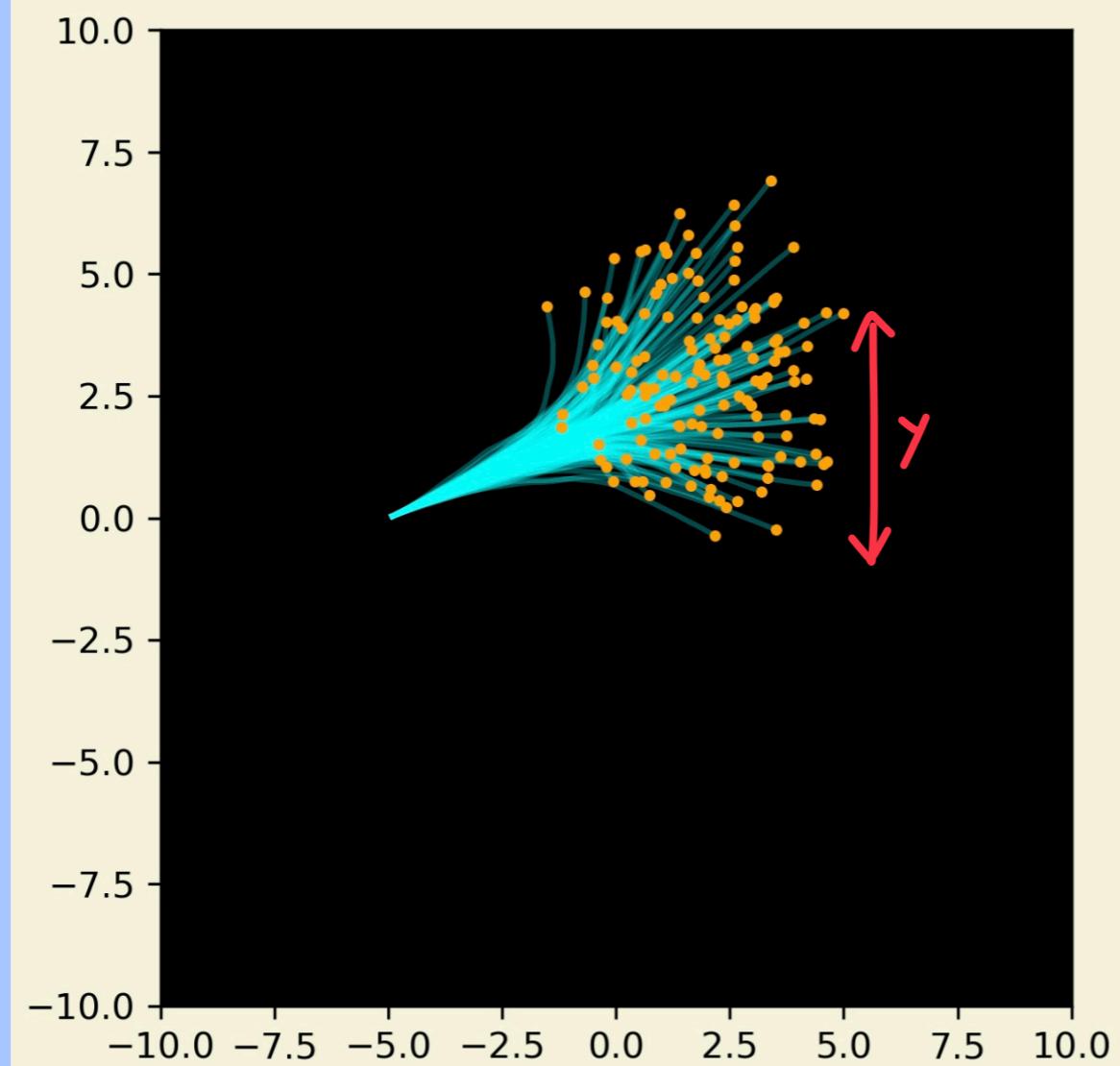


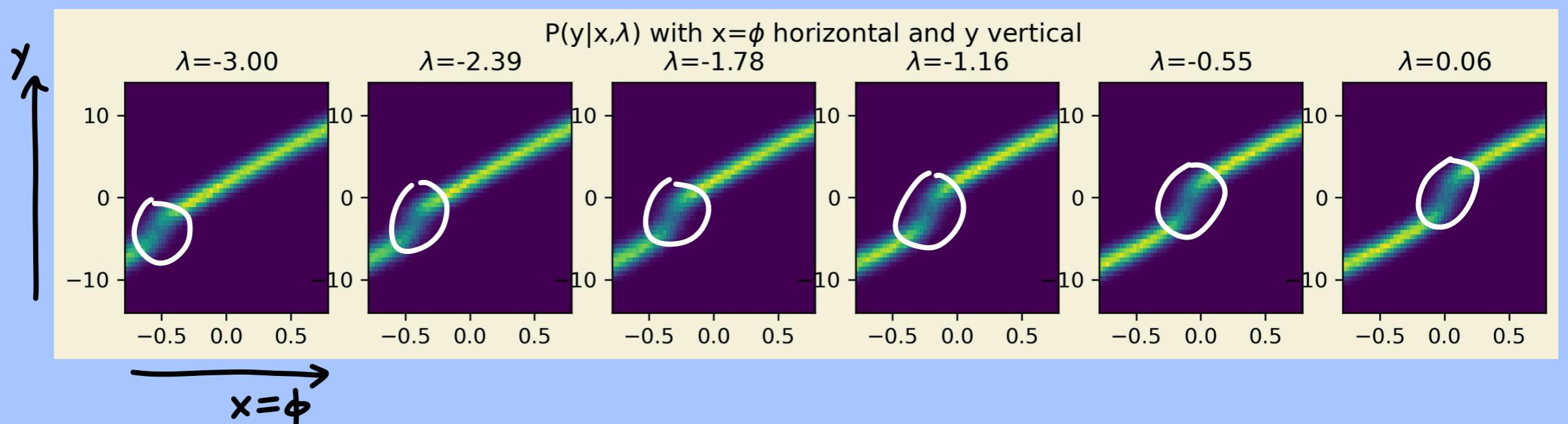
$$P(y(x)=y|\lambda)$$

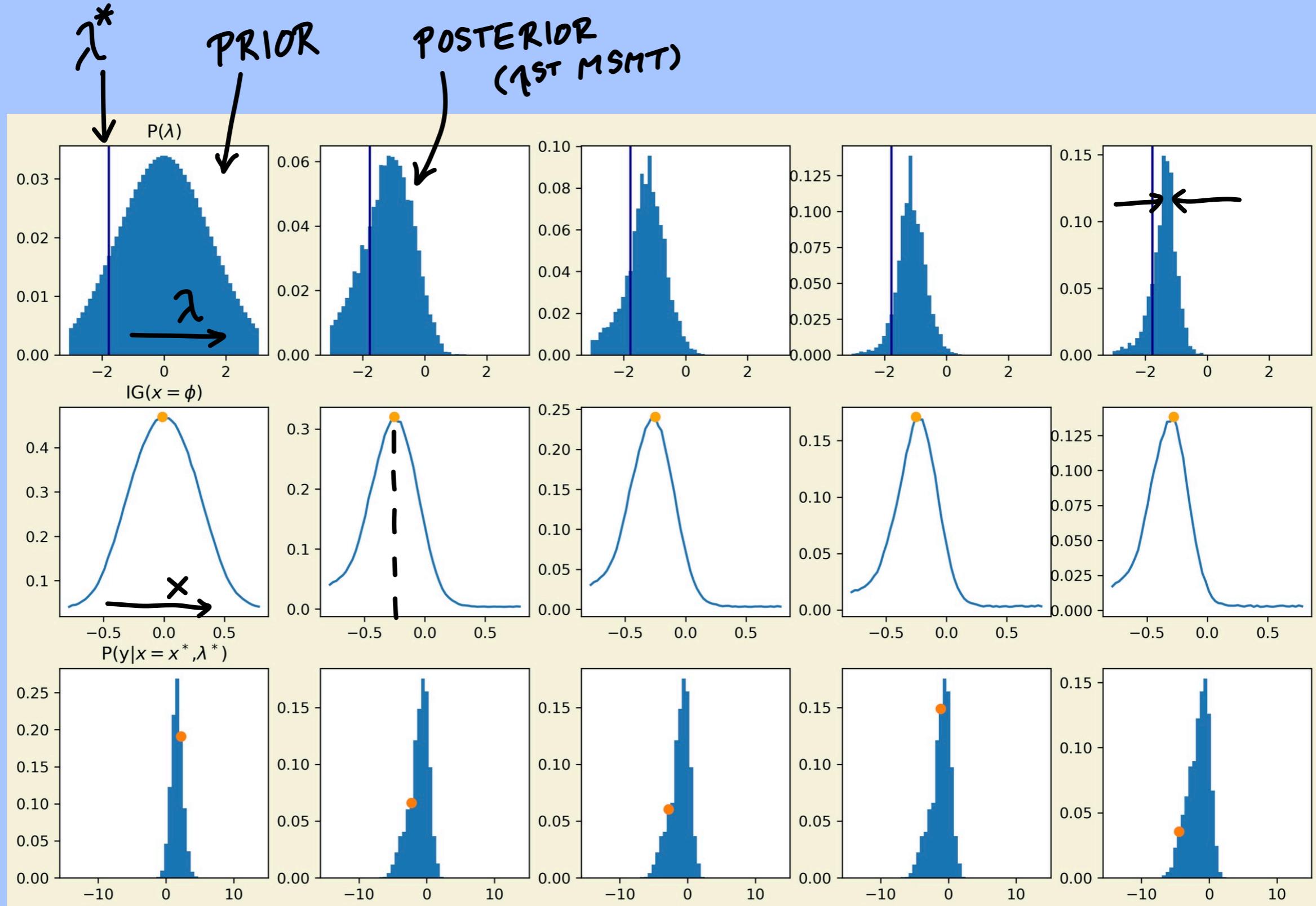




$$\mathcal{P}(Y(x)=y \mid \lambda^*)$$







$y$  HIGH-DIM.  $\rightarrow$  SAMPLE  $y$

$\lambda$  HIGH-DIM.  
(AND/OR  $x$ )  $\rightarrow$  NORMALIZING FLOW  
 $P(\lambda | Y(x)=y)$

OPTIMIZING  $x$ :

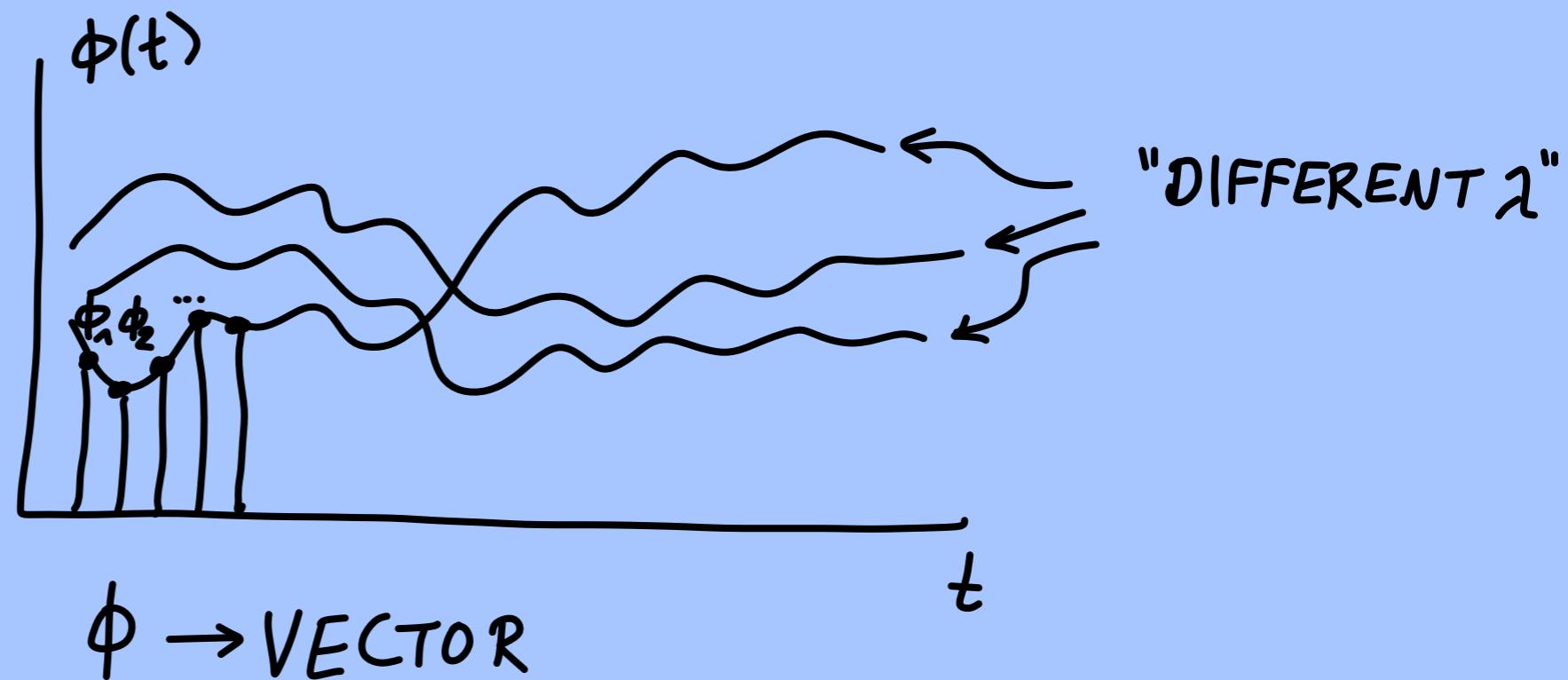
- GRID: OK ✓
- GRAD. ASC. IN IG
- GRAD. -FREE METHODS

10.3

## BAYESIAN ACTIVE LEARNING FOR GAUSSIAN RANDOM PROCESSES AND NEURAL NETWORKS

→ HIGH.-DIM. MODELS!

### GAUSSIAN RANDOM PROCESSES



BAYES

$$P(\phi \mid Y(t_1)=y_1, Y(t_2)=y_2, \dots, Y(t_M)=y_M)$$

$$Y(t) = \phi(t) + \underbrace{\text{NOISE}}_{\text{(GAUSSIAN)}}$$

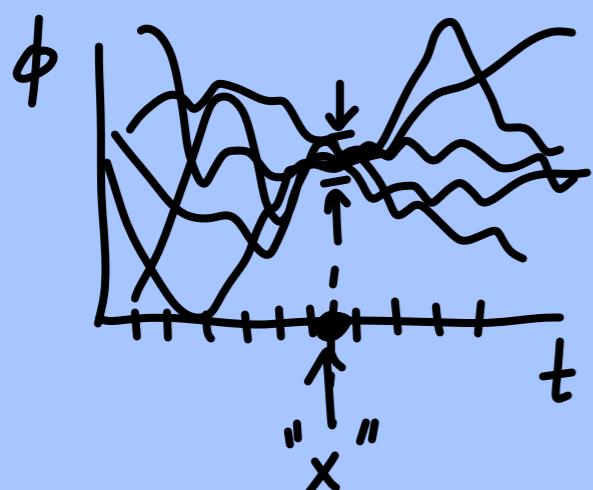
$$P(\phi) = \frac{1}{\sqrt{(2\pi)^N / \det C}}} \exp \left[ -\frac{1}{2} (\phi - \bar{\phi})^T C^{-1} (\phi - \bar{\phi}) \right]$$

(N: GRID POINTS) CORR.  
MATRIX

$$C_{ij} = \langle (\phi_i - \bar{\phi}_i)(\phi_j - \bar{\phi}_j) \rangle$$

$$H(P(\phi)) = \frac{N}{2}(1 + \ln 2\pi) + \frac{1}{2} \ln |\det C|$$

AFTER MSMT:



$$(C^{-1})^{NEW} = C^{-1} + \frac{1}{G^2} |x\rangle\langle x|$$

MSMT LOCATION

$$|x\rangle\langle x| = \begin{bmatrix} x & \cdot & \cdot \\ \cdot & \ddots & \cdot \\ \cdot & \cdot & 0 \end{bmatrix}$$

$$(|x\rangle\langle x|)_{nm} = S_{n,x} S_{m,x}$$

MSMT UNCERTAINTY

$\Rightarrow$  INDEP. OF MSMT RESULT  $y(x)$  !  
NO NEED TO  
AVG. OVER Y

$$H(P(\phi)) - H(P(\phi | \text{MSMT}))$$

$$= \frac{1}{2} \left\{ \ln |\det C| - \ln |\det C^{\text{NEW}}| \right\}$$

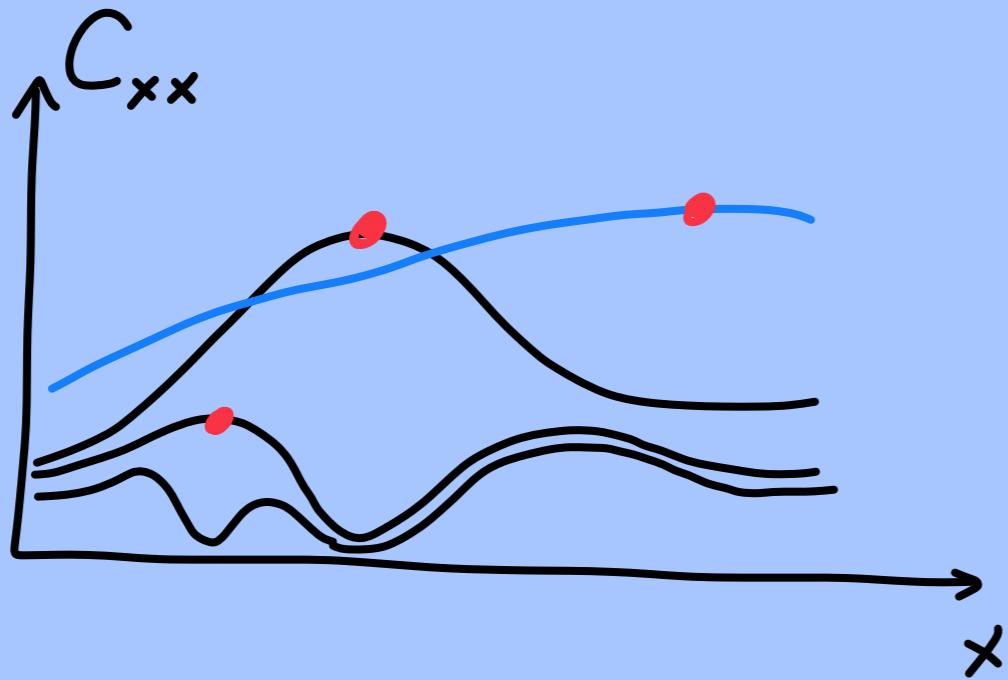
$$= \frac{1}{2} \ln \left| \det \underbrace{C(C^{\text{NEW}})^{-1}}_{C^{-1}} \right| + \frac{1}{2} \ln |x><x|$$

$$= \frac{1}{2} \ln \det \left[ I + \frac{1}{Z^2} C |x><x| \right]$$

$$= \frac{1}{2} \ln \left( 1 + \frac{\langle x | C | x \rangle}{Z^2} \right)$$


---

$C_{xx}$   
 $= \langle (\phi_x - \bar{\phi}_x)^2 \rangle$   
 UNCERTAINTY  
 AT MSMT  
 LOCATION!



NOTE: PREVIOUS MSMT RESULT  $y$   
DOES NOT AFFECT  $C$

$\Rightarrow$  A.L. - STRATEGY  
FOR GAUSSIAN  
RAND. PROC.

IS INDEP. OF RESULTS!

# NEURAL NETWORKS

(McKay '92)

## MODEL

$$y = F_{\theta}(x) + \underbrace{\text{NOISE}}_{\text{GAUSSIAN, } \mathcal{Z}}$$

↑ "λ"

$$P(Y(x_1)=y_1, Y(x_2)=y_2, \dots | \theta) = \frac{1}{\sqrt{(2\pi\sigma^2)^M}} \exp \left[ -\frac{1}{2\sigma^2} \sum_{j=1}^M (y_j - F_{\theta}(x_j))^2 \right]$$

PRIOR?

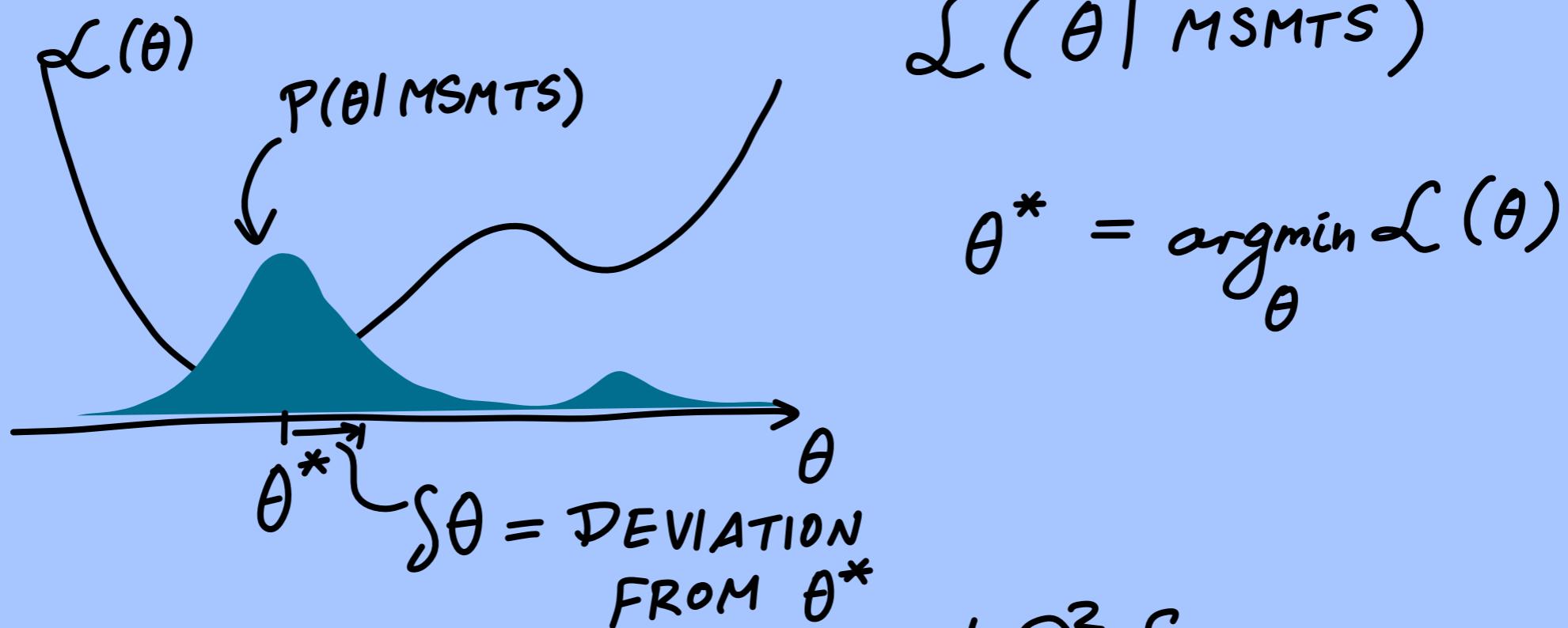
$$P_{\theta}(\theta) = \frac{1}{Z} \exp \left( -\underbrace{\mathcal{L}_0(\theta)}_{\text{REGULARIZATION LOSS (e.g. } L_1\text{)}} \right)$$

$$\mathcal{L}_{\text{MSE}}(\theta | \underbrace{\{(x_i, y_i)\}}_{\text{EXISTING SAMPLES}})$$

REGULARIZATION  
LOSS (e.g.  $L_1$ )

BAYES:

$$P(\theta | \text{MSMT}) \sim \exp \left[ - \underbrace{(\mathcal{L}_0 + \mathcal{L}_{\text{MSE}})}_{\mathcal{L}(\theta | \text{MSMTS})} \right]$$



$$\theta^* = \underset{\theta}{\operatorname{argmin}} \mathcal{L}(\theta)$$

$$\mathcal{L} = \mathcal{L}(\theta^*) + \frac{1}{2} \underbrace{\delta\theta^t \frac{\partial^2 \mathcal{L}}{\partial \theta^2} \delta\theta}_{\text{HESCIAN}} + \dots$$
$$\frac{\partial^2 \mathcal{L}}{\partial \theta_k \partial \theta_j} \xrightarrow{\text{C}^{-1}}$$

FLUCTUATIONS:

$$\langle \delta\theta_i \delta\theta_j \rangle = C_{ij}$$

NEXT MSMT  $\Rightarrow$   
NEW TERM  $\mathcal{L}^{NEW} = \mathcal{L} + \frac{1}{2\zeta^2} (y - F_\theta(x))^2$

NEW PART OF  $C^{-1}$ :

$$\begin{aligned} (C^{-1})^{NEW} &= C^{-1} + \frac{\partial^2}{\partial\theta_i \partial\theta_j} " \\ &= C^{-1} + \frac{1}{\zeta^2} \frac{\partial F_\theta(x)}{\partial\theta_i} \boxed{\frac{\partial F_\theta(x)}{\partial\theta_j}} \\ &\quad - \frac{1}{\zeta^2} (y - F_\theta(x)) \frac{\partial^2 F_\theta(x)}{\partial\theta_i \partial\theta_j} \end{aligned}$$

ASSUME: SMALL

$g_j$

## ENTROPY REDUCTION

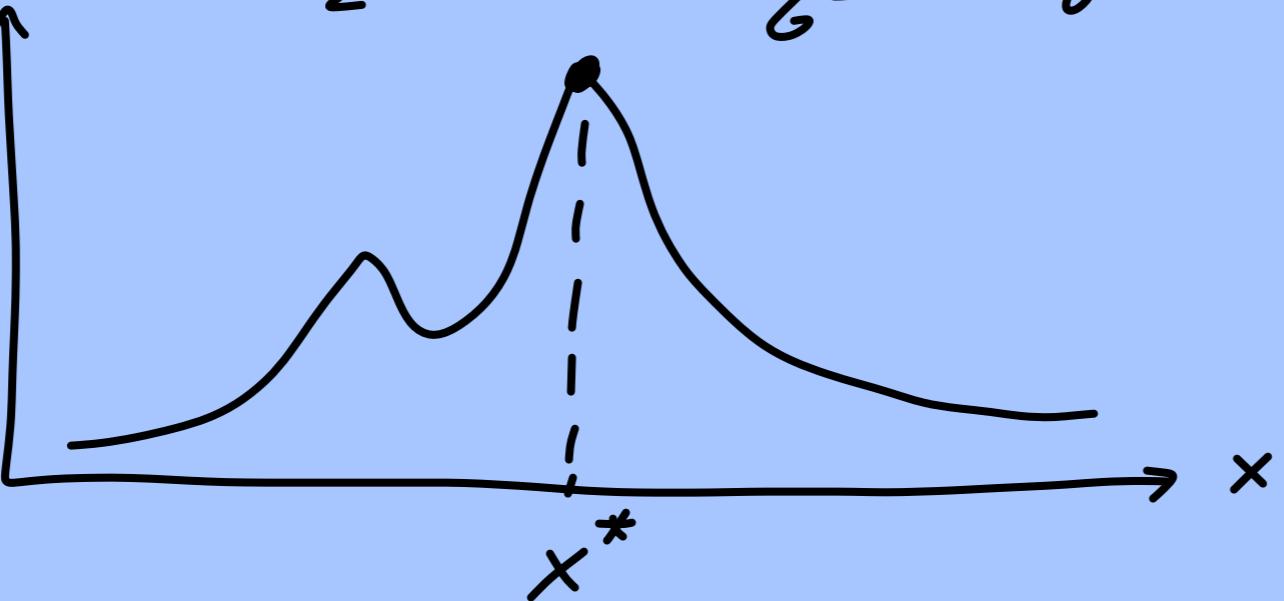
$$\begin{aligned} & \frac{1}{2} \ln \det C(C^{\text{NEW}})^{-1} \\ \approx & \frac{1}{2} \ln \left( 1 + \frac{1}{Z^2} \boxed{g^T C g} \right) \end{aligned}$$

SENSITIVITY  $\frac{\partial F_\theta}{\partial \theta}$

FLUCTUATION OF  $F_\theta(x)$  AT  $x$ :

$$\begin{aligned} \text{Var } F_\theta(x) & \approx \frac{1}{2} \left\langle \left( \underline{s_{\theta_e}} \frac{\partial F_\theta}{\partial \theta_e} \right) \left( \underline{s_{\theta_g}} \frac{\partial F_\theta}{\partial \theta_g} \right) \right\rangle \\ & = \frac{1}{2} g^T C g \end{aligned}$$

$$IG = \frac{1}{2} \ln \left( 1 + \frac{1}{G^2} \text{Var } F_\theta(x) \right)$$



$$C^{-1} = \frac{\partial^2 \mathcal{L}}{\partial \theta^2}$$

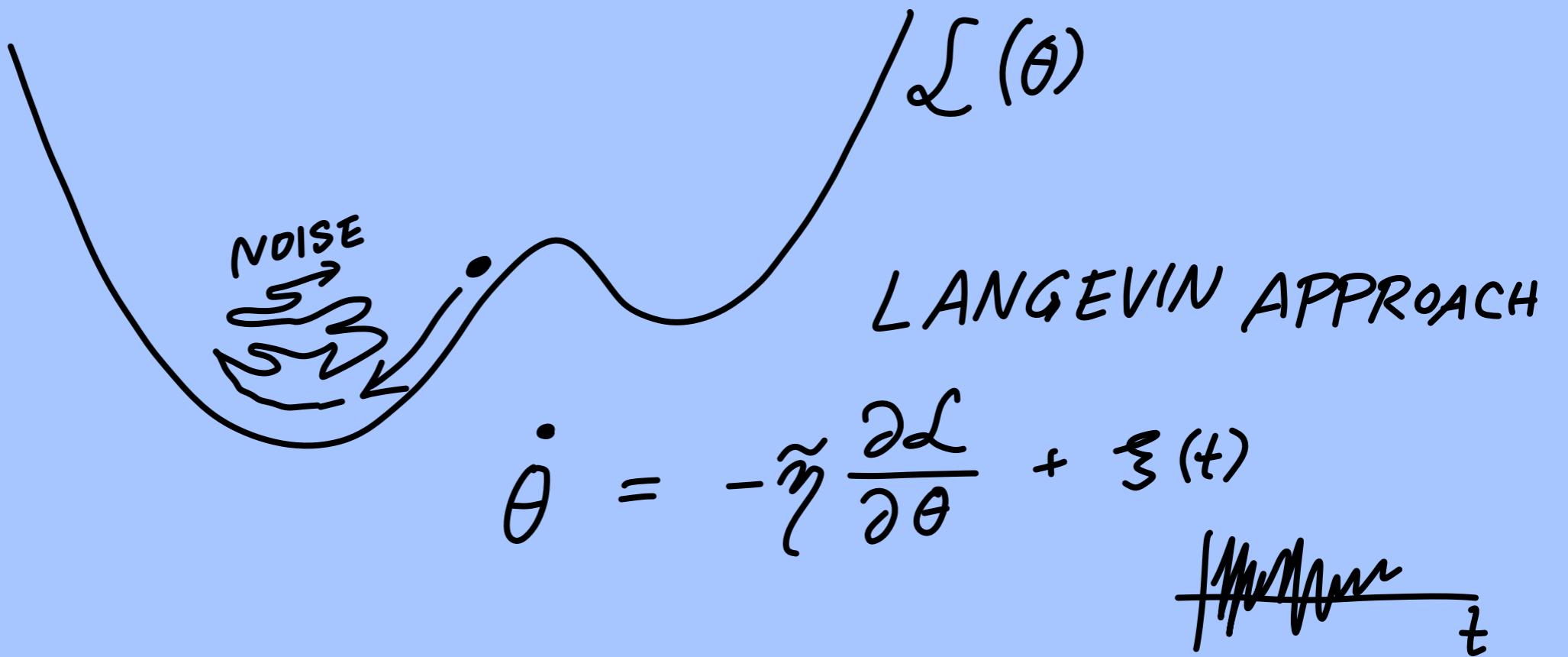
$$g = \frac{\partial F}{\partial \theta}$$

NOW:  $C$  DEPENDS ON PREVIOUS MSMTs  $\Rightarrow$  ALSO FOR  $x^* \Rightarrow$  ADAPTIVE

EXPENSIVE: NEED HESSIAN

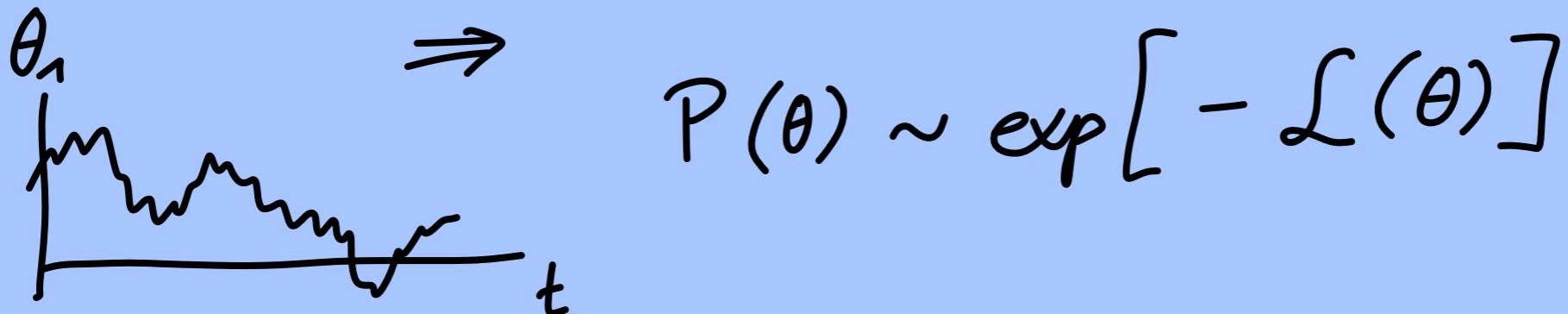
$$\frac{\partial^2 \mathcal{L}}{\partial \theta^2}$$





$$\theta^{(t+1)} = \theta^{(t)} - \gamma \frac{\partial \mathcal{L}}{\partial \theta} + \xi^{(t)}$$

$$\langle [\xi^{(t)}]^2 \rangle = 2\gamma$$

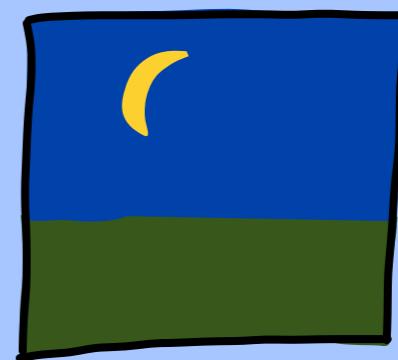
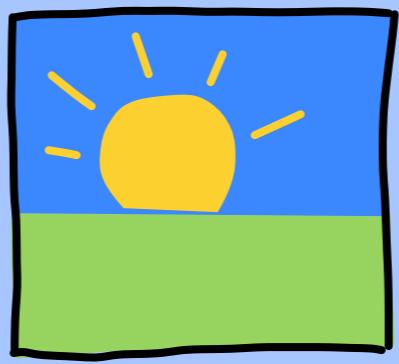
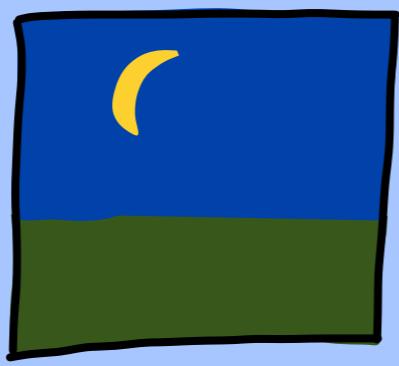
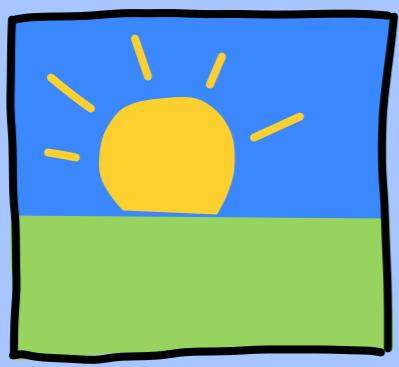


11.

# COMPLEXITY & PREDICTION & OCCAM'S RAZOR (ALGORITHMIC INFORMATION THEORY)

USE OBSERVED  
REGULARITIES FOR  
EFFICIENT DESCRIPTIONS  
AND PREDICTIONS

(RECALL SHANNON, BAYES )



?

01010101010101 01 ?

1011001 01 0101 01 ... ?  
~~~~~

1 011 0111 01111 011111 0 ... ?

1011101001101000010 ... ?  
~~~~~

"HOW DOES THIS CONTINUE?"

"IS THIS RANDOM?"

# OCCAM'S RAZOR

(ALREADY: ARISTOTLE, PTOLEMY, AQUINAS,...)

PREFER SIMPLE MODEL

SHORT ?

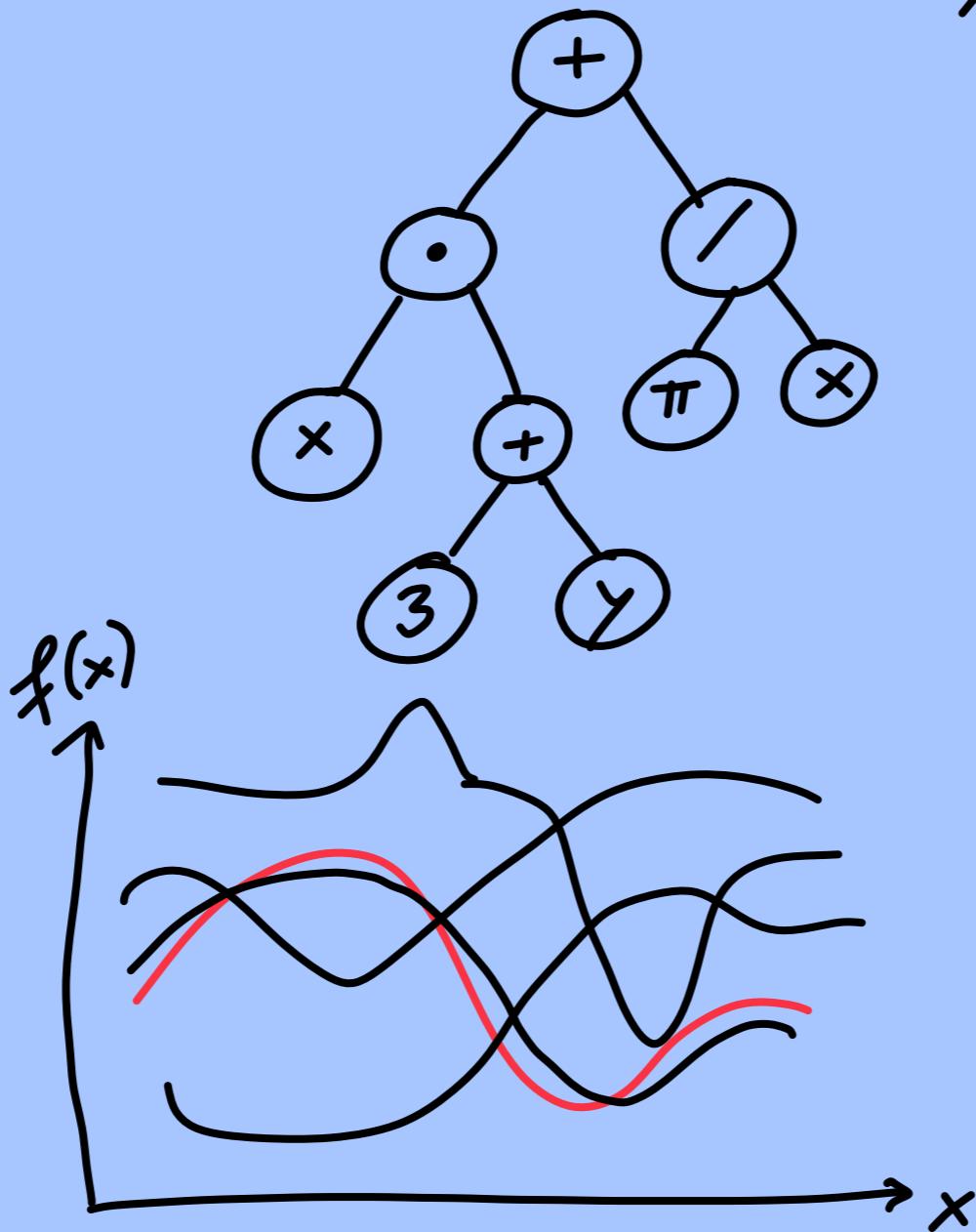
DEPENDS ON LANGUAGE

11.1

# SYMBOLIC EXPRESSIONS

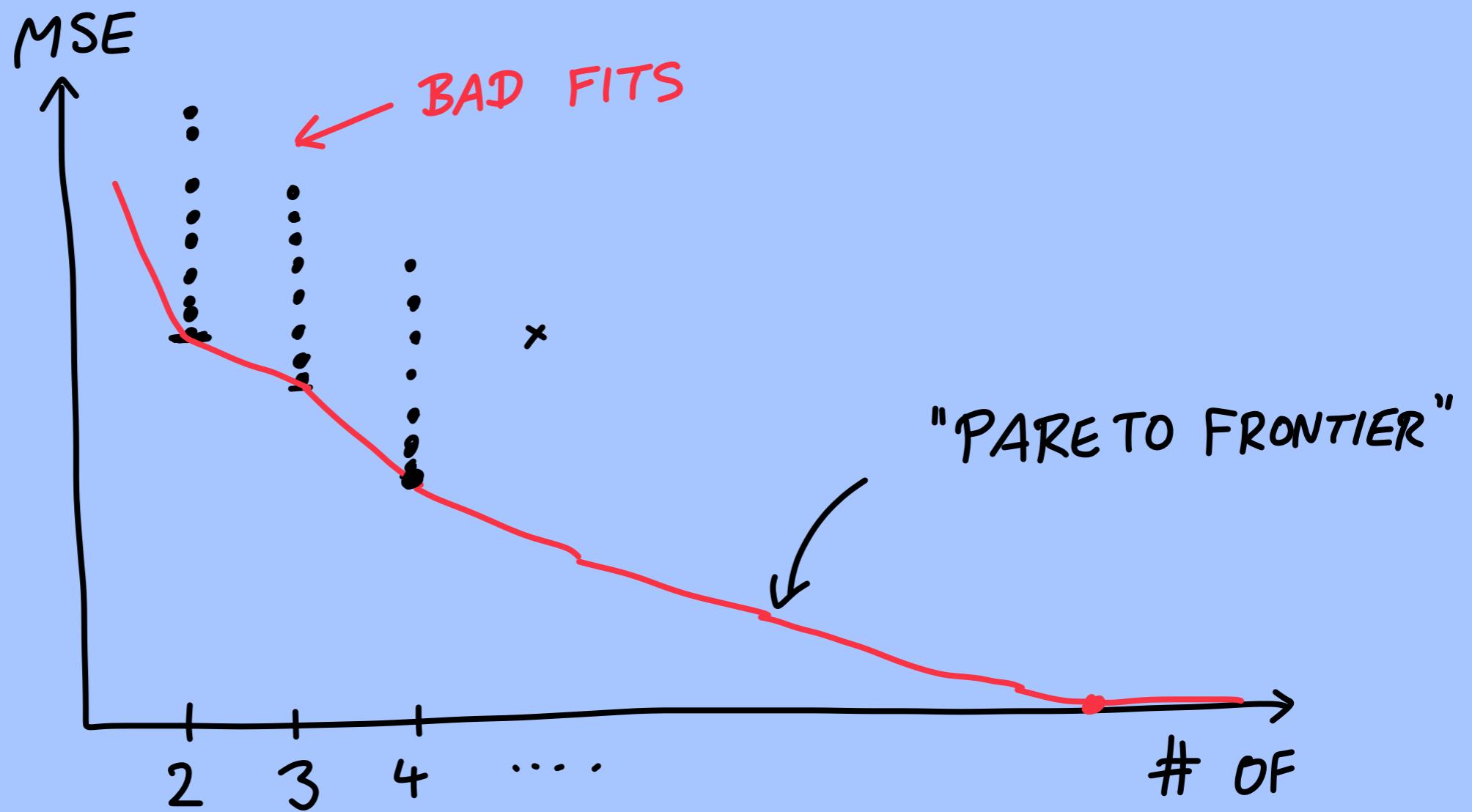
## EXPRESSION TREES

$$x \cdot (3+y) + \frac{\pi}{x}$$



TRY TO FIT  
USING SYM.  
EXPR. (THAT  
INCLUDE  $x$ )

MORE TERMS (NODES)  $\Rightarrow$  BETTER FIT

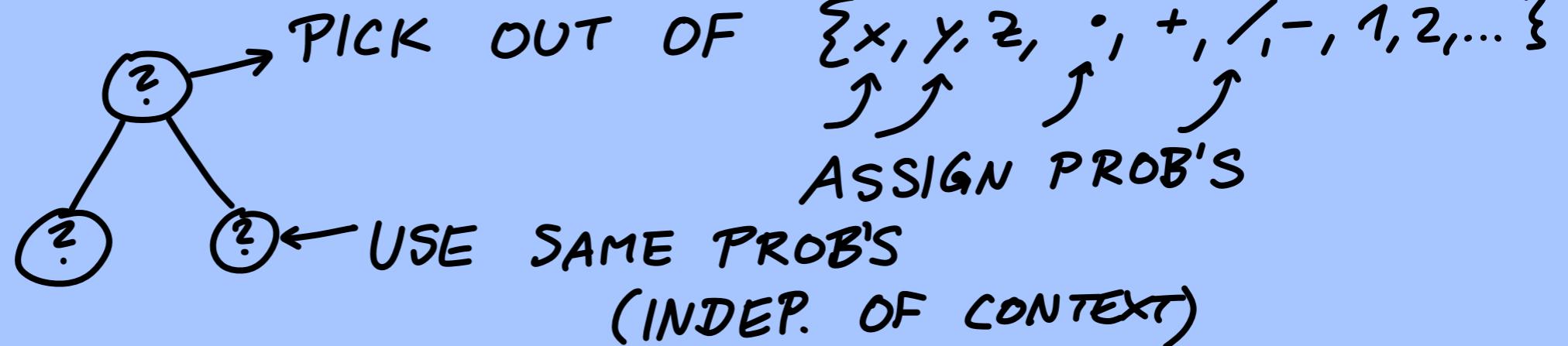


OPTIMIZING TWO THINGS  
AT THE SAME TIME :

OPTIMIZE HOLDING THE  
SECOND QUANTITY  
FIXED

(SOME  
MEASURE OF  
COMPLEXITY)

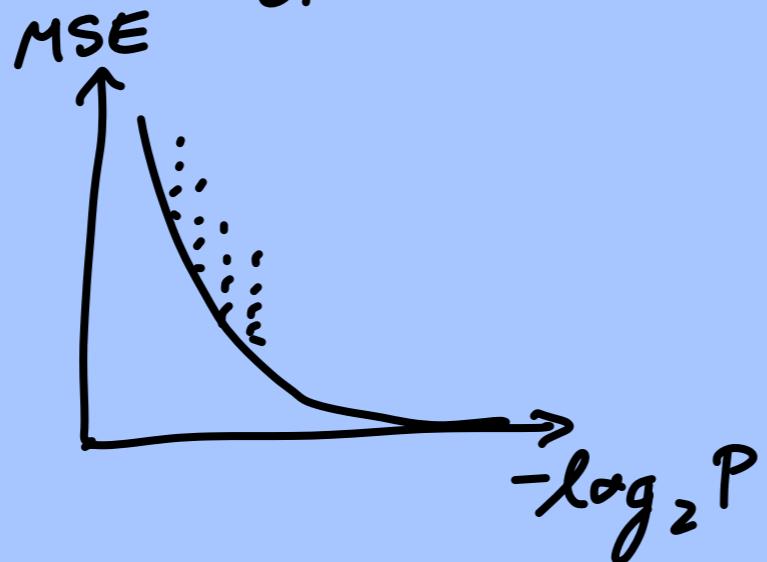
PRIOR PROBABILITY



$P(\text{EXPR.}) = \text{PRODUCT OF PROBS}$

$-\log_2 P(\text{EXPR.}) = \text{BITS NEEDED FOR OPTIMAL COMPRESSION}$

$\Rightarrow$  'MEASURE OF COMPLEXITY'

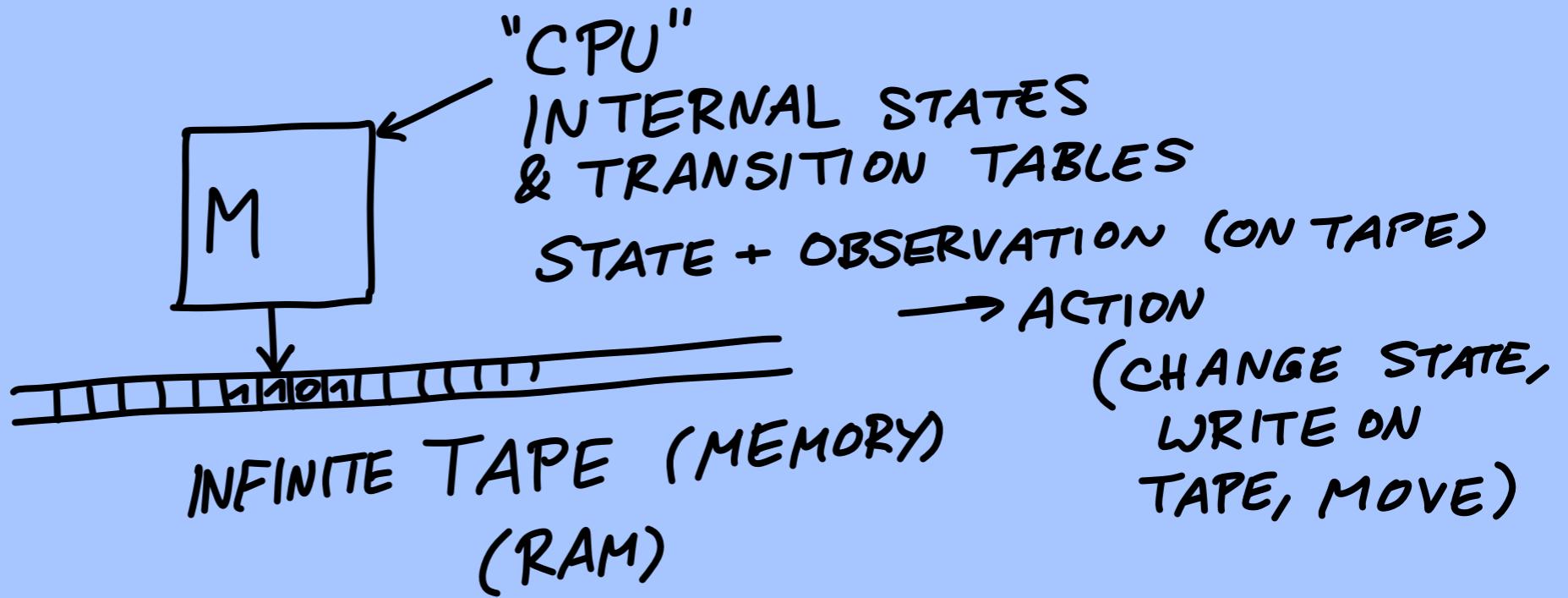


PROBLEMS:

- PRIOR SUBJECTIVE
- ARITHM. EXPRESSIONS NOT POWERFUL ENOUGH

11.2

## TURING MACHINES



### UNIVERSAL TURING MACHINE

CAN EMULATE ANY  
OTHER TURINGM. RUNNING  
ON ANY INPUT  $x$ , BY  
USING AS INPUT

$P_x$   $\Rightarrow$  FIXED FINITE  
OVERHEAD  
'EMULATION  
PROGRAM'

# HALTING PROBLEM (TURING)

GIVEN A PRG. P :

WILL IT RUN FOREVER  
OR WILL IT STOP?

CANNOT WRITE ALGORITHM  
THAT CAN DECIDE  
THIS QUESTION FOR  
ARBITRARY p!

## PREFIX-FREE CODES

$\left[ \begin{array}{c} 0 \\ 10 \\ \boxed{110} \\ 1110 \\ \dots \end{array} \right]$  CODE WORDS

$\boxed{110} \mid \boxed{1110} \mid \boxed{10}$

$\left[ \begin{array}{c} 01 \underline{11} \\ \boxed{10} \underline{01} \underline{00} \underline{01} \underline{11} \\ 01 \underline{01} 10 \underline{00} \underline{00} 11 \\ \dots \end{array} \right]$

$$\sum_{p \in \text{PR.FREE CODE}} 2^{-\overbrace{l(p)}^{\text{LENGTH OF } p \text{ IN BITS}}} \leq 1$$

KRAFT'S  
INEQUALITY

$\left[ \begin{array}{c} \boxed{10101} \underline{0} \\ 01 \underline{110} \underline{0} \\ \dots \\ 10111 \underline{1} \underline{01001} \underline{0} \\ 01100 \underline{1} \underline{01101} \underline{0} \\ \dots \end{array} \right]$

11.3

## ALGORITHMIC COMPLEXITY (KOLMOGOROFF-SOLOMONOFF-CHAITIN)

$x = 0110101101\dots \rightarrow \text{"HOW RANDOM?"}$

$$K(x) = \min_u \left\{ \ell(p) \mid U(p) = \begin{matrix} \xrightarrow{\text{OUTPUT}} \\ x \end{matrix} \right\}$$

$\downarrow$        $\downarrow$   
MACHINE      PRG  
UNIVERSAL  
TURING MACHINE

$$|K_u(x) - K_{u'}(x)| \leq \underbrace{C_{uu'}}_{\text{DEPENDS ON LENGTH OF 'EMULATOR'}}$$

$\rightarrow$  COMPRESSION!

BUT: UNCOMPUTABLE!  
HALTING PROBLEM

UPPER BOUND:

$$K(x) \leq l(x) + 2 \log_2 l(x)$$

(&  $K(x) \geq l(x)$  FOR 'MOST'  $x$ )

GOAL: FIND PRG.  $p$  THAT OUTPUTS  $x$

NAIVE: RUN ALL  $p$  OF  
INCREASING LENGTH  $\ell(p)$   
AND CHECK WHETHER  $U(p) = x$

↳ HALTING PROBLEM

MINIMAL DESCRIPTION LENGTH (MDL):

$p \sim \text{MODEL} + \text{DEVIATION}$

11.4

## UNIVERSAL SEARCH ALGORITHM (LEVIN)

GOAL: FIND PROGRAM  $P$   
THAT IS "SHORT & FAST"  
THAT OUTPUTS  $x$

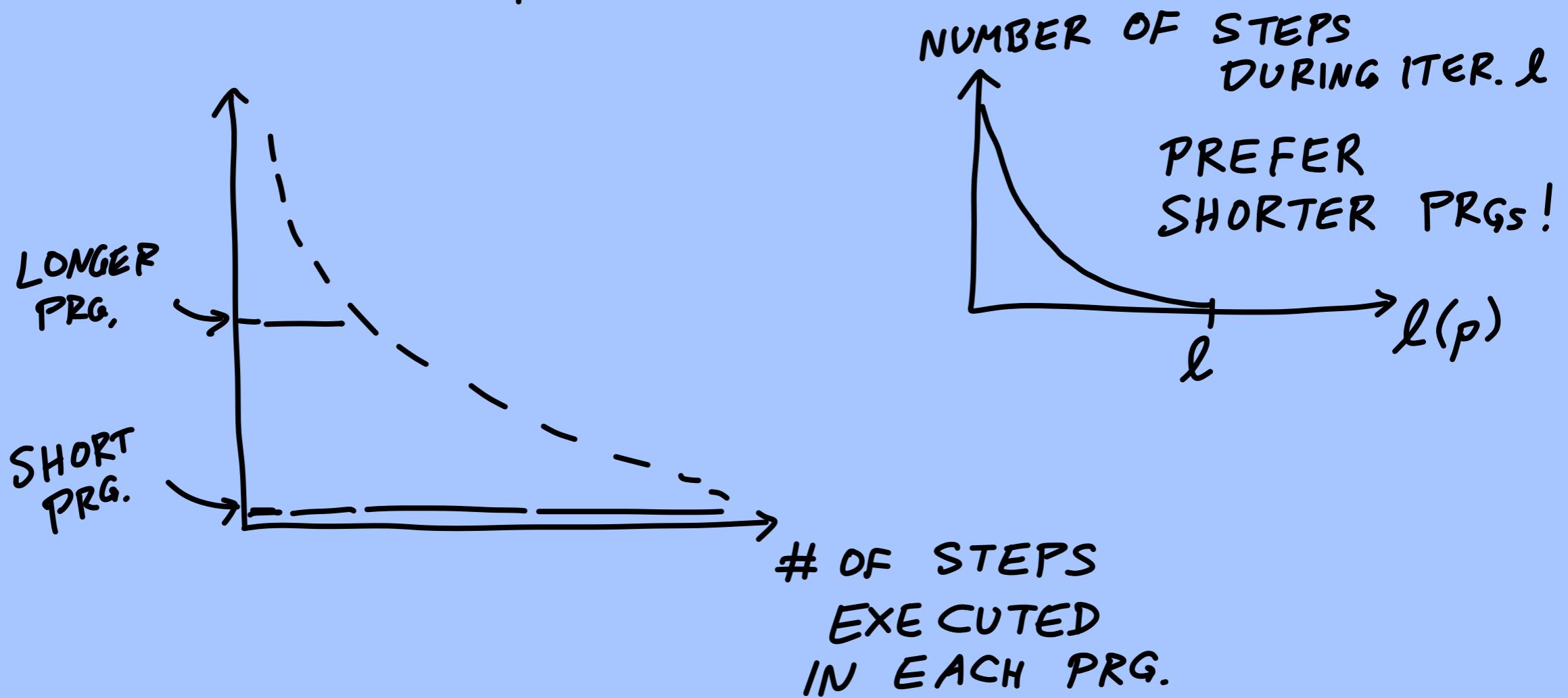
(OR: APPLY FCT.  $\phi$   
AND FIND  $P$  THAT  
PRODUCES  $x$  WITH  
 $\phi(x)=y$ )

INVERSION PROBLEM

EQ. SOLVING  
INTEGRATION  
ETC.

AT ITERATION  $\ell$ :

EXECUTE EVERY  
PRG.  $p$  WITH LENGTH  
 $l(p) \leq \ell$  FOR  $2^{\ell - l(p)}$  STEPS!



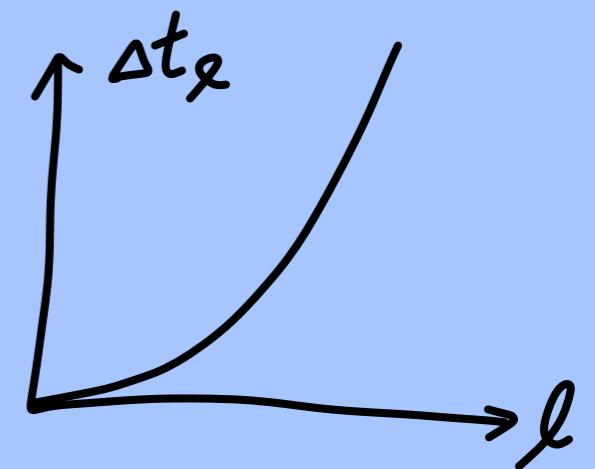
$\Rightarrow$  TOTAL STEPS IN ITERATION  $l$ :

$$\Delta t_l \leq \sum_{l(p) \leq l} 2^{l-l(p)}$$

STEPS/  
PRG.

$\leq 2^l$

↑  
P PREFIX-FREE  
PRGs



TOTAL STEPS UP TO (INCL.)  $l$ :

$$t_l = \sum_{l'=1}^l \Delta t_{l'} \leq 2^{l+1} - 2 \approx \underline{2^{l+1}}$$

(LARGER  $l$ )

$=$

TOTAL STEPS FOR PRG  $p$ :  
 (UP TO  $\ell$ )

$$\tau_{\ell, \ell(p)} = \sum_{\ell'=\ell(p)}^{\ell} 2^{\ell' - \ell(p)} = 2^{\ell - \ell(p) + 1} (-1)$$

IF  $p$  PRINTS  $\times$  AFTER  $\tau(p)$  STEPS  
 $\Rightarrow$  REAL TIME NEEDED?

$$\tau(p) = \tau_{\ell, \ell(p)} = 2^{\ell - \ell(p) + 1}$$

$$\underline{2^\ell} = \underline{2^{\ell(p)-1}} \cdot \underline{\tau(p)}$$

$$\underline{t} \leq \underline{2^{\ell+1}} = \underline{2^{\ell(p)}} \cdot \underline{\tau(p)}$$

EXPONENTIAL IN  $\ell(p)$  / LINEAR IN RUNTIME

TWO PROGS

$P$	$\ell(P)$	$\tau(P)$
$P'$	$\ell(P')$	$\tau(P')$

WHICH IS COMPLETED FIRST?

$$\underbrace{2^{\ell(P)} \tau(P)} < 2^{\ell(P')} \tau(P')$$

$\Rightarrow$  WILL PICK  $P$  THAT MINIMIZES

$$\underline{\ell(P) + \log_2 \tau(P)}$$

$\Rightarrow$  LEVIN COMPLEXITY OF  $x$ :

$$K_U(x) = \min_P \{ \ell(P) + \log_2 \tau(P) \mid U(P) = x \}$$

CAN BE COMPUTED!

SLIGHT GENERALIZATION :

$P(p)$       PRIOR  
↓  
PRG.

⇒ OPTIMAL COMPRESSION

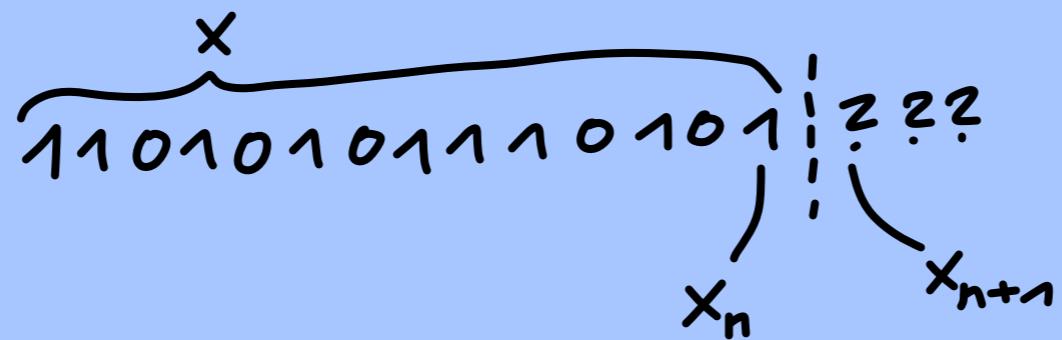
$$P_{\text{COMPR.}} \mapsto P' : l(P') \approx -\log_2 P(p)$$

$$\text{MINIMIZE } 2^{l(P')} \tau(P')$$

$$\Rightarrow \text{MINIMIZE } \frac{\tau(p)}{P(p)}$$

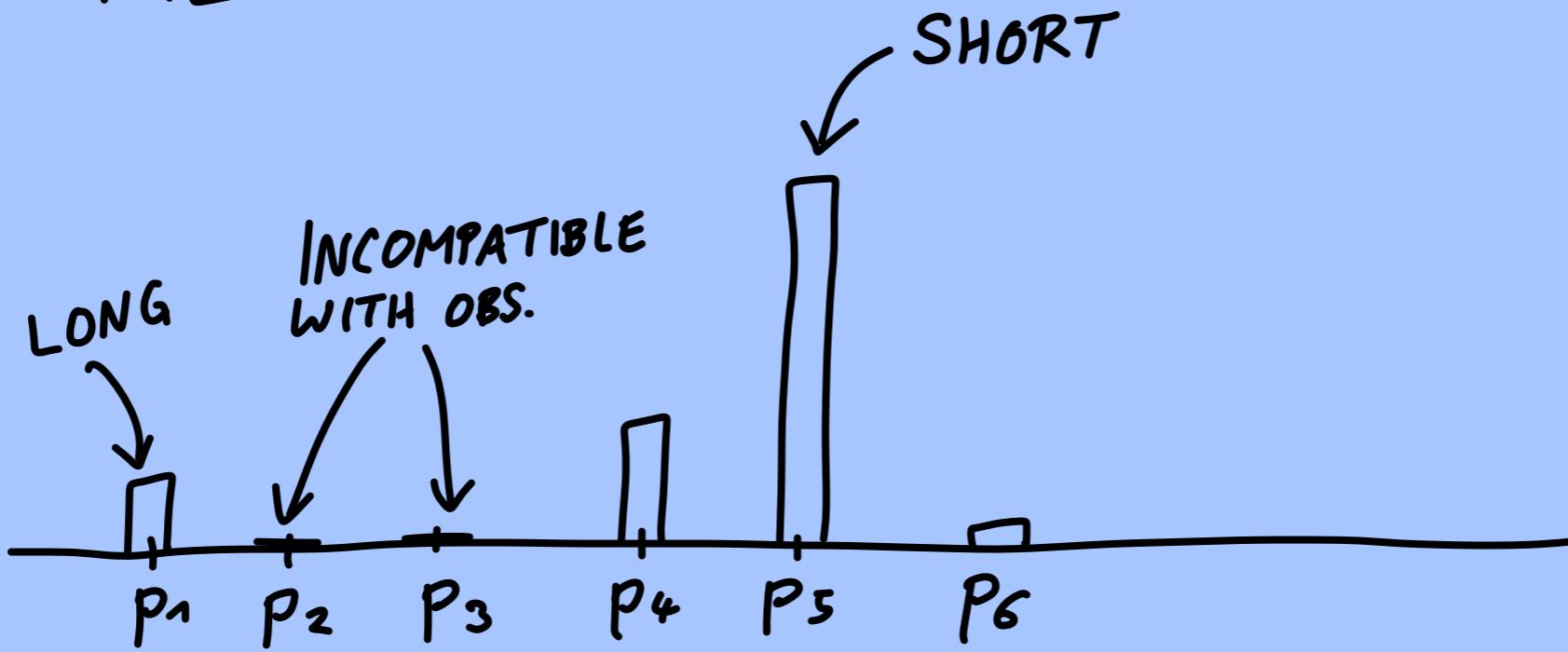
11.5

## SOLOMONOFF'S GENERAL THEORY OF INDUCTION (ALGORITHMIC PROBABILITY)



OBSERVE → HYPOTHESES → PREDICT  
= PROGRAMS  
(MANY!)  
KEEP ALL(!)  
BUT PREFER SHORT  
PRG

$\Rightarrow$  FILTER



$\rightsquigarrow$  UPDATE WHEN OBS.  $x_{n+n}$

01 01 01 01 | 1

BAYES:

$$P(x_{n+1} | \underbrace{x_1, \dots, x_n}_x) = \sum_p P(x_{n+1} | p) \underbrace{P(p | x)}_{P(p|x)}$$

$$P(p|x) = \frac{P(x|p) P(p)}{P(x)}$$

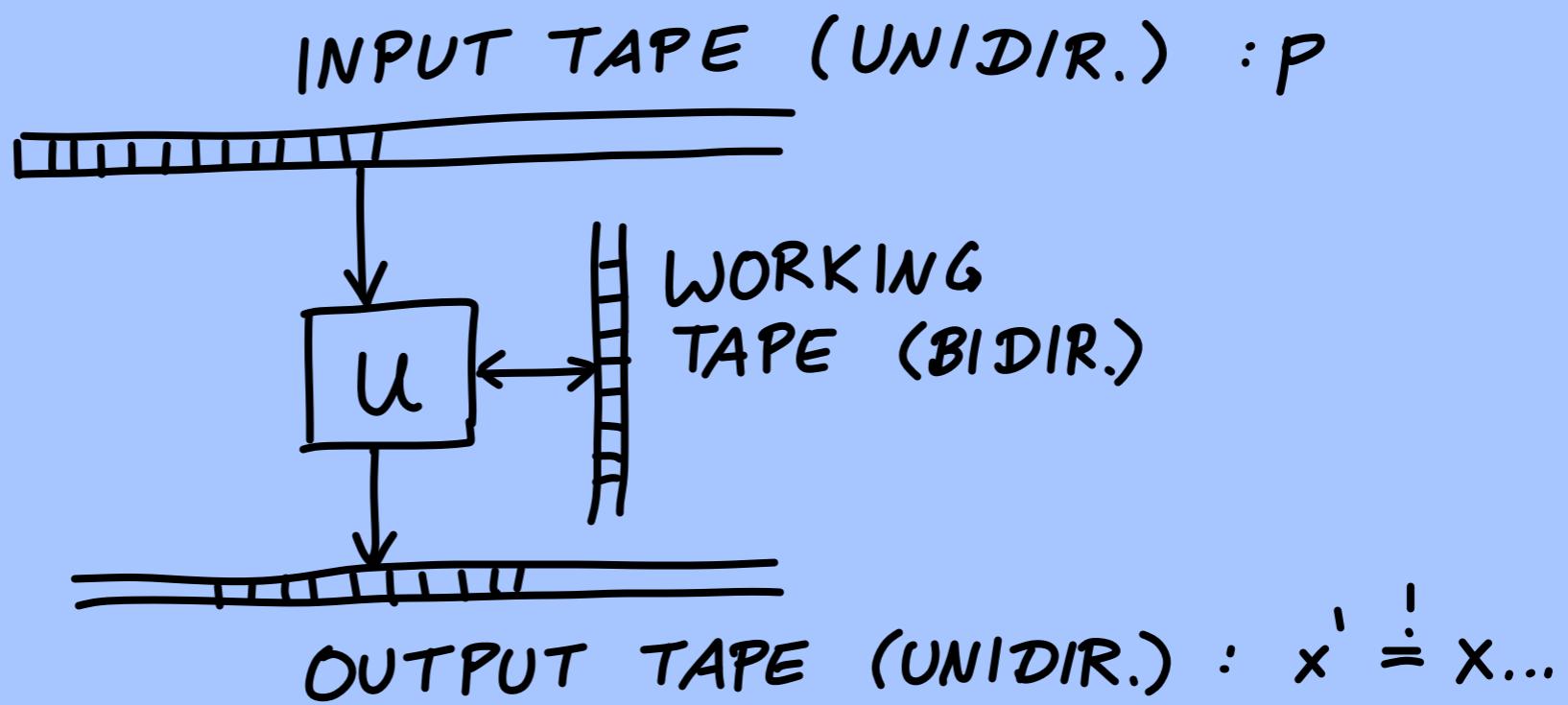
PRIOR?

ROUGHLY:  $P(p) \sim 2^{-l(p)}$

IN WORDS: SOLOMONOFF "ALGORITHMIC PROBAB."

$P(x)$  = PROB. OF  $x$  BEING  
OUTPUT BY A TURING  
MACHINE THAT IS FED  
A RANDOM PRG.  $p$ !

## SPECIAL TURING MACHINE:



$p$  = "MINIMAL CODE FOR  $x$ " ( $p \in \mathcal{M}(x, U)$ )  
IF  $U(p) = x\dots$   
& YOU CANNOT DROP LAST BIT OF  $p$

$$P_u(x) = \frac{\sum_{p \in M(x, u)} 2^{-\ell(p)}}{\sum_{x'} \sum_{p \in M(x', u)} 2^{-\ell(p)}$$

$\ell(x') = \ell(x)$

"ALGORITHMIC  
PROBABILITY"

OBSERVE:

$$\sum_{x'} \sum_{p \in M(x, u)} 2^{-\ell(p)} \leq 1$$

$\ell(x') = \ell$

↓  
KRAFT

BUT: HALTING PROBLEM!

~ MODIFY

MACHINE  $M_T$  STOPS AT CUTOFF TIME  $T$

$$\tilde{P}_u(x) = \lim_{T \rightarrow \infty} \sum_{p \in M(x, M_T)} 2^{-\ell(p)}$$

INDUCTION:

$$P_u(x_{n+1} | x_1, \dots, x_n) = \frac{\tilde{P}_u(x_1, \dots, x_n, x_{n+1})}{\tilde{P}_u(x_1, \dots, x_n, 0) + \tilde{P}_u(x_1, \dots, x_n, 1)}$$

SOLOMONOFF:

RANDOM PROCESS

$$P(x_{n+1} | x_1, \dots, x_n)$$

$$E_P \left[ \sum_{i=1}^n (P(x_{i+1}=1 | x_1, \dots, x_i) - P_u(x_{i+1}=1 | x_1, \dots))^2 \right] \leq \frac{k}{2} \ln 2$$

(n → ∞)

$P_u$  CONVERGES TO CORRECT  $P$ !

FIXED  
DEPENDS  
ON  $U$  &  $P$

$P_1$  — ✓  
 $P_2$  — — ✓  
 $P_3$  —  
 $P_4$  ✗

## NOTES:

- CAN USE LEVIN SEARCH
- COMPRESS PRGS DISCOVERED SO FAR

$p = A \underbrace{BC}_{BA}, CDE \underbrace{BA} \dots \Rightarrow$  SEARCH  
TIME  $\sim \frac{\tau(p)}{P(p)}$   
NEW SYMBOL DECREASES

- DESIGN SMART TRAINING SEQUENCES

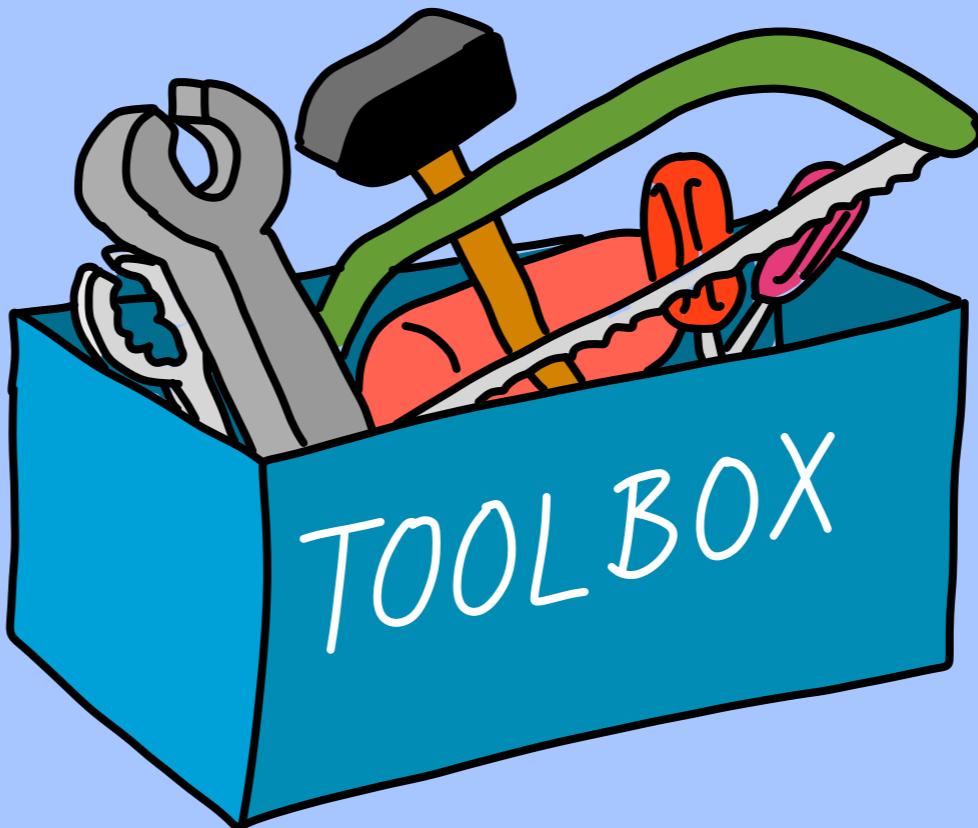
## RELATED:

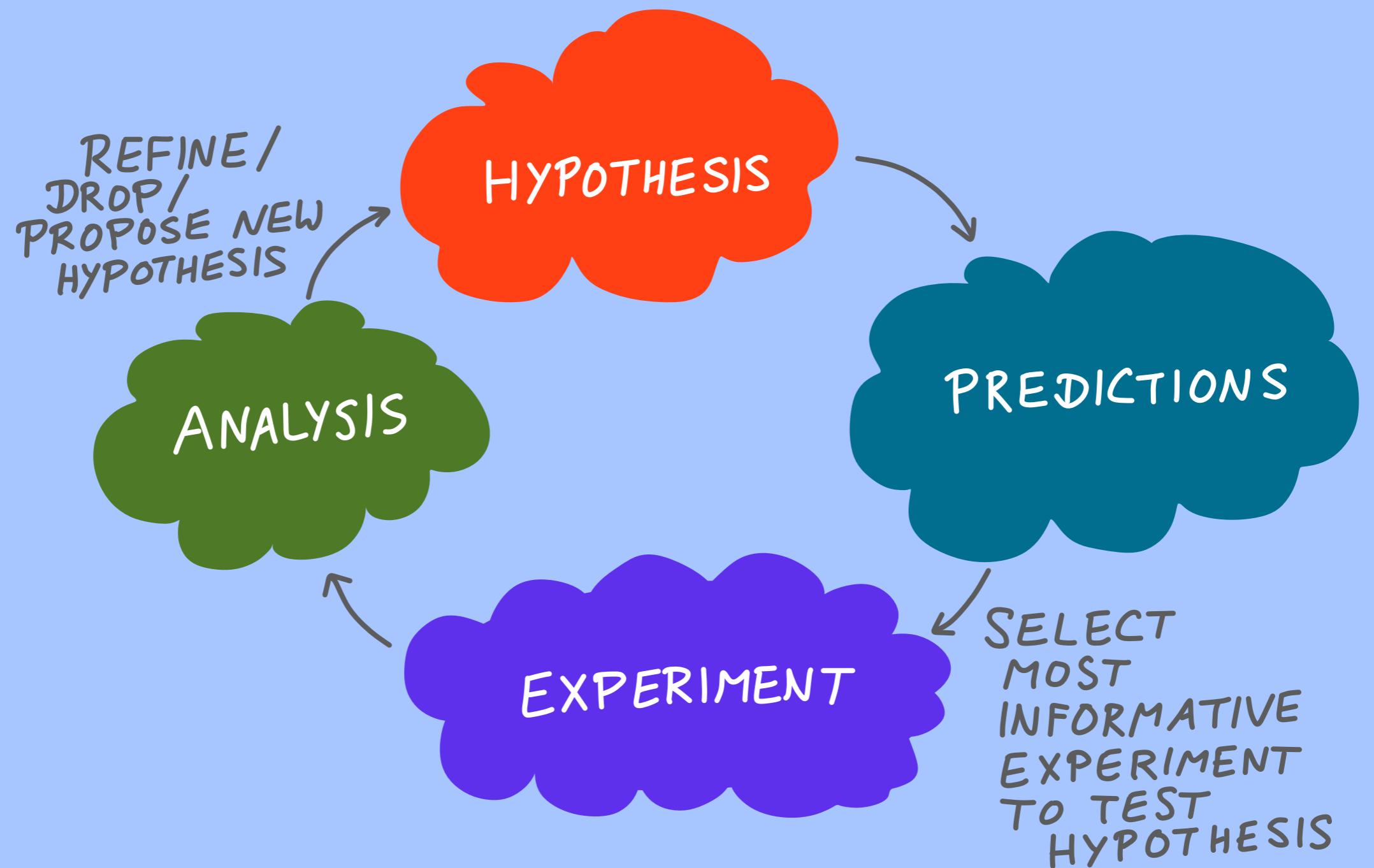
- OOPS (SCHMIDHUBER)
- AIXI (HUTTER)
- PROGRAM SYNTHESIS

12.

## CONCLUDING OVERVIEW & OUTLOOK

ADVANCED MACHINE LEARNING FOR  
PHYSICS, SCIENCE, AND  
ARTIFICIAL SCIENTIFIC DISCOVERY





SUPERVISED  
LEARNING  
INCL. GRAPH NN,  
TRANSFORMERS

DEEP IMPLICIT  
LAYERS  
NEURAL DIFF. EQS.

LEARNING PROBABILITIES  
INCL. BOLTZMANN M.,  
NORMALIZING FLOWS,  
VAE, GAN

REPRESENTATION  
LEARNING  
AUODECODERS

MUTUAL  
INFORMATION

HYPOTHESIS

BAYES

ALGORITHMIC  
INFORMATION  
THEORY

ANALYSIS

PREDICTIONS

EXPERIMENT

REINFORCEMENT  
LEARNING

ACTIVE LEARNING  
OPTIMAL EXPERIM.  
DESIGN

# SOME INTERESTING APPROACHES / ASPECTS WE DID NOT COVER → EXPLORE!

DECISION TREES  
KERNEL MACHINES

UNSUPERVISED  
CLUSTERING TECHNIQUES

STATISTICAL LEARNING  
THEORY

INSPECTING NN  
(INTERPRETABLE AI)

CAUSAL INFERENCE

MORE SYMBOLIC/LOGICAL  
APPROACHES

LOGIC PROGRAMMING LANGUAGES

AUTOMATED THEOREM  
PROVERS

PROGRAM SYNTHESIS

NEW PHYSICAL HARDWARE  
FOR MACHINE LEARNING  
CLASSICAL / QUANTUM

# OUTLOOK / CHALLENGES FOR "ARTIFICIAL SCIENTIFIC DISCOVERY"

IDENTIFY SUITABLE  
SCIENTIFIC TOPICS

COMPLEXITY  
(GENETICS, PROTEIN  
FOLDING, MATERIALS,  
KNOT THEORY,  
QUANTUM EXPERIMENTS &  
CONTROL,...)

?/  
FOUNDATIONAL/BASIC  
QUESTIONS?

HOW TO ENCODE  
EXISTING KNOWLEDGE?  
(UNSTRUCTURED & UNRELIABLE)

NEEDED TO IDENTIFY &  
PRESENT ADVANCES

HUMAN / MACHINE TEAM

WHAT IS CONSIDERED  
SCIENTIFICALLY INTERESTING?  
(RL GOAL)

HOW CAN THE MACHINE  
EXPLAIN/TEACH WHAT IT  
DISCOVERED?

