A physicist's perspective on generative models

Lei Wang (王磊)
Institute of Physics, CAS
https://wangleiphy.github.io

Plan

1. **Physics perspective**

- Autoregressive models
- Flow models

2. **Physics applications**

- Crystalline materials design
- Variational free-energy calculation

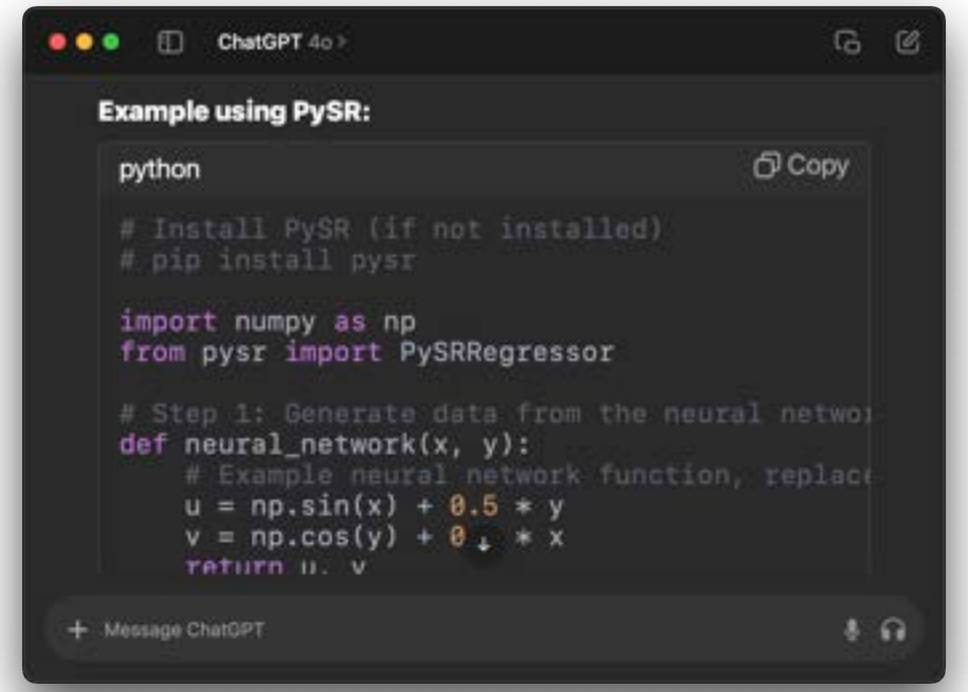
Probabilistic modeling with generative AI

$$p(\boldsymbol{X})$$

pixels, words, atoms, ...

How to express, learn, and sample from a high-dimensional probability distribution ?

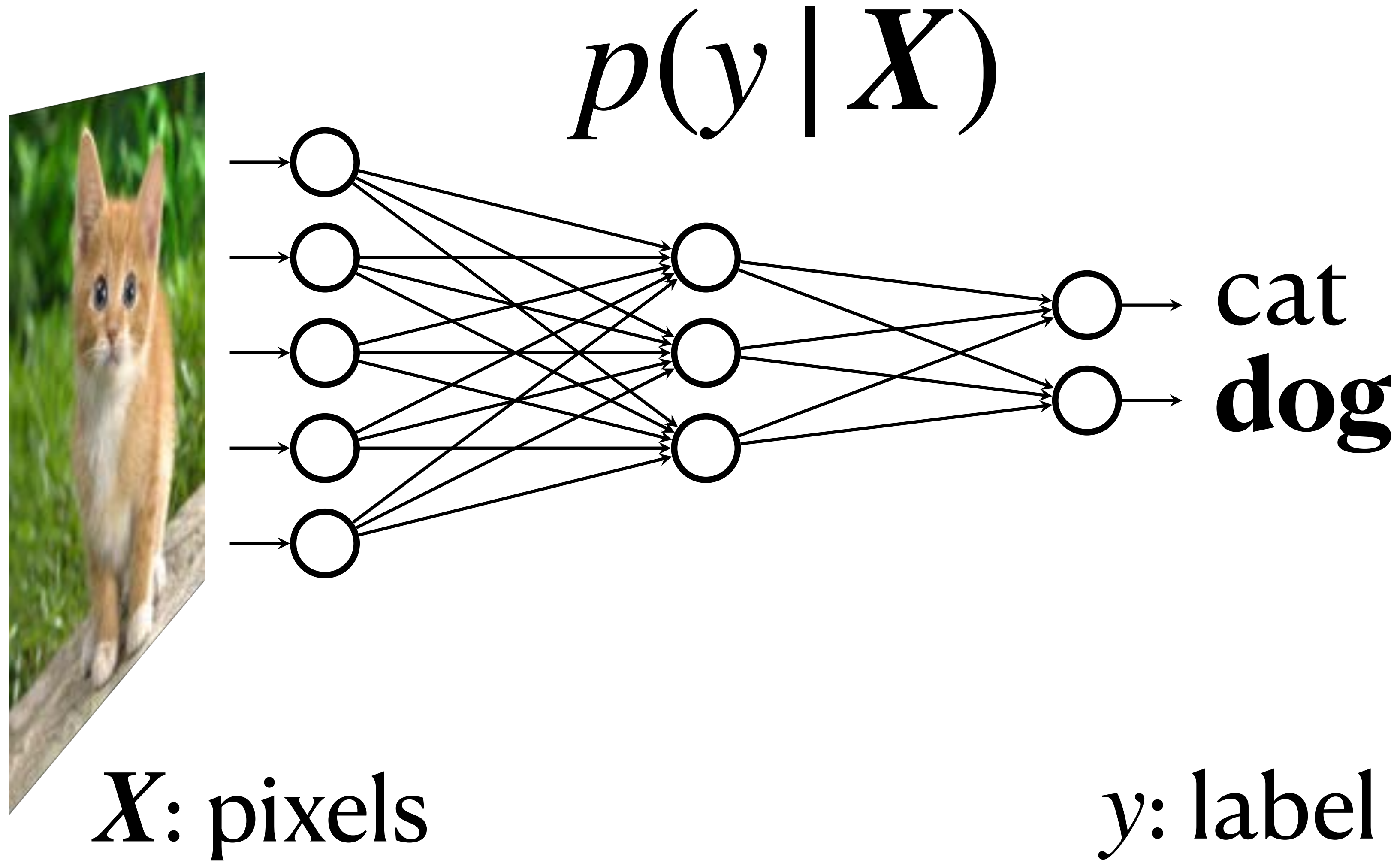
DaLL-E



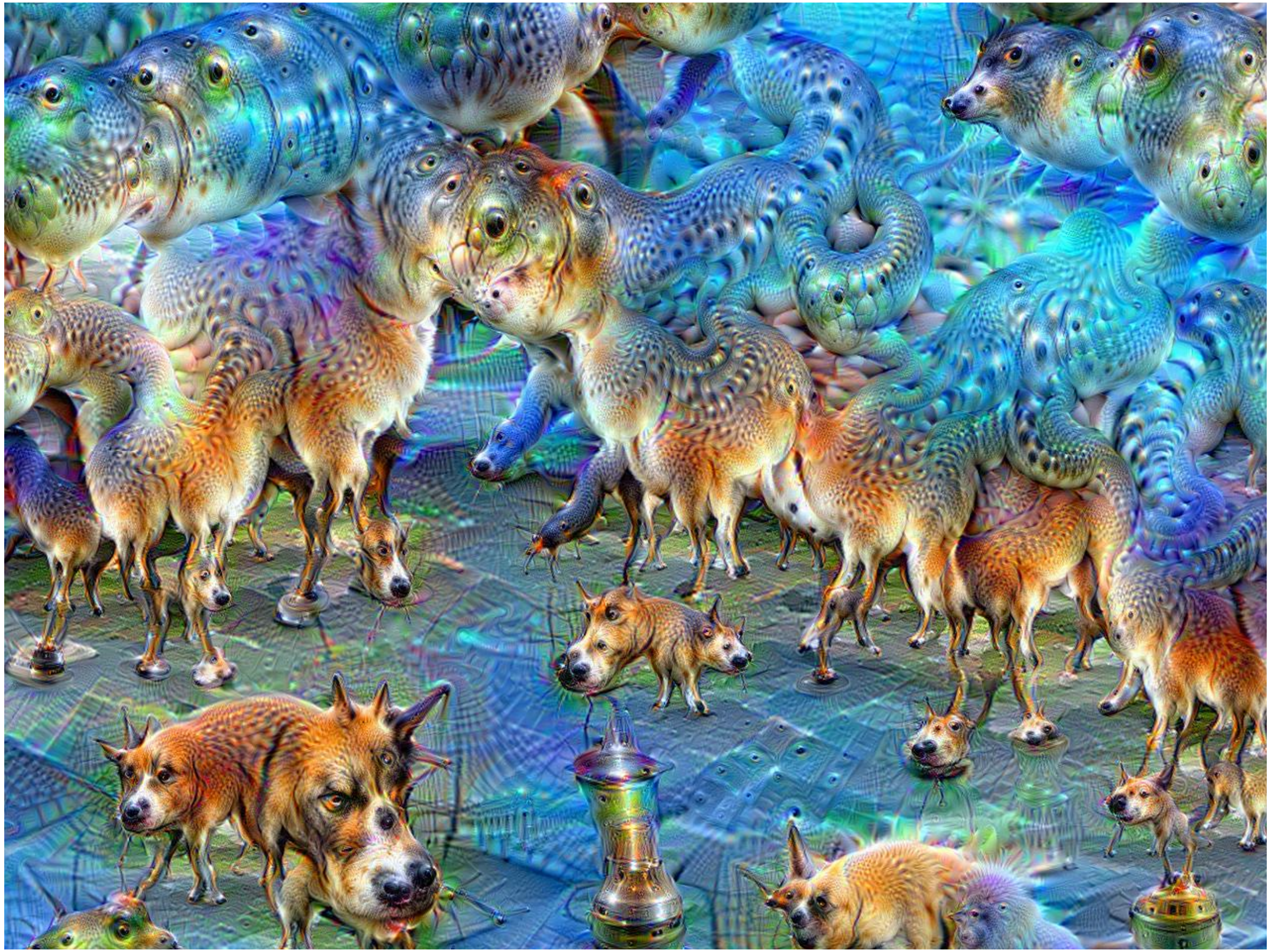
ChatGPT

AlphaFold3

Discriminative AI is not enough



$$\nabla_{\text{pixels}} p(\text{dog} \mid \text{pixels})$$



Deepdream: http://googleresearch.blogspot.ch/2015/06/inceptionism-going-deeper-into-neural.html

Bayes rule

posterior prior likelihood

$$p(\boldsymbol{X} \mid y) \propto p(\boldsymbol{X})p(y \mid \boldsymbol{X})$$

Inverse design Forward prediction


generative model $p(\boldsymbol{X})$
data $\boldsymbol{X}$

Two sides of the same coin

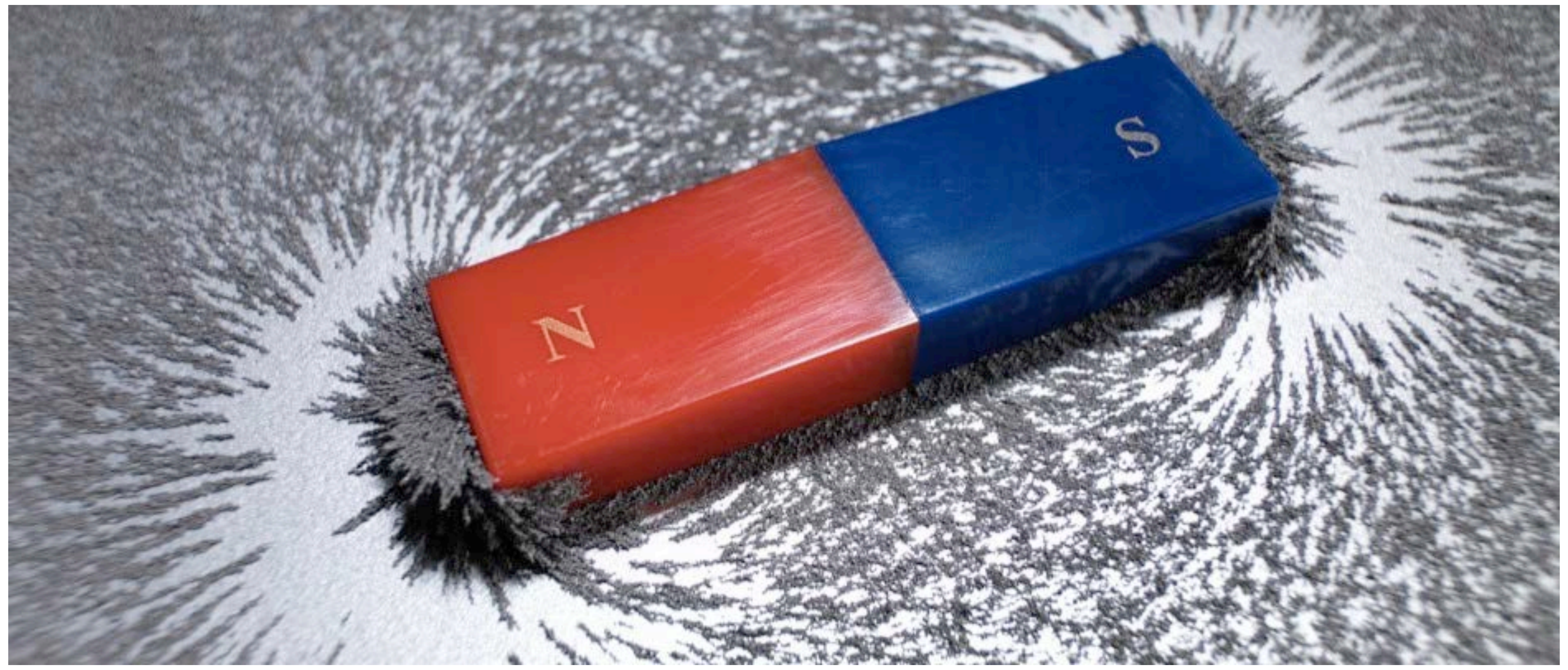
Generative modeling

"learn from data"

Mamixmum likelihood estimation

$$\mathcal{L} = -\mathbb{E}_{\boldsymbol{X}\sim\mathrm{data}}\left[\ln p(\boldsymbol{X})\right]$$

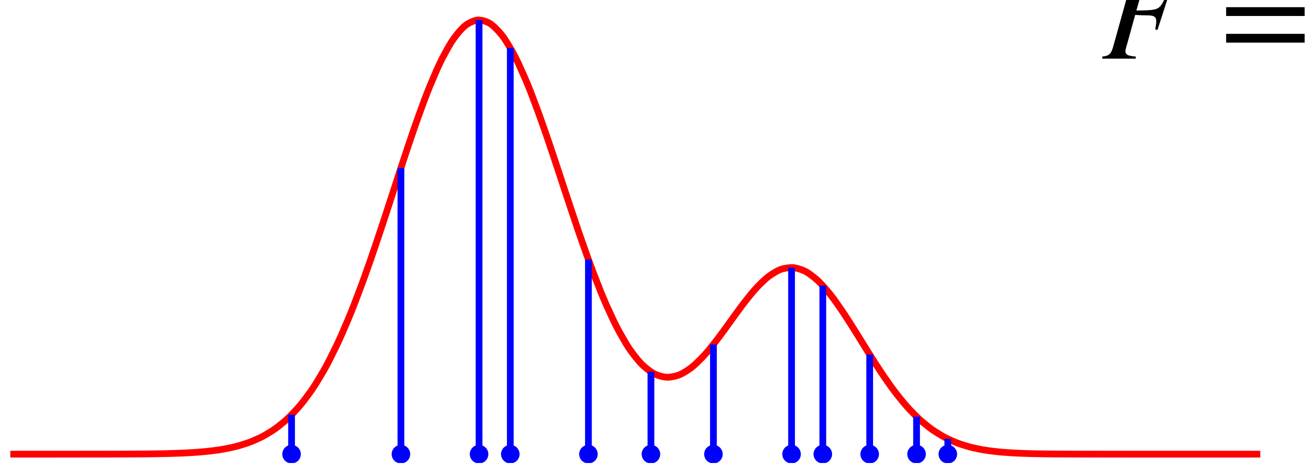
Statistical physics



"learn from energy"

Variational free energy

$$F = \mathop{\mathbb{E}}_{\boldsymbol{X}\sim p(\boldsymbol{X})}\left[E(\boldsymbol{X}) + k_B T \ln p(\boldsymbol{X})\right]$$



$\mathbb{KL}(\mathrm{data} \,\|\, p)$ vs $\mathbb{KL}(p \,\|\, e^{-E/k_BT})$

Kullback–Leibler divergence

$$\mathbb{KL}(\pi \parallel p) \equiv \sum_{\boldsymbol{X}} \pi(\boldsymbol{X}) \left[\ln \pi(\boldsymbol{X}) - \ln p(\boldsymbol{X}) \right]$$

$$\mathbb{KL}(\pi \parallel p) \geq 0$$

$$\mathbb{KL}(\pi \parallel p) = 0 \iff \pi(\boldsymbol{X}) = p(\boldsymbol{X})$$

$$\mathbb{KL}(\pi \parallel p) \neq \mathbb{KL}(p \parallel \pi)$$

Learn from data

$$\pi(\boldsymbol{X}) \propto \sum_{\boldsymbol{d} \in \text{dataset}} \delta(\boldsymbol{X} - \boldsymbol{d})$$

$$\min_{\theta} \mathbb{KL}(\underset{\text{target}}{\pi} \parallel \underset{\text{model}}{p_\theta}) \iff \underset{\text{Maximum likelihood estimation}}{\min_{\theta} \left\{ \mathbb{E}_{\boldsymbol{X} \sim \text{dataset}} \left[-\ln p_\theta(\boldsymbol{X}) \right] \right\}}$$

The lower bound is the entropy of the dataset: complete memorization

Learn from energy

$$\pi(\boldsymbol{X}) \propto e^{-E/k_BT}$$

$$\min_{\theta} \mathbb{KL}(p_\theta \parallel \pi) \iff \min_{\theta} \left\{ \mathop{\mathbb{E}}_{\boldsymbol{X} \sim p_\theta(\boldsymbol{X})} \left[E(\boldsymbol{X}) + k_BT \ln p_\theta(\boldsymbol{X}) \right] \right\}$$

model target

Variational free energy

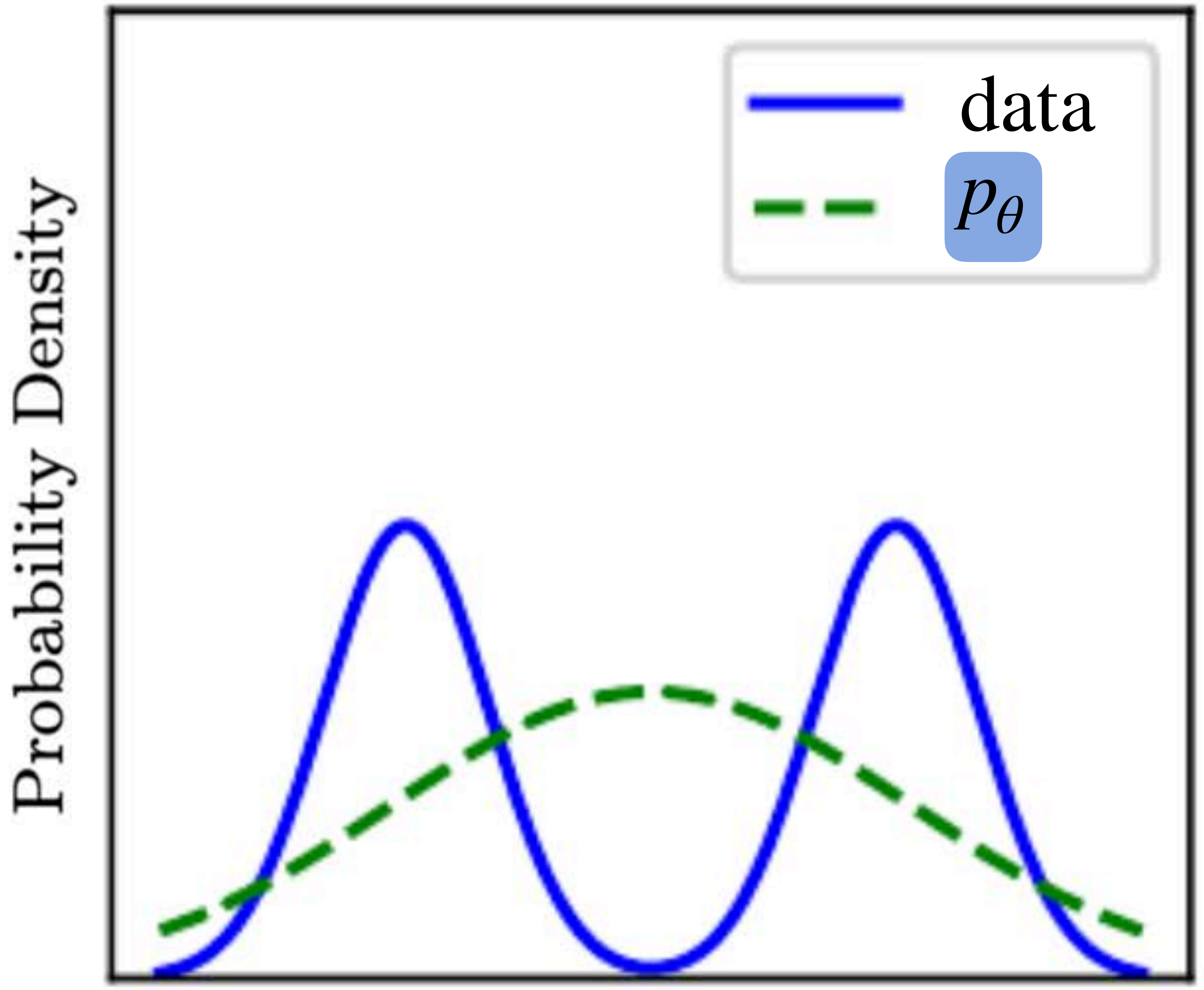
The lower bound is the true free energy: exact solution

Forward KL or Reverse KL ?

Maximum likelihood estimation

$$\min_\theta \mathbb{KL}(\text{data} \,||\, p_\theta)$$

Mode covering

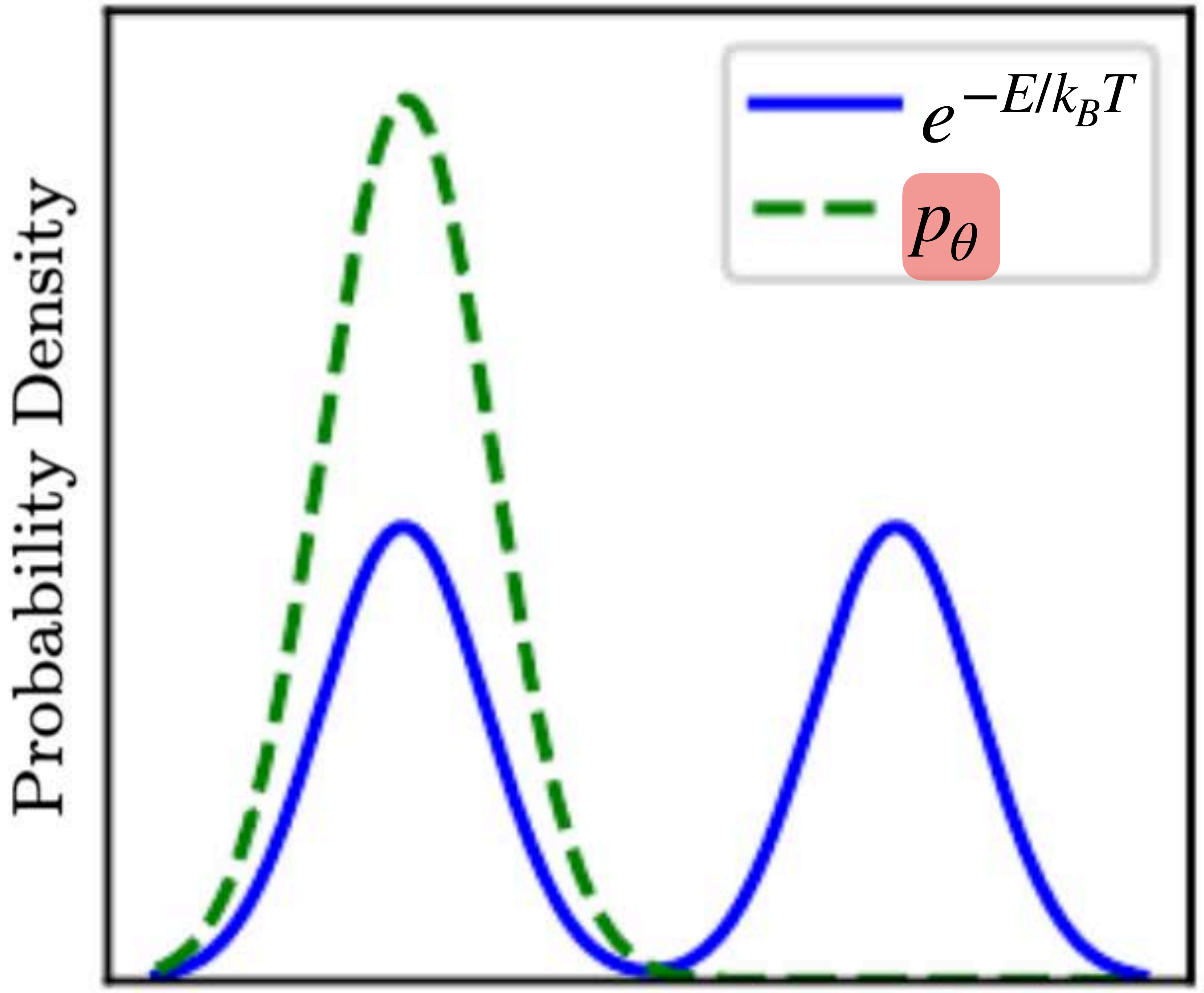


Failure mode: hallucination

Variational free energy

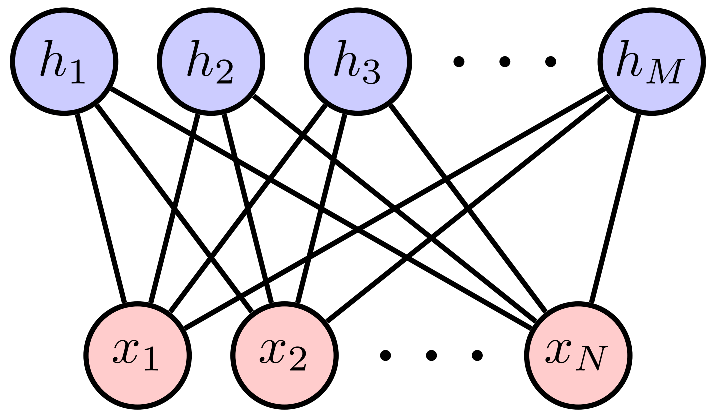
$$\min_\theta \mathbb{KL}(p_\theta \,||\, e^{-E/k_BT})$$

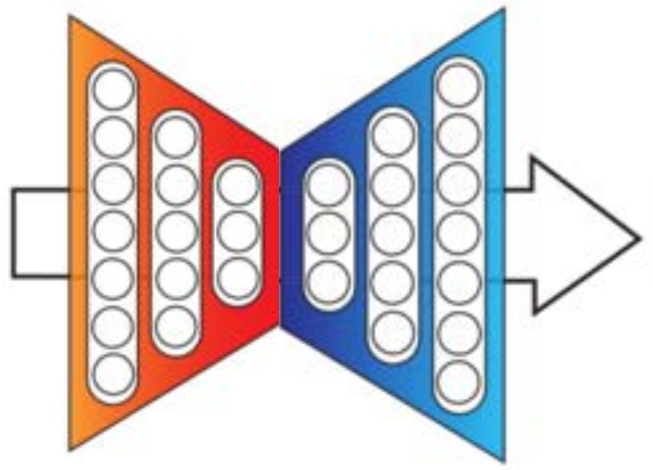
Mode seeking

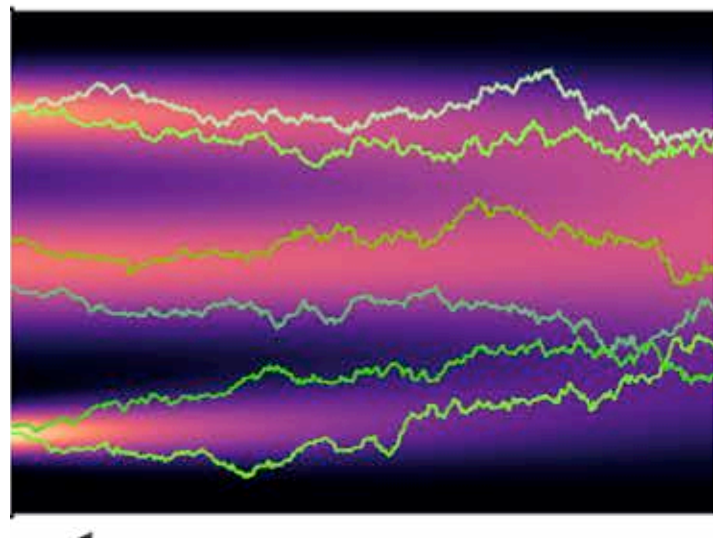


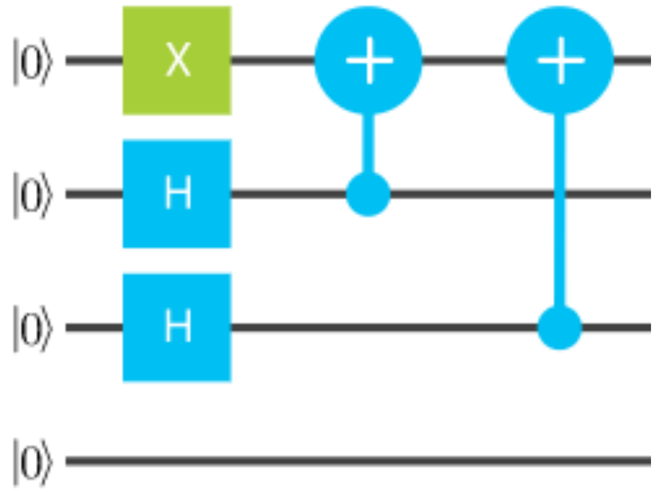
Failure mode: local minima

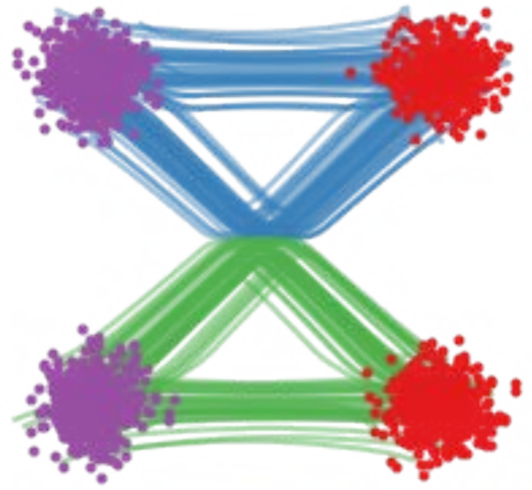
Goodfellow et al, Deep Learning











Boltzmann Machine — 1985

Variational Autoencoder — 2013

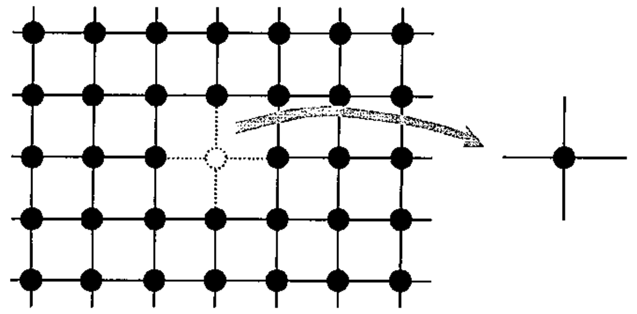
Diffusion Model — 2015

Born Machine — 2017

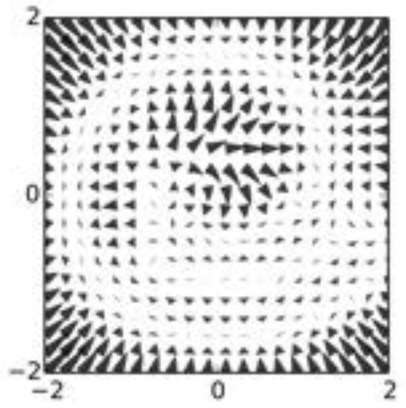
Flow Matching — 2022

Monte Carlo Ising model

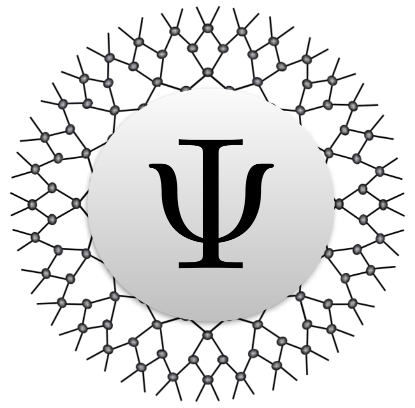
Variational mean field



Nonequilibrium thermodynamics



Tensor networks Quantum circuits



Fluid optimal transportation

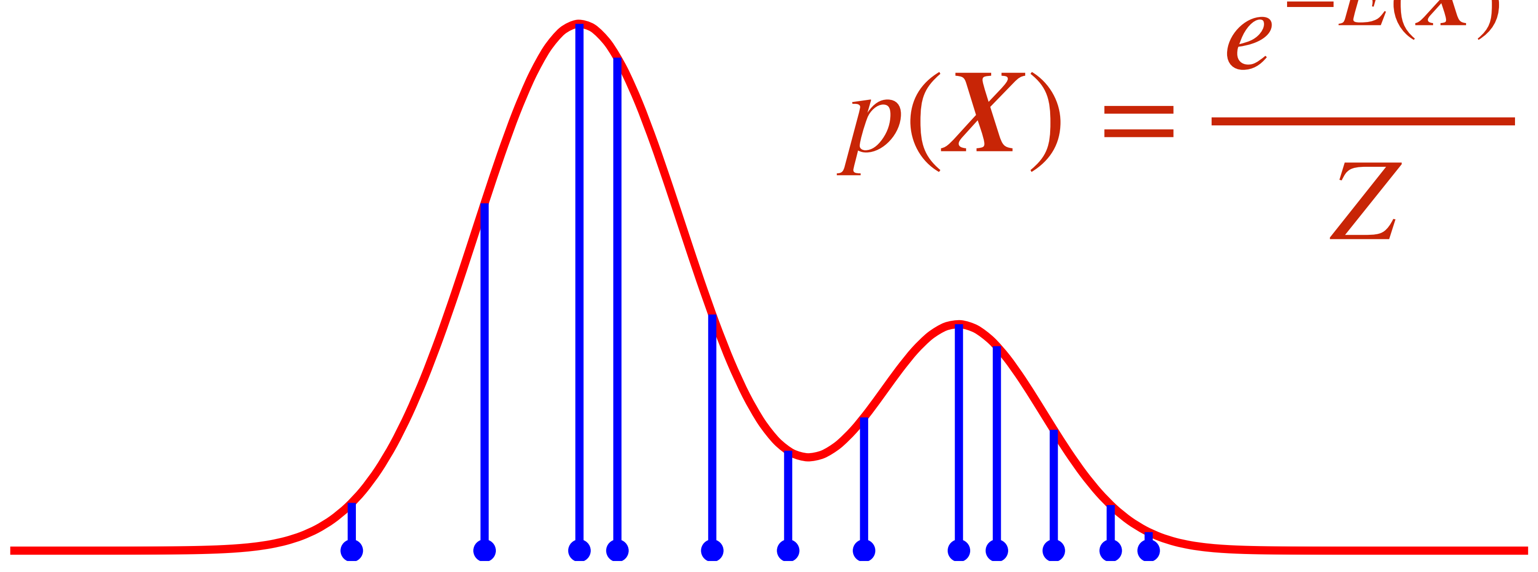
$$\frac{\partial p(\boldsymbol{X},t)}{\partial t} + \nabla \cdot \left[p(\boldsymbol{X},t)\boldsymbol{v}\right] = 0$$

Statistical, quantum, fluid, ... physics insights into generative models

Leverage the power of modern generative models for science

Generative models	Statistical physics
Log-likelihood	Energy function
Score function	Force
Latent variables	Collective variables
Partition function	Free energy calculation
Sampling	MD/MC

Boltzmann machines 

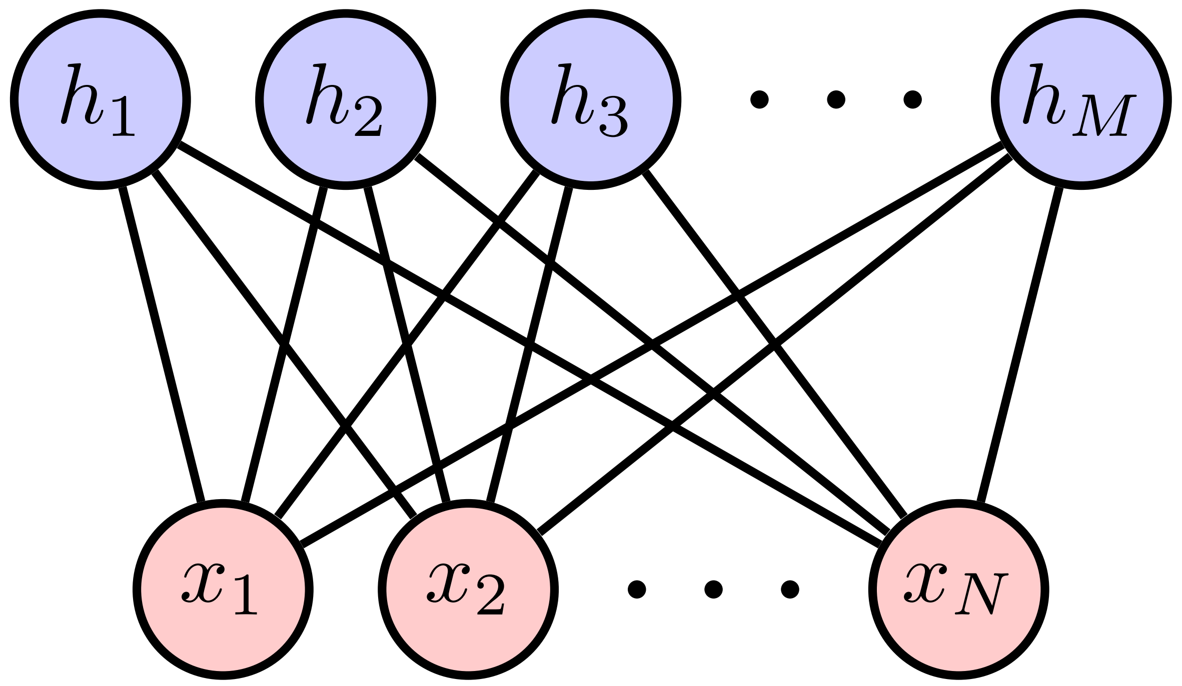


$$p(\boldsymbol{X}) = \frac{e^{-E(\boldsymbol{X})}}{Z}$$

Ackley, Hinton, Sejnowski, 1985

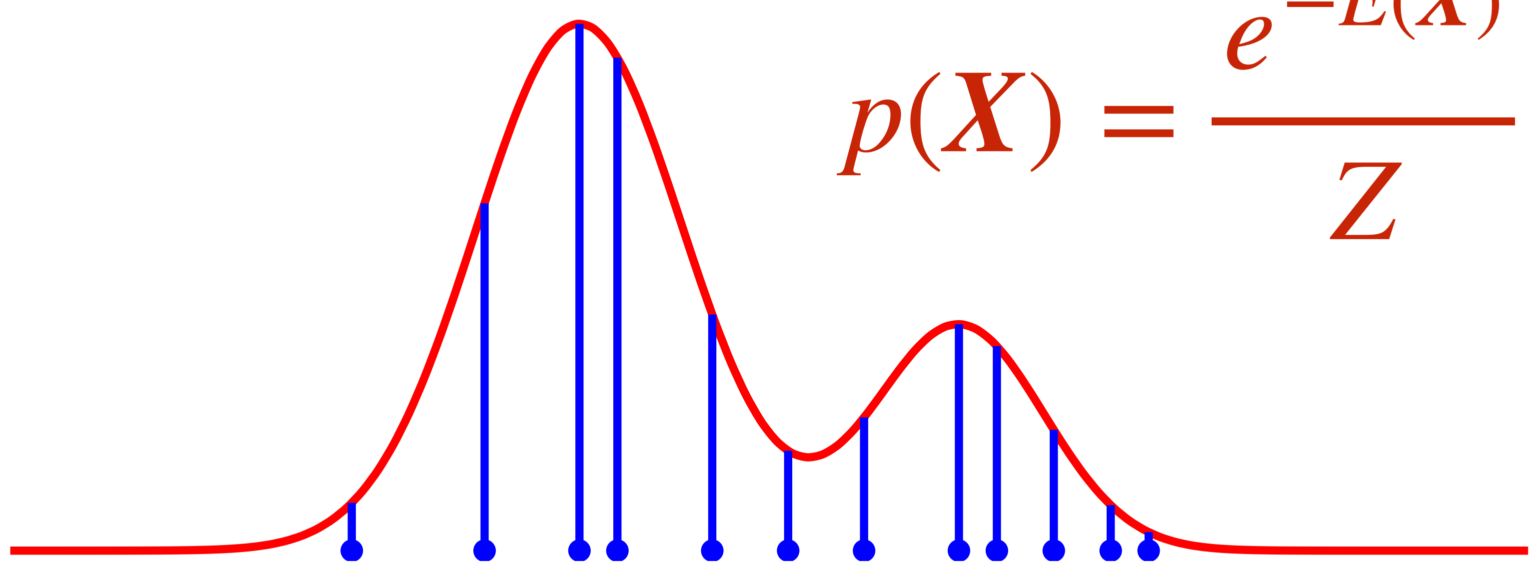
Learn

$$\mathscr{L} = \mathbb{E}_{\boldsymbol{X}\sim\text{data}}\left[-\ln p(\boldsymbol{X})\right]$$



$$\nabla_\theta \mathscr{L} = \mathop{\mathbb{E}}_{\boldsymbol{X}\sim\text{dataset}}\left[\nabla_\theta E\right] - \mathop{\mathbb{E}}_{\boldsymbol{X}\sim p(\boldsymbol{X})}\left[\nabla_\theta E\right]$$

Boltzmann machines 

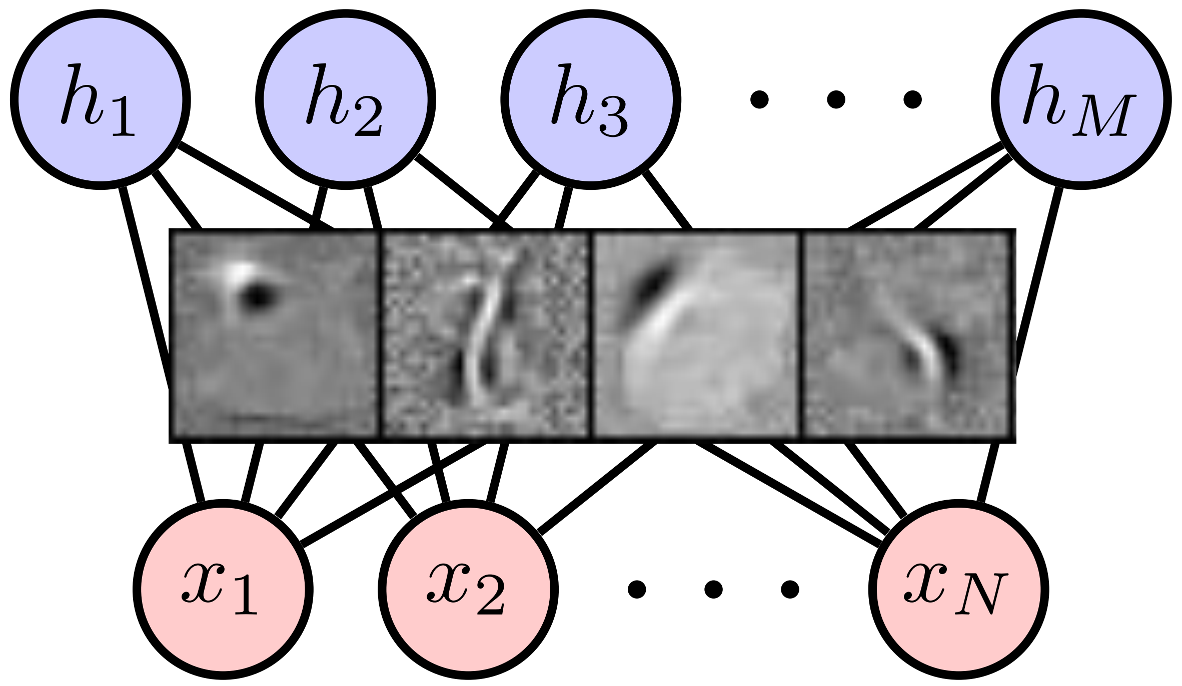


$$p(\boldsymbol{X}) = \frac{e^{-E(\boldsymbol{X})}}{Z}$$

Ackley, Hinton, Sejnowski, 1985

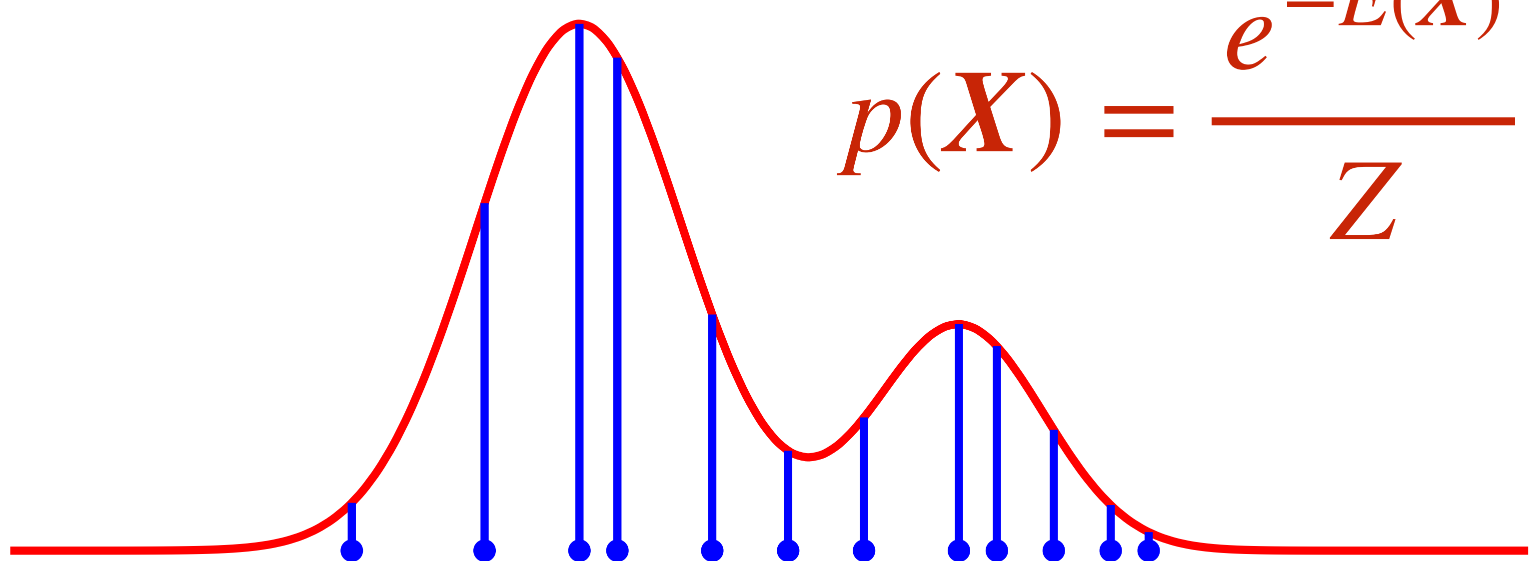
Learn

$$\mathscr{L} = \mathbb{E}_{\boldsymbol{X}\sim\text{data}}\left[-\ln p(\boldsymbol{X})\right]$$



$$\nabla_\theta \mathscr{L} = \mathop{\mathbb{E}}_{\boldsymbol{X}\sim\text{dataset}}\left[\nabla_\theta E\right] - \mathop{\mathbb{E}}_{\boldsymbol{X}\sim p(\boldsymbol{X})}\left[\nabla_\theta E\right]$$

Boltzmann machines 

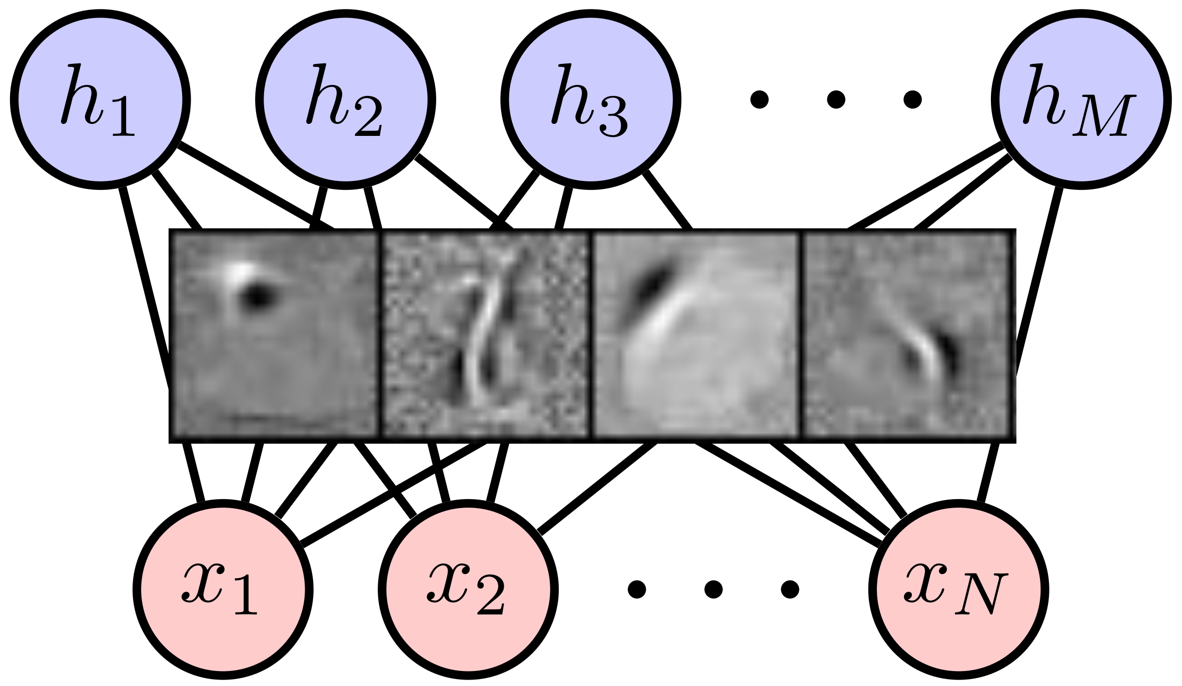


$$p(\boldsymbol{X}) = \frac{e^{-E(\boldsymbol{X})}}{Z}$$

Ackley, Hinton, Sejnowski, 1985

Learn

$$\mathscr{L} = \mathbb{E}_{X\sim\text{data}}\left[-\ln p(\boldsymbol{X})\right]$$

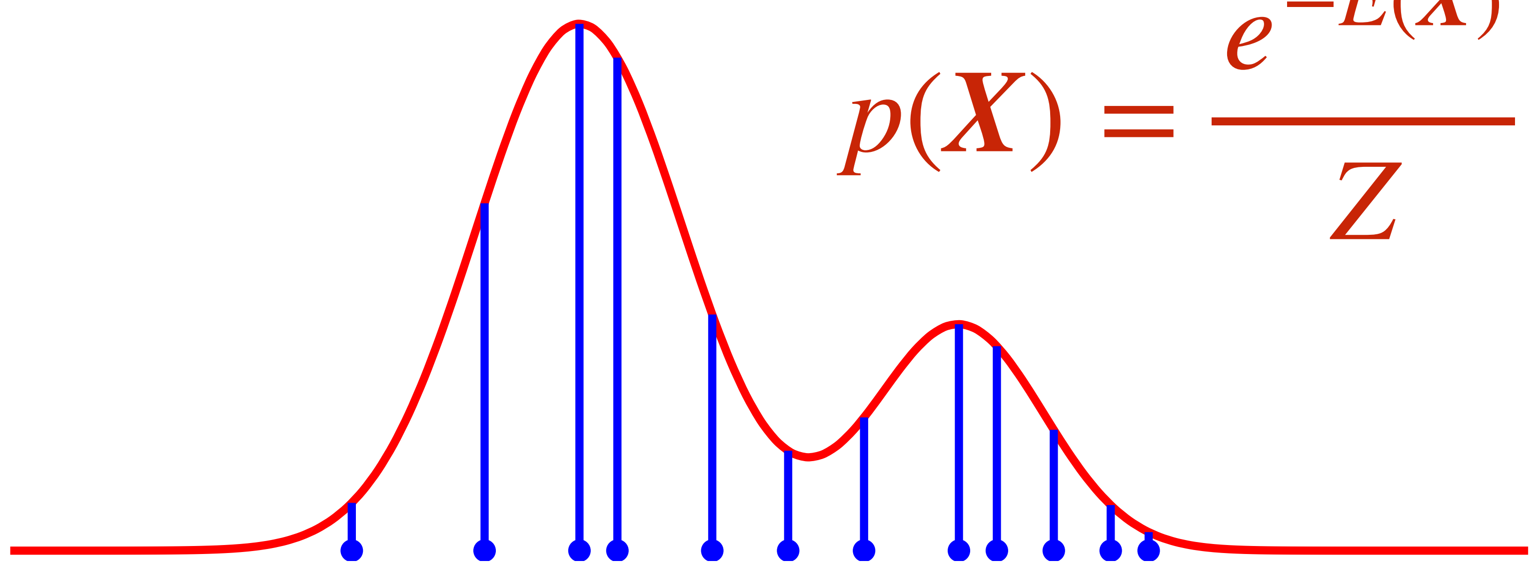


Generate

$$\boldsymbol{X} \sim p(\boldsymbol{X})$$

$$\nabla_\theta \mathscr{L} = \underset{X\sim\text{dataset}}{\mathbb{E}}\left[\nabla_\theta E\right] - \underset{X\sim p(X)}{\mathbb{E}}\left[\nabla_\theta E\right]$$

Boltzmann machines 

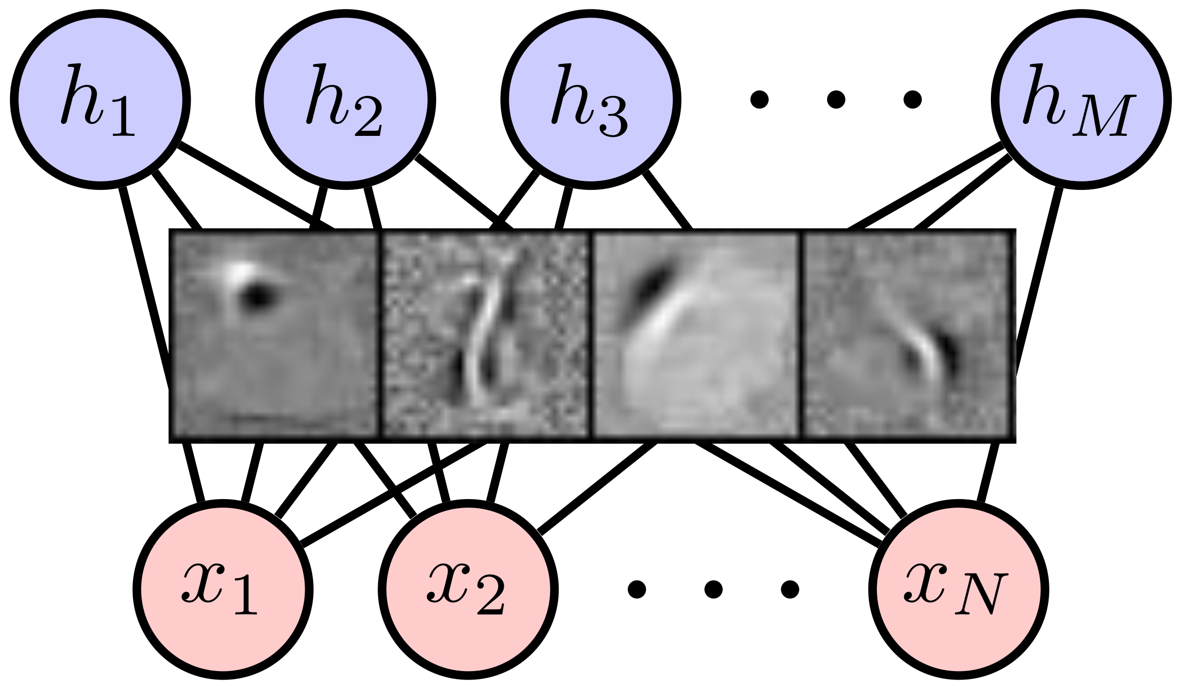


$$p(\boldsymbol{X}) = \frac{e^{-E(\boldsymbol{X})}}{Z}$$

Ackley, Hinton, Sejnowski, 1985

Learn

$$\mathscr{L} = \mathbb{E}_{X\sim\text{data}}\left[-\ln p(\boldsymbol{X})\right]$$

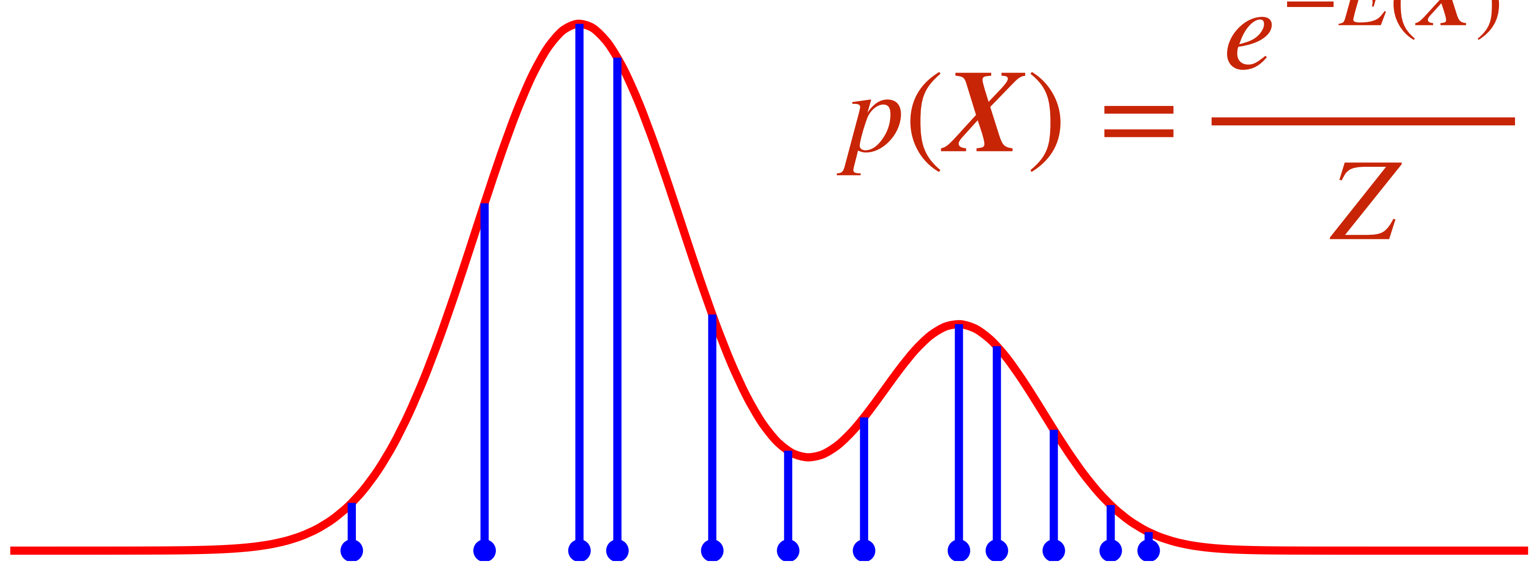


Generate

$$\boldsymbol{X} \sim p(\boldsymbol{X})$$

$$\nabla_\theta \mathscr{L} = \mathop{\mathbb{E}}_{X\sim\text{dataset}}\left[\nabla_\theta E\right] - \mathop{\mathbb{E}}_{X\sim p(X)}\left[\nabla_\theta E\right]$$

Boltzmann machines 



$$p(\boldsymbol{X}) = \frac{e^{-E(\boldsymbol{X})}}{Z}$$

Ackley, Hinton, Sejnowski, 1985

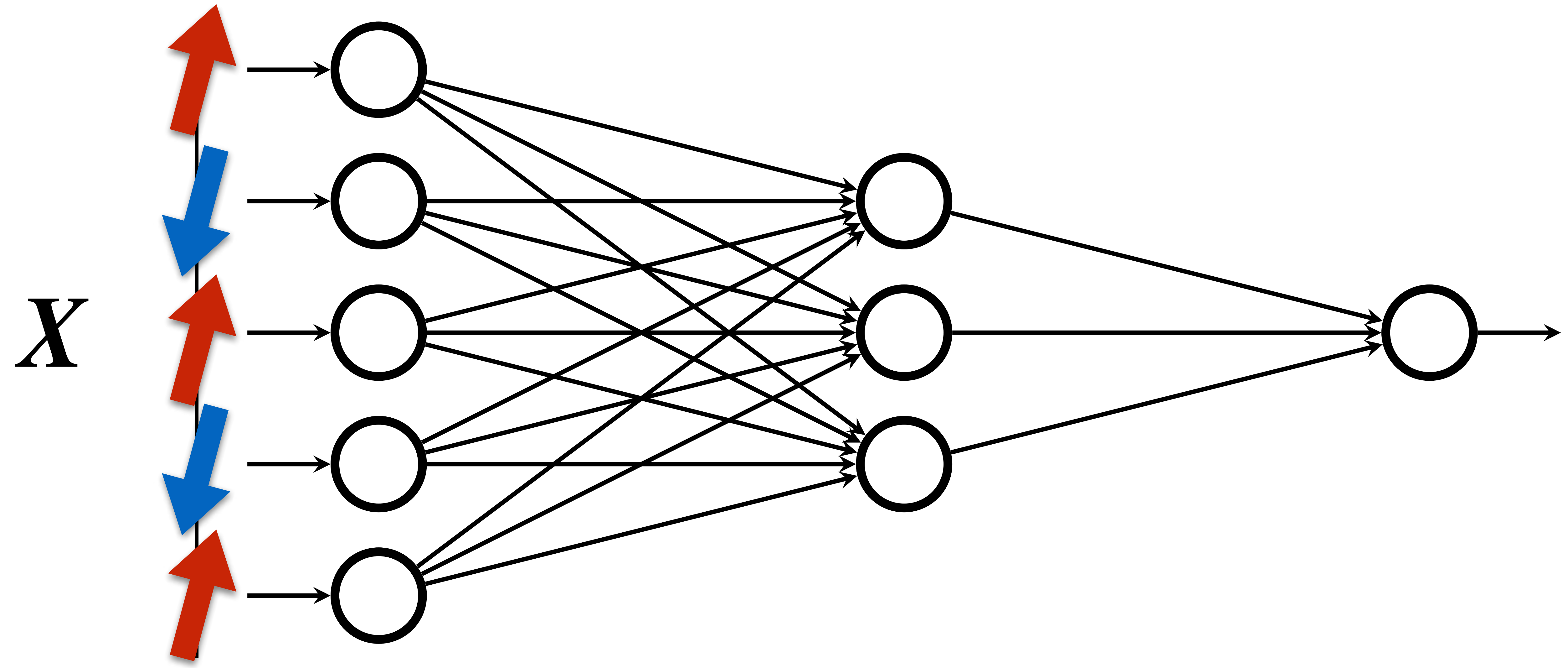
2210.10318

GAUSSIAN-BERNOULLI RBMS WITHOUT TEARS 😂

Renjie Liao*[1], Simon Kornblith[2], Mengye Ren[3], David J. Fleet[2,4,5], Geoffrey Hinton[2,4,5]

$$\nabla_\theta \mathscr{L} = \underset{X\sim\text{dataset}}{\mathbb{E}} \left[\nabla_\theta E\right] - \underset{X\sim p(X)}{\mathbb{E}} \left[\nabla_\theta E\right]$$

So, why bother ?



$$p(\boldsymbol{X}) \geq 0$$

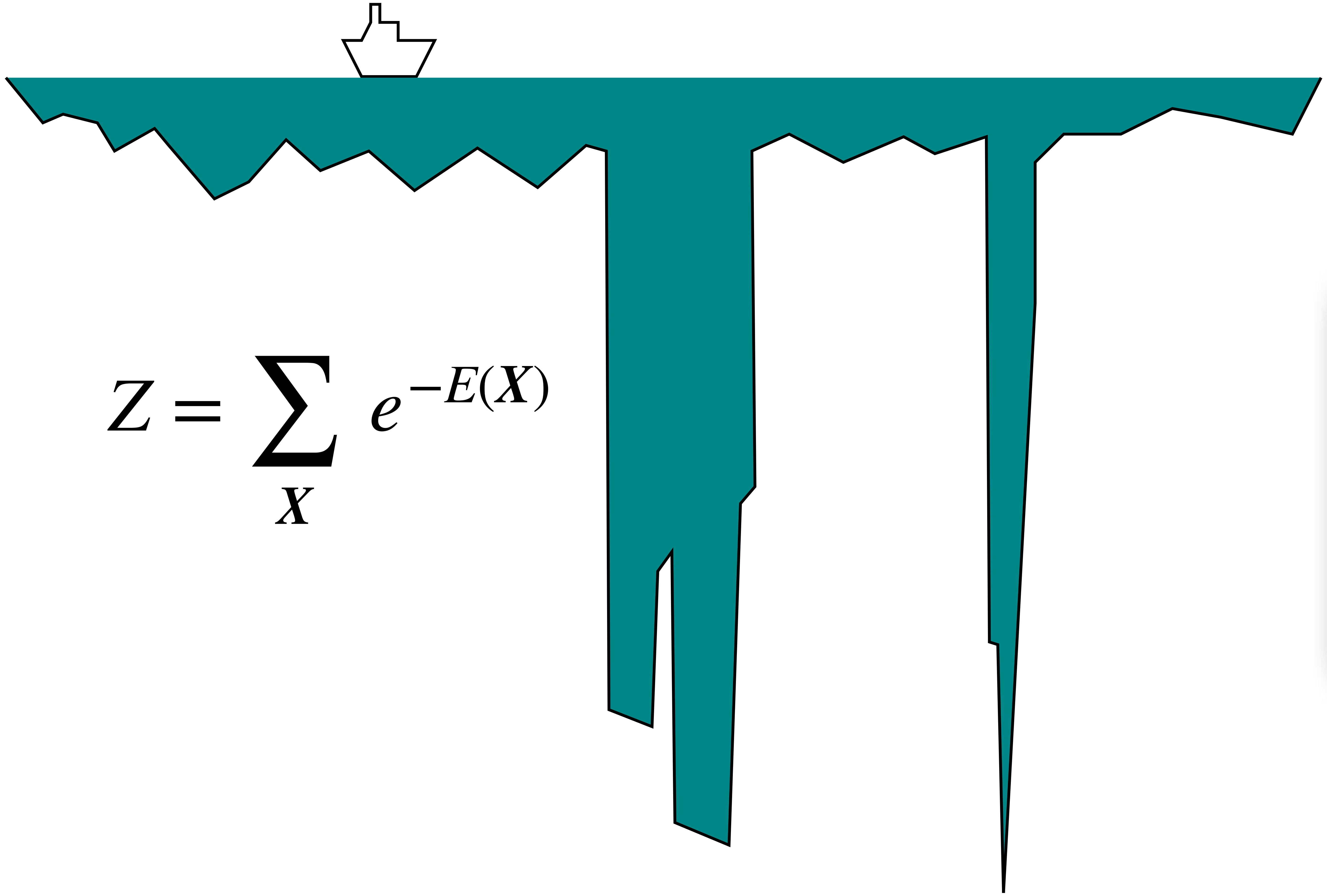
Normalization ?

$$\sum_{X} p(X)$$

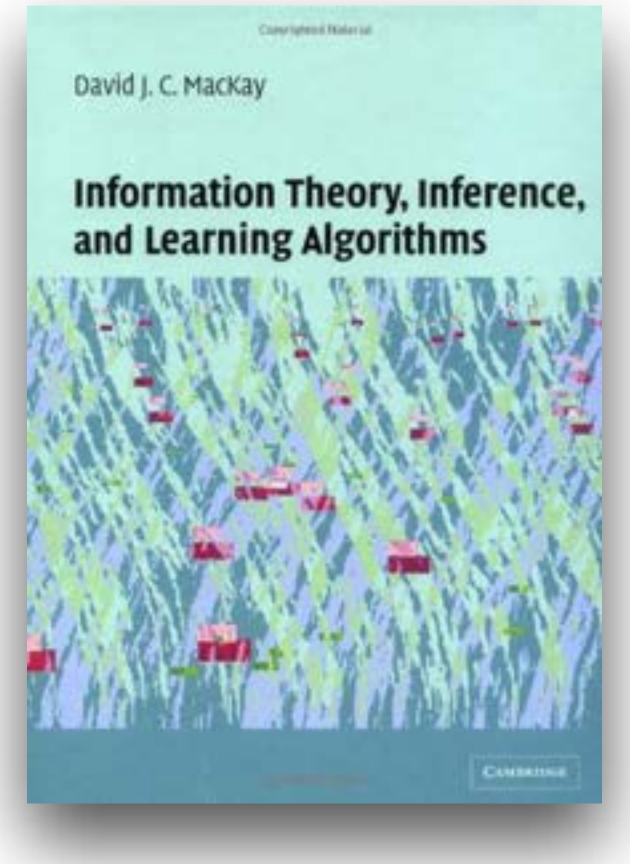
Sampling ?

$$\boldsymbol{X} \sim p(\boldsymbol{X})$$

The difficulty of normalization



$$Z = \sum_{X} e^{-E(X)}$$

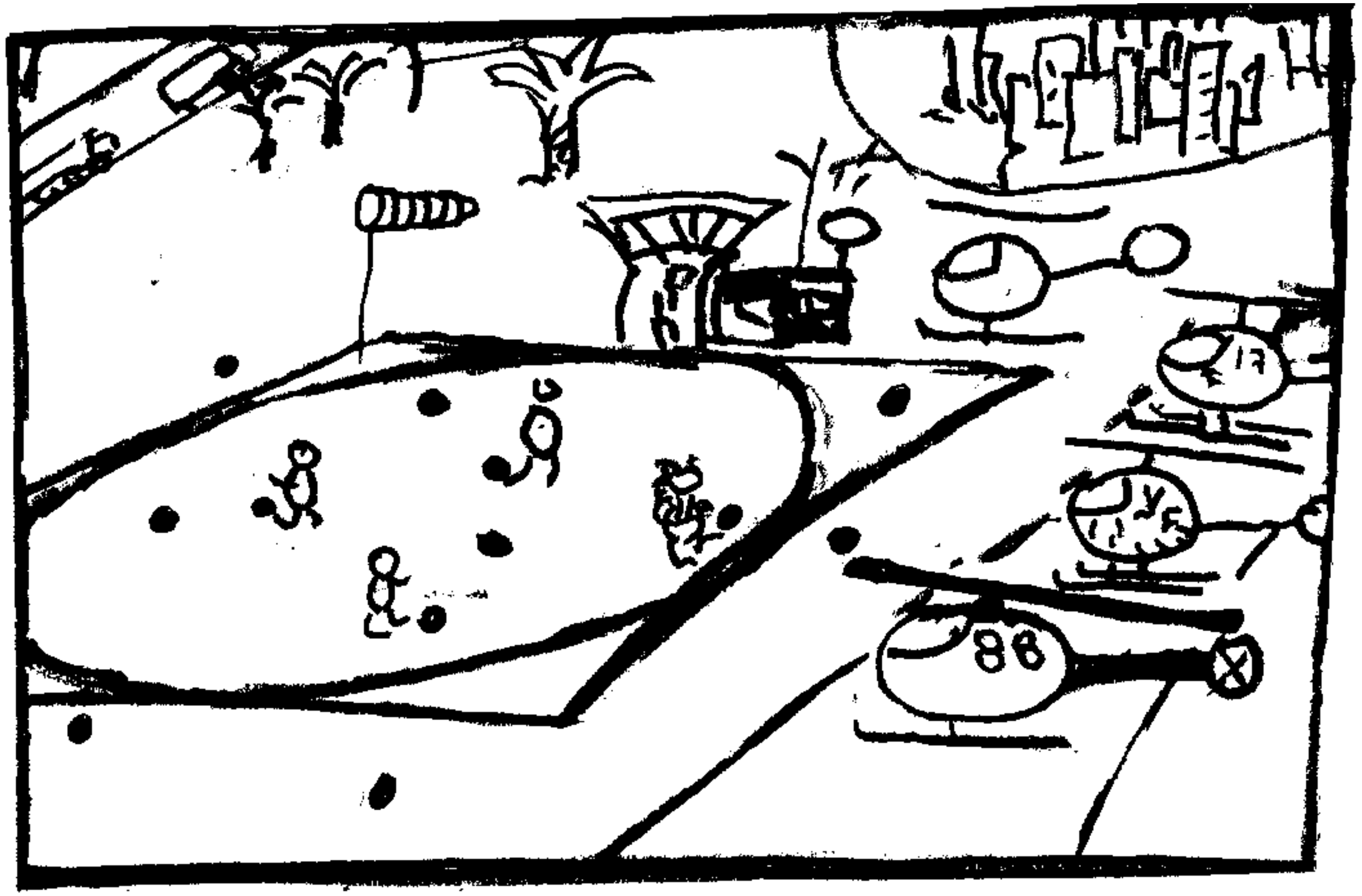


“Intractable” partition function Z

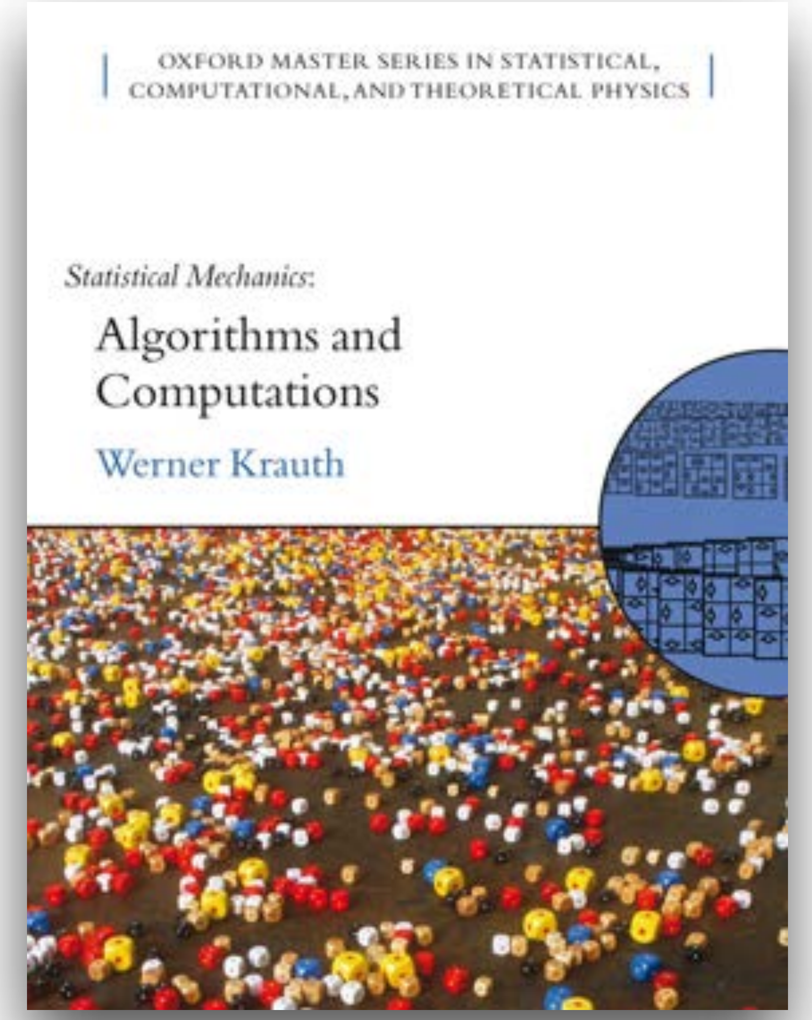
appears widely in machine learning and statistical physics (entropy and free energy calculation)

The difficulty of sampling

$$X \sim p(X)$$



Adults computing the number π at the Monte Carlo heliport.



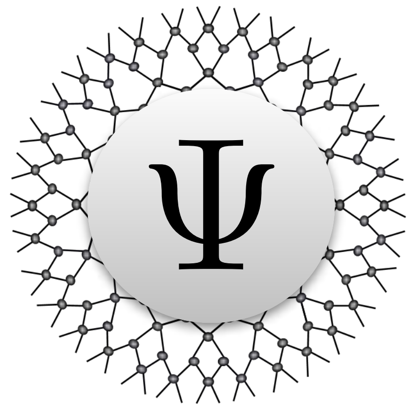
Direct sampling is generally difficult in high-dimensional space

Generative models and their physics genes

Goodfellow,
NIPS tutorial, 1701.00160

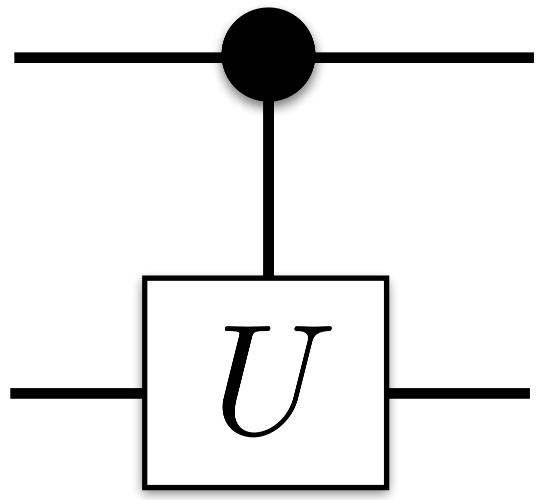
$p(\boldsymbol{X})$

- Explicit density
  - Tractable density
    - -Fully visible belief nets
      - -NADE
      - -MADE
      - -PixelRNN
    - -Change of variables models (nonlinear ICA)
  - Approximate density
    - Variational
      - Variational autoencoder
    - Markov Chain
      - Boltzmann machine
- Implicit density
  - Direct
    - GAN
  - Markov Chain
    - GSN



Tensor Networks

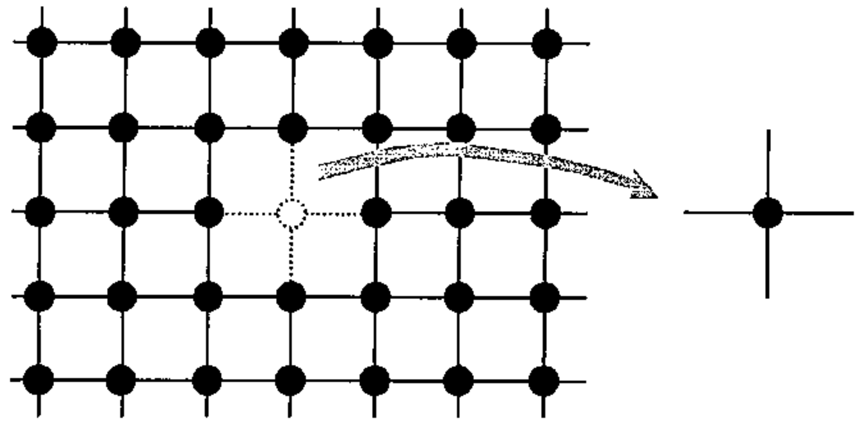
Han et al, PRX '18

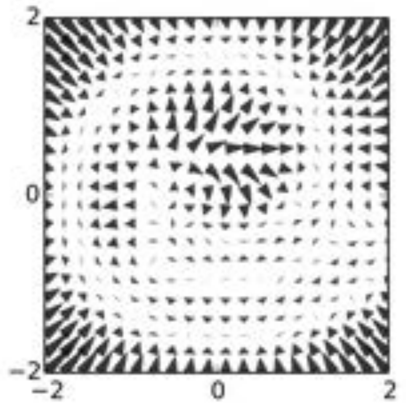


Quantum Circuits

Liu et al PRA '18

+Diffusion models





Autoregressive model

$$p(\boldsymbol{X}) = p(x_1)p(x_2 \,|\, x_1)p(x_3 \,|\, x_1, x_2)\cdots$$

"... the murderer is ___"

$$p(\_ \,|\, \ldots)$$

Normalization

$$\sum_{x_1} p(x_1) \sum_{x_2} p(x_2 \,|\, x_1) \sum_{x_3} p(x_3 \,|\, x_1, x_2)\cdots$$

Sampling

$$x_1 \sim p(x_1)$$

$$x_2 \sim p(x_2 \,|\, x_1)$$

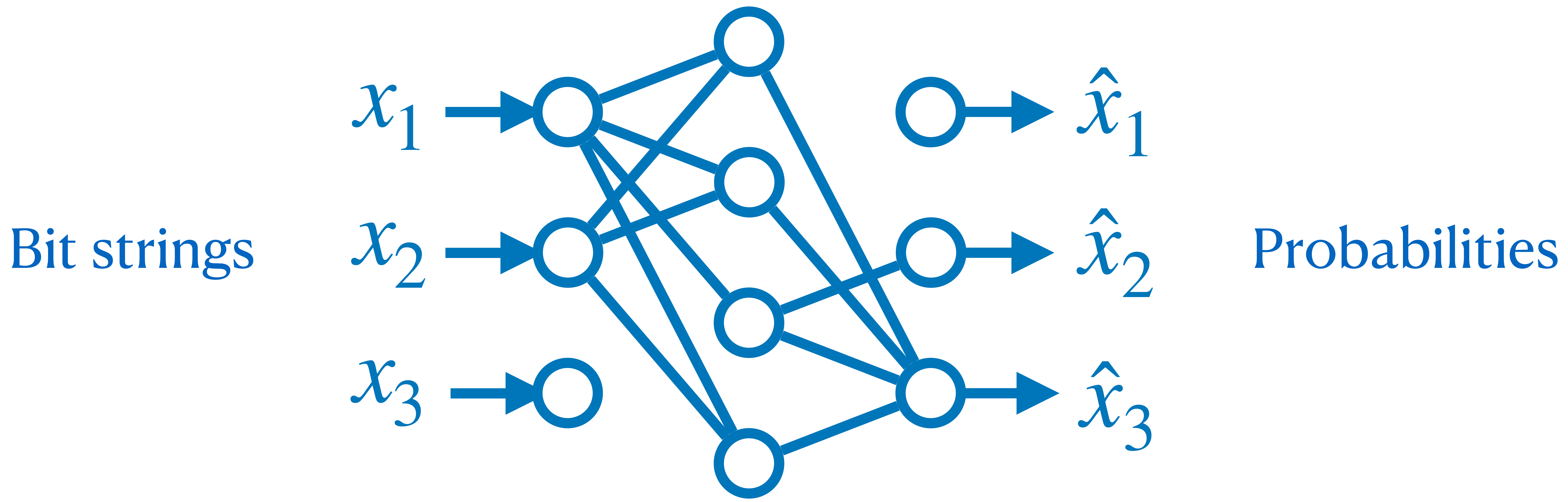
$$\vdots$$

Autoregressive models: alphabets, actions, and atoms

A unified perspective to LLM, RL, and atomistic modeling

Implementation: autoregressive masks

Masked Autoencoder Germain et al, 1502.03509



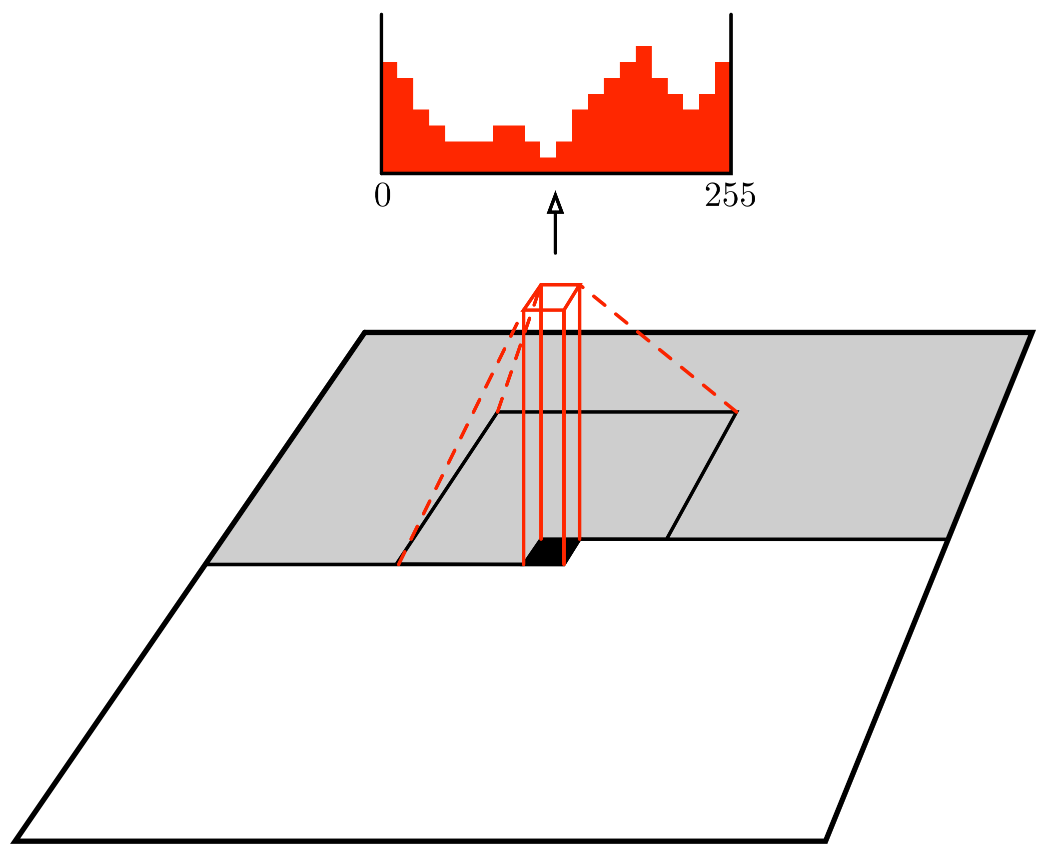
$p(x_1) = \text{Bernoulli}(\hat{x}_1)$ $\quad p(x_2 \mid x_1) = \text{Bernoulli}(\hat{x}_2)$ $\quad p(x_3 \mid x_1, x_2) = \text{Bernoulli}(\hat{x}_3)$

https://github.com/karpathy/pytorch-made

Implementation: autoregressive masks

Mask convolutional kernel

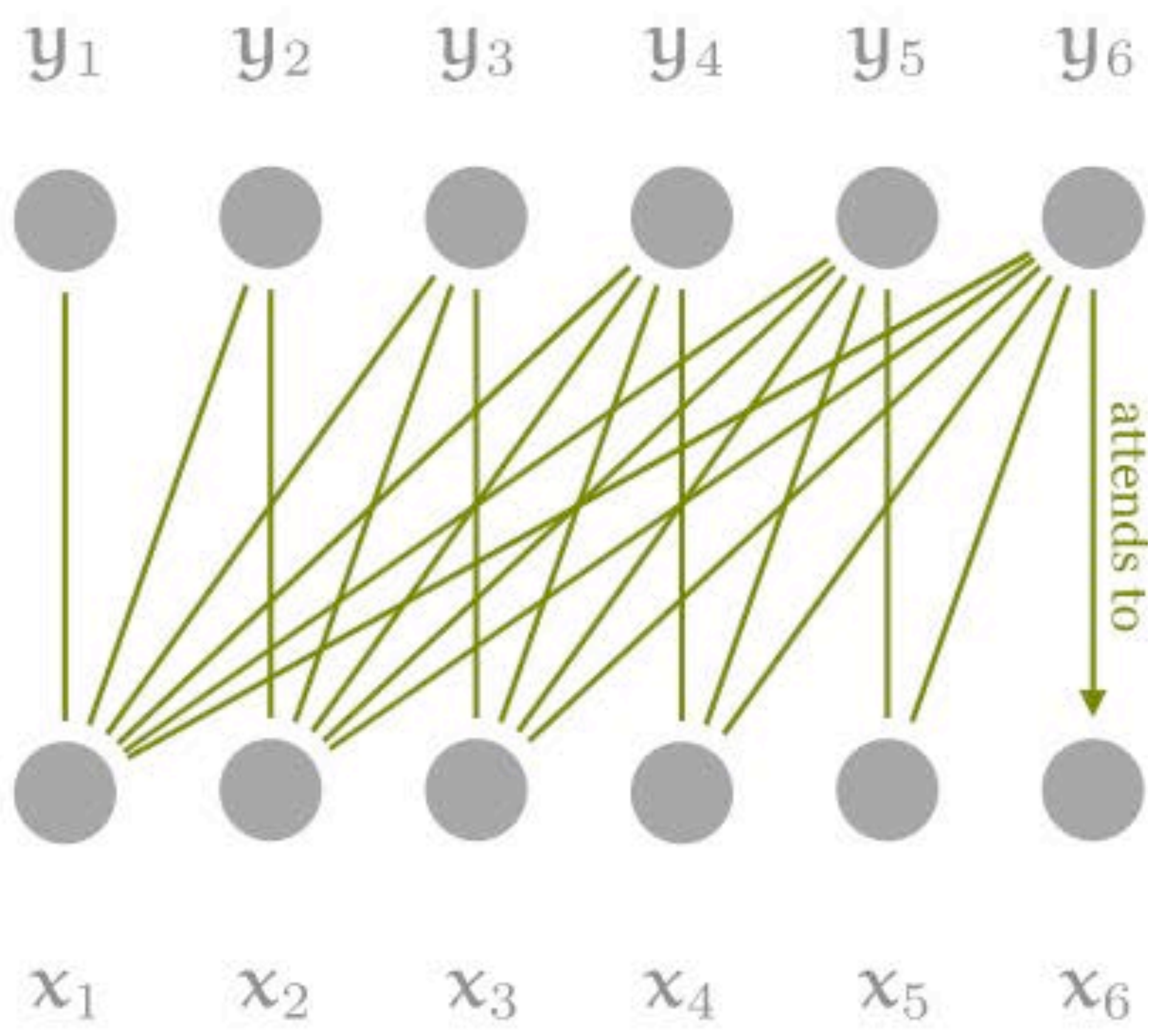
PixelCNN, van den Oord et al, 1601.06759



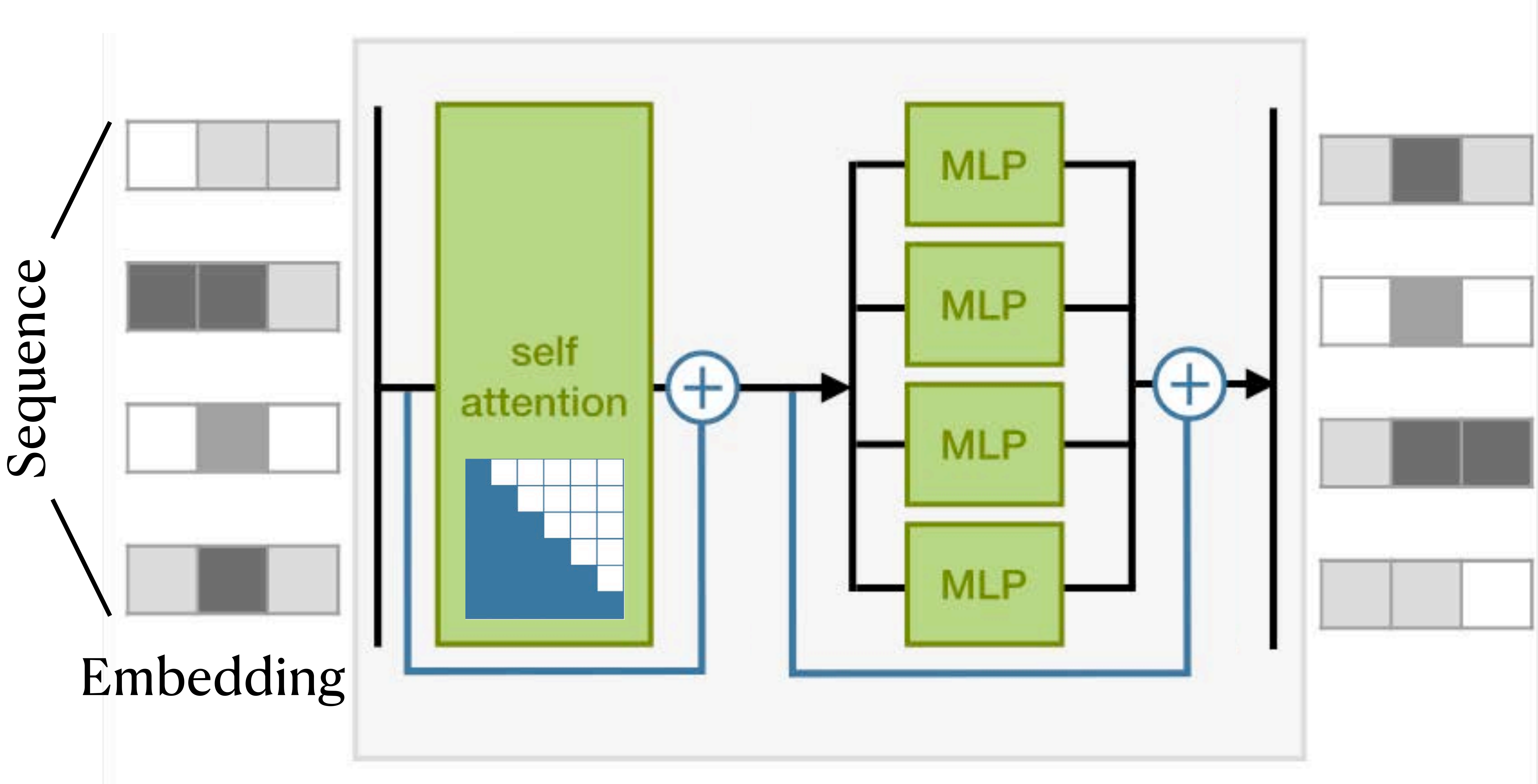
1	1	1	1	1
1	1	1	1	1
1	1	0	0	0
0	0	0	0	0
0	0	0	0	0

Mask self-attention matrix

Causal transformer, Vaswani et al 1706.03762



The autoregressive transformer



Masked attention matrix => lower triangular Jacobian matrix => autoregressive model
Great at capturing long-range dependence; friendly to backpropagation and GPUs

Transformers from scratch, https://peterbloem.nl/blog/transformers picoGPT — wanglei@bright90:~ — vi gpt2_pico.py — 116×63

```python
import numpy as np

def gelu(x):
    return 0.5 * x * (1 + np.tanh(np.sqrt(2 / np.pi) * (x + 0.044715 * x**3)))

def softmax(x):
    exp_x = np.exp(x - np.max(x, axis=-1, keepdims=True))
    return exp_x / np.sum(exp_x, axis=-1, keepdims=True)

def layer_norm(x, g, b, eps: float = 1e-5):
    mean = np.mean(x, axis=-1, keepdims=True)
    variance = np.var(x, axis=-1, keepdims=True)
    return g * (x - mean) / np.sqrt(variance + eps) + b

def linear(x, w, b):
    return x @ w + b

def ffn(x, c_fc, c_proj):
    return linear(gelu(linear(x, **c_fc)), **c_proj)

def attention(q, k, v, mask):
    return softmax(q @ k.T / np.sqrt(q.shape[-1]) + mask) @ v

def mha(x, c_attn, c_proj, n_head):
    x = linear(x, **c_attn)
    qkv_heads = list(map(lambda x: np.split(x, n_head, axis=-1), np.split(x, 3, axis=-1)))
    causal_mask = (1 - np.tri(x.shape[0], dtype=x.dtype)) * -1e10
    out_heads = [attention(q, k, v, causal_mask) for q, k, v in zip(*qkv_heads)]
    x = linear(np.hstack(out_heads), **c_proj)
    return x

def transformer_block(x, mlp, attn, ln_1, ln_2, n_head):
    x = x + mha(layer_norm(x, **ln_1), **attn, n_head=n_head)
    x = x + ffn(layer_norm(x, **ln_2), **mlp)
    return x

def gpt2(inputs, wte, wpe, blocks, ln_f, n_head):
    x = wte[inputs] + wpe[range(len(inputs))]
    for block in blocks:
        x = transformer_block(x, **block, n_head=n_head)
    return layer_norm(x, **ln_f) @ wte.T

def generate(inputs, params, n_head, n_tokens_to_generate):
    from tqdm import tqdm
    for _ in tqdm(range(n_tokens_to_generate), "generating"):
        logits = gpt2(inputs, **params, n_head=n_head)
        next_id = np.argmax(logits[-1])
        inputs.append(int(next_id))
    return inputs[len(inputs) - n_tokens_to_generate :]

def main(prompt: str, n_tokens_to_generate: int = 40, model_size: str = "124M", models_dir: str = "models"):
    from utils import load_encoder_hparams_and_params
    encoder, hparams, params = load_encoder_hparams_and_params(model_size, models_dir)
    input_ids = encoder.encode(prompt)
    assert len(input_ids) + n_tokens_to_generate < hparams["n_ctx"]
    output_ids = generate(input_ids, params, hparams["n_head"], n_tokens_to_generate)
    output_text = encoder.decode(output_ids)
    return output_text

if __name__ == "__main__":
    import fire
    fire.Fire(main)
"gpt2_pico.py" 62L, 2330B
```

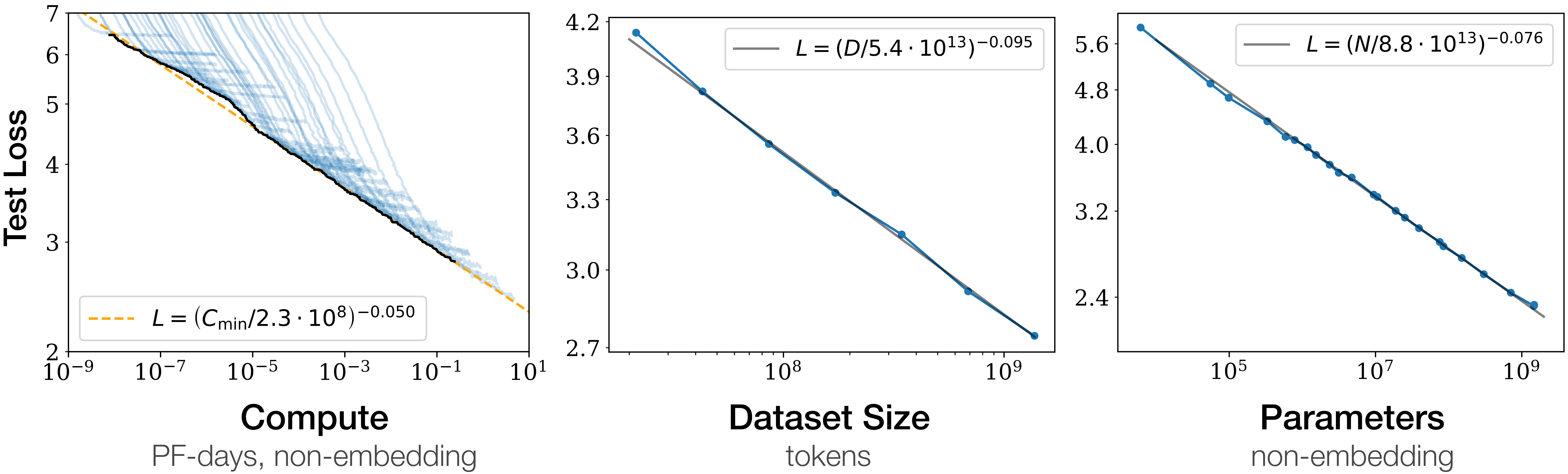
GPT2 in 60 lines of numpy

https://jaykmody.com/blog/gpt-from-scratch

Scaling law of the loss function

$$\mathscr{L} = \mathop{\mathbb{E}}_{X \sim \text{dataset}} \left[-\ln p(X) \right]$$

Kaplan, McCandlish, et al
2001.08361



"Predict resouces needed to sovle increasingly difficult tasks"
— Sam McCandlish at Aspen workshop "theoretical physics for machine learning" '19

https://sites.google.com/view/phys4ml/home

Scaling Laws for Neural Language Models, Kaplan et al, 2001.08361
Scaling Laws for Autoregressive Generative Modeling, Henighan et al, 2010.14701

It is natural to conjecture that the scaling relations will apply to other generative modeling tasks with a maximum likelihood loss, and perhaps in other settings as well. To this purpose, it will be interesting to test these relations on other domains, such as images, audio, and video models, and perhaps also for random network distillation. At this point we do not know which of our results depend on the structure of natural language data, and which are universal. It would also be exciting to find a theoretical framework from which the scaling relations can be derived: a 'statistical mechanics' underlying the 'thermodynamics' we have observed. Such a theory might make it possible to derive other more precise predictions, and provide a systematic understanding of the limitations of the scaling laws.

In the domain of natural language, it will be important to investigate whether continued improvement on the loss translates into improvement on relevant language tasks. Smooth quantitative change can mask major qualitative improvements: "more is different". For example, the smooth aggregate growth of the economy provides no indication of the specific technological developments that underwrite it. Similarly, the smooth improvements in language model loss may hide seemingly qualitative changes in capability.

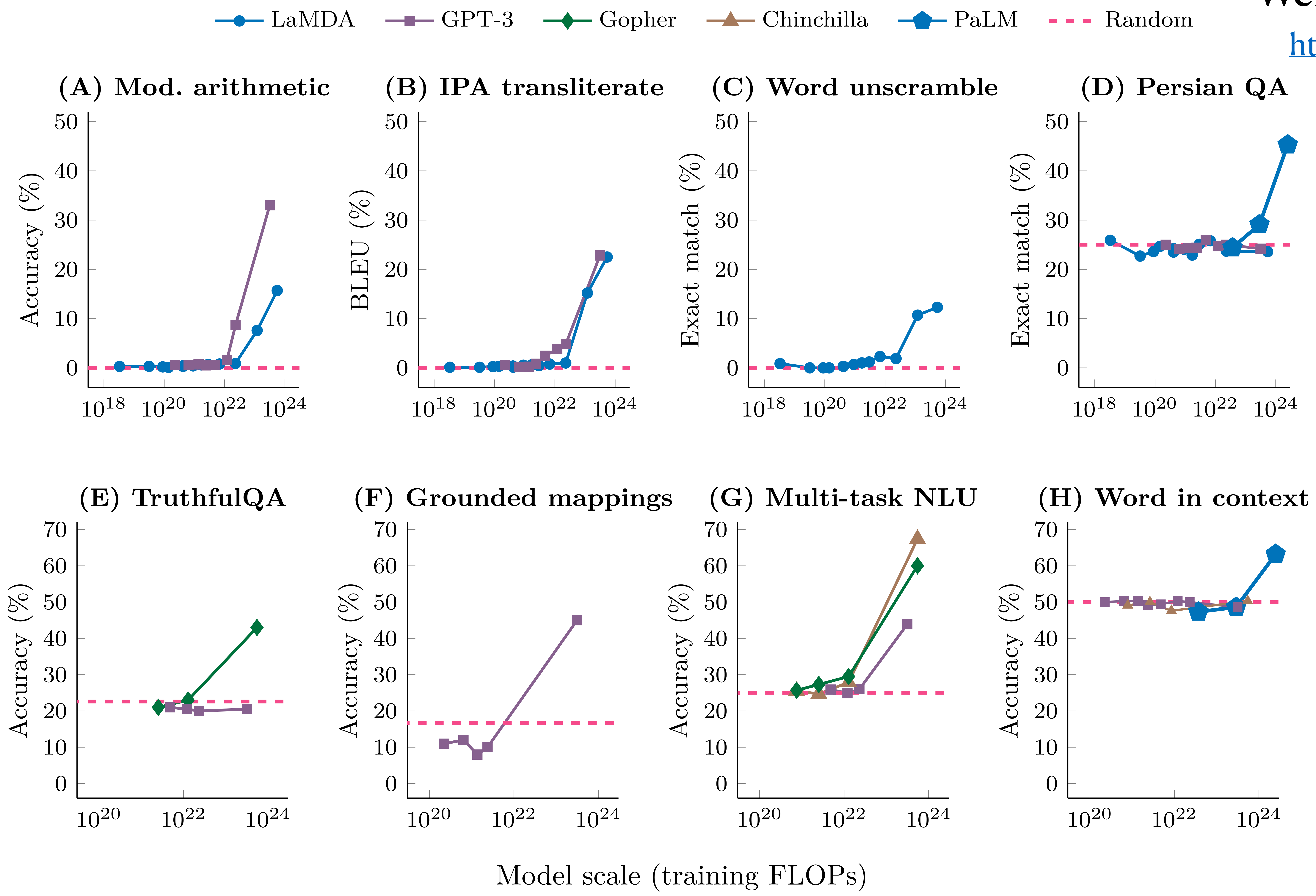
GPT-1 GPT-2 GPT-3 GPT-3.5 GPT-4

2018 2019 2020 2021 2022 2023

Emergent abilities: more is different

Wei et al, 2206.07682

https://www.jasonwei.net/blog/emergence



Avogadro constant number of FLOPs

2304.15004

Are Emergent Abilities of Large Language Models a Mirage?

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo

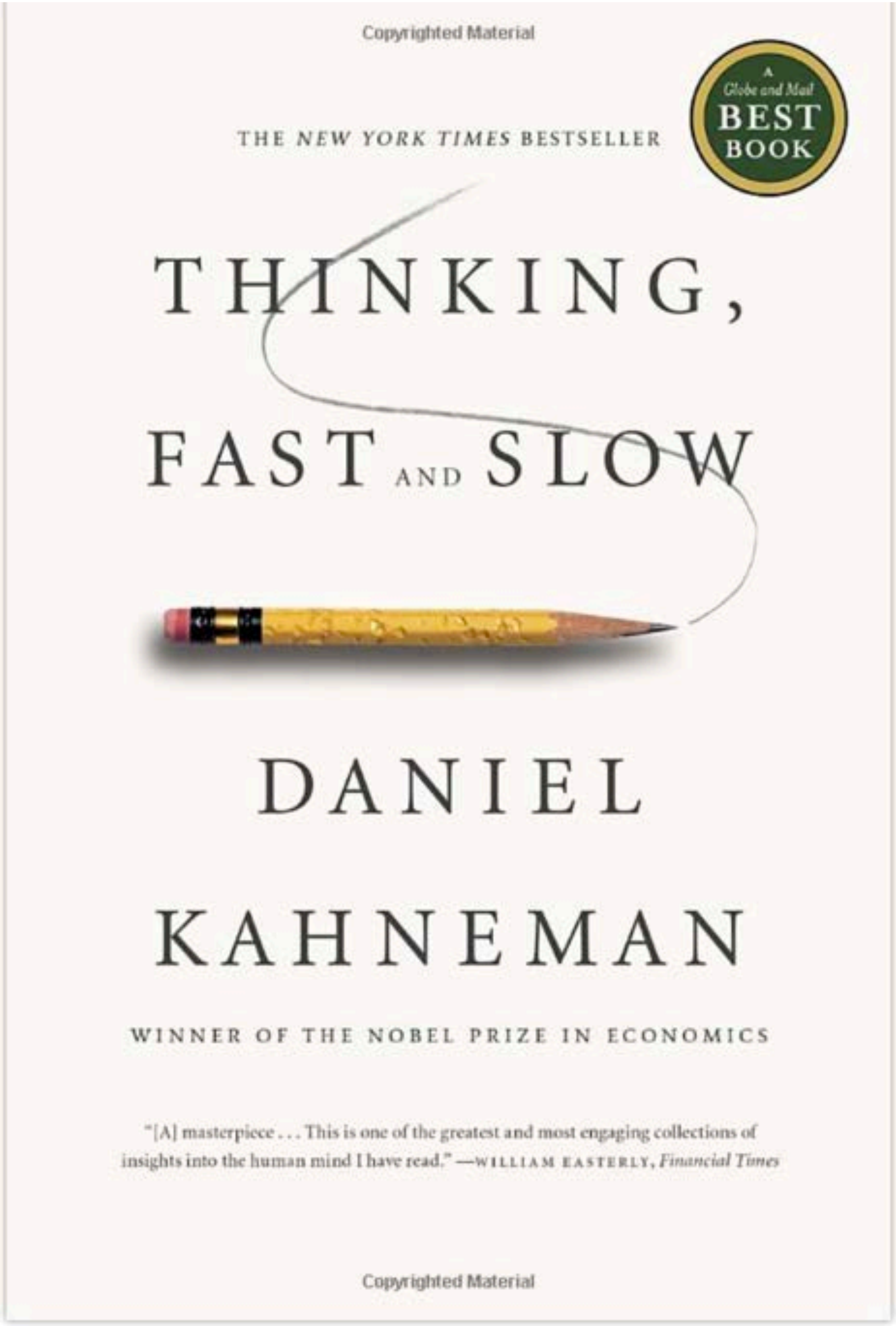
Computer Science, Stanford University

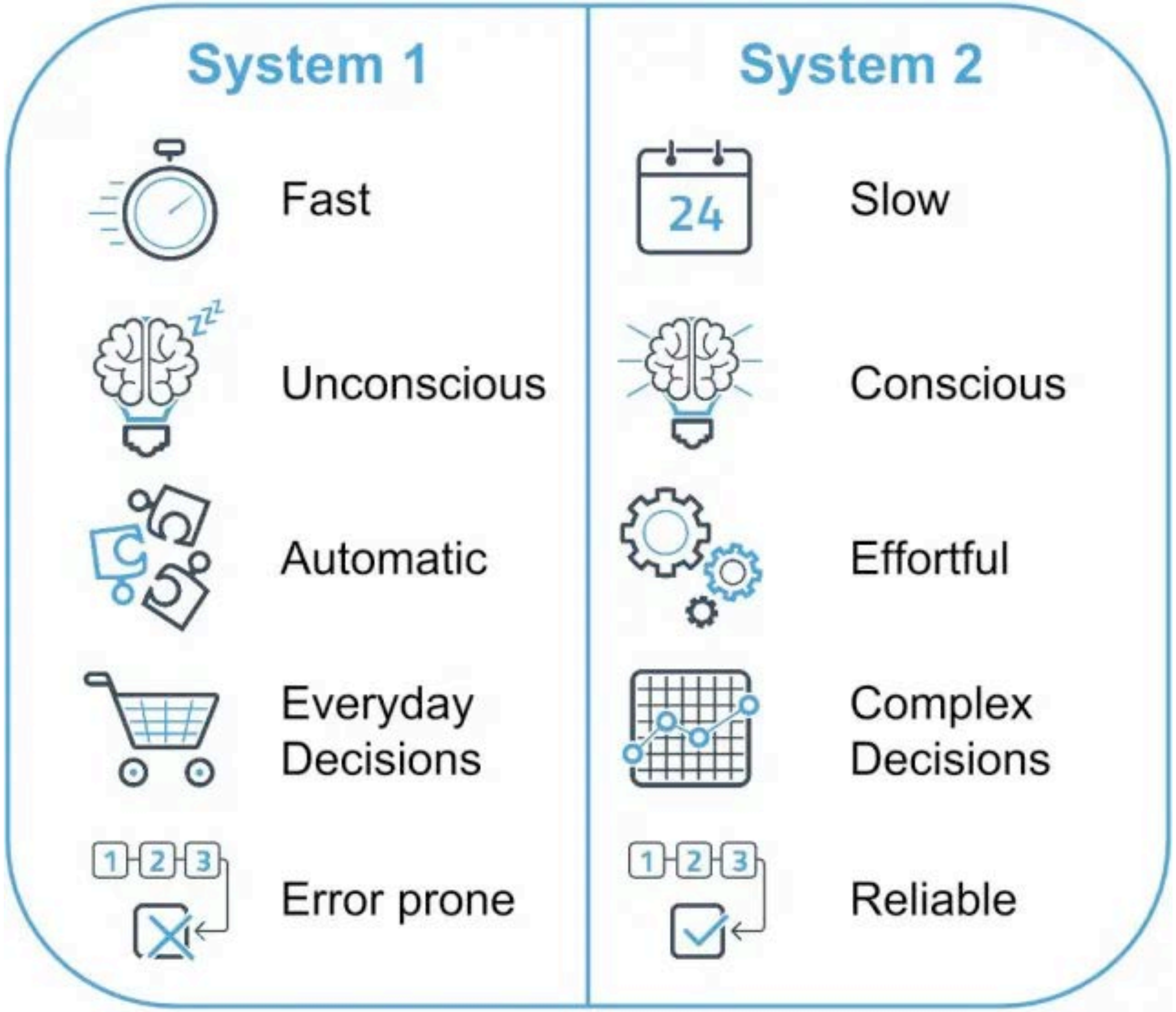
$$\text{Accuracy}(N) \approx p_N(\text{single token correct})^{\text{num. of tokens}} = \exp\Big(-(N/c)^{\alpha}\Big)^{L}$$

"The researcher’s choice of metric can nonlinearly and/or discontinuously transform the error rate in a manner that causes the model performance to appear sharp and unpredictable.“

Can LLM reason ?

Autoregressive language models are fast thinkers





Fast thinkers rely on good intuitions

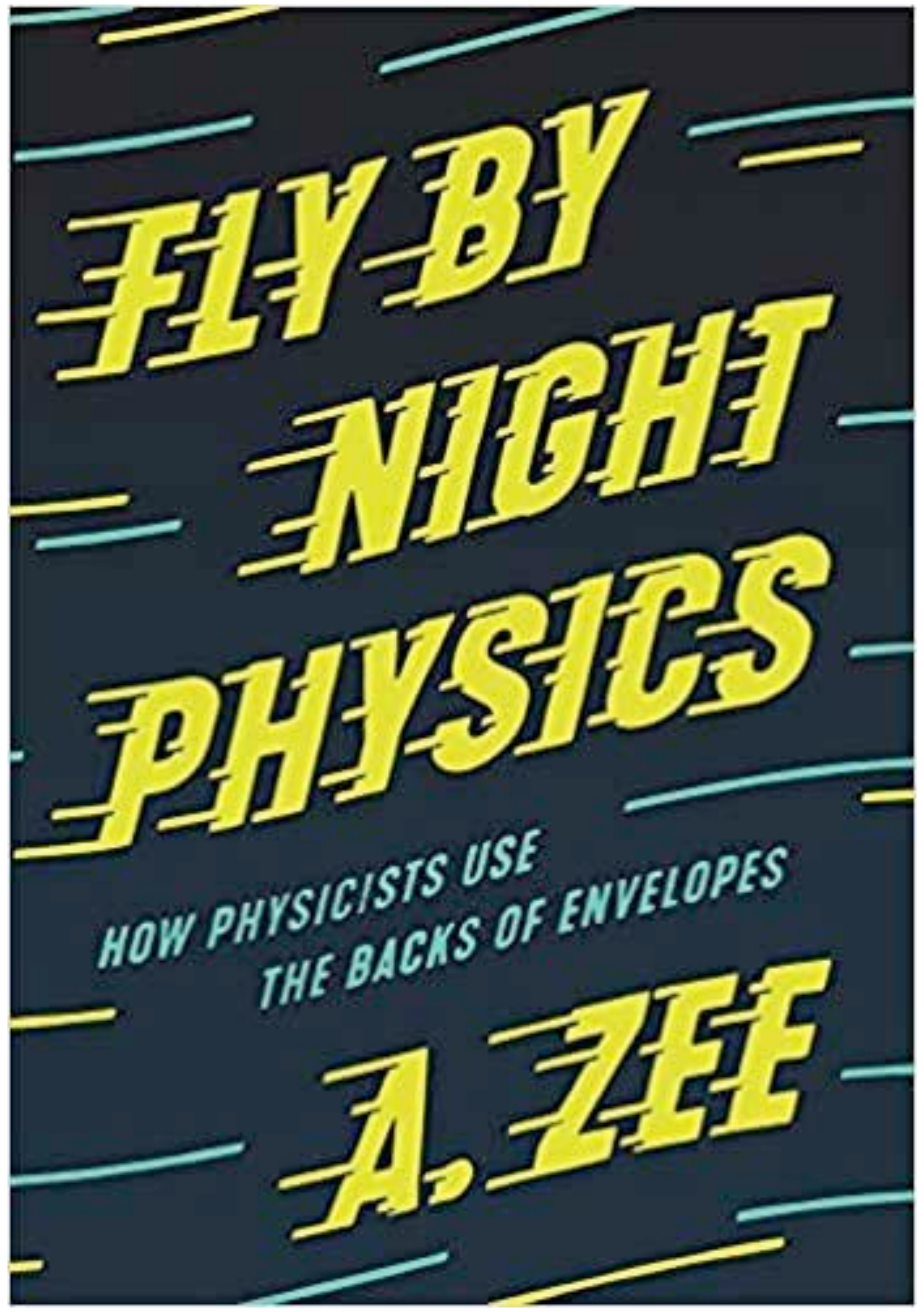
物理直觉是如何养成的？徐一鸿：思考比计算要困难得多

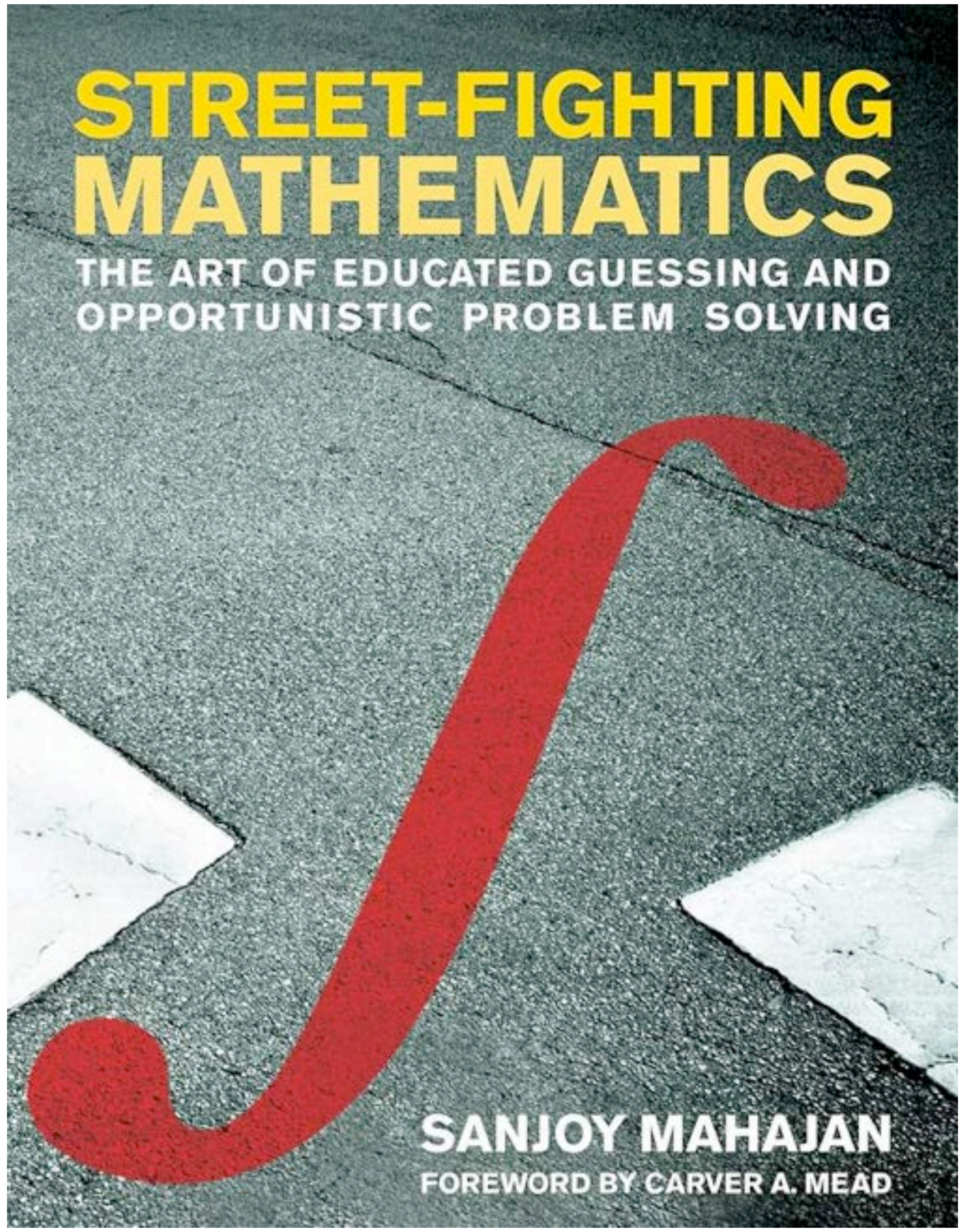
澎湃新闻记者 曹年润
2022-12-14 08:28 来源：澎湃新闻

· “重要的是要帮助学生培养更多的物理直觉，这也是我写这本书的原因之一。如何培养物理直觉？我在书里提到了两种可能性：天生就拥有物理直觉，或者通过不断练习发展物理直觉。”

徐一鸿教授。

“你应该先思考。思考比计算要困难得多，大多数人都可以坐下来计算，但不计算就思考问题，这是极其困难的。”





System 1 thinking in physics and math:
getting answers quickly without lengthy calculations

“Never never calculate unless you already know the answer! ”—John Wheeler

Autoregressive model is more than language modeling

“Language” => token sequence => bitstream => **ANYTHING**

Speech: WaveNet 1609.03499

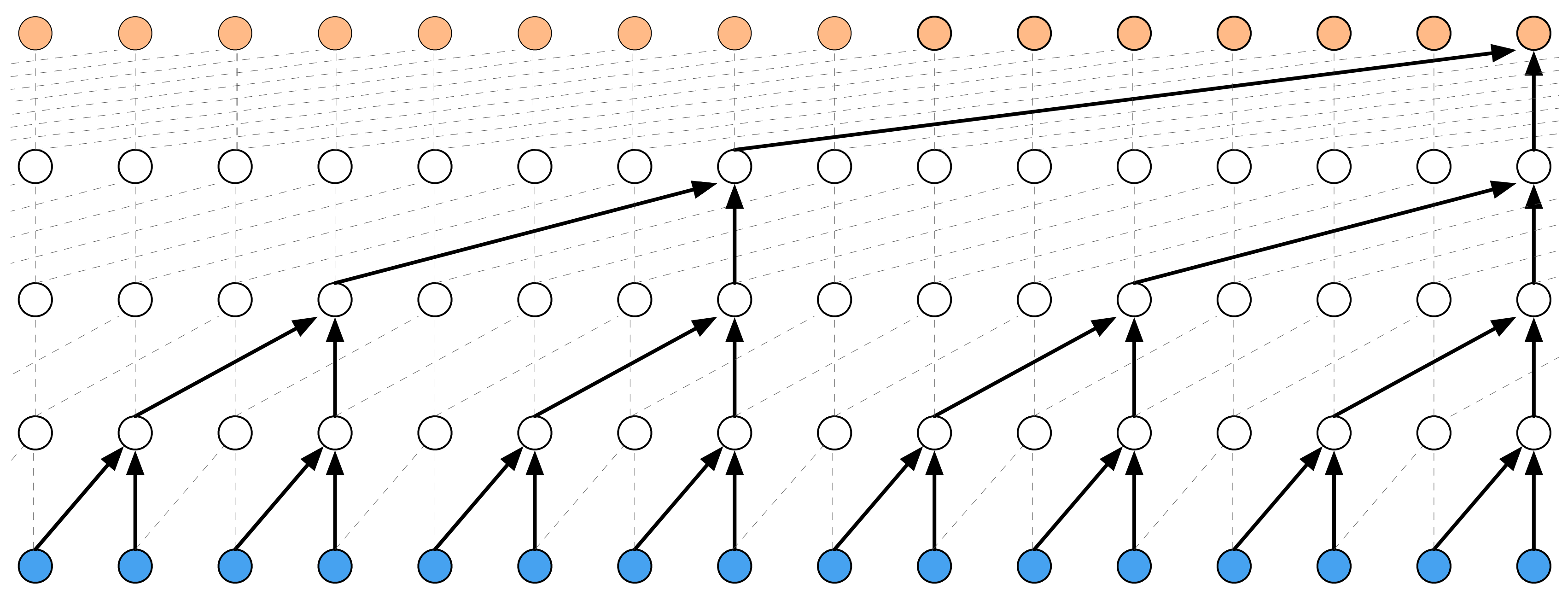
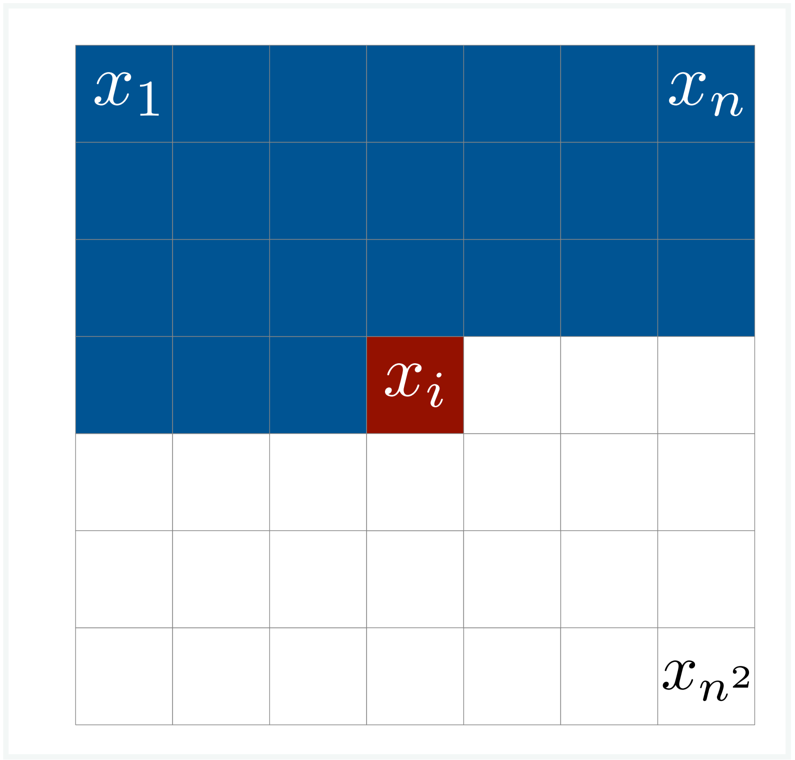
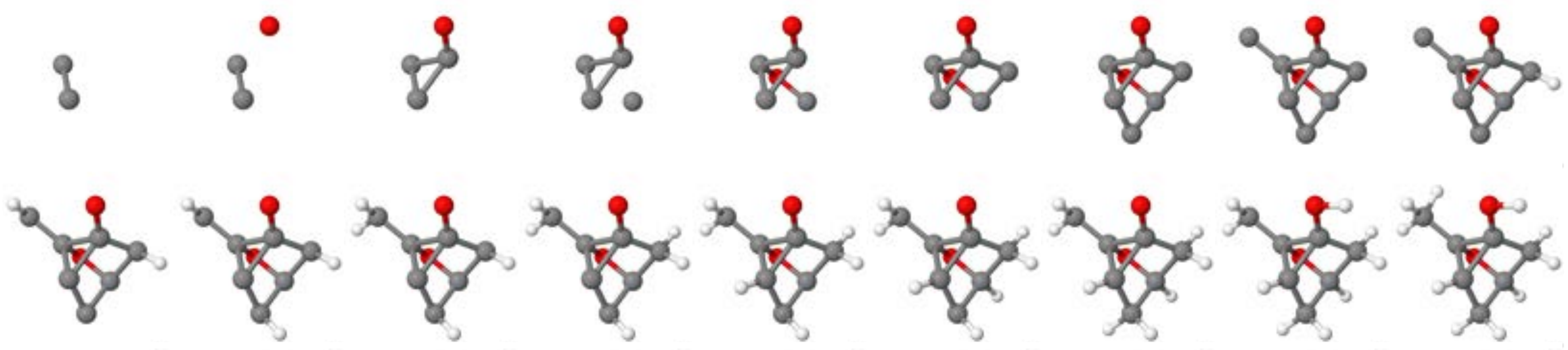


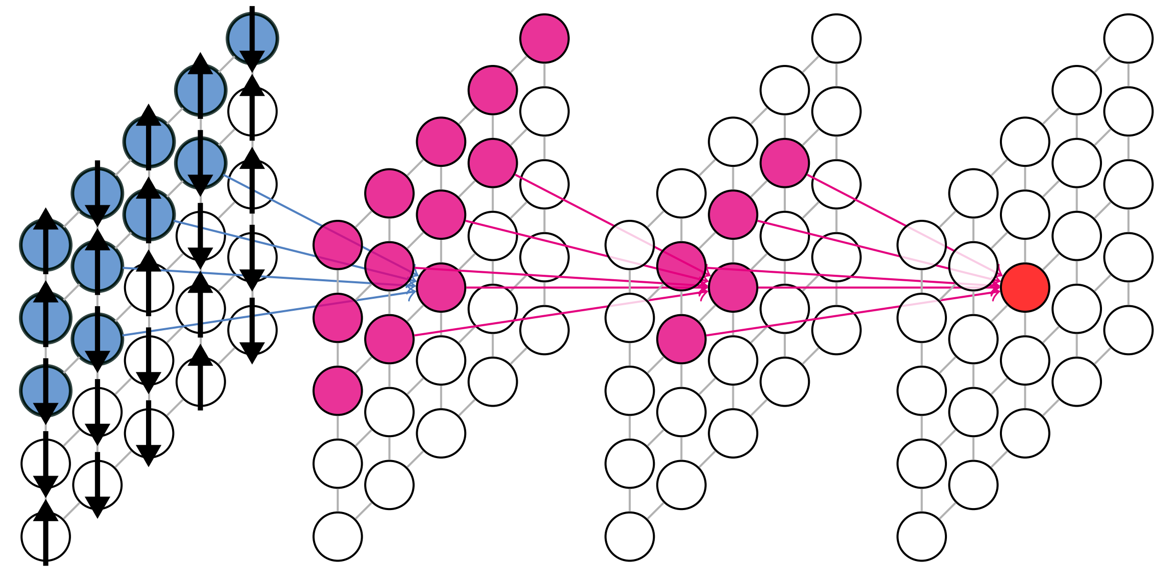
Image: PixelCNN 1601.06759



Molecules: 1810.11347

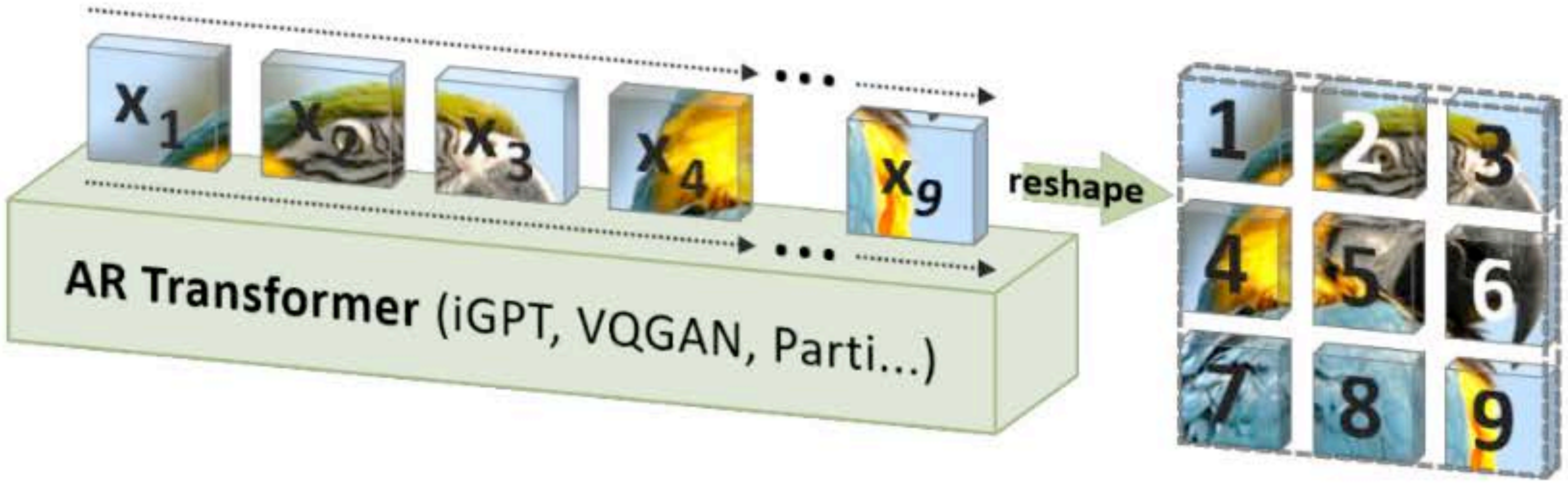


Ising spins: 1809.10606



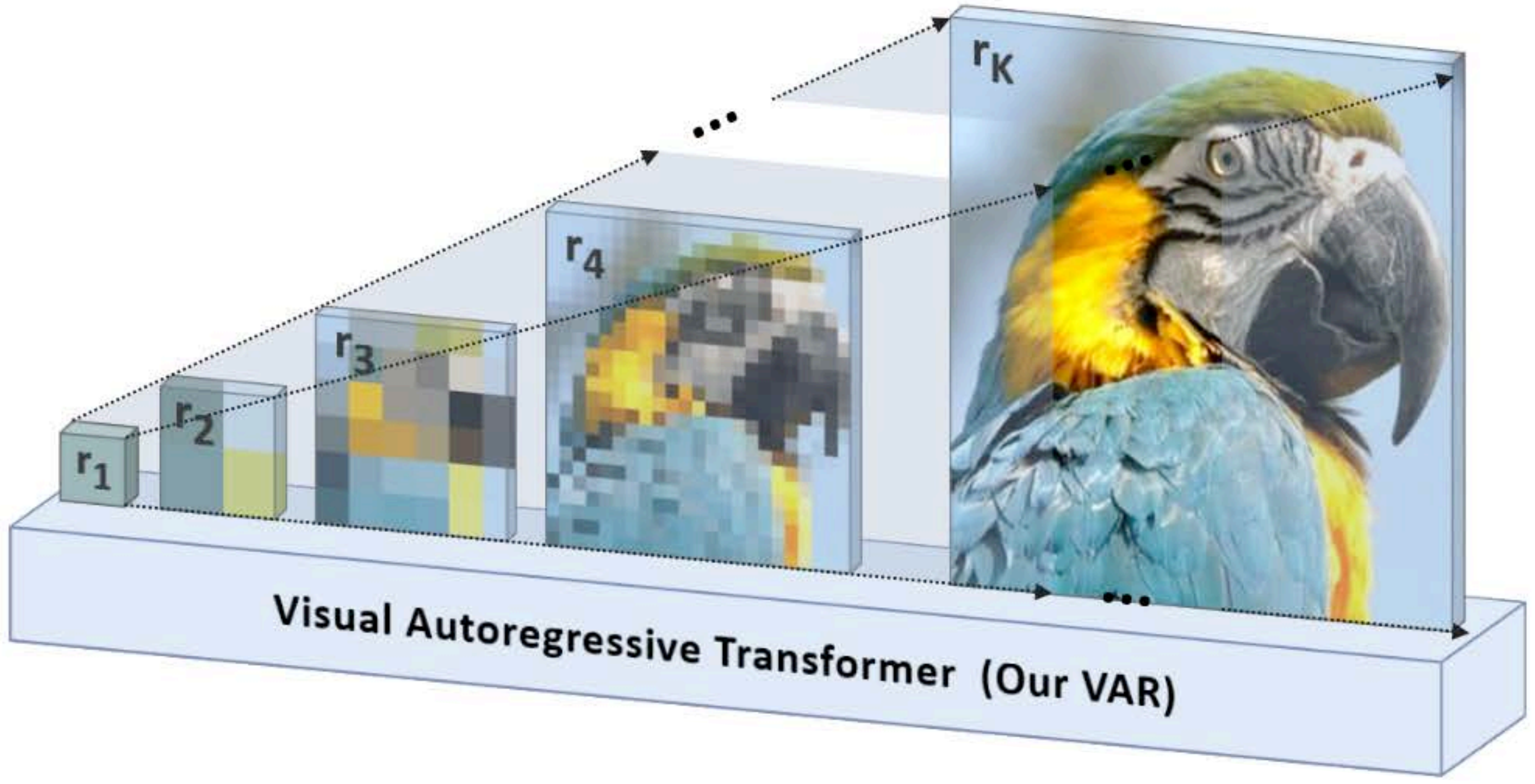
Autoregressive models for images

Chen et al, PMLR '20, Esser et al, 2012.09841



Next pixel (patch) prediction

Reed et al, 1703.03664
Tian et al, 2404.02905, Li et al, 2502.17437



Next **scale** prediction

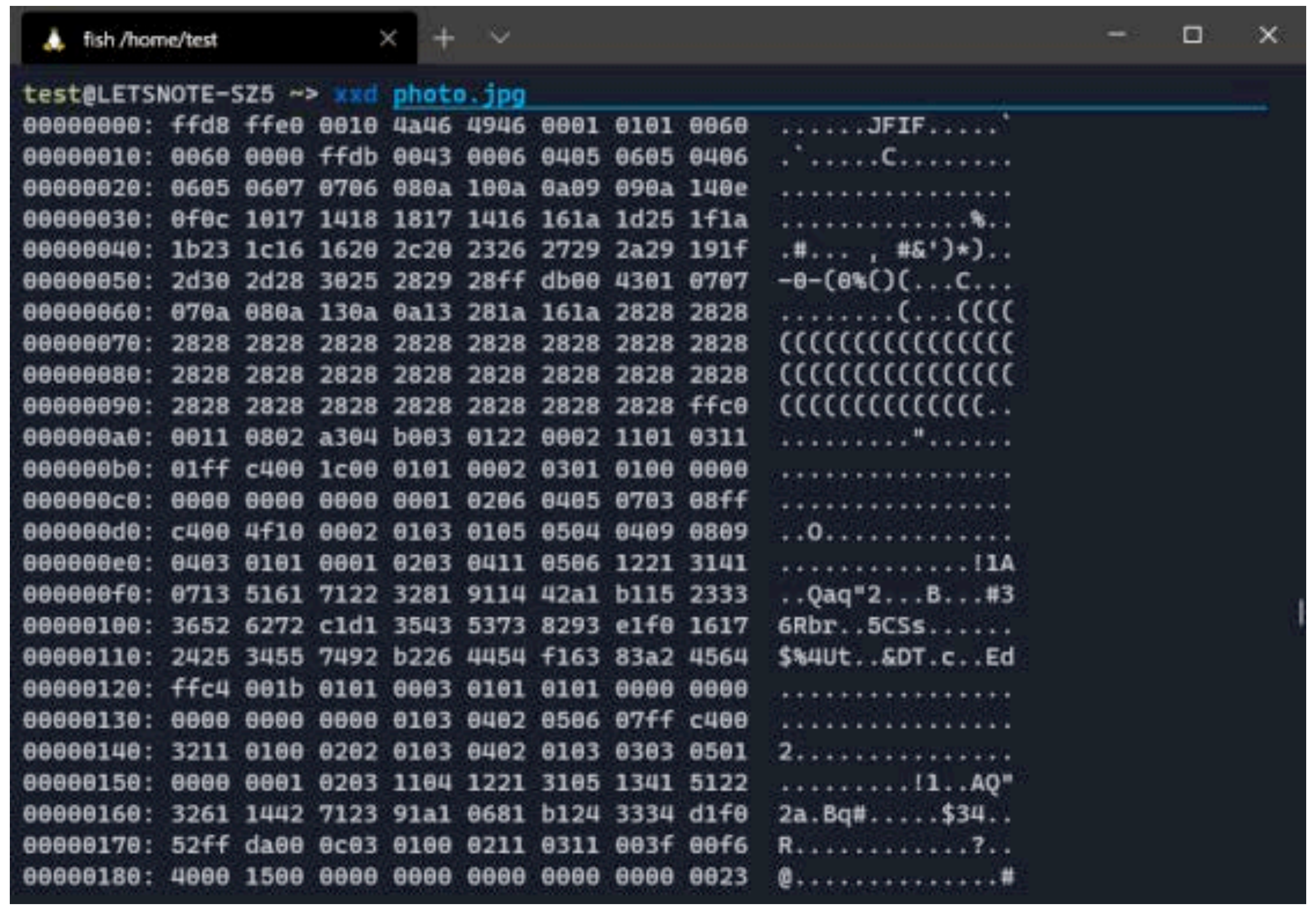
What is the suitable 1D ordering of 2D images ?

Autoregressive model for images

Han et al, 2408.08459

"Language" => token sequence => bitstream => **ANYTHING**

Compress



westlake.jpeg

Further compress

JPEG-LM

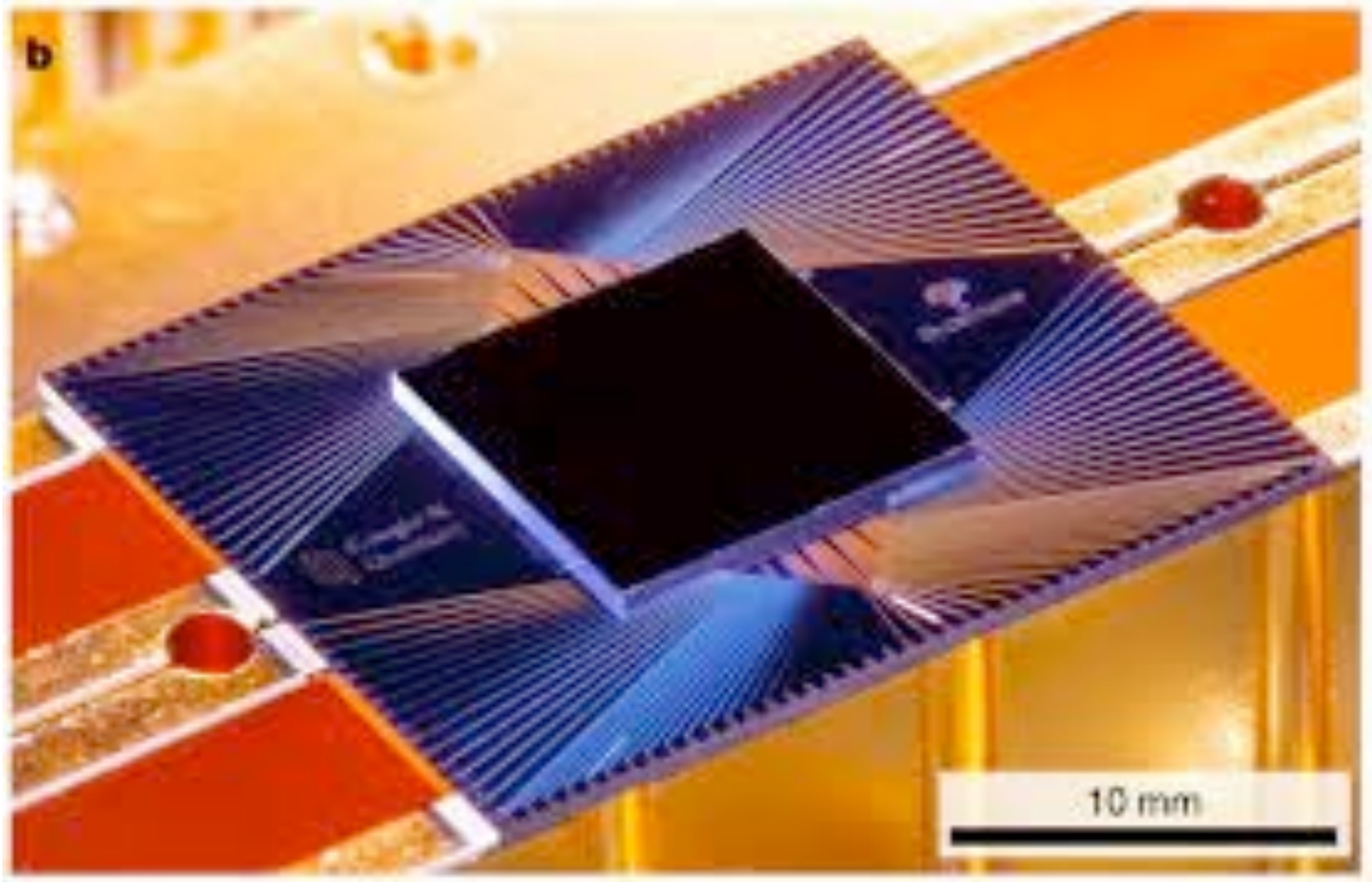
jpeg is a common lossy compression format for digital images

1) compute weights on predefined high-and-low frequency patches

2) throw away high-frequency weights; lossless compress low-frequency weights

Demo: Generative model of Sycamore data

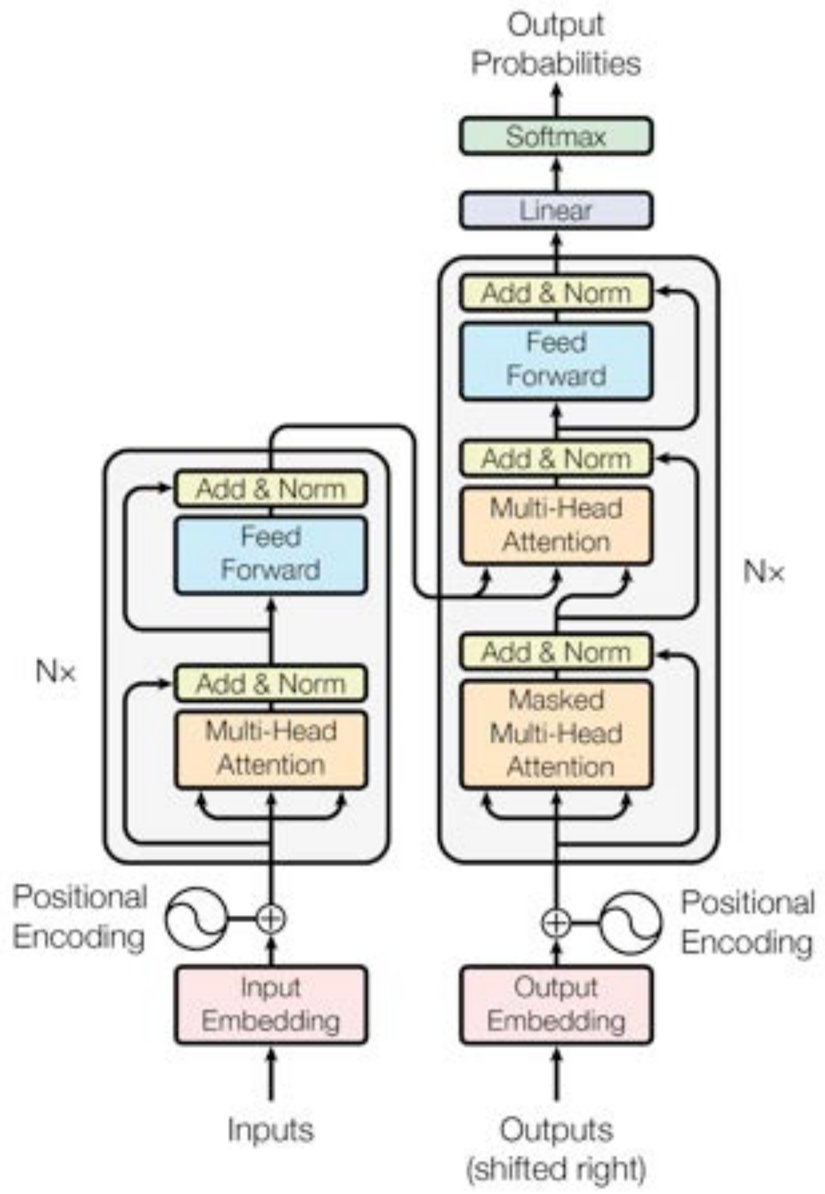
Quantum chip



bitstrings $\sim |\Psi(X)|^2$

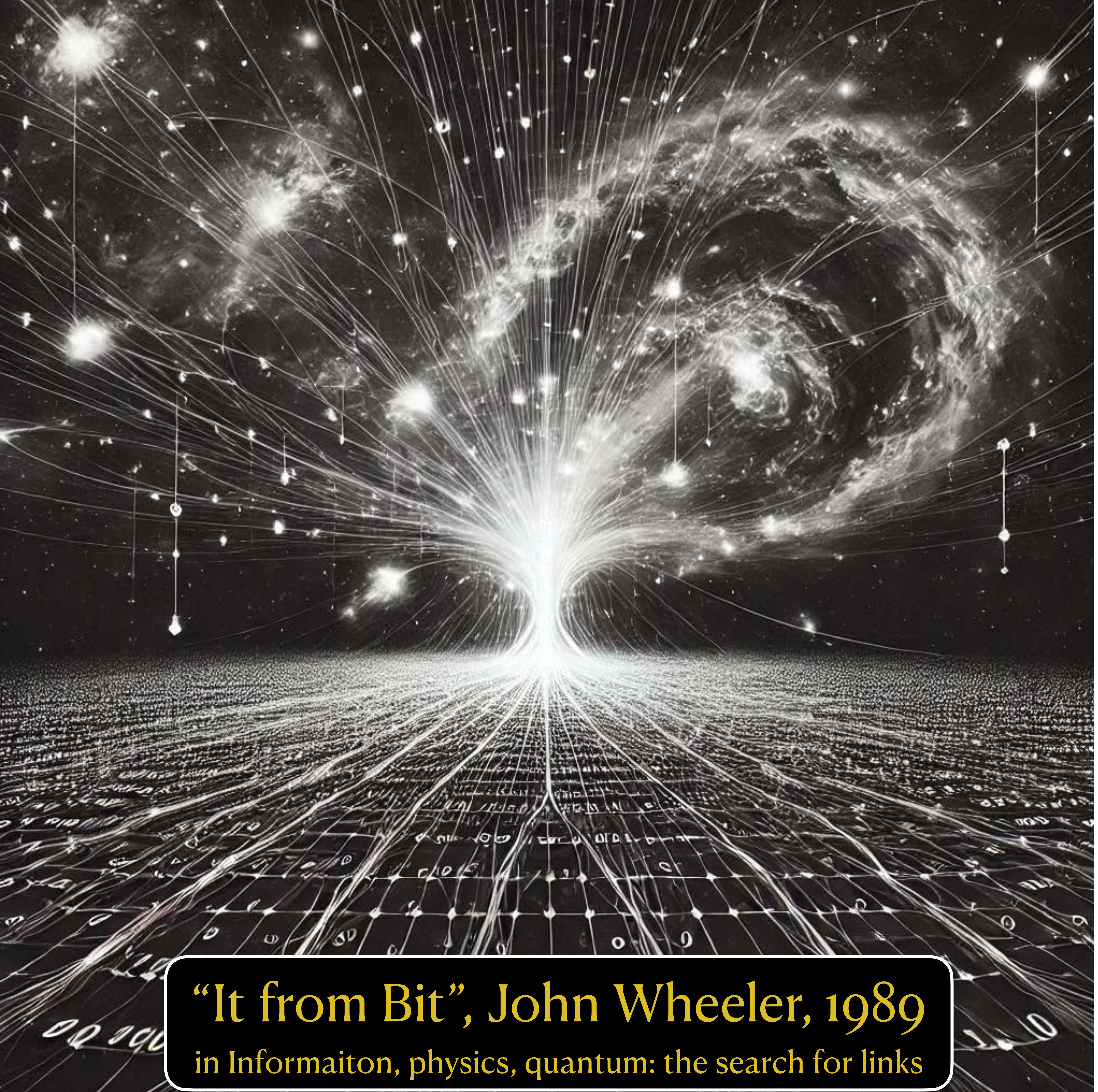
```
011110110100
100001111011
100110110111
100110100010
010100011000
010001000000
010101101100
100001111000
100101001001
001000001010
```

Transformer



Can we fake the measurement of the sycamore quantum circuit by training a transformer?

https://colab.research.google.com/drive/11WaroqULkudKT3h2i5J6r_EmA4wFKkoZ?usp=sharing



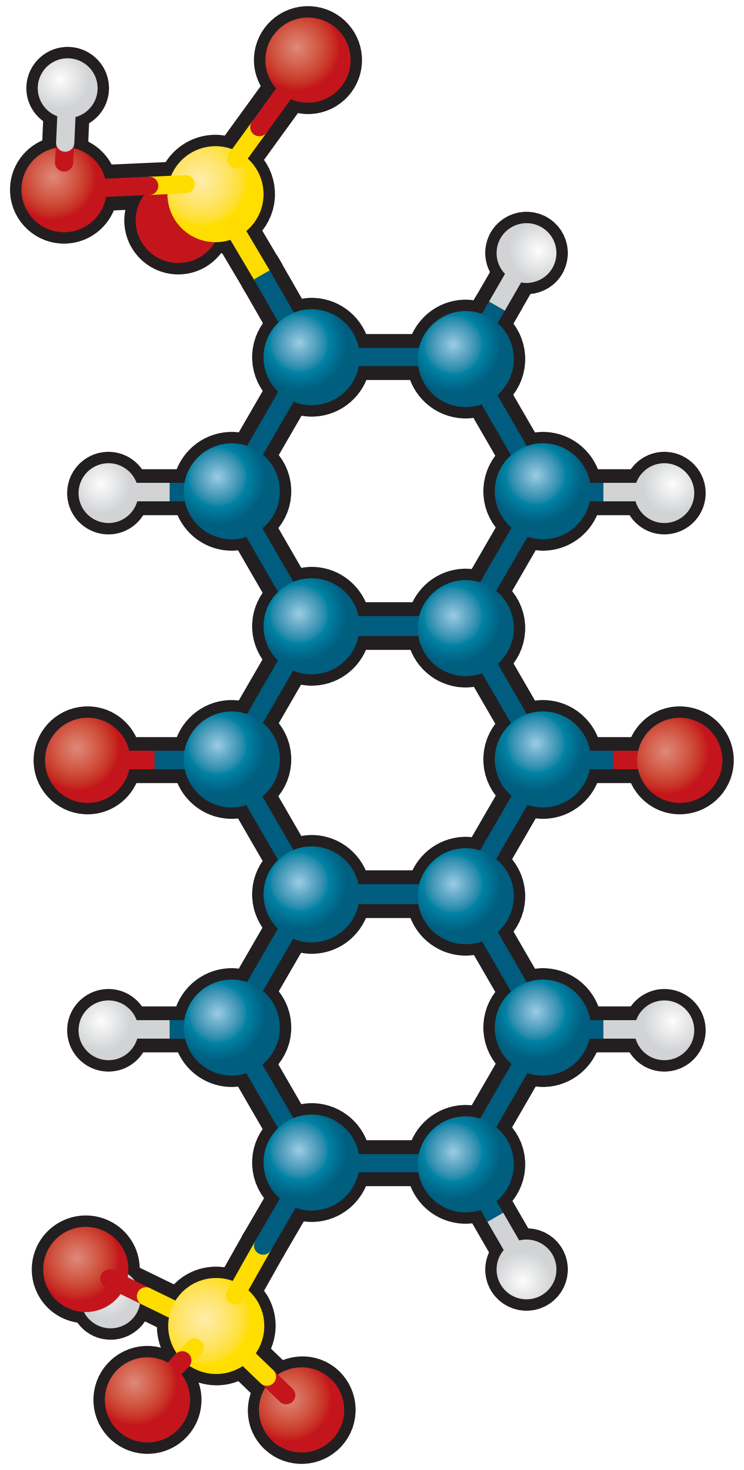
"It from Bit", John Wheeler, 1989

in Informaiton, physics, quantum: the search for links

Generative AI for It

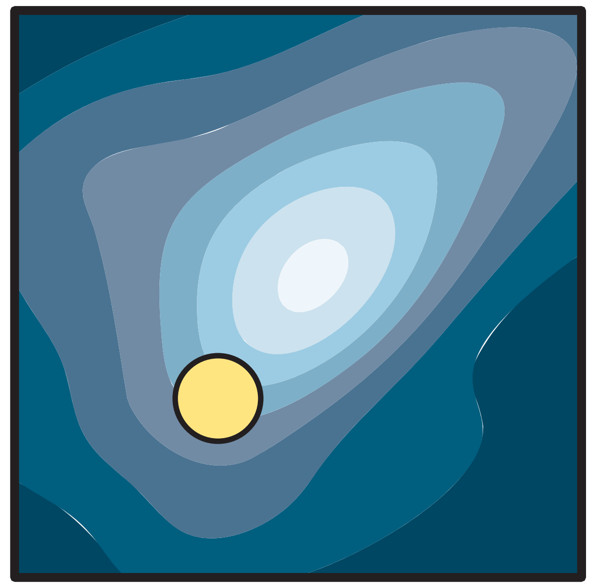
Generative AI for matter engineering

Chemical space X



Prediction $p(y \mid \boldsymbol{X})$

Generation $p(\boldsymbol{X} \mid y)$

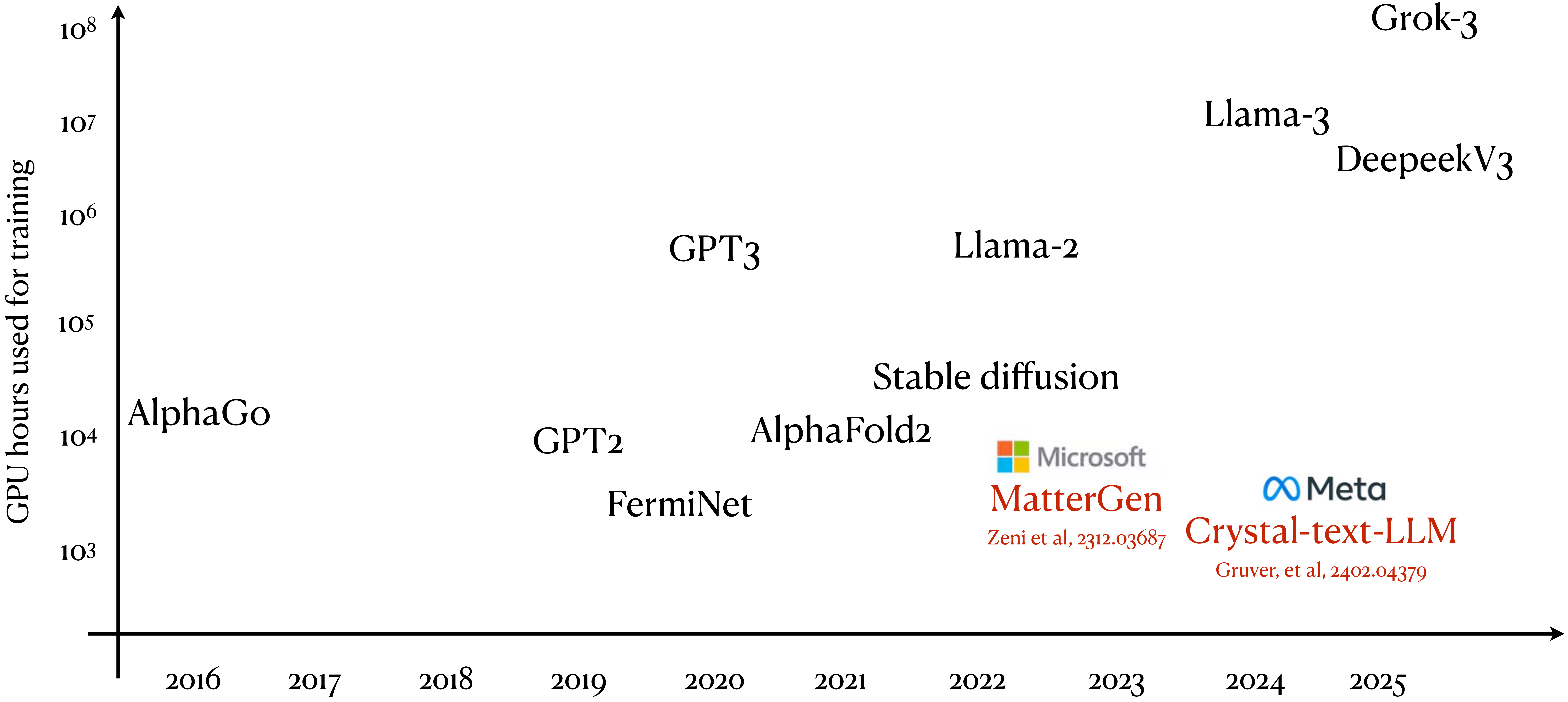


Control variable y

Inverse molecular design using machine learning, Sanchez-Lengeling & Aspuru-Guzik, Science '18
Inverse design in search of materials with target functionalities, Zunger, Nature Reviews Chemistry '18

"an image of beautiful crystals in 16:9"

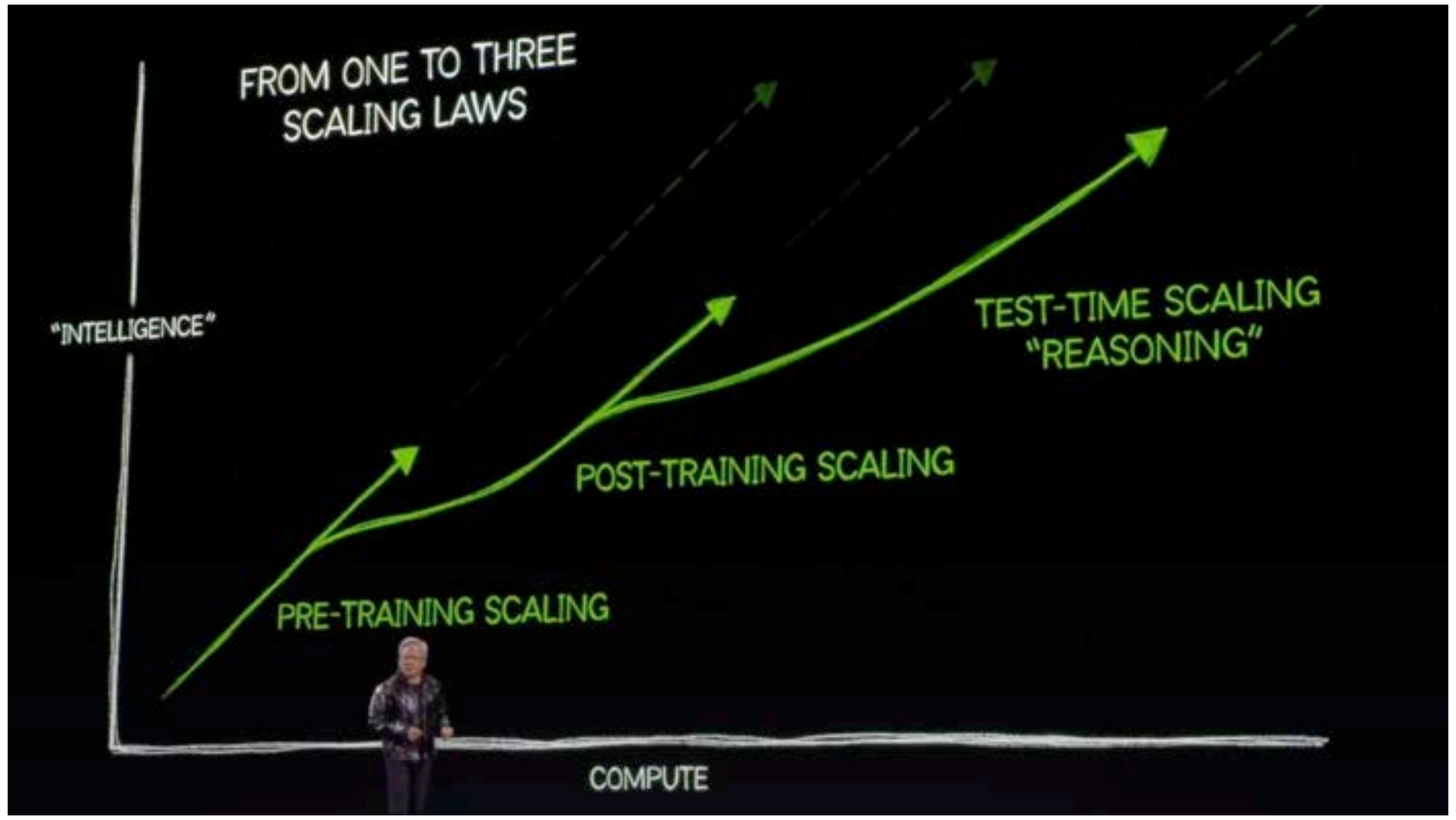
$$\text{pixels} \sim p(\text{pixels} \mid \text{texts})$$

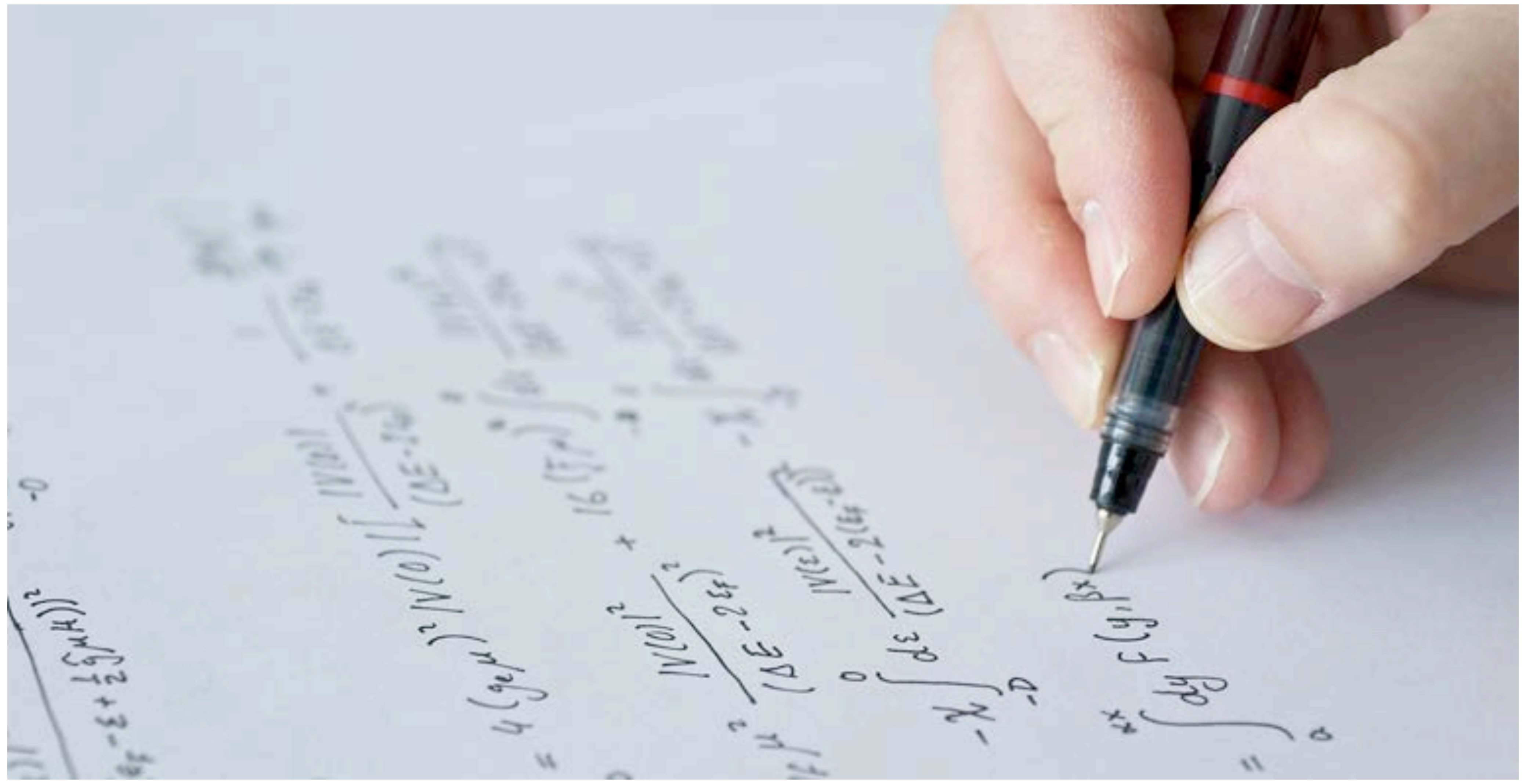


Is there a bitter lesson ?

"The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective"

—Rich Sutton 2019





more data and compute

more physics and symmetries

How are crystals different from languages/images/proteins ?

https://crystalsymmetry.files.wordpress.com/2014/08/poster_230_the_space_group_list_project.png

We have much less crystal data

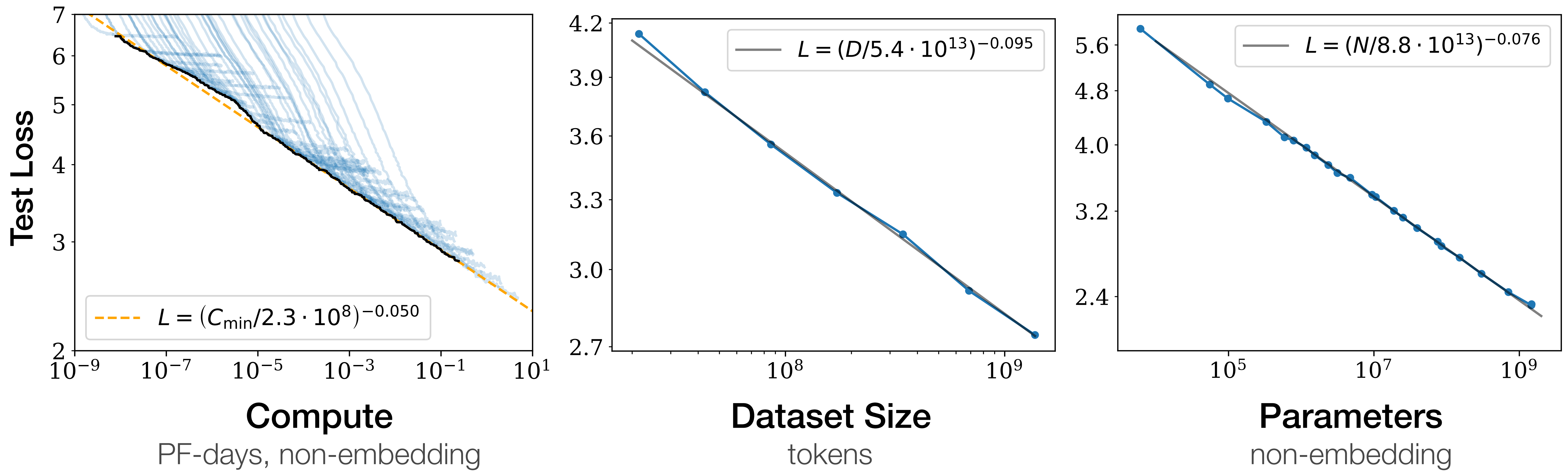
COMMON CRAWL

ICSD

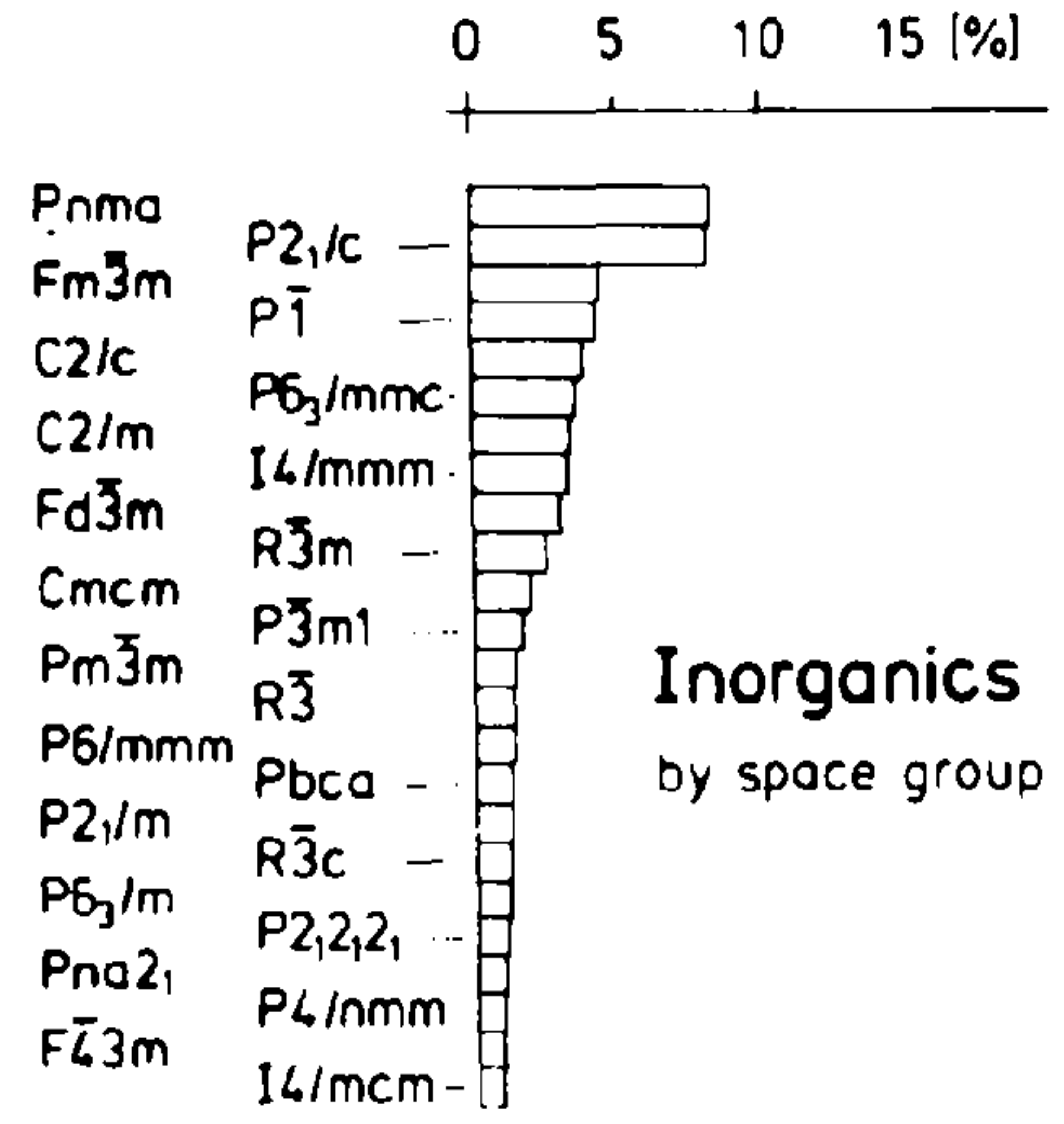
Over 250 billion pages

> 291,000 crystal structures

Data, compute, and parameters need to scale simultaneously Kaplan et al, 2001.08361



Space groups quantify Nature's preference over symmetry



Wyckoff Positions of Group *P*1 (No. 1)

Multiplicity	Wyckoff letter	Site symmetry	Coordinates
1	a	1	(x,y,z)

P1 is rare!

...

Wyckoff Positions of Group *Ia-3d* (No. 230)

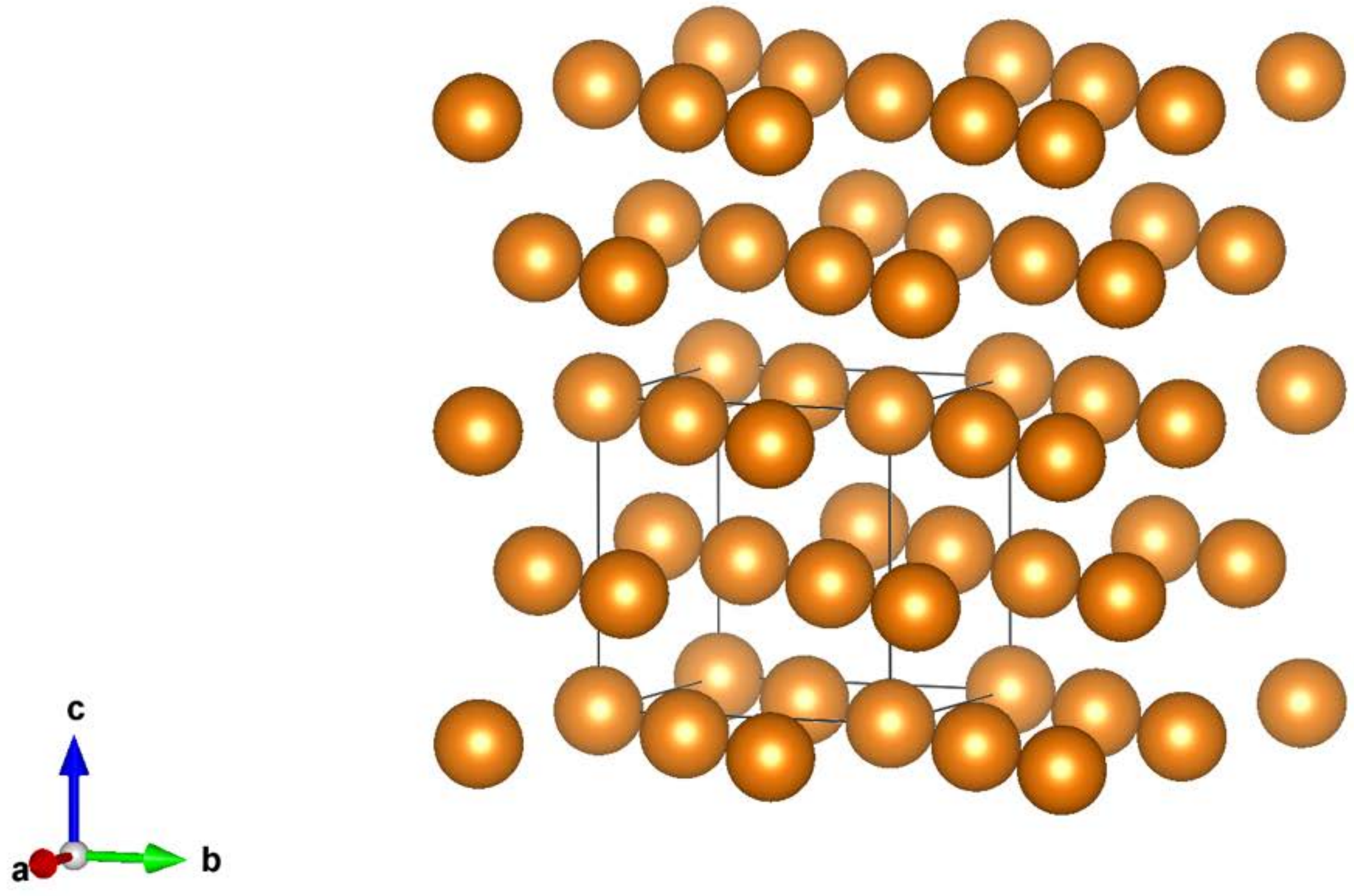
Multiplicity	Wyckoff letter	Site symmetry	Coordinates
			(0,0,0) + (1/2,1/2,1/2) +
96	h	1	(x,y,z) (-x+1/2,-y,z+1/2) (-x,y+1/2,-z+1/2) (x+1/2,-y+1/2,-z) (z,x,y) (z+1/2,-x+1/2,-y) (-z+1/2,-x,y+1/2) (-z,x+1/2,-y+1/2) (y,z,x) (-y,z+1/2,-x+1/2) (y+1/2,-z+1/2,-x) (-y+1/2,-z,x+1/2) (y+3/4,x+1/4,-z+1/4) (-y+3/4,-x+3/4,-z+3/4) (y+1/4,-x+1/4,z+3/4) (-y+1/4,x+3/4,z+1/4) (x+3/4,z+1/4,-y+1/4) (-x+1/4,z+3/4,y+1/4) (-x+3/4,-z+3/4,-y+3/4) (x+1/4,-z+1/4,y+3/4) (z+3/4,y+1/4,-x+1/4) (z+1/4,-y+1/4,x+3/4) (-z+1/4,y+3/4,x+1/4) (-z+3/4,-y+3/4,-x+3/4) (-x,-y,-z) (x+1/2,y,-z+1/2) (x,-y+1/2,z+1/2) (-x+1/2,y+1/2,z) (-z,-x,-y) (-z+1/2,x+1/2,y) (z+1/2,x,-y+1/2) (z,-x+1/2,y+1/2) (-y,-z,-x) (y,-z+1/2,x+1/2) (-y+1/2,z+1/2,x) (y+1/2,z,-x+1/2) (-y+1/4,-x+3/4,z+3/4) (y+1/4,x+1/4,z+1/4) (-y+3/4,x+3/4,-z+1/4) (y+3/4,-x+1/4,-z+3/4) (-x+1/4,-z+3/4,y+3/4) (x+3/4,-z+1/4,-y+3/4) (x+1/4,z+1/4,y+1/4) (-x+3/4,z+3/4,-y+1/4) (-z+1/4,-y+3/4,x+3/4) (-z+3/4,y+3/4,-x+1/4) (z+3/4,-y+1/4,-x+3/4) (z+1/4,y+1/4,x+1/4)
48	g	..2	(1/8,y,-y+1/4) (3/8,-y,-y+3/4) (7/8,y+1/2,y+1/4) (5/8,-y+1/2,y+3/4) (-y+1/4,1/8,y) (-y+3/4,3/8,-y) (y+1/4,7/8,y+1/2) (y+3/4,5/8,-y+1/2) (y,-y+1/4,1/8) (-y,-y+3/4,3/8) (y+1/2,y+1/4,7/8) (-y+1/2,y+3/4,5/8) (7/8,-y,y+3/4) (5/8,y,y+1/4) (1/8,-y+1/2,-y+3/4) (3/8,y+1/2,-y+1/4) (y+3/4,7/8,-y) (y+1/4,5/8,y) (-y+3/4,1/8,-y+1/2) (-y+1/4,3/8,y+1/2) (-y,y+3/4,7/8) (y,y+1/4,5/8) (-y+1/2,-y+3/4,1/8) (y+1/2,-y+1/4,3/8)
48	f	2..	(x,0,1/4) (-x+1/2,0,3/4) (1/4,x,0) (3/4,-x+1/2,0) (0,1/4,x) (0,3/4,-x+1/2) (3/4,x+1/4,0) (3/4,-x+3/4,1/2) (x+3/4,1/2,1/4) (-x+1/4,0,1/4) (0,1/4,-x+1/4) (1/2,1/4,x+3/4) (-x,0,3/4) (x+1/2,0,1/4) (3/4,-x,0) (1/4,x+1/2,0) (0,3/4,-x) (0,1/4,x+1/2) (1/4,-x+3/4,0) (1/4,x+1/4,1/2) (-x+1/4,1/2,3/4) (x+3/4,0,3/4) (0,3/4,x+3/4) (1/2,3/4,-x+1/4)
32	e	.3.	(x,x,x) (-x+1/2,-x,x+1/2) (-x,x+1/2,-x+1/2) (x+1/2,-x+1/2,-x) (x+3/4,x+1/4,-x+1/4) (-x+3/4,-x+3/4,-x+3/4) (x+1/4,-x+1/4,x+3/4) (-x+1/4,x+3/4,x+1/4) (-x,-x,-x) (x+1/2,x,-x+1/2) (x,-x+1/2,x+1/2) (-x+1/2,x+1/2,x) (-x+1/4,-x+3/4,x+3/4) (x+1/4,x+1/4,x+1/4) (-x+3/4,x+3/4,-x+1/4) (x+3/4,-x+1/4,-x+3/4)
24	d	-4..	(3/8,0,1/4) (1/8,0,3/4) (1/4,3/8,0) (3/4,1/8,0) (0,1/4,3/8) (0,3/4,1/8) (3/4,5/8,0) (3/4,3/8,1/2) (1/8,1/2,1/4) (7/8,0,1/4) (0,1/4,7/8) (1/2,1/4,1/8)
24	c	2.2 2	(1/8,0,1/4) (3/8,0,3/4) (1/4,1/8,0) (3/4,3/8,0) (0,1/4,1/8) (0,3/4,3/8) (7/8,0,3/4) (5/8,0,1/4) (3/4,7/8,0) (1/4,5/8,0) (0,3/4,7/8) (0,1/4,5/8)
16	b	.32	(1/8,1/8,1/8) (3/8,7/8,5/8) (7/8,5/8,3/8) (5/8,3/8,7/8) (7/8,7/8,7/8) (5/8,1/8,3/8) (1/8,3/8,5/8) (3/8,5/8,1/8)
16	a	.-3.	(0,0,0) (1/2,0,1/2) (0,1/2,1/2) (1/2,1/2,0) (3/4,1/4,1/4) (3/4,3/4,3/4) (1/4,1/4,3/4) (1/4,3/4,1/4)

https://www.cryst.ehu.es/cryst/get_wp.html

Wyckoff Positions of Group *Fm*-3*m* (No. 225)

Multiplicity	Wyckoff letter	Site symmetry	Coordinates (0,0,0) + (0,1/2,1/2) + (1/2,0,1/2) + (1/2,1/2,0) +
192	l	1	(x,y,z) (-x,-y,z) (-x,y,-z) (x,-y,-z) (z,x,y) (z,-x,-y) (-z,-x,y) (-z,x,-y) (y,z,x) (-y,z,-x) (y,-z,-x) (-y,-z,x) (y,x,-z) (-y,-x,-z) (y,-x,z) (-y,x,z) (x,z,-y) (-x,z,y) (-x,-z,-y) (x,-z,y) (z,y,-x) (z,-y,x) (-z,y,x) (-z,-y,-x) (-x,-y,-z) (x,y,-z) (x,-y,z) (-x,y,z) (-z,-x,-y) (-z,x,y) (z,x,-y) (z,-x,y) (-y,-z,-x) (y,-z,x) (-y,z,x) (y,z,-x) (-y,-x,z) (y,x,z) (-y,x,-z) (y,-x,-z) (-x,-z,y) (x,-z,-y) (x,z,y) (-x,z,-y) (-z,-y,x) (-z,y,-x) (z,-y,-x) (z,y,x)
96	k	..m	(x,x,z) (-x,-x,z) (-x,x,-z) (x,-x,-z) (z,x,x) (z,-x,-x) (-z,-x,x) (-z,x,-x) (x,z,x) (-x,z,-x) (x,-z,-x) (-x,-z,x) (x,x,-z) (-x,-x,-z) (x,-x,z) (-x,x,z) (x,z,-x) (-x,z,x) (-x,-z,-x) (x,-z,x) (z,x,-x) (z,-x,x) (-z,x,x) (-z,-x,-x)
96	j	m..	(0,y,z) (0,-y,z) (0,y,-z) (0,-y,-z) (z,0,y) (z,0,-y) (-z,0,y) (-z,0,-y) (y,z,0) (-y,z,0) (y,-z,0) (-y,-z,0) (y,0,-z) (-y,0,-z) (y,0,z) (-y,0,z) (0,z,-y) (0,z,y) (0,-z,-y) (0,-z,y) (z,y,0) (z,-y,0) (-z,y,0) (-z,-y,0)
48	i	m.m 2	(1/2,y,y) (1/2,-y,y) (1/2,y,-y) (1/2,-y,-y) (y,1/2,y) (y,1/2,-y) (-y,1/2,y) (-y,1/2,-y) (y,y,1/2) (-y,y,1/2) (y,-y,1/2) (-y,-y,1/2)
48	h	m.m 2	(0,y,y) (0,-y,y) (0,y,-y) (0,-y,-y) (y,0,y) (y,0,-y) (-y,0,y) (-y,0,-y) (y,y,0) (-y,y,0) (y,-y,0) (-y,-y,0)
48	g	2.m m	(x,1/4,1/4) (-x,3/4,1/4) (1/4,x,1/4) (1/4,-x,3/4) (1/4,1/4,x) (3/4,1/4,-x) (1/4,x,3/4) (3/4,-x,3/4) (x,1/4,3/4) (-x,1/4,1/4) (1/4,1/4,-x) (1/4,3/4,x)
32	f	.3m	(x,x,x) (-x,-x,x) (-x,x,-x) (x,-x,-x) (x,x,-x) (-x,-x,-x) (x,-x,x) (-x,x,x)
24	e	4m. m	(x,0,0) (-x,0,0) (0,x,0) (0,-x,0) (0,0,x) (0,0,-x)
24	d	m.m m	(0,1/4,1/4) (0,3/4,1/4) (1/4,0,1/4) (1/4,0,3/4) (1/4,1/4,0) (3/4,1/4,0)
8	c	-43m	(1/4,1/4,1/4) (1/4,1/4,3/4)
4	b	m-3m	(1/2,1/2,1/2)
4	a	m-3m	(0,0,0)

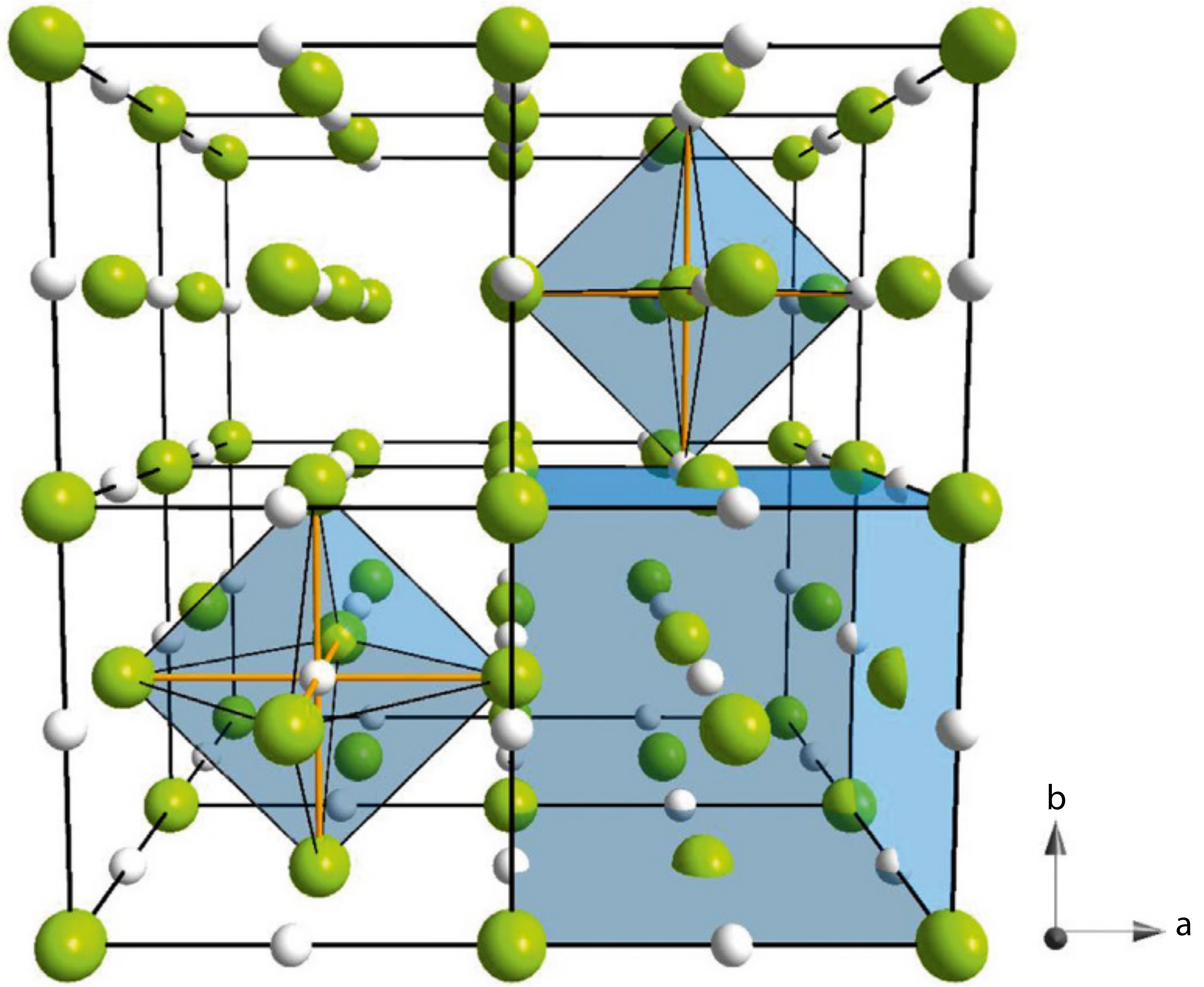
Copper



Wyckoff Positions of Group *Fm*-3*m* (No. 225)

Multiplicity	Wyckoff letter	Site symmetry	Coordinates (0,0,0) + (0,1/2,1/2) + (1/2,0,1/2) + (1/2,1/2,0) +
192	l	1	(x,y,z) (-x,-y,z) (-x,y,-z) (x,-y,-z) (z,x,y) (z,-x,-y) (-z,-x,y) (-z,x,-y) (y,z,x) (-y,z,-x) (y,-z,-x) (-y,-z,x) (y,x,-z) (-y,-x,-z) (y,-x,z) (-y,x,z) (x,z,-y) (-x,z,y) (-x,-z,-y) (x,-z,y) (z,y,-x) (z,-y,x) (-z,y,x) (-z,-y,-x) (-x,-y,-z) (x,y,-z) (x,-y,z) (-x,y,z) (-z,-x,-y) (-z,x,y) (z,x,-y) (z,-x,y) (-y,-z,-x) (y,-z,x) (-y,z,x) (y,z,-x) (-y,-x,z) (y,x,z) (-y,x,-z) (y,-x,-z) (-x,-z,y) (x,-z,-y) (x,z,y) (-x,z,-y) (-z,-y,x) (-z,y,-x) (z,-y,-x) (z,y,x)
96	k	..m	(x,x,z) (-x,-x,z) (-x,x,-z) (x,-x,-z) (z,x,x) (z,-x,-x) (-z,-x,x) (-z,x,-x) (x,z,x) (-x,z,-x) (x,-z,-x) (-x,-z,x) (x,x,-z) (-x,-x,-z) (x,-x,z) (-x,x,z) (x,z,-x) (-x,z,x) (-x,-z,-x) (x,-z,x) (z,x,-x) (z,-x,x) (-z,x,x) (-z,-x,-x)
96	j	m..	(0,y,z) (0,-y,z) (0,y,-z) (0,-y,-z) (z,0,y) (z,0,-y) (-z,0,y) (-z,0,-y) (y,z,0) (-y,z,0) (y,-z,0) (-y,-z,0) (y,0,-z) (-y,0,-z) (y,0,z) (-y,0,z) (0,z,-y) (0,z,y) (0,-z,-y) (0,-z,y) (z,y,0) (z,-y,0) (-z,y,0) (-z,-y,0)
48	i	m.m 2	(1/2,y,y) (1/2,-y,y) (1/2,y,-y) (1/2,-y,-y) (y,1/2,y) (y,1/2,-y) (-y,1/2,y) (-y,1/2,-y) (y,y,1/2) (-y,y,1/2) (y,-y,1/2) (-y,-y,1/2)
48	h	m.m 2	(0,y,y) (0,-y,y) (0,y,-y) (0,-y,-y) (y,0,y) (y,0,-y) (-y,0,y) (-y,0,-y) (y,y,0) (-y,y,0) (y,-y,0) (-y,-y,0)
48	g	2.m m	(x,1/4,1/4) (-x,3/4,1/4) (1/4,x,1/4) (1/4,-x,3/4) (1/4,1/4,x) (3/4,1/4,-x) (1/4,x,3/4) (3/4,-x,3/4) (x,1/4,3/4) (-x,1/4,1/4) (1/4,1/4,-x) (1/4,3/4,x)
32	f	.3m	(x,x,x) (-x,-x,x) (-x,x,-x) (x,-x,-x) (x,x,-x) (-x,-x,-x) (x,-x,x) (-x,x,x)
24	e	4m. m	(x,0,0) (-x,0,0) (0,x,0) (0,-x,0) (0,0,x) (0,0,-x)
24	d	m.m m	(0,1/4,1/4) (0,3/4,1/4) (1/4,0,1/4) (1/4,0,3/4) (1/4,1/4,0) (3/4,1/4,0)
8	c	-43m	(1/4,1/4,1/4) (1/4,1/4,3/4)
4	b	m-3m	(1/2,1/2,1/2)
4	a	m-3m	(0,0,0)

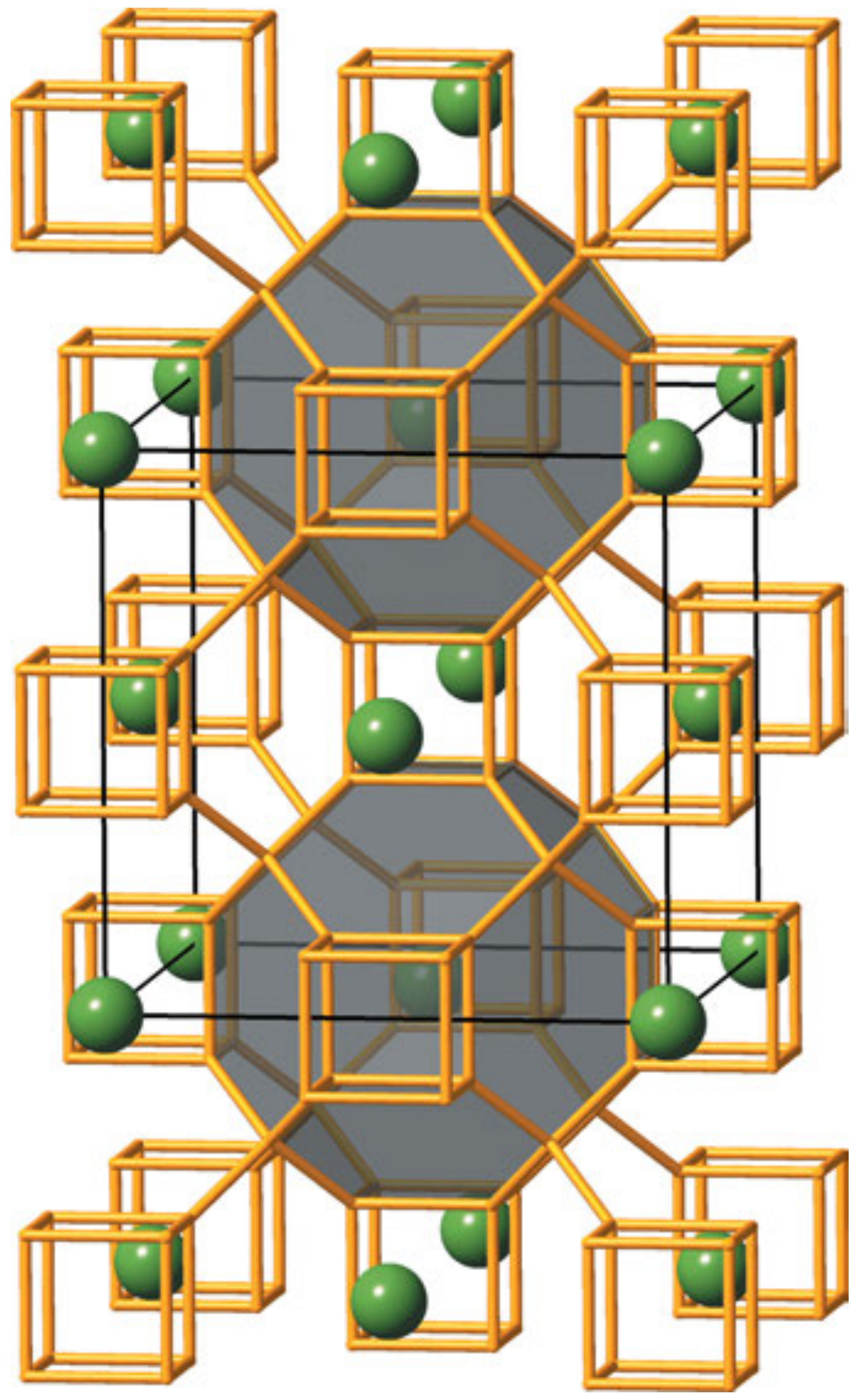
NaCl



Wyckoff Positions of Group *Fm*-3*m* (No. 225)

Multiplicity	Wyckoff letter	Site symmetry	Coordinates (0,0,0) + (0,1/2,1/2) + (1/2,0,1/2) + (1/2,1/2,0) +
192	l	1	(x,y,z) (-x,-y,z) (-x,y,-z) (x,-y,-z) (z,x,y) (z,-x,-y) (-z,-x,y) (-z,x,-y) (y,z,x) (-y,z,-x) (y,-z,-x) (-y,-z,x) (y,x,-z) (-y,-x,-z) (y,-x,z) (-y,x,z) (x,z,-y) (-x,z,y) (-x,-z,-y) (x,-z,y) (z,y,-x) (z,-y,x) (-z,y,x) (-z,-y,-x) (-x,-y,-z) (x,y,-z) (x,-y,z) (-x,y,z) (-z,-x,-y) (-z,x,y) (z,x,-y) (z,-x,y) (-y,-z,-x) (y,-z,x) (-y,z,x) (y,z,-x) (-y,-x,z) (y,x,z) (-y,x,-z) (y,-x,-z) (-x,-z,y) (x,-z,-y) (x,z,y) (-x,z,-y) (-z,-y,x) (-z,y,-x) (z,-y,-x) (z,y,x)
96	k	..m	(x,x,z) (-x,-x,z) (-x,x,-z) (x,-x,-z) (z,x,x) (z,-x,-x) (-z,-x,x) (-z,x,-x) (x,z,x) (-x,z,-x) (x,-z,-x) (-x,-z,x) (x,x,-z) (-x,-x,-z) (x,-x,z) (-x,x,z) (x,z,-x) (-x,z,x) (-x,-z,-x) (x,-z,x) (z,x,-x) (z,-x,x) (-z,x,x) (-z,-x,-x)
96	j	m..	(0,y,z) (0,-y,z) (0,y,-z) (0,-y,-z) (z,0,y) (z,0,-y) (-z,0,y) (-z,0,-y) (y,z,0) (-y,z,0) (y,-z,0) (-y,-z,0) (y,0,-z) (-y,0,-z) (y,0,z) (-y,0,z) (0,z,-y) (0,z,y) (0,-z,-y) (0,-z,y) (z,y,0) (z,-y,0) (-z,y,0) (-z,-y,0)
48	i	m.m 2	(1/2,y,y) (1/2,-y,y) (1/2,y,-y) (1/2,-y,-y) (y,1/2,y) (y,1/2,-y) (-y,1/2,y) (-y,1/2,-y) (y,y,1/2) (-y,y,1/2) (y,-y,1/2) (-y,-y,1/2)
48	h	m.m 2	(0,y,y) (0,-y,y) (0,y,-y) (0,-y,-y) (y,0,y) (y,0,-y) (-y,0,y) (-y,0,-y) (y,y,0) (-y,y,0) (y,-y,0) (-y,-y,0)
48	g	2.m m	(x,1/4,1/4) (-x,3/4,1/4) (1/4,x,1/4) (1/4,-x,3/4) (1/4,1/4,x) (3/4,1/4,-x) (1/4,x,3/4) (3/4,-x,3/4) (x,1/4,3/4) (-x,1/4,1/4) (1/4,1/4,-x) (1/4,3/4,x)
32	f	.3m	(x,x,x) (-x,-x,x) (-x,x,-x) (x,-x,-x) (x,x,-x) (-x,-x,-x) (x,-x,x) (-x,x,x)
24	e	4m. m	(x,0,0) (-x,0,0) (0,x,0) (0,-x,0) (0,0,x) (0,0,-x)
24	d	m.m m	(0,1/4,1/4) (0,3/4,1/4) (1/4,0,1/4) (1/4,0,3/4) (1/4,1/4,0) (3/4,1/4,0)
8	c	-43m	(1/4,1/4,1/4) (1/4,1/4,3/4)
4	b	m-3m	(1/2,1/2,1/2)
4	a	m-3m	(0,0,0)

LaH_{10}

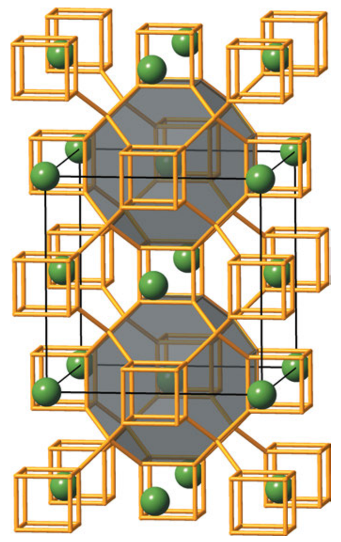


CrystalFormer

Zhendong Cao, Xiaoshan Luo, Jian Lv, and LW, 2403.15734

deepmodeling/CrystalFormer

Space Group Informed Transformer for Crystals



"225-a-La-0-0-0-c-H-1/4-1/4-1/4-f-H-0.375-0.375-0.375-X-5.1-5.1-5.1-90-90-90"

"Grammar" ~ Solid state chemistry

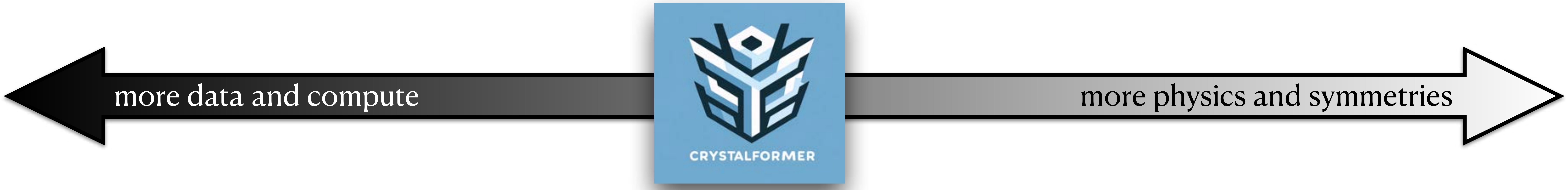
"Synonyms" ~ Elememt substitution

"Idioms" ~ Coordination polyhedra

"Rhythm" ~ Wyckoff positions

Nature's codebook for tokenization

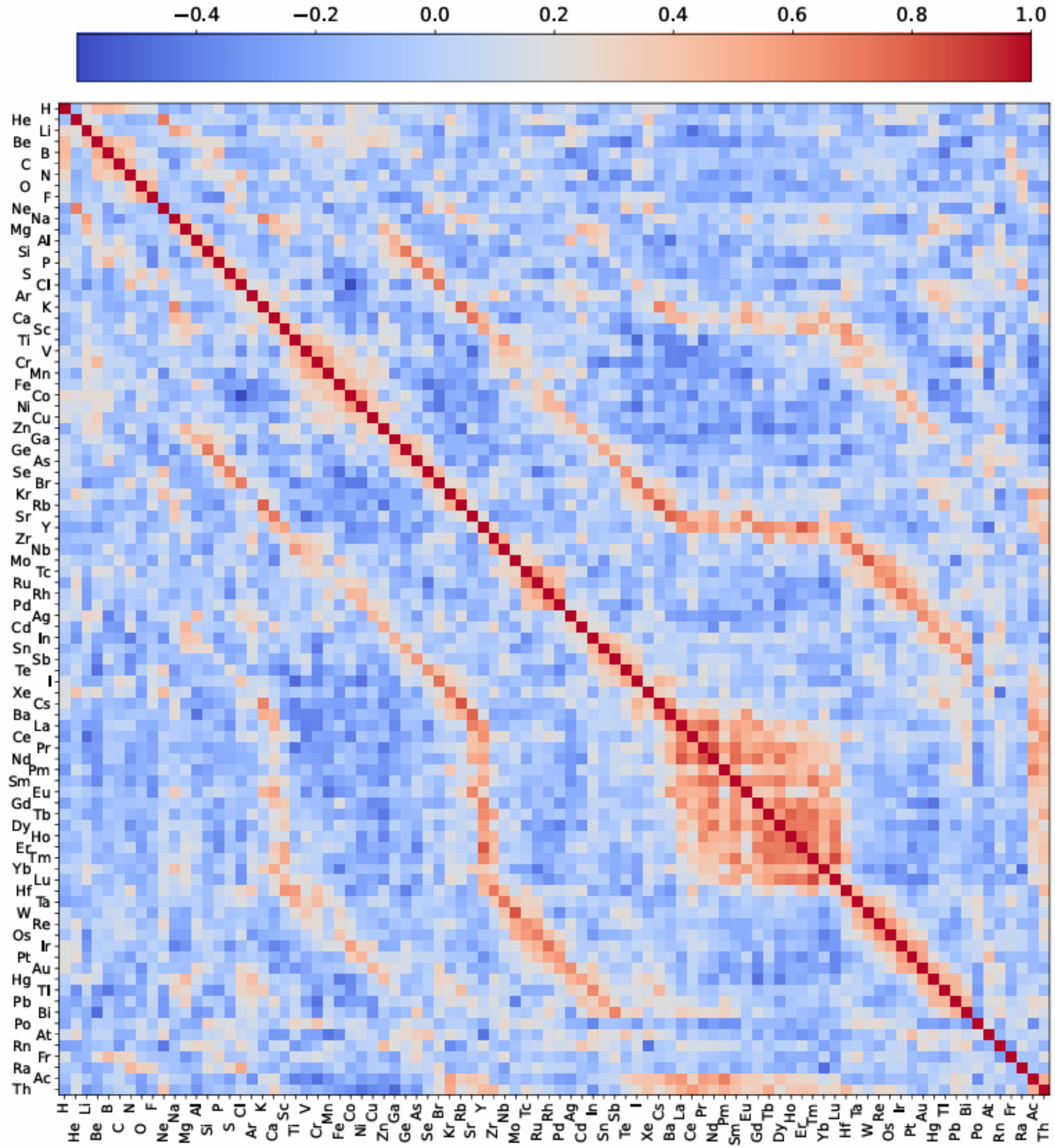
discrete, pre-compression

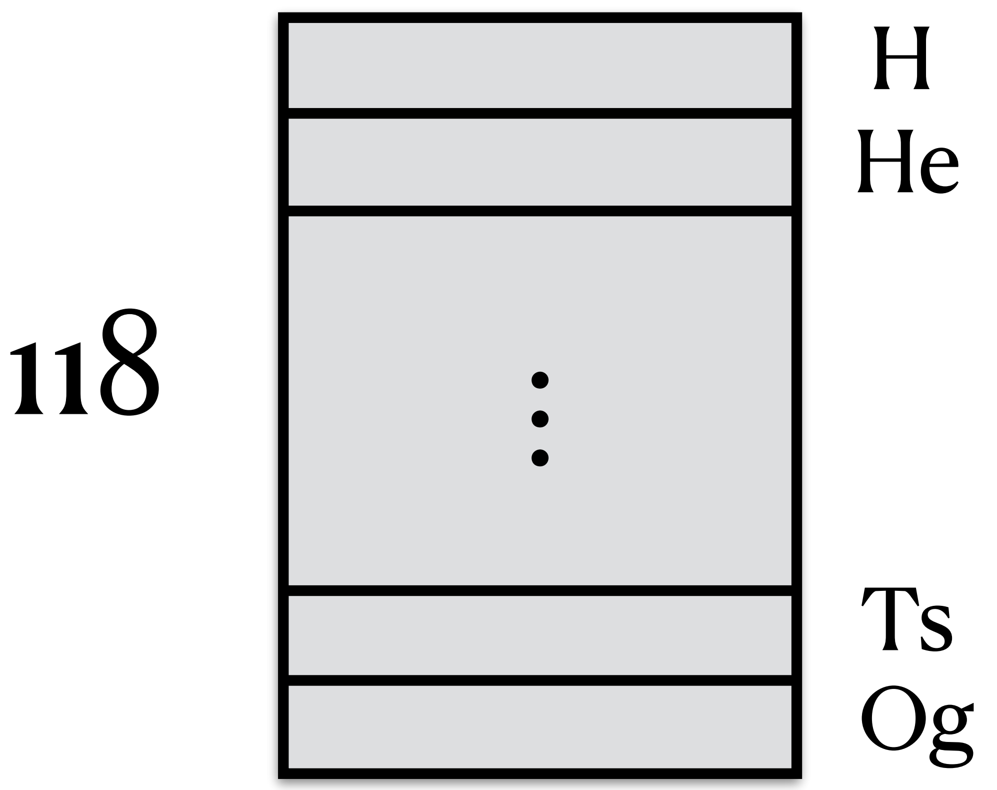


Not a large language model, **nor** a potential energy surface

Compress material database into transformer parameters

The model has to gain chemical intuition for such compression





Element embedding table

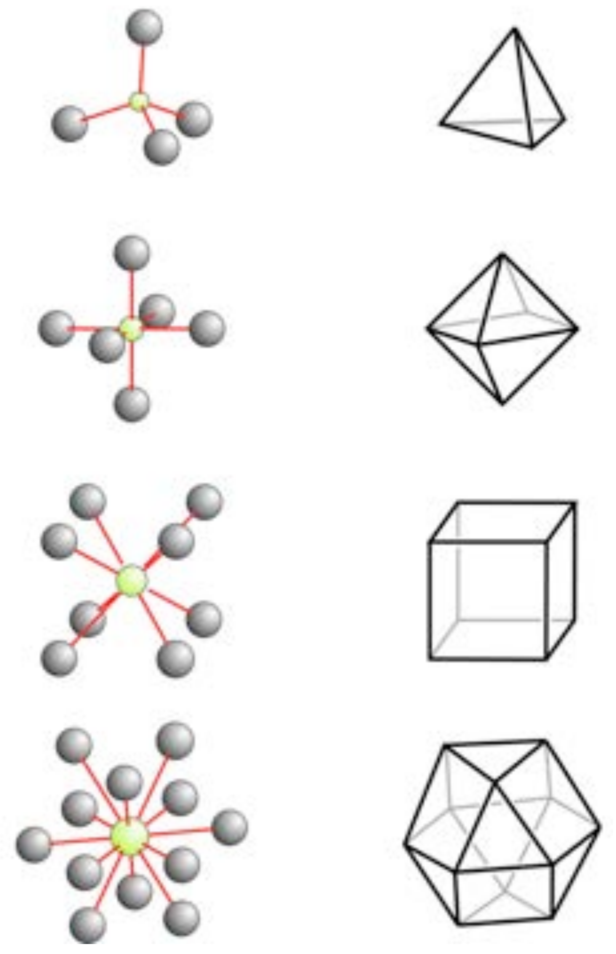
$$\frac{\boldsymbol{a} \cdot \boldsymbol{b}}{|\boldsymbol{a}| \cdot |\boldsymbol{b}|}$$

Cosine similarity

Solid state chemistry as "n-gram" in the crystal language

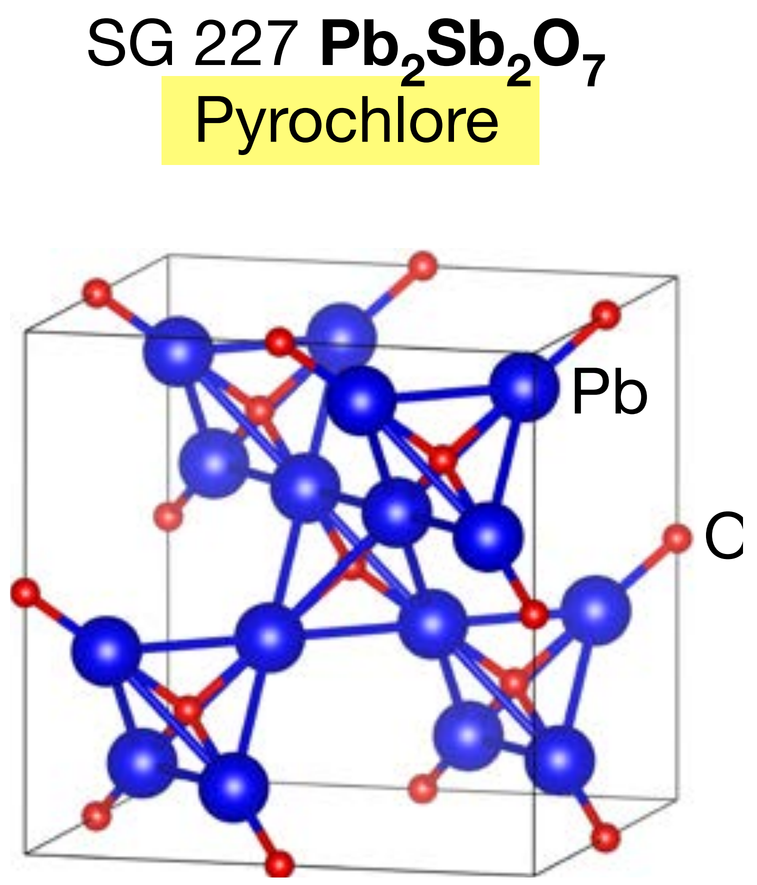
$$g\text{-}W_1\text{-}A_1\text{-}X_1\text{-}W_2\text{-}A_2\text{-}X_2\text{-}\cdots\text{-a-b-c-}\alpha\text{-}\beta\text{-}\gamma$$

Coordination polyhedra



Polyhedra in Chemistry
Gongdu Zhou 2009

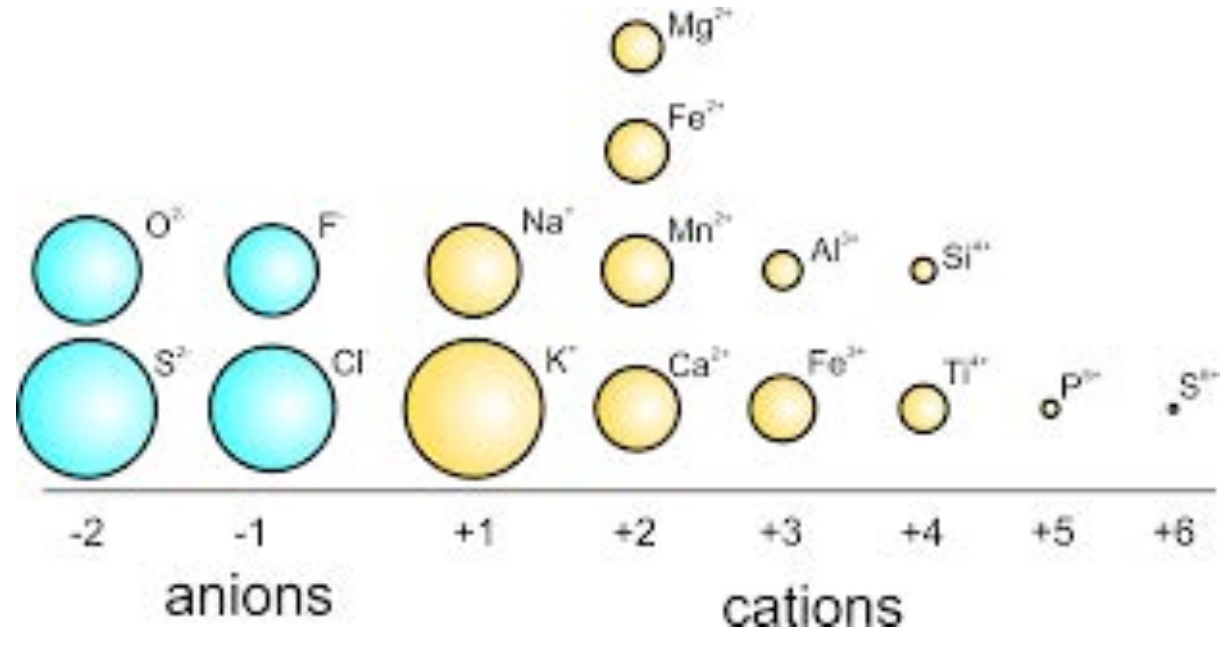
Lattices



Regnault et al, catalogue of
flat-band materials, Nature '22

Valence

"anions are in less symmetric
positions than cations"



Urusov and Nadezhina,
J. Struct. Chem. 2009

Crystals by intuition vs by minimization

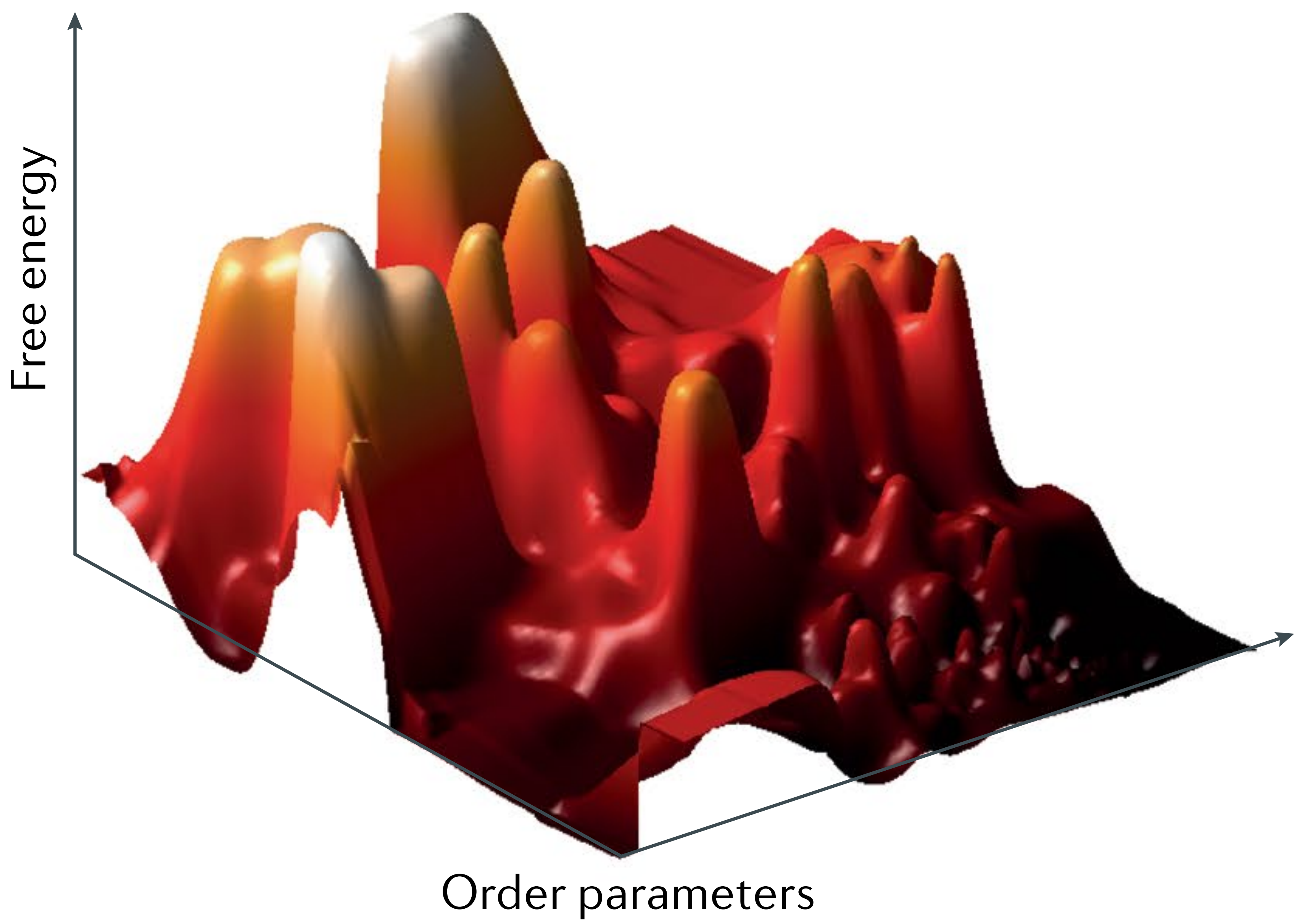
Data-driven (system 1)

Chemical intuitions (e.g. Pauling rules) from compression

CDVAE, Mattergen, Unimat, DiffCSP, CrystaLLM...

Physics-based (system 2)

energy minimization

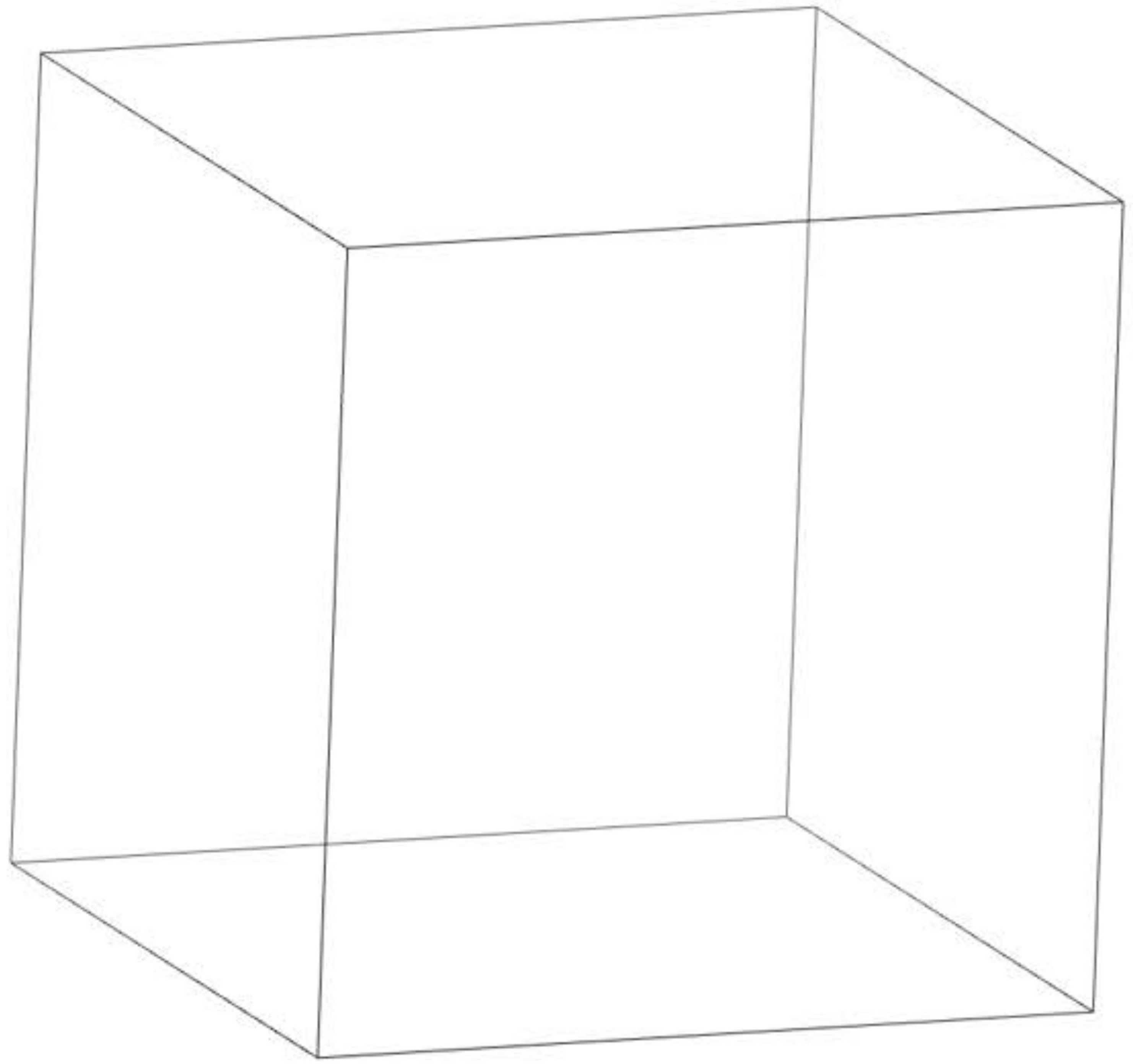


CALYPSO, USPEX, ARISS,...

DPA, MACE, LASP, GNoME,...

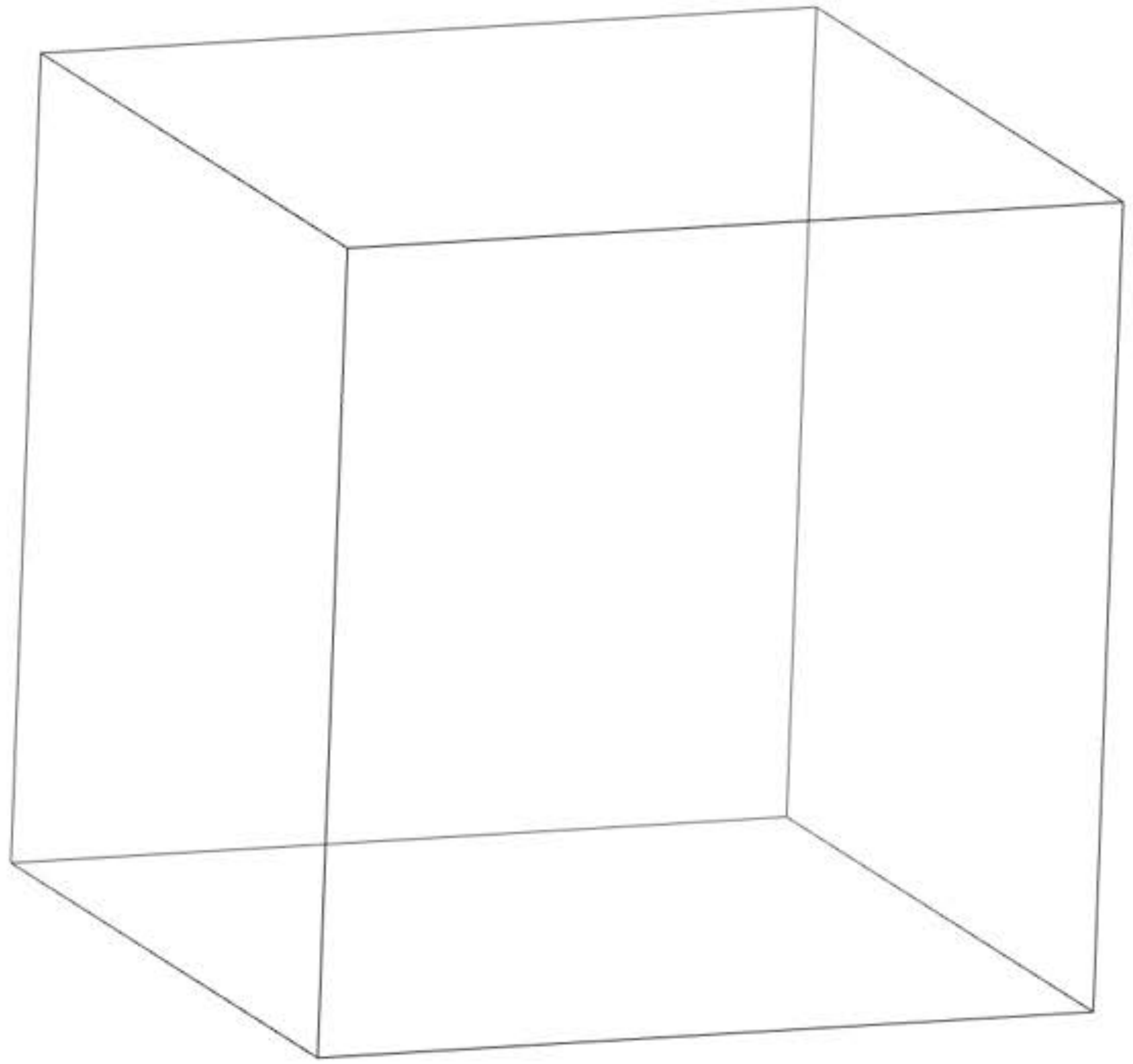
Autoregressive sampling of a crystal

$Cs_2ZnFe(CN)_6$



Autoregressive sampling of a crystal

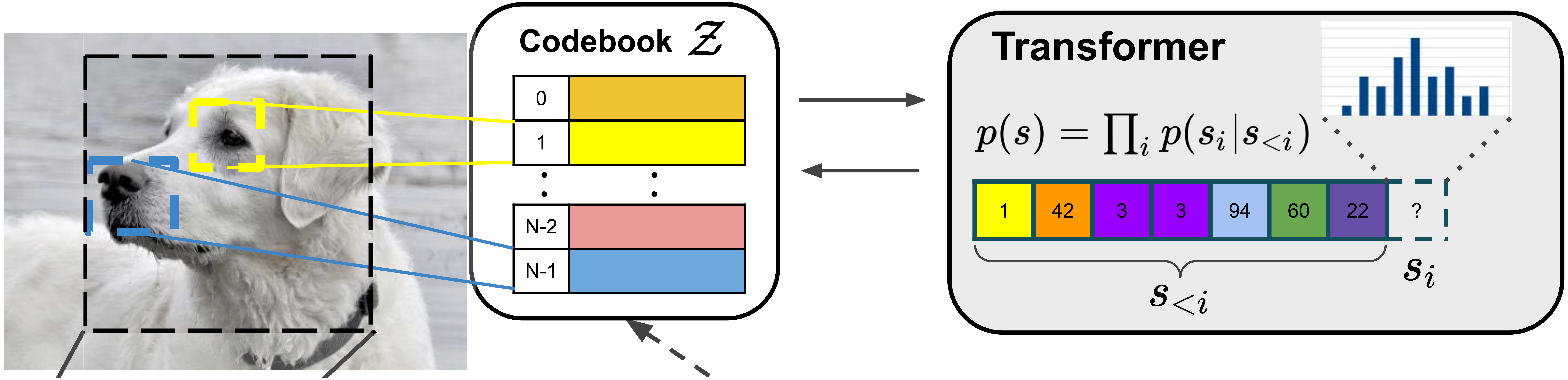
$Cs_2ZnFe(CN)_6$



225-a-Fe-0-0-0-b-Zn-1/2-1/2-1/2-c-Cs-1/4-1/4-1/4-e-C-0.18-0-0-e-N-0.29-0-0-X-10.45-10.45-10.45-90-90-90

Aside: autoregressive transformer for images

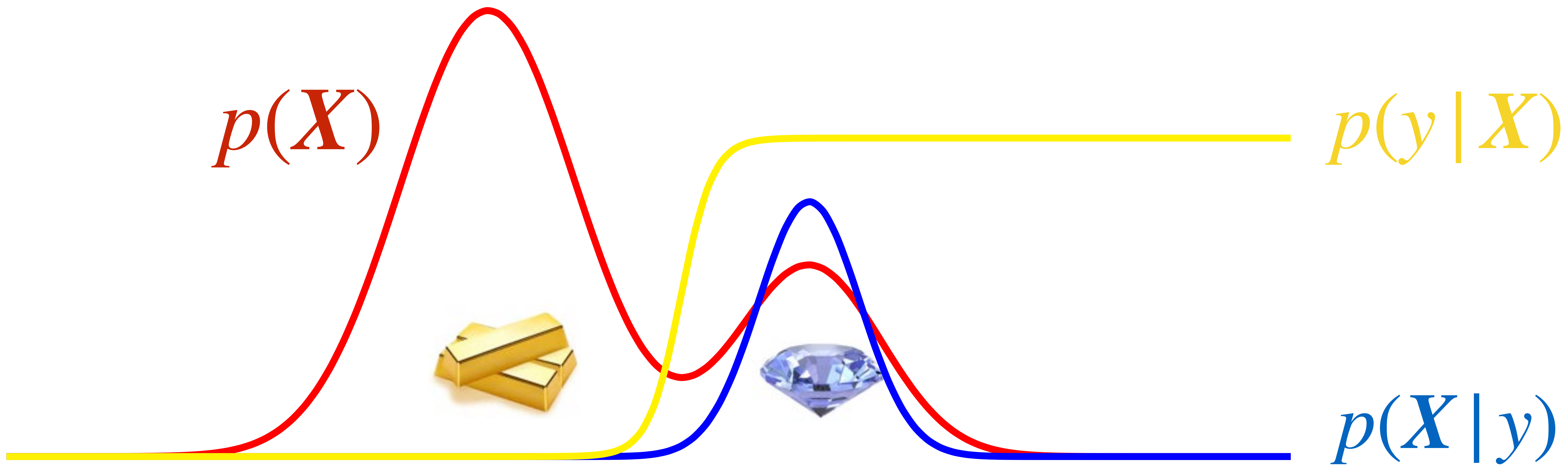
Esser et al, Taming Transformers for High-Resolution Image Synthesis (VQGAN), 2012.09841



learned codebook, see also Tian et al, 2404.02905

CrystalFormer leverages Nature's codebook: the Wyckoff position table

Bayes rule for materials inverse design



How to sample from $p(\boldsymbol{X}\,|\,y)$? Two approaches originated in physics

Markov chain Monte Carlo

Metropolis et al, 1953, Hastings 1970

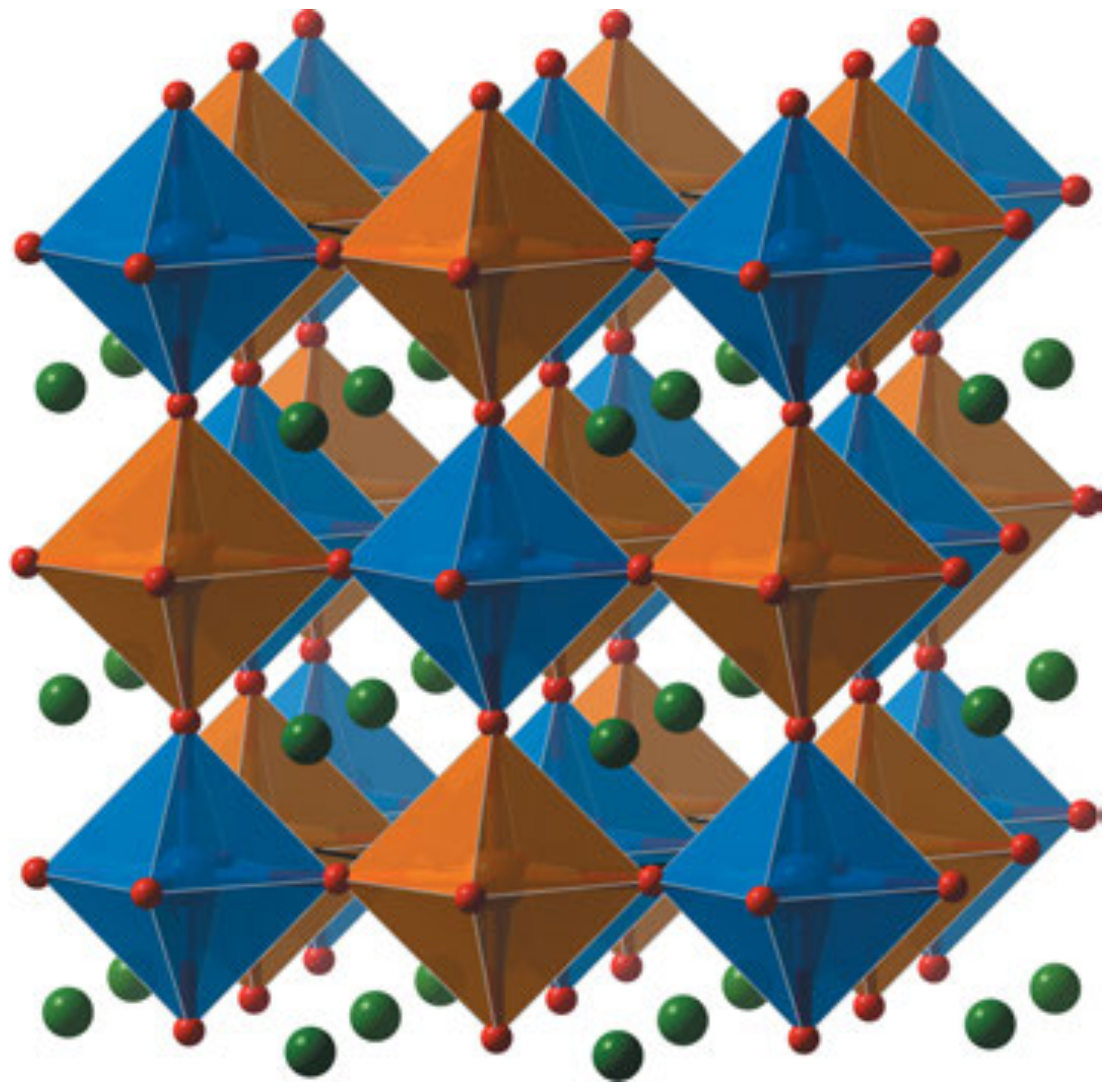
$$\nabla F$$

Variational inference

Gibbs, Feynman, Bogoliubov,..., Jordan et al 1999

MCMC sampling from the posterior

Generate more double perovskites $A_2BB'O_6$



225-a-[?]-0-0-0-b-[?]-1/2-1/2-1/2-c-[?]-1/4-1/4-1/4-e-O-[?]-0-0

$$A(X \to X') = \min\left[1, \frac{p(X'|y)}{p(X|y)}\right]$$

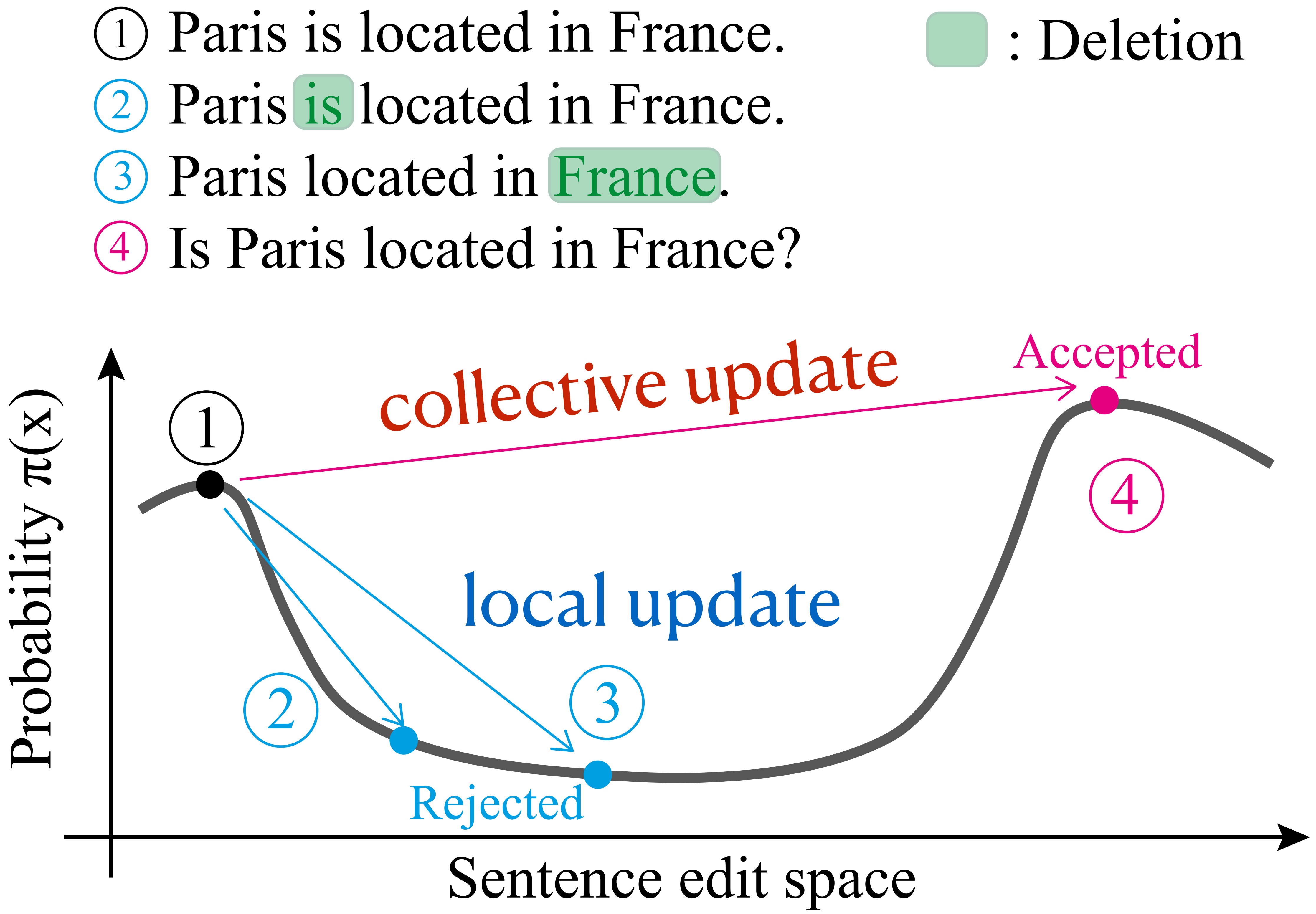
Solve crystal cloze test via MCMC sweep through the “crystal string”

Aside: Constrained sentence generation in languate modeling

$$\pi(x) \propto P_{\mathrm{LM}}(x) \cdot \mathrm{Constraint}(x)$$

“traverses the probabilistic space of high-quality sentences more effectively”

Miao et al, 1811.10996, Zhang et al, 2011.12334



The sequence length for inorganic crystals is ~100 with vocabulary size ~100

So, even naive Metropolis-Hastings with annealing works fine

Variational inference the posterior

$$\mathbb{KL}\left(q(\boldsymbol{X}) \,\|\, p(\boldsymbol{X} \,|\, y)\right) = \underset{\boldsymbol{X}\sim q(\boldsymbol{X})}{\mathbb{E}} \left[-\ln p(y \,|\, \boldsymbol{X})\right] + \mathbb{KL}\left(q(\boldsymbol{X}) \| p(\boldsymbol{X})\right)$$

Variational probability — Likelihood function — Prior

$q(\boldsymbol{X})$ is easy to sample, e.g. another autoregressive model

Variational inference turns a sampling problem into a stochastic optimization problem

Also known as: reinforcement fine-tuning

$$\mathbb{KL}\left(q(\boldsymbol{X}) \parallel p(\boldsymbol{X} \mid y)\right) = \mathop{\mathbb{E}}_{\boldsymbol{X} \sim q(\boldsymbol{X})} \left[-r(\boldsymbol{X})\right] + \mathbb{KL}\left(q(\boldsymbol{X}) \| p(\boldsymbol{X})\right)$$

Fine-tuned model

Reward function

Remain close to the pretrained model

"RL with KL penalties is better viewed as Bayesian inference" Korbar et al, 2205.11275

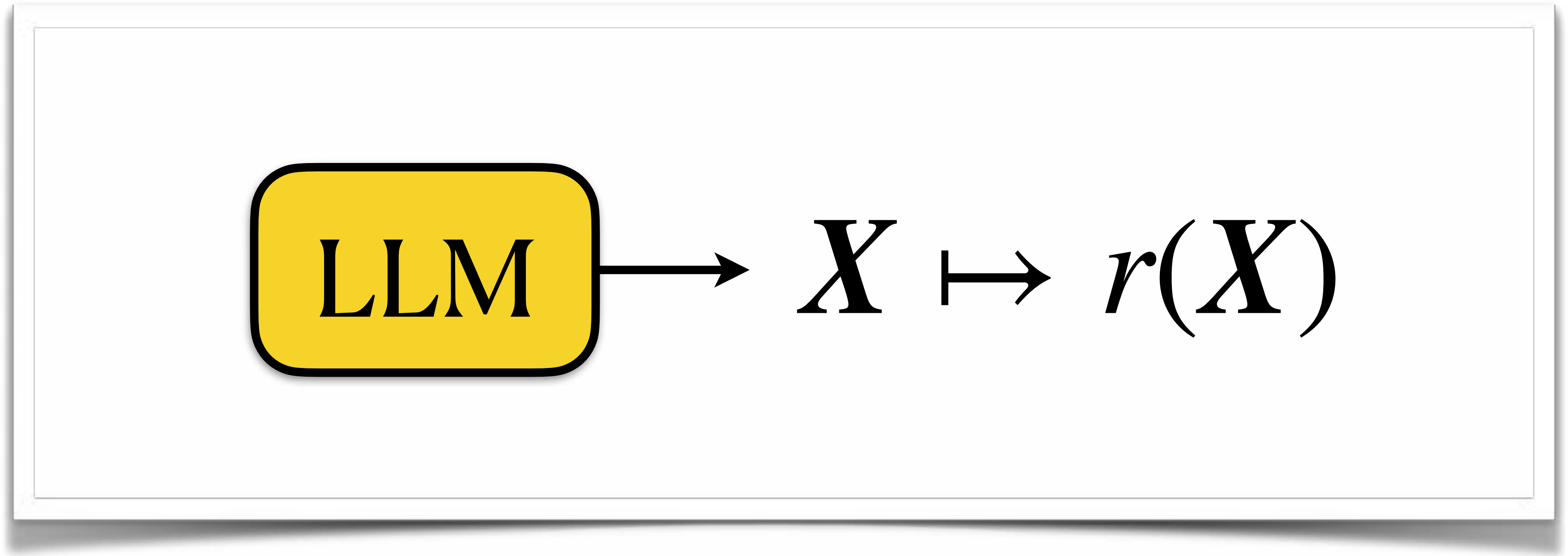
Two sides of the same coin

❶ Pre-training

learn from data
to be a generalist

$$\mathscr{L} = -\,\mathbb{E}_{X\sim\text{data}}\left[\ln p(X)\right]$$

❷ Post-training

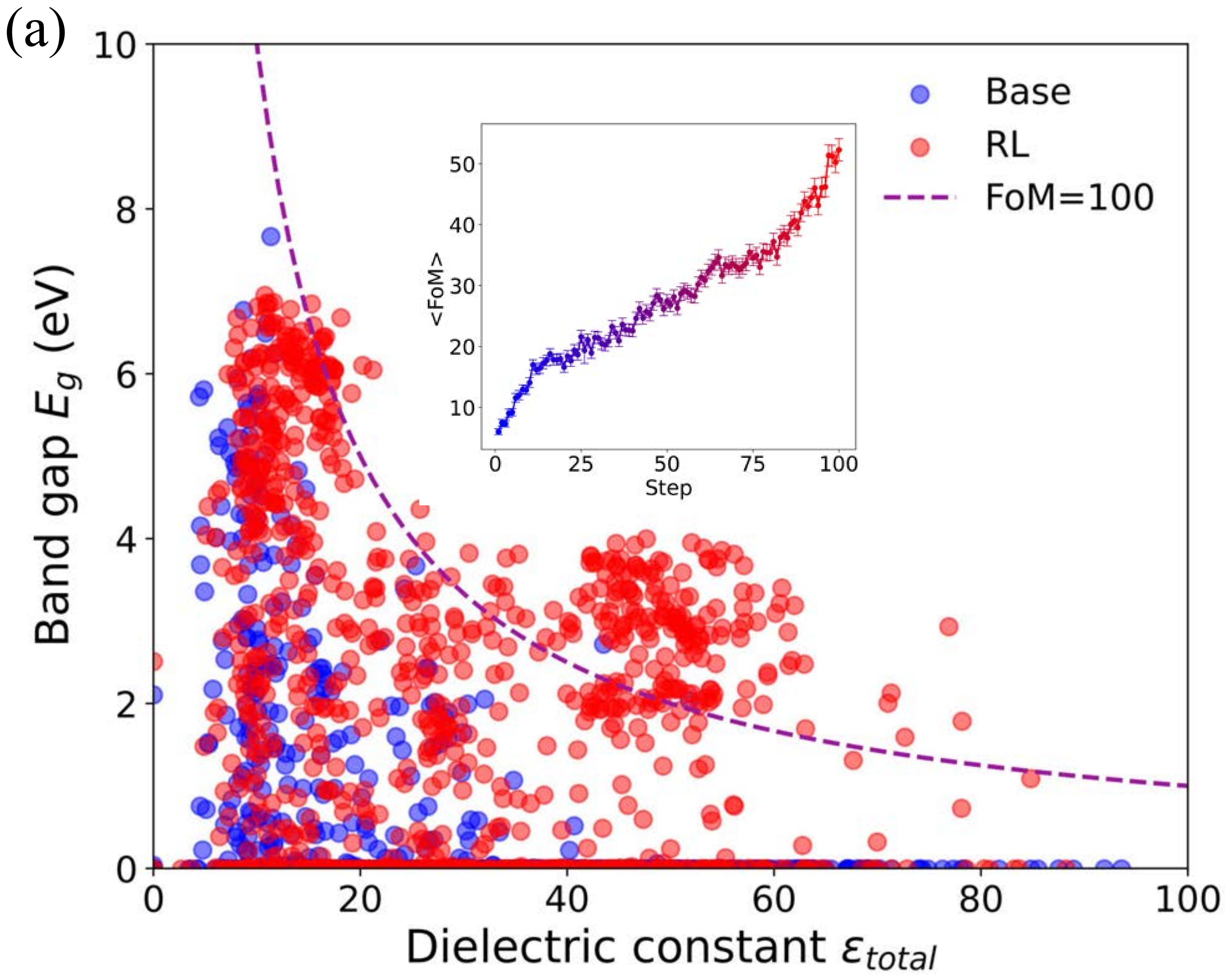


learn from reward
to be a specialized generalist

$$\mathscr{L} = \mathop{\mathbb{E}}_{X\sim q(X)}\left[-r(X)\right] + \mathbb{KL}\left(q(X)\|p(X)\right)$$

$\mathbb{KL}(\text{data} \,\|\, p)$ vs $\mathbb{KL}(q \,\|\, pe^{r})$

Reinforcement fine-tuning for materials design



CrystalFormer-RL, Cao et al, 2504.02367

$$\underset{X\sim q(X)}{\mathbb{E}}\left[\underbrace{r(\boldsymbol{X})}_{\text{Reward}}\right]+\mathbb{KL}\left(q(\boldsymbol{X})\|\underbrace{p(\boldsymbol{X})}_{\text{Pretrained Crystalformer}}\right)$$

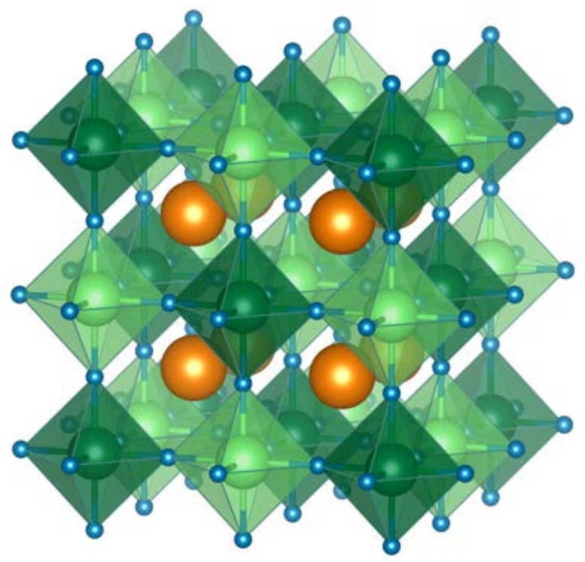
Reward = Band gap x dielectric constant

(Two usually anti-correlated properties)

ε_{total} = 21.32

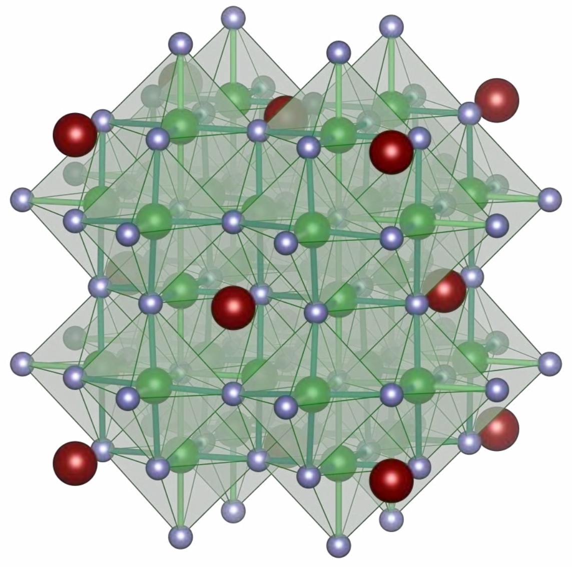
E_g = 5.38eV

E_{hull} = 0.04eV/atom



225-a-Sr-b-Ba-c-Cs-e-F

Cs_2BaSrF_6



225-a-Pb-c-Cl-d-Li-e-Cl

Li_6PbCl_8

ε_{total} = 25.19

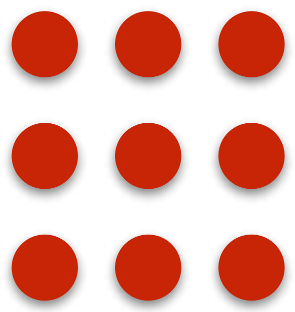
E_g = 4.64 eV

E_{hull} = 0.08eV/atom

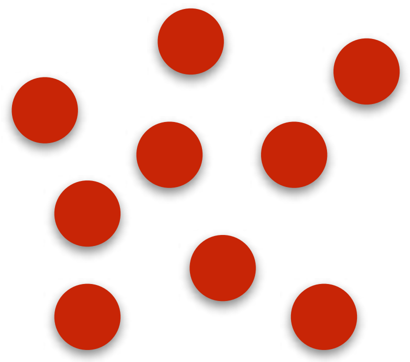
Nature tries to minimize free energy

$$F = E - TS$$

energy



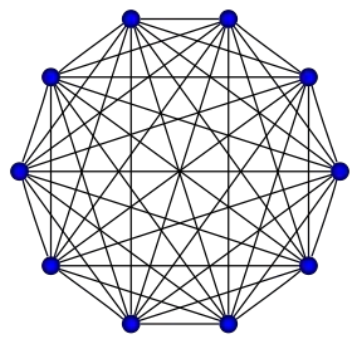
entropy



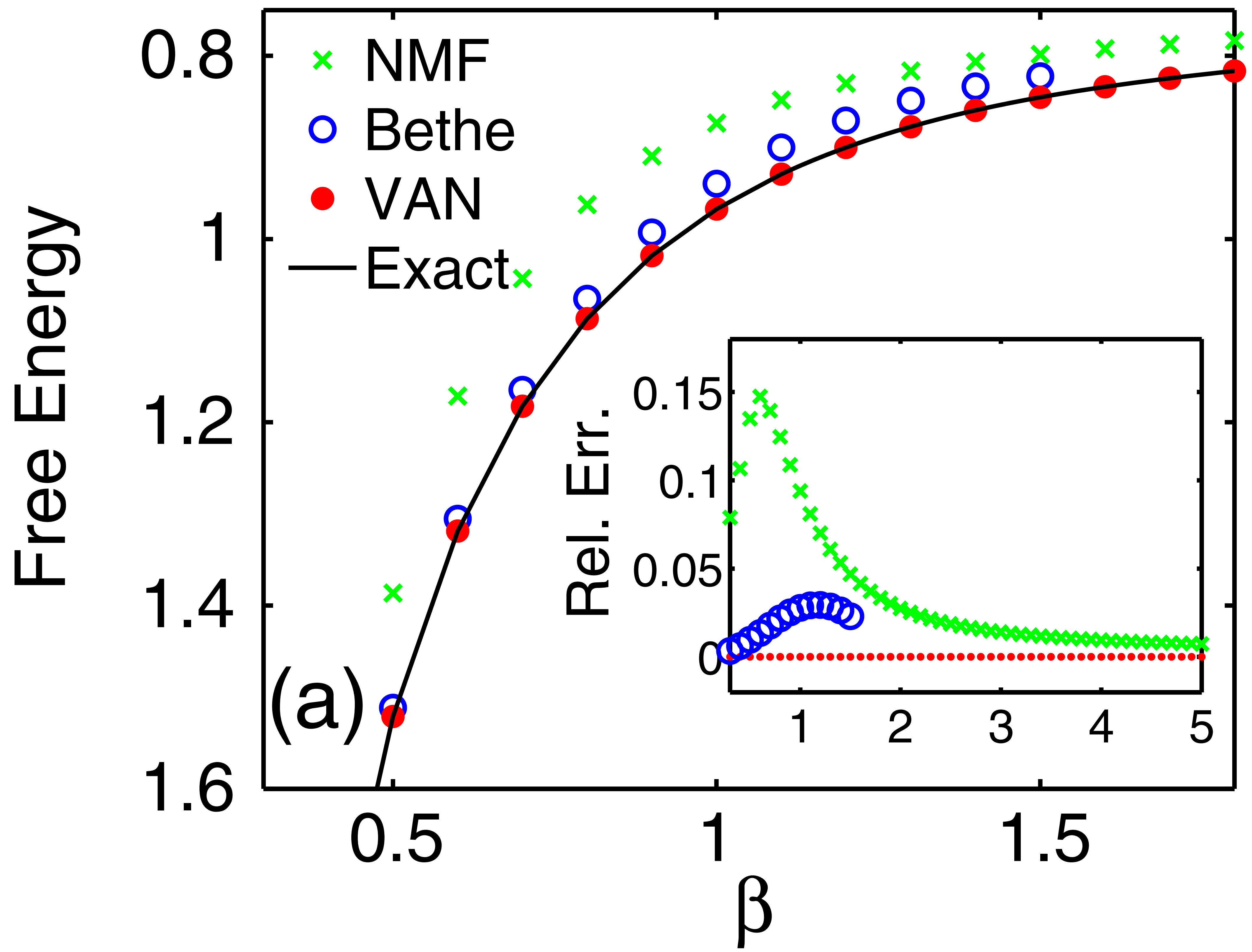
F is a cost function given by Nature

The *same* cost function for training deep generative models

Variational autoregressive network for statistical mechanics



Sherrington-Kirkpatrick spin glass



Objective function: variational free-energy

$$F = \mathop{\mathbb{E}}_{\boldsymbol{X}\sim p(\boldsymbol{X})} \left[E(\boldsymbol{X}) + k_B T \ln p(\boldsymbol{X}) \right]$$

Naive mean-field factorized probability $\quad p(\boldsymbol{X}) = \prod_i p(x_i)$

Bethe approximation pairwise interaction $\quad p(\boldsymbol{X}) = \prod_i p(x_i) \prod_{(i,j)\in E} \frac{p(x_i, x_j)}{p(x_i)p(x_j)}$

Variational autoregressive network $\quad p(\boldsymbol{X}) = \prod_i p(x_i | \boldsymbol{x}_{<i})$

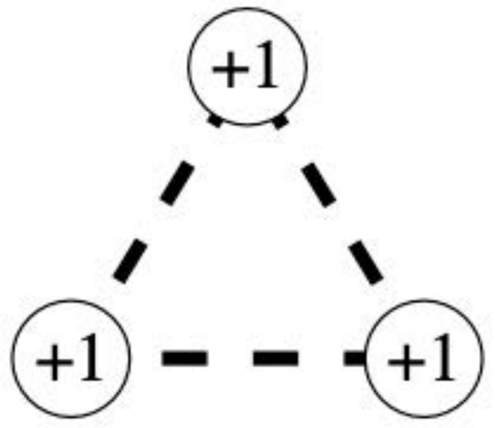
Wu, LW, Zhang, PRL '19

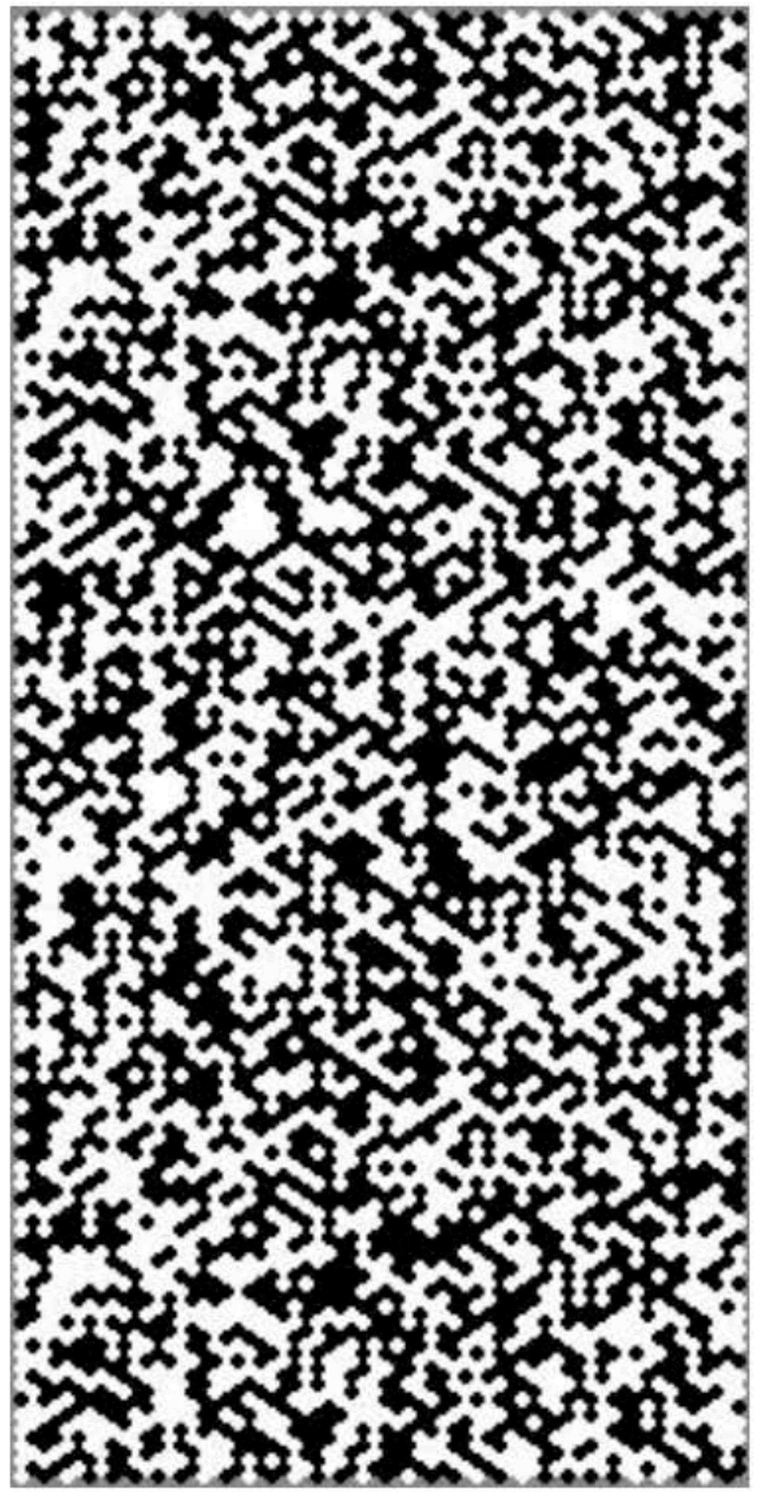
github.com/wdphy16/stat-mech-van

VAN for triangular Ising

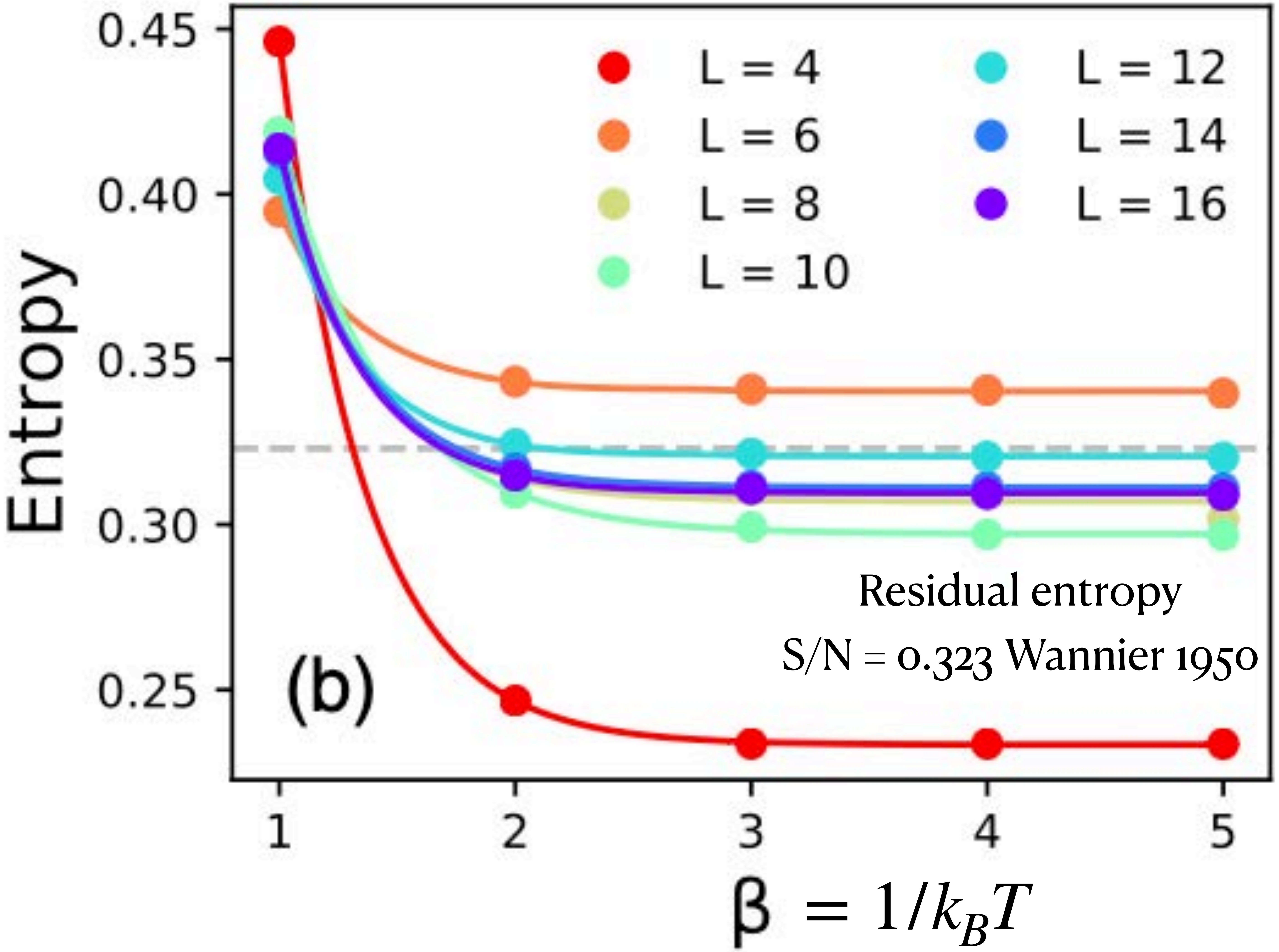
Wu, LW, Zhang, PRL '19

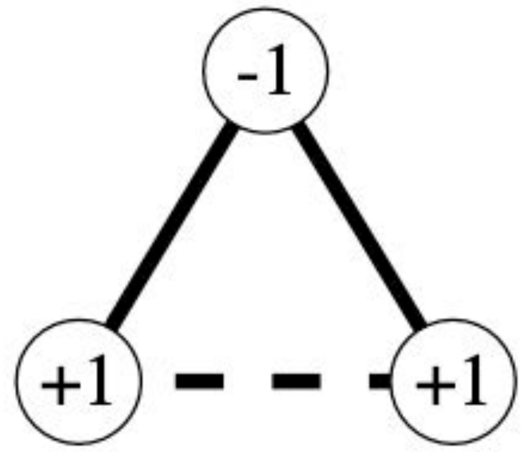
$$F = \mathop{\mathbb{E}}_{\boldsymbol{X} \sim p(\boldsymbol{X})} \left[E(\boldsymbol{X}) + k_B T \ln p(\boldsymbol{X}) \right]$$

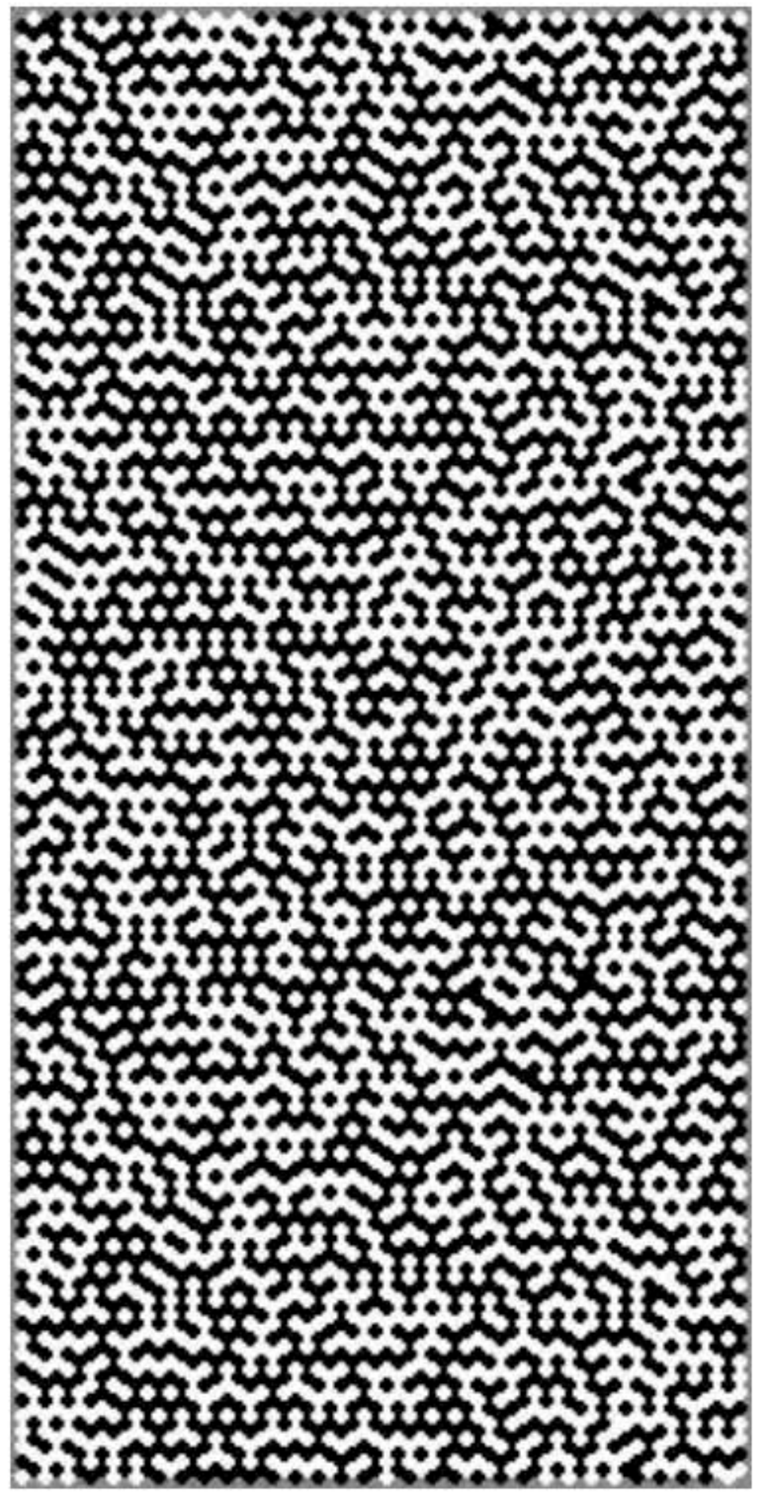




Hot configuration







Cold configuration
MacKay, 2006

VAN (aka RL) for 8-queens problem

$$\mathscr{L} = \mathop{\mathbb{E}}_{X\sim q(X)} \Big[-r(X) + \ln q(X) \Big]$$

Energy exploitation

Entropy exploration

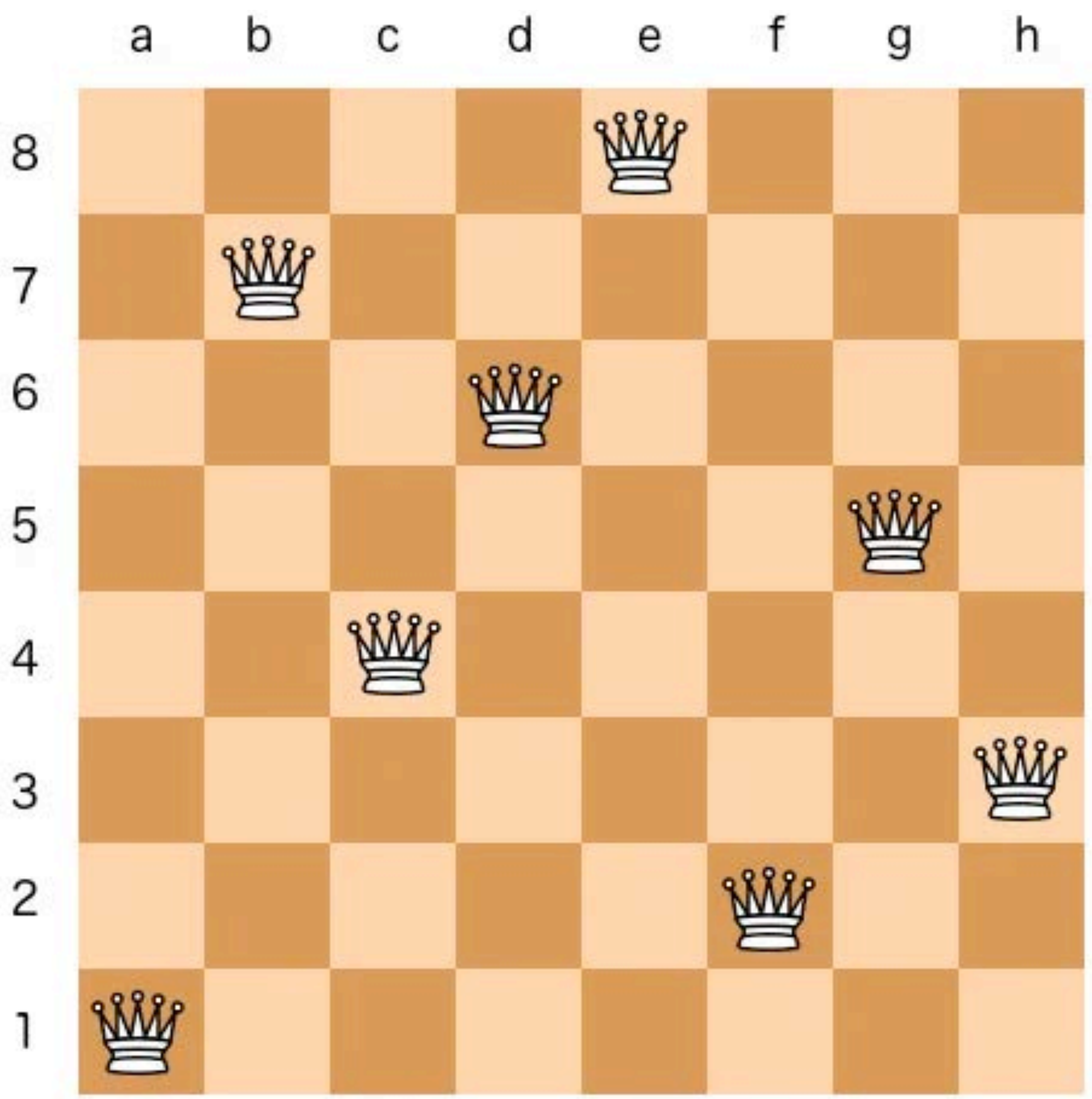
X: a sequence of actions

a1—b7—c4—d6—e8—f2—g5—h3

Reward $r(X) = \begin{cases} 1 & \text{if no attack} \\ 0 & \text{otherwise} \end{cases}$

Policy network

$q(X) = q(x_1)q(x_2 | x_1)\ldots$

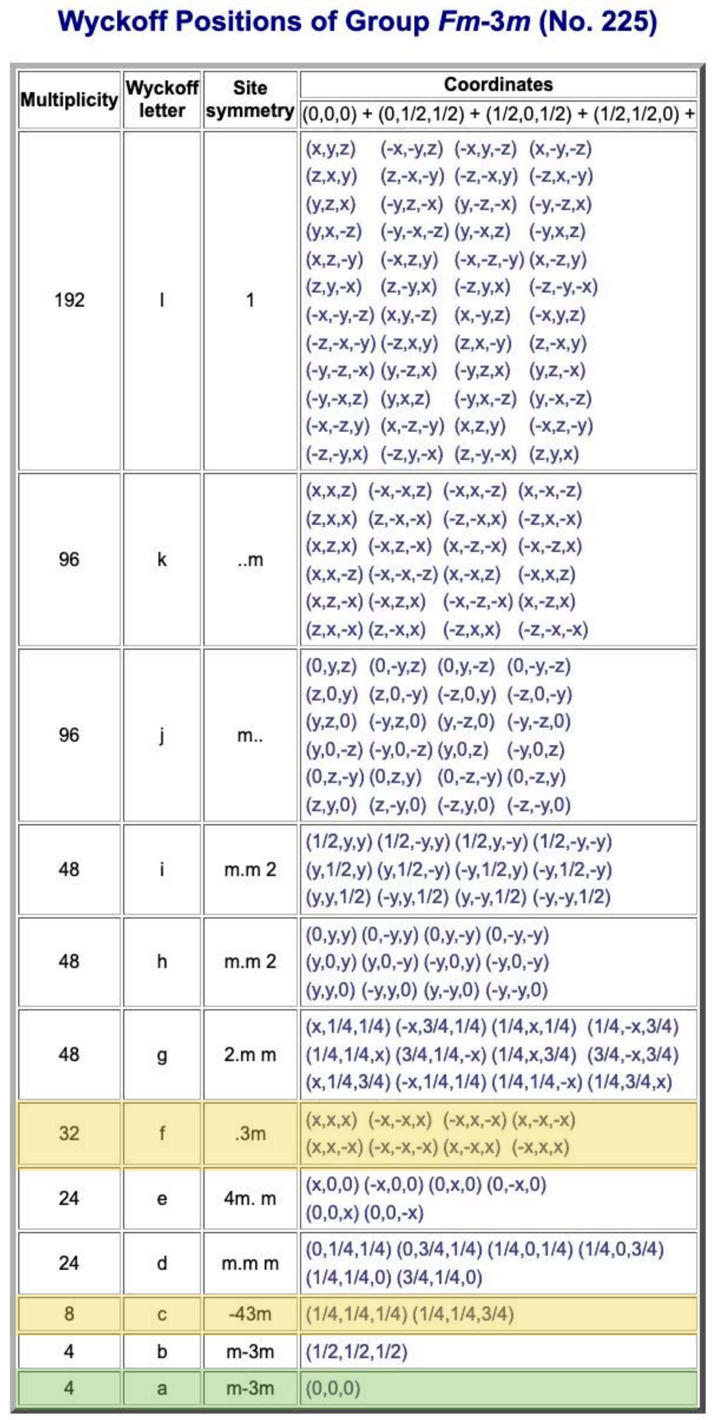


Board size	Solutions
8	92
12	14,200
16	14,772,512
20	39,029,188,884
24	227,514,171,973,736
28	???

Can you solve it ?

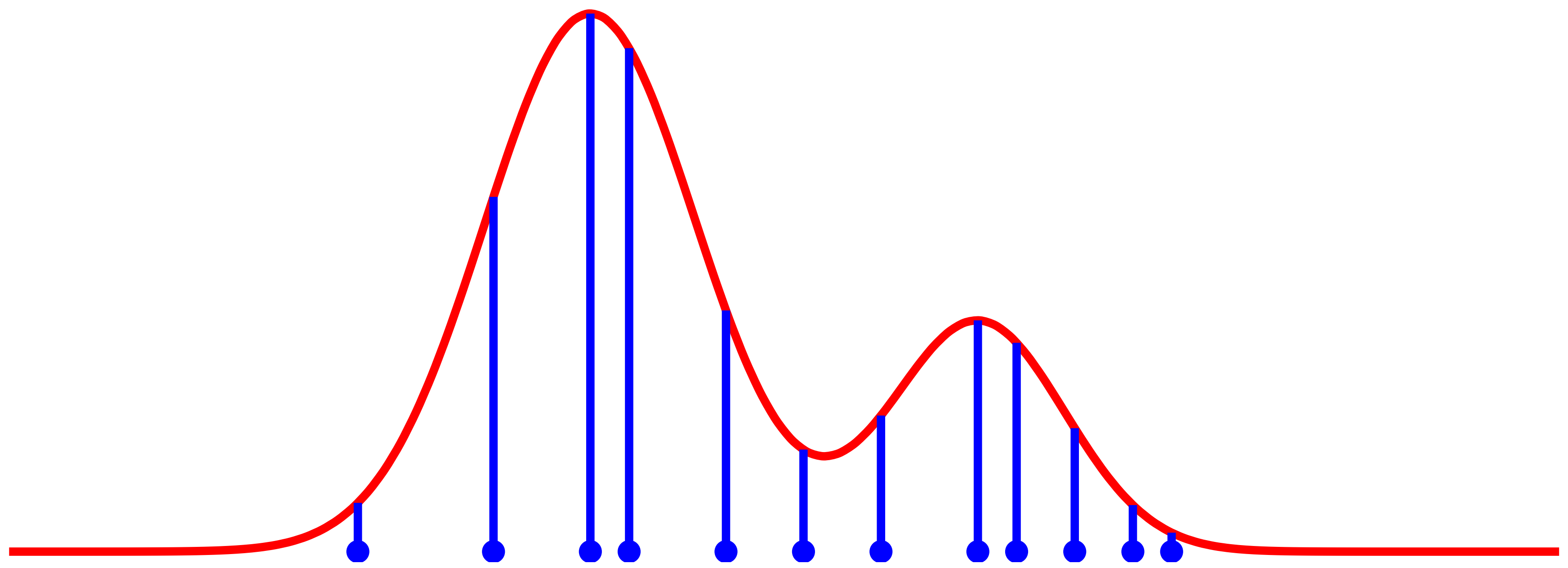
Autoregressive model blueprint

1. Ordering
2. Tokenization
3. Objective function
4. Inference



Wyckoff Positions of Group *Fm*-3*m* (No. 225)

Multiplicity	Wyckoff letter	Site symmetry	Coordinates (0,0,0) + (0,1/2,1/2) + (1/2,0,1/2) + (1/2,1/2,0) +
192	l	1	(x,y,z) (-x,-y,z) (-x,y,-z) (x,-y,-z) (z,x,y) (z,-x,-y) (-z,-x,y) (-z,x,-y) (y,z,x) (-y,z,-x) (y,-z,-x) (-y,-z,x) (y,x,-z) (-y,-x,-z) (y,-x,z) (-y,x,z) (x,z,-y) (-x,z,y) (-x,-z,-y) (x,-z,y) (z,y,-x) (z,-y,x) (-z,y,x) (-z,-y,-x) (-x,-y,-z) (x,y,-z) (x,-y,z) (-x,y,z) (-z,-x,-y) (-z,x,y) (z,x,-y) (z,-x,y) (-y,-z,-x) (y,-z,x) (-y,z,x) (y,z,-x) (-y,-x,z) (y,x,z) (-y,x,-z) (y,-x,-z) (-x,-z,y) (x,-z,-y) (x,z,y) (-x,z,-y) (-z,-y,x) (-z,y,-x) (z,-y,-x) (z,y,x)
96	k	..m	(x,x,z) (-x,-x,z) (-x,x,-z) (x,-x,-z) (z,x,x) (z,-x,-x) (-z,-x,x) (-z,x,-x) (x,z,x) (-x,z,-x) (x,-z,-x) (-x,-z,x) (x,x,-z) (-x,-x,-z) (x,-x,z) (-x,x,z) (x,z,-x) (-x,z,x) (-x,-z,-x) (x,-z,x) (z,x,-x) (z,-x,x) (-z,x,x) (-z,-x,-x)
96	j	m..	(0,y,z) (0,-y,z) (0,y,-z) (0,-y,-z) (z,0,y) (z,0,-y) (-z,0,y) (-z,0,-y) (y,z,0) (-y,z,0) (y,-z,0) (-y,-z,0) (y,0,-z) (-y,0,-z) (y,0,z) (-y,0,z) (0,z,-y) (0,z,y) (0,-z,-y) (0,-z,y) (z,y,0) (z,-y,0) (-z,y,0) (-z,-y,0)
48	i	m.m 2	(1/2,y,y) (1/2,-y,y) (1/2,y,-y) (1/2,-y,-y) (y,1/2,y) (y,1/2,-y) (-y,1/2,y) (-y,1/2,-y) (y,y,1/2) (-y,y,1/2) (y,-y,1/2) (-y,-y,1/2)
48	h	m.m 2	(0,y,y) (0,-y,y) (0,y,-y) (0,-y,-y) (y,0,y) (y,0,-y) (-y,0,y) (-y,0,-y) (y,y,0) (-y,y,0) (y,-y,0) (-y,-y,0)
48	g	2.m m	(x,1/4,1/4) (-x,3/4,1/4) (1/4,x,1/4) (1/4,-x,3/4) (1/4,1/4,x) (3/4,1/4,-x) (1/4,x,3/4) (3/4,-x,3/4) (x,1/4,3/4) (-x,1/4,1/4) (1/4,1/4,-x) (1/4,3/4,x)
32	f	.3m	(x,x,x) (-x,-x,x) (-x,x,-x) (x,-x,-x) (x,x,-x) (-x,-x,-x) (x,-x,x) (-x,x,x)
24	e	4m. m	(x,0,0) (-x,0,0) (0,x,0) (0,-x,0) (0,0,x) (0,0,-x)
24	d	m.m m	(0,1/4,1/4) (0,3/4,1/4) (1/4,0,1/4) (1/4,0,3/4) (1/4,1/4,0) (3/4,1/4,0)
8	c	-43m	(1/4,1/4,1/4) (1/4,1/4,3/4)
4	b	m-3m	(1/2,1/2,1/2)
4	a	m-3m	(0,0,0)



$$\mathbb{KL}(\text{data} \,\|\, p) \quad \text{vs} \quad \mathbb{KL}(p \,\|\, e^{-E/k_B T})$$