HEP for AI and AI for HEP

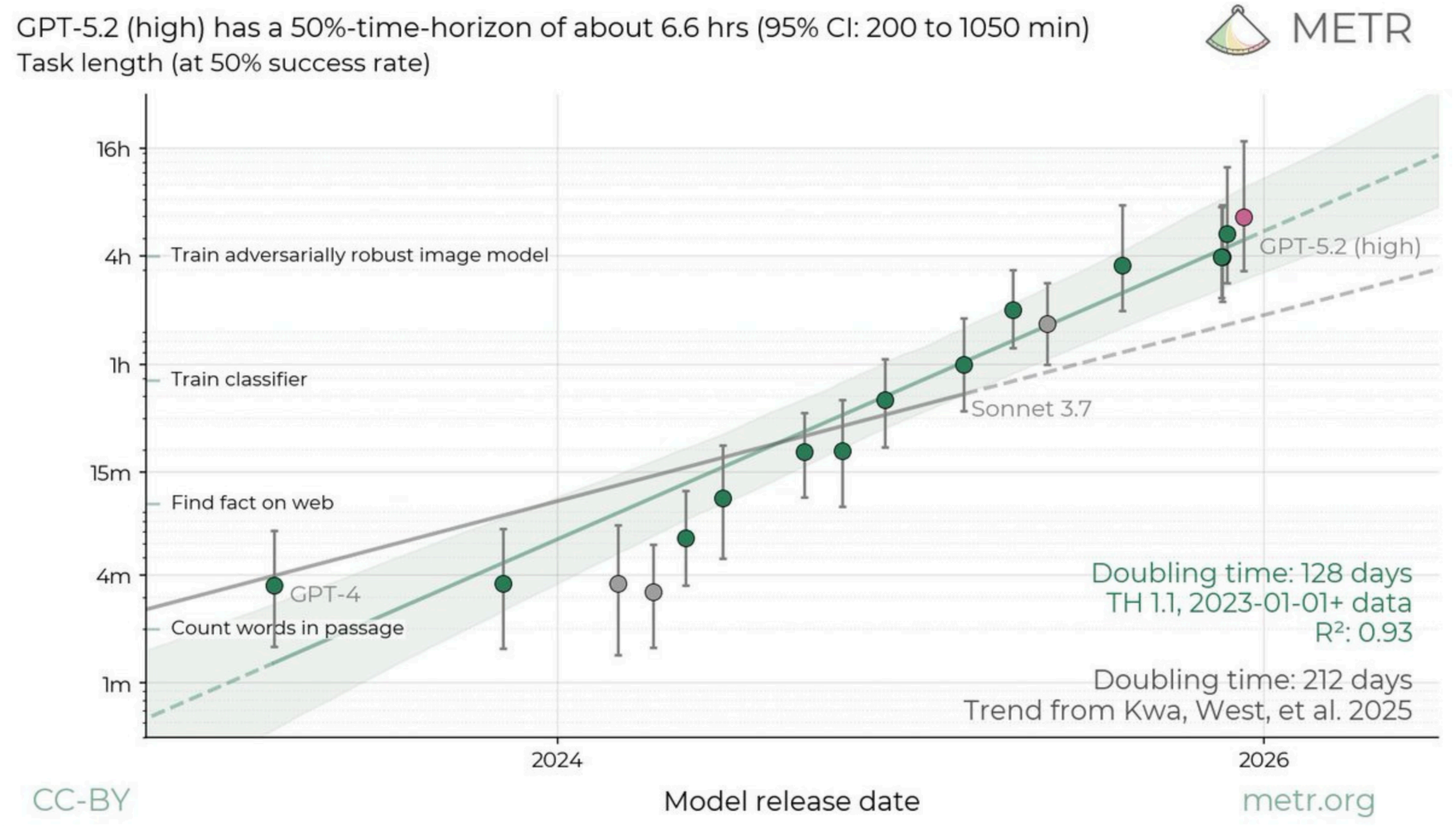
Yoni Kahn
University of Toronto/Vector Institute
WNPPC 2026
February 15, 2026

work supported by the Canadian AI Safety Institute
Research Program at CIFAR

The dawn of the AI age

Prompt: Create a storyboard for this scene

New tools continue to impress on a daily basis: large language models can now reliably solve graduate problems in physics, write code, even start to act and plan autonomously



Outcome. Within ~18 minutes, the model produced the correct curved-space generators closing into $\mathsf{SL}(2,\mathbb{R})$:

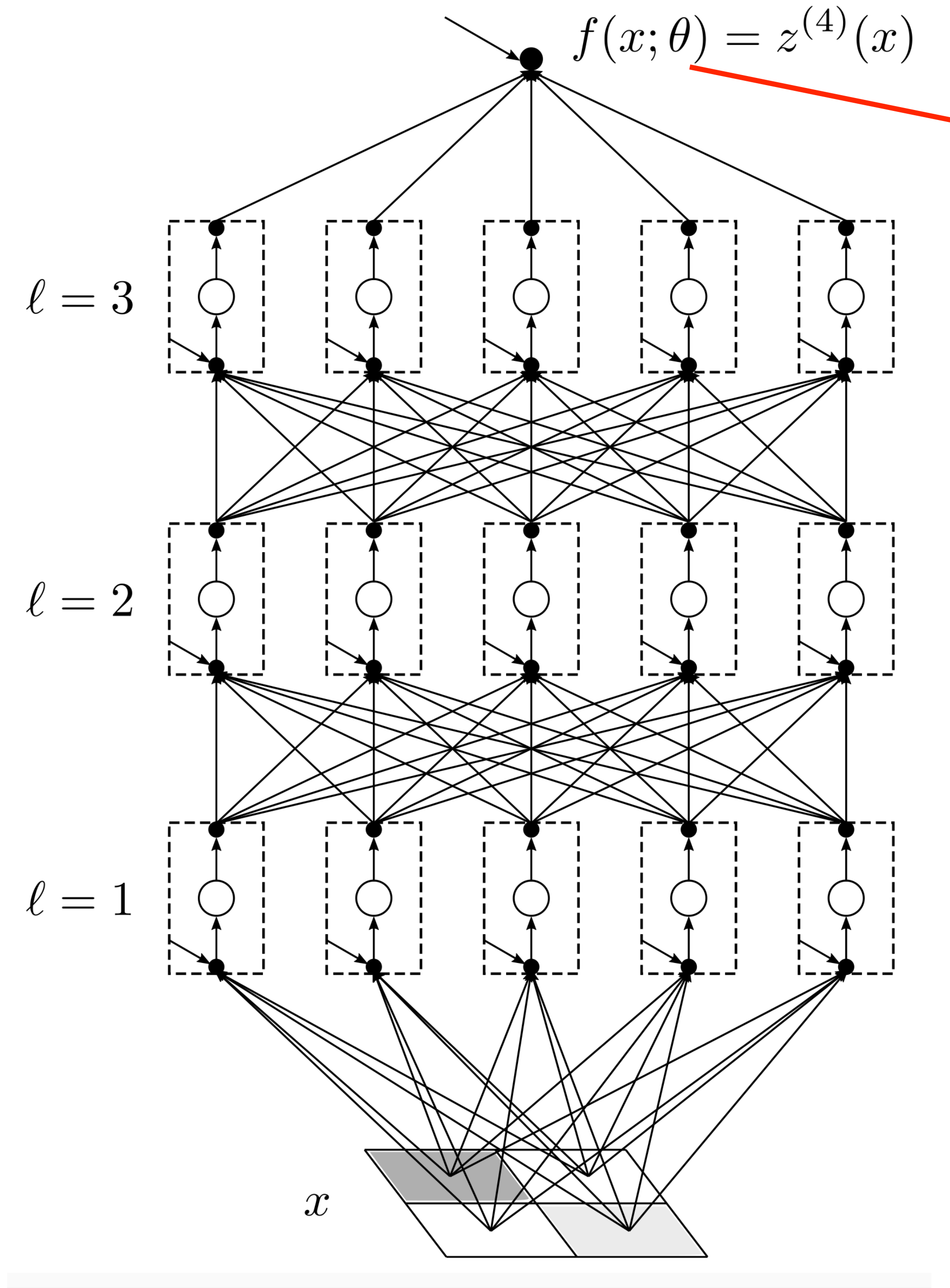
$$H_+ = \frac{x\,\Delta\,\partial_r + (r-M)(1-x^2)\,\partial_x}{(r-M)^2-(M^2-a^2)x^2}, \qquad \text{(I.4a)}$$

$$H_0 = \frac{(r-M)\,\Delta\,\partial_r + (M^2-a^2)x(1-x^2)\,\partial_x}{(r-M)^2-(M^2-a^2)x^2} + \frac{1}{2}, \qquad \text{(I.4b)}$$

$$H_- = \frac{(M^2-a^2)x\,\Delta\,\partial_r - (r-M)(1-x^2)[\Delta-(M^2-a^2)x^2]\,\partial_x}{(r-M)^2-(M^2-a^2)x^2} + x\,\Delta\,\partial_r + (r-M)\,x. \qquad \text{(I.4c)}$$

The hard problem

Why does a trained neural network give the results it does?



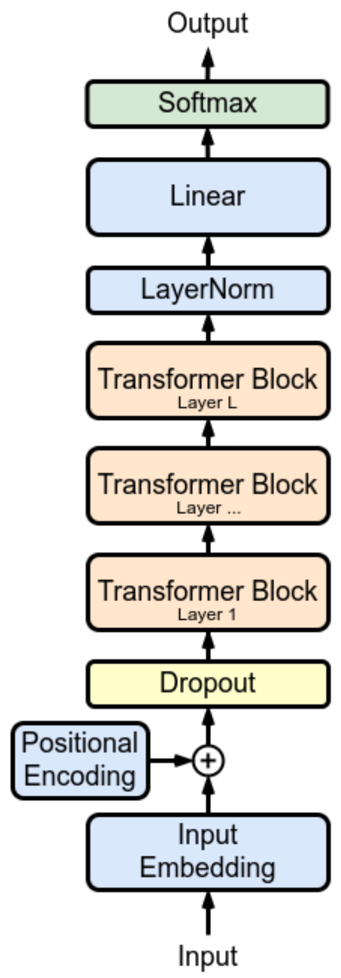
$$\theta \equiv \{W_{ij}^{(\ell)}, b_i^{(\ell)}\}$$

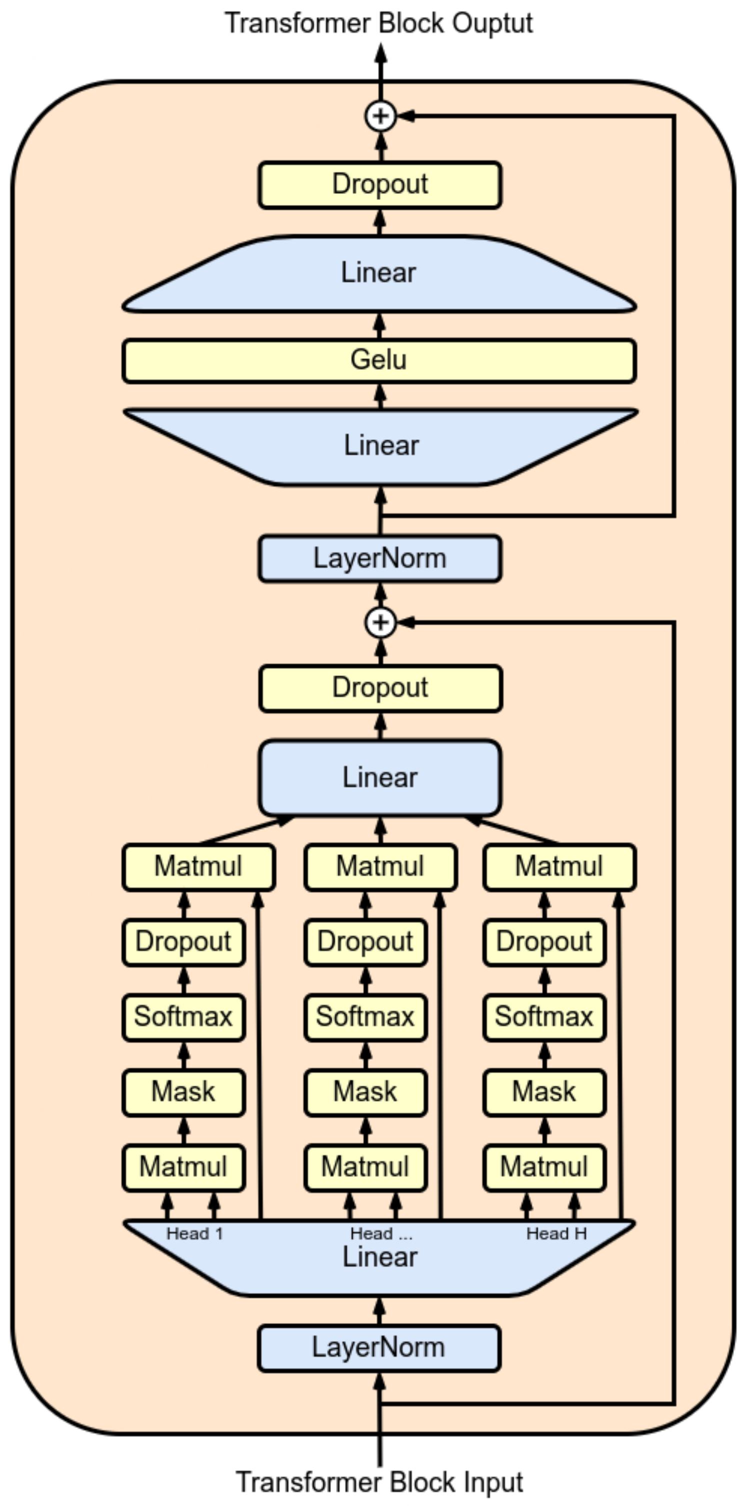
$$z_i^{(\ell+1)} = W_{ij}^{(\ell+1)} \sigma_j^{(\ell)} + b_j^{(\ell+1)}$$

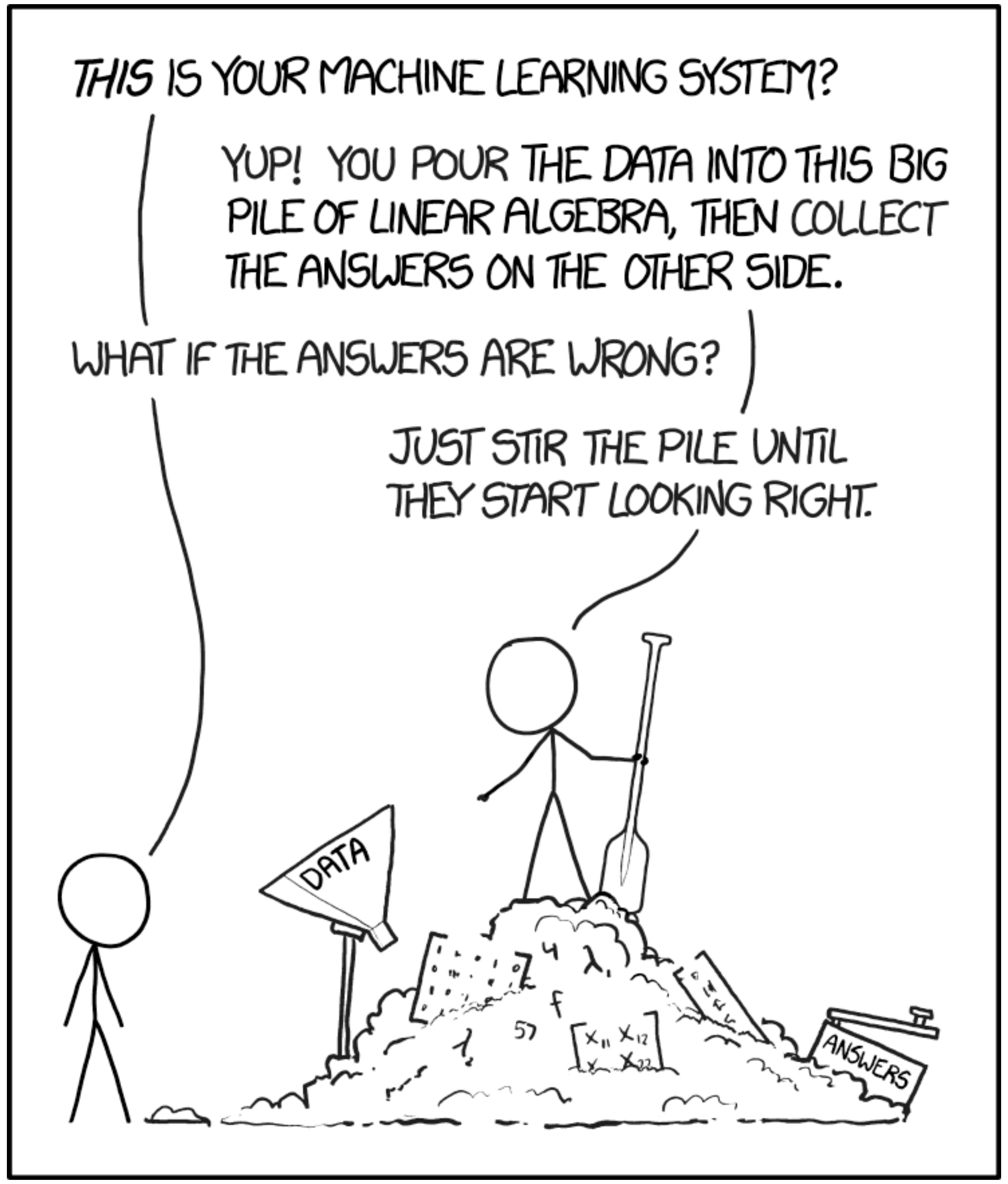
$$\vdots$$

$$\sigma_i^{(1)} = \sigma(z_i^{(1)})$$

$$z_i^{(1)} = W_{ij}^{(1)} x_j + b_i^{(1)}$$



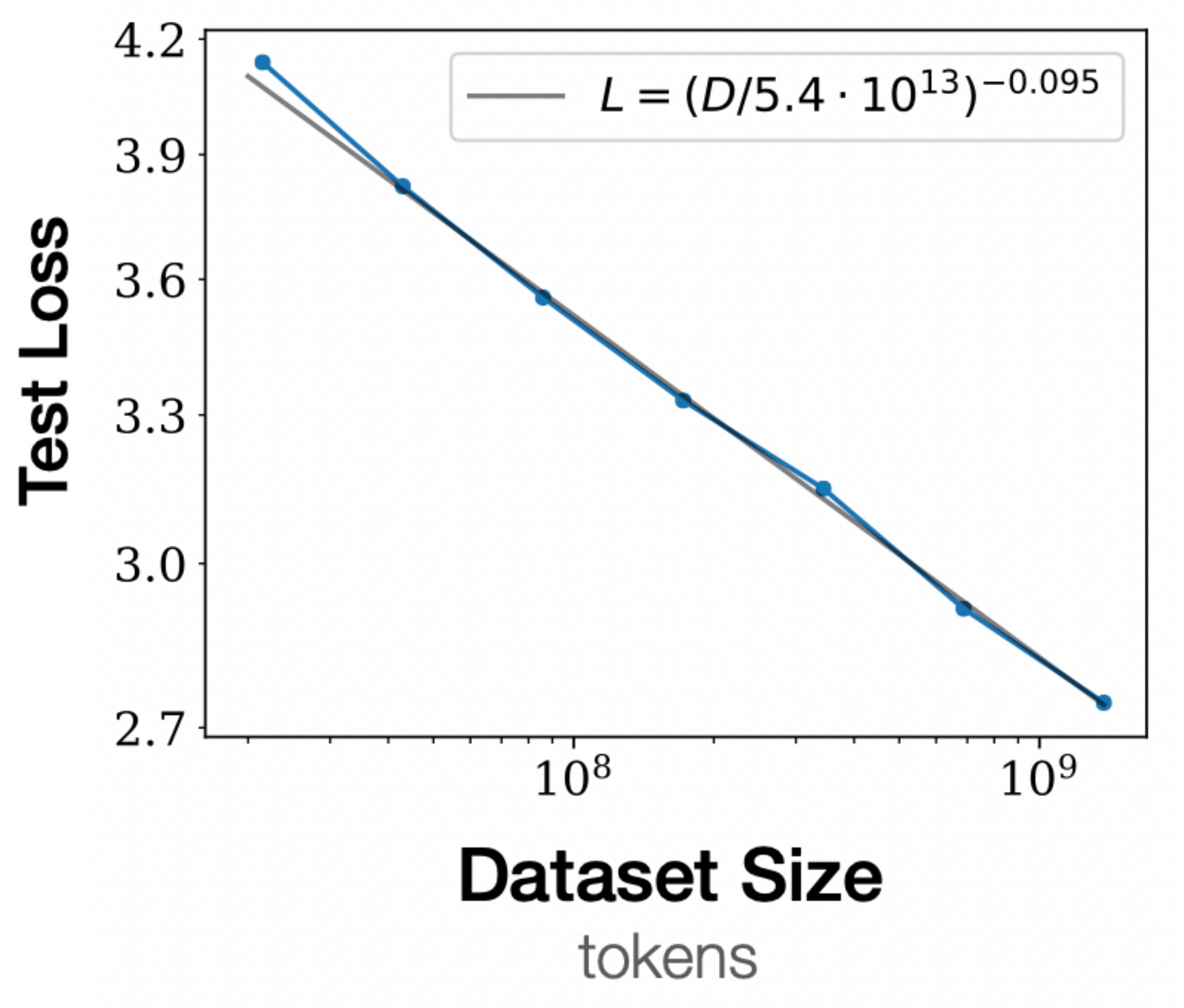


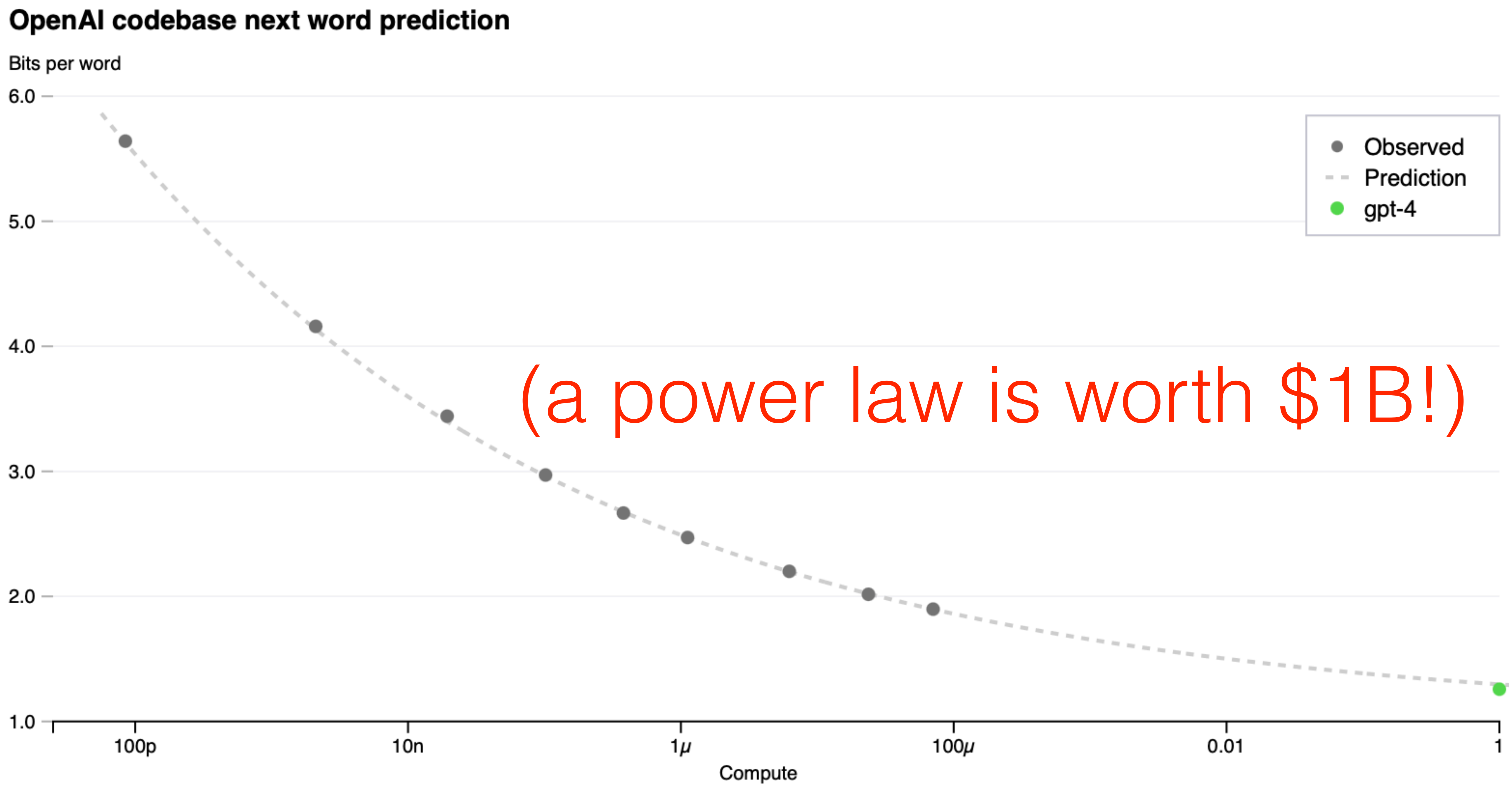


Basic structure: some set of operations repeated L times ("layers")

Hard subproblem 1

Where do scaling laws come from?



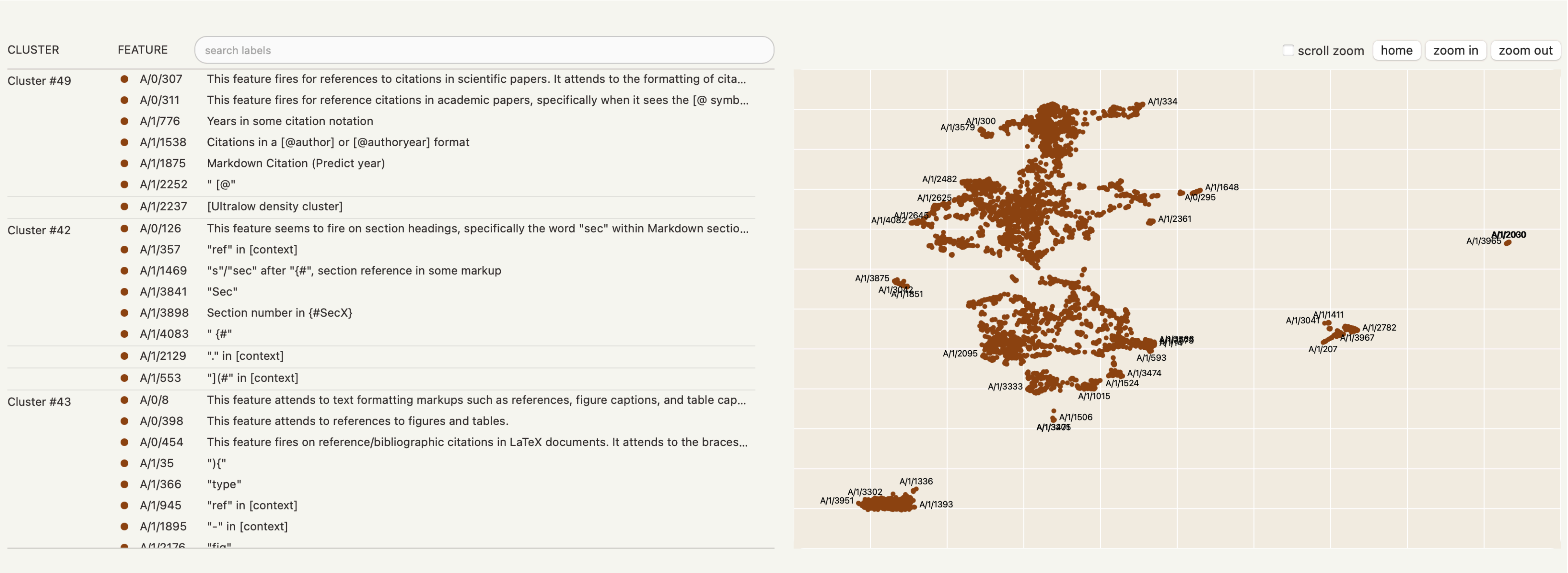


(a power law is worth $1B!)

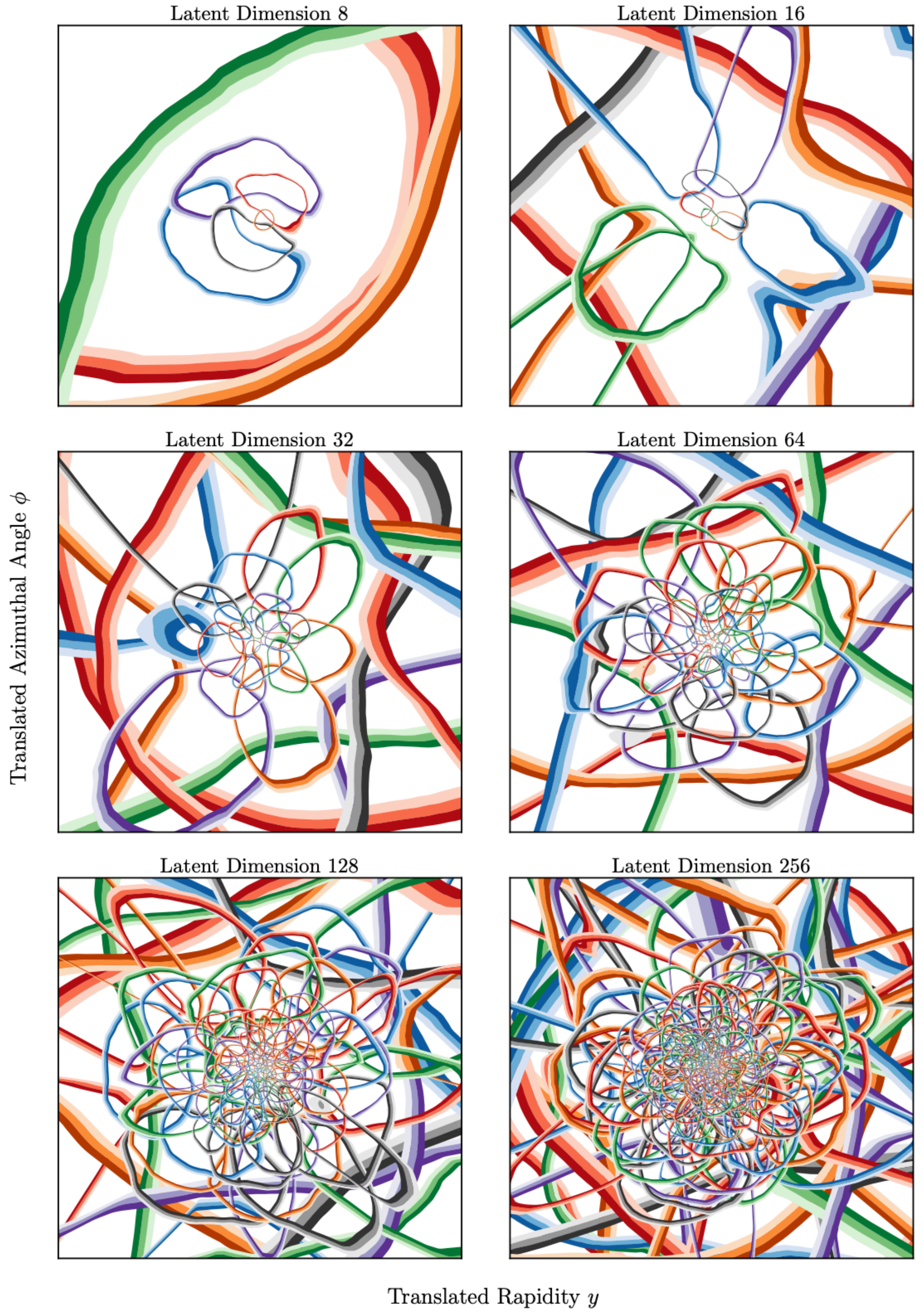
[Kaplan, McCandlish, et al., arXiv:2001.08361; OpenAI GPT-4 blog post]

Hard subproblem 2

Can we interpret what was learned?



A “mind map” of Anthropic’s Claude



Learned features from high-energy physics data

If we want to use AI for science (or make sure it does what we want in society at large), this is mandatory.

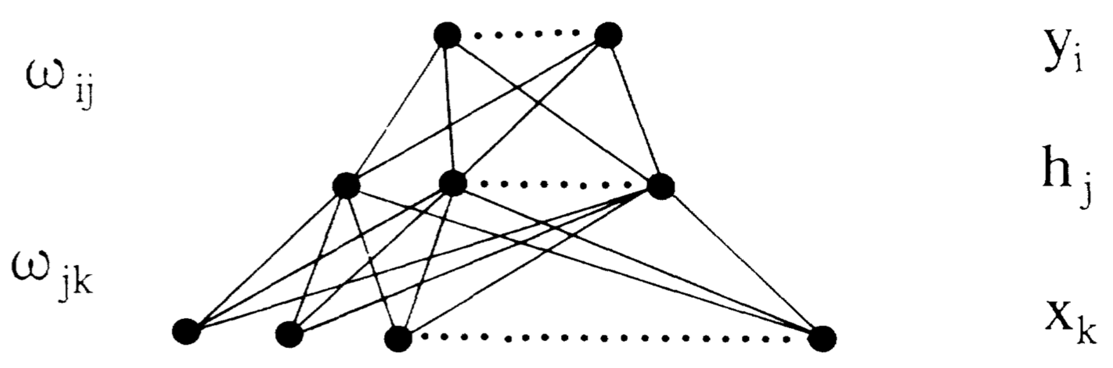
[Anthropic blog post; Komiske, Metodiev, Thaler, arXiv:1810.05165]

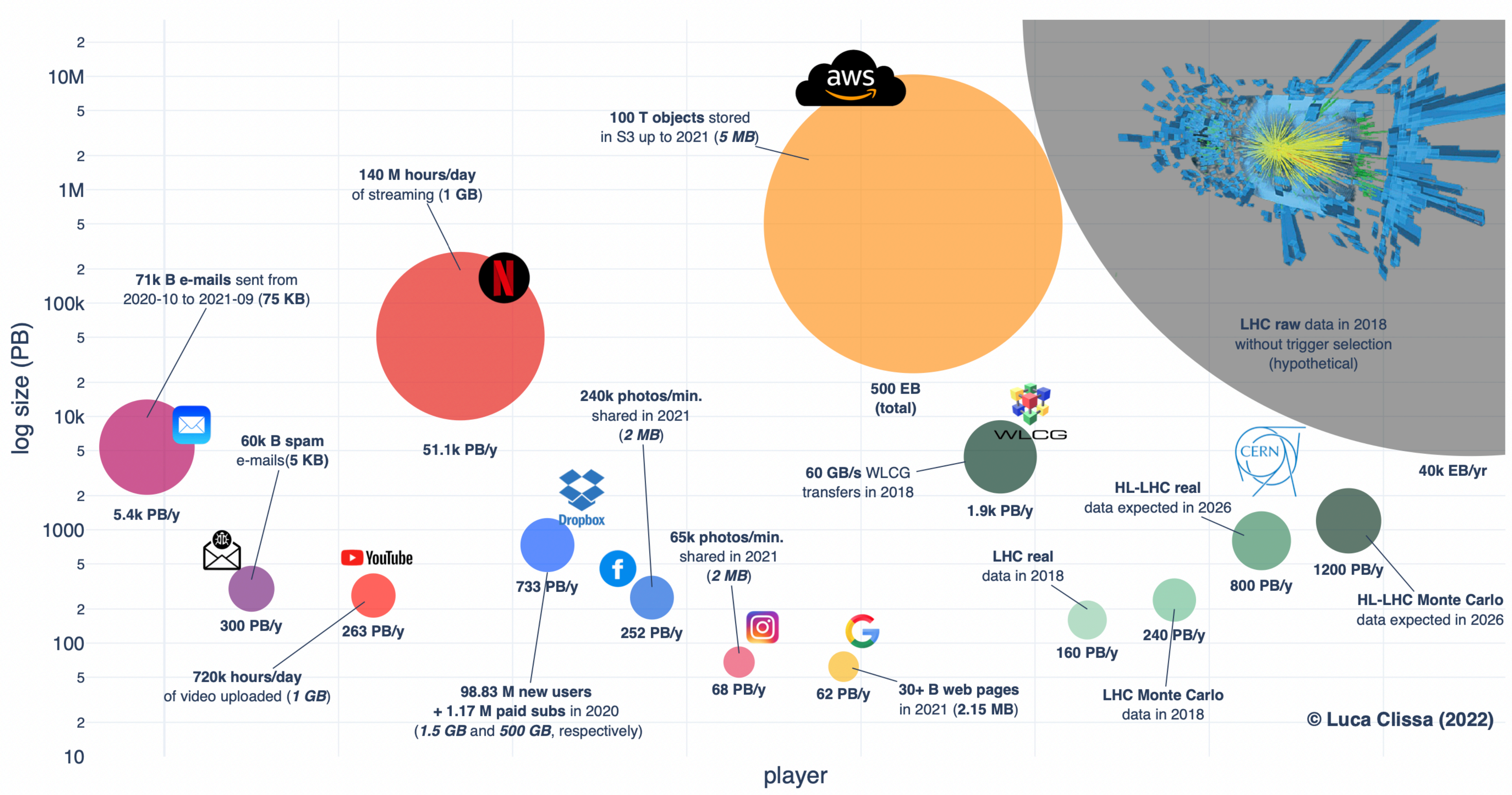
Where is our ChatGPT or AlphaFold moment in (high-energy) physics?

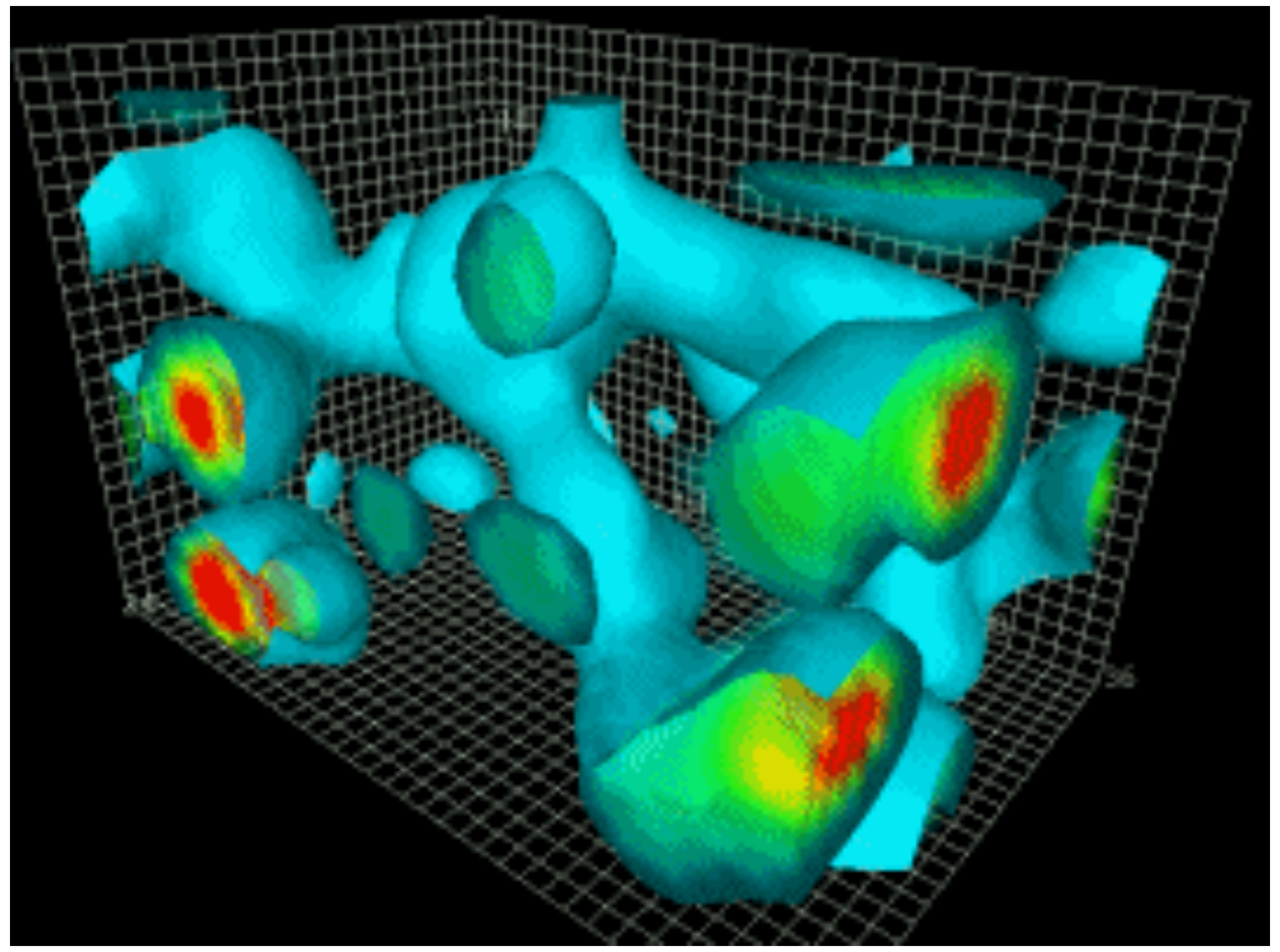
Finding Gluon Jets with a Neural Trigger

Leif Lönnblad,[a] Carsten Peterson,[b] and Thorsteinn Rögnvaldsson[c]
Department of Theoretical Physics, University of Lund, Sölvegatan 14A, S-22362 Lund, Sweden
(Received 6 April 1990)

Using a neural-network classifier we are able to separate gluon from quark jets originating from Monte Carlo–generated e^+e^- events with 85%–90% accuracy.





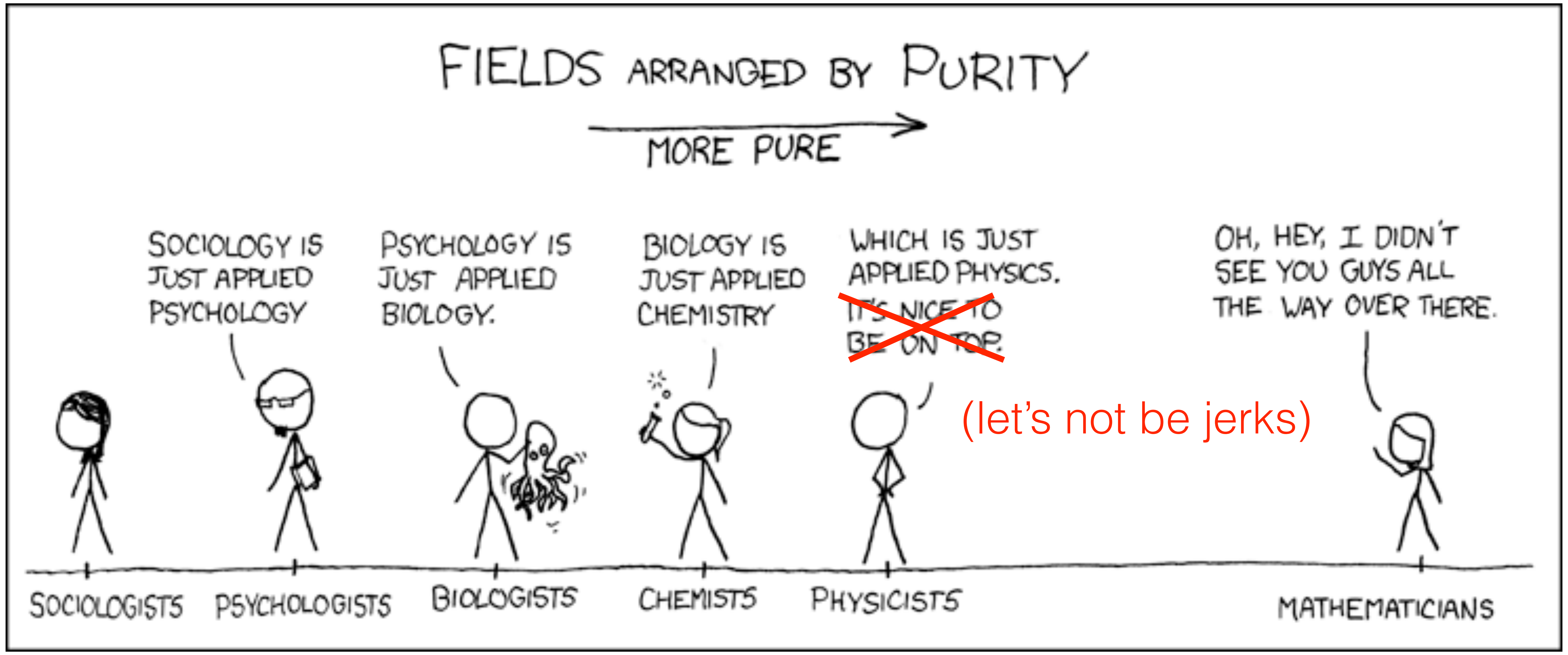


HEP has been using neural networks for 35 years.
AI certainly helps, but arguably not at the transformative level of language -> LLMs.
How do we improve this?

[Lönnblad, Peterson, Rögnvaldsson, PRL 1990; L. Clissa, arXiv:2202.07659; University of Adelaide, Lattice QCD visualizations]

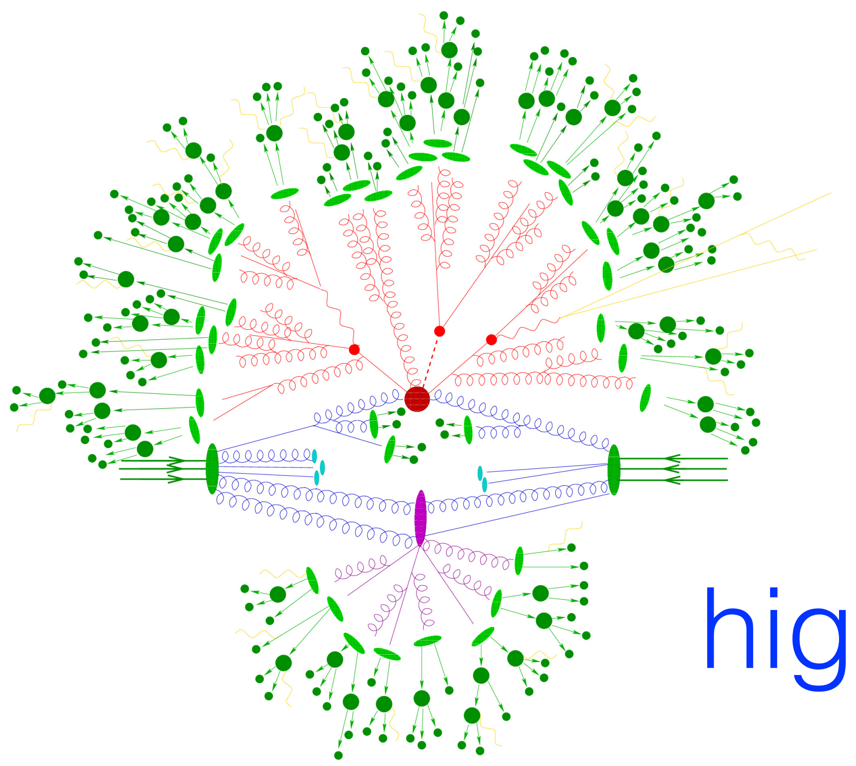
Why might physics be useful?

Natural language: too complex



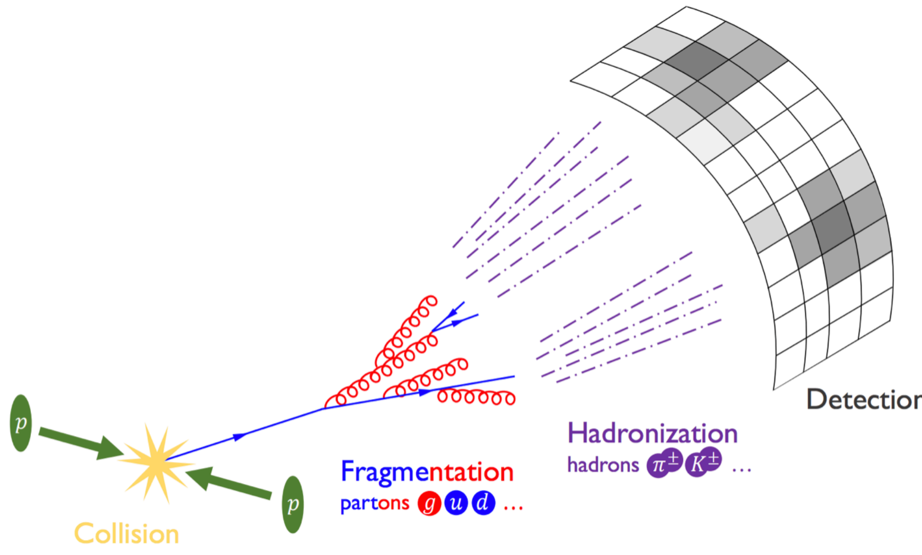
Gaussian data: too simple

Physics data (just right?)



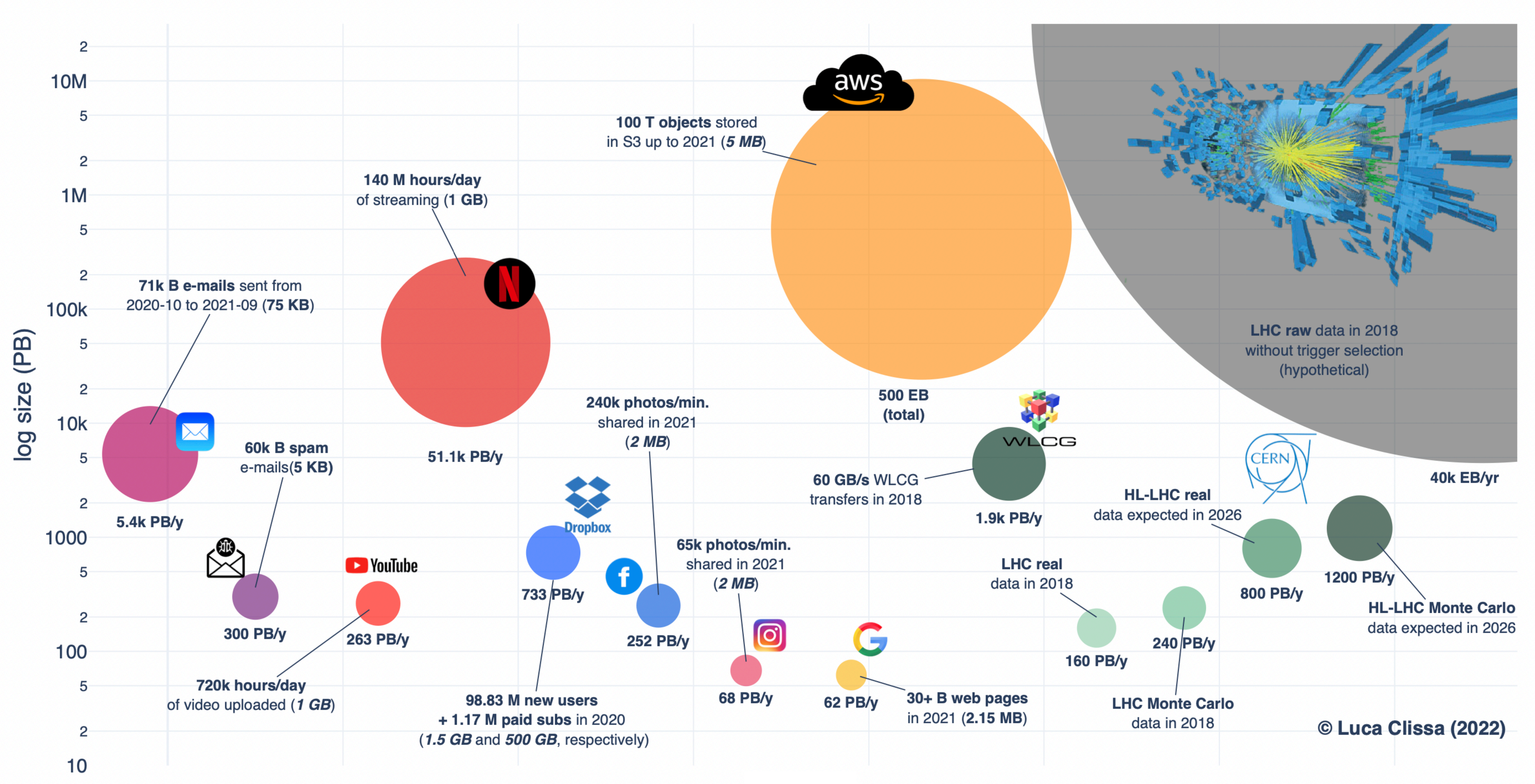
...dressed by high-dimensional “fluff”

low-dimensional core…



[Image credits: XKCD, Eric Metodiev]

Physics: the infinite-data limit

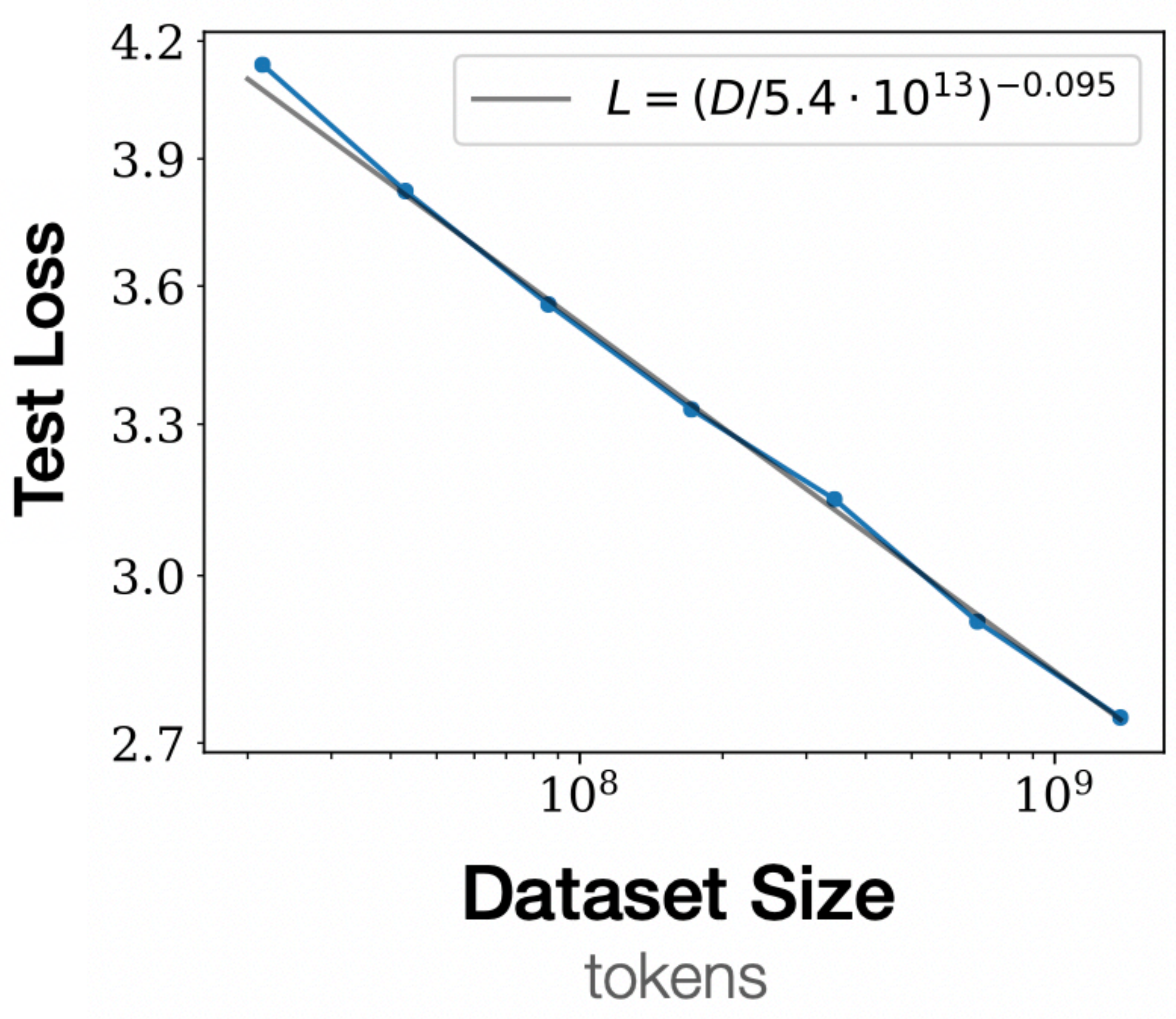


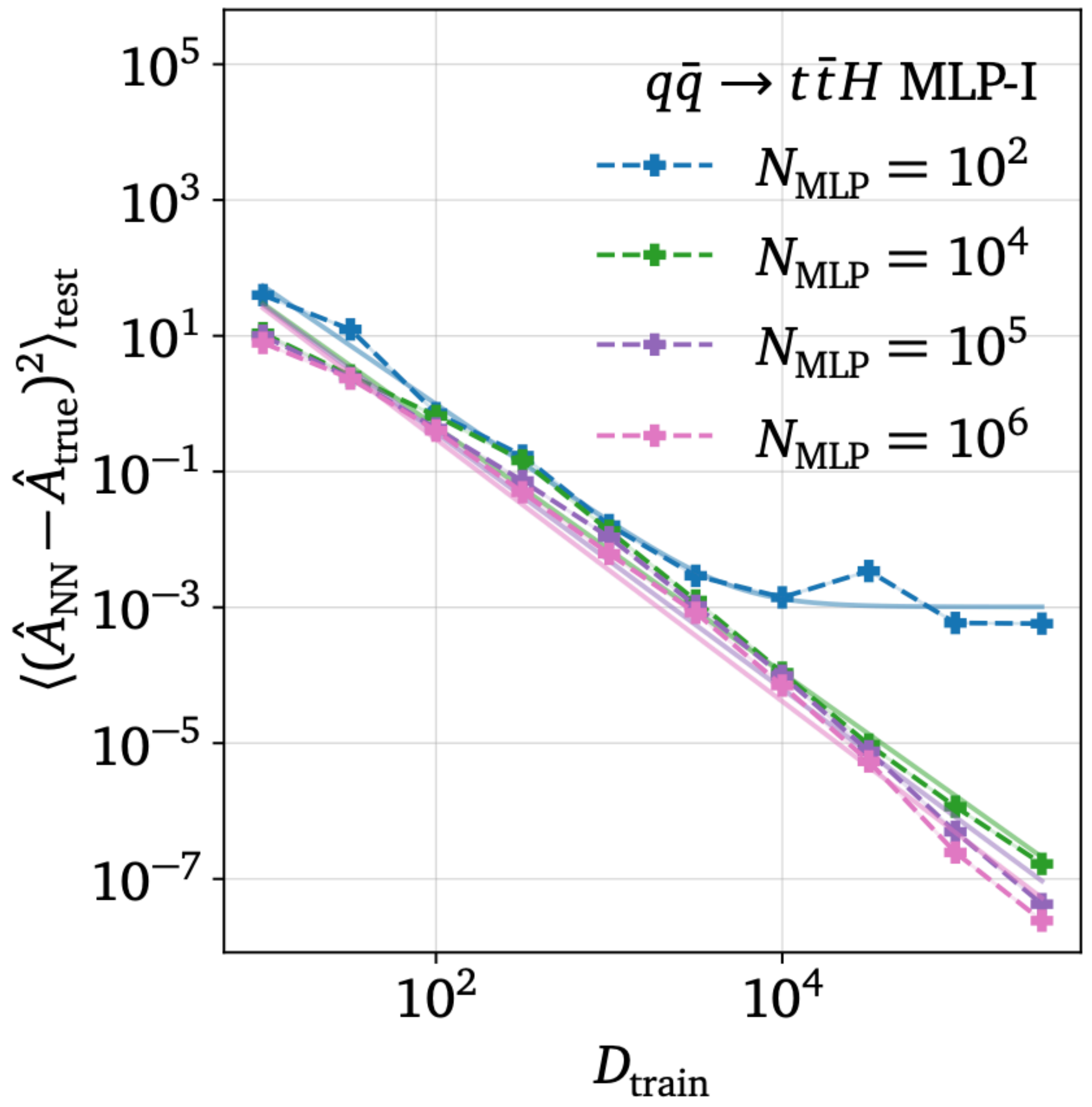
Simulated data is even more abundantly available! Physicists are really good at this

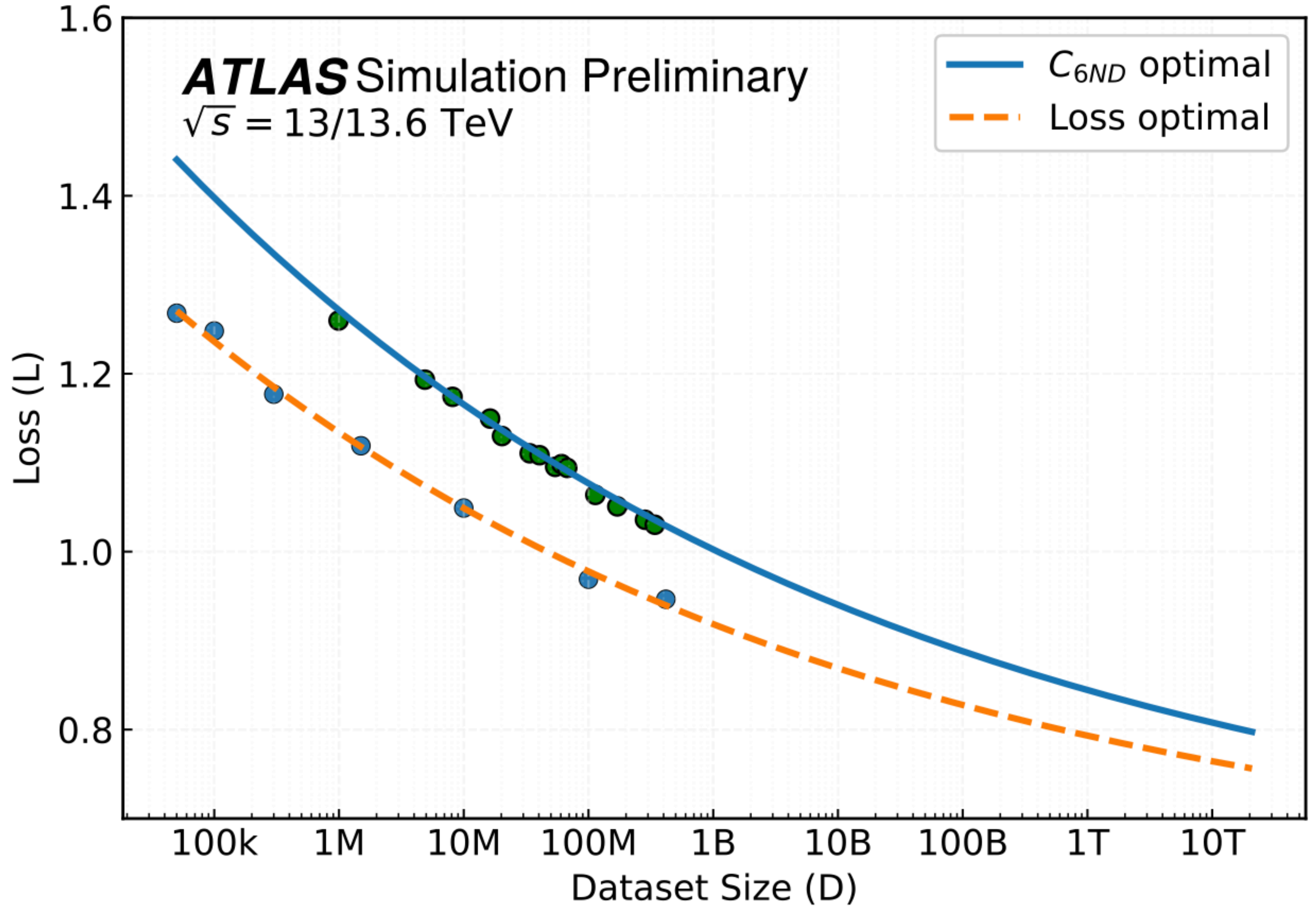
[L. Clissa, arXiv:2202.07659]

Physics for AI: origin of scaling laws

More data, parameters, compute time = better performance







first seen in large language models

now seen everywhere, including physics applications

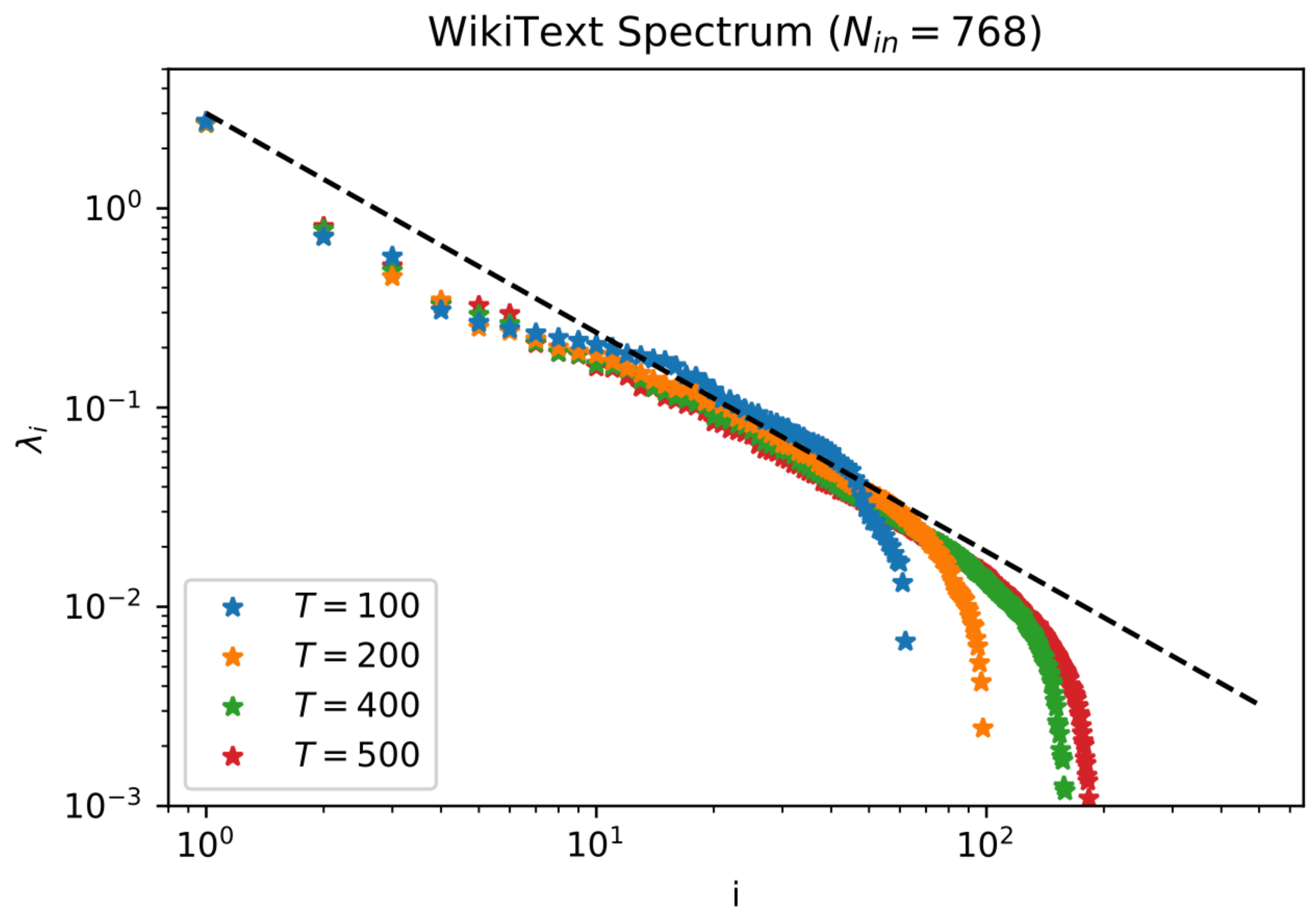
This power law is now driving an O(1) fraction of the US stock market.

What determines the slope?

[Kaplan, McCandlish, et al., arXiv:2001.08361; Bahl et al., arXiv:2601.13308; ATLAS PUB note ATL-SOFT-PUB-2026-002]

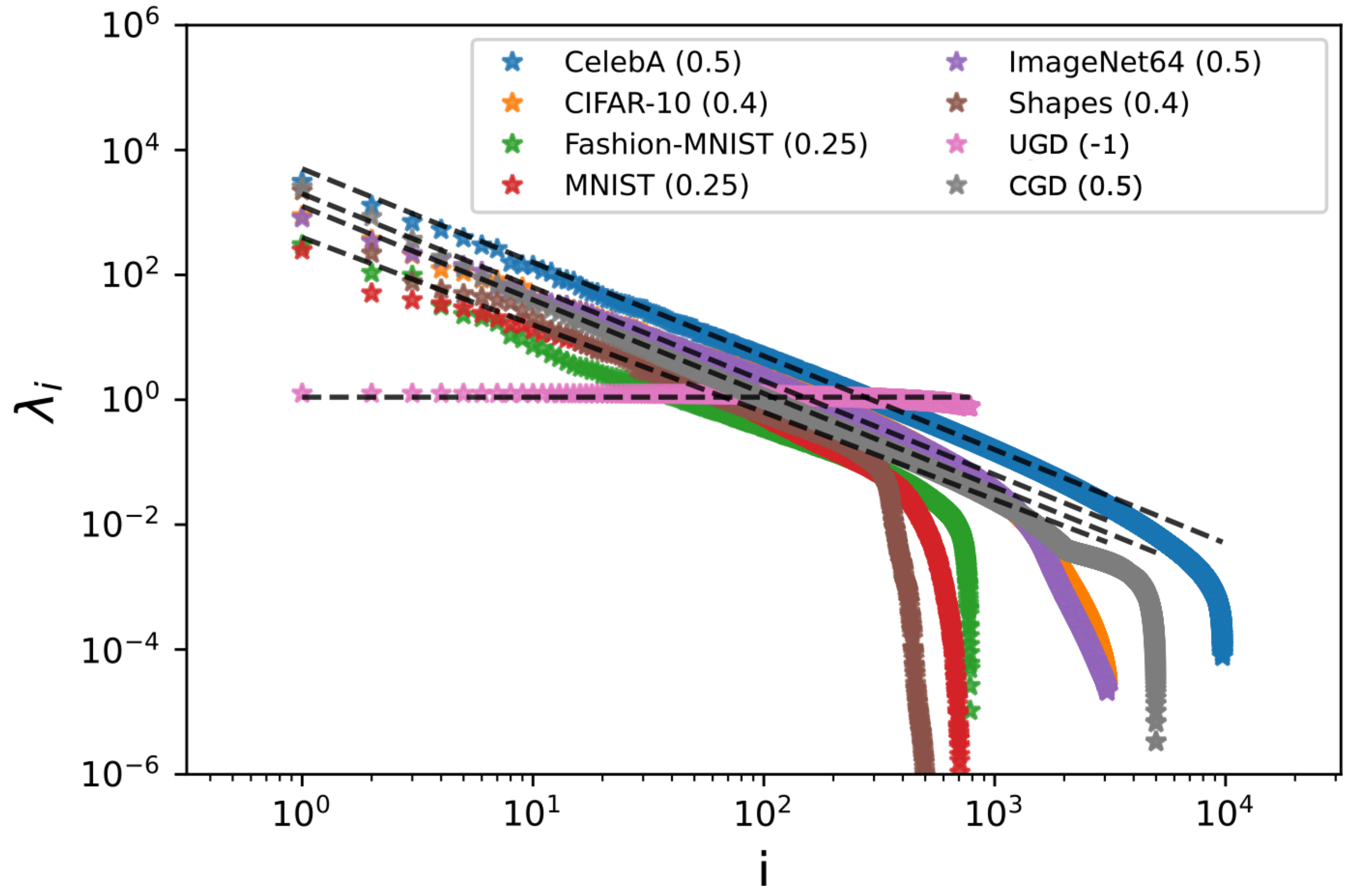
Toy model: high-dimensional regression

Empirical covariance matrix of natural data tends to have power-law eigenvalue spectrum



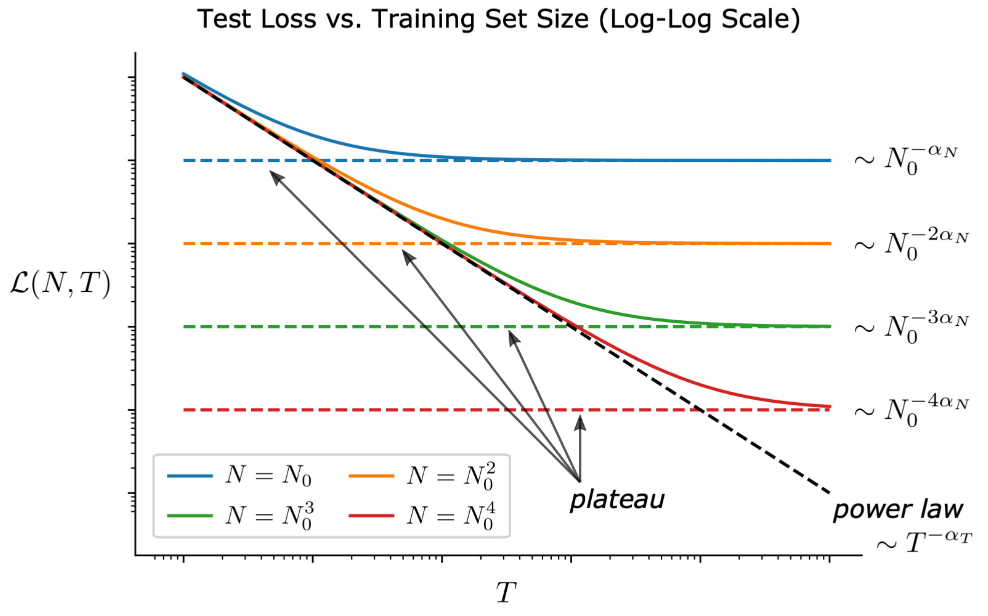
$$\Sigma_{\alpha\beta} = \vec{x}_\alpha \cdot \vec{x}_\beta$$

eigenvalues $\lambda_k \propto k^{-\alpha}$



Performing linear regression and averaging over the dataset with random matrix theory:

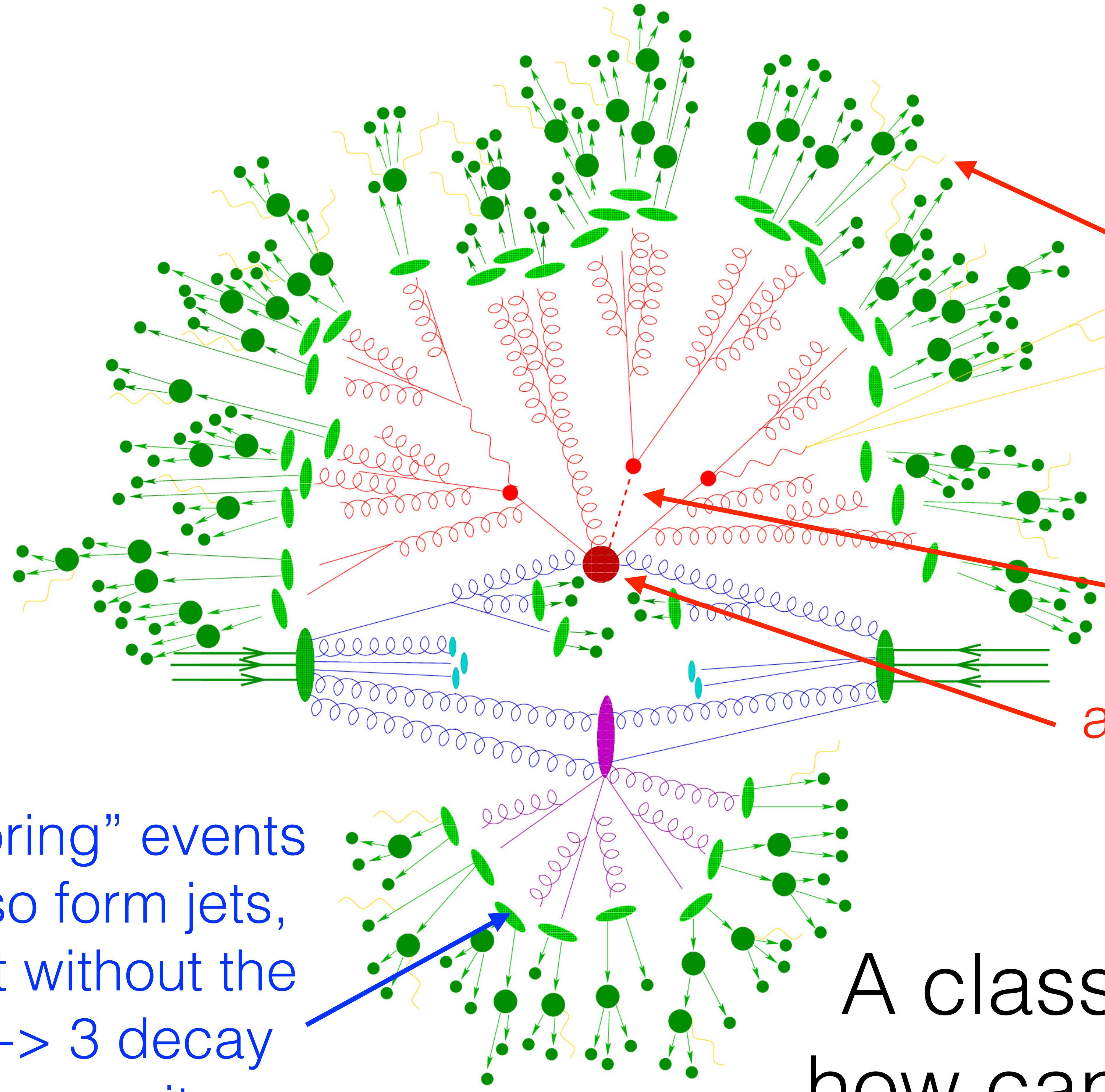
$$E_g \sim \begin{cases} \dfrac{N^{-2\alpha\min(r,1/2)}}{1-N/P}, & \alpha < l;\ N^{-2\alpha\min(r,1/2)} \gg \sigma_\epsilon^2 N/P & \text{Signal dominated} \\ \dfrac{N^{-2l\min(r,1/2)}}{1-N/P}, & l < \alpha;\ N^{-2l\min(r,1/2)} \gg \sigma_\epsilon^2 N^{l/\alpha}/P & \text{Ridge dominated} \\ \sigma_\epsilon^2 \dfrac{N}{P} & \alpha < l;\ N^{-2\min(\alpha,l)\min(r,1/2)} \ll \sigma_\epsilon^2 N/P & \text{Noise dominated} \\ \sigma_\epsilon^2 N^{l/\alpha}/P & l < \alpha;\ N^{-2\min(\alpha,l)\min(r,1/2)} \ll \sigma_\epsilon^2 N^{l/\alpha}/P & \text{Noise mitigated} \end{cases}$$



Fully solvable in terms of covariance power law index

[Maloney, Roberts, Sully, arXiv:2210.16859; Levi and Oz, arXiv:2306.14975; Atanasov, Zavatone-Veth, Pehlevan, arXiv:2405.00592]

AI for physics: learning jets



which eventually are detected as a spray of hundreds of particles (a jet)

decays into three others…

a single strongly-interacting particle…

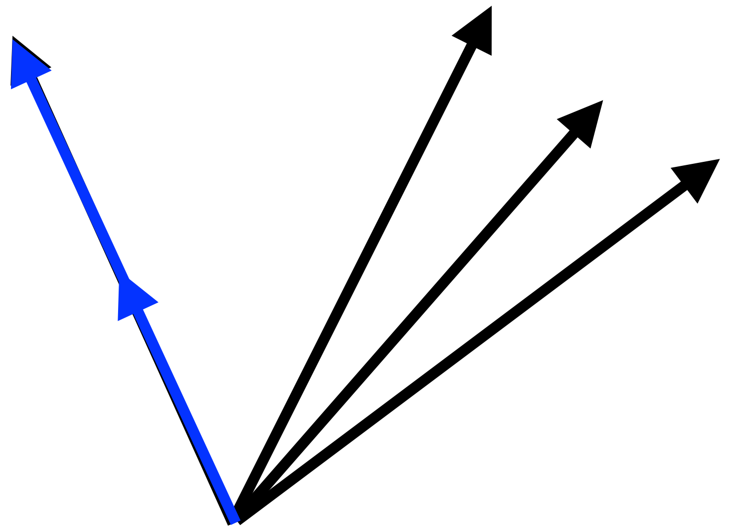
“boring” events also form jets, but without the 1 -> 3 decay progenitor

```
print(X[2][:,0:3])
```

```
[[ 0.21682942 -0.99705702  0.53256891]
 [ 0.2313594  -1.59192211  0.20290659]
 [ 0.3415725  -1.34588077  0.17991001]
 [16.82434455 -1.44341629  0.1983383 ]
 [ 3.88668157 -1.40565011  0.20535495]
 [ 1.48644645 -1.44413084  0.21409881]
 [ 9.3856368  -1.38781108  0.21308547]
 [19.86195738 -1.43333281  0.22595456]
 [ 1.15015947 -1.37975003  0.22350402]
 [ 9.62330424 -1.51326457  0.25372503]
 [11.09433991 -1.42098188  0.23969255]
 [ 0.62370459 -1.57651711  0.34558    ]
 [ 4.64977756 -1.44984434  0.28598831]
 [ 2.61376518 -1.40445959  0.27877904]
 [ 2.00224566 -1.4019      0.28213292]
 [ 3.31956822 -1.54040698  0.33040497]
 [ 1.34283334 -1.49964736  0.82643411]
 [ 4.42455039 -1.47730468  0.3387453 ]
 [ 1.1745804  -1.12311379  0.5969829 ]
 [ 0.12070414 -1.15907443  0.65859174]
 [ 0.10445965 -1.15730904  0.53805857]
 [ 0.27883771 -1.54480515  0.53203865]
 [ 0.74119026 -1.21943139  0.54154753]
 [ 1.02516475 -1.31617244  0.43983998]
 [ 0.40229068 -1.51899693  0.55556759]
 [ 0.52385688 -1.3505163   0.71414581]
 [ 2.54897776 -1.35765739  0.46719657]
 [ 1.09042071 -1.26333935  0.60527719]
 [ 0.71729668 -1.26732054  0.59983503]
 [ 0.13870153 -1.48712285  0.58126095]
 [ 6.58555777 -1.30438334  0.51994192]
 [ 0.43662182 -1.32084426  0.50975518]
 [ 0.68491052 -1.46491663  0.61566227]
 [15.25639084 -1.32019095  0.55301947]
 [ 9.60192677 -1.34177416  0.55731791]
 [17.25212412 -1.34202677  0.55728155]
 [ 1.89835954 -1.34675898  0.62160968]
 [ 7.93255976 -1.36902236  0.54372975]
 [16.54838134 -1.36051182  0.55187618]
 [ 3.60274988 -1.34467532  0.59952559]
 [20.74558186 -1.34706257  0.60579601]
 [15.35961558 -1.34678192  0.59190014]
 [ 5.5243889  -1.36198034  0.55557767]
 [ 7.54390784 -1.37840486  0.55684761]
```

A classic ML for physics application: how can we best distinguish these two classes of events?

The manifold of jet data



a zero-energy ("infrared") particle
might as well not be there

two collinear particles
might as well be one

$$\mathcal{M}_{4-\text{particle}} \simeq S^8 \supset S^5$$

"How many particles" is not a well-defined question in perturbative QFT!

$$\mathcal{M}_{\text{jet}} \sim S^2 \subset S^5 \subset S^8 \cdots \subset S^{596} \subset \cdots$$

Squared matrix elements are
probability distributions on phase space

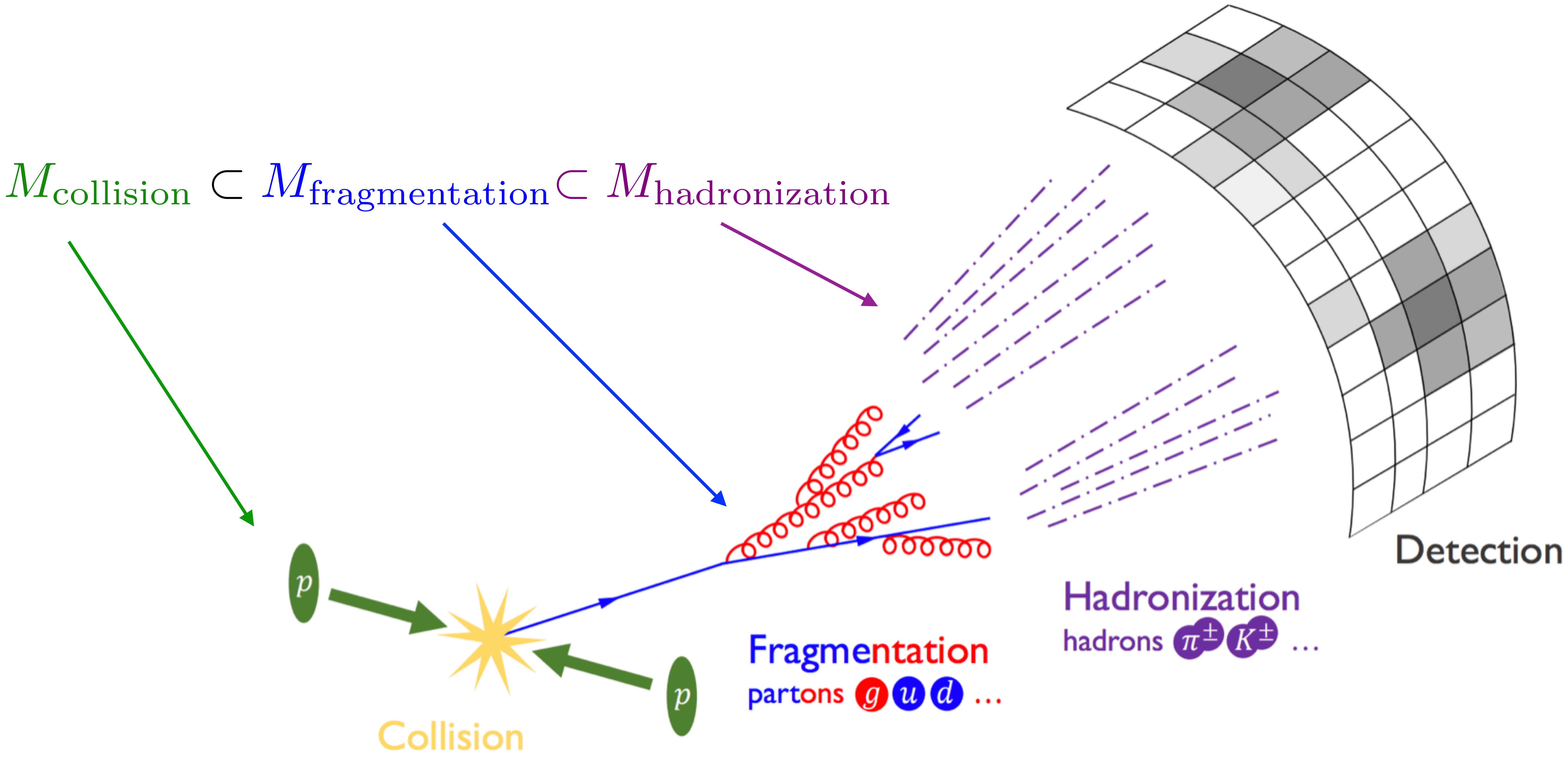
$$|\mathcal{M}(q \to \bar{q}_1' q_2' q_3)|^2 \propto -\frac{m_W^2}{m_t^2}\left(\frac{2z_1}{z_1+z_2}\frac{m_t^2-m_W^2}{m_W^2} - \frac{2s_{13}}{m_W^2} + \frac{z_1-z_2}{z_1+z_2}\right)^2 + \frac{4+(z_1-z_2)^2}{z_1+z_2} + z_1 + z_2 - \frac{m_W^2}{m_t^2} - 4$$

can compute features on the
full manifold as long as
they are "infrared and collinear (IRC) safe"

$$\mathrm{EFP}_G = \sum_{i_1=1}^{M} \cdots \sum_{i_N=1}^{M} z_{i_1} \cdots z_{i_N} \prod_{(k,\ell)\in G} \theta_{i_k i_\ell}$$

[Larkoski, arXiv:2411.00104; Komiske, Metodiev, Thaler, arXiv:1712.07124]

The benefits of calculability



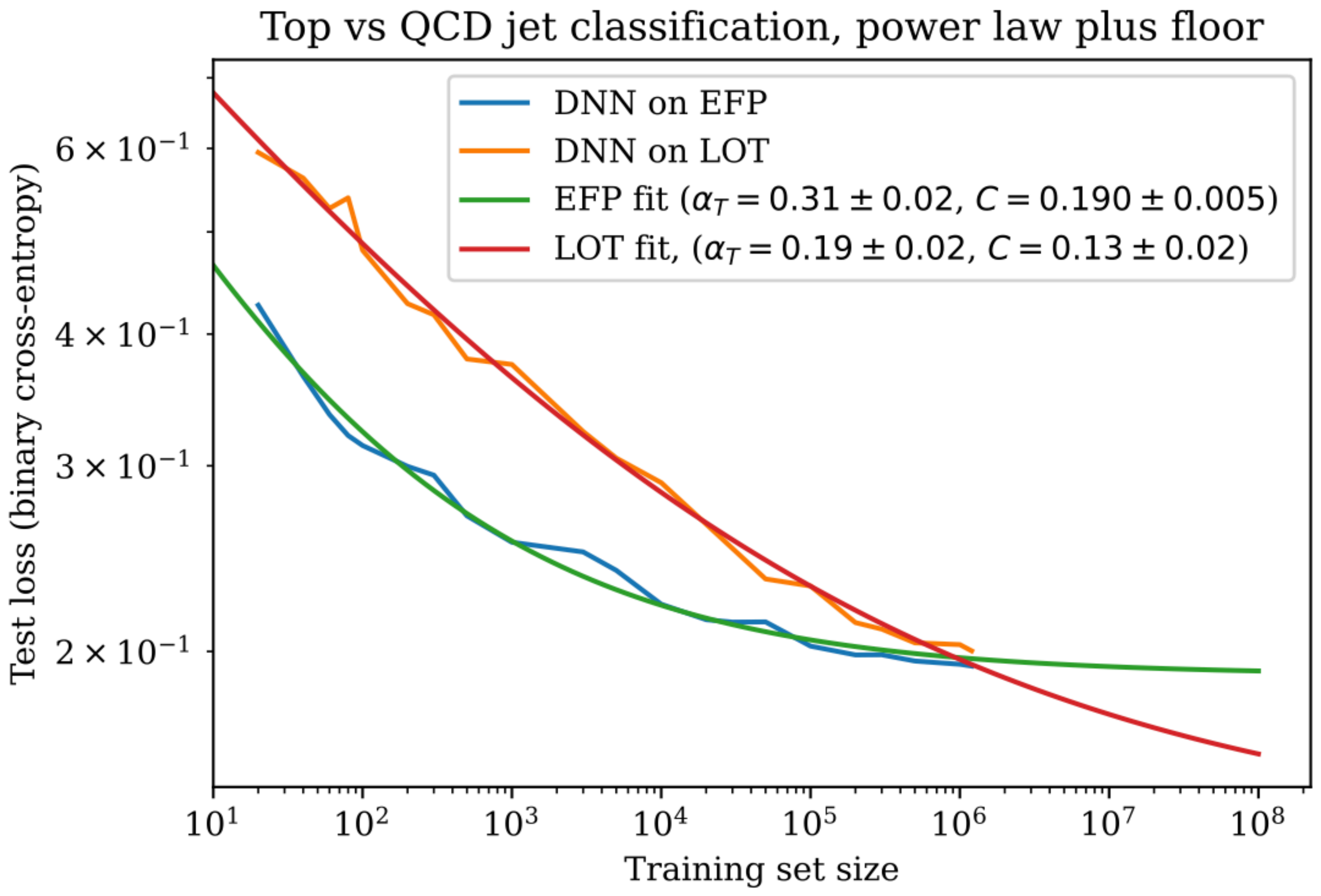
can't calculate $p(\text{images})$

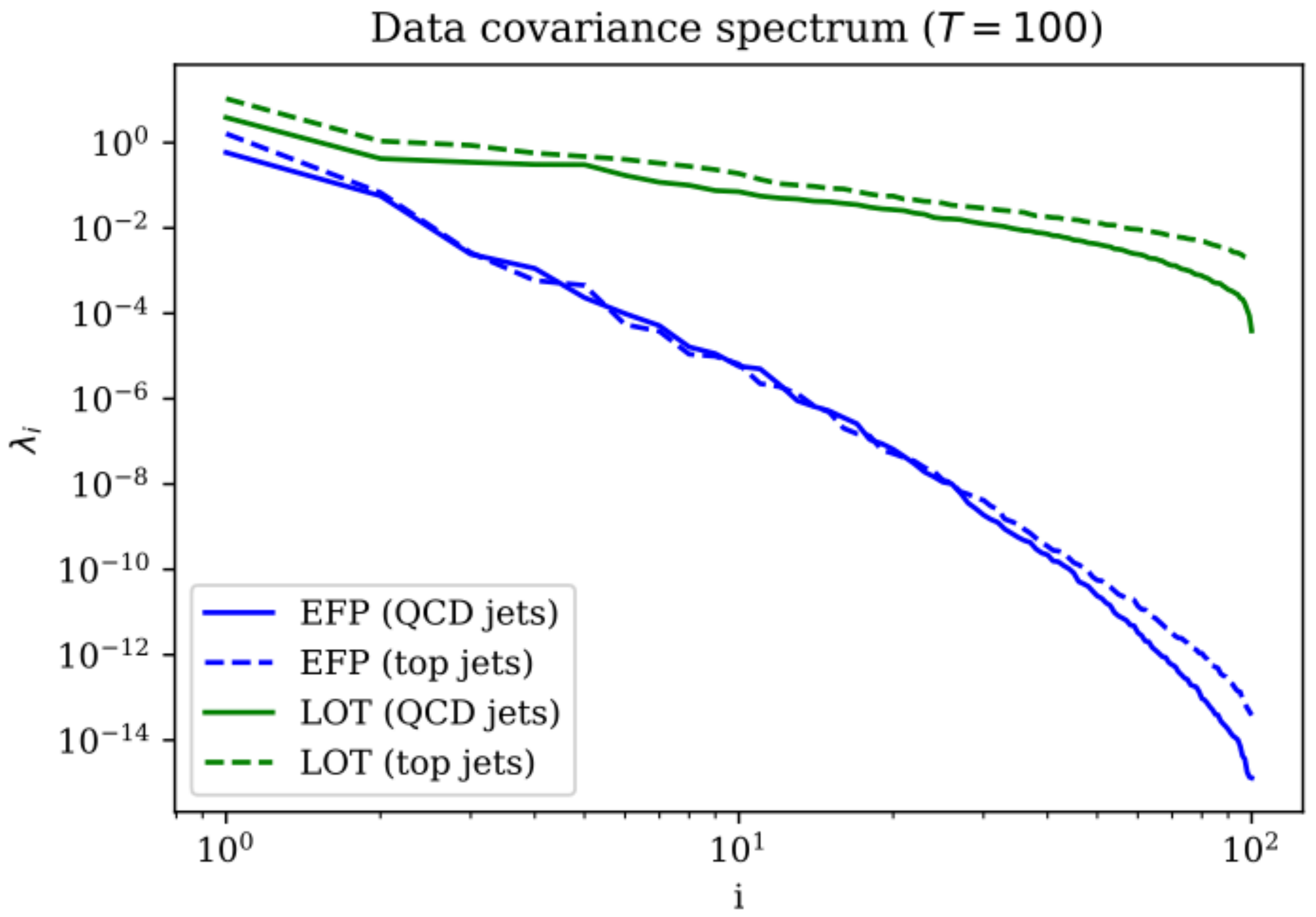
can calculate $p(\text{jets})$ (perturbatively)

We know the data manifold ($\mathscr{M}_{N\text{−particles}} \sim S^{3N-4}$), there are natural embeddings, and QFT tells us how to compute distributions on it

[Image credits Nano Banana Pro; E. Metodiev]

Jet scaling laws



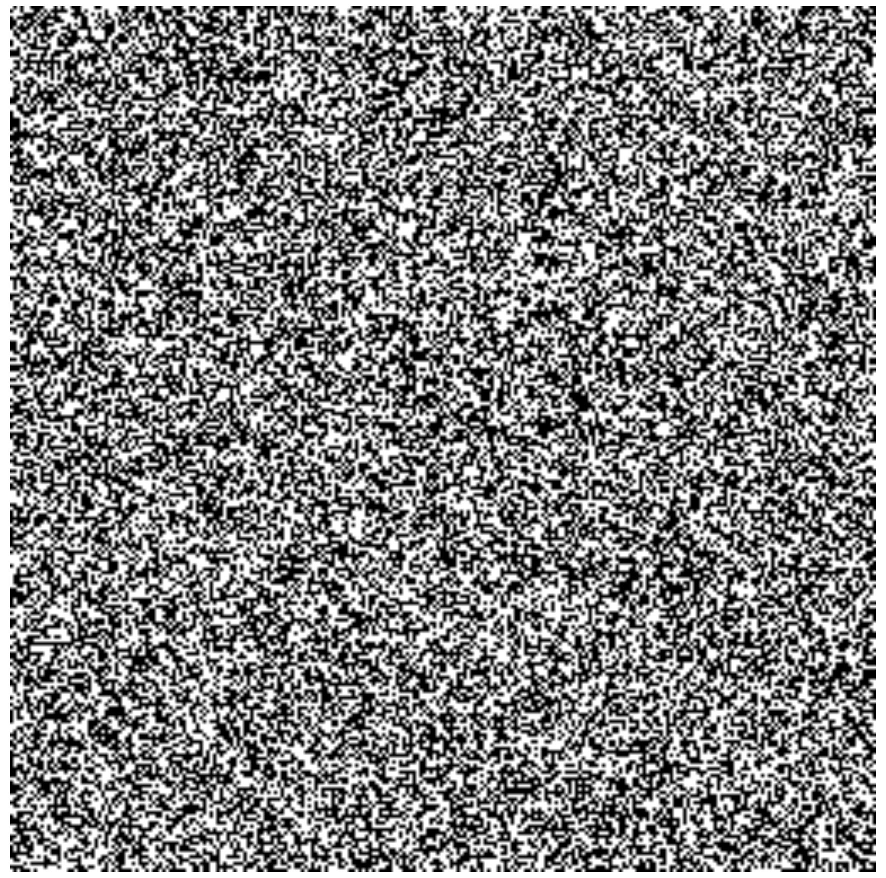


Ongoing work: **compute** data covariance spectrum in simulated data from first principles, for different processes and parameters, relate to scaling laws in the loss

[Batson, **YK**, arXiv:2312.02264; Humphrey, **YK,** Larkoski, Rasovic, in prep]

HEP for AI: jets as a model of data

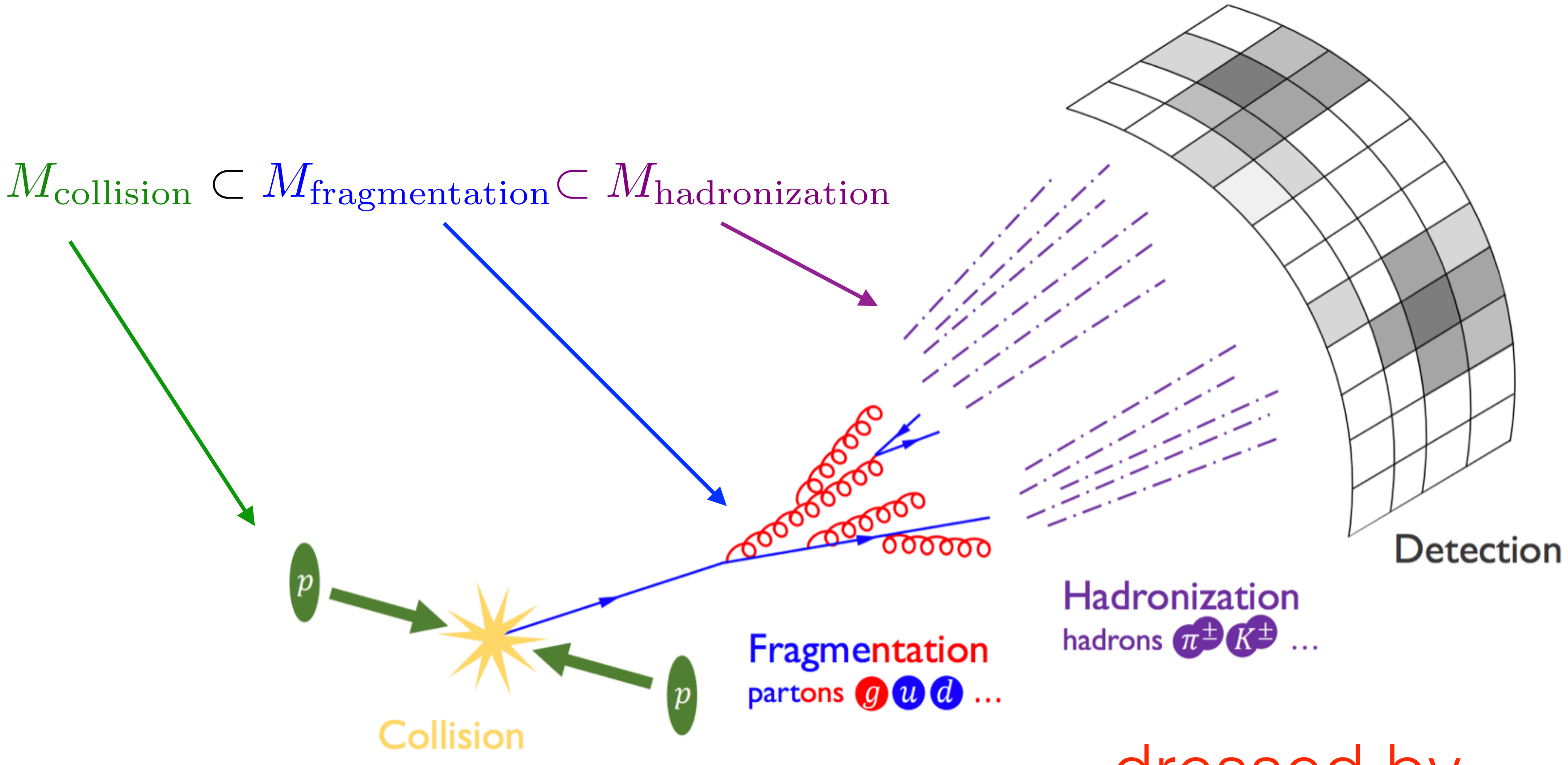
Random data:



2^{256} possible images,
$2^{2^{256}}$ image/binary label pairs

Real-world data:

Occupies a vastly smaller subspace of the 2^{256}-dimensional data manifold

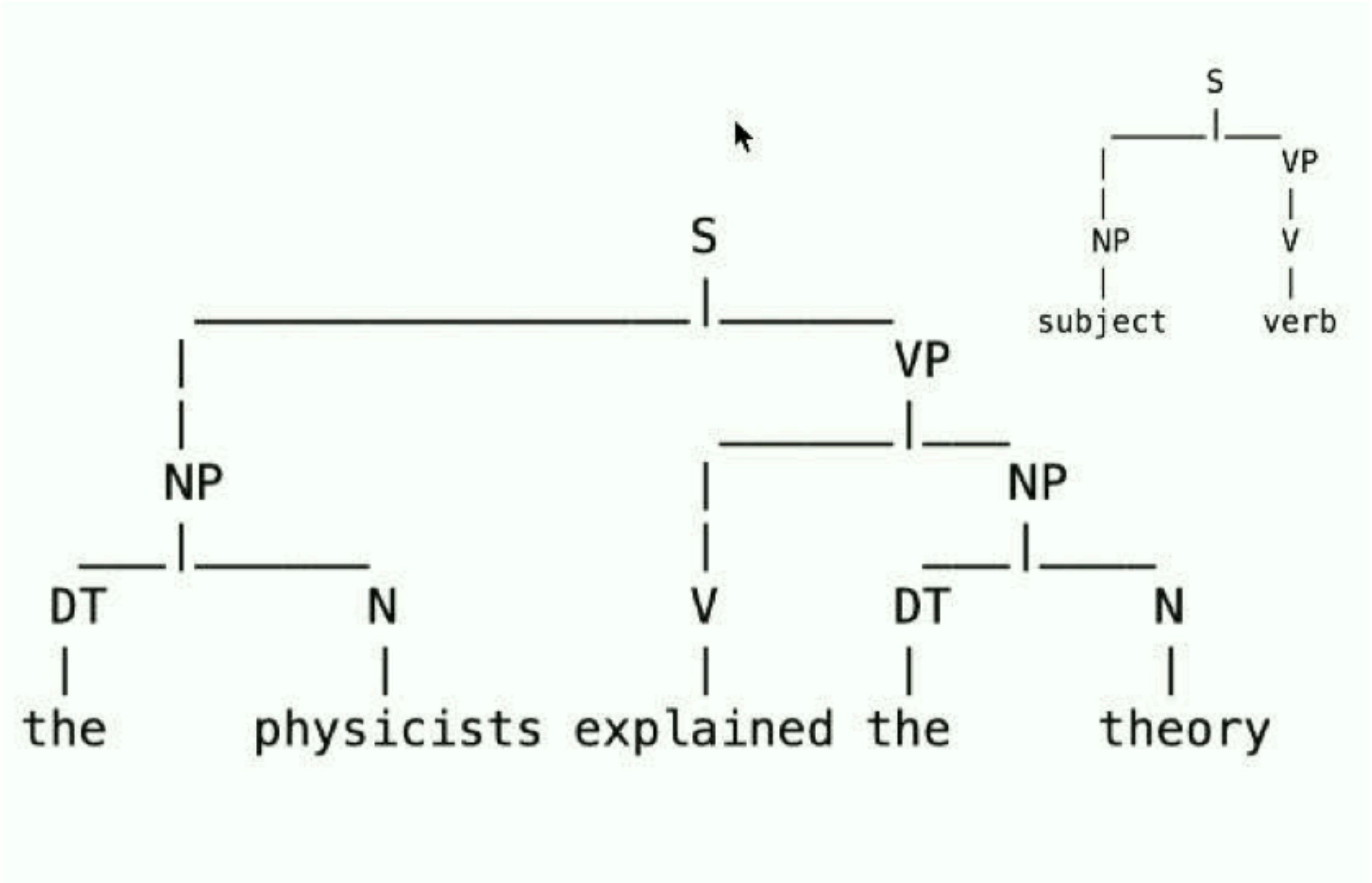


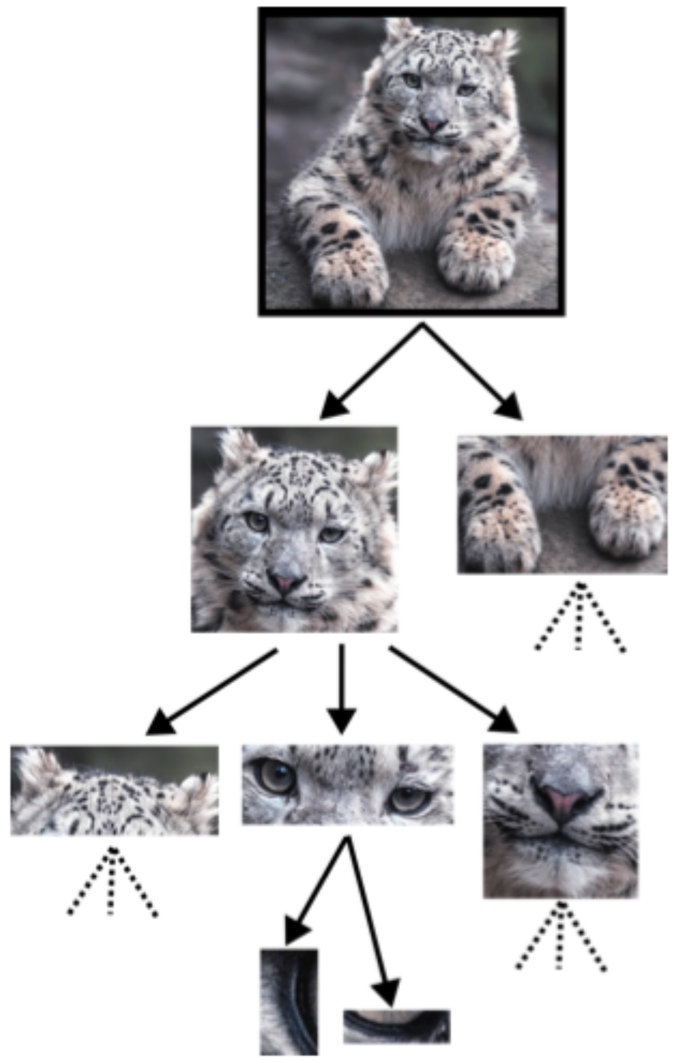
low-dimensional core

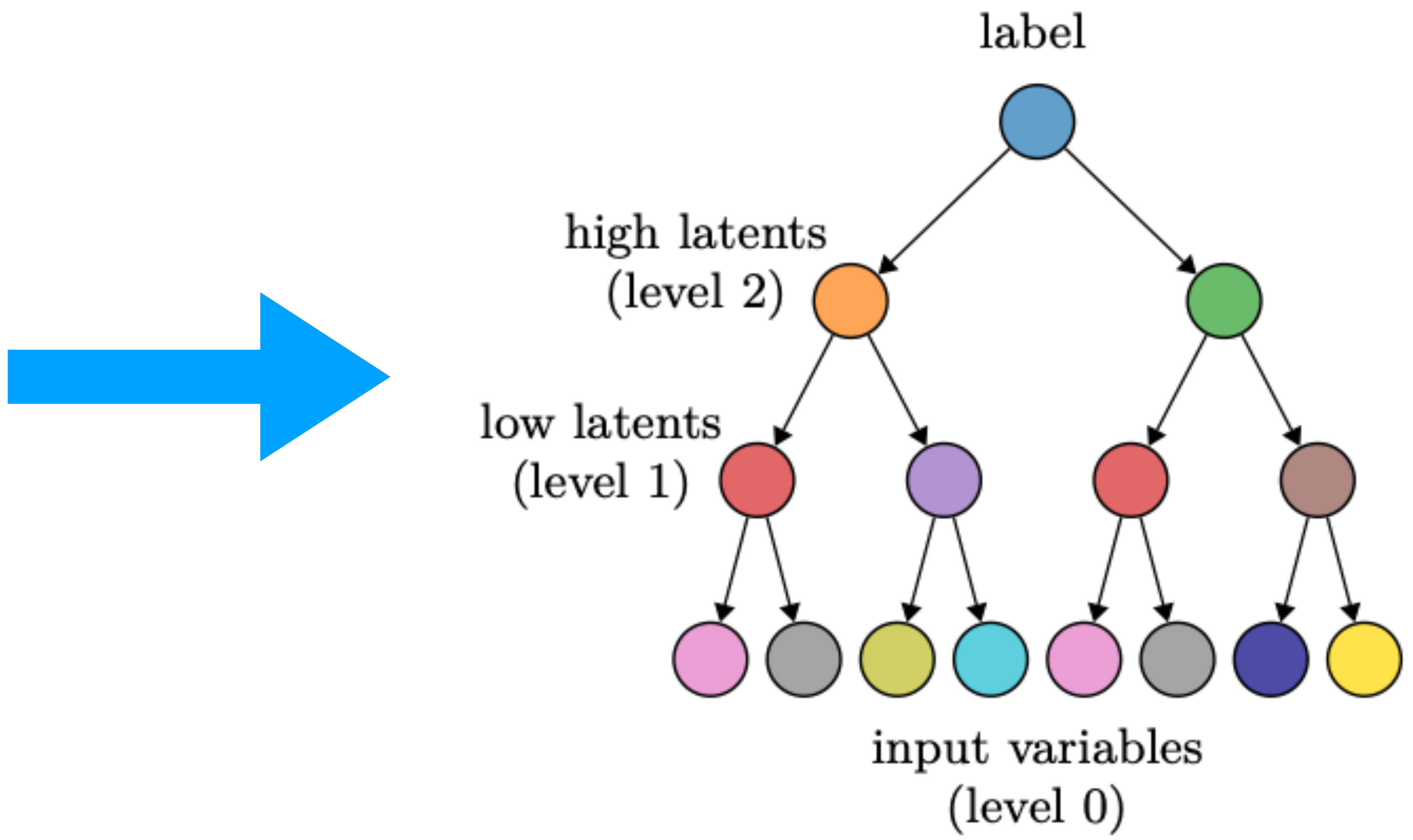
dressed by high-dimensional fluff

Other excellent examples: condensed matter physics, climate physics, cosmology, …
physics is full of potential “models of data”!

Toy model: random hierarchy







For vocabulary size v, splittings s, m rules per symbol, each input sequence is associated to a unique tree if $m \leq v^{s-1}$

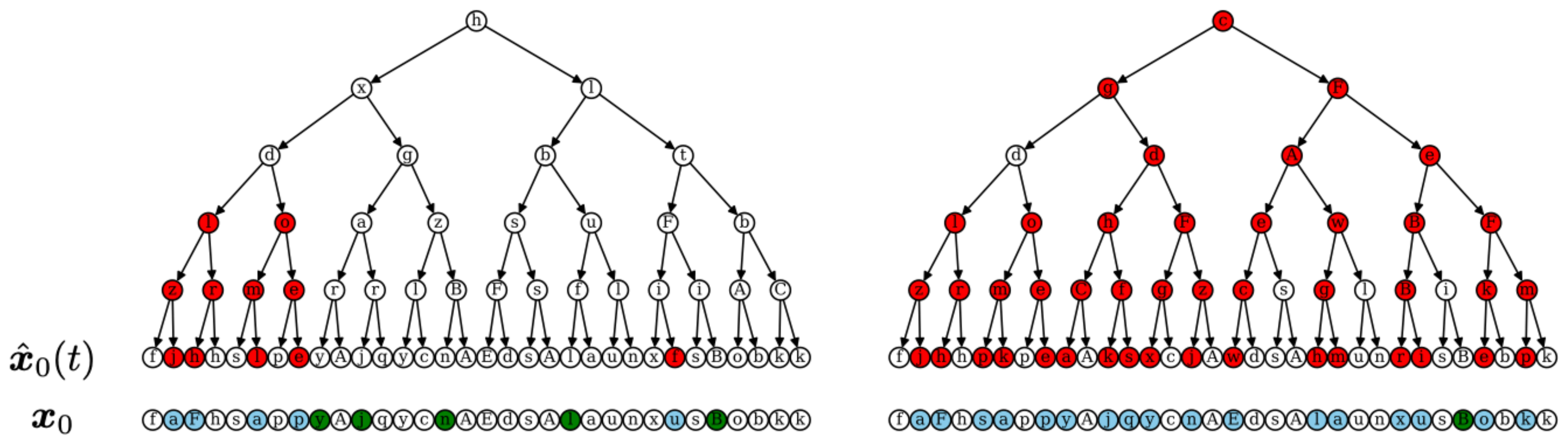
Exponentially many input sequences, but allowed sequences are lower-dimensional subset

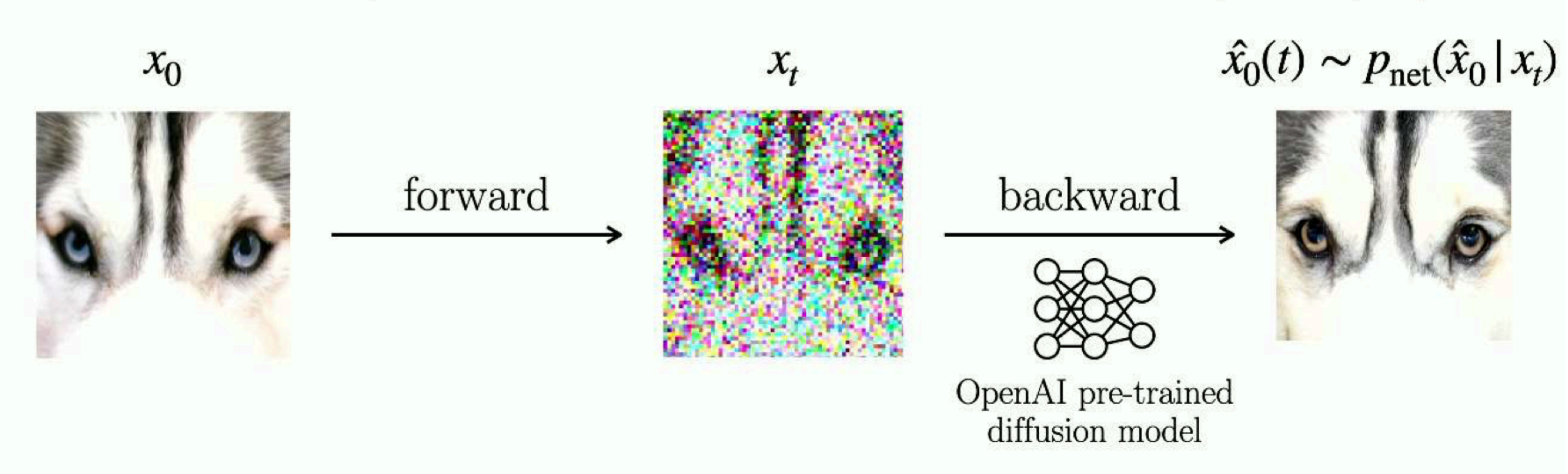
[Cagnetta et al., arXiv:2307.02129; Schlocchi, Favero, Wyart, arXiv:2402.16991; Sclocchi, Favero, Levi, Wyart, arXiv:2410.13770; A. Favero, PI Theory+AI workshop April 2025]

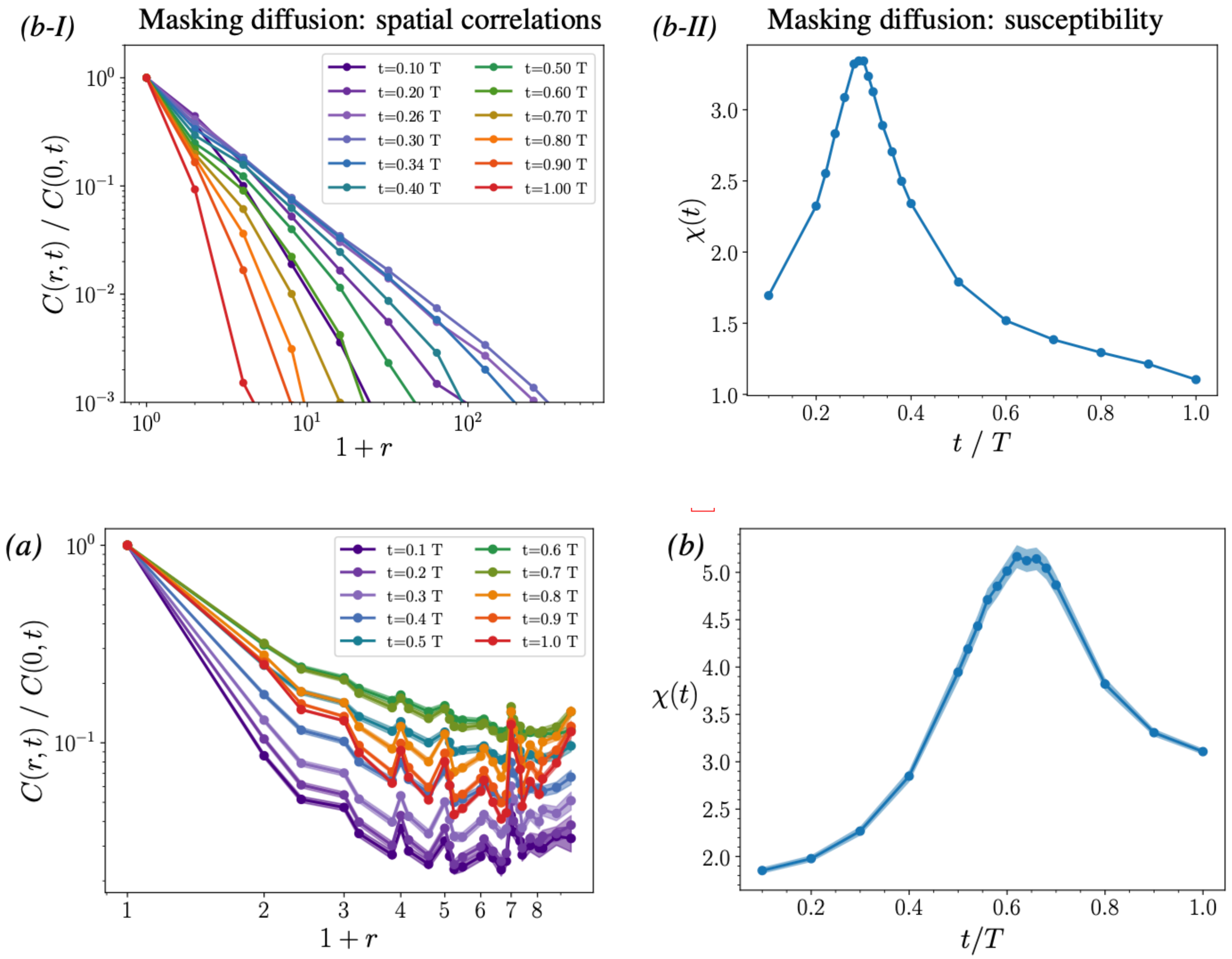
Diffusion models detect hierarchical structure

A diffusion model is a neural network trained to de-noise a dataset

(this is the basis for most AI image generators)







Spatially-correlated changes propagate to give a susceptibility peak, but only for hierarchical data

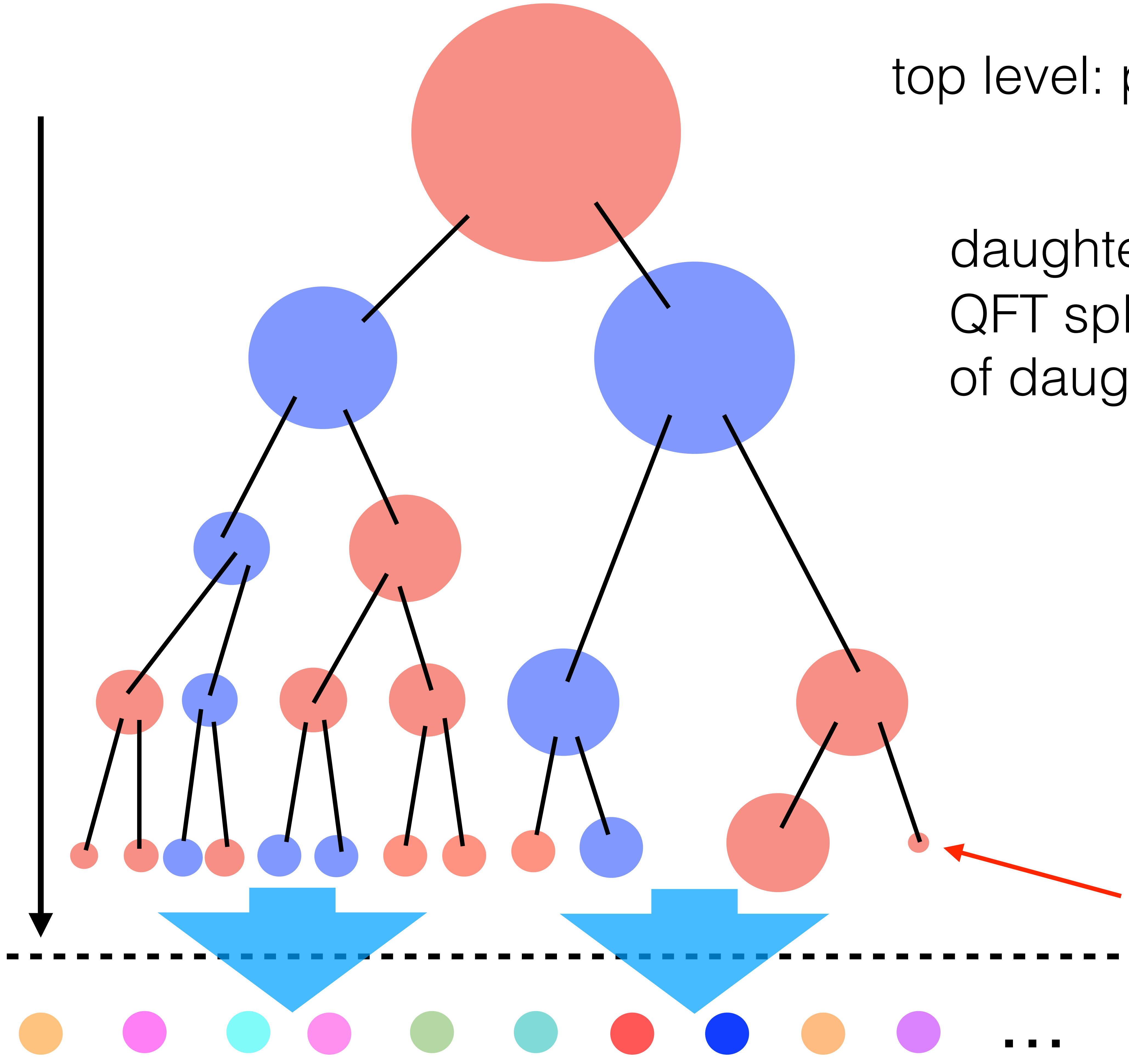
[A. Favero, PI Theory+AI workshop April 2025; Sclocchi, Favero, Levi, Wyart, arXiv:2410.13770]

Jets have hierarchical structure

time t
(a.k.a. virtuality)

energies of particles decrease, splitting probability $\alpha(t)$ increases

$\alpha(t) = 1$: hadronization



top level: progenitor particle

daughter momenta drawn from $p(\vec{q}\,|\,\alpha, t)$. QFT splitting functions govern identity of daughters

truncating tree with N particles in layer gives data manifold of dimension 3N-4

zero-momentum particles vanish from IRC-safe observables

~200 particles at lowest level

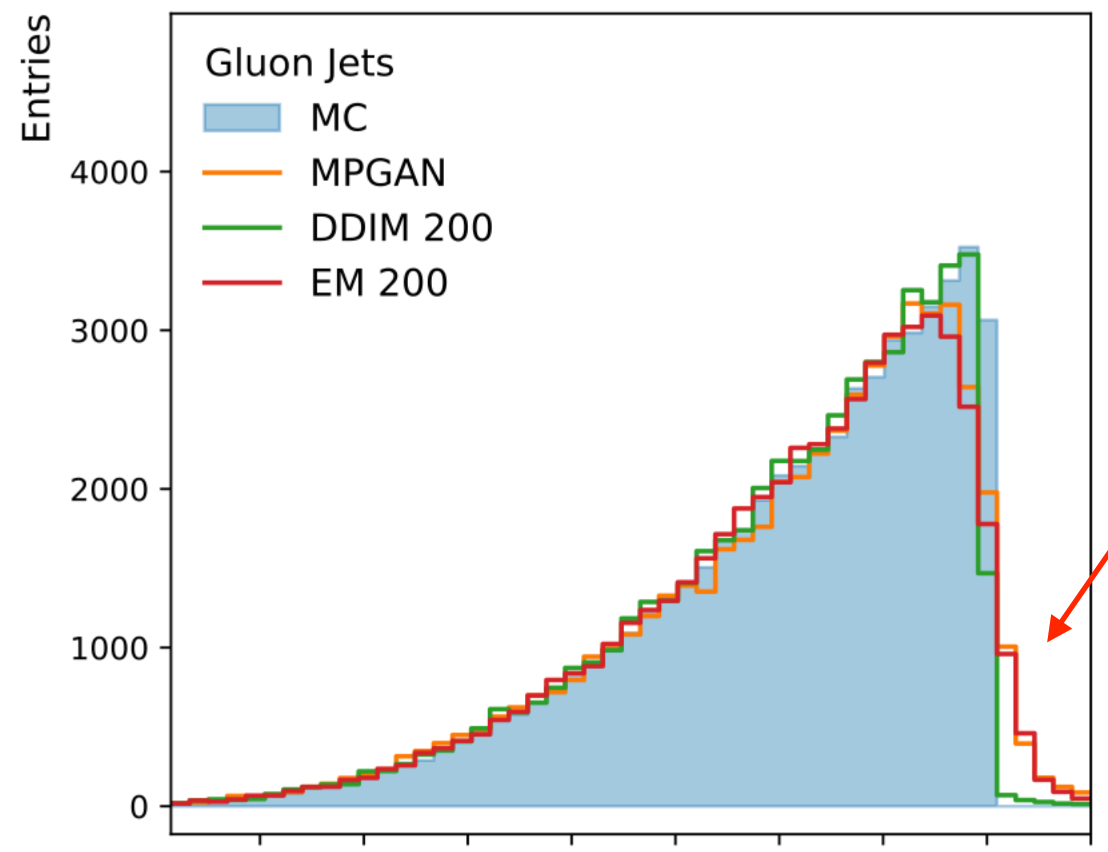
Correlations encode class label, first few splittings, latent variable α, symmetries, conservation laws

Very first steps: diffusion on phase space

If we want to exploit everything we know about physics, we had better make sure that a generative model conserves energy and momentum and respects Lorentz invariance

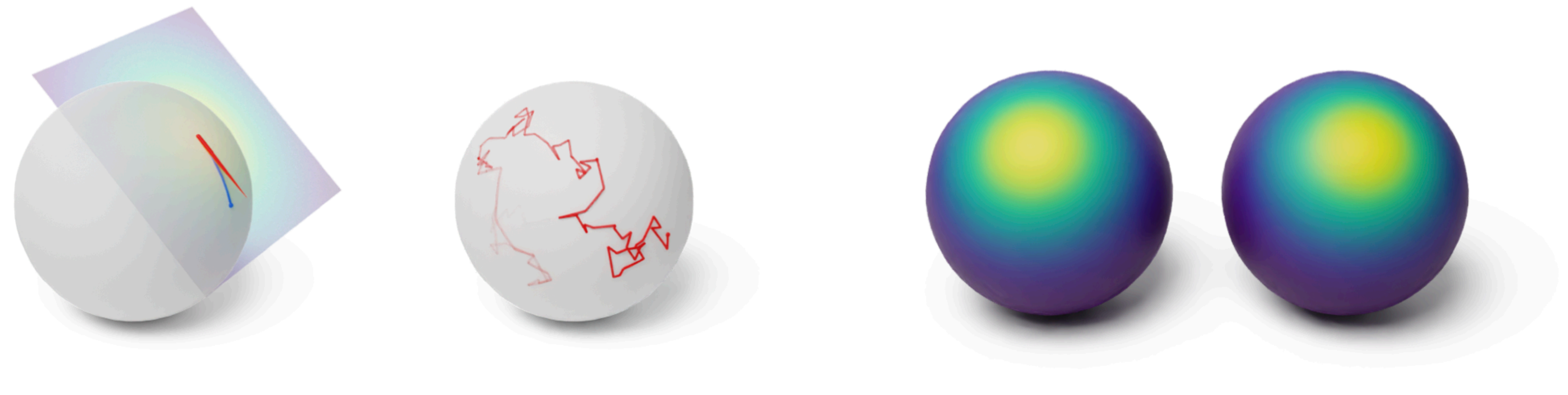
$$d\Pi_N = \left(\prod_{I=1}^{N} \delta^{(4)}(p_I^2 - m_I^2)\Theta(p_I^0) \right) \delta^3 \left(\sum_{I=1}^{N} \mathbf{p}_I \right) \delta \left(\left\{ \sum_{I=1}^{N} p_I^0 \right\} - 1 \right)$$

“Vanilla” unconstrained flat-space diffusion:



tail violates energy conservation!

Riemannian diffusion?



(a) A single step of a Geodesic Random Walk.

(b) Many steps yield an approximate trajectory.

(c) Gaussian Random Walk [Left] and the Brownian motion density [Right] agree well for small time steps.

Beautiful idea in principle,
fails numerically in practice for massless particles

[Leigh et al., arXiv:2303.05376; De Bortoli et al., arXiv:2202.02763]

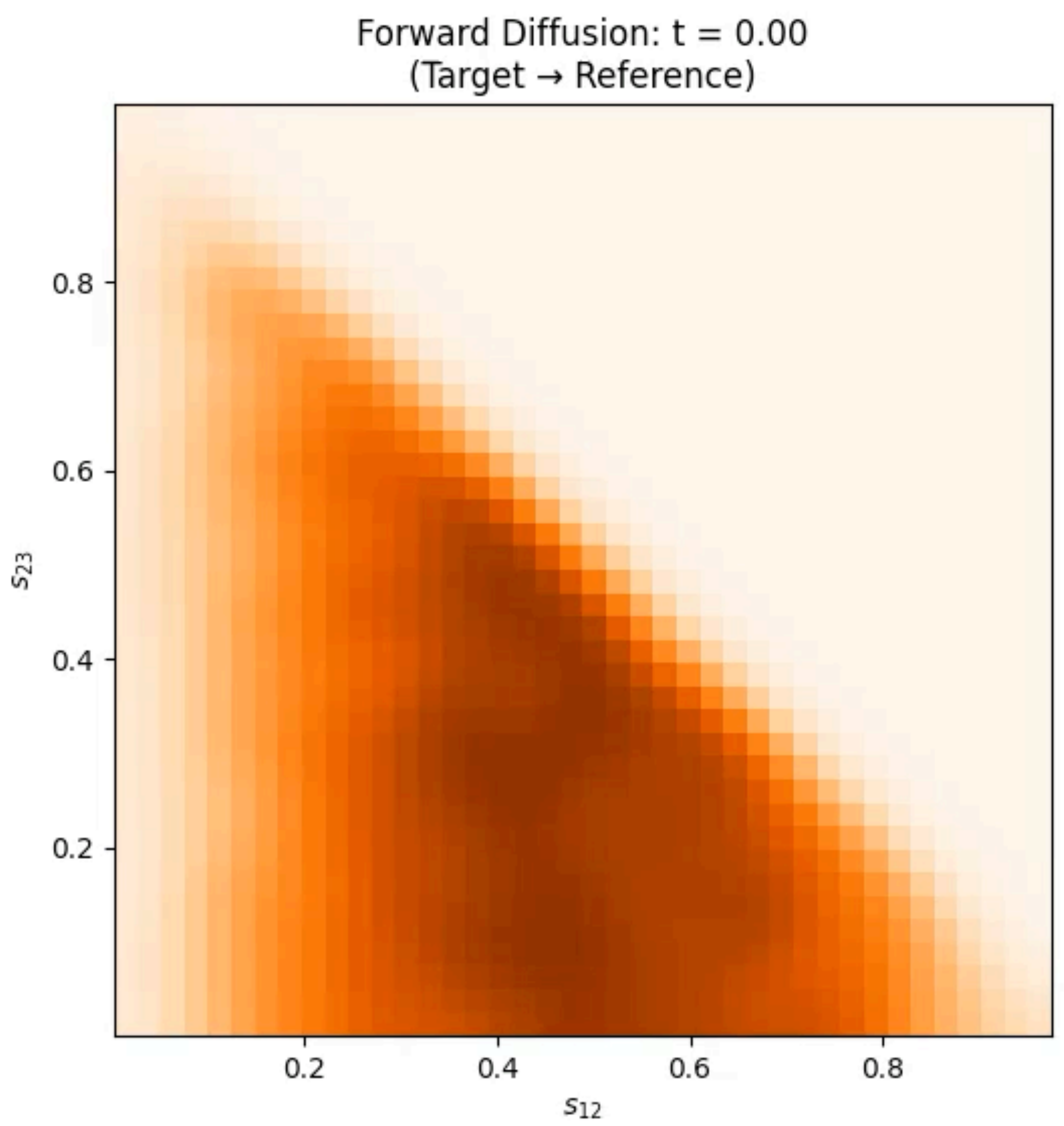
Ancient physics tricks: auxiliary space diffusion

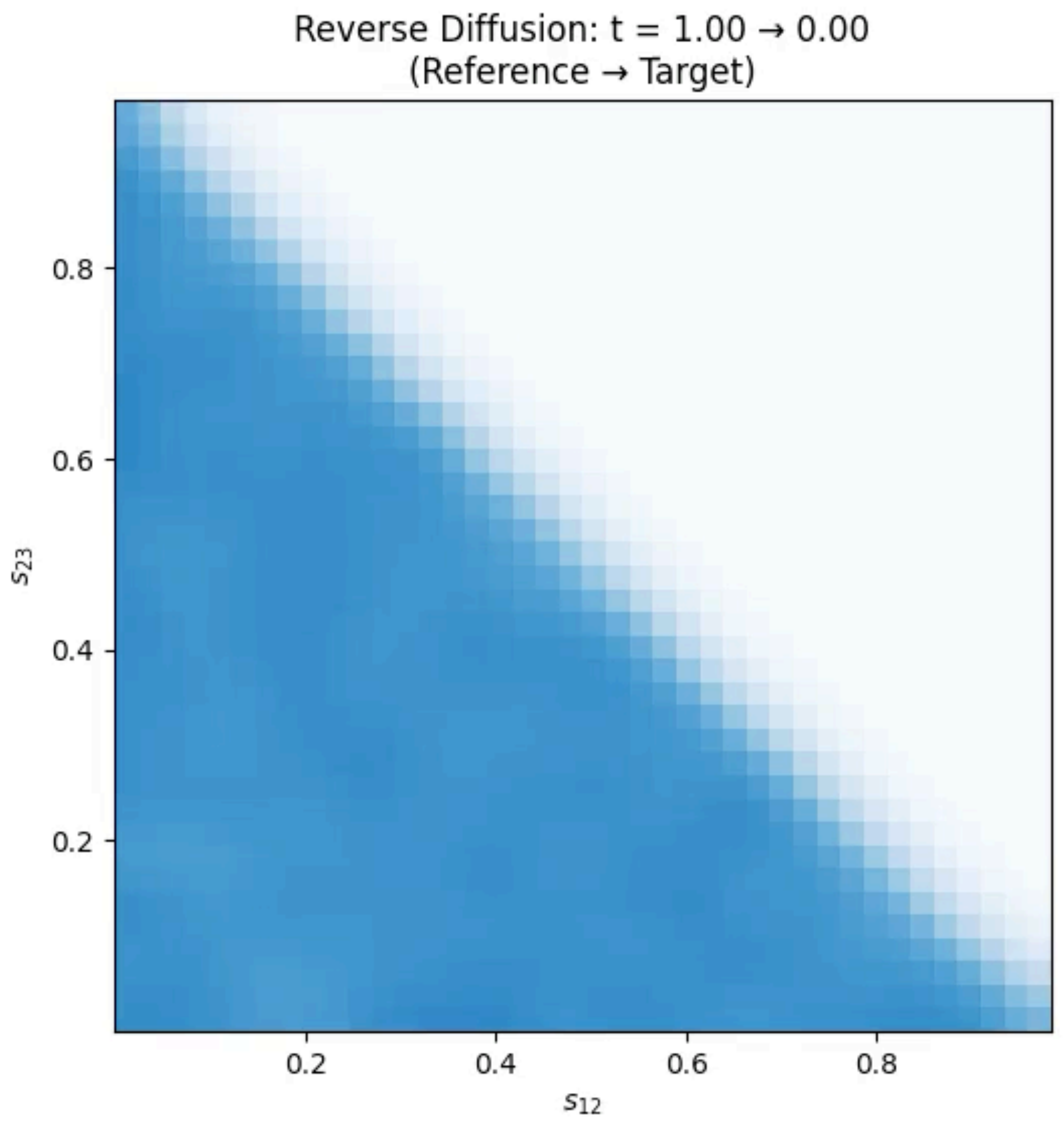
Uniform w.r.t. nonlinear constraints $\Longrightarrow$ non-uniform with no constraints

$\{\mathbf{p}_I\} =$ Lorentz transformation to CM frame, rescale energies $\Longleftarrow$ $p_{\text{ref}}(\{\mathbf{q}_I\}) = \prod_{I=1}^{N} \frac{e^{-q_I}}{q_I}$

This algorithm (RAMBO) is how massless phase space has been sampled since the 1980's!

forward process: targets p_{ref} in q-space, gives uniform distribution in phase space as "pure noise"





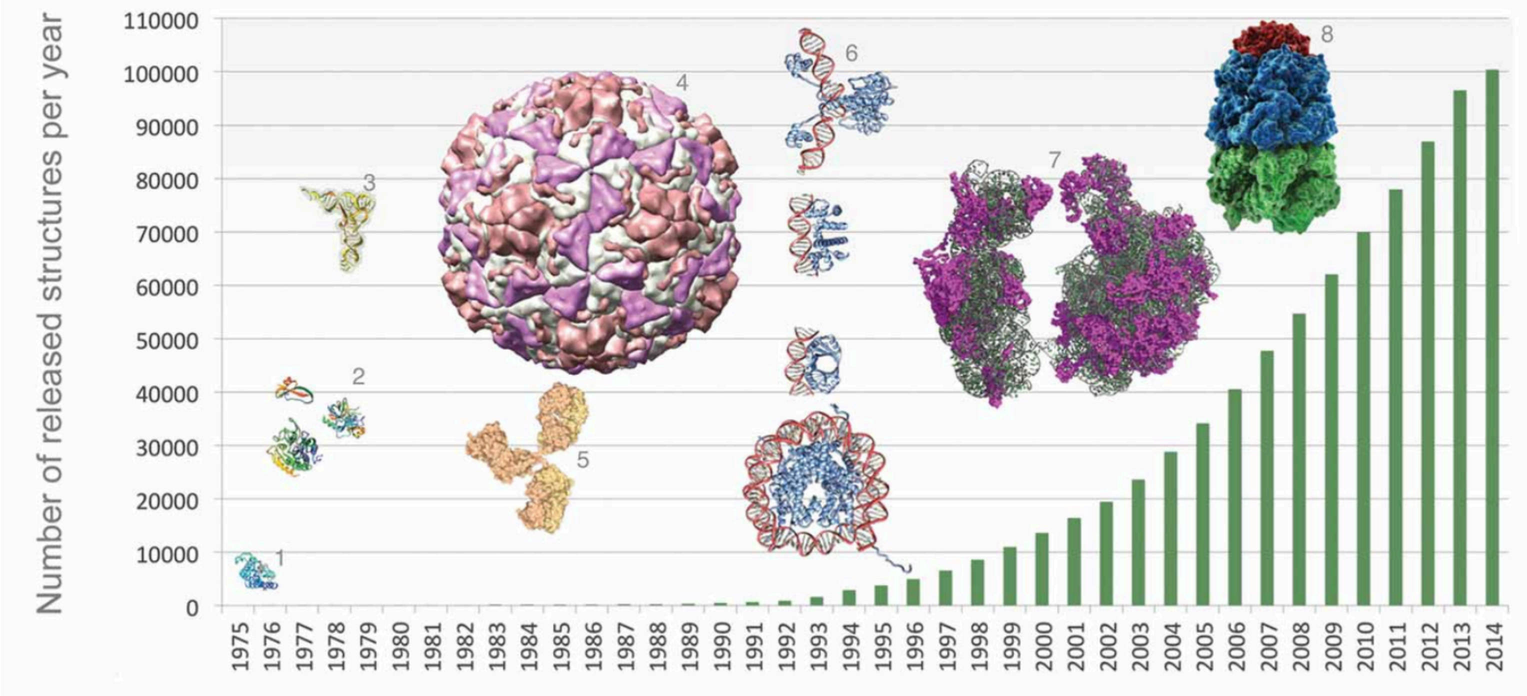
reverse process: sample pure noise, recover target distribution

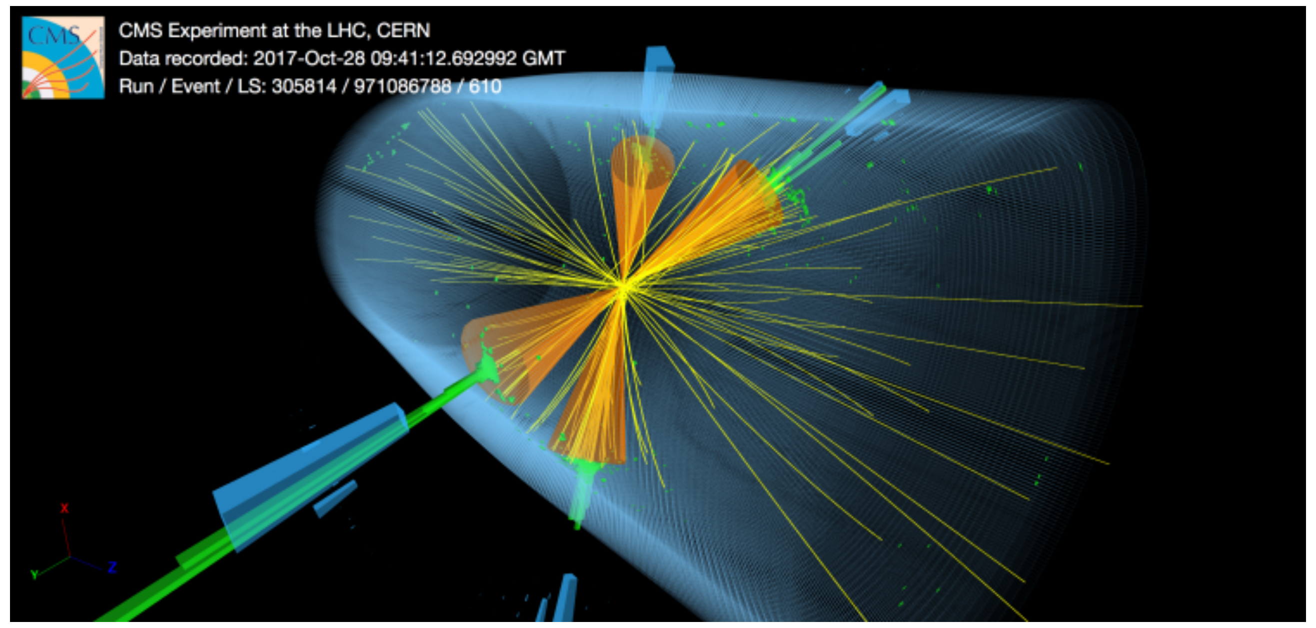
[Kleis, Stirling, Ellis, Comput. Phys. Commun. (1986); Bogorad, Elsharkawy, **YK**, Larkoski, Levi, arXiv:2602.xxxxx]

Back to AI for HEP

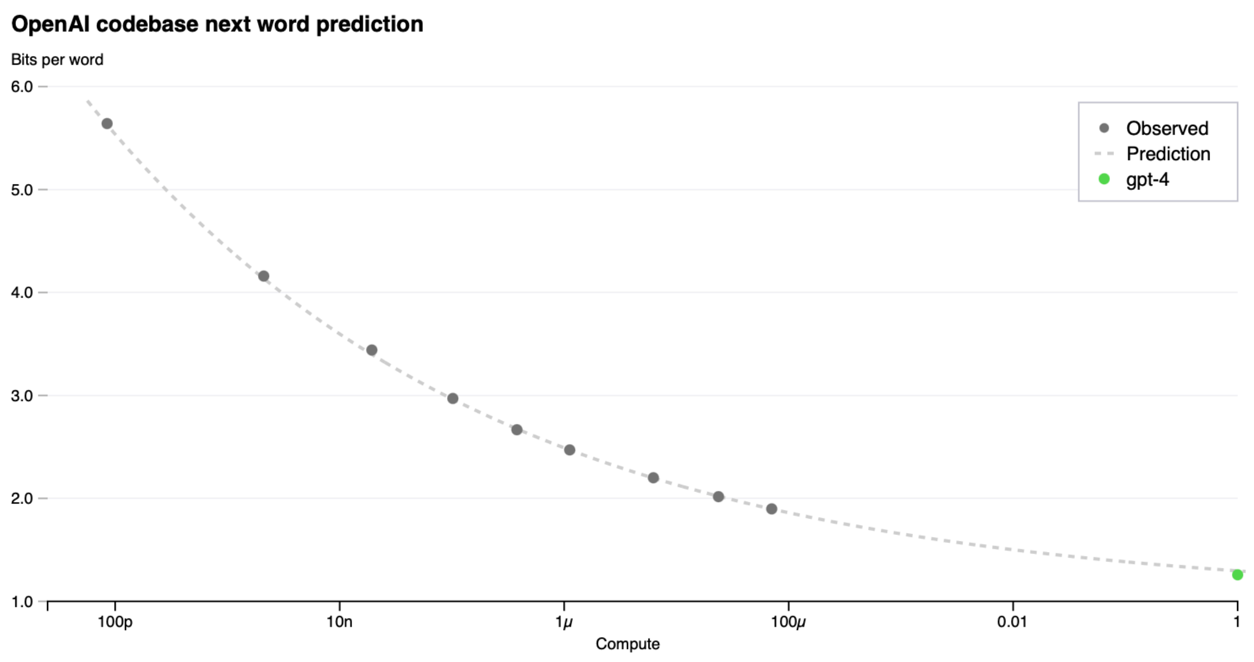
Lessons from industry:

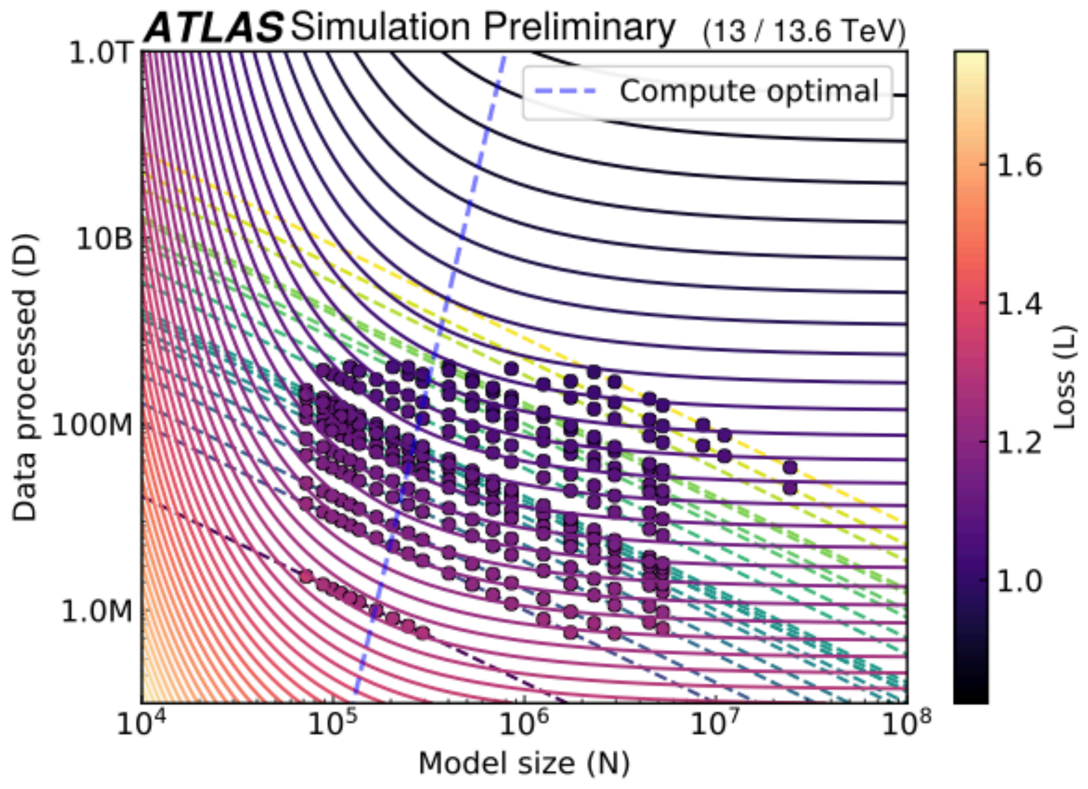
High-quality data with hierarchical structure is key





Scaling data, model size, and compute predictably improves performance





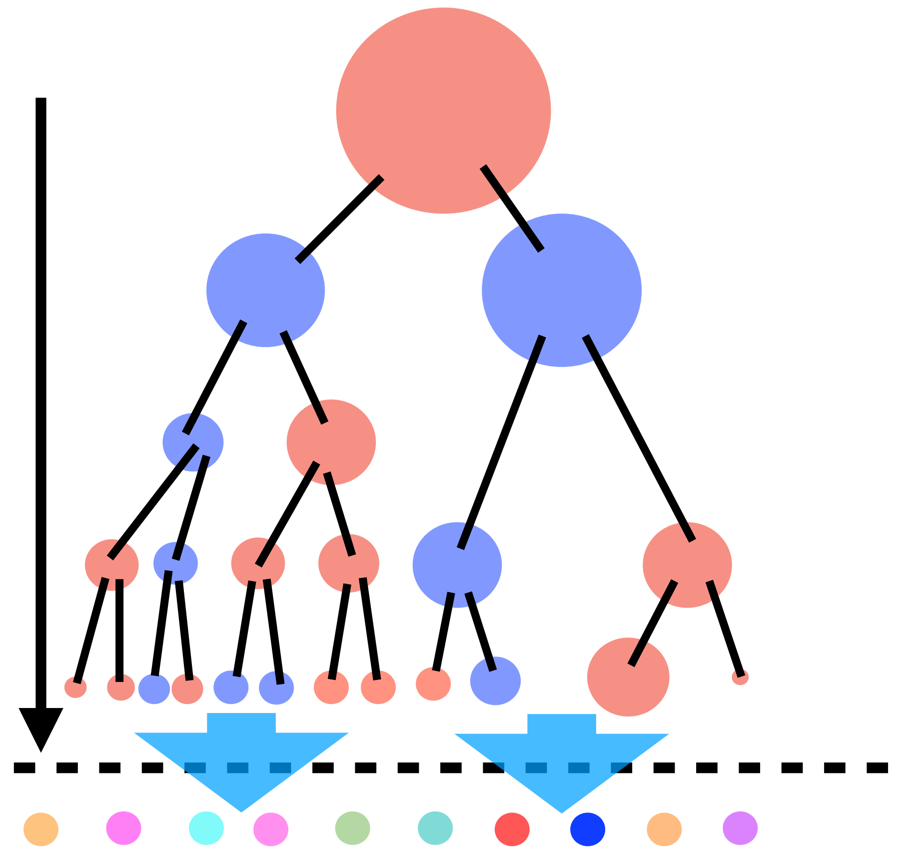
Correlations are everything

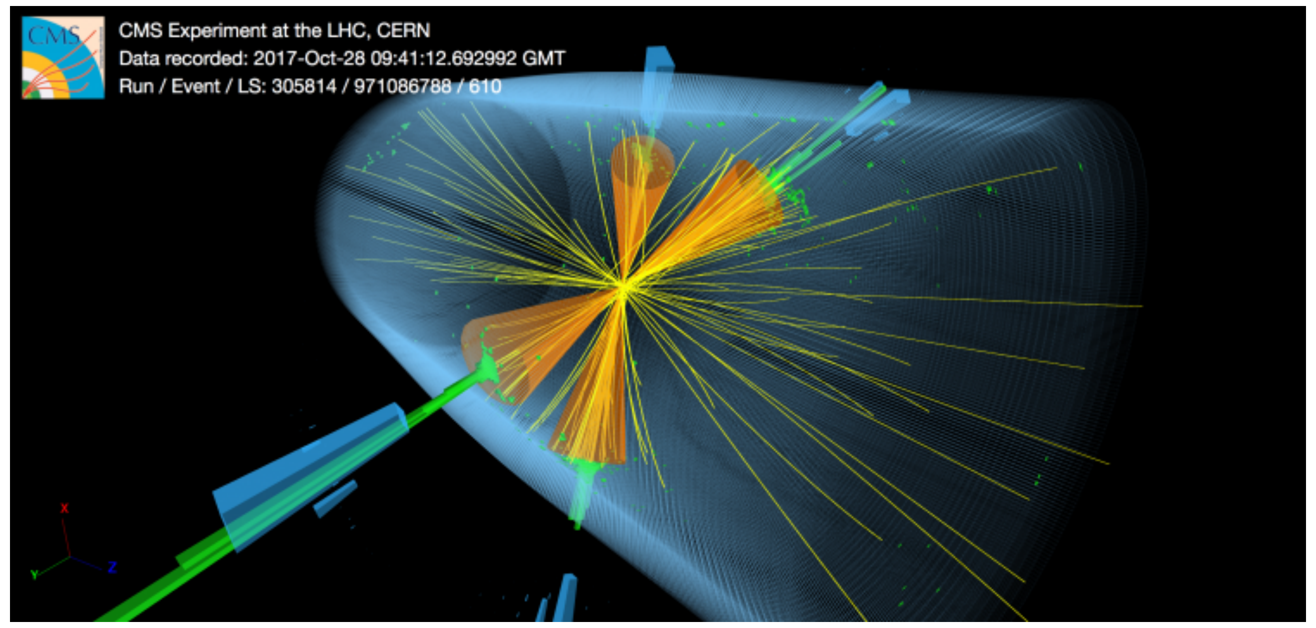
$\sim p(\text{“a cat riding a horse”} \mid \text{Internet})$

all of physics???

HEP for AI for HEP

HEP for AI: we have the right tools and philosophy to solve interesting problems in a “pre-scientific” field where experiment is WAY ahead of theory





AI for HEP: data is a gold mine. We have centuries of practice taking, manipulating, storing, simulating high-quality data: scaling lets us tap that resource for discovery

This is a virtuous cycle!

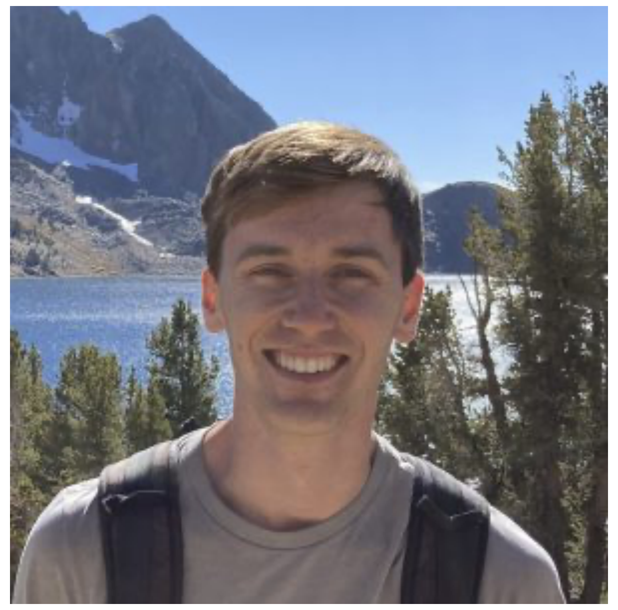
Collaborators

Hannah Day
(Ph.D. student, UIUC)

Ibrahim Elsharkawy
(Ph.D. student, U ofT)

Andrija Rasovic
(Ph.D. student, U of T)

Keegan Humphrey
(Ph.D. student, U of T)



Zachary Bogorad
(postdoc Fermilab)

Dan Roberts
(MIT -> OpenAI)

Joshua Batson
(Anthropic)

Andrew Larkoski
(APS)

Noam Levi
(EPFL/Meta AI)

David Curtin
(U of T)