Theory Overview

with a personal bias

Sven Krippendorf, 8.11.2024, Paris ML4Jets

UNIVERSITY OF CAMBRIDGE

Why?

Theorists are cheap but developing the Standard Model of Particle Physics has a non-negligible price tag and is resource limited.
Can we replace this with a single year of compute on an A100?

Theory ∩ ML
A growing landscape

ML for inference on pheno models *

ML for exploration of theories around the corner

ML for mathematics discovery

Formalising TP and proving

TP for improved ML

* covered widely in a large fraction of talks at ML4Jets. Exciting developments but excluded in this talk for time reasons.

Theory ∩ ML

A growing landscape

ML for inference on pheno models *

ML for exploration of theories around the corner

ML for mathematics discovery

Formalising TP and proving

TP for improved ML

* covered widely in a large fraction of talks at ML4Jets. Exciting developments but excluded in this talk for time reasons.

As physics students we learn formalisms/algorithms to describe dynamical systems

As physicists we develop and teach formalisms/algorithms to describe dynamical systems

$$\dot{p} = -\frac{\partial H}{\partial q}, \quad \dot{q} = \frac{\partial H}{\partial p}$$

As physics students we learn formalisms/algorithms to describe dynamical systems

As physicists we develop and teach formalisms/algorithms to describe dynamical systems

$$\dot{p} = -\frac{\partial H}{\partial q}, \quad \dot{q} = \frac{\partial H}{\partial p}$$

but why? What makes these formalisms/algorithms special,
i.e. how can we search for them?

As physics students we learn formalisms/algorithms to describe dynamical systems

As physicists we develop and teach formalisms/algorithms to describe dynamical systems

$$\dot{p} = -\frac{\partial H}{\partial q}, \quad \dot{q} = \frac{\partial H}{\partial p}$$

but why? What makes these formalisms/algorithms special, i.e. how can we search for them?

They are efficient in describing these systems.

As physics students we learn formalisms/algorithms to describe dynamical systems

As physicists we develop and teach formalisms/algorithms to describe dynamical systems

$$\dot{p} = -\frac{\partial H}{\partial q}, \quad \dot{q} = \frac{\partial H}{\partial p}$$

but why? What makes these formalisms/algorithms special, i.e. how can we search for them?

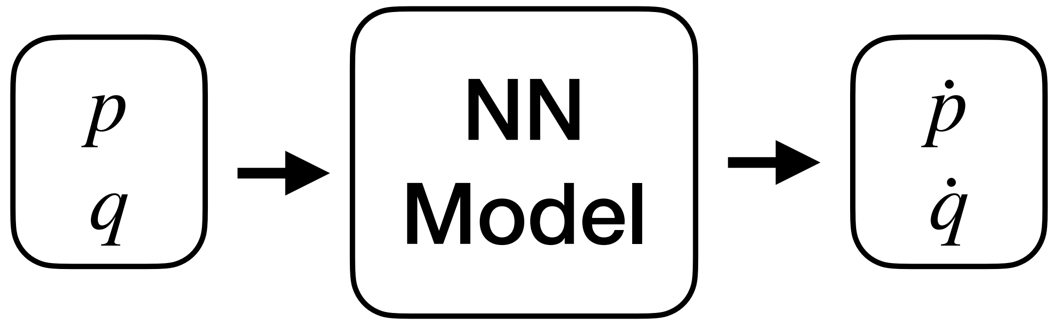
They are efficient in describing these systems.

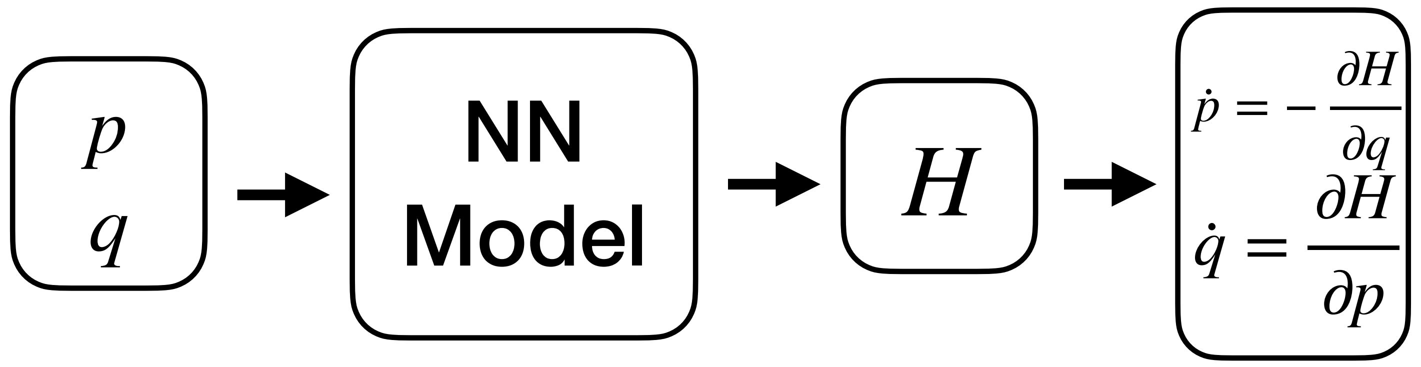
This makes such formalisms susceptible for optimisation.

Example: predicting trajectories

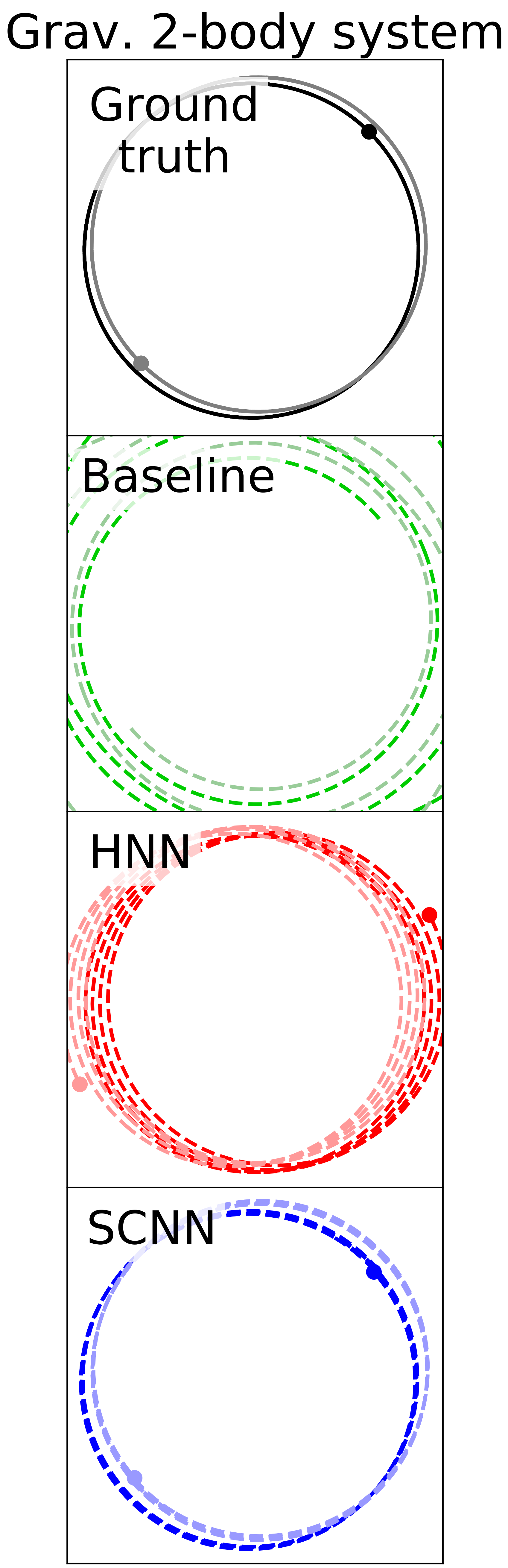
Networks with physical bias are more efficient

- Networks with correct functional bias show better generalization:





- Here: functional bias has been built in. Can we learn/generate the formalism as well?



Battaglia et al 2016 (1612.00222)
Greydanus et al. 2019
…

Can we get symbolic descriptions?

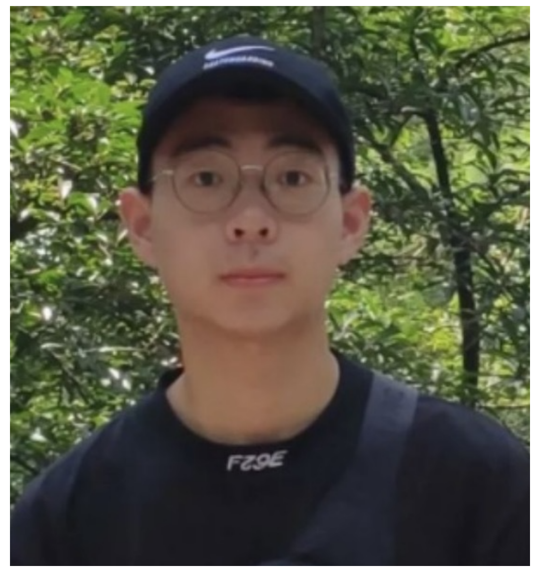
Current approaches

- Learn NN and then use your favourite symbolic regressor (e.g. PySR, cf. M. Cranmer et al.). **Problem:** *inference always from scratch (genetic algorithm)*

- Learn transformer model on known symbolic descriptions (cf. Charton et al.). **Problem:** *in general we do not know the symbolic description*

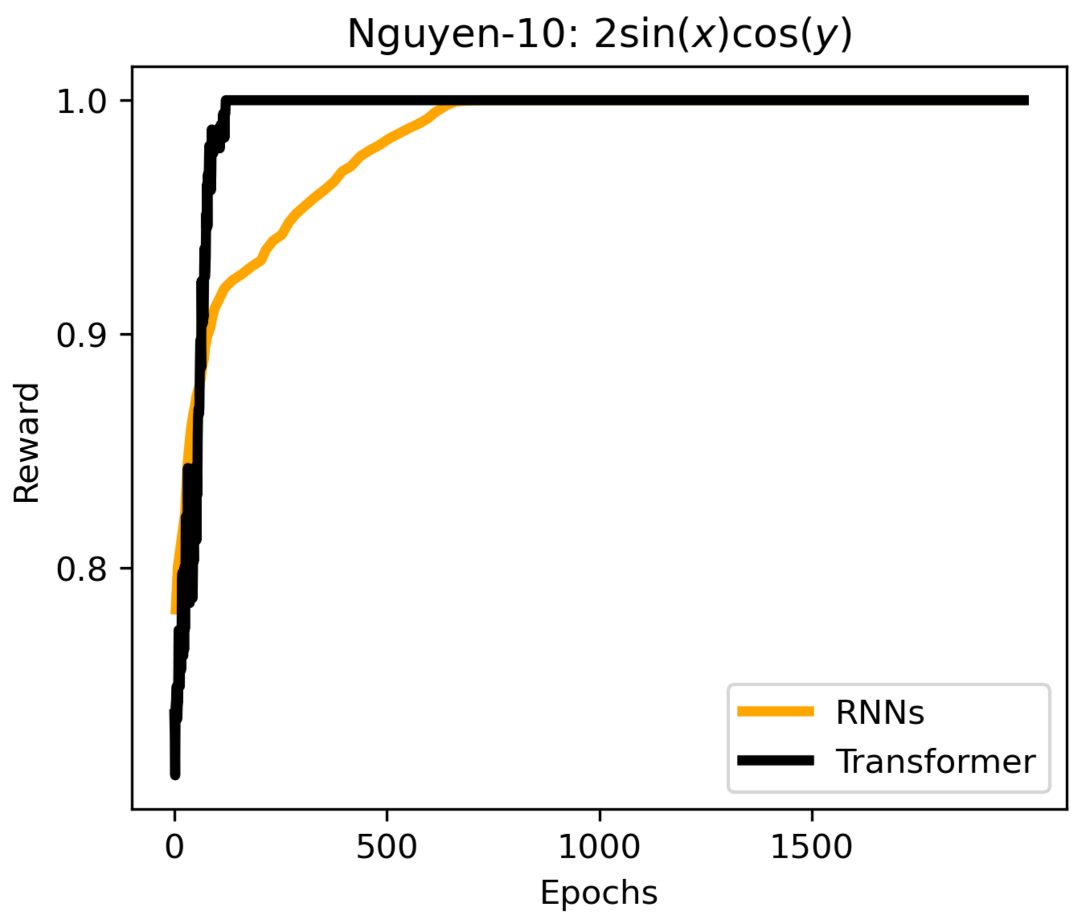
- Combine both in one step (cf. 1912.04871)?

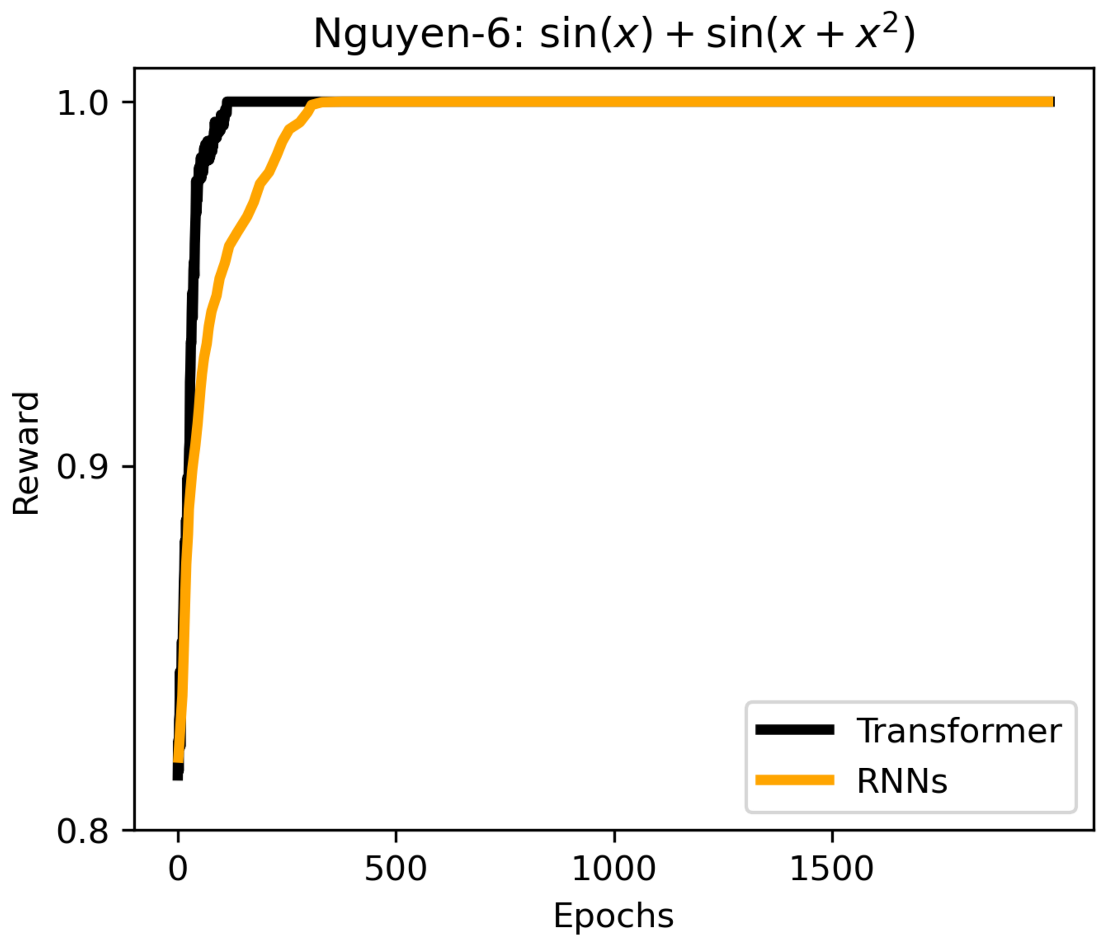
Getting symbolic expressions directly with transformers



- Combine both in one step (cf. 1912.04871)? Here: transformers [wip with Gu, Kiendl]

Input: data sample (x,f(x)) → **Model:** Encoder - Decoder transformer → **Output:** $P(\text{token}_{i+i} \,|\, \text{data}, \text{token}_{0:i})$





Optimisation:

- Generate sample expressions
- Take top expressions
- Loss

$$L_1 = \sum_i (R_i - R_{\text{threshold}}) \nabla_\theta \log \left(P(\text{expression} \,|\, \text{data})\right),$$

$$L_2 = \sum_i^{\#\text{tokens}} \gamma^i H_i,$$

$$R = \frac{1}{1 + \text{RMSE}/\sigma_y}, \quad H_i = \sum_{\text{topexp.}} P(\text{token}_i \,|\, \text{data}, \text{token}_{0:i-1})$$

- Update with $L_1 + \alpha L_2$.

- Open: further benchmarking, scaling to interesting expressions

From toy models to benchmarks

Benchmark: (Symbolic) Calabi-Yau metrics

- Yau (70s): Ricci-flat metrics on Calabi-Yau manifolds exist but no explicit construction to this date. CY manifolds are of interest as compactifications in string theory.
- Problem to solve: Solving Einsteins equations on compact six-dimensional manifolds
- NNs for efficient solutions (active field with various packages [2410.19728,2211.12520, 2205.13408] and phenomenological applications are started to be explored [2407.13836, 2411.00962]). In special cases (Fermat quintic) down to machine precision [0908.2635].
- Challenge: For precision metrics, can we find symbolic expressions? Issue, overcome combinatorial explosion due to high number of variables, e.g.:

$$K = -\log\left(1 + \sum_{i=1}^{4} |z_i|^2\right)$$

Theory ∩ ML
A growing landscape

ML for inference on pheno models *

ML for exploration of theories around the corner

ML for mathematics discovery

Formalising TP and proving

TP for improved ML

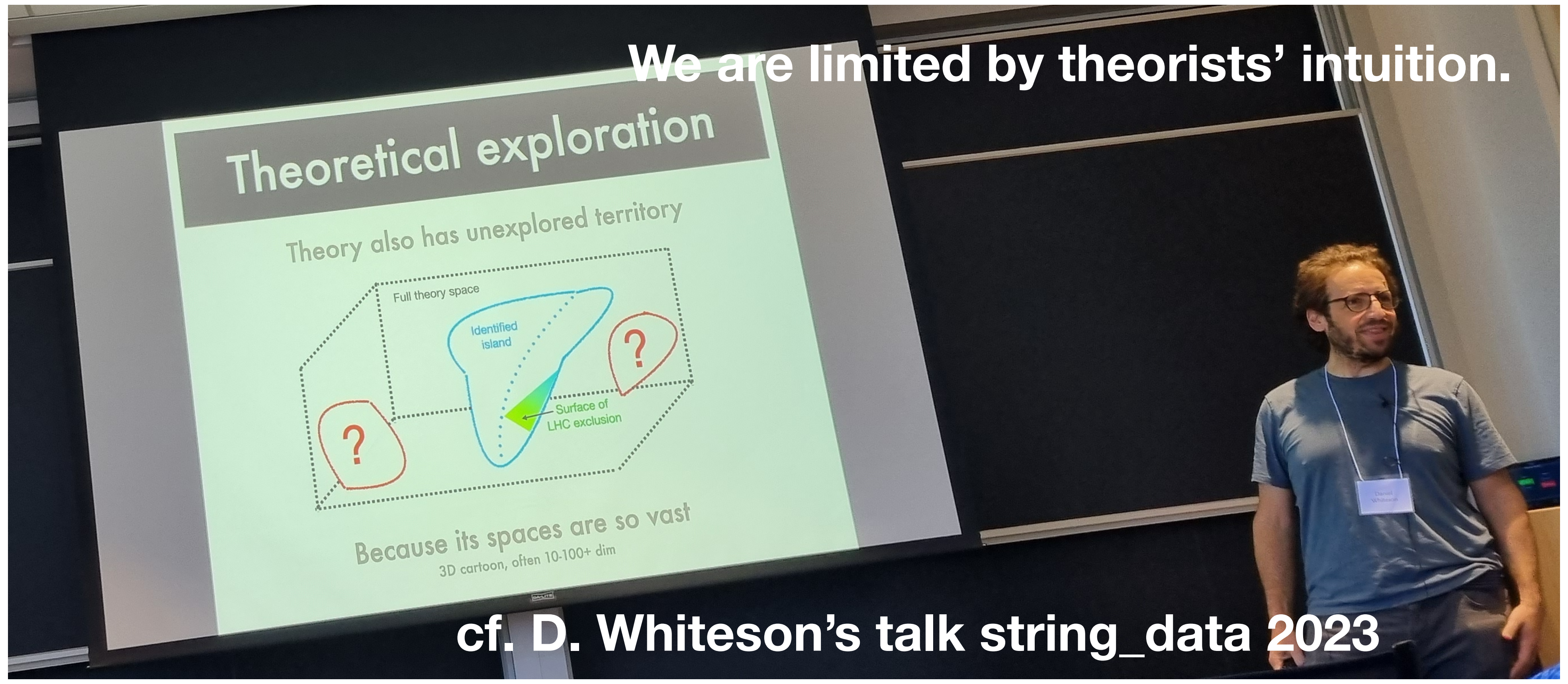
* covered widely in a large fraction of talks at ML4Jets. Exciting developments but excluded in this talk for time reasons.

ML for Exploration in Theory Space

- We have many theories but we have not yet explored their phenomenology. Why bother? Unexplored whether they contain new methods to address our old problems (e.g. EW hierarchy problem, cosmological constant)

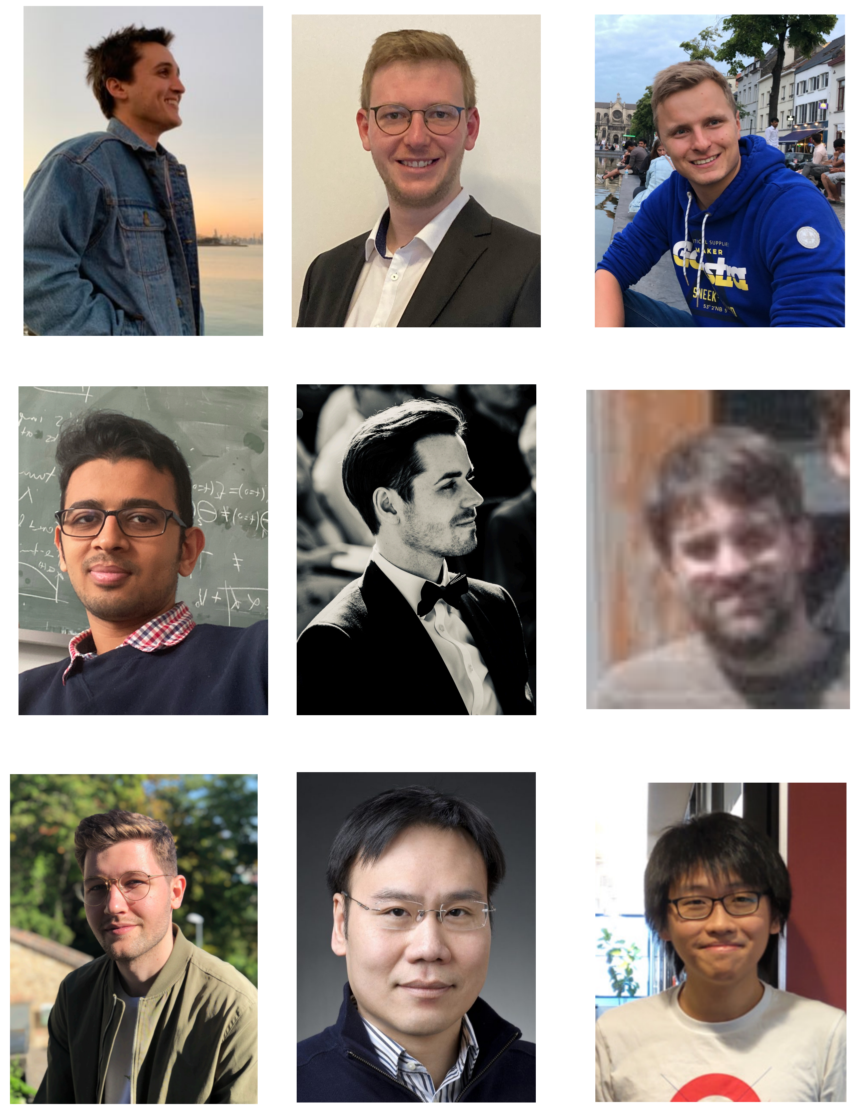
- Can we search existing theory space efficiently?
  Not until recently (e.g. string theory model space ⊂ BSM models) as tools were missing.

- Case study: Flux compactification of type IIB string theory (see also work on IIA (e.g. Loges, Shiu) and heterotic string theory (e.g. Abel, Constantin, Fraser-Taliente, Harvey, Lukas…))



Team & Papers

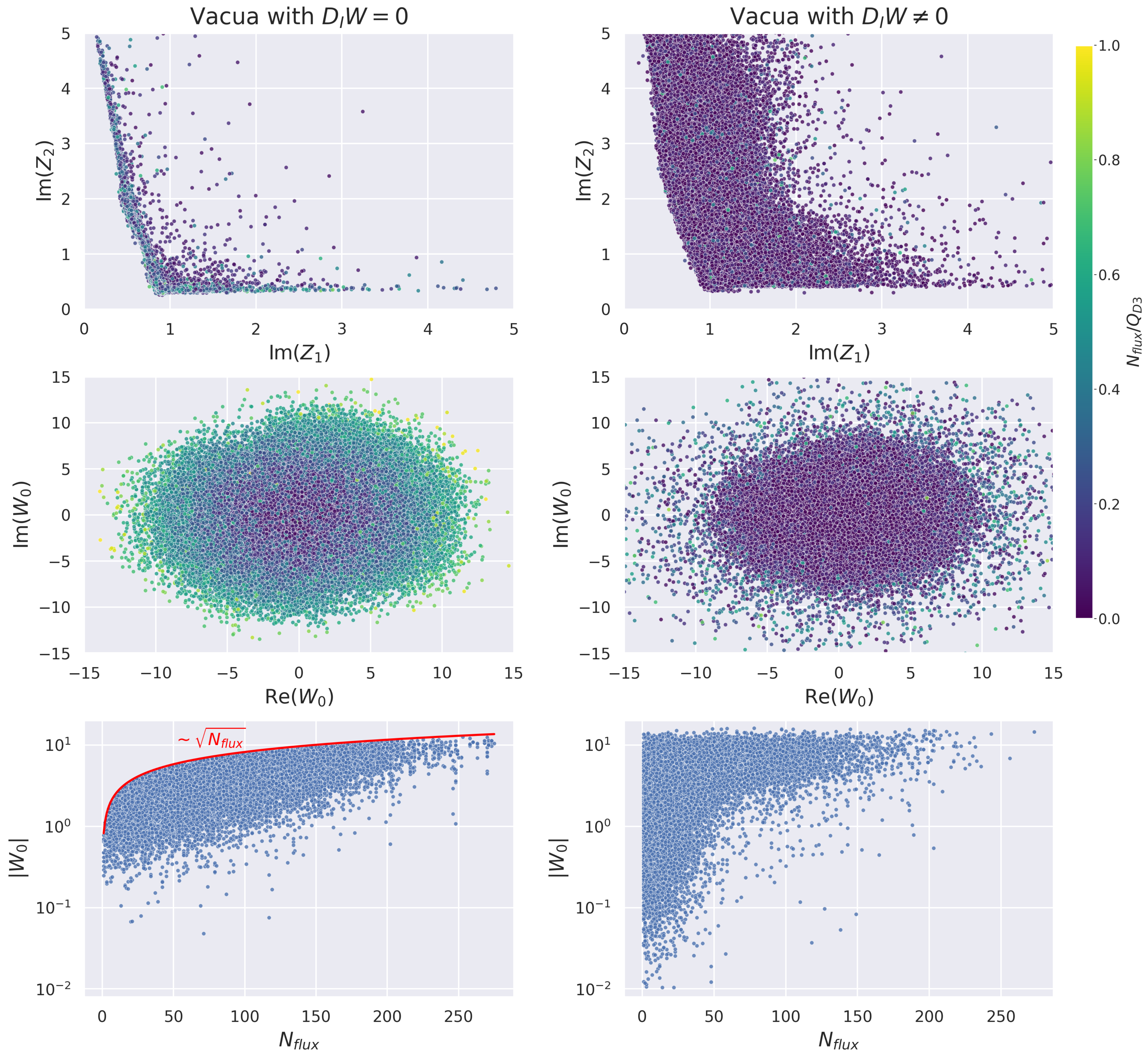
2107.04039,
2111.11466,
2209.15433
2306.06160,
2307.15749,
2308.15525



Tools for string theory model space exploration

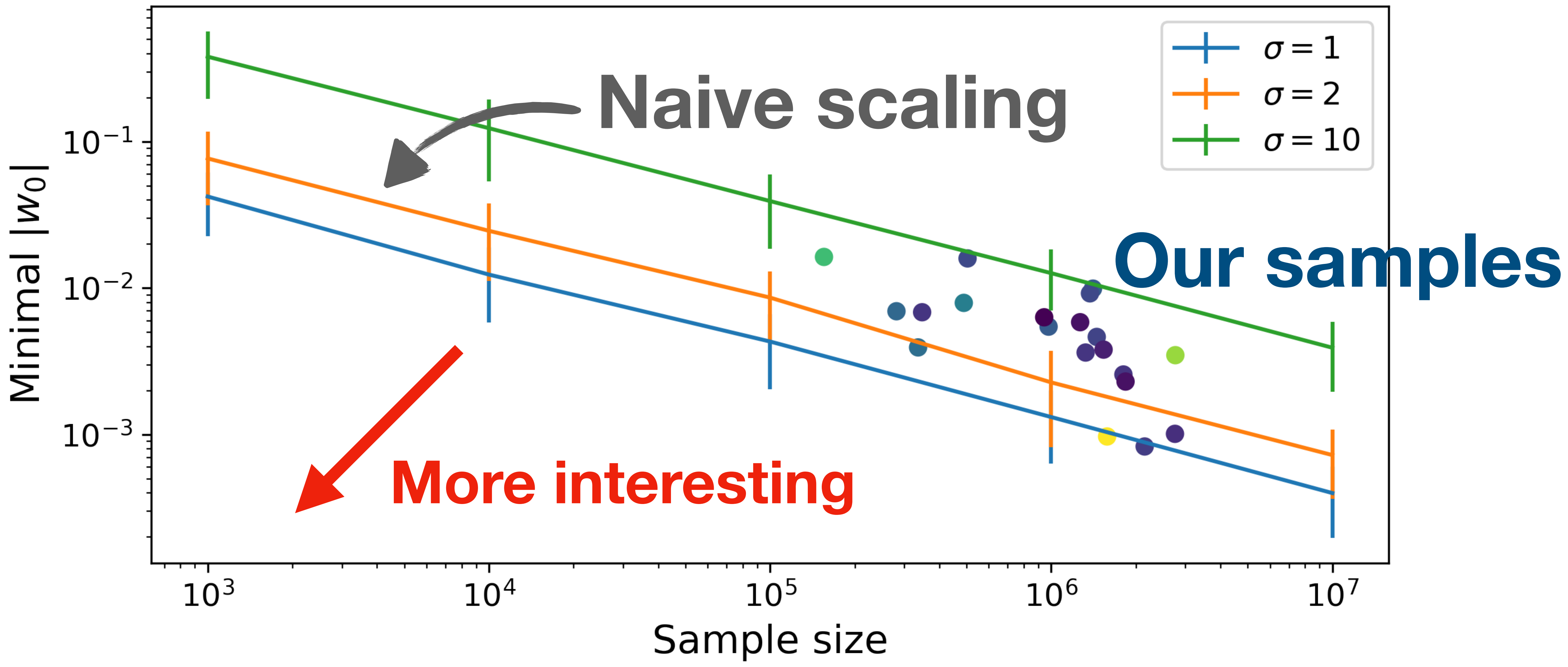
- Model space: (Geometry, Local sources); Local sources are subject to consistency constraints such as anomaly cancellation
- EFT algorithm "known" and can be evaluated using appropriate derivatives with respect to fields parametrizing the extra-dimensions:

  (discrete input to prepotential → Kähler potential, superpotential → scalar potential)
- Optimisation
- Tools to efficiently access many of such models: custom JAX code for vectorised and compiled machinery



First applications

- We finally can sample from this space efficiently:



- Next steps: explore with appropriate numerical tools to efficiently sample from these model spaces.

Theory ∩ ML

A growing landscape

ML for inference on pheno models *

ML for exploration of theories around the corner

ML for mathematics discovery

Formalising TP and proving

TP for improved ML

* covered widely in a large fraction of talks at ML4Jets. Exciting developments but excluded in this talk for time reasons.

Formalising TP and proving

Joseph Tooby-Smith, 2405.08863

- Idea: LLMs for automated theorem proving (silver medal Math olympiad this year); TP also has theorems and conjectures. Can this be useful?

- **LeanBSM:** first steps in formalizing HEP questions in Lean. E.g. proving that our Higgs potential has a minimum.

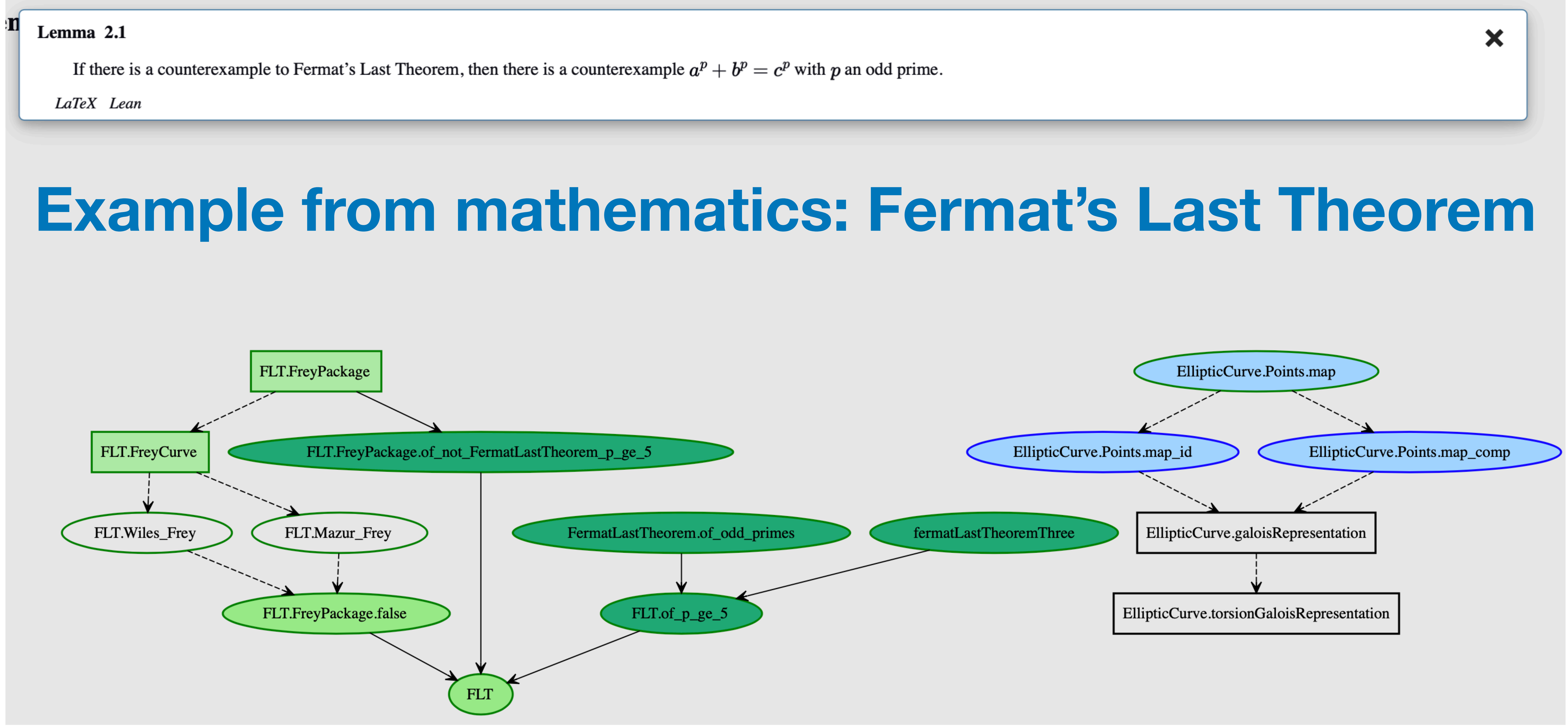
```
lemma IsMinOn_potential_iff_of_μSq_nonneg {μSq lambda : ℝ}                    (source)
    (hLam : 0 < lambda) (hμSq : 0 ≤ μSq) :
    IsMinOn (potential μSq lambda) Set.univ φ ↔ ‖φ‖ ^ 2 = μSq /(2 * lambda) := by
  ...
```

Blueprints

Splitting proves into parts — Roadmaps for larger proofs

- Roadmaps for theoretical physics: we often do not actually know why a particular question in TP is relevant (e.g. why your favourite string theory colleague cares about the KKLT scenario in string theory).

- What can be included? Assumptions, experimental data.



https://imperialcollegelondon.github.io/FLT/blueprint/index.html

Theory ∩ ML

A growing landscape

ML for inference on pheno models *

ML for exploration of theories around the corner

ML for mathematics discovery

Formalising TP and proving

TP for improved ML

* covered widely in a large fraction of talks at ML4Jets. Exciting developments but excluded in this talk for time reasons.

TP for improved ML

Gerdes, Cheng, Welling 2410.02667

Designing diffusion models with renormalisation group methods

- Designing diffusion models inspired by comparison with renormalisation group methods:

Standard diffusion

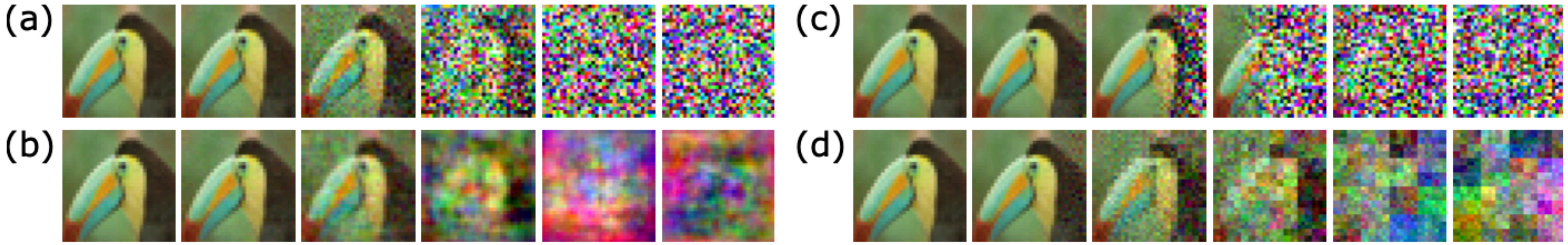
Data $\xrightarrow{\text{Diagonal pixel basis}}$ White Noise

Renormalisation Group

Data $\xrightarrow{\text{Diagonal frequency basis}}$ Scale-invariant distribution

Erases information from high to low-frequencies

1) Basis, 2) Prior distribution, 3) Noise Scheduling



based on 2202.11104 (MLST), 2305.00995 (MLST), and 2410.07451:

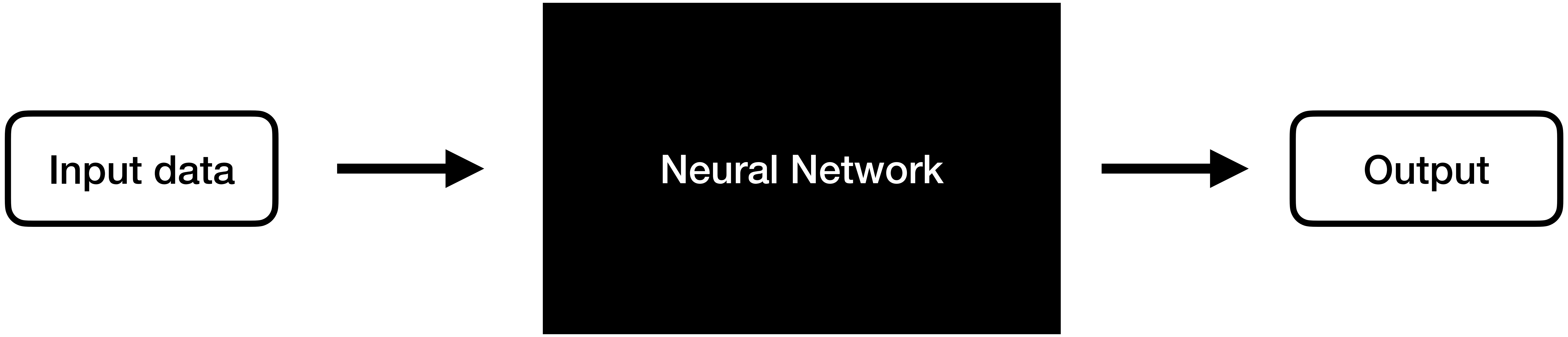
Michael Spannowsky

Sam Tovey

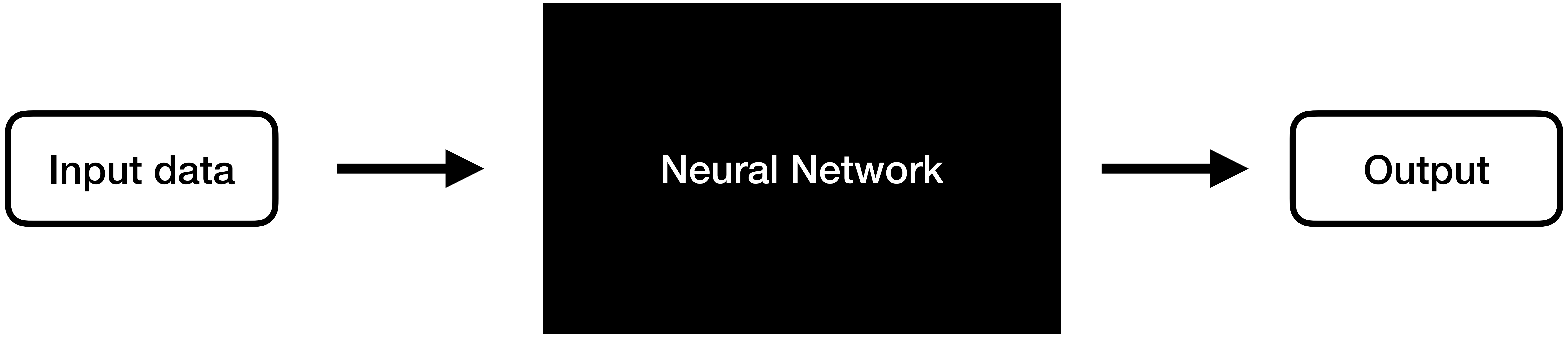
Konstantin Nikolaou

Christian Holm

Are neural networks black boxes?

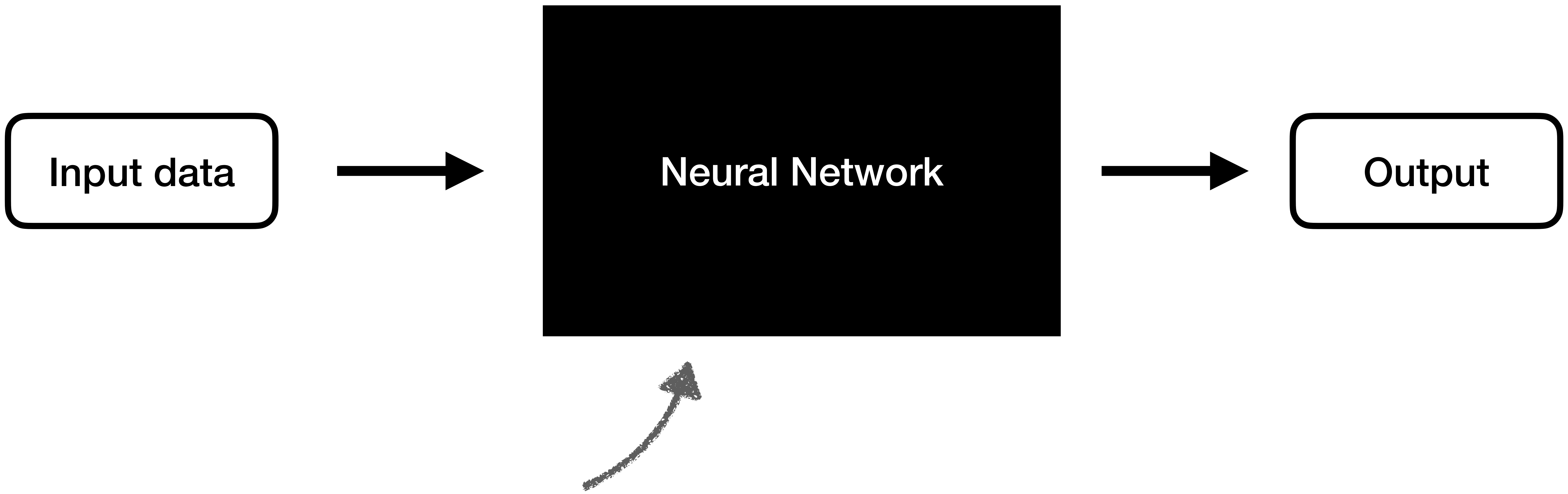


Are neural networks black boxes?



Analytic function, but many parameters so it’s not a simple function.

Are neural networks black boxes?



Analytic function, but many parameters so it's not a simple function.

Do we know what is going on inside them?

Some hints: scaling laws

e.g. performance improves with more parameters

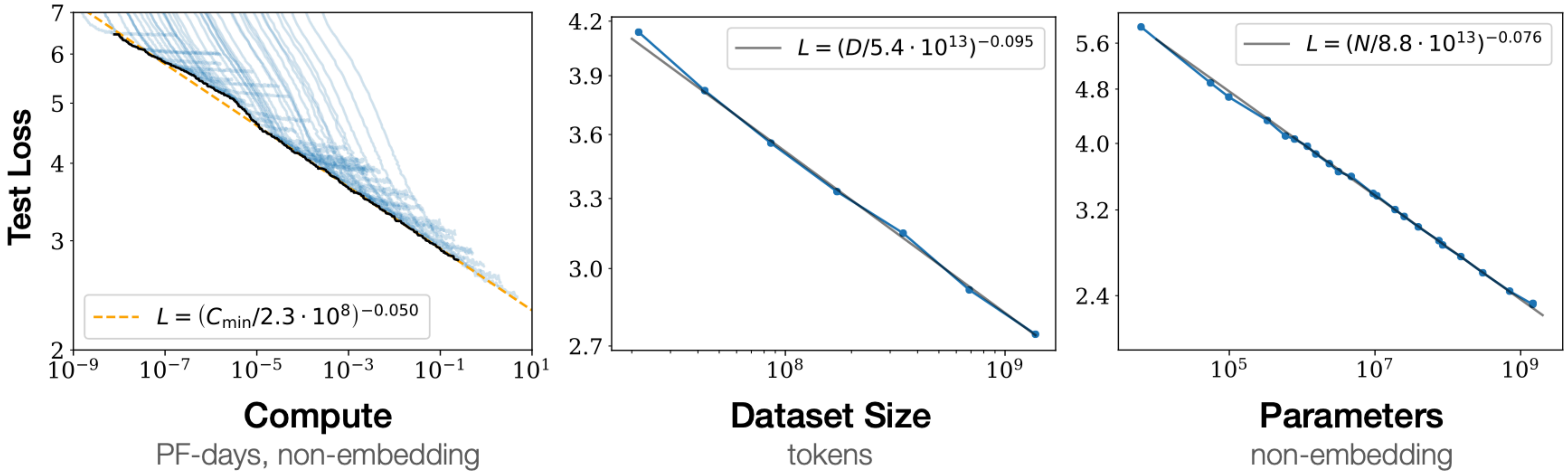


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

e.g. Kaplan et al 2020 Scaling laws for neural language models

We were able to precisely model the dependence of the loss on N and D, and alternatively on N and S, when these parameters are varied simultaneously. We used these relations to derive the compute scaling, magnitude of overfitting, early stopping step, and data requirements when training large language models. So our scaling relations go beyond mere observation to provide a predictive framework. One might interpret these relations as analogues of the ideal gas law, which relates the macroscopic properties of a gas in a universal way, independent of most of the details of its microscopic consituents.

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.

Kaplan et al 2020 Scaling laws for neural language models

We were able to precisely model the dependence of the loss on N and D, and alternatively on N and S, when these parameters are varied simultaneously. We used these relations to derive the compute scaling, magnitude of overfitting, early stopping step, and data requirements when training large language models. So our scaling relations go beyond mere observation to provide a predictive framework. One might interpret these relations as analogues of the ideal gas law, which relates the macroscopic properties of a gas in a universal way, independent of most of the details of its microscopic consituents.

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.

Do we know what is going inside NNs?

For us becomes: Theoretical framework to quantify dynamical behaviour of NNs?

Kaplan et al 2020 Scaling laws for neural language models

Physics to understand NN dynamics

Problems and our approach

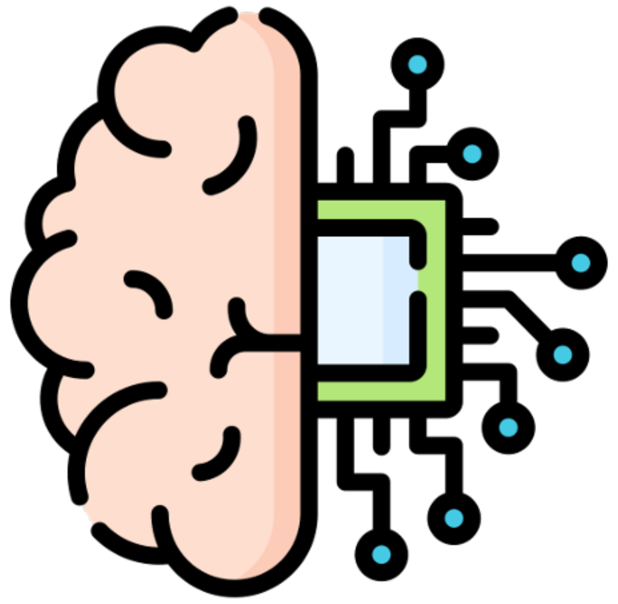
Parameters	175 billion
Training Time	Several months
Training Cost	~ $4.6 million

OpenAI

- We cannot afford hyperparameter scans for such large networks. *How to successfully predict training performance?*

Neurons	86 billion
Object recognition time[2]	150 ms
Energy cost[1]	< 20 W

[1] (Sterling & Laughlin, 2015), [2] (Thorpe et al., 1996)



- Our NN networks are not energy efficient. *How to improve efficiency of NNs to make them useful with less computational resources?*

cf. Lahiri, Sohl-Dickstein, Ganguli 1603.07758

Physics to understand NN dynamics

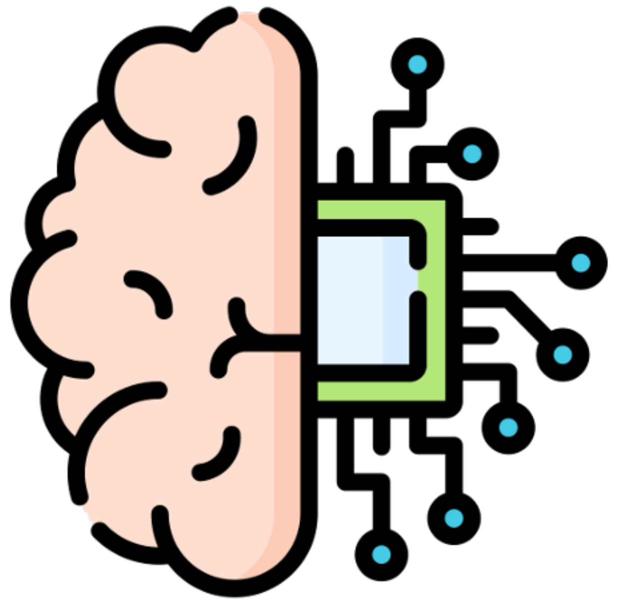
Problems and our approach

Parameters	175 billion
Training Time	Several months
Training Cost	~ $4.6 million

OpenAI

Neurons	86 billion
Object recognition time[2]	150 ms
Energy cost[1]	< 20 W

[1] (Sterling & Laughlin, 2015), [2] (Thorpe et al., 1996)



- We cannot afford hyperparameter scans for such large networks. *How to successfully predict training performance?*

- Our NN networks are not energy efficient. *How to improve efficiency of NNs to make them useful with less computational resources?*

cf. Lahiri, Sohl-Dickstein, Ganguli 1603.07758

Describe neural networks & dynamics via dynamics of collective variables.
Aim: control and improve learning of NNs.

How do we link dynamics of NNs and collective variables?

Understand NN dynamics via empirical NTK

Simplification of dynamics in large width limit

- The dynamics of a neural network $f(x, \theta)$ simplify in the infinite width limit.
- The NN equations in continuous time limit:

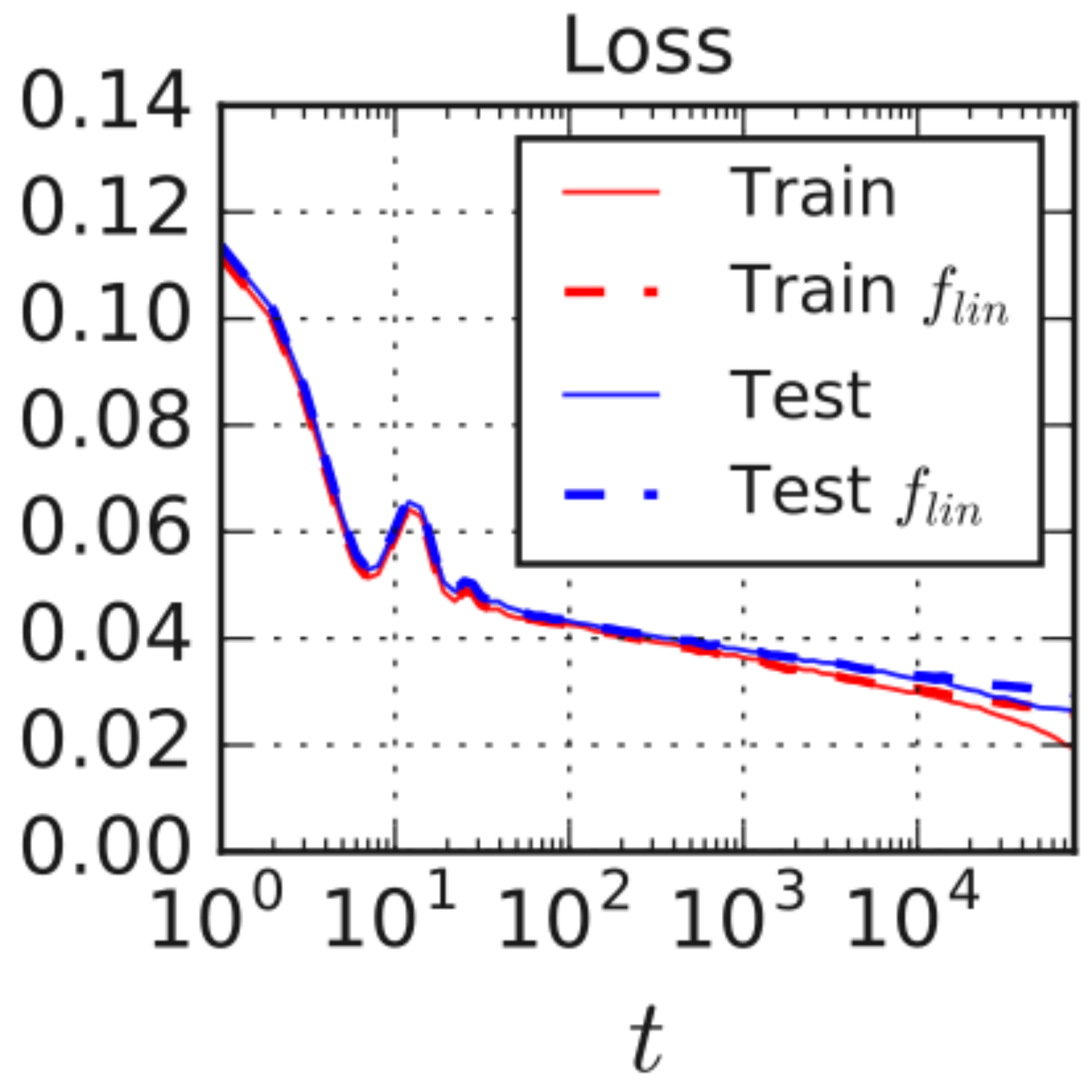
$$\dot{\theta} = -\eta \nabla_\theta \mathscr{L} = -\eta \nabla_\theta f(y) \nabla_{f(y)} \mathscr{L}$$

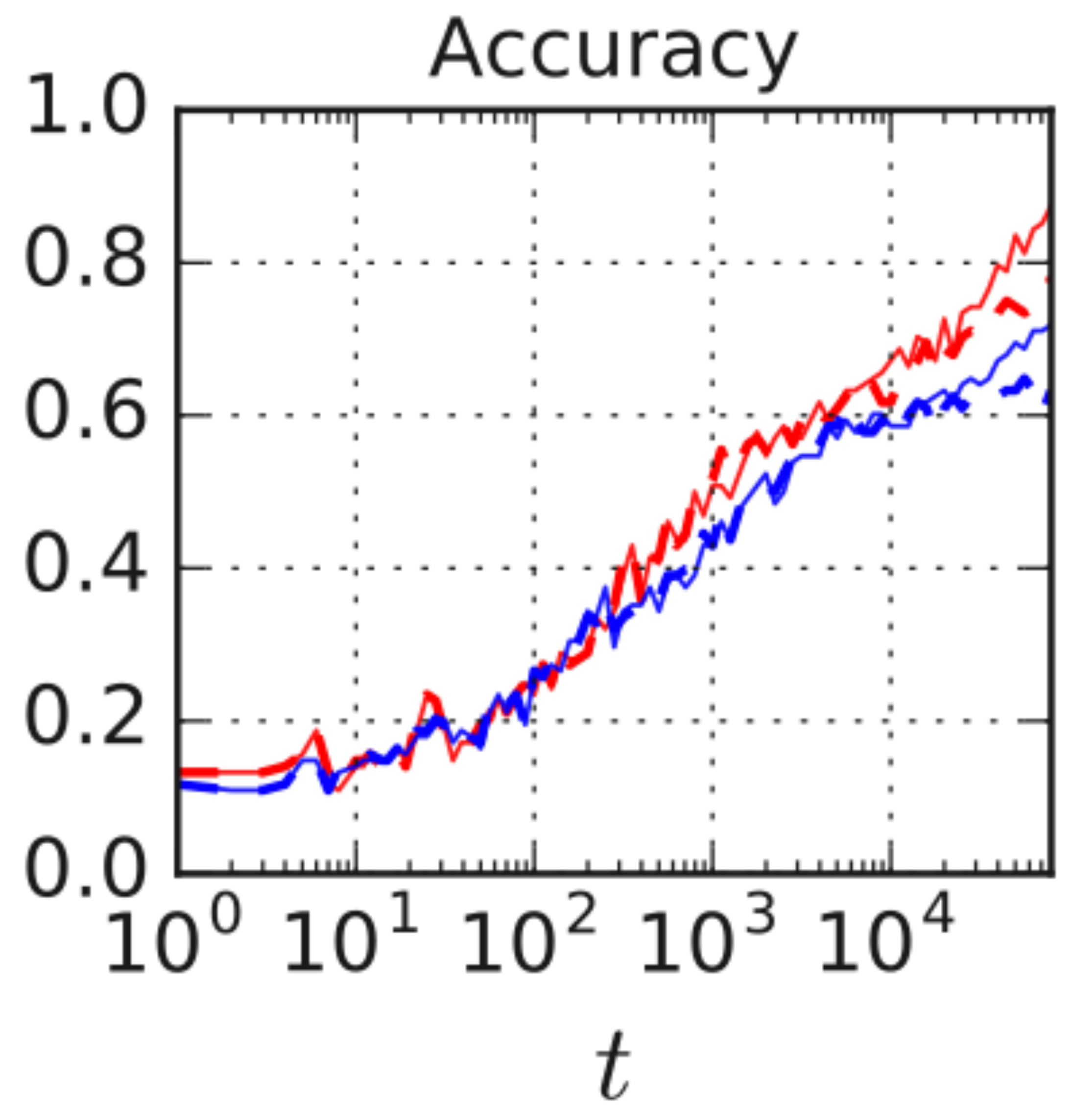
$$\dot{f}(x) = \nabla_\theta f(x)\, \dot{\theta} = -\eta \nabla_\theta f(x) \nabla_\theta f(y) \nabla_{f(y)} \mathscr{L} = -\eta \Theta(x, y) \nabla_{f(y)} \mathscr{L}$$

- NN update simplify in large width limit: Neural tangent kernel remains constant (empirical and analytical):

$$\Theta(t, x, y) = \Theta(t = 0, x, y)$$

- Complete as all learning components included: finite data, optimisers, and NN architecture
- Not sufficient (e.g. not capturing feature learning), in practice $\Theta(t, x, y) \approx \Theta(t = 0, x, y)$ at finite but large width. Which simple model describes the dynamics of NTK?





Wide resnet trained by SGD with momentum on CIFAR-10 (from 1902.06720)

Krippendorf, Spannowsky: 2202.11104
Tovey, Krippendorf, Nikolou, Holm: 2305.00995

Jacot, Gabrial, Hongler
Lee, Xiao, Schoenholz, Bahri, Novak, Sohl-Dickstein, Pennington
Novak, Xiao, Hron, Lee, Alemi, Sohl-Dickstein, Schoenholz

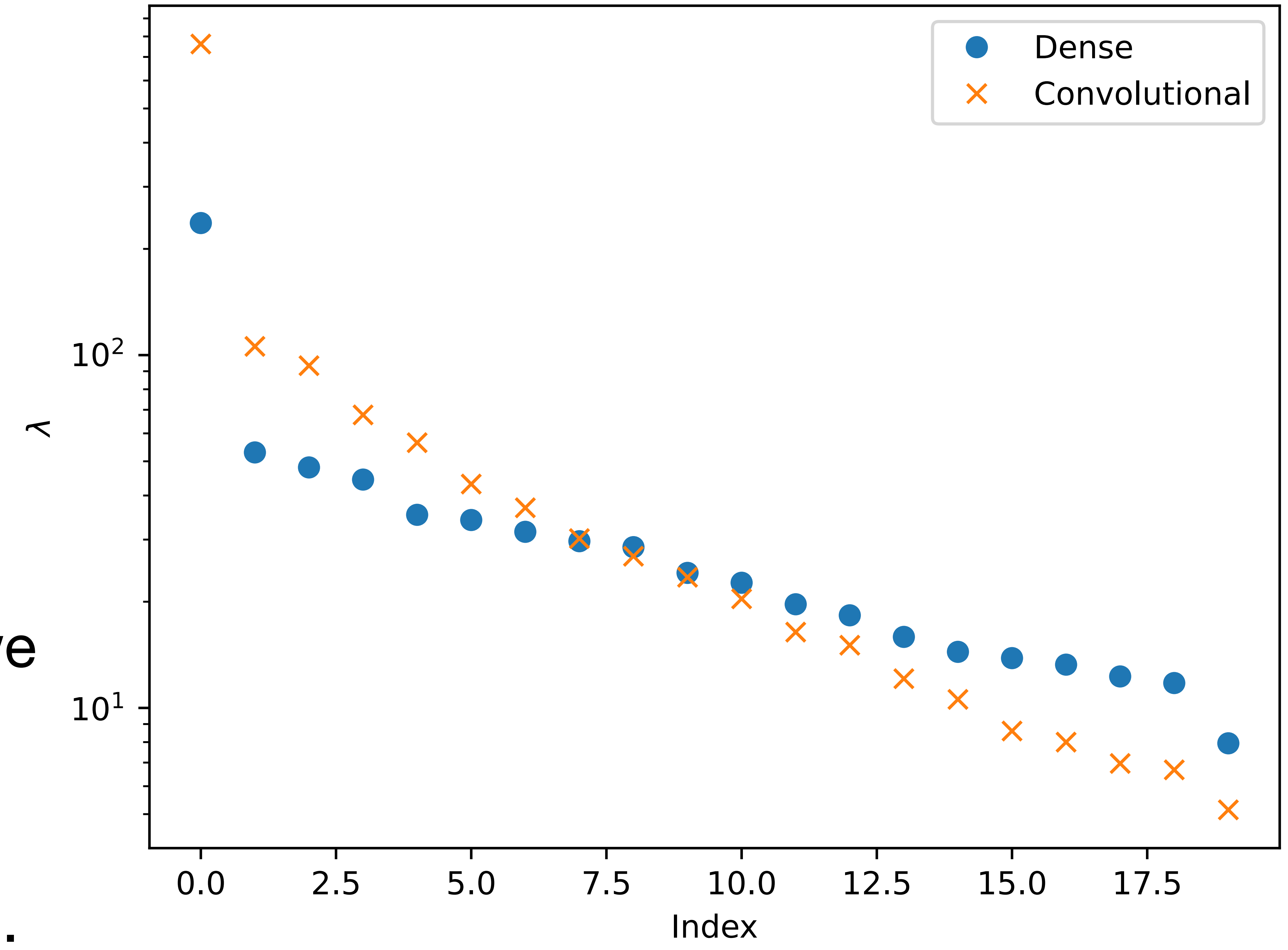
Scales in NN dynamics

Hierarchical spectrum in NTK → EFT (coll. variable) approach promising

- Diagonalise NTK (Θ_{NTK}) NN-update equation:

$$\dot{\tilde{f}}(\mathscr{D}) = -\eta\ \mathrm{diag}(\lambda_1, \ldots, \lambda_{\mathrm{N}})\ \mathscr{L}'(\mathscr{D})$$

- Largest changes in modes with largest eigenvalues.
- Hierarchical spectrum in NTK, consequences:
  - Effectively dynamics take place in lower-dimensional subspace.
  - There are few "collective" variables in NTK which determine the dynamics. Their time evolution is what we need to understand.
  - Limit: adding more data does not change dynamics if non-vanishing eigenvalues are not changed (naturally cut-offs do appear analogy with effective field theories).



cf. Gur-Ari, Roberts, Dyer 2018

spectrum perspective: 2202.11104 (MLST)

Variables to capture significant changes in spectrum

Overall magnitude of NTK (trace) and diversity entropy

- We see that the maximal eigenvalues of the NTK is very dominant and was relevant in the mean evolutions of the network:

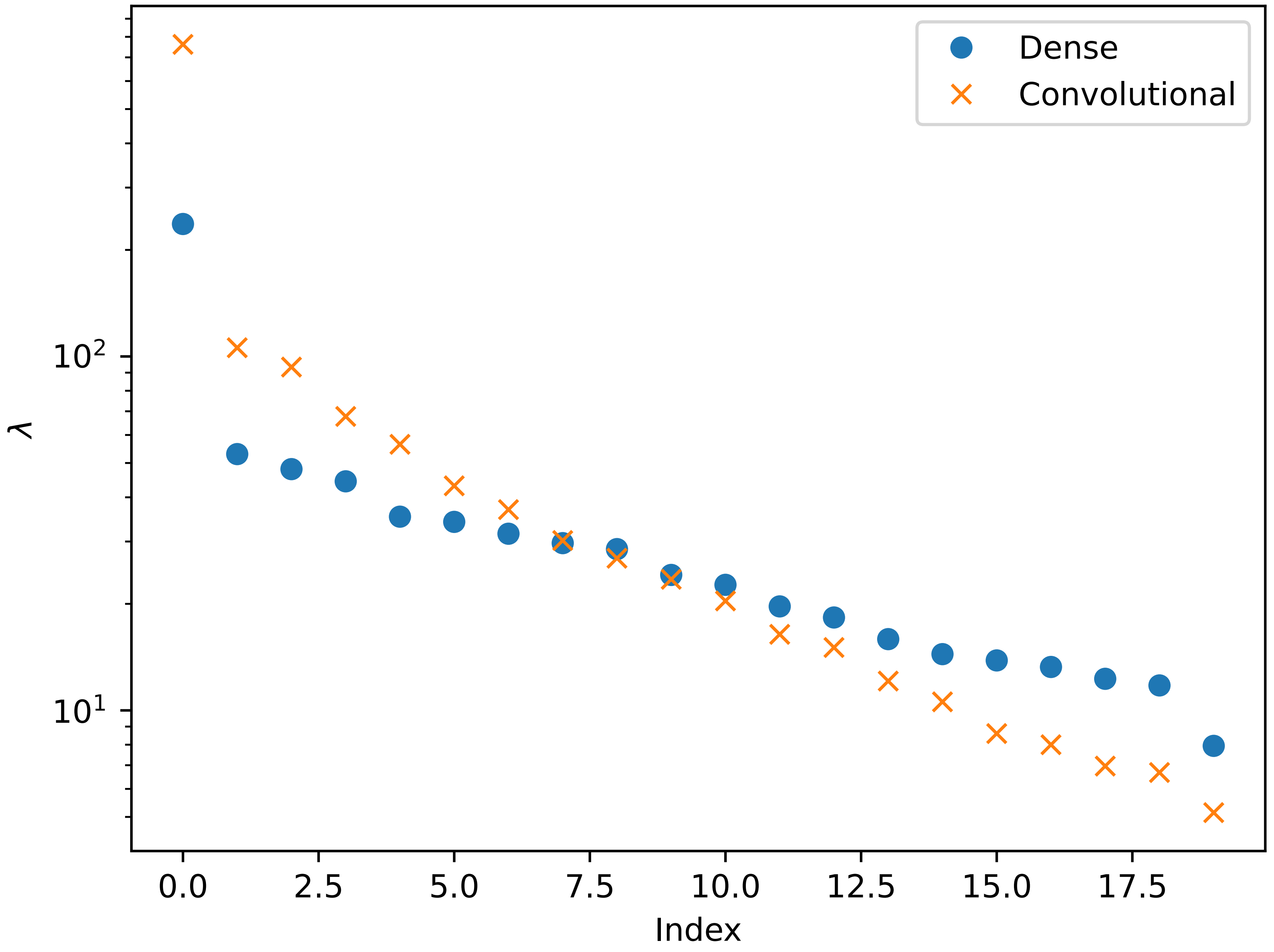
$$\mathrm{Tr}(\Theta_{\mathrm{NTK}}) = \sum_i \lambda_i \approx \lambda_{\mathrm{max}}$$

- The # of relevant modes differs between tasks. A variable which is independent of the # of modes is the following entropy:

$$S^{VN} = -\sum_i \hat{\lambda}_i \log \hat{\lambda}_i$$

(here: $\hat{\lambda}_i$ normalised eigenvalues of Θ_{NTK})

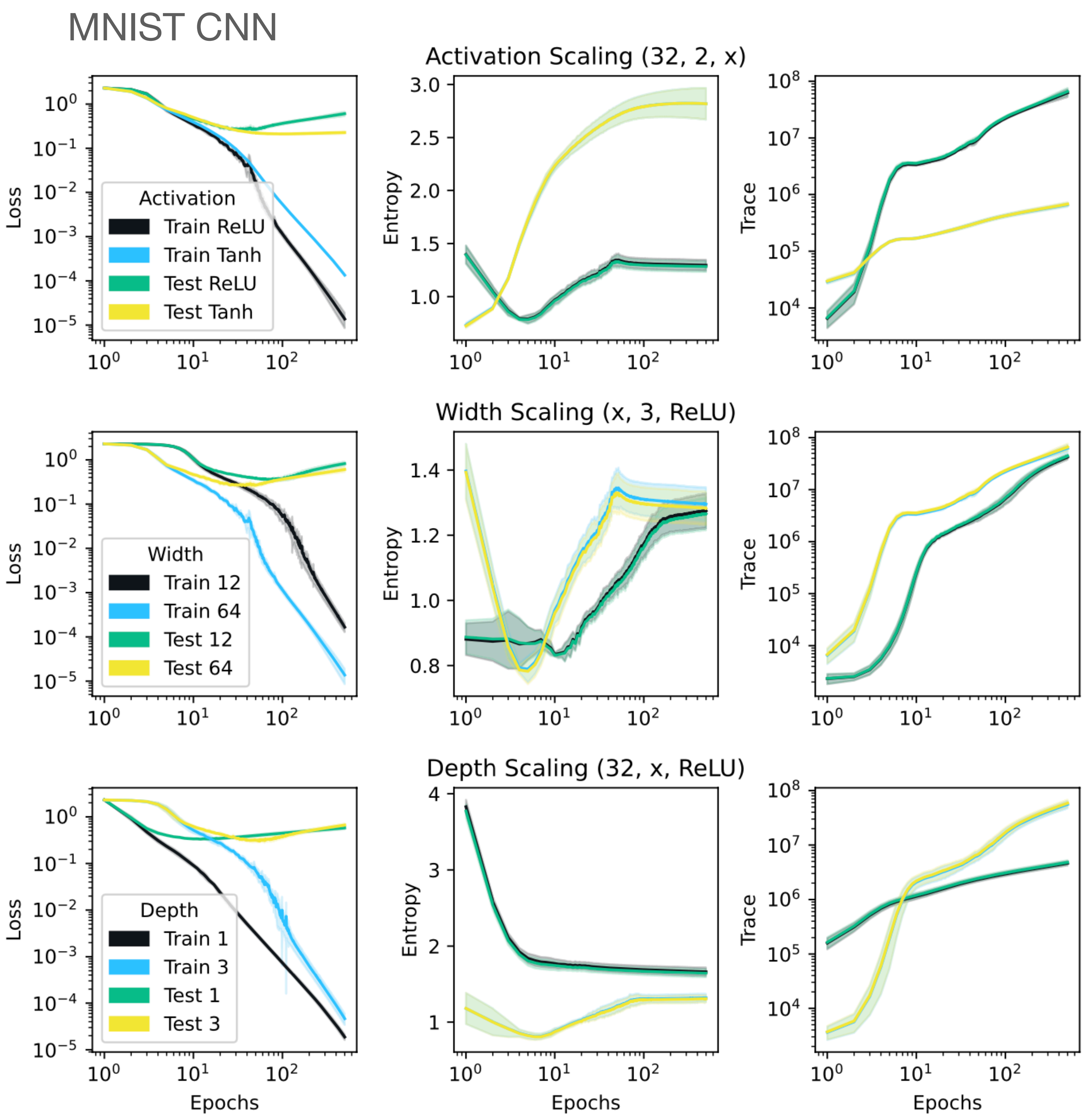
‣ **How do these two variables correlate with neural network behaviour? How do they evolve during training?**

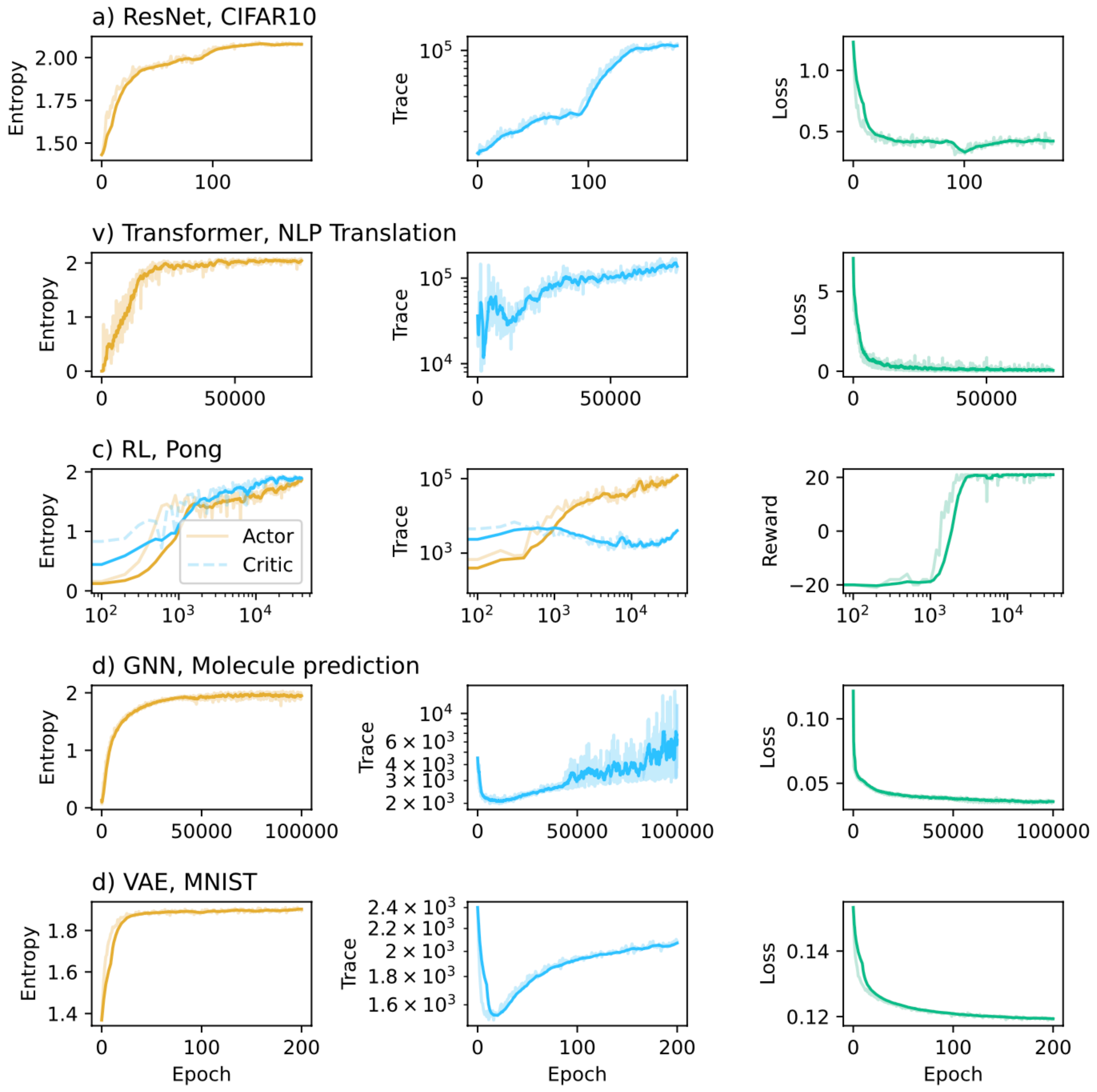


NTK evolution study

Collective variables

- Universal behavior of training dynamics: information compression at the beginning of training and then structure formation (increased trace and entropy for large model)
- Definition of deep learning regime via entropy behaviour





A biased selection for BSM

- LLMs for symbolic regression (formalism search): beyond next word prediction
- Automated theorem proving: LEAN meets physics (HEPLean)
- Benchmark for symbolic regression: CY-metrics
- Flux-vacua: exploring BSM model spaces
- Improving diffusion models with RG
- Theory for ML: collective variables of NTK

Thank you!

<u>Advertisement: ML ∩ Physics growing in Cambridge</u>

- DIS MPhil (1 year Master)
- DIS CDT (graduate school)
- MOU: Cambridge-Infosys AI Lab [postdocs, students]
- And a lot of cool people to work with…