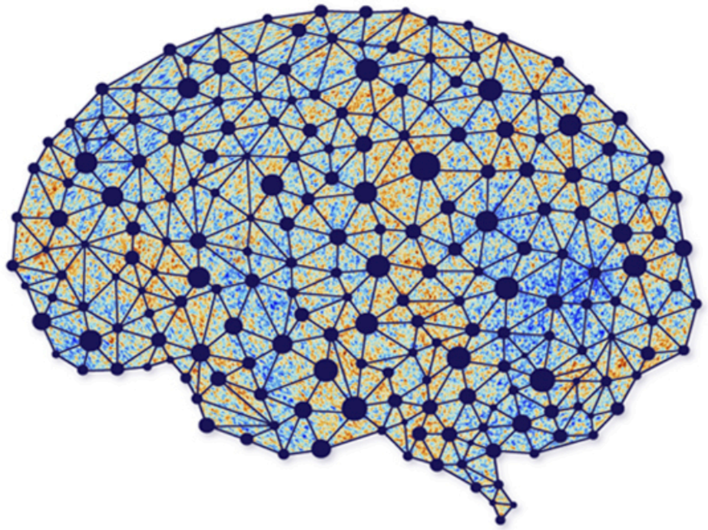
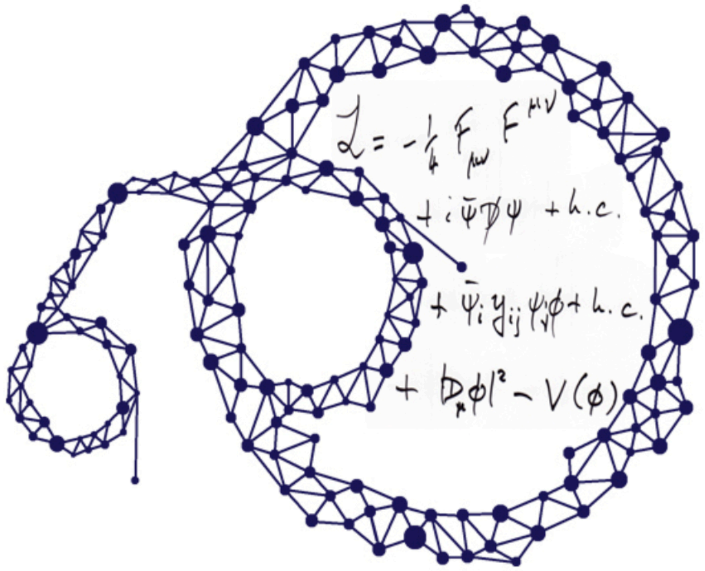
Physics 361 - Machine Learning in Physics

Lecture 17 – Foundation Models for Science

March 18th 2025



AI
∩
Universe


$\mathcal{L} = -\frac{1}{4} F_{\mu\nu} F^{\mu\nu}$
$+ i \bar{\psi} \not{D} \psi + h.c.$
$+ \bar{\psi}_i y_{ij} \psi_j \phi + h.c.$
$+ |D_\mu \phi|^2 - V(\phi)$

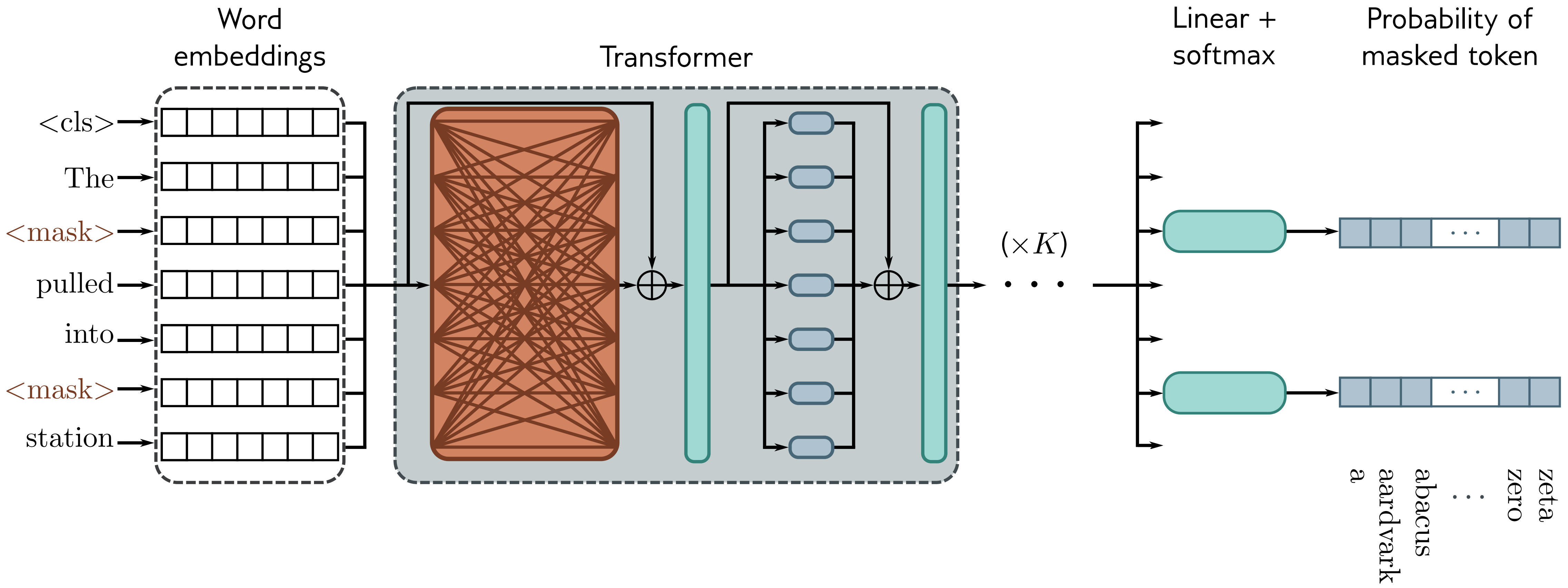
Moritz Münchmeyer

Transformers

Recall: LLM Encoders and Decoders

Encoder Pre-training

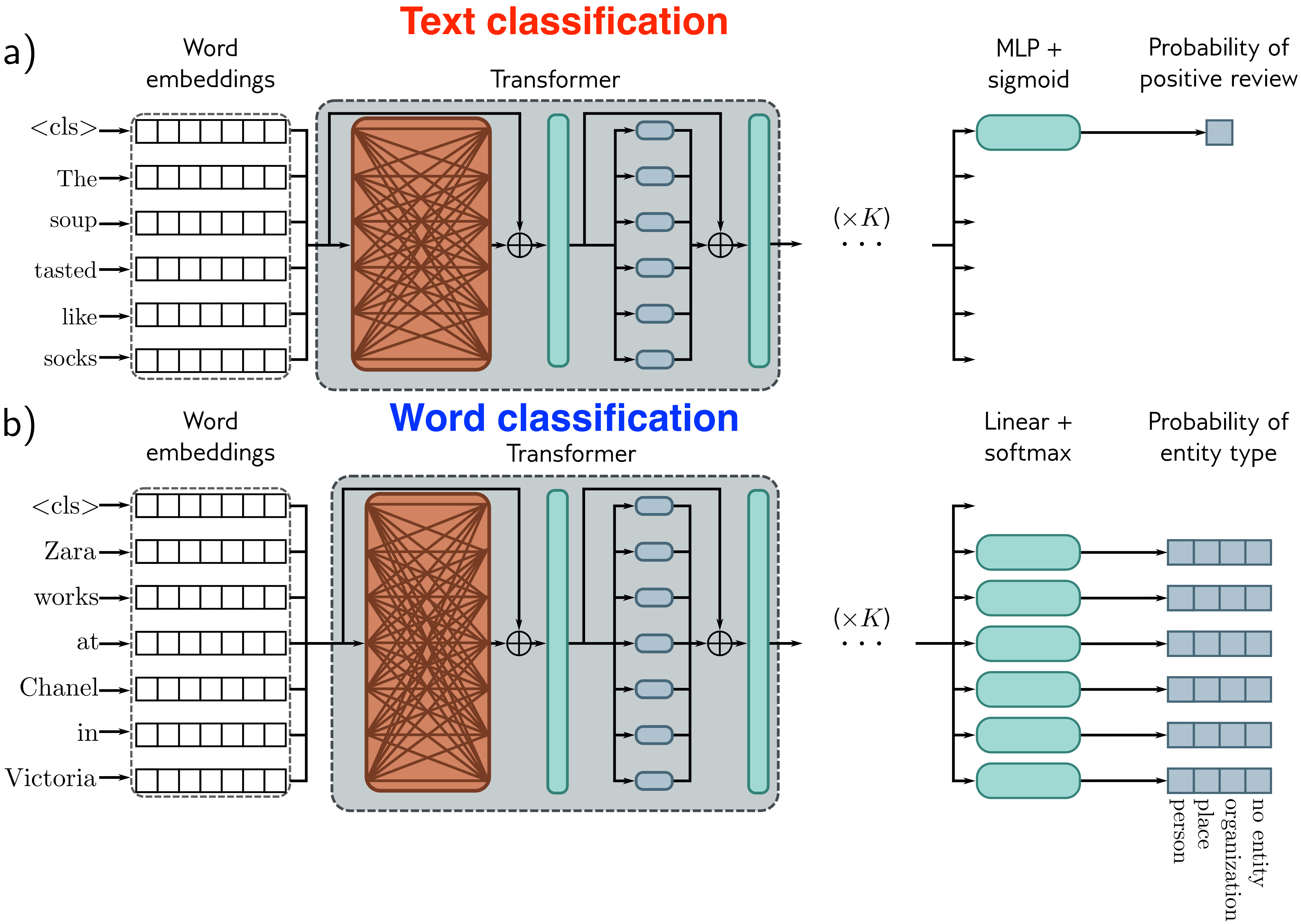
- For BERT, the self-supervision task consists of predicting missing words from sentences from a large internet corpus.



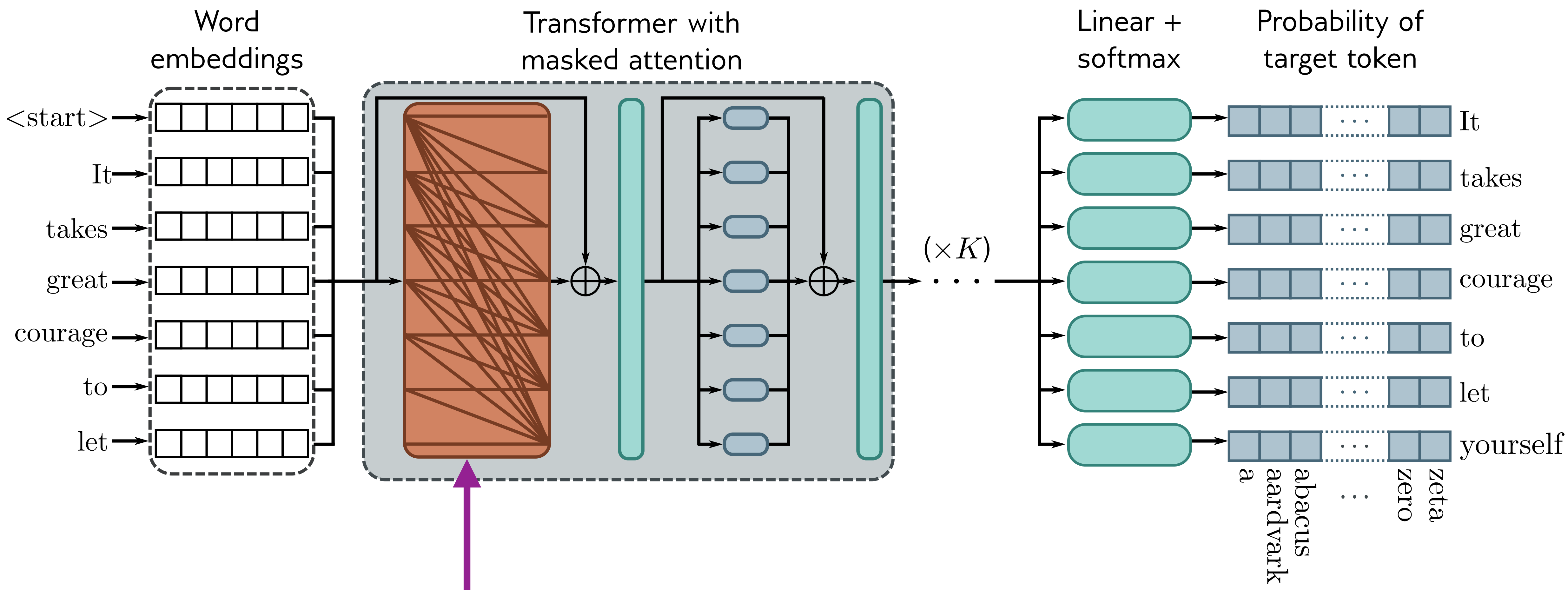
- Predicting missing words forces the transformer model to understand some syntax. For example, red is often found before car or dress than swim. In the above example, train is more likely than lasagna.

Above plots are from Prince Deep Learning book

Encoder Fine-tuning



Decoder Pre-Training



attend only to the current and previous tokens

Decoder: Text generation via Sampling

- The autoregressive language model is a **generative model**.
- Start with an input sequence of text, beginning with a <start> token.
- The outputs are the probabilities over possible subsequent tokens. **We can either pick the most likely token or sample from this probability distribution**.
- The new extended sequence can be fed back into the decoder network that outputs the probability distribution over the next token.
- At each step, **the decoder takes the entire sequence generated so far (including all previous tokens) as input and produces a probability distribution for the next token**. Then, you typically select one token from that distribution and append it to the sequence. This updated sequence, now containing the newly generated token, is then fed back into the decoder for the next prediction.
- Other strategies (instead of greedy search): **beam search and top-k sampling**, etc.

Transformers

Multimodal Transformers

Multi-modality

- Due to their generality, transformers **have become the state-of-the-art for many different modalities, including text, image, video, point cloud, and audio data**, and **have been used for both discriminative and generative applications** within each of these.
- The core architecture of the transformer layer has remained relatively constant, both over time and across applications. Therefore, **the key innovations that enabled the use of transformers in areas other than natural language have largely focused on the representation and encoding of the inputs and outputs**.
- One big advantage of a single architecture that is capable of processing many different kinds of data is that **it makes multimodal computation relatively straightforward**.
- For example, we may wish to generate an image from a text prompt.

Vision Transformer (Encoder)

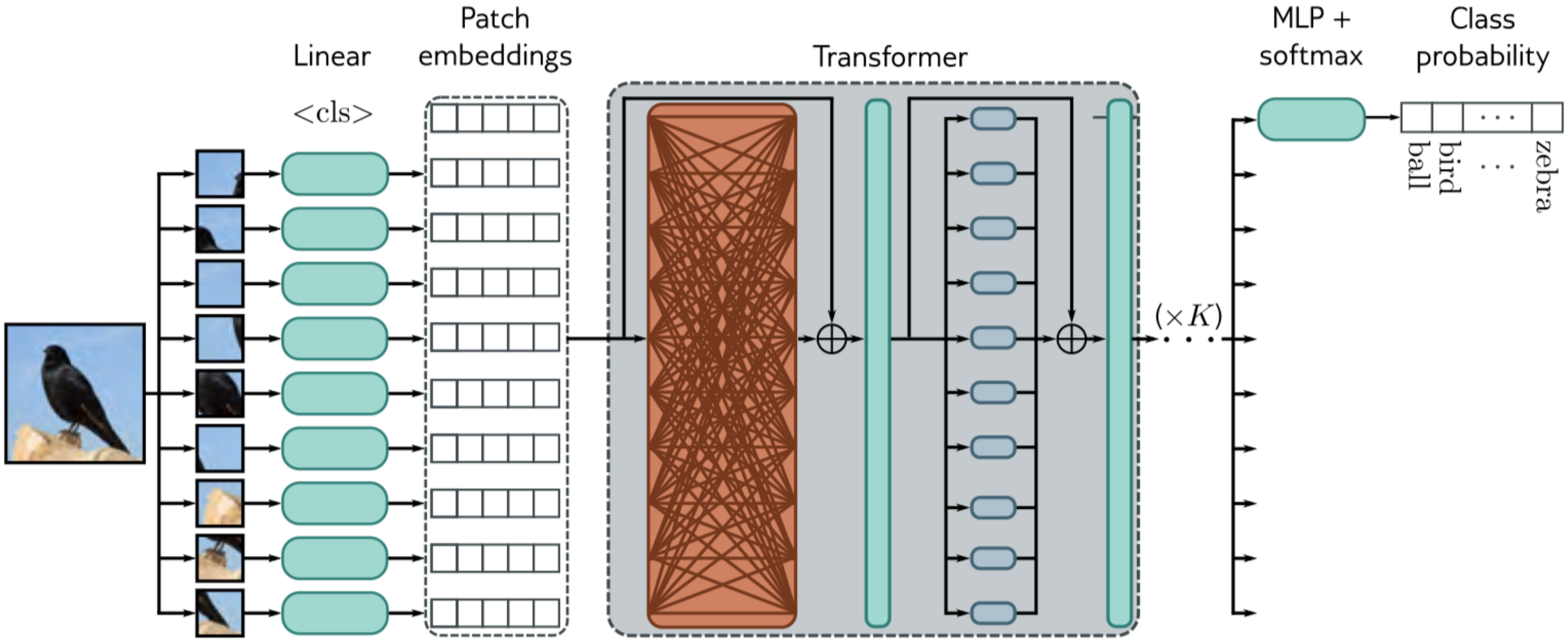
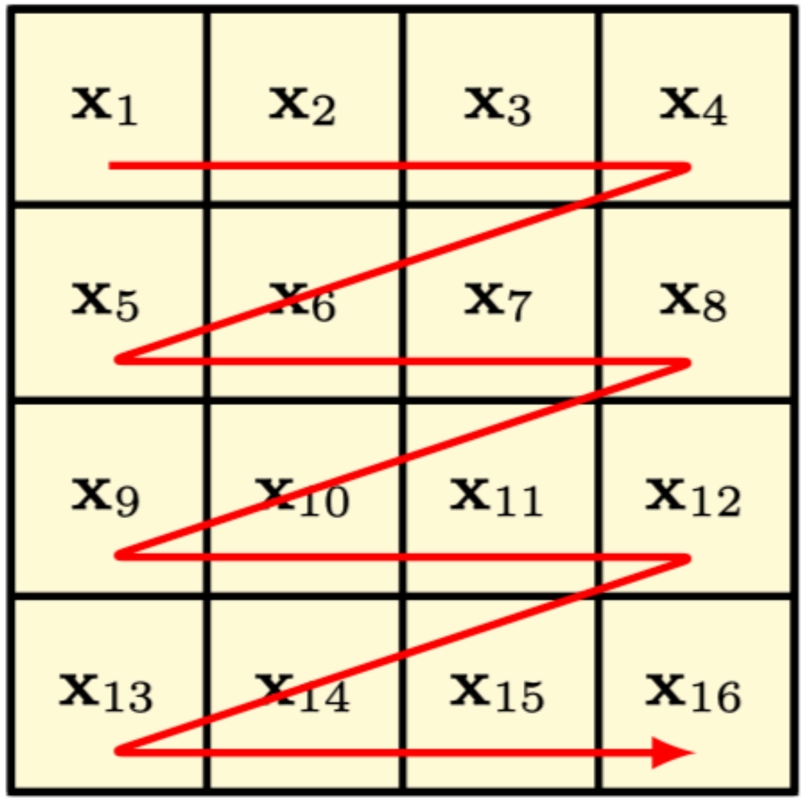


Figure 12.17 Vision transformer. The Vision Transformer (ViT) breaks the image into a grid of patches (16×16 in the original implementation). Each of these is projected via a learned linear transformation to become a patch embedding. These patch embeddings are fed into a transformer encoder network, and the <cls> token is used to predict the class probabilities.

Input vectors:
Each patch has dimension: 16 × 16 × 3 = 768 (flattened into a vector).
Each patch is not a discrete token from a fixed vocabulary (like words in NLP) but rather **a continuous-valued vector**.
Positional embeddings are learned. The length of the sequence is fixed.

Generating images (Decoder)

Illustration of a *raster scan* that defines a specific linear ordering of the pixels in a two-dimensional image.



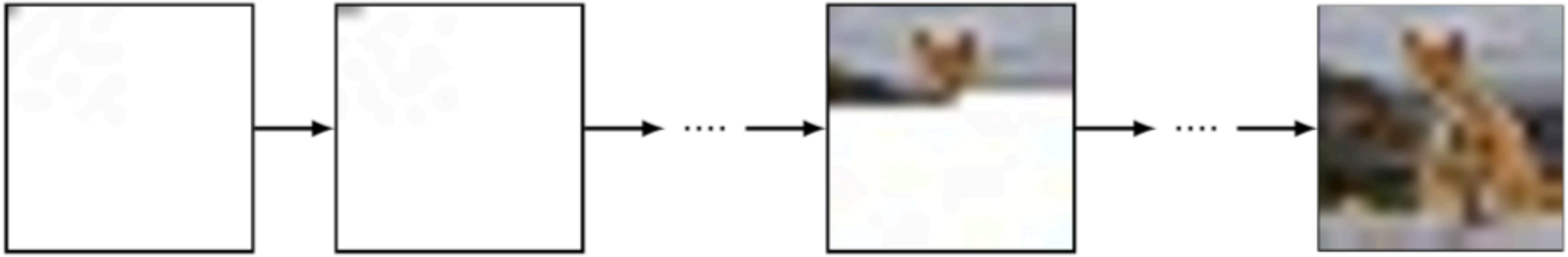


Figure 12.24 An illustration of how an image can be sampled from an autoregressive model. The first pixel is sampled from the marginal distribution $p(\mathbf{x}_{11})$, the second pixel from the conditional distribution $p(\mathbf{x}_{12}|\mathbf{x}_{11})$, and so on in raster scan order until we have a complete image.

Text-to-speech

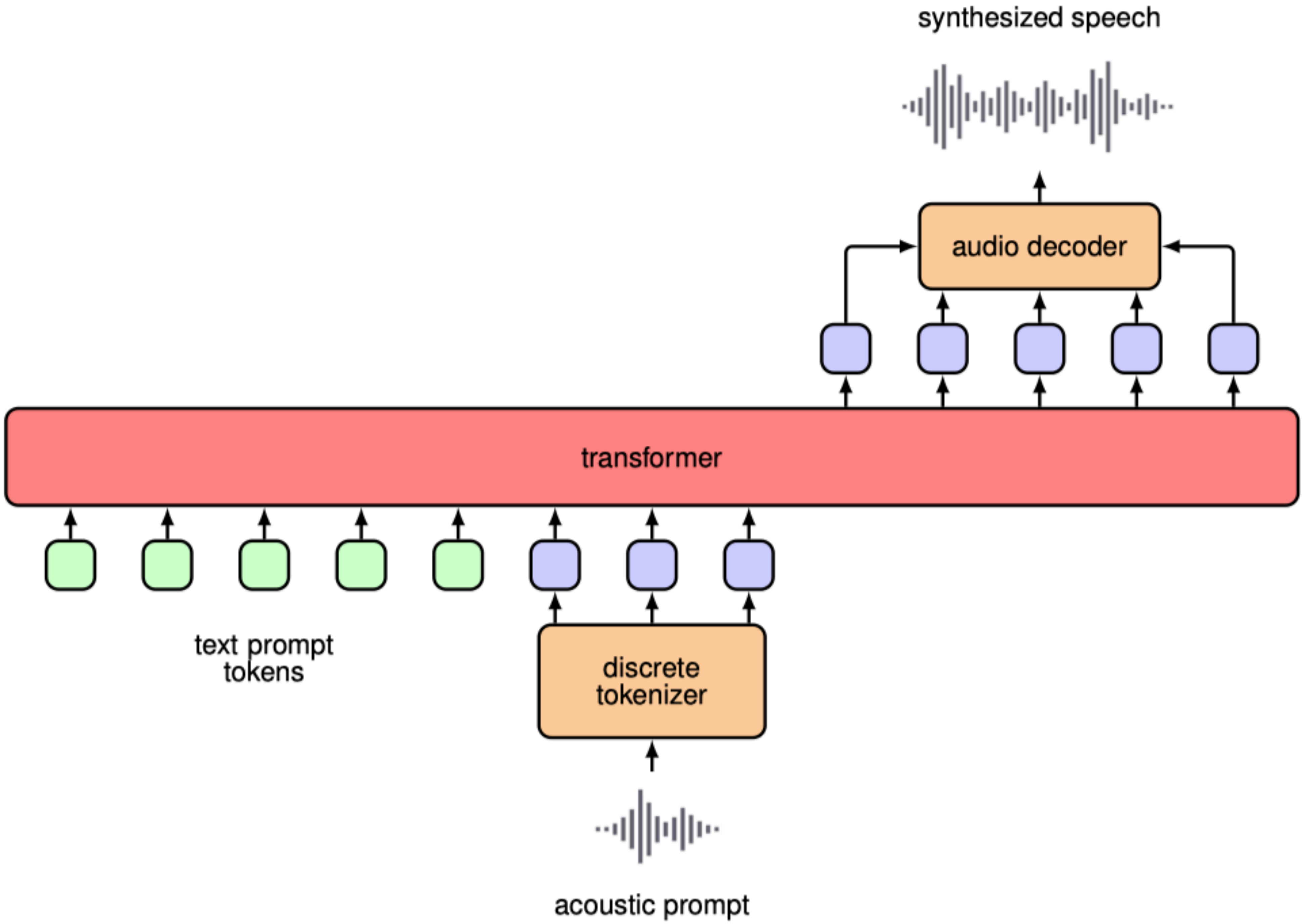


Figure 12.26 A diagram showing the high-level architecture of Vall-E. The input to the transformer model consists of standard text tokens, which prompt the model as to what words the synthesized speech should contain, together with acoustic prompt tokens that determine the speaker style and tone information. The sampled model output tokens are decoded back to speech with the learned decoder. For simplicity, the positional encodings and linear projections are not shown.

Above plots are from Bishop Deep Learning book

Transformers

Foundation Models for Science

Foundation Models for Science

- Like we do for language (LLMs), we can **pre-train large multimodal transformers with scientific data from simulations, publications, experiments etc.**
- After self-supervised pre-training, **the model should have some understanding of the data**. It will have found useful representations and connections between them.
- We can then fine-tune the model for various **“downstream tasks”**, as we did for the transformer encoder BERT.
- There is a lot of ongoing research in this area.

Potential benefits

- Foundation models for science provide a powerful, general-purpose framework for accelerating discovery across multiple scientific domains.
- Some potential benefits
  - Foundation models (FMs) **can be trained on vast amounts** of structured and unstructured scientific data, including research papers, experimental results, and simulations.
  - FMs **can speed up traditional simulations by acting as efficient surrogates** for computationally expensive processes
  - They can optimize lab experiments by recommending parameters to test, reducing time and costs.
  - Models **may suggest novel hypotheses, design experiments, and even predict unknown scientific relationships** by leveraging their training data and making connections that humans may have overlooked.
- We will look at a few examples now.

https://arxiv.org/pdf/2306.00258

Towards Foundation Models for Scientific Machine Learning: Characterizing Scaling and Transfer Behavior

Shashank Subramanian
shashanksubramanian@lbl.gov
Lawrence Berkeley National Lab

Peter Harrington
pharrington@lbl.gov
Lawrence Berkeley National Lab

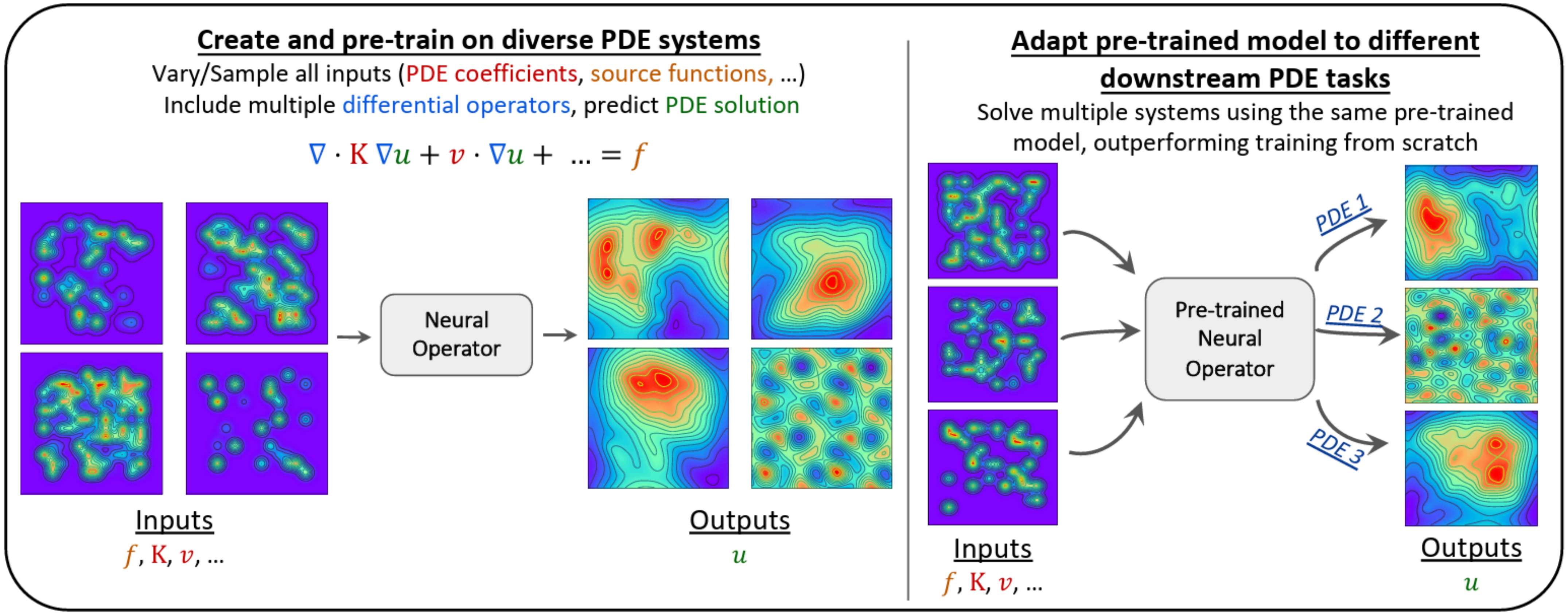
Kurt Keutzer
keutzer@eecs.berkeley.edu
UC Berkeley

Wahid Bhimji
wbhimji@lbl.gov
Lawrence Berkeley National Lab

Dmitriy Morozov
dmorozov@lbl.gov
Lawrence Berkeley National Lab

Michael W. Mahoney
mmahoney@stat.berkeley.edu
LBNL, ICSI, and UC Berkeley

Amir Gholami
amirgh@berkeley.edu
ICSI, UC Berkeley



Create and pre-train on diverse PDE systems
Vary/Sample all inputs (PDE coefficients, source functions, ...)
Include multiple differential operators, predict PDE solution
$\nabla \cdot K \nabla u + v \cdot \nabla u + \ \dots = f$
Neural Operator
Inputs
f, K, v, ...
Outputs
u
Adapt pre-trained model to different downstream PDE tasks
Solve multiple systems using the same pre-trained model, outperforming training from scratch
Pre-trained Neural Operator
PDE 1
PDE 2
PDE 3
Inputs
f, K, v, ...
Outputs
u

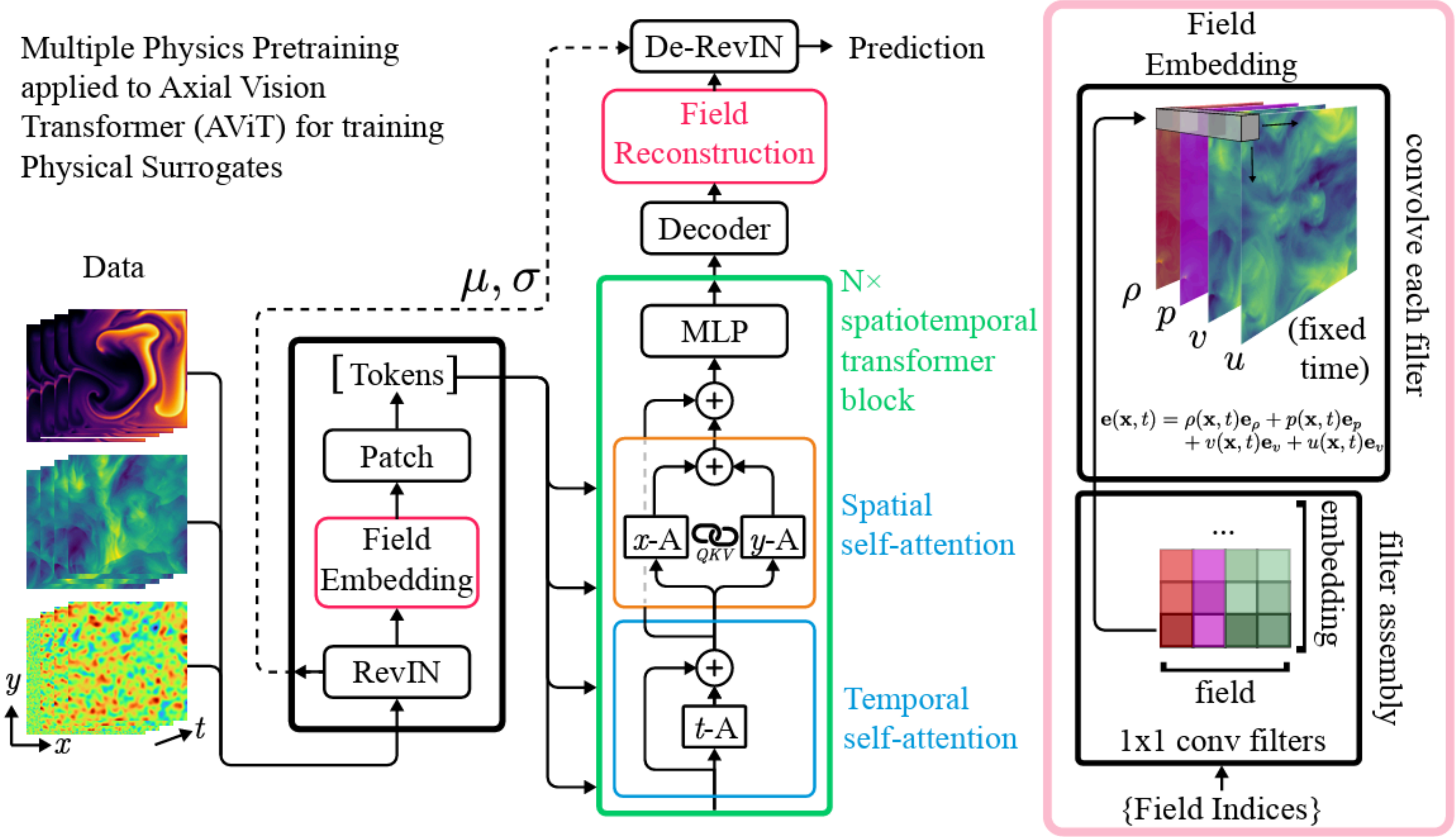
Figure 1: *Our setup consists of creating diverse training datasets, sampling both PDE coefficients and source functions simultaneously with different PDE operators and input data (coefficients, sources) distributions for pre-training. A neural operator is then pre-trained to predict the PDE solutions given these inputs and the ground truth solutions (computed through PDE solvers). The pre-trained model is then adapted with minimal fine-tuning (zero-shot or few-shot), and it is used in various downstream tasks (PDE systems) that can be in-domain or out-of-domain from the pre-training datasets. The pre-training with multiple solution operators allows the same model to transfer to several very different systems. For instance, PDE 2 (Helmholtz) manifests highly oscillatory solutions compared to, say, PDE 1 (Advection-Diffusion) or PDE 3 (Poisson's). We further characterize the scaling and transfer properties of this model as a function of downstream data scale and model size scale.*

Computer Science > Machine Learning

[Submitted on 4 Oct 2023 (v1), last revised 10 Dec 2024 (this version, v2)]

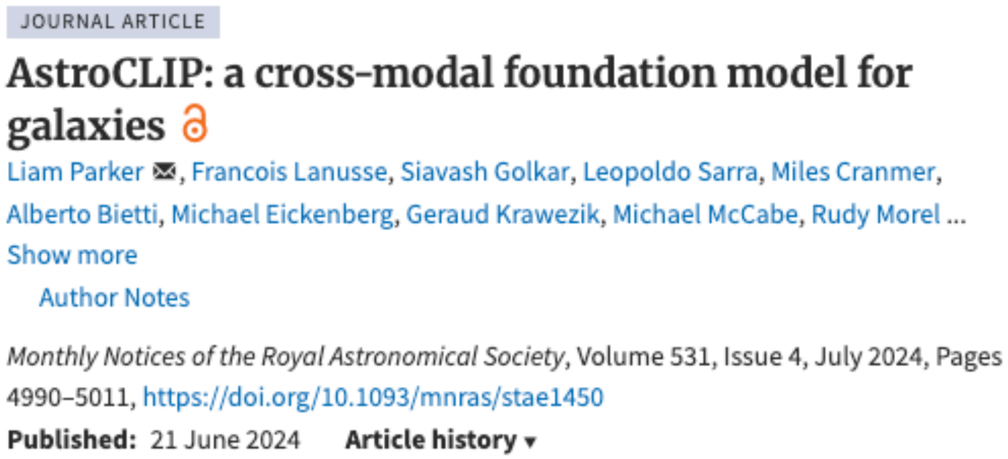
Multiple Physics Pretraining for Physical Surrogate Models

Michael McCabe, Bruno Régaldo-Saint Blancard, Liam Holden Parker, Ruben Ohana, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Siavash Golkar, Geraud Krawezik, Francois Lanusse, Mariel Pettee, Tiberiu Tesileanu, Kyunghyun Cho, Shirley Ho



Multiple Physics Pretraining applied to Axial Vision Transformer (AViT) for training Physical Surrogates

Figure 2: (Left) MPP works by individually normalizing each example using Reversible Instance Normalization (RevIN) then embedding each field individually into a shared, normalized space. A single transformer backbone can then predict the next step for multiple sets of physics. We use an AViT backbone which attends over space and time axis sequentially. Spatial attention is further split by axis, though these share linear projection weights. (Right) The embedding and reconstruction matrices are formed by subsampling a larger 1×1 convolutional filter based on input fields.



JOURNAL ARTICLE

AstroCLIP: a cross-modal foundation model for galaxies

Liam Parker, Francois Lanusse, Siavash Golkar, Leopoldo Sarra, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Geraud Krawezik, Michael McCabe, Rudy Morel ...
Show more

Author Notes

Monthly Notices of the Royal Astronomical Society, Volume 531, Issue 4, July 2024, Pages 4990–5011, https://doi.org/10.1093/mnras/stae1450

Published: 21 June 2024 **Article history** ▾

https://arxiv.org/abs/2310.03024

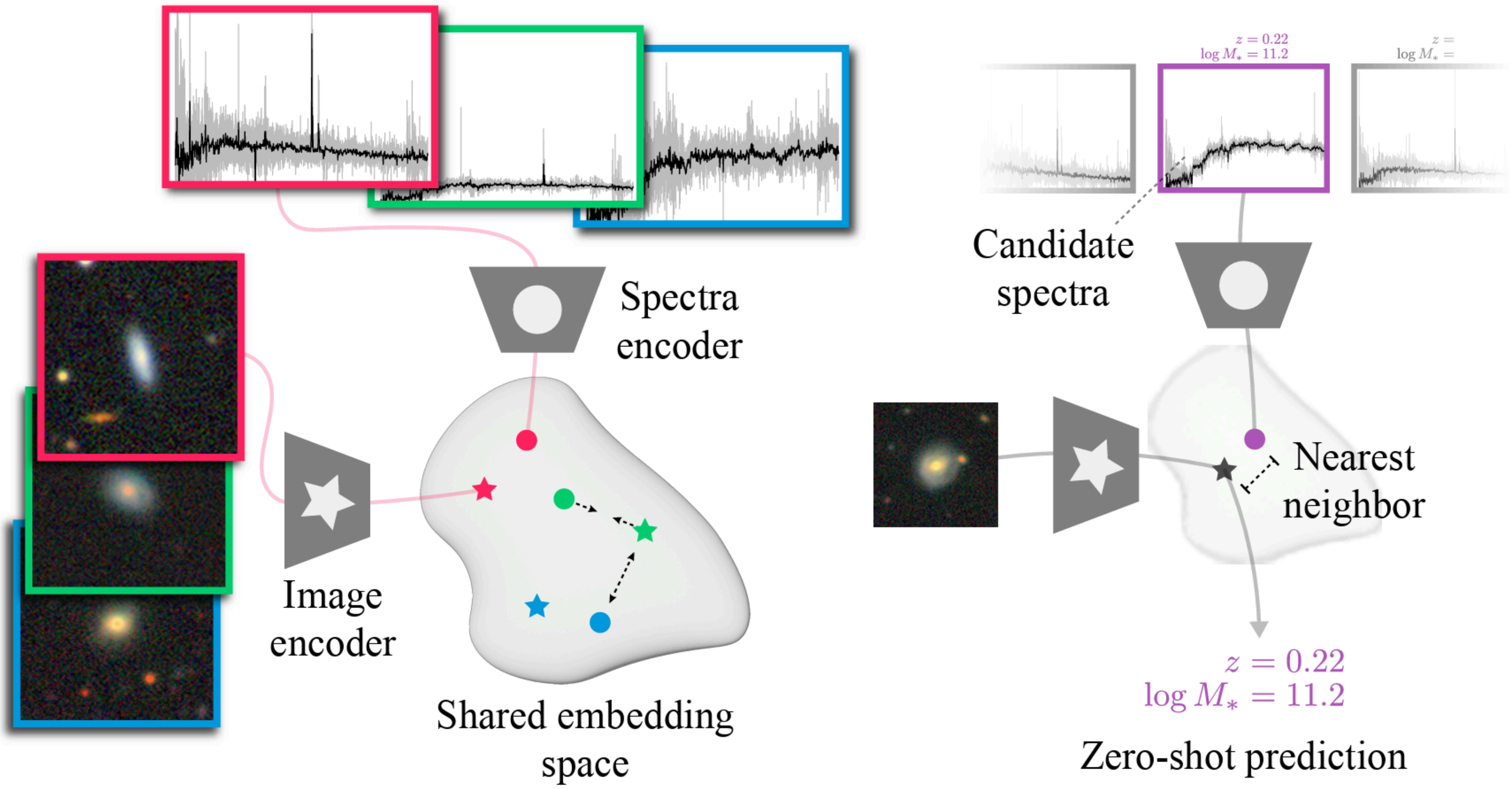
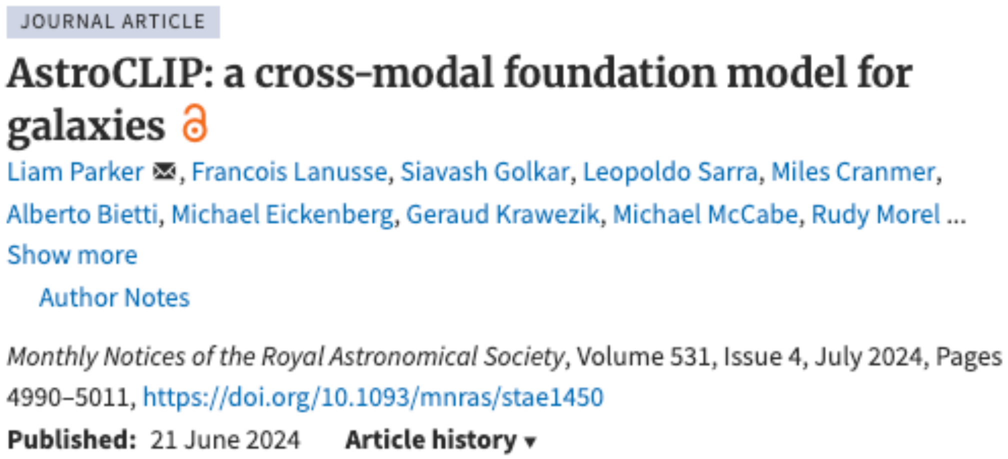


Figure 1. Illustration of the *AstroCLIP* cross-modal training strategy. This approach consists of two steps. First, galaxy images and spectra are embedded separately by pretraining both an image and a spectrum encoder in a SSL setting. Then, these encoders are aligned using a cross-modal contrastive loss. Once aligned, these embeddings allow us to connect and compare cross-modal representations. Moreover, they possess physically meaningful high-level information which can be used for a variety of downstream tasks on which the model was neither trained nor fine-tuned.



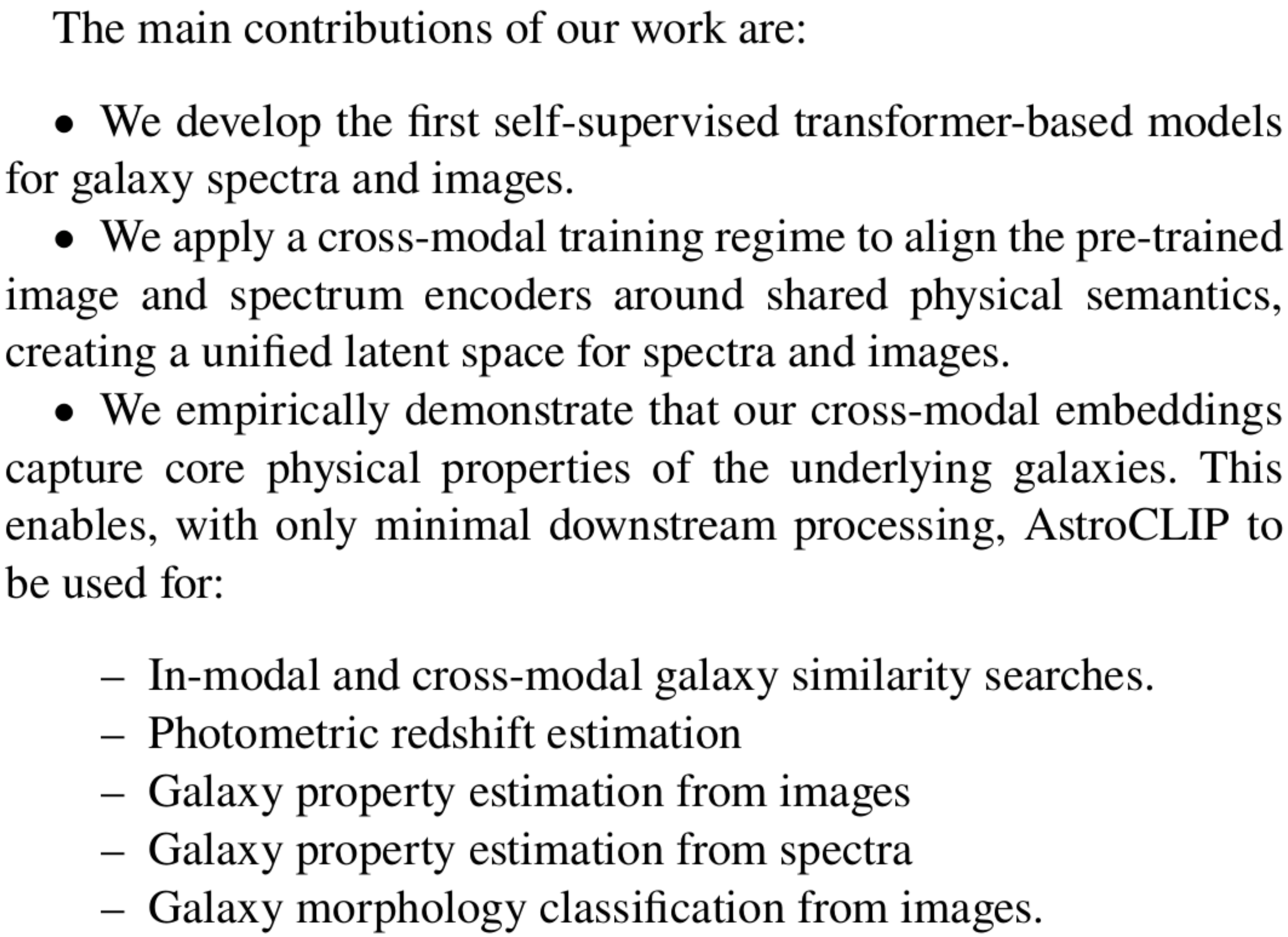
JOURNAL ARTICLE

AstroCLIP: a cross-modal foundation model for galaxies

Liam Parker ✉, Francois Lanusse, Siavash Golkar, Leopoldo Sarra, Miles Cranmer, Alberto Bietti, Michael Eickenberg, Geraud Krawezik, Michael McCabe, Rudy Morel ...
Show more
Author Notes

Monthly Notices of the Royal Astronomical Society, Volume 531, Issue 4, July 2024, Pages 4990–5011, https://doi.org/10.1093/mnras/stae1450
Published: 21 June 2024 **Article history** ▾

https://arxiv.org/abs/2310.03024



The main contributions of our work are:

- We develop the first self-supervised transformer-based models for galaxy spectra and images.
- We apply a cross-modal training regime to align the pre-trained image and spectrum encoders around shared physical semantics, creating a unified latent space for spectra and images.
- We empirically demonstrate that our cross-modal embeddings capture core physical properties of the underlying galaxies. This enables, with only minimal downstream processing, AstroCLIP to be used for:
  - In-modal and cross-modal galaxy similarity searches.
  - Photometric redshift estimation
  - Galaxy property estimation from images
  - Galaxy property estimation from spectra
  - Galaxy morphology classification from images.

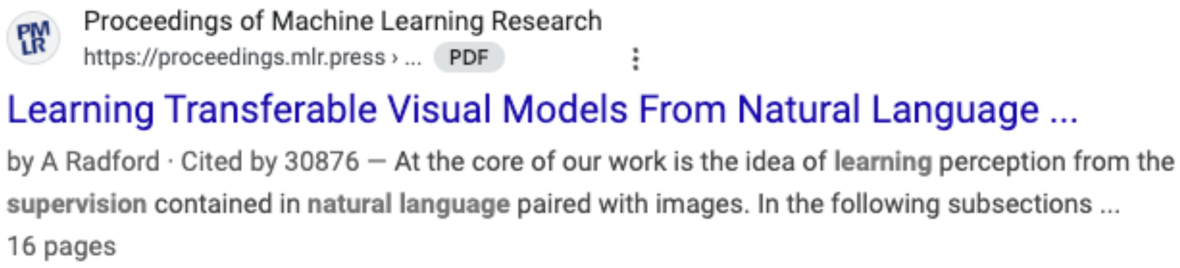
Cross-Modal Contrastive Techniques

- There are several techniques to combine representations across modalities
- A popular one is Contrastive Language–Image Pretraining (CLIP), designed to align language-based descriptions with their corresponding images. CLIP learns **general visual concepts** by contrasting correct image-text pairs against incorrect ones.

Training Pipeline:

- **Encoders. CLIP uses two separate models:**
  - Image Encoder (Vision Transformer or ResNet): Converts an image into an embedding.
  - Text Encoder (Transformer like GPT): Converts a caption into an embedding.
- **Contrastive Learning Objective:**
  - For a given batch of N image-text pairs:
    - Compute embeddings for all N images and N text captions.
    - Compute the similarity (dot product) between every image-text pair.
    - Optimize the model such that:
      - Correct (image, caption) pairs have high similarity.
      - Incorrect pairs have low similarity.
  - This is done using a contrastive loss function (e.g., InfoNCE loss).

CLIP objective



Proceedings of Machine Learning Research
https://proceedings.mlr.press › ... PDF

Learning Transferable Visual Models From Natural Language ...

by A Radford · Cited by 30876 — At the core of our work is the idea of learning perception from the supervision contained in natural language paired with images. In the following subsections ...

16 pages

- Assume we have the **observations X and Y from two different modalities.**
- We would like to find a new shared embedding space where these observations are mapped to embeddings **x and y**.
- In this new embedding space, we want x and y to be close if they come from the same sample (e.g. matching text and image).
- Closeness can be measured by the normalized **cosine similarity**:

$$S_C(x_i, y_j) = \frac{x_i^T y_j}{\|x_i\|_2 \|y_j\|_2}$$

$$\|x_i\|_2 = \sqrt{\sum_j x_{i,j}^2}$$

- We want to optimize the normalized cosine similarity on the training data set. More precisely, we use the **InfoNCE loss**, which is a lower bound on the mutual information:

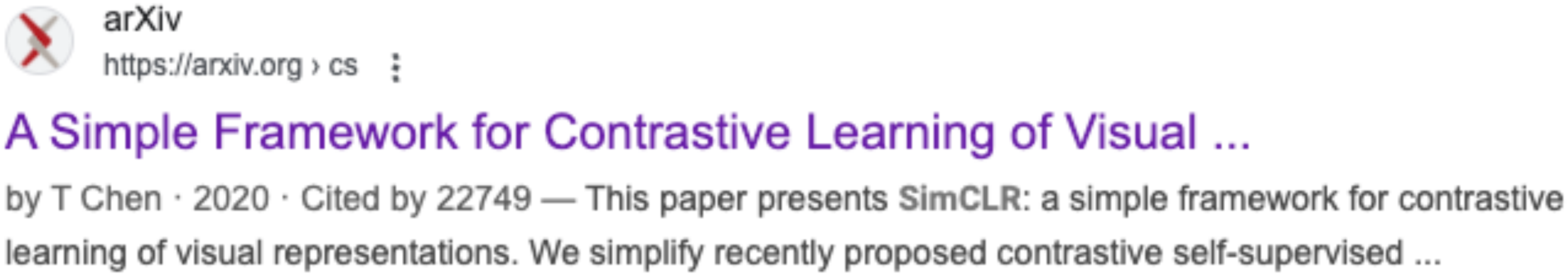
$$\mathcal{L}_{InfoNCE}(\mathbf{X}, \mathbf{Y}) = -\frac{1}{K} \sum_{i=1}^{K} \log \frac{\exp(S_C(\mathbf{x}_i, \mathbf{y}_i)/\tau)}{\sum_j^K \exp(S_C(\mathbf{x}_i, \mathbf{y}_j)/\tau)}$$

Here tau is a smoothing parameter called the temperature.

- Intuitively, the **InfoNCE objective works by bringing together points in the embedding space that correspond to the same underlying physical object and pushing points in the embedding space away from each other if they correspond to different underlying physical objects**.

Contrastive learning of Representations

- As we have seen, contrastive learning can be used in multimodal training to align different modalities.
- It is also used in “single modal” applications **with the goal of learning better representations**, especially when training data is sparse.
- In this case the model learns to differentiate **augmented views** of the same input from different samples.
  - This can be useful in physics. **We can corrupt the same image by different “systematics” and learn representations that are invariant under these systematics**.



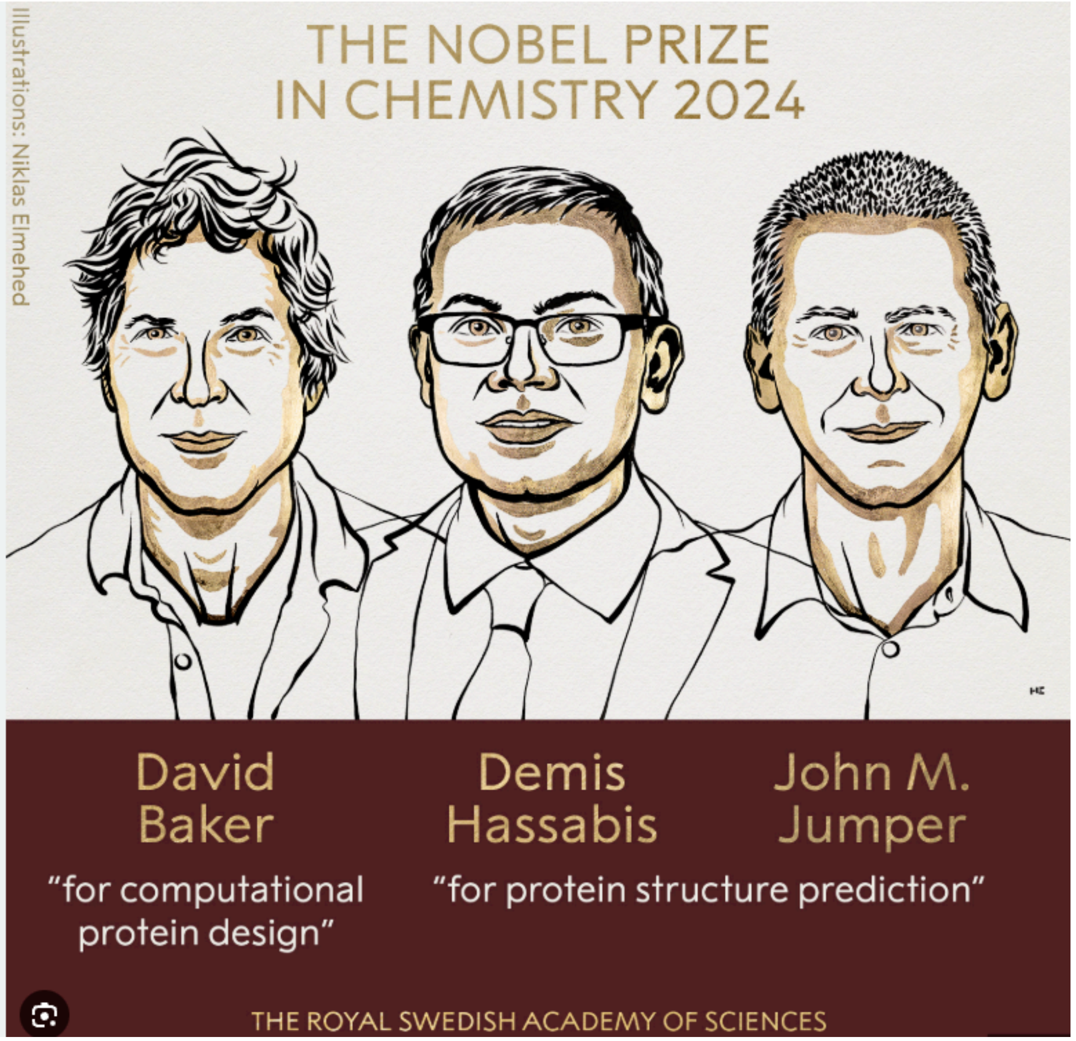
arXiv
https://arxiv.org › cs

A Simple Framework for Contrastive Learning of Visual ...

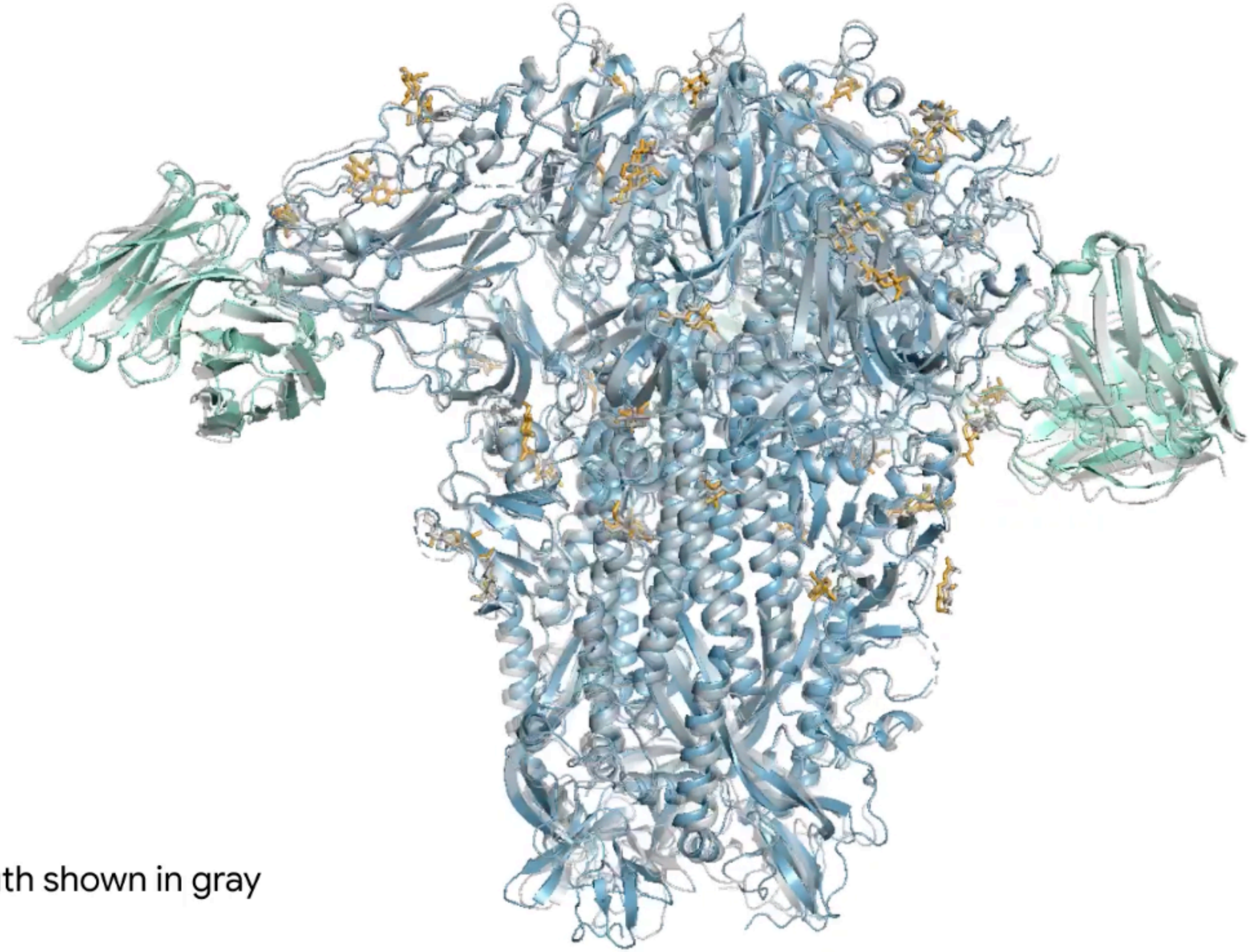
by T Chen · 2020 · Cited by 22749 — This paper presents SimCLR: a simple framework for contrastive learning of visual representations. We simplify recently proposed contrastive self-supervised ...

AlphaFold

- **Industrial scale AI for science example (involving hundreds of scientists)**
- AlphaFold is a deep learning-based foundation model for predicting protein 3D structures from amino acid sequences. Developed by DeepMind, it has revolutionized computational biology by providing near-experimental accuracy predictions, solving a decades-old scientific challenge.
- Trained on massive protein databases, AlphaFold can predict structures for almost any protein without task-specific fine-tuning.
- Uses a modified Transformer architecture (Evoformer) to model residue-residue interactions.
- AlphaFold is largely pretrained in a self-supervised manner, but it also incorporates supervised learning elements.
- For example it uses Contrastive Learning on Protein Representations: AlphaFold learns to distinguish between physically plausible and implausible protein structures based on data-driven constraints.


THE NOBEL PRIZE
IN CHEMISTRY 2024
Illustrations: Niklas Elmehed
David Baker
"for computational protein design"
Demis Hassabis
John M. Jumper
"for protein structure prediction"
THE ROYAL SWEDISH ACADEMY OF SCIENCES

7PNM



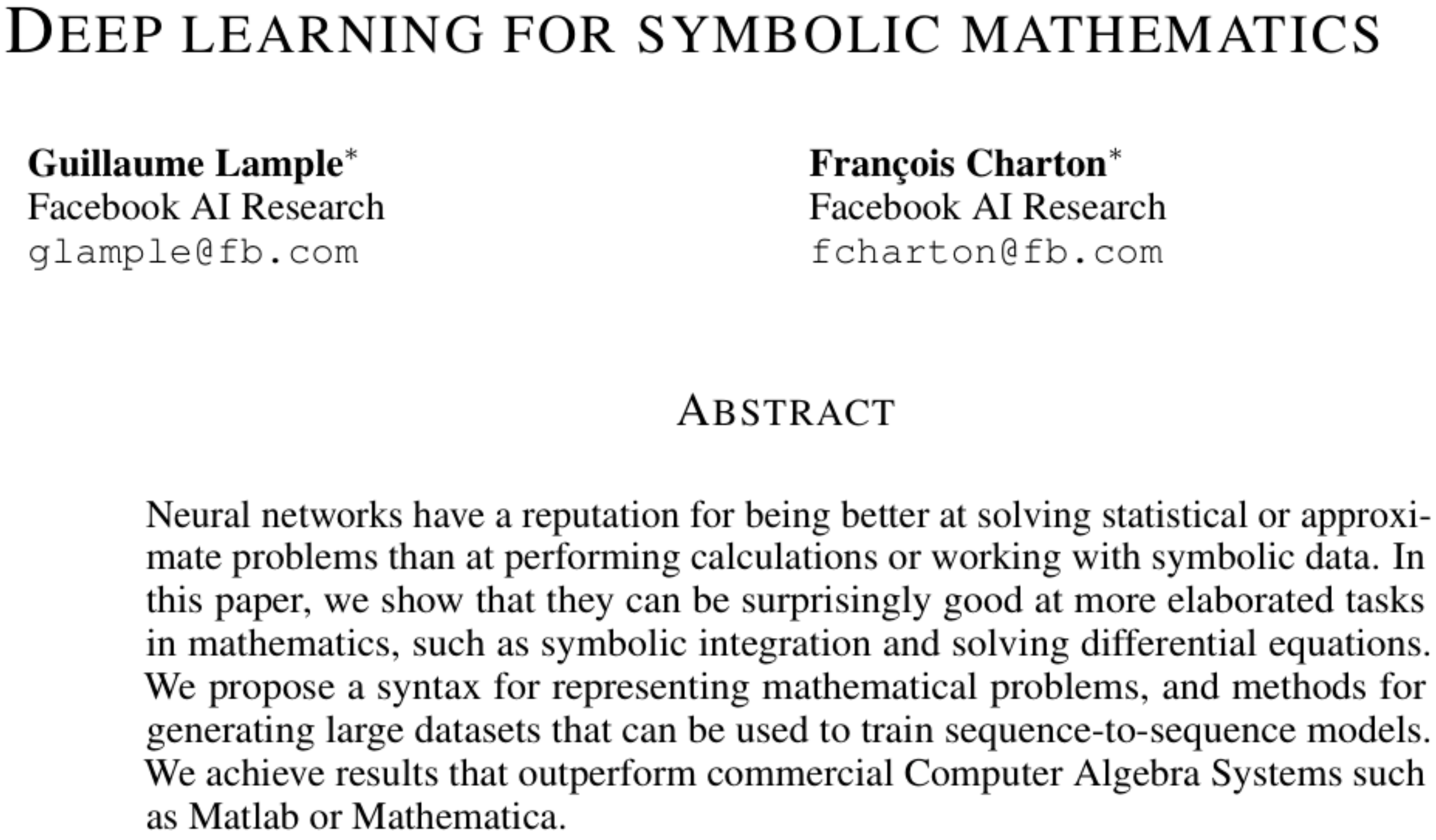
Ground truth shown in gray

7PNM - Spike protein of a common cold virus (Coronavirus OC43): AlphaFold 3's structural prediction for a spike protein (blue) of a cold virus as it interacts with antibodies (turquoise) and simple sugars (yellow), accurately matches the true structure (gray). The animation shows the protein interacting with an antibody, then a sugar. Advancing our knowledge of such immune-system processes helps better understand coronaviruses, including COVID-19, raising possibilities for improved treatments.

Transformers

Examples of Transformers for Math

Famous early work



DEEP LEARNING FOR SYMBOLIC MATHEMATICS

Guillaume Lample*
Facebook AI Research
glample@fb.com

François Charton*
Facebook AI Research
fcharton@fb.com

ABSTRACT

Neural networks have a reputation for being better at solving statistical or approximate problems than at performing calculations or working with symbolic data. In this paper, we show that they can be surprisingly good at more elaborated tasks in mathematics, such as symbolic integration and solving differential equations. We propose a syntax for representing mathematical problems, and methods for generating large datasets that can be used to train sequence-to-sequence models. We achieve results that outperform commercial Computer Algebra Systems such as Matlab or Mathematica.

https://arxiv.org/pdf/1912.01412

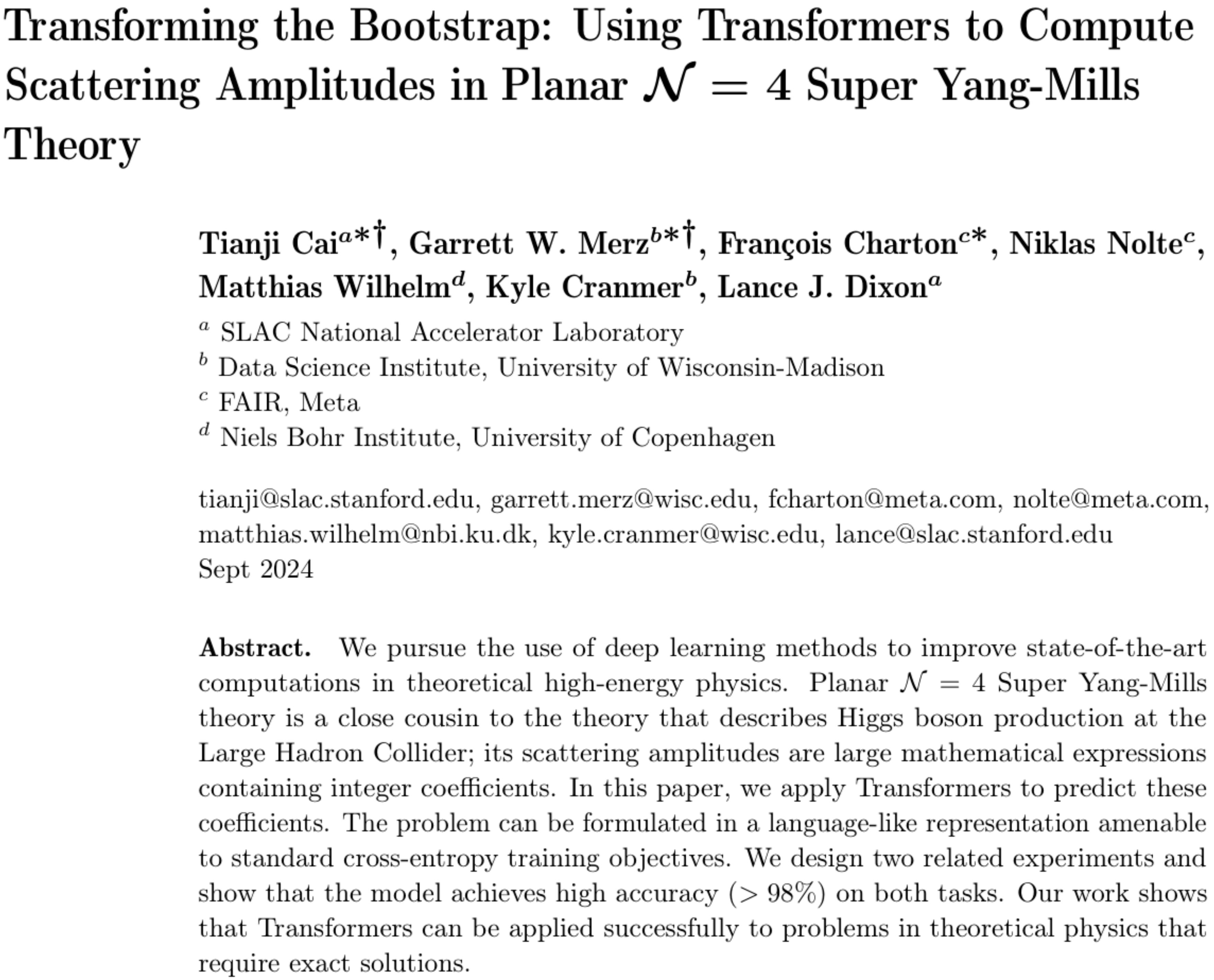
The paper includes methods to tokenize mathematical expressions and numbers.

Side note: For practically useful problems the above approach cannot replace CAS systems

Guessing the answer

- A **common problem setup** where transformers are helpful is the following:
    - For some mathematical problems, it is hard to find a solution, but easy to check if a solution is correct once it is found.
    - A famous example of that sort is the RSA algorithm based on prime factoring prime numbers.
- In such cases, we can try to train a transformer on pairs of problems and solutions of similar sort. The transformer **may be able to see patterns** in this data and get good at guessing answers to new problems.
- We can then let the transformer make lots of guesses, **and hopefully find some correct ones**.
- More than that, we **may be able to interpret the patterns that the transformer found theoretically**. This can be considered **“experimental mathematics”**.

An example of this sort from UW Physics



Transforming the Bootstrap: Using Transformers to Compute Scattering Amplitudes in Planar $\mathcal{N} = 4$ Super Yang-Mills Theory

Tianji Cai[a]*†, Garrett W. Merz[b]*†, François Charton[c]*, Niklas Nolte[c], Matthias Wilhelm[d], Kyle Cranmer[b], Lance J. Dixon[a]

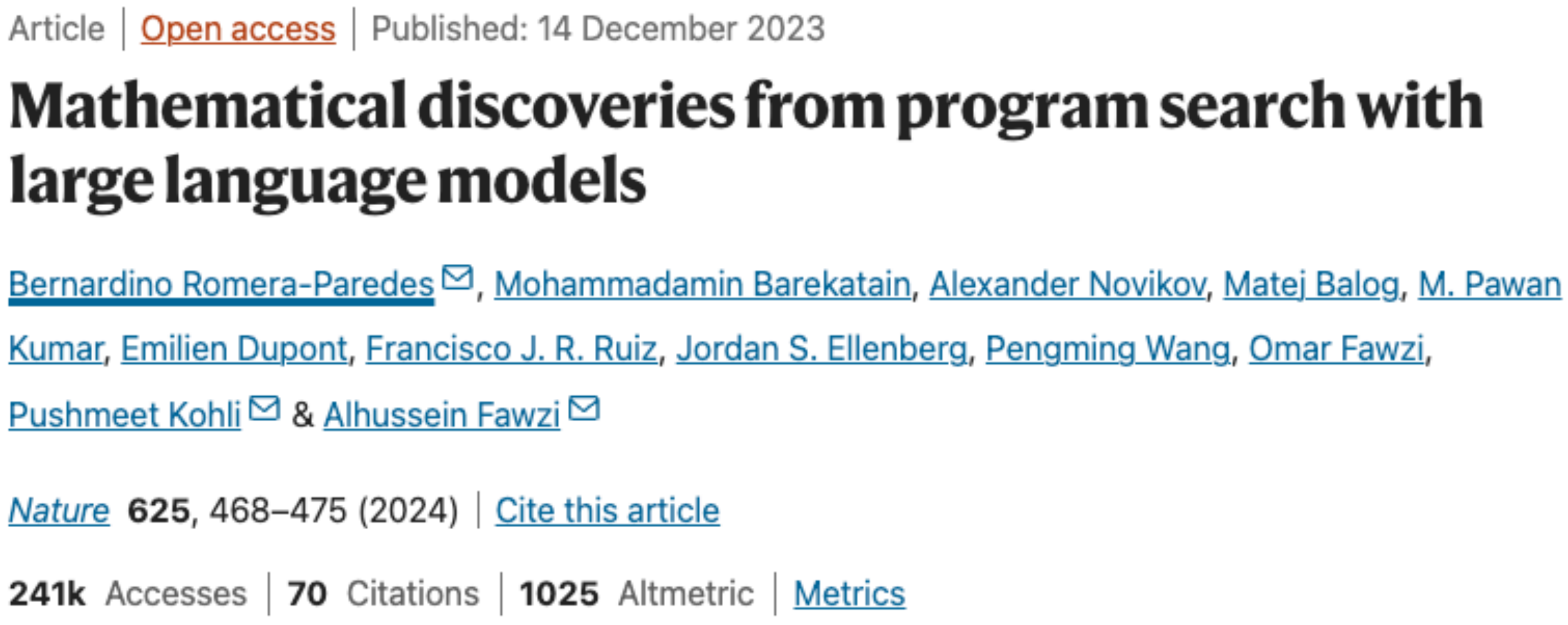
[a] SLAC National Accelerator Laboratory
[b] Data Science Institute, University of Wisconsin-Madison
[c] FAIR, Meta
[d] Niels Bohr Institute, University of Copenhagen

tianji@slac.stanford.edu, garrett.merz@wisc.edu, fcharton@meta.com, nolte@meta.com, matthias.wilhelm@nbi.ku.dk, kyle.cranmer@wisc.edu, lance@slac.stanford.edu
Sept 2024

Abstract. We pursue the use of deep learning methods to improve state-of-the-art computations in theoretical high-energy physics. Planar $\mathcal{N} = 4$ Super Yang-Mills theory is a close cousin to the theory that describes Higgs boson production at the Large Hadron Collider; its scattering amplitudes are large mathematical expressions containing integer coefficients. In this paper, we apply Transformers to predict these coefficients. The problem can be formulated in a language-like representation amenable to standard cross-entropy training objectives. We design two related experiments and show that the model achieves high accuracy ($> 98\%$) on both tasks. Our work shows that Transformers can be applied successfully to problems in theoretical physics that require exact solutions.

https://arxiv.org/abs/2405.06107

FunSearch (also including UW)



Article | Open access | Published: 14 December 2023

Mathematical discoveries from program search with large language models

Bernardino Romera-Paredes ✉, Mohammadamin Barekatain, Alexander Novikov, Matej Balog, M. Pawan Kumar, Emilien Dupont, Francisco J. R. Ruiz, Jordan S. Ellenberg, Pengming Wang, Omar Fawzi, Pushmeet Kohli ✉ & Alhussein Fawzi ✉

Nature **625**, 468–475 (2024) | Cite this article

241k Accesses | **70** Citations | **1025** Altmetric | Metrics

FunSearch is a method to search for new solutions in mathematics and computer science. FunSearch works by pairing a pre-trained LLM, whose goal is to provide creative solutions in **the form of computer code**, **with an automated “evaluator”**, which guards against hallucinations and incorrect ideas. By iterating back-and-forth between these two components, initial solutions “evolve” into new knowledge. The system searches for “functions” written in computer code; hence the name FunSearch.

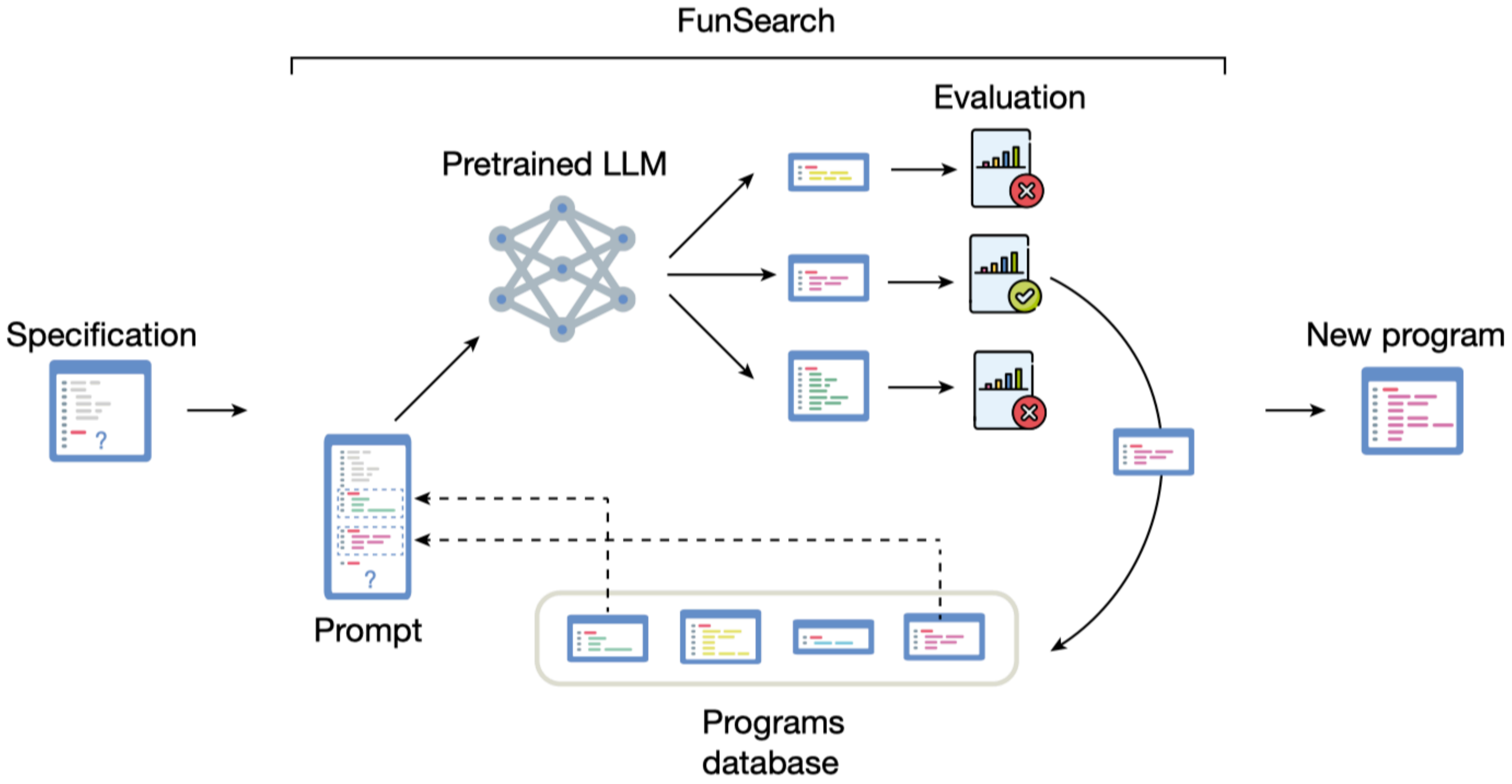
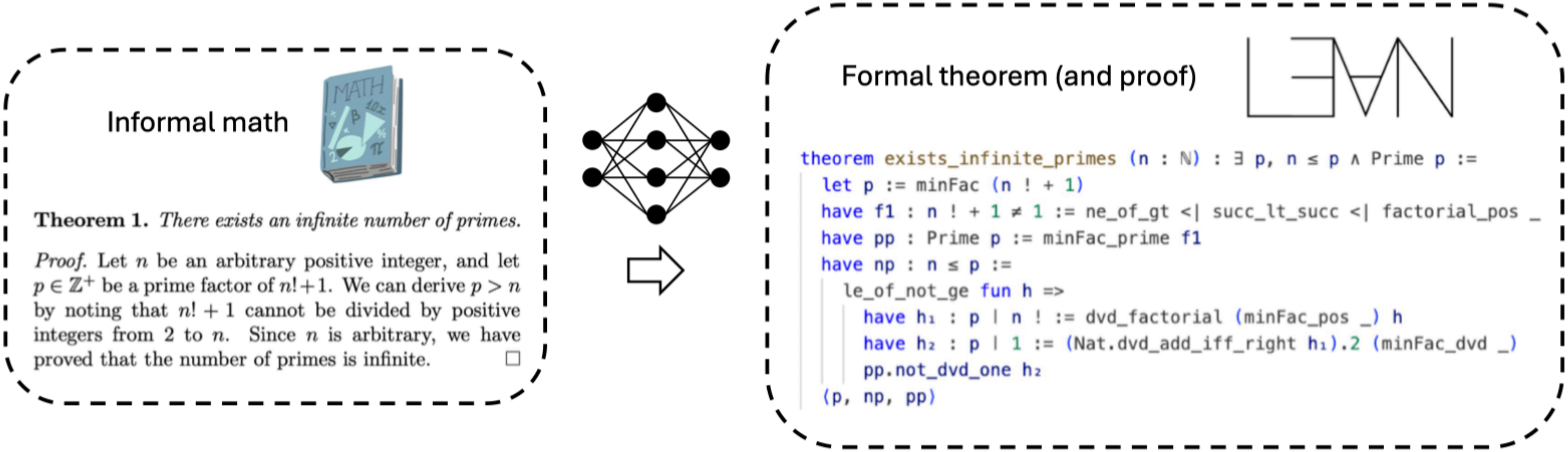


Fig. 1 | Overview of FunSearch. The input to FunSearch is a specification of the problem in the form of an 'evaluate' function, an initial implementation of the function to evolve, which can be trivial, and potentially a skeleton. At each iteration, FunSearch builds a prompt by combining several programs sampled from the programs database (favouring high-scoring ones). The prompt is then fed to the pretrained LLM and new programs are created. Newly created programs are then scored and stored in the programs database (if correct), thus closing the loop. The user can at any point retrieve the highest-scoring programs discovered so far.

Formal Theorem Proving

- Mathematical theorems can be formalized in languages and proof assistants such as LEAN, which allow proofs to be verified automatically.
- In principle, LLMs can come up with new candidate proofs, and then the proof can be automatically verified.



Theorem 1. *There exists an infinite number of primes.*

Proof. Let n be an arbitrary positive integer, and let $p \in \mathbb{Z}^+$ be a prime factor of $n!+1$. We can derive $p > n$ by noting that $n! + 1$ cannot be divided by positive integers from 2 to n. Since n is arbitrary, we have proved that the number of primes is infinite. □

```
theorem exists_infinite_primes (n : ℕ) : ∃ p, n ≤ p ∧ Prime p :=
  let p := minFac (n ! + 1)
  have f1 : n ! + 1 ≠ 1 := ne_of_gt <| succ_lt_succ <| factorial_pos _
  have pp : Prime p := minFac_prime f1
  have np : n ≤ p :=
    le_of_not_ge fun h =>
      have h₁ : p ∣ n ! := dvd_factorial (minFac_pos _) h
      have h₂ : p ∣ 1 := (Nat.dvd_add_iff_right h₁).2 (minFac_dvd _)
      pp.not_dvd_one h₂
  ⟨p, np, pp⟩
```

arXiv:2412.16075v1

- Can this be useful for physics too? Open question.

Course logistics

- This lecture did not follow a particular book.