CPSC 488/588, Fall 2023
AI Foundation Models

Yale University

Instructor: Arman Cohan

Logistics

- Instructor: Arman Cohan
- Course website: https://yale-nlp.github.io/cpsc488/

- Time: Tuesdays and Thursdays 9-10:15am
- Location: WTS A30

- Office hours:
  - Arman's office hours: Tuesday 10:45-11:45
  - (appointment-based) https://calendly.com/arman-cohan/office-hours

TAs

Yixin Liu
Simeng Han
Chen Liu

Office hours on the course website: https://yale-nlp.github.io/cpsc488/
If you need additional office hours reach out to us.

The TAs are all CS PhD students with a lot of relevant research experience!

Communications

- We use slack for communications
  - Workspace: cpsc488.slack.com
  - We will send invitation

- Announcements will be posted on Canvas

Anonymous questions, comments, thoughts

- Submit your questions, comments, or any feedback you have here:

  https://forms.gle/FFrnza7oPKcK8tDN9

Completely anonymous!

Course structure

- This course is about current topics in AI, NLP, and Generative AI (including several topics from other modalities such as vision)
- The objective is to cover state-of-the-art materials
  - Some materials might change as we progress through the semester
  - We will give 2 weeks notice
- Several exciting guest lectures from leaders in the field!

Colin Raffel
(UNC and Huggingface)

Sewon Min
(University of Washington)

Tushar Khot
(Allen AI)

Course structure

- Most of the course will focus on (written) natural language
- Why language:
  - Natural language data is abundant
  - Language is a universal communication medium: Bridges cultural and social gaps, empowering global interaction and collaboration
  - Many of the innovations, techniques and core concepts behind LLMs are transferable and used in models developed for other modalities
  - Largest scale models are designed for language
- We will also discuss key innovations behind Generative AI and foundation models in other modalities such as vision, speech, and biomedicine

Main objectives of the course:

- Getting familiar with the fundamentals and innovations that derive latest foundation models
- Discussion of latest foundation models and their properties
- Prepare students to perform cutting-edge research in NLP and beyond
- Help students build or improve research skills (from literature reviews and critiquing prior work, to brainstorming ideas and implementing them).

Main objectives of the course:

- Getting familiar with the fundamentals and innovations that derive latest foundation models
- Discussion of latest foundation models and their properties
- Prepare students to perform cutting-edge research in NLP and beyond
- Help students build or improve research skills (from literature reviews and critiquing prior work, to brainstorming ideas and implementing them).

Overview of the material: https://yale-nlp.github.io/cpsc488/schedule

Course structure - Prerequisites

- Students should feel comfortable with basics of Machine Learning
  - Fundamentals of ML including supervised learning, linear models, basic neural networks
  - Training ML models including gradient based learning and backpropagation
  - We will review some of the material as refresher
- Having taken an NLP or computer vision class is a plus:
  - CPSC 477/577 Natural Language Processing
  - CPSC 480/580 Computer vision
- Related courses
  - CPSC 670 Topics in NLP
  - CPSC 477/577 NLP

Course structure - Resources

- No required textbook. But if you are interested in textbooks or book chapters:
  - On the Opportunities and Risks of Foundation Models
    https://arxiv.org/pdf/2108.07258.pdf
  - Optional:
    Probabilistic Machine Learning, Kevin Murphy
    Available for free: https://probml.github.io/pml-book/
  - Optional:
    A Primer on Neural Network Models for Natural Language Processing.
    https://u.cs.biu.ac.il/~yogo/nnlp.pdf

- We will be reading research papers from premier conferences in the field
  E.g., ACL, EMNLP, NAACL, ICLR, NeurIPS, ICML, …

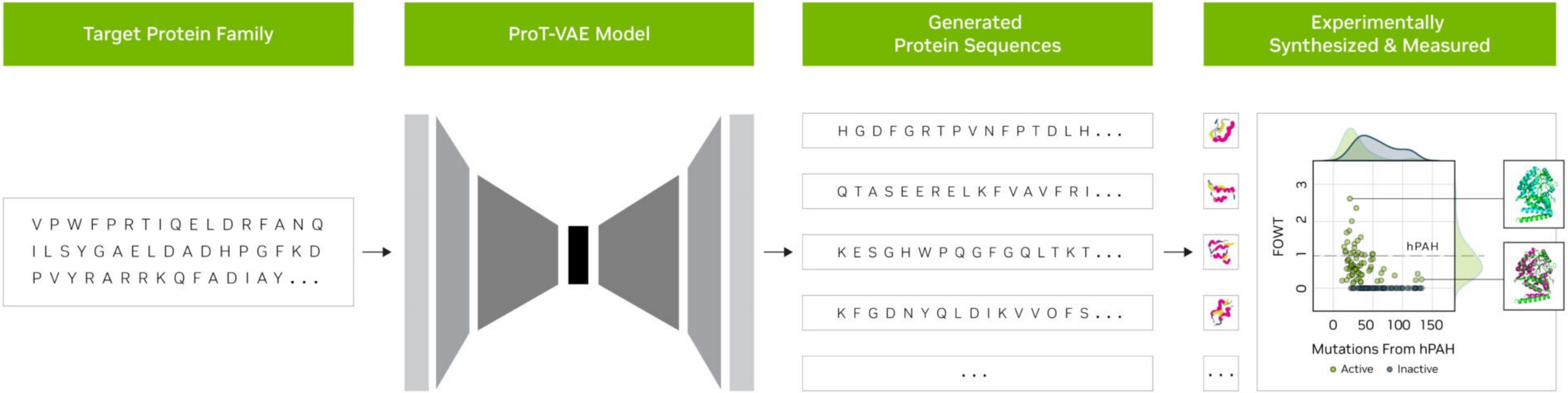
Current State of AI

Example generated with Midjourney

Prompt: a fallout isolated nuclear world

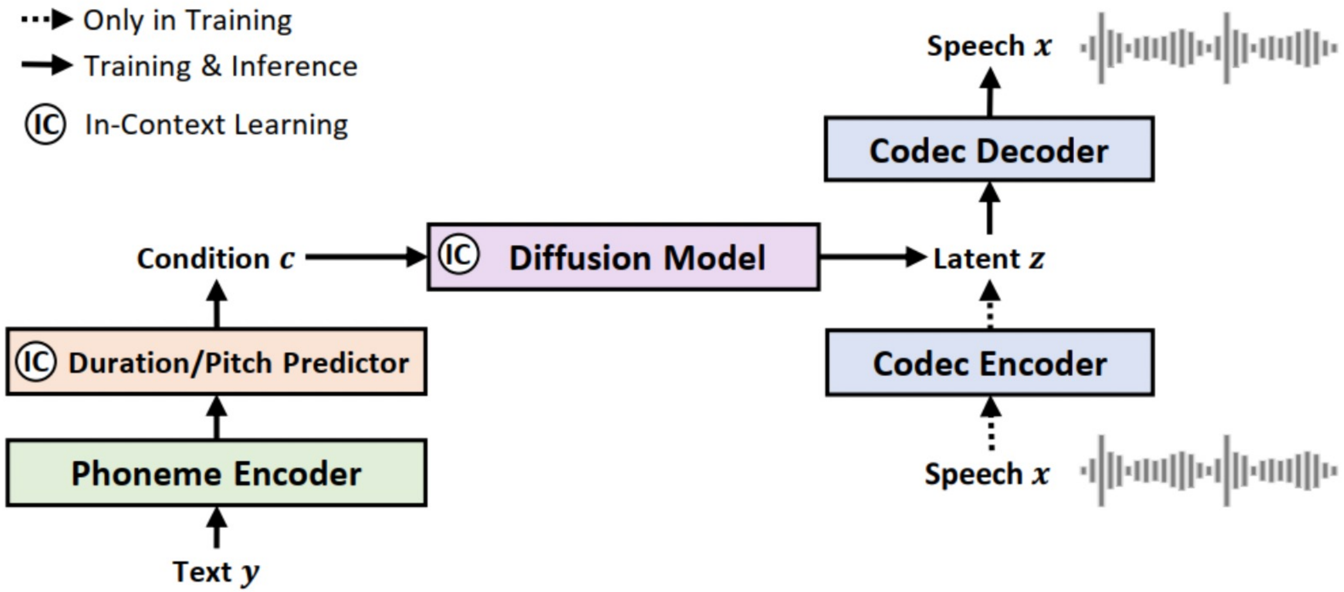
Example generated with Midjourney

Generative models for proteins



https://www.youtube.com/watch?v=oxXpB9pSETo&ab_channel=DiepNep

Text to speech



NaturalSpeech 2 (Shen et al., 2023)

Sample text: I had always known him to be restless in his manner, but on this particular occasion he was in such a state of uncontrollable agitation that it was clear something very unusual had occurred.

Example from: https://speechresearch.github.io/naturalspeech2/

Large Language Models

A Write a sentence about human landing on the moon where every word starts with an S.

Sure thing! Here's a sentence:

"Sixty-seven space scientists successfully surveyed Saturn's satellite, selecting safe sites."

Large Language Models

A

Write a sentence about human landing on the moon where every word starts with an S.

Sure thing! Here's a sentence:

"Sixty-seven space scientists successfully surveyed Saturn's satellite, selecting safe sites."

A

Do the same thing but make every word start with a B

Absolutely! Here you go:

"Brave Bob's bold balloon breached big blue beyond, boarding barren base."

Large Language Models

A

Write a paragraph (instead of a sentence). And start every word with B.

Certainly! Here's a paragraph where every word begins with the letter B:

"Brave Barry's big balloon bravely breached beautiful blue beyond. Boldly boarding barren base, Barry beheld breathtaking beauty. Below, boundless black blended brilliantly with bright burning bulbs. Bizarre, but bewitching, Barry's boldness became both breathtaking benchmark. Bafflingly, Barry believed boundless boundaries beckoned; beyond, brave beings believed. Becoming brave, Barry built bridges between baffling barriers, boldly breaking boundaries."

Large Language Models

A

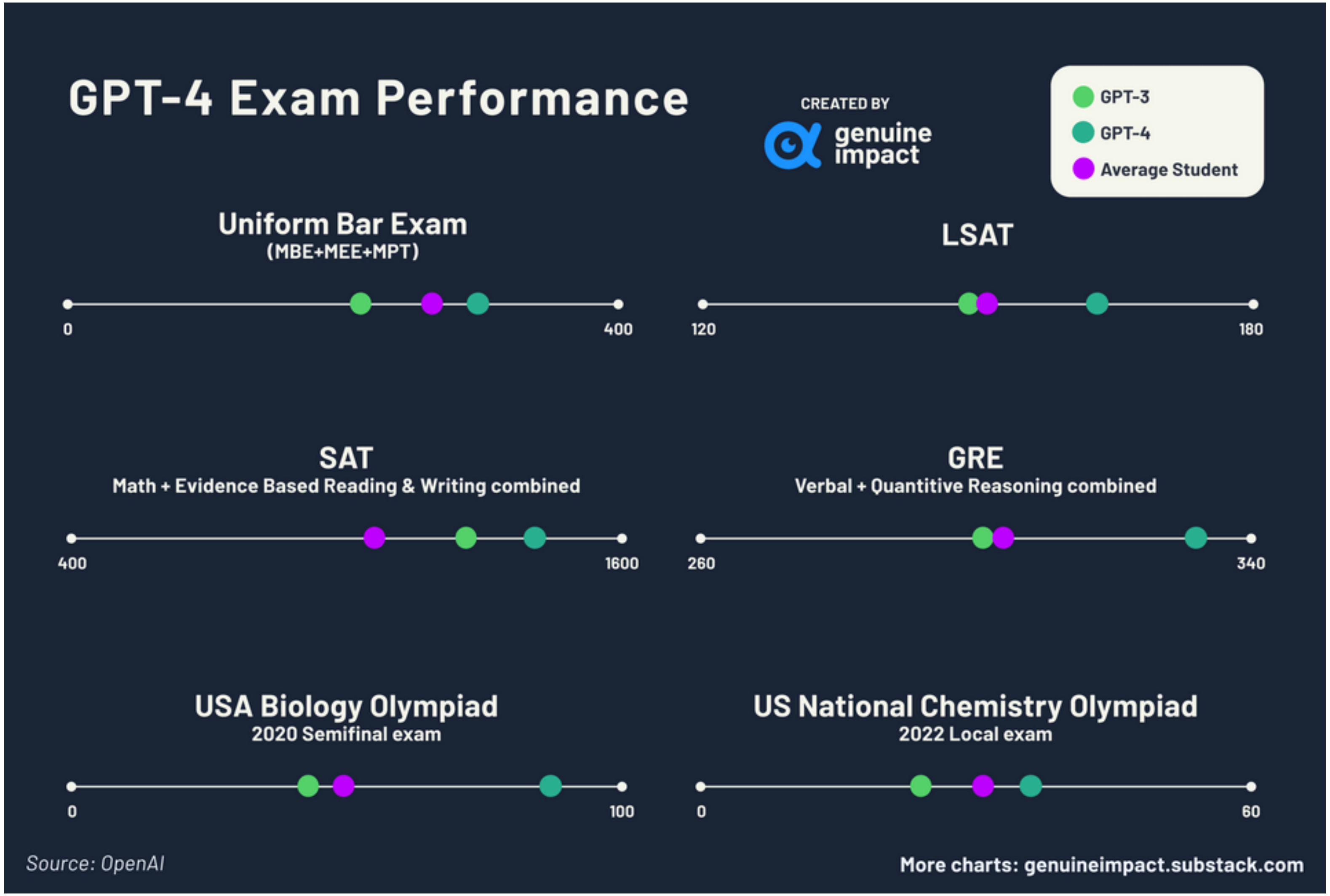
Do the same thing with U

Sure thing! The letter 'U' is a challenge, but here's a whimsical attempt:

"Ulysses undertook unusual underwater urchin upkeeping. Under unassuming umbrellas, urchins uniquely united, undergoing unexpected upheaval. Understanding Ulysses' unending urge, urchins uplifted, unveiling underwater universes unknown. Unbelievable, unlikely, unmatched unity unfolded. Unfaltering, Ulysses undertook uncharted underwater universes, unearthing unexplained undersea utopias."

This paragraph doesn't exactly relate to the moon, but it's a fun and imaginative exploration with the unique constraint! Enjoy!

Large Language Models



Progress in AI

- Traditional AI systems could run on single machines
- Modern AI systems need vast amounts of compute
- Many advances are due to neural networks
- How old are neural networks?
    - When was the first NN paper published?

Join by Web
PollEv.com/cohan

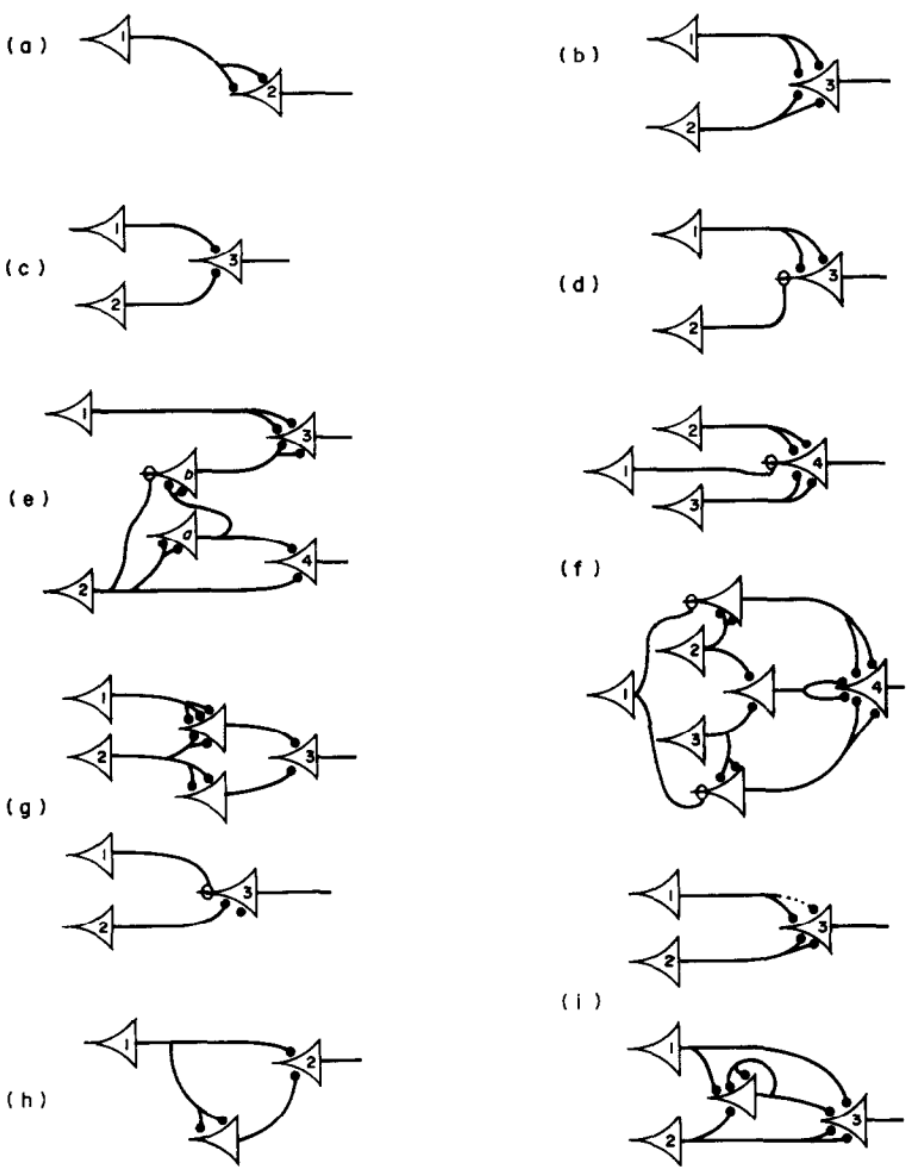
Progress in AI

McCullough & Pitts (1943)

A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY*

■ WARREN S. MCCULLOCH AND WALTER PITTS
University of Illinois, College of Medicine,
Department of Psychiatry at the Illinois Neuropsychiatric Institute,
University of Chicago, Chicago, U.S.A.

Because of the "all-or-none" character of nervous activity, neural events and the relations among them can be treated by means of propositional logic. It is found that the behavior of every net can be described in these terms, with the addition of more complicated logical means for nets containing circles; and that for any logical expression satisfying certain conditions, one can find a net behaving in the fashion it describes. It is shown that many particular choices among possible neurophysiological assumptions are equivalent, in the sense that for every net behaving under one assumption, there exists another net which behaves under the other and gives the same results, although perhaps not in the same time. Various applications of the calculus are discussed.



Progress in AI

- How old are neural networks?
  - They’ve been around since 1940s
  - But why only recently we saw major breakthroughs?

Progress in AI

- How old are neural networks?
  - They’ve been around since 1940s
  - But why only recently we saw major breakthroughs?

Current state of AI

- But what allowed such rapid progress recently?
  - Paradigm shift to self-supervised learning

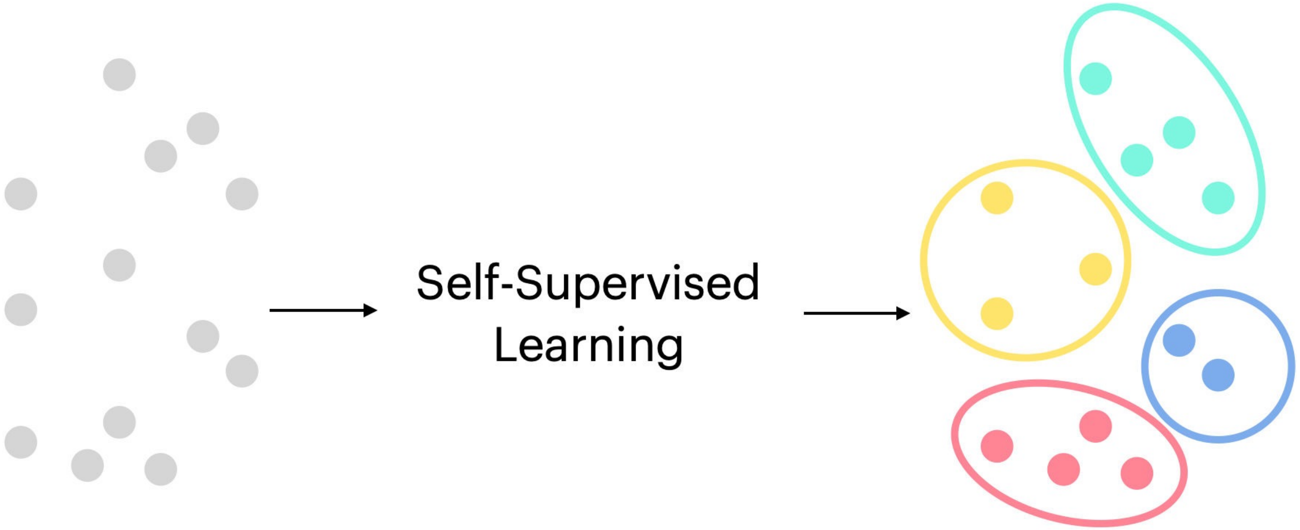
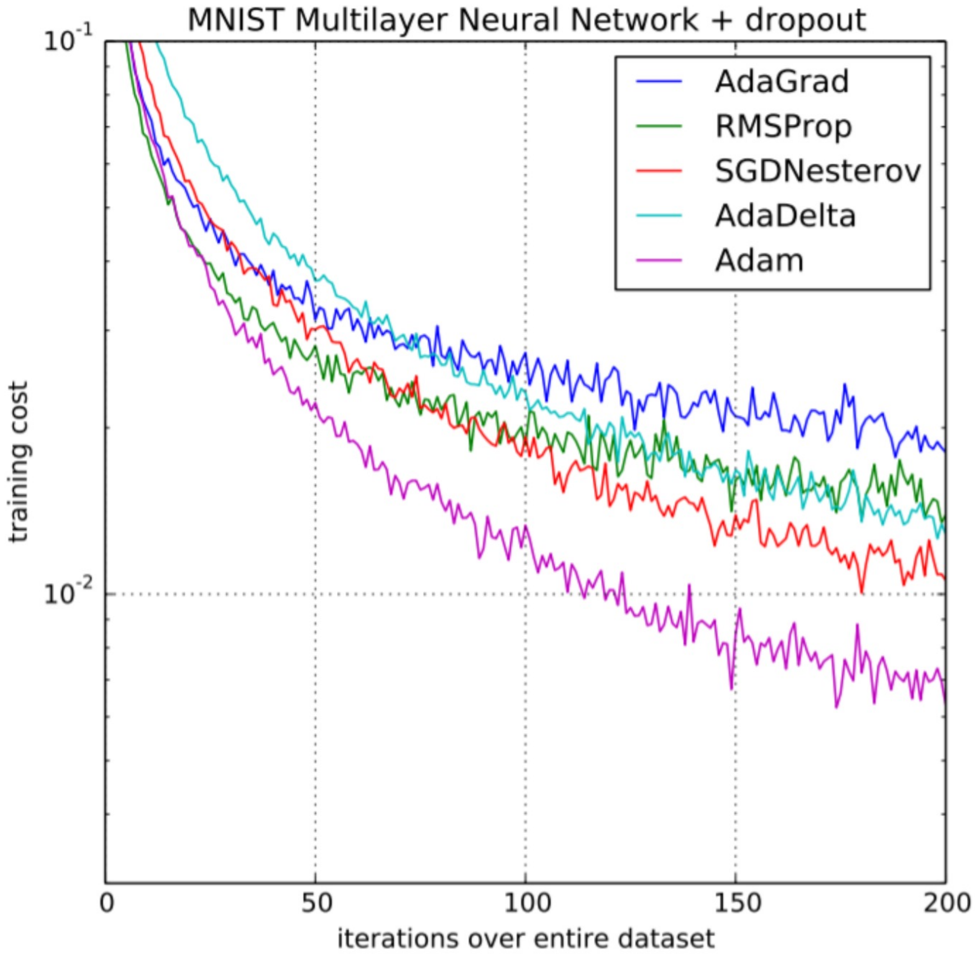


Image source: http://multicomp.cs.cmu.edu/research/self-supervised-learning/

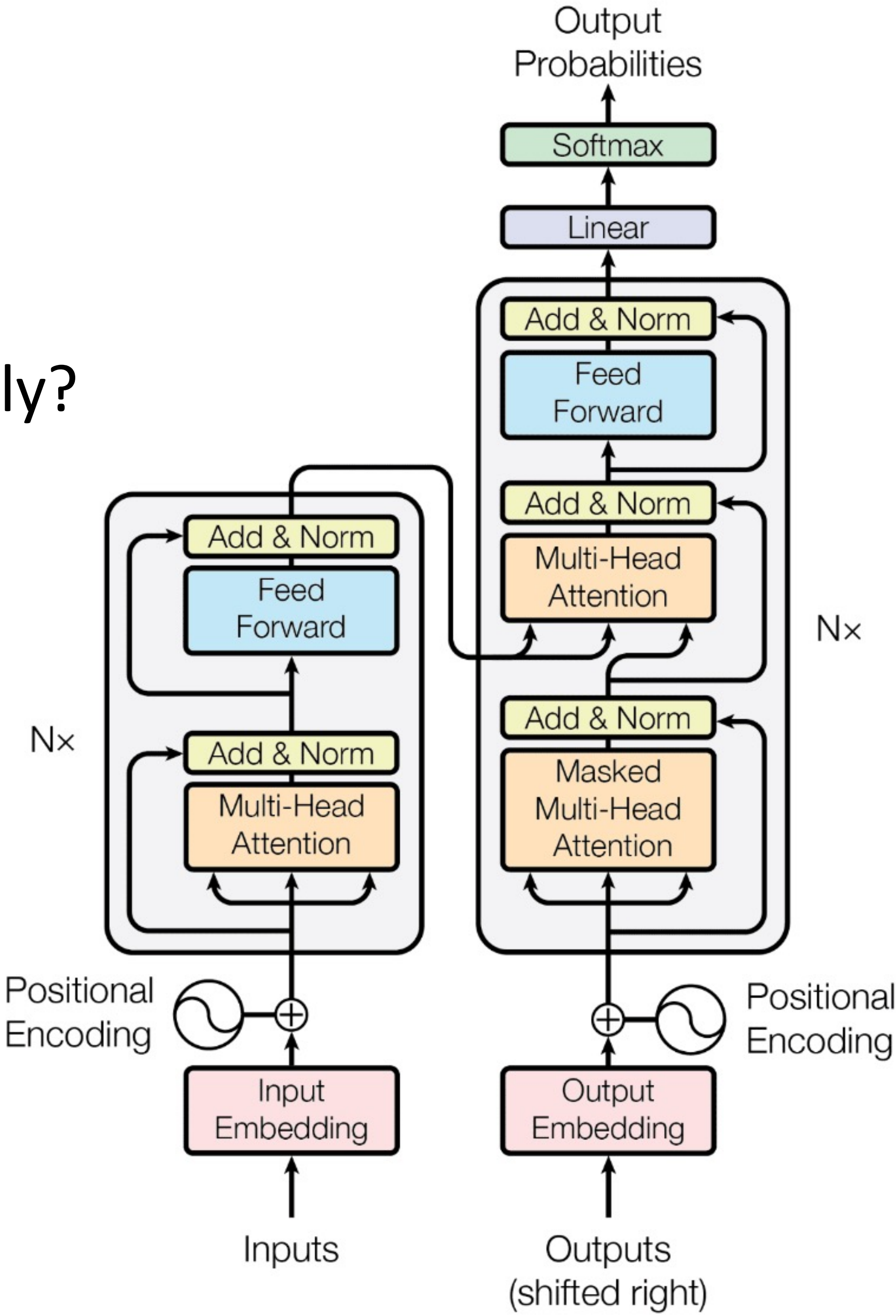
Current state of AI

- But what allowed such rapid progress recently?
    - Paradigm shift to self-supervised learning
    - Advances in optimization



Current state of AI

- But what allowed such rapid progress recently?
    - Paradigm shift to self-supervised learning
    - Advances in optimization
    - Innovations in model architectures



Current state of AI

- But what allowed such rapid progress recently?
  - Paradigm shift to self-supervised learning
  - Advances in optimization
  - Innovations in model architectures
  - Large scale training data
  - Large models

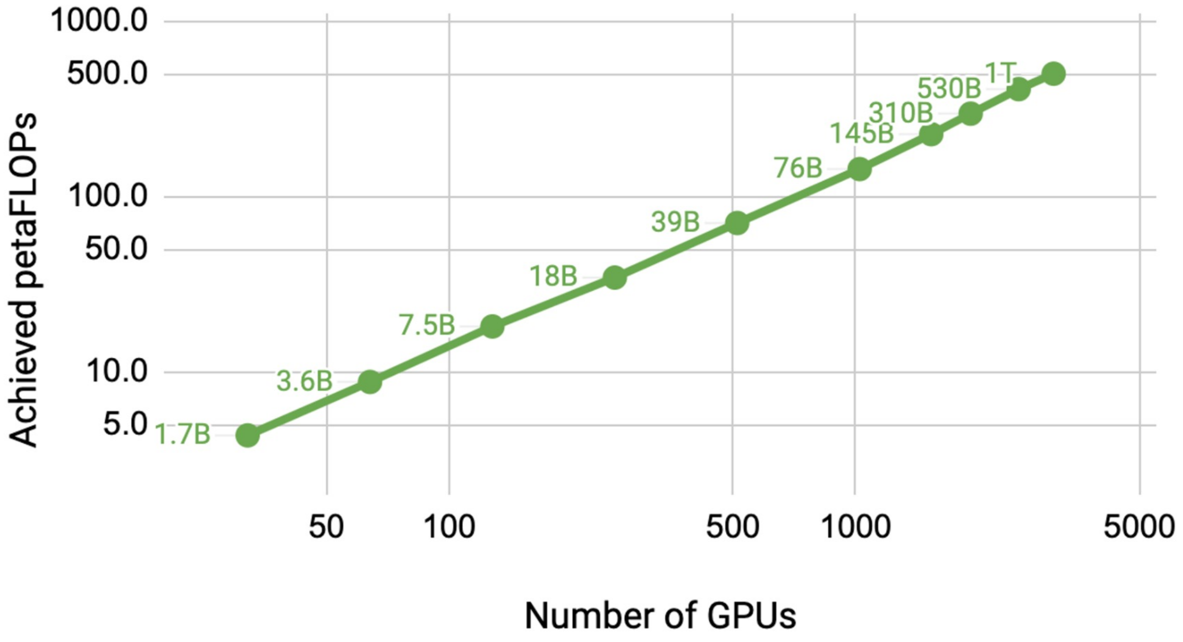


Image source: https://developer.nvidia.com/blog/scaling-language-model-training-to-a-trillion-parameters-using-megatron/

Current state of AI

- But what allowed such rapid progress recently?
  - Paradigm shift to self-supervised learning
  - Advances in optimization
  - Innovations in model architectures
  - Large scale training data
  - Large models
  - Advances in hardware accelerator and DL systems

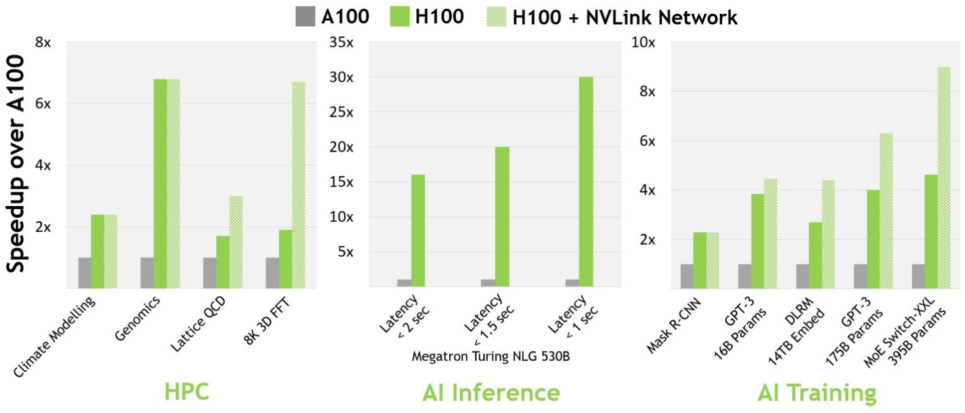


Image source: https://developer.nvidia.com/blog/nvidia-hopper-architecture-in-depth/

Current state of AI

- Almost every AI model is based on Neural networks
- Performance is consistently improving with scale
  - More training data
  - Larger models (number of neural network parameters)

- Question: How many parameters does a fully connected 12 layer neural network has (with hidden size of 768)?

Current state of AI

- Almost every AI model is based on Neural networks
- Performance is consistently improving with scale
  - More training data
  - Larger models (number of neural network parameters)

- Question: How many parameters does a fully connected 12 layer neural network has (with hidden size of 768)?
  - Each layer: 768 x 768 (weights) + 768 (biases)
  - Total: 12 x (768 x 768 + 768) = **7.1M parameters**

Current state of AI

State-of-the-art models are hundreds of billions of parameters

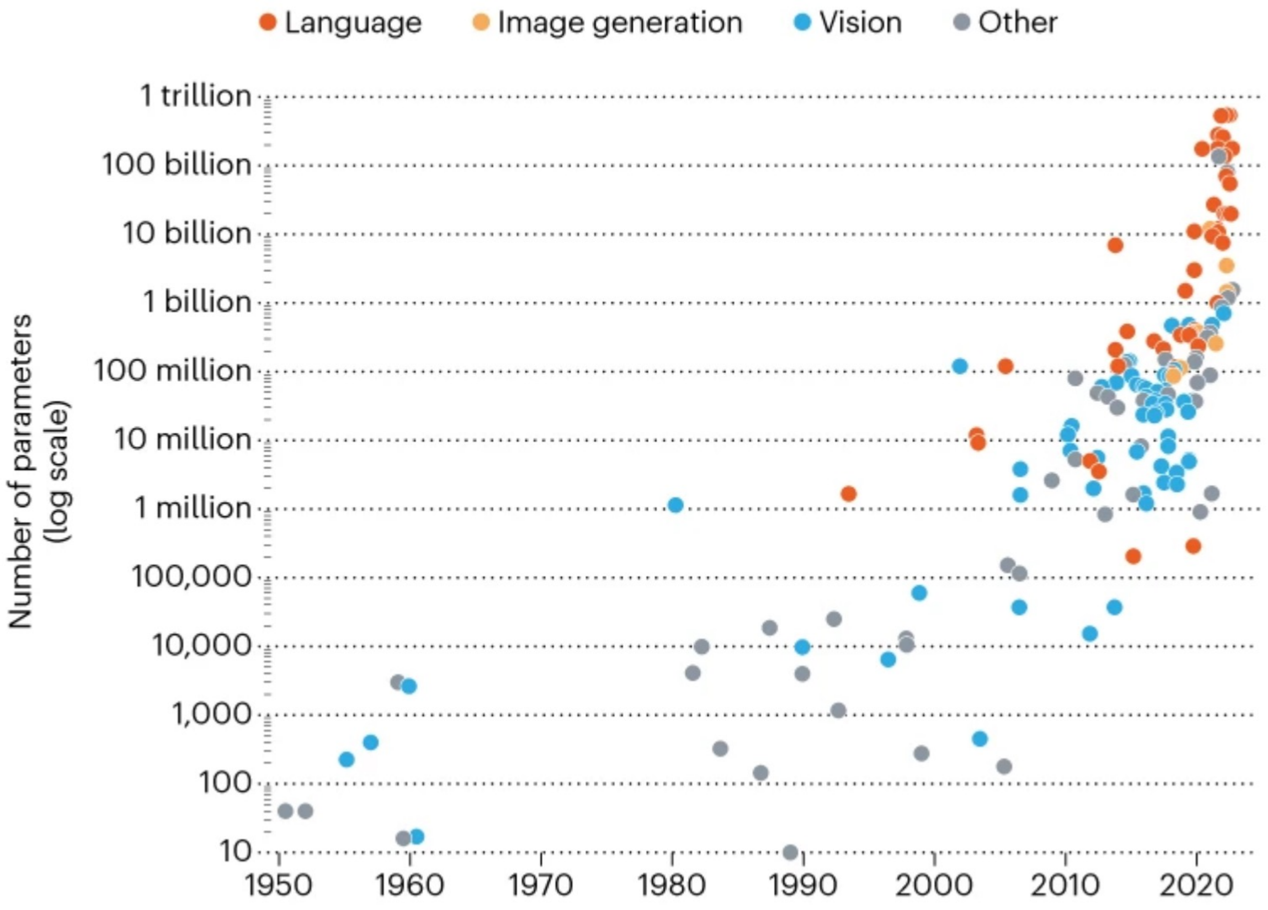


Image from: In AI, is bigger always better? https://www.nature.com/articles/d41586-023-00641-w

Current state of AI

State-of-the-art models are hundreds of billions of parameters

Trained on vast amounts of data (Trillions of tokens)

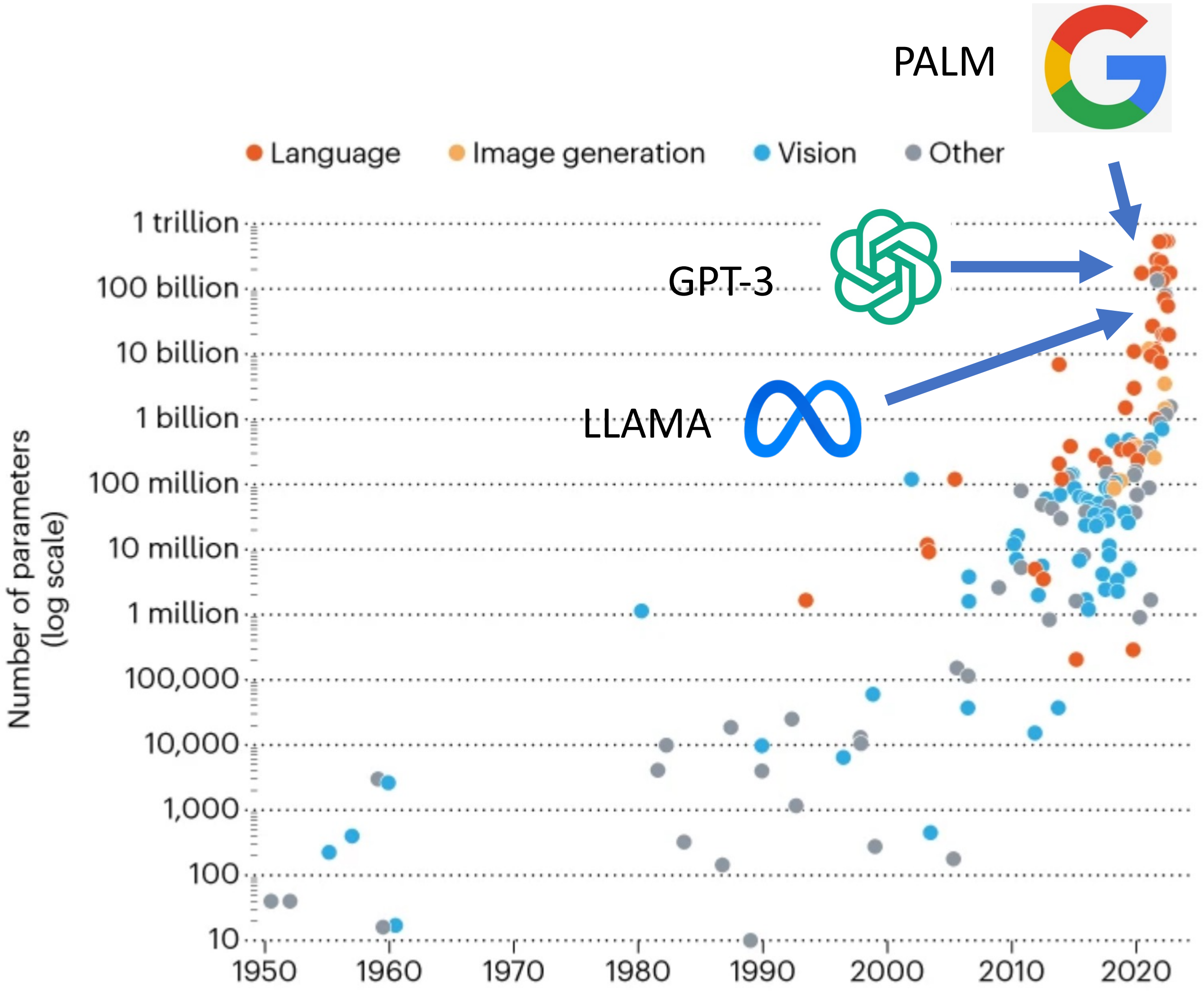


Image from: In AI, is bigger always better? https://www.nature.com/articles/d41586-023-00641-w

What is a foundation model?

On the terminology

Foundation models

- “Foundation models” is an informal label for a large family of models
  - “Foundation” models that can be used as part of modern AI systems
- Alternative names (none are perfect)
  - LLMs
  - Self-supervised models
  - Generative AI models
  - pretrained models

On the Opportunities and Risks of Foundation Models

Rishi Bommasani* Drew A. Hudson Ehsan Adeli Russ Altman Simran Arora Sydney von Arx Michael S. Bernstein Jeannette Bohg Antoine Bosselut Emma Brunskill Erik Brynjolfsson Shyamal Buch Dallas Card Rodrigo Castellon Niladri Chatterji Annie Chen Kathleen Creel Jared Quincy Davis Dorottya Demszky Chris Donahue Moussa Doumbouya Esin Durmus Stefano Ermon John Etchemendy Kawin Ethayarajh Li Fei-Fei Chelsea Finn Trevor Gale Lauren Gillespie Karan Goel Noah Goodman Shelby Grossman Neel Guha Tatsunori Hashimoto Peter Henderson John Hewitt Daniel E. Ho Jenny Hong Kyle Hsu Jing Huang Thomas Icard Saahil Jain Dan Jurafsky Pratyusha Kalluri Siddharth Karamcheti Geoff Keeling Fereshte Khani Omar Khattab Pang Wei Koh Mark Krass Ranjay Krishna Rohith Kuditipudi Ananya Kumar Faisal Ladhak Mina Lee Tony Lee Jure Leskovec Isabelle Levent Xiang Lisa Li Xuechen Li Tengyu Ma Ali Malik Christopher D. Manning Suvir Mirchandani Eric Mitchell Zanele Munyikwa Suraj Nair Avanika Narayan Deepak Narayanan Ben Newman Allen Nie Juan Carlos Niebles Hamed Nilforoshan Julian Nyarko Giray Ogut Laurel Orr Isabel Papadimitriou Joon Sung Park Chris Piech Eva Portelance Christopher Potts Aditi Raghunathan Rob Reich Hongyu Ren Frieda Rong Yusuf Roohani Camilo Ruiz Jack Ryan Christopher Ré Dorsa Sadigh Shiori Sagawa Keshav Santhanam Andy Shih Krishnan Srinivasan Alex Tamkin Rohan Taori Armin W. Thomas Florian Tramèr Rose E. Wang William Wang Bohan Wu Jiajun Wu Yuhuai Wu Sang Michael Xie Michihiro Yasunaga Jiaxuan You Matei Zaharia Michael Zhang Tianyi Zhang Xikun Zhang Yuhui Zhang Lucia Zheng Kaitlyn Zhou Percy Liang*[1]

Center for Research on Foundation Models (CRFM)
Stanford Institute for Human-Centered Artificial Intelligence (HAI)
Stanford University

AI is undergoing a paradigm shift with the rise of models (e.g., BERT, DALL-E, GPT-3) trained on broad data (generally using self-supervision at scale) that can be adapted to a wide range of downstream tasks. We call these models foundation models *to underscore their critically central yet incomplete character. This report provides a thorough account of the opportunities and risks of foundation models, ranging from their capabilities (e.g., language, vision, robotic manipulation, reasoning, human interaction) and technical principles (e.g., model architectures, training procedures, data, systems, security, evaluation, theory) to their applications (e.g., law, healthcare, education) and societal impact (e.g., inequity, misuse, economic and environmental impact, legal and ethical considerations). Though foundation models are based on standard deep learning and transfer learning, their scale results in new emergent capabilities, and their effectiveness across so many tasks incentivizes homogenization. Homogenization provides powerful leverage but demands caution, as the defects of the foundation model are inherited by all the adapted models downstream. Despite the impending widespread deployment of foundation models, we currently lack a clear understanding of how they work, when they fail, and what they are even capable of due to their emergent properties. To tackle these questions, we believe much of the critical research on foundation models will require deep interdisciplinary collaboration commensurate with their fundamentally sociotechnical nature.*

[1]Corresponding author: pliang@cs.stanford.edu

*Equal contribution.

Foundation models

- **The term "Foundation models" Doesn't mean that these models are foundation of AI!!**
- Means majority of AI systems build on top of these models
- These models are general models designed for a wide range of capabilities
- The terminology shows the significance of architectural stability, safety, and security: poorly-constructed foundations are a recipe for disaster

On the Opportunities and Risks of Foundation Models

arXiv:2108.07258v3 [cs.LG] 12 Jul 2022

Rishi Bommasani* Drew A. Hudson Ehsan Adeli Russ Altman Simran Arora Sydney von Arx Michael S. Bernstein Jeannette Bohg Antoine Bosselut Emma Brunskill Erik Brynjolfsson Shyamal Buch Dallas Card Rodrigo Castellon Niladri Chatterji Annie Chen Kathleen Creel Jared Quincy Davis Dorottya Demszky Chris Donahue Moussa Doumbouya Esin Durmus Stefano Ermon John Etchemendy Kawin Ethayarajh Li Fei-Fei Chelsea Finn Trevor Gale Lauren Gillespie Karan Goel Noah Goodman Shelby Grossman Neel Guha Tatsunori Hashimoto Peter Henderson John Hewitt Daniel E. Ho Jenny Hong Kyle Hsu Jing Huang Thomas Icard Saahil Jain Dan Jurafsky Pratyusha Kalluri Siddharth Karamcheti Geoff Keeling Fereshte Khani Omar Khattab Pang Wei Koh Mark Krass Ranjay Krishna Rohith Kuditipudi Ananya Kumar Faisal Ladhak Mina Lee Tony Lee Jure Leskovec Isabelle Levent Xiang Lisa Li Xuechen Li Tengyu Ma Ali Malik Christopher D. Manning Suvir Mirchandani Eric Mitchell Zanele Munyikwa Suraj Nair Avanika Narayan Deepak Narayanan Ben Newman Allen Nie Juan Carlos Niebles Hamed Nilforoshan Julian Nyarko Giray Ogut Laurel Orr Isabel Papadimitriou Joon Sung Park Chris Piech Eva Portelance Christopher Potts Aditi Raghunathan Rob Reich Hongyu Ren Frieda Rong Yusuf Roohani Camilo Ruiz Jack Ryan Christopher Ré Dorsa Sadigh Shiori Sagawa Keshav Santhanam Andy Shih Krishnan Srinivasan Alex Tamkin Rohan Taori Armin W. Thomas Florian Tramèr Rose E. Wang William Wang Bohan Wu Jiajun Wu Yuhuai Wu Sang Michael Xie Michihiro Yasunaga Jiaxuan You Matei Zaharia Michael Zhang Tianyi Zhang Xikun Zhang Yuhui Zhang Lucia Zheng Kaitlyn Zhou Percy Liang*[1]

Center for Research on Foundation Models (CRFM)
Stanford Institute for Human-Centered Artificial Intelligence (HAI)
Stanford University

AI is undergoing a paradigm shift with the rise of models (e.g., BERT, DALL-E, GPT-3) trained on broad data (generally using self-supervision at scale) that can be adapted to a wide range of downstream tasks. We call these models foundation models *to underscore their critically central yet incomplete character. This report provides a thorough account of the opportunities and risks of foundation models, ranging from their capabilities (e.g., language, vision, robotic manipulation, reasoning, human interaction) and technical principles (e.g., model architectures, training procedures, data, systems, security, evaluation, theory) to their applications (e.g., law, healthcare, education) and societal impact (e.g., inequity, misuse, economic and environmental impact, legal and ethical considerations). Though foundation models are based on standard deep learning and transfer learning, their scale results in new emergent capabilities, and their effectiveness across so many tasks incentivizes homogenization. Homogenization provides powerful leverage but demands caution, as the defects of the foundation model are inherited by all the adapted models downstream. Despite the impending widespread deployment of foundation models, we currently lack a clear understanding of how they work, when they fail, and what they are even capable of due to their emergent properties. To tackle these questions, we believe much of the critical research on foundation models will require deep interdisciplinary collaboration commensurate with their fundamentally sociotechnical nature.*

[1]Corresponding author: pliang@cs.stanford.edu *Equal contribution.

More discussion on naming (Bommasani et al., 2021) https://arxiv.org/pdf/2108.07258.pdf

Foundation models

However, this terminology is not perfect and many researchers don’t like it!

Yi Tay @YiTayML

Frontier models vs foundation models.

"This time, at least the name is coined by groups that actually have a solid track record building such models, so I don't need to be grumpy right away :)"

Solid burn 🤭 💯

> **Lucas Beyer** @giffmana · Jul 26
>
> Stop everything and change all your slides from "foundation model" to "frontier model".
>
> Curious what comes out of this!
>
> This time, at least the name is coined by groups that actually have a solid track record building such models, so I don't need to be grumpy right away :)
> twitter.com/ShaneLegg/stat...

Foundation models

However, this terminology is not perfect and many researchers don't like it!

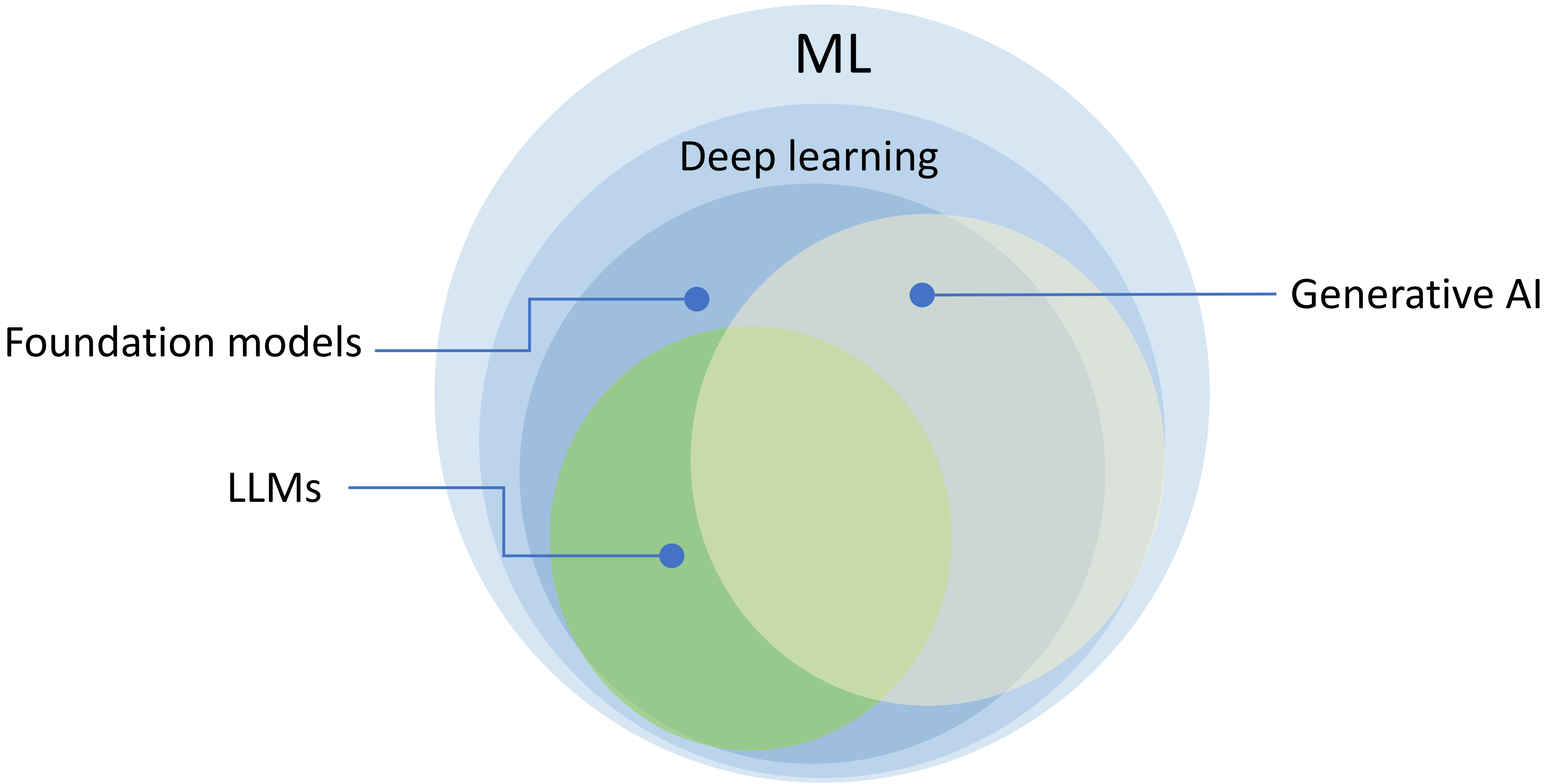
Yi Tay
@YiTayML

Frontier models vs foundation models.

"This time, at least the name is coined by groups that actually have a solid track record building such models, so I don't need to be grumpy right away :)"

Solid burn 🤭 💯

> **Lucas Beyer** @giffmana · Jul 26
> Stop everything and change all your slides from "foundation model" to "frontier model".
>
> Curious what comes out of this!
>
> This time, at least the name is coined by groups that actually have a solid track record building such models, so I don't need to be grumpy right away :)
> twitter.com/ShaneLegg/stat...


ML
Deep learning
Generative AI
Foundation models
LLMs

Modern AI: SCALE solves lots of problems!!

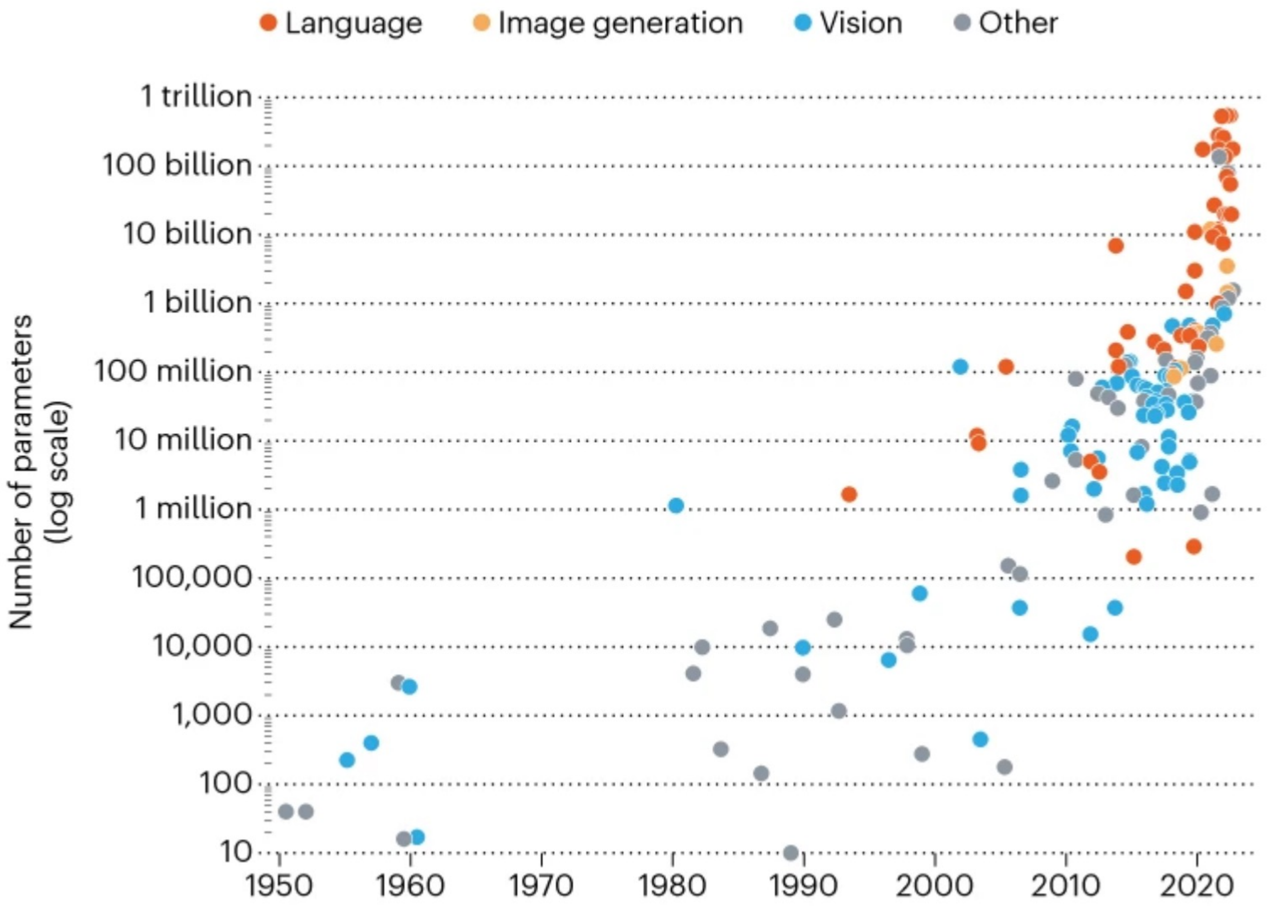
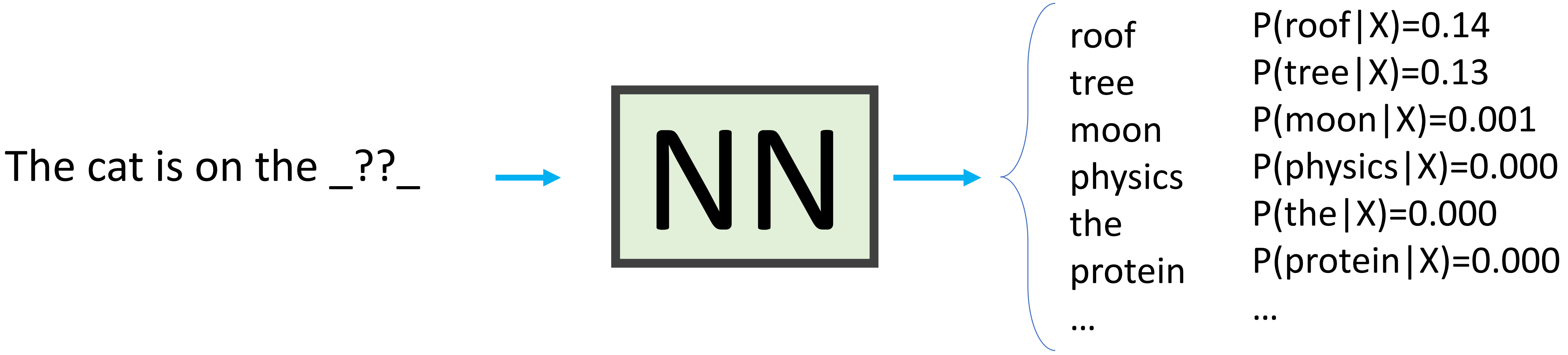


Image from: In AI, is bigger always better? https://www.nature.com/articles/d41586-023-00641-w

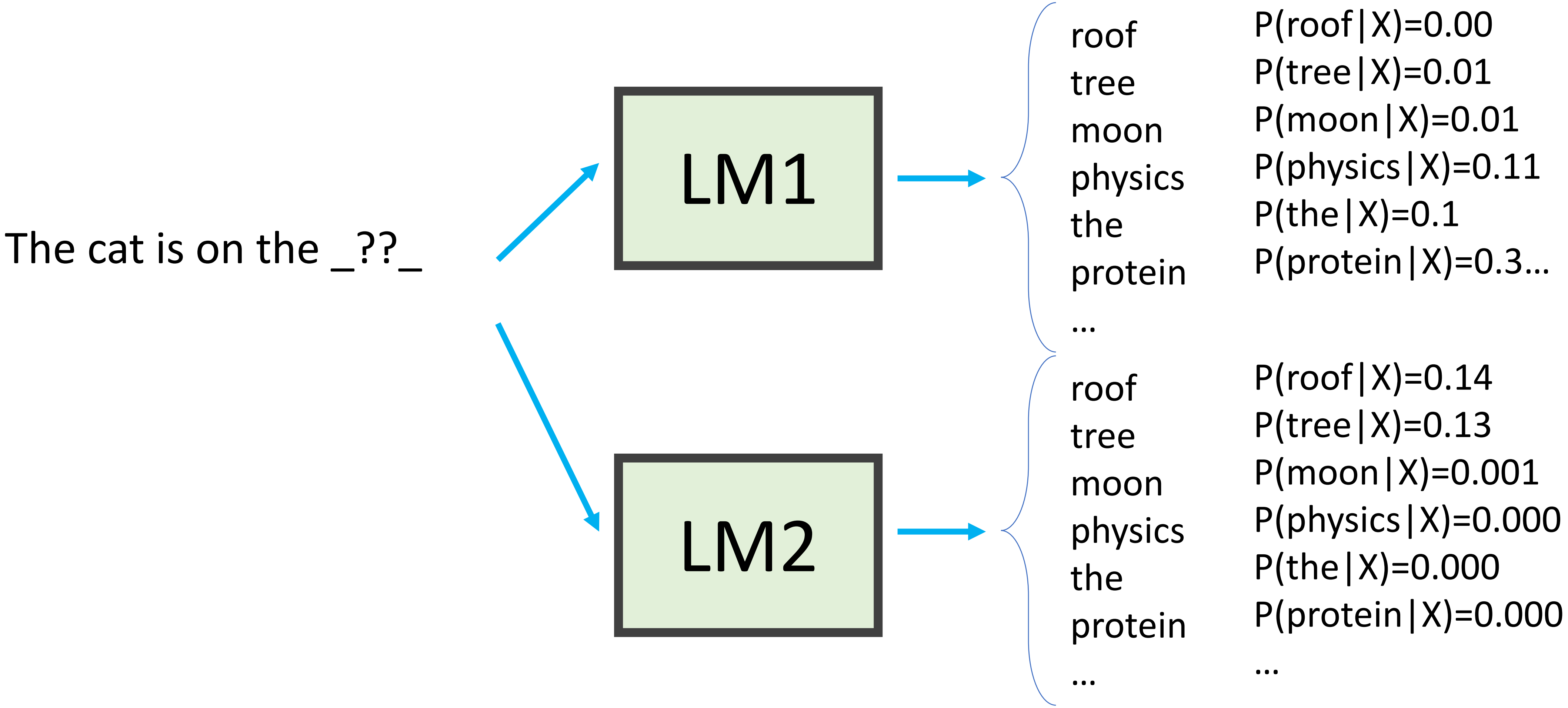
Large Language Models

A language model can predict the next word based on the given context



Large Language Models

A language model can predict the next word based on the given context



Large Language Models

A language model can predict the next word based on the given context

The cat is on the _??_

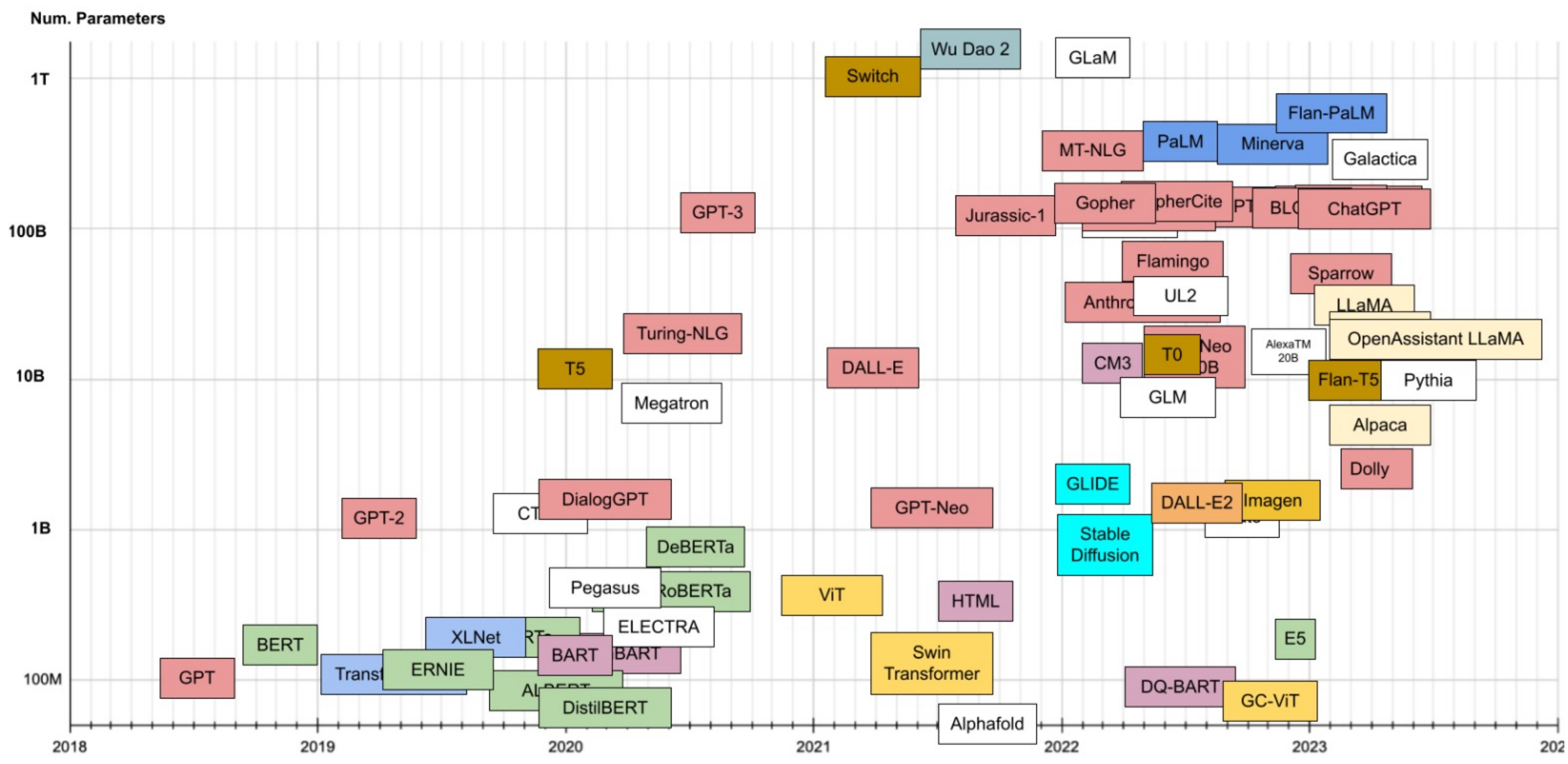
Which LM is better?

LM1

roof	P(roof|X)=0.00
tree	P(tree|X)=0.01
moon	P(moon|X)=0.01
physics	P(physics|X)=0.11
the	P(the|X)=0.1
protein	P(protein|X)=0.3…
…	

LM2

roof	P(roof|X)=0.14
tree	P(tree|X)=0.13
moon	P(moon|X)=0.001
physics	P(physics|X)=0.000
the	P(the|X)=0.000
protein	P(protein|X)=0.000
…	…



[Fig from: Transformer models: an introduction and catalog (2023)]()

Rank the LMs!

PollEv.com/cohan

A

Large language models are a great way to learn a new language.

The real reason that the real state market has seen such a decline is because the bankers who created the market have been removed from both the government and the financial system.

B

Large language models are effective at capturing much of the syntactic and semantic

The real reason that the real state market has seen such a decline is due to the fact that the coronavirus pandemic has had a large impact on the economy.

C

Large language models are important for natural language processing and machine translation, as

The real reason that the real state market has seen such a decline is because of the lack of demand. Interest rates have been at historic lows for an extended period of time.

D

Large language models are also popular in natural language processing. These models are

The real reason that the real state market has seen such a decline is because of the Fed's low Interest Rate Policy. low interest rate policy has created a financial

Rank the LMs!

PollEv.com/cohan

Large language models are a great way to learn a new language.

The real reason that the real state market has seen such a decline is because the bankers who created the market have been removed from both the government and the financial system.

A

Large language models are effective at capturing much of the syntactic and semantic

The real reason that the real state market has seen such a decline is due to the fact that the coronavirus pandemic has had a large impact on the economy.

B

B > C > D > A

Large language models are important for natural language processing and machine translation, as

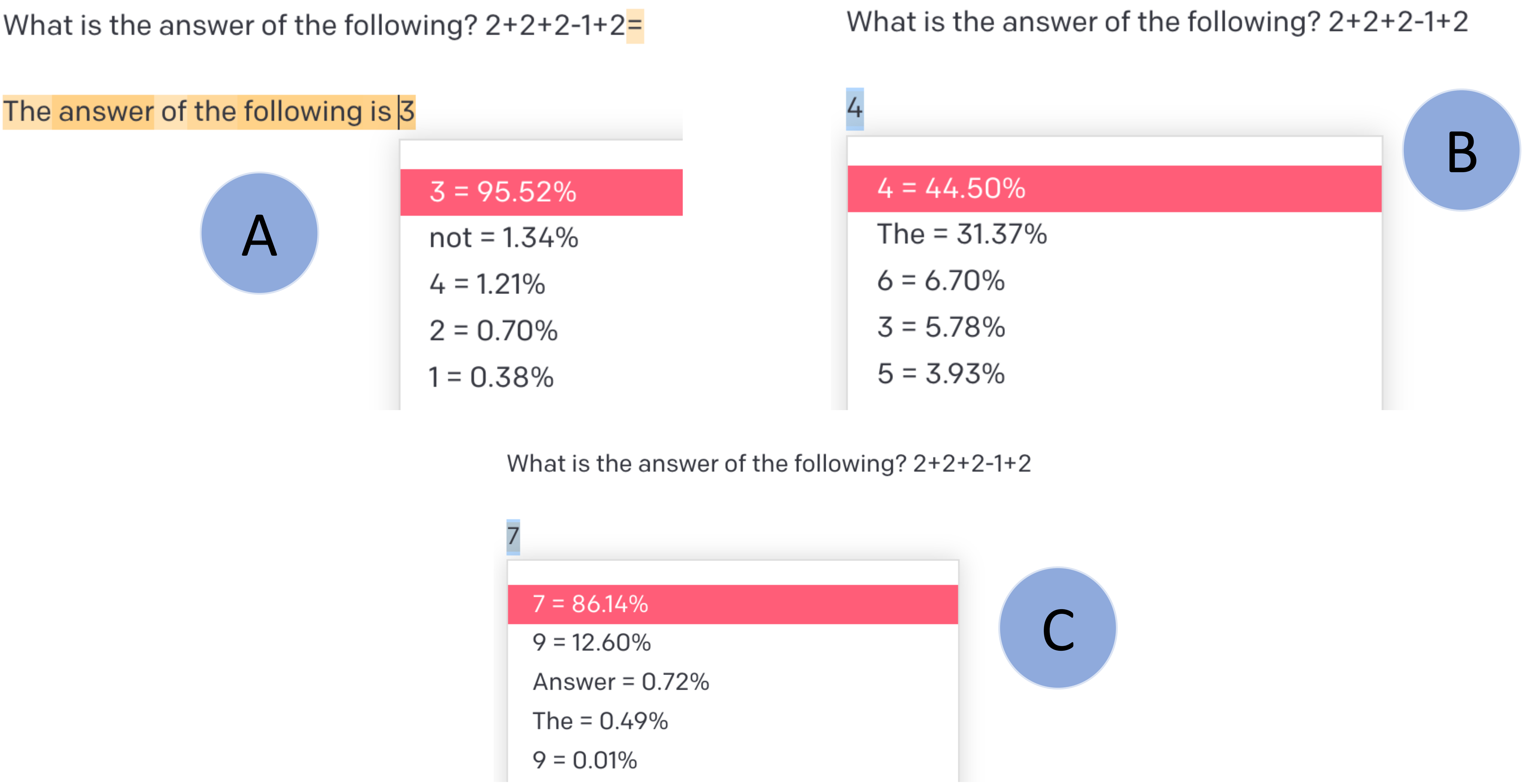
The real reason that the real state market has seen such a decline is because of the lack of demand. Interest rates have been at historic lows for an extended period of time.

C

Large language models are also popular in natural language processing. These models are

The real reason that the real state market has seen such a decline is because of the Fed's low Interest Rate Policy. low interest rate policy has created a financial

D


What is the answer of the following? 2+2+2-1+2=
The answer of the following is 3
3 = 95.52%
not = 1.34%
4 = 1.21%
2 = 0.70%
1 = 0.38%
A
What is the answer of the following? 2+2+2-1+2
4
4 = 44.50%
The = 31.37%
6 = 6.70%
3 = 5.78%
5 = 3.93%
B
What is the answer of the following? 2+2+2-1+2
7
7 = 86.14%
9 = 12.60%
Answer = 0.72%
The = 0.49%
9 = 0.01%
C

Language model's scale

- The larger the scale of an LM:
    - it has better chance to learn diverse linguistic patterns
    - it has more capacity to capture more abstract concepts
    - new emerging capabilities

Questions so far?

Grading breakdown

- 30%: Assignments
  - Includes both written and coding problem sets
- 20%: Midterm, in person, closed book
- 10%: Participation and quizzes
- 40%: Final projects – open ended (more on this)
  - Project proposal (5%)
  - Progress report (5%)
  - Final report and presentation (20%)
  - Code and reproducibility checklist (10%)

Final project

- Group projects (team size = 2 to 3 students)

- What is the goal of the final project?
  - Getting familiar on conducting research on AI foundation models

Class project and timeline

September 19
Form teams

October 13
Project proposal

November 17
Progress report

Dec 18
Final report

Final project – teams

- The projects should be done in teams of 2-3 students
- We encourage you to form your own groups and tell us by 9/19
- But if you don’t submit your group, we will randomly assign you a group on 9/20
- You can also use slack to find teammates

Final project - proposal

- 1-2 page proposal about the topic of the project and execution plan
  - Brief description of the topic, relevant related work, the proposed approach, and experimental plan
- The project is open ended!
- We will discuss some possible directions and tips on choosing the project topics in the next few weeks
- Due October 13

Final projects – types of projects

- Several possibilities (more on this in future lectures)
  - Reimplementing and reproducing results from another research paper and adding some missing experiments – (undergrads only)
  - Identifying a shortcoming in a paper and proposing a small extension
  - Novel research idea
  - Applications to other domains, interdisciplinary projects, and novel user facing applications are encouraged
  - Possibility for writing a survey paper on a specific relevant topic
    (talk to us early if you want to work on survey papers)
- Guidelines
  - Do not choose projects that require a lot of compute!
  - Data collection projects are harder to finish on time
  - Graduate students need to have a more extensive literature review component in their report

Final projects - resources

- Yale HPC cluster
- Your own or your lab's compute resources
- We have 55 seats each with $50 GCP cloud credits.
    - In your proposal let us know how many seats you need (max 1 per person)
    - If demand is higher, we will randomly assign

Final project – final report

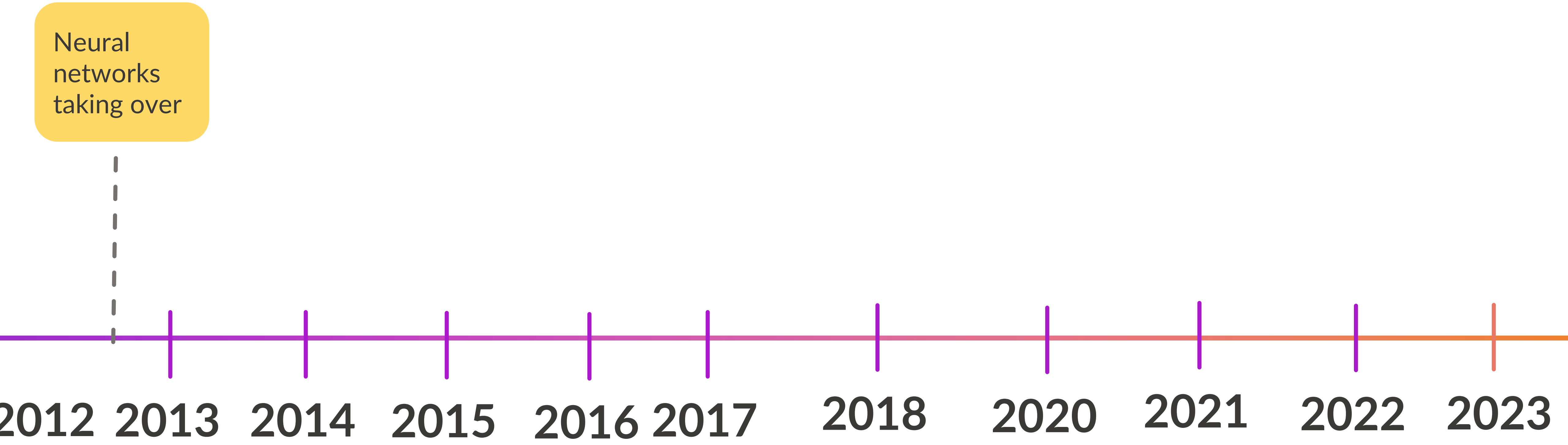
- 4-6 pages, ML conference format report
  - e.g., NeurIPS template (https://www.overleaf.com/latex/templates/neurips-2023/vstgtvjwgdng)
  - Max 6 pages. Page limit doesn’t include references and appendix
  - Due December 18
- Submit a short 3-minute pre recorded video presentation

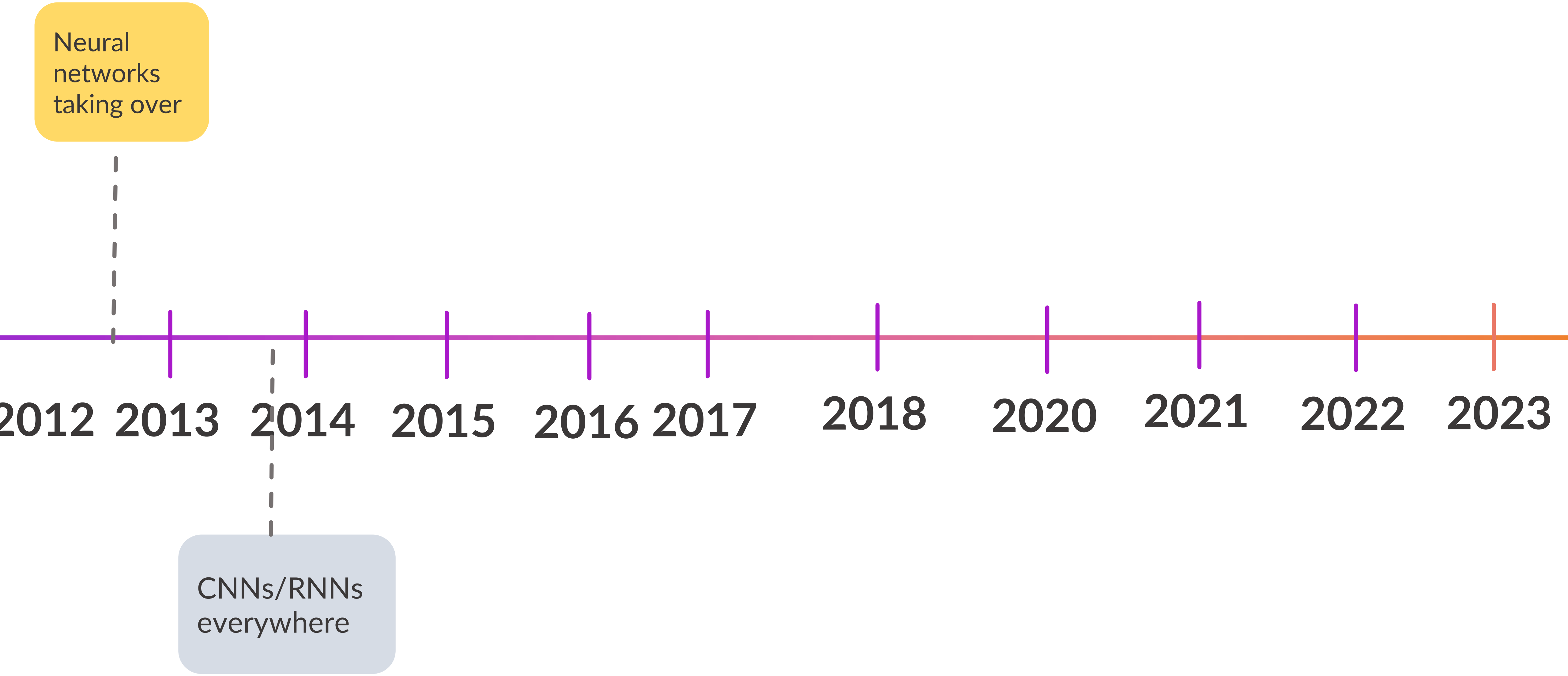
Final project – code and reproducibility

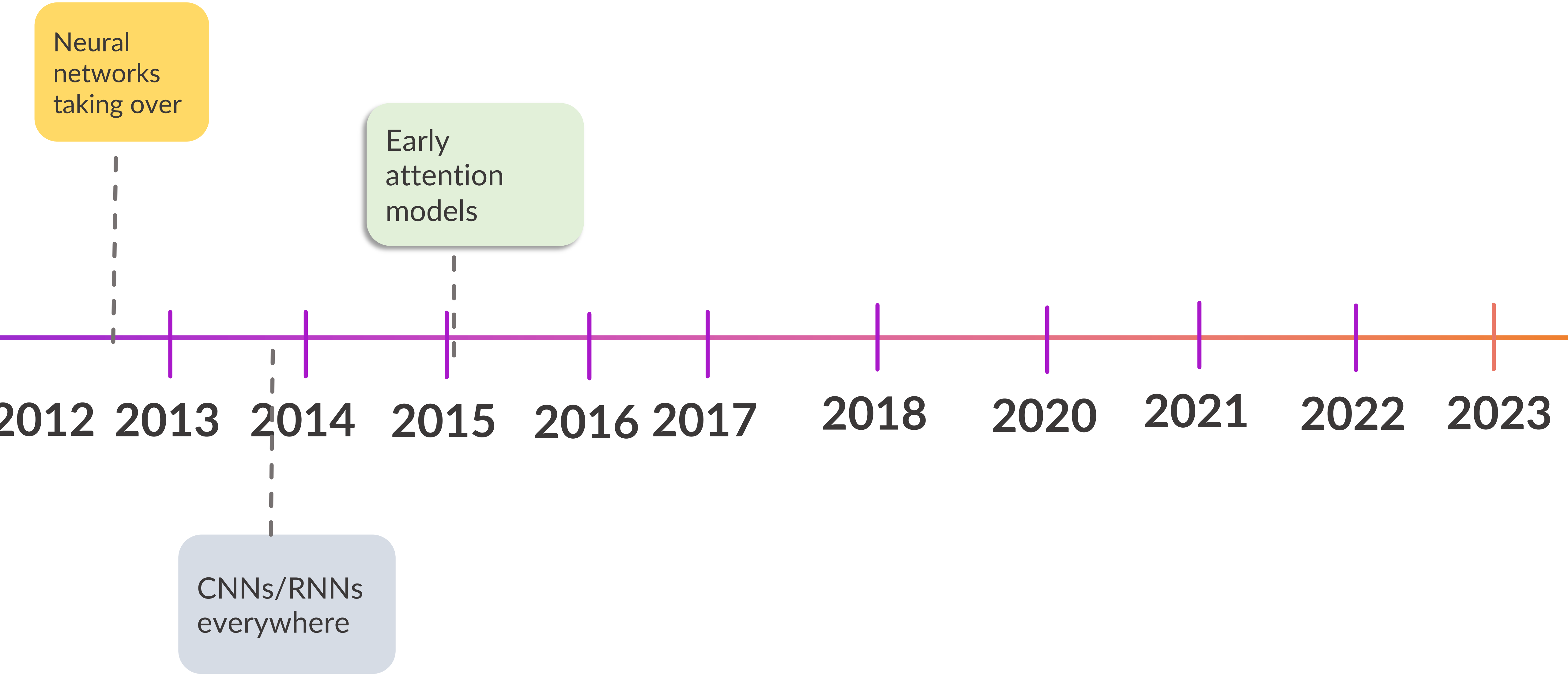
- Code should be uploaded to a github repo and linked in the report
- Please include a clear README on how to setup the environment, run the code, and reproduce the results
- We will distribute a reproducibility checklist that need to be submitted with your final report
- Also submit a contribution statement that clearly describes contributions of your team members

Questions?

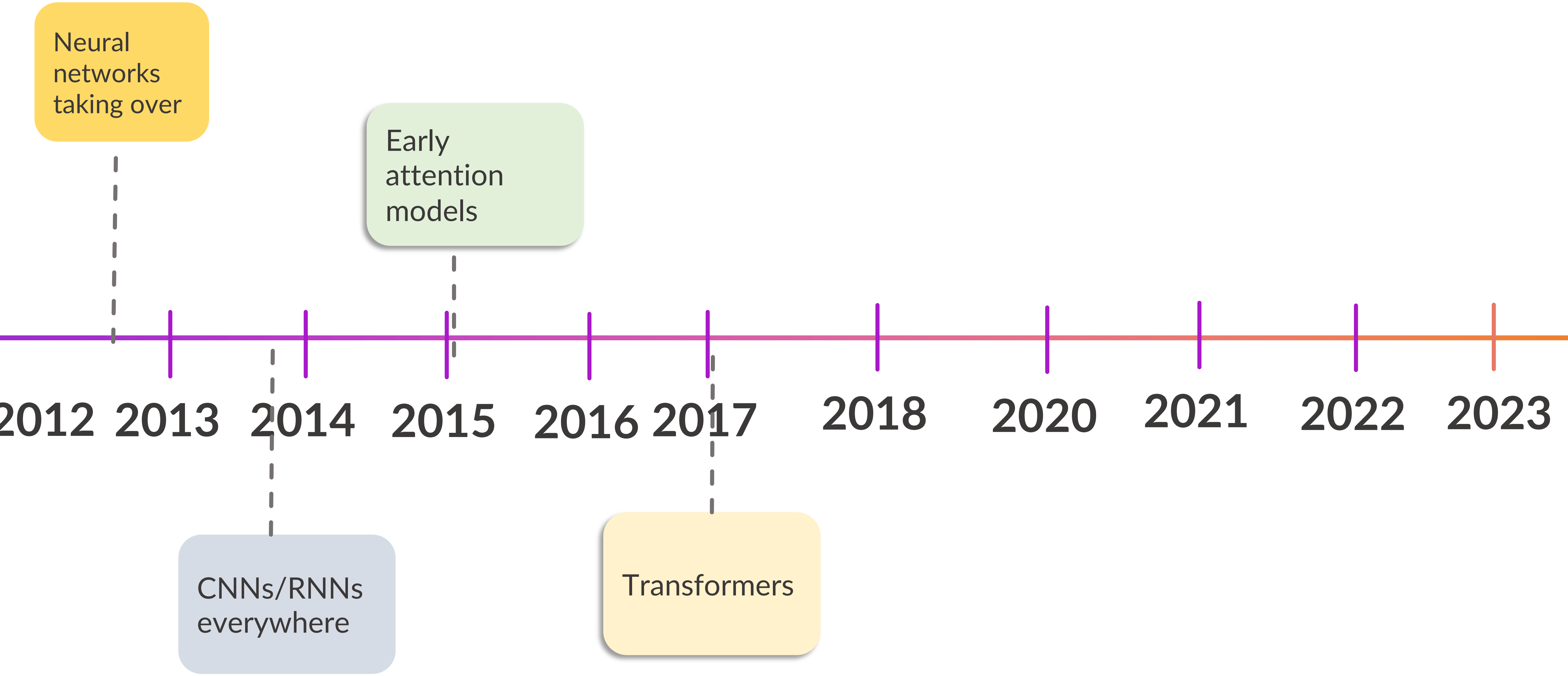
A brief timeline

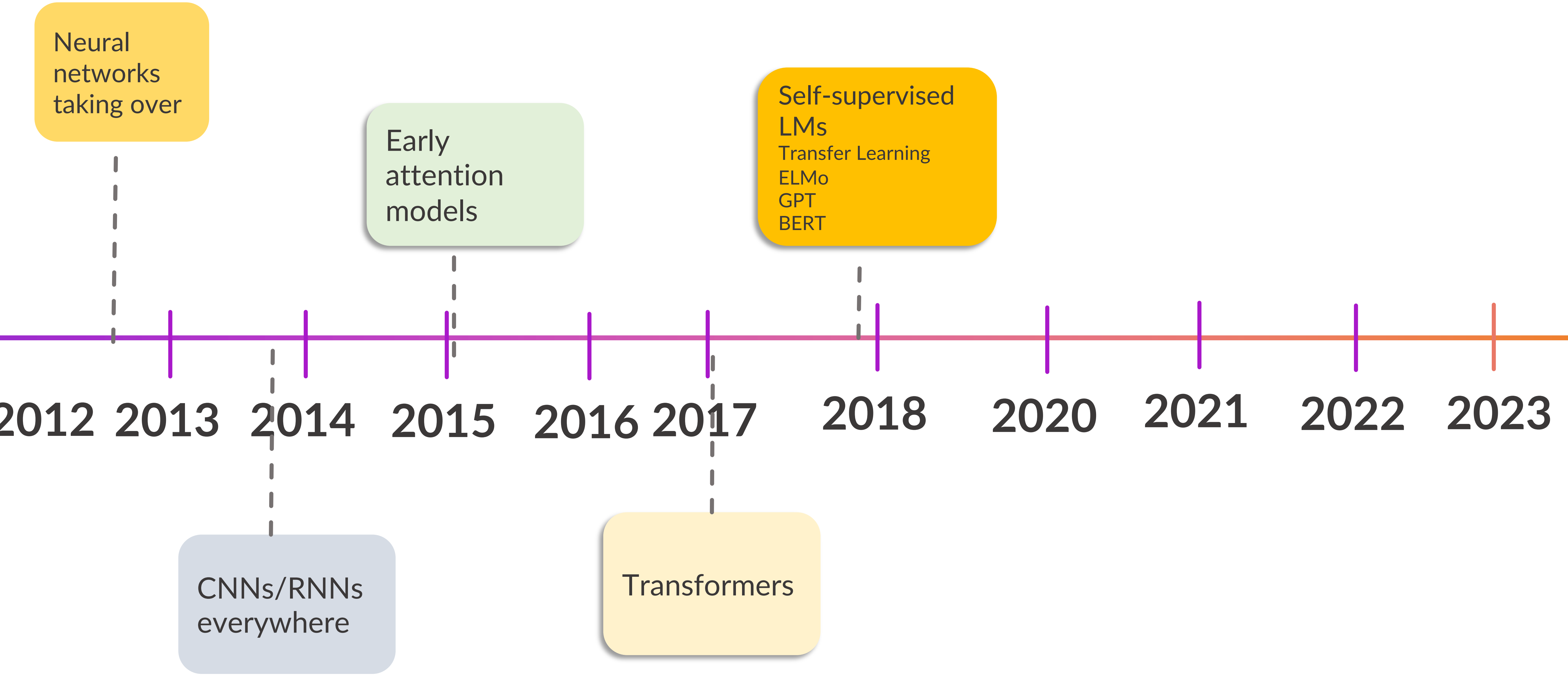

Neural networks taking over
2012
2013
2014
2015
2016
2017
2018
2020
2021
2022
2023

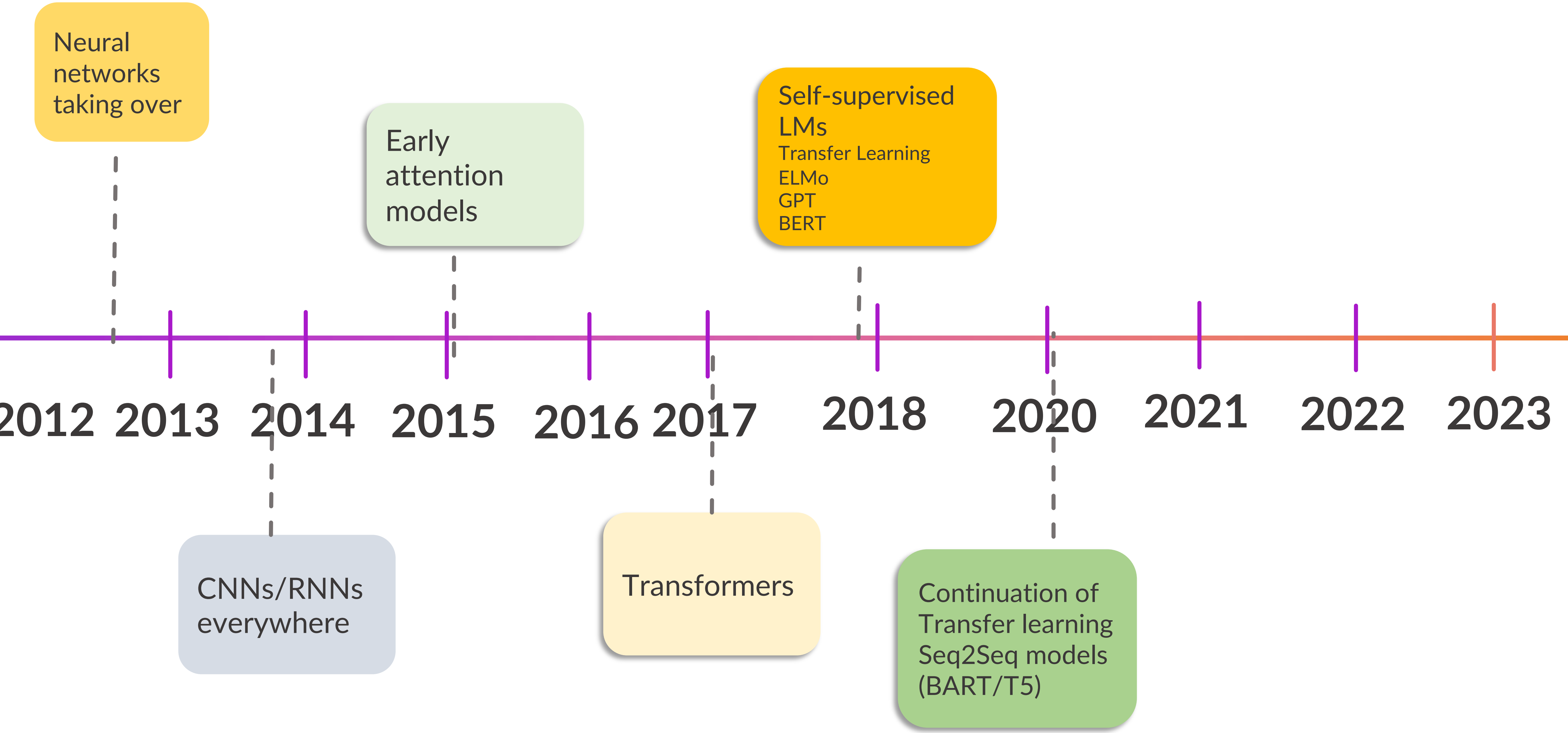

Neural networks taking over
CNNs/RNNs everywhere
2012
2013
2014
2015
2016
2017
2018
2020
2021
2022
2023

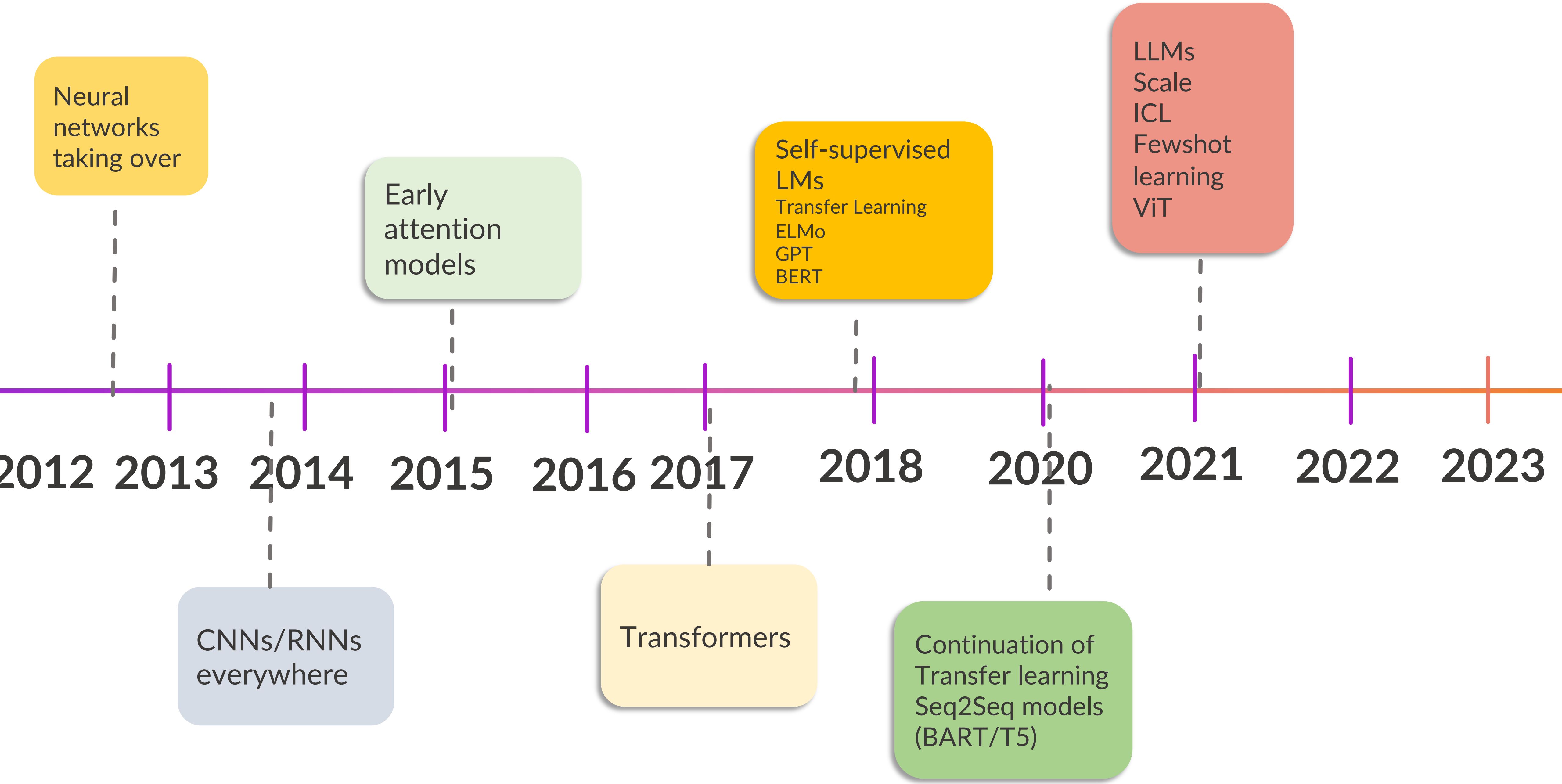


Neural networks taking over
Early attention models
2012
2013
2014
2015
2016
2017
2018
2020
2021
2022
2023
CNNs/RNNs everywhere

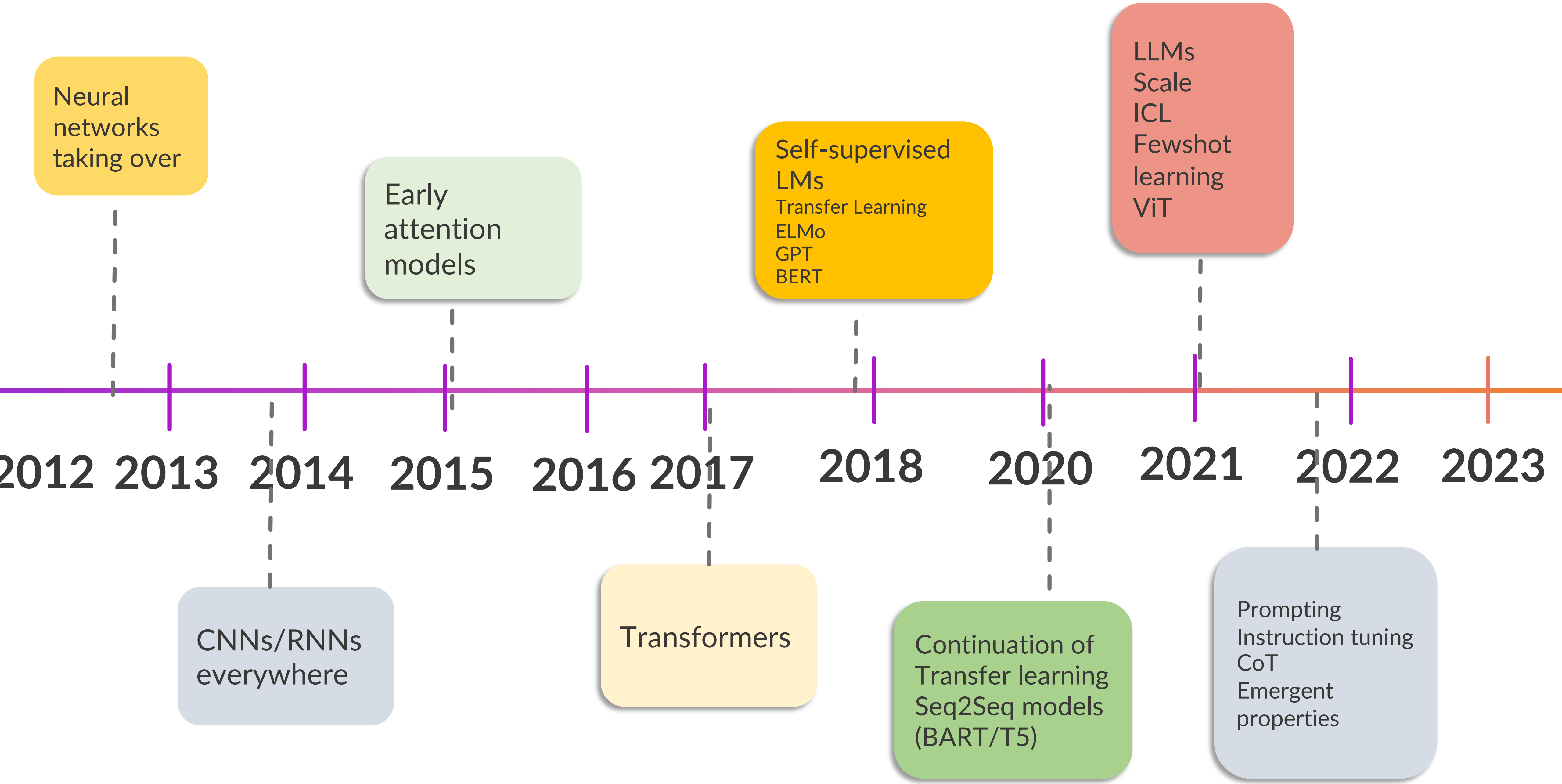

Neural networks taking over
Early attention models
2012
2013
2014
2015
2016
2017
2018
2020
2021
2022
2023
CNNs/RNNs everywhere
Transformers

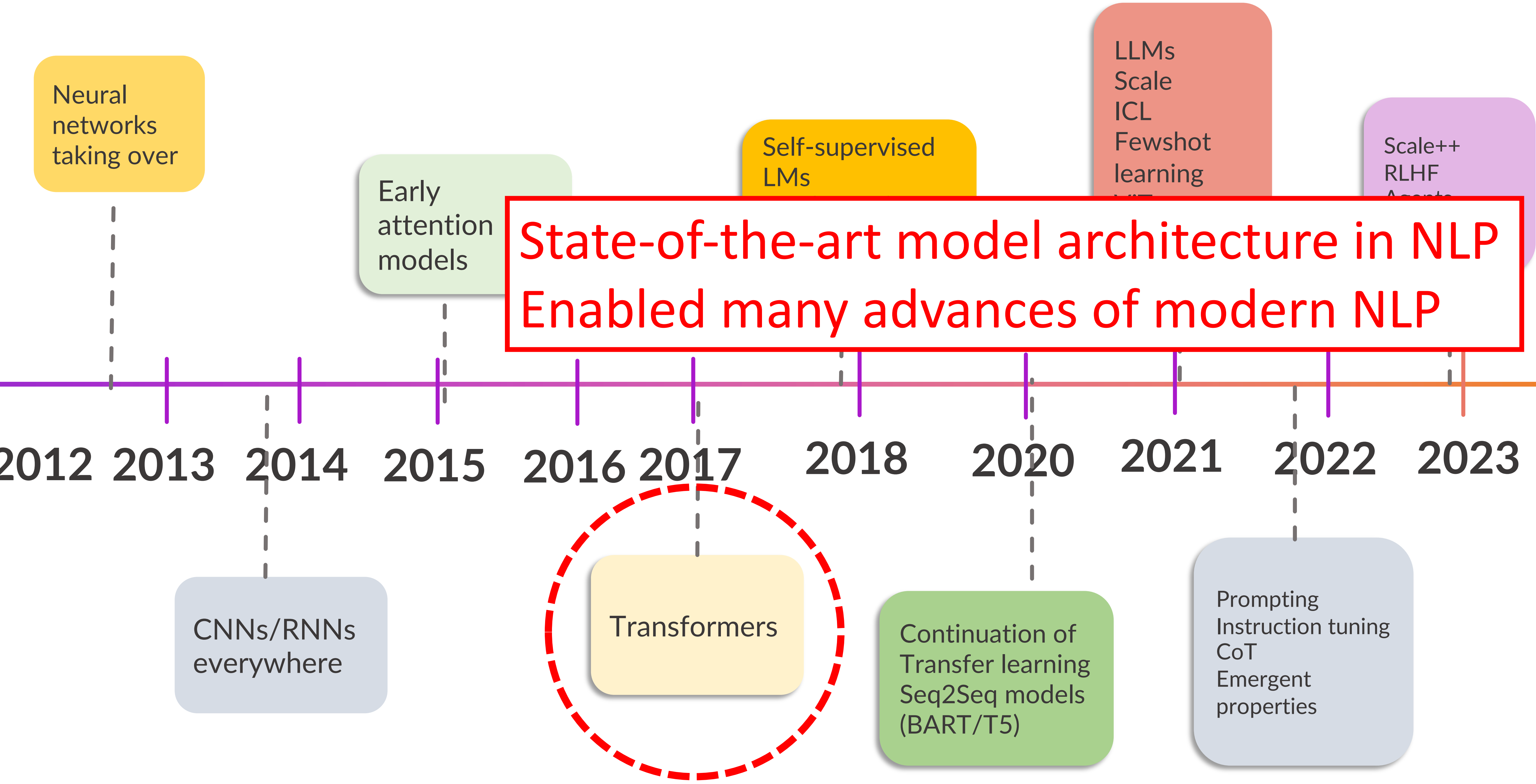

Neural networks taking over
Early attention models
Self-supervised LMs
Transfer Learning
ELMo
GPT
BERT
2012
2013
2014
2015
2016
2017
2018
2020
2021
2022
2023
CNNs/RNNs everywhere
Transformers


Neural networks taking over
Early attention models
Self-supervised LMs
Transfer Learning
ELMo
GPT
BERT
2012
2013
2014
2015
2016
2017
2018
2020
2021
2022
2023
CNNs/RNNs everywhere
Transformers
Continuation of Transfer learning
Seq2Seq models (BART/T5)



Neural networks taking over
Early attention models
Self-supervised LMs
Transfer Learning
ELMo
GPT
BERT
LLMs
Scale
ICL
Fewshot learning
ViT
2012
2013
2014
2015
2016
2017
2018
2020
2021
2022
2023
CNNs/RNNs everywhere
Transformers
Continuation of Transfer learning
Seq2Seq models (BART/T5)


Neural networks taking over
Early attention models
Self-supervised LMs
Transfer Learning
ELMo
GPT
BERT
LLMs
Scale
ICL
Fewshot learning
ViT
2012
2013
2014
2015
2016
2017
2018
2020
2021
2022
2023
CNNs/RNNs everywhere
Transformers
Continuation of Transfer learning
Seq2Seq models
(BART/T5)
Prompting
Instruction tuning
CoT
Emergent properties

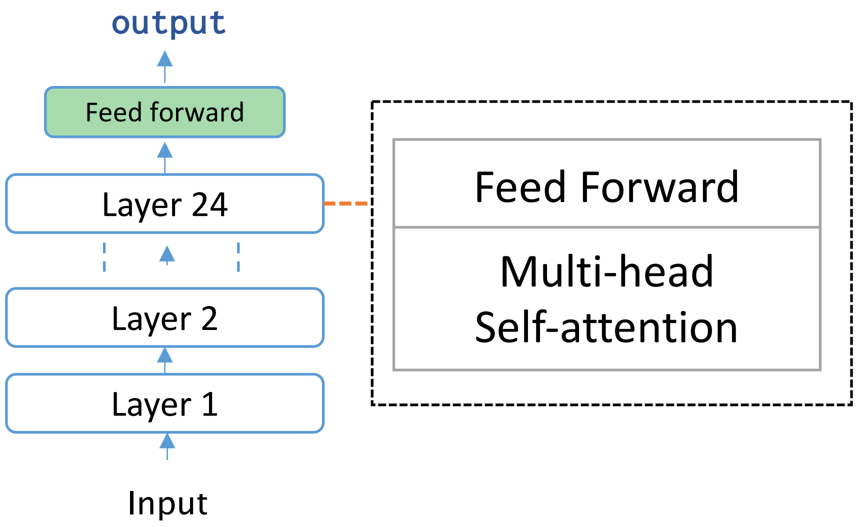

Neural networks taking over
Early attention models
Self-supervised LMs
LLMs
Scale
ICL
Fewshot learning
Scale++
RLHF
State-of-the-art model architecture in NLP
Enabled many advances of modern NLP
2012
2013
2014
2015
2016
2017
2018
2020
2021
2022
2023
CNNs/RNNs everywhere
Transformers
Continuation of Transfer learning
Seq2Seq models
(BART/T5)
Prompting
Instruction tuning
CoT
Emergent properties

Transformer (Vaswani, et al 2017)

- A type of neural network designed primarily for sequence modeling
- Gradually build representations of the input sequence in each layer of the network
- A mechanism to let the model learn how to combine representations across the sequence (Attention)

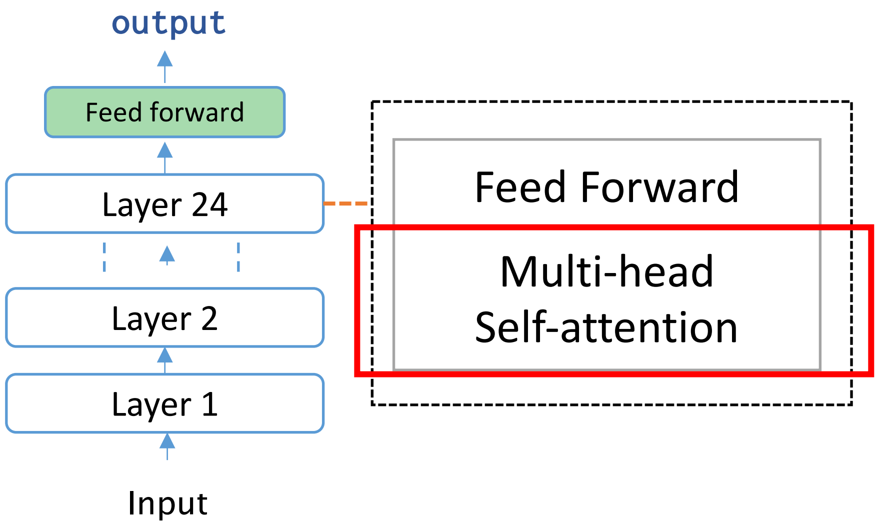
Transformer (Vaswani et al., 2017)

- Each layer consists of two major components
  - Self-attention and feed forward



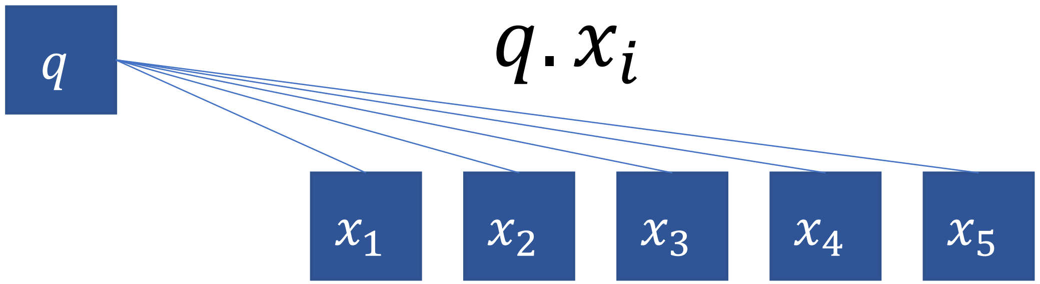
Transformer (Vaswani et al., 2017)

- Each layer consists of two major components
  - Self-attention and feed forward



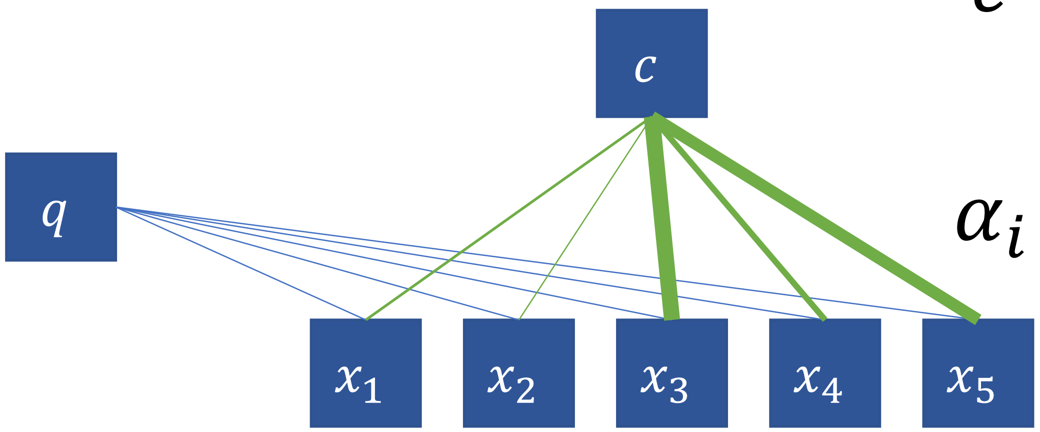
Attention mechanism (Bahdanau et al 2015)

- On high-level allows the model to weigh the importance of different parts of the input to build a contextual representation with respect to a query



Attention mechanism (Bahdanau et al 2015)

- On high-level allows the model to weigh the importance of different parts of the input to build a contextual representation with respect to a query



$$c = \sum_i \alpha_i x_i$$

$$\alpha_i = \mathrm{softmax}(q . x_i)$$

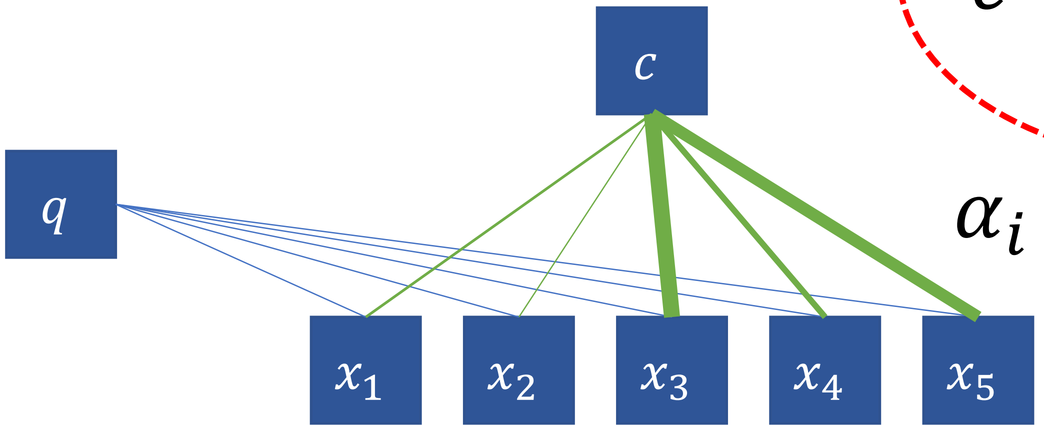
$$\mathrm{softmax}(z_i) = \frac{e_i^z}{\sum_j e^j}$$

Attention mechanism (Bahdanau et al 2015)

- On high-level allows the model to weigh the importance of different parts of the input to build a **contextual representation** with respect to a **query**

contextual representation of the sequence w.r.t the query

$$c = \sum_i \alpha_i x_i$$

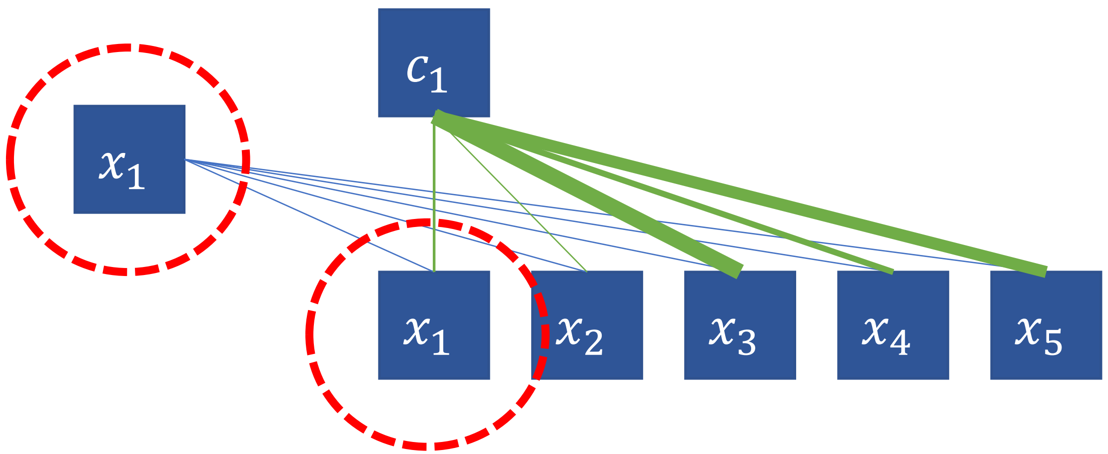


$$\alpha_i = \mathrm{softmax}(q . x_i)$$

$$\mathrm{softmax}(z_i) = \frac{e_i^z}{\sum_j e^j}$$

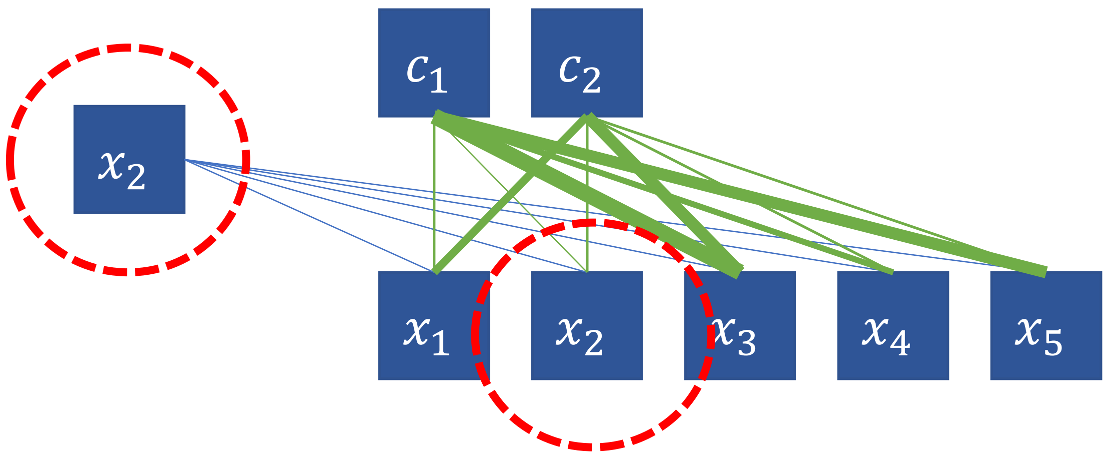
Transformer and self-attention

- Self-attention
- The **query** is the same sequence
  - **For each element x_i in the sequence we compute the contextual representation of the sequence w.r.t. x_i**



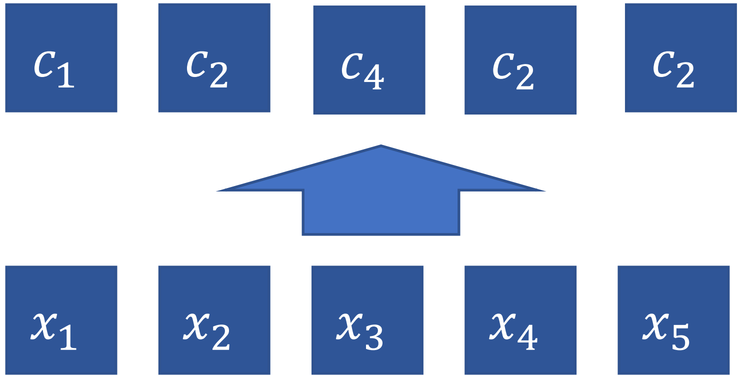
Transformer and self-attention

- Self-attention
- The **query** is the same sequence
  - **For each element x_i in the sequence we compute the contextual representation of the sequence w.r.t. x_i**



Transformer and self-attention

- Self-attention
- The **query** is the same sequence
  - **For each element x_i in the sequence we compute the contextual representation of the sequence w.r.t. x_i**



$$c_i = \sum_j \alpha_{ij} x_j$$

$$\alpha_{ij} = \text{softmax}_j(x_i . x_j)$$

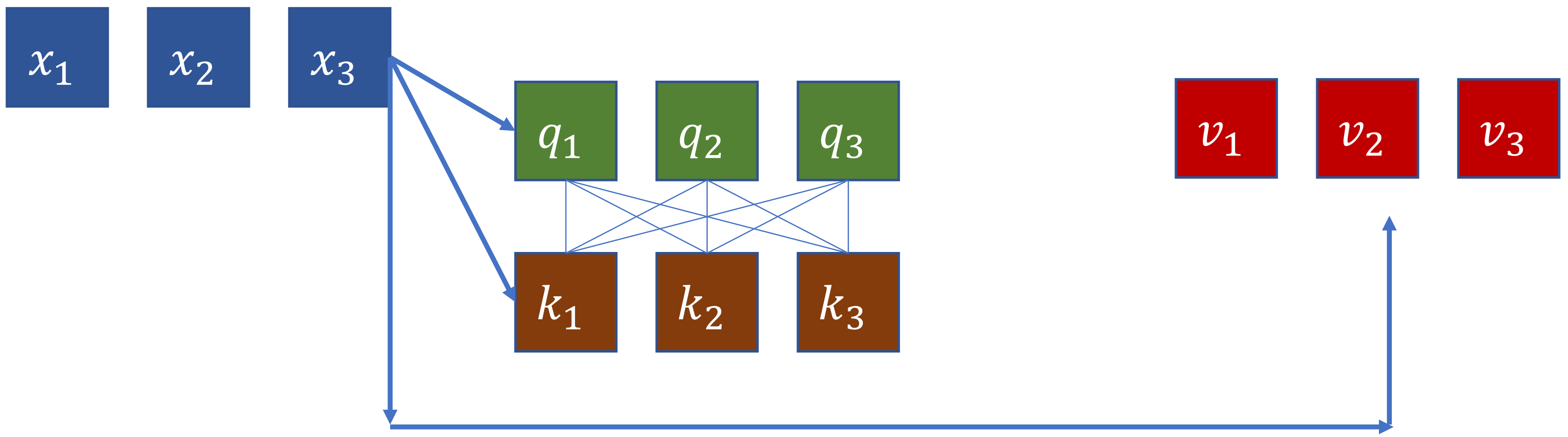
attention scores

Transformer

- Instead of using one vector x_i for each input location
  - Each input vector is projected into three vectors
    - Query vector $q_i = W_q x_i$
    - Key vector $k_i = W_k x_i$
    - Value vector $v_i = W_v x_i$

Transformer

- Instead of using one vector x_i for each input location
  - Each input vector is projected into three vectors
    - Query vector $q_i = W_q x_i$
    - Key vector $k_i = W_k x_i$
    - Value vector $v_i = W_v x_i$

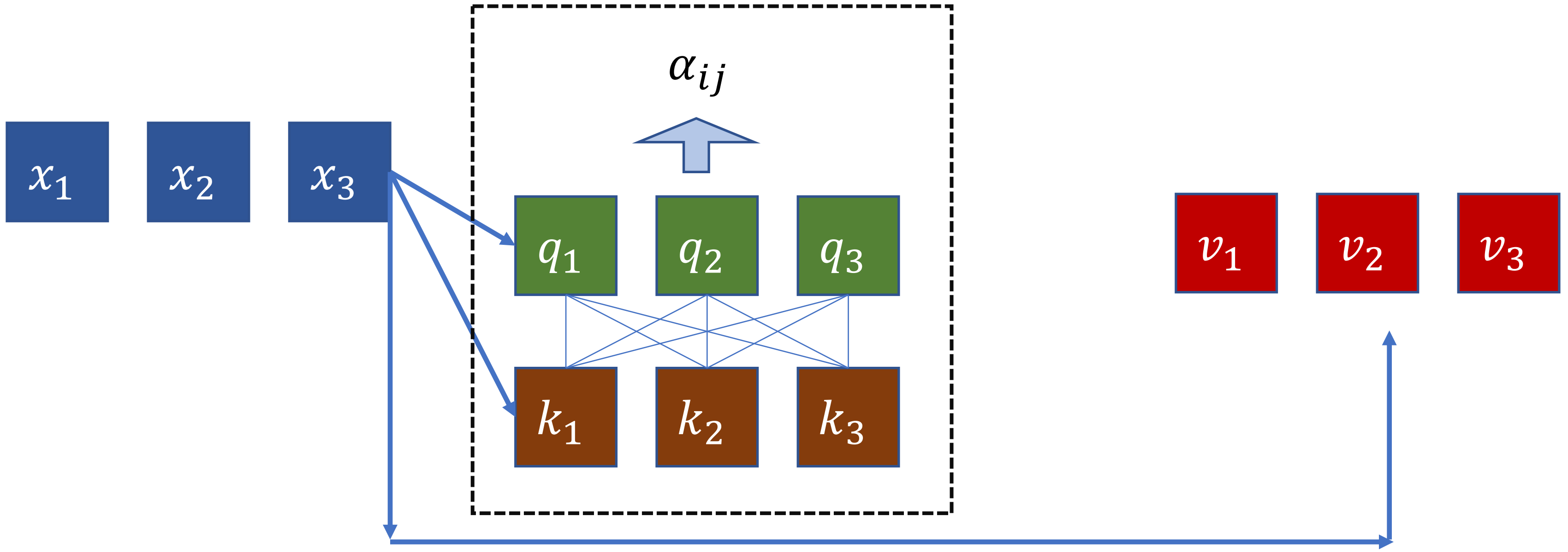


Transformer

- Instead of using one vector x_i for each input location
  - Each input vector is projected into three vectors
    - Query vector $q_i = W_q x_i$
    - Key vector $k_i = W_k x_i$
    - Value vector $v_i = W_v x_i$

$$\alpha_{ij} = \mathrm{softmax}_j(q_i . k_j)$$

attention scores

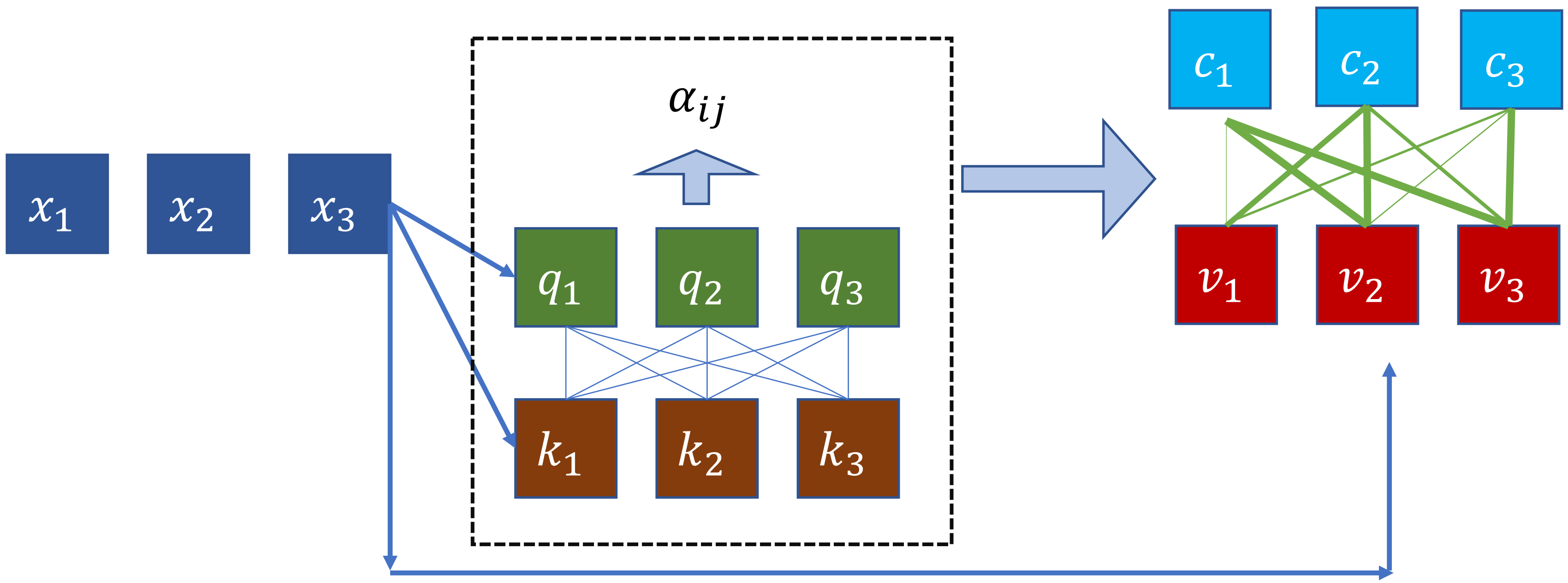


Transformer

- Instead of using one vector x_i for each input location
  - Each input vector is projected into three vectors
    - Query vector $q_i = W_q x_i$
    - Key vector $k_i = W_k x_i$
    - Value vector $v_i = W_v x_i$

$$\alpha_{ij} = \mathrm{softmax}_j(q_i . k_j)$$

$$c_i = \sum_j \alpha_{ij} v_j$$



Transformer

- Vaswani et al., 2017 used a slightly modified version of attention
- They re-scale the logits by a factor of $\frac{1}{\sqrt{d_k}}$ where d_k is the dimension of the key vector

$$\alpha_{ij} = \mathrm{softmax}_j(q_i . k_j) \longrightarrow \alpha_{ij} = \mathrm{softmax}_j\left(\frac{q_i . k_j}{\sqrt{d_k}}\right)$$

Putting it all together: Implementation

```python
def self_attention(query, key, value):
    d_k = query.size(-1)
    attn_logits = torch.matmul(query, key.transpose(-2, -1)) / math.sqrt(d_k)
    attn_scores = F.softmax(attn_logits, dim=-1)
    context_vectors = torch.matmul(attn_scores, value)
    return context_vectors, attn_scores
```

Multi-head attention

- Vaswani et al., (2017) use multiple attention heads instead of one
  - Perform self_attention times: N number of heads
    - Query vector $q_i^n = W_q^n x_i$ (W_q^n: Query projection weights for head $1 \leq n \leq N$)
    - Key vector $k_i^n = W_k^n x_i$
    - Value vector $v_i^n = W_v^n x_i$

  - Concatenate the resulting context vectors and project the output

$$c_i^n = \sum_j \text{softmax}\left(\frac{q_i^n k_j^n}{\sqrt{d_k}}\right) v_j^n$$

$$c_i = \text{concat}\left(c_i^1, \dots, c_i^N\right). W^o$$

Multi-head attention

- Example dimensions (used in original Transformer)
  - $d = 512$ (input vector dimension)
  - $h = 8$ (number of attention heads)
  - $d_v = d_k = d_q = \frac{512}{8} = 64$

$$c_i^n = \sum_j \operatorname{softmax}\left(\frac{q_i^n k_j^n}{\sqrt{d_k}}\right) v_j^n$$

$$c_i = \operatorname{concat}\left(c_i^1, \ldots, c_i^N\right). W^o \qquad W^o \in \mathbb{R}^{h.d_v \times d}$$

- What is the output vector dimension c_i?

Multi-head attention

- Example dimensions (used in original Transformer)
  - $d = 512$ (input vector dimension)
  - $h = 8$ (number of attention heads)
  - $d_v = d_k = d_q = \frac{512}{8} = 64$

$$c_i^n = \sum_j \mathrm{softmax}\left(\frac{q_i^n k_j^n}{\sqrt{d_k}}\right) v_j^n$$

$$c_i = \mathrm{concat}\left(c_i^1, \dots, c_i^N\right). W^o \qquad W^o \in \mathbb{R}^{h.d_v \times d}$$

  - What is the output vector dimension c_i? Output dimension is *d=512*

Multi-head attention

```
class TransformeMultiHeadAttention:
    def __init__(self, d_model, n_heads, dropout=0.1):
        self.d_model = d_model
        self.n_heads = n_heads
        self.d_k = d_model // n_heads
        self.w_q = nn.Linear(d_model, d_model)
        self.w_k = nn.Linear(d_model, d_model)
        self.w_v = nn.Linear(d_model, d_model)
        self.w_o = nn.Linear(d_model, d_model)
        self.dropout = nn.Dropout(dropout)

    def forward(self, q, k, v):
        batch_size = q.size(0)
        q = self.w_q(q).view(batch_size, -1, self.n_heads, self.d_k).transpose(1, 2)
        k = self.w_k(k).view(batch_size, -1, self.n_heads, self.d_k).transpose(1, 2)
        v = self.w_v(v).view(batch_size, -1, self.n_heads, self.d_k).transpose(1, 2)
        context_vectors, attn_scores = self_attention(q, k, v)
        context_vectors = context_vectors.transpose(1, 2).contiguous().view(batch_size, -1, self.d_model)
        return self.w_o(context_vectors), attn_scores
```

Multi-head attention

```python
class TransformeMultiHeadAttention:
    def __init__(self, d_model, n_heads, dropout=0.1):
        self.d_model = d_model
        self.n_heads = n_heads
        self.d_k = d_model // n_heads
        self.w_q = nn.Linear(d_model, d_model)
        self.w_k = nn.Linear(d_model, d_model)
        self.w_v = nn.Linear(d_model, d_model)
        self.w_o = nn.Linear(d_model, d_model)
        self.dropout = nn.Dropout(dropout)

    def forward(self, q, k, v):
        batch_size = q.size(0)
        q = self.w_q(q).view(batch_size, -1, self.n_heads, self.d_k).transpose(1, 2)
        k = self.w_k(k).view(batch_size, -1, self.n_heads, self.d_k).transpose(1, 2)
        v = self.w_v(v).view(batch_size, -1, self.n_heads, self.d_k).transpose(1, 2)
        context_vectors, attn_scores = self_attention(q, k, v)
        context_vectors = context_vectors.transpose(1, 2).contiguous().view(batch_size, -1, self.d_model)
        return self.w_o(context_vectors), attn_scores
```

q,k,v projections

output projection

Multi-head attention

```python
class TransformeMultiHeadAttention:
    def __init__(self, d_model, n_heads, dropout=0.1):
        self.d_model = d_model
        self.n_heads = n_heads
        self.d_k = d_model // n_heads
        self.w_q = nn.Linear(d_model, d_model)
        self.w_k = nn.Linear(d_model, d_model)
        self.w_v = nn.Linear(d_model, d_model)
        self.w_o = nn.Linear(d_model, d_model)
        self.dropout = nn.Dropout(dropout)

    def forward(self, q, k, v):
        batch_size = q.size(0)
        q = self.w_q(q).view(batch_size, -1, self.n_heads, self.d_k).transpose(1, 2)
        k = self.w_k(k).view(batch_size, -1, self.n_heads, self.d_k).transpose(1, 2)
        v = self.w_v(v).view(batch_size, -1, self.n_heads, self.d_k).transpose(1, 2)
        context_vectors, attn_scores = self_attention(q, k, v)
        context_vectors = context_vectors.transpose(1, 2).contiguous().view(batch_size, -1, self.d_model)
        return self.w_o(context_vectors), attn_scores
```

q,k,v projections

output projection

Main self-attention function

Final projection

Feed forward layer

- A position-wise transformation consisting of:
  - A linear transformation, non-linear activation f (e.g., ReLU), and another linear transformation.

$$FF(c) = f(cW_1 + b_1)W_2 + b_2$$

Feed forward layer

- A position-wise transformation consisting of:
  - A linear transformation, non-linear activation f (e.g., ReLU), and another linear transformation.

$$FF(c) = f(cW_1 + b_1)W_2 + b_2$$

  - This allows the model to apply another transformation to the contextual representations (or "post-process" them)
  - Usually the dimensionality of the hidden feedforward layer is 2-8 times larger than the input dimension

Feed forward layer

```
class TransformerFeedForward:

    def __init__(self, d_model, d_ff, dropout=0.1):
        self.w_1 = nn.Linear(d_model, d_ff)
        self.w_2 = nn.Linear(d_ff, d_model)
        self.dropout = nn.Dropout(dropout)

    def forward(self, x):
        x = self.w_2(F.relu(self.w_1(x)))
        return self.dropout(x)
```