Scaling Language Models: Memorization, Compression, and Emergence Abilities

Dani Yogatama

University of Southern California, Reka

The State of Natural Language Processing

State-of-the-art models are based on increasingly larger neural networks.

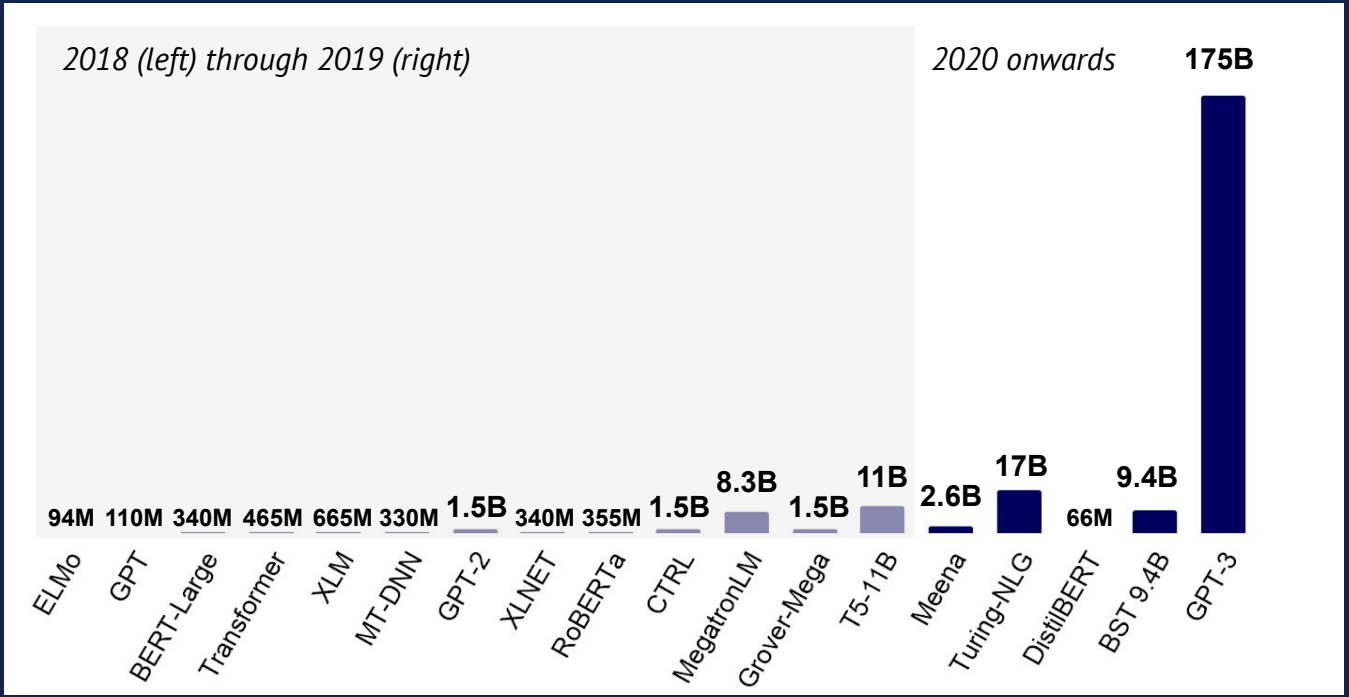


Figure taken from State of AI Report 2020.

- Scaling up is not everything as recent advances have shown (e.g., Chinchilla; Hoffmann et al., 2022).

- But it’s clear that it is a necessary ingredient for future progress (at least in the short term).

This Talk

- How to scale efficiently (Tay et al., ICLR 2022; Tay et al., arXiv 2022).

This Talk

- How to scale efficiently (Tay et al., ICLR 2022; Tay et al., arXiv 2022).
- What do we get from scaling up (Wei et al., TMLR 2022).

This Talk

- How to scale efficiently (Tay et al., ICLR 2022; Tay et al., arXiv 2022).
- What do we get from scaling up (Wei et al., TMLR 2022).
- Scaling down: memorization, compression, distillation (Yogatama et al., TACL 2021; Sachan et al., NeurIPS 2021; Peng et al., ACL 2022; Liu et al., TACL 2022).

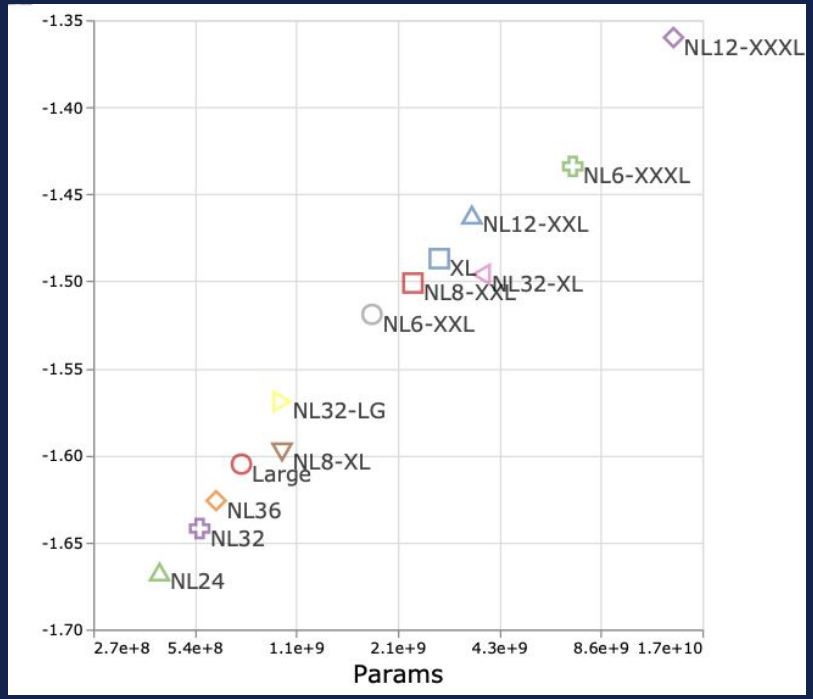
Scale efficiently: insights from pretraining and finetuning transformers

ICLR 2022

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, Donald Metzler

Google

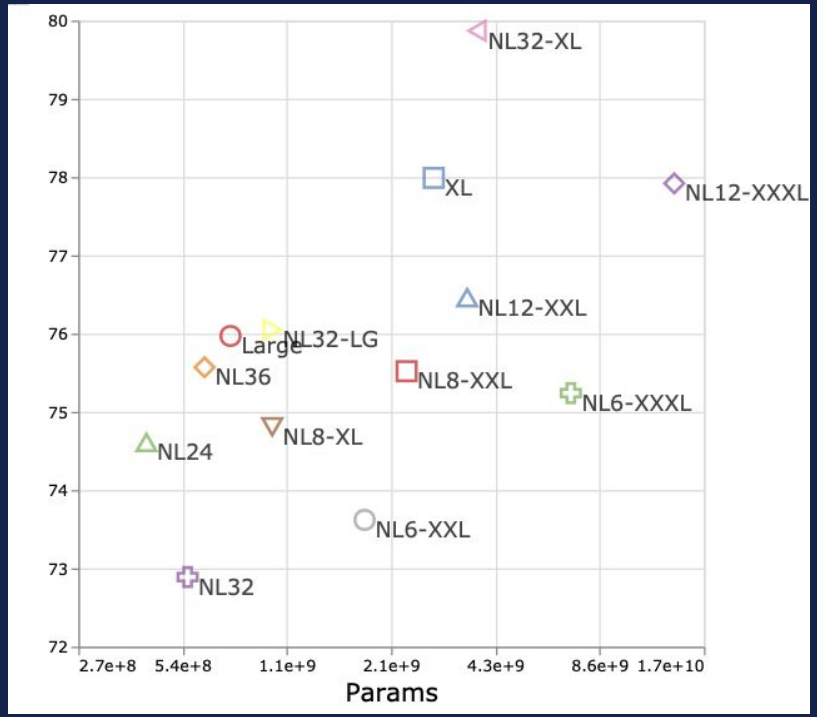
Upstream vs downstream performance



Transformers from 16M to 30B (XXXL) parameters.

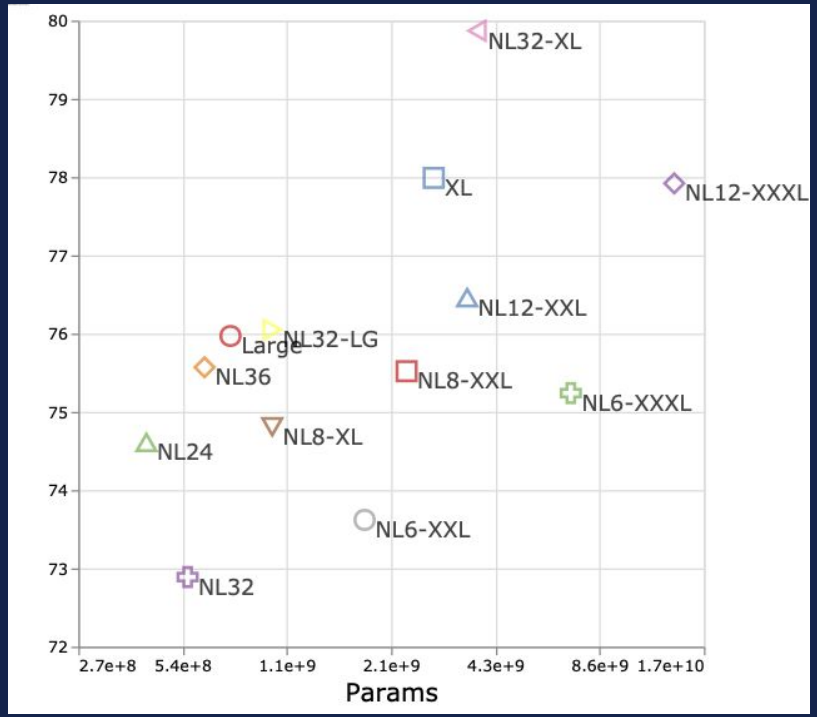
Predictable decrease in perplexity as we increase the model size.

Upstream vs downstream performance



Unpredictable downstream performance (SuperGLUE)

Upstream vs downstream performance

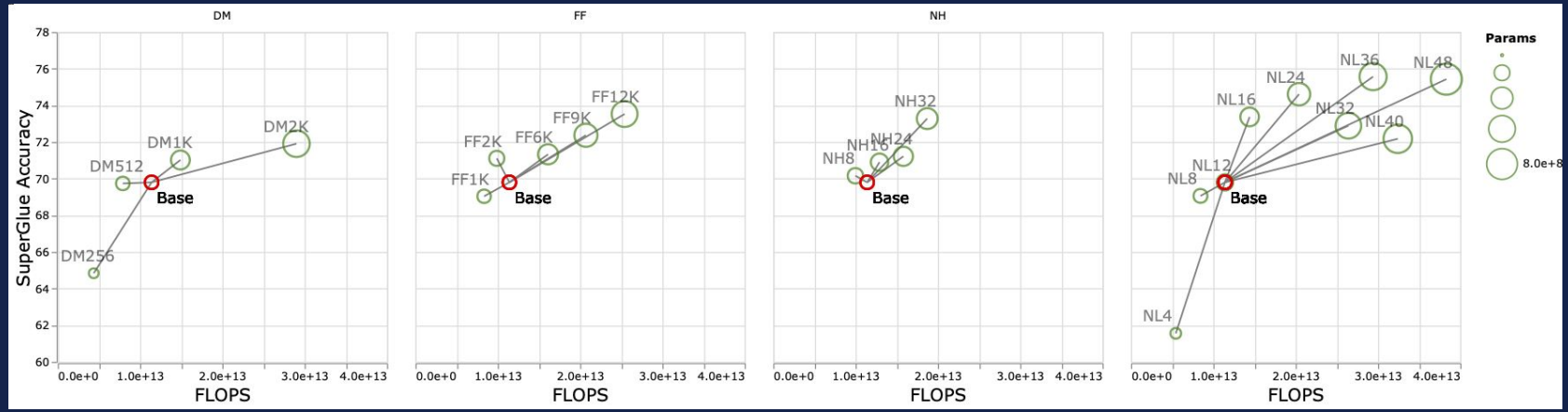


Unpredictable downstream performance (SuperGLUE)

Model shape matters when scaling up

Upstream vs downstream performance

Best strategy is *deep narrow*

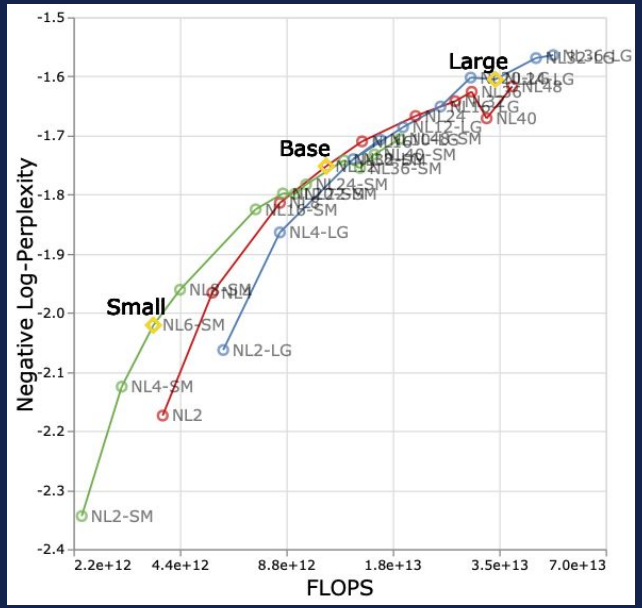


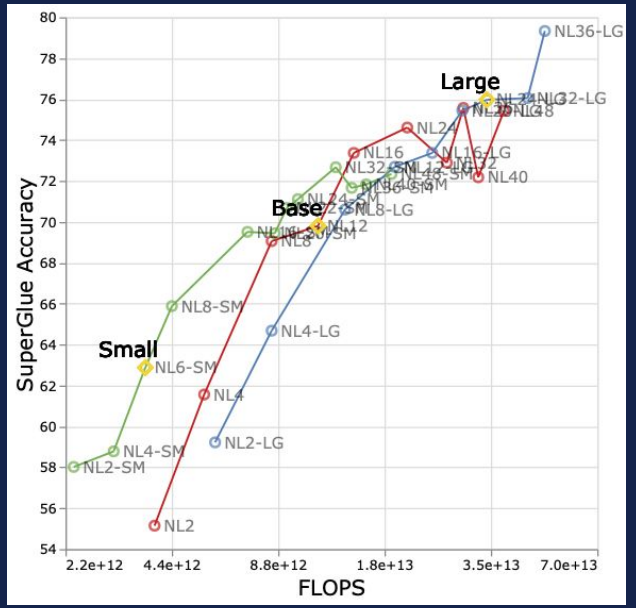
Upstream vs downstream performance

Best strategy is *deep narrow*

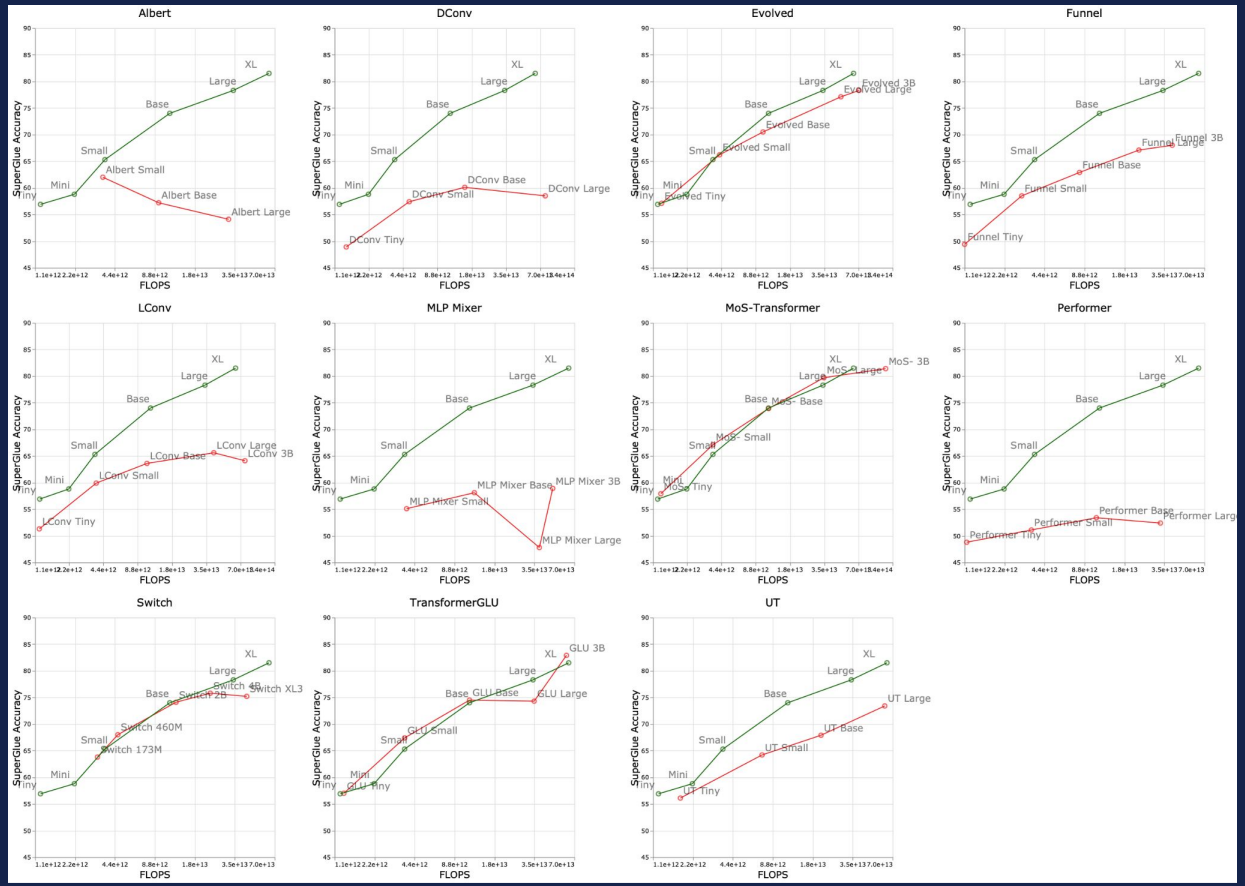
Model	#Params	#TFlops	Steps/s	Ppl (C4)	GLUE	SGLUE	SQuAD	AVG
Small	61M	3.7	23	-2.021	77.45	62.88	80.39	73.57
Mini-8L	50M	3.2	24	-2.056	77.11	63.35	80.12	73.52
Base	223M	11	9	-1.752	82.53	69.80	85.14	79.16
Small 16L	134M	7.2	13	-1.825	82.57	69.51	84.12	78.73
Small 20L	164M	8.6	11	-1.798	83.22	69.44	85.23	79.30
Small 22L	179M	9.3	10	-1.798	82.52	70.68	85.39	79.54
Small 24L	193M	10	9	-1.783	83.11	71.11	85.45	79.92
Small 32EL	143M	10	10	-1.897	82.77	70.66	86.01	79.81
Large	738M	34	4	-1.605	85.08	75.97	87.55	82.87
Base 36L	621M	29	3	-1.626	85.26	75.57	87.84	82.89
XL	2.9B	64	1	-1.487	86.49	77.99	88.70	84.38
Large 36L	1.1B	50	2	-1.564	87.22	79.34	89.21	85.27
XXL	11.3B	367	1	-1.430	86.91	79.20	89.50	85.20
XL 32L	3.8B	169	3	-1.500	86.94	79.87	89.46	85.42

Pareto efficient models





How does architectural bias affect scaling?



Emergence Abilities of Large Language Models

Is scaling unavoidable?

Are there things we can only do with large scale models?

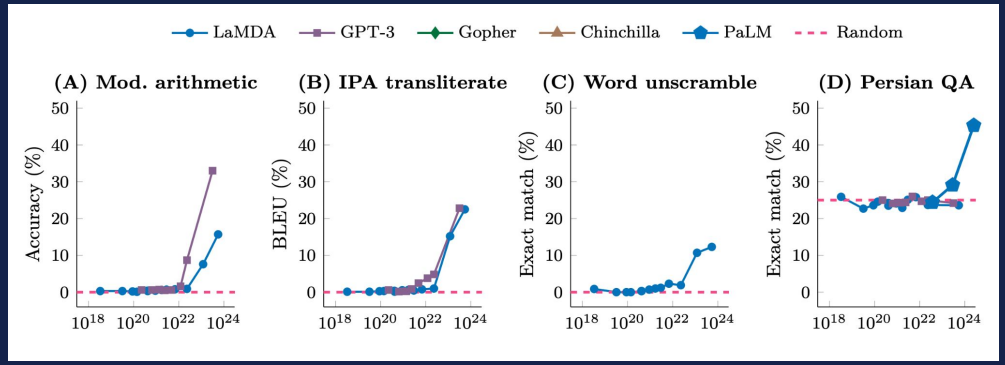
Emergence Abilities of Large Language Models

TMLR 2022

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, William Fedus

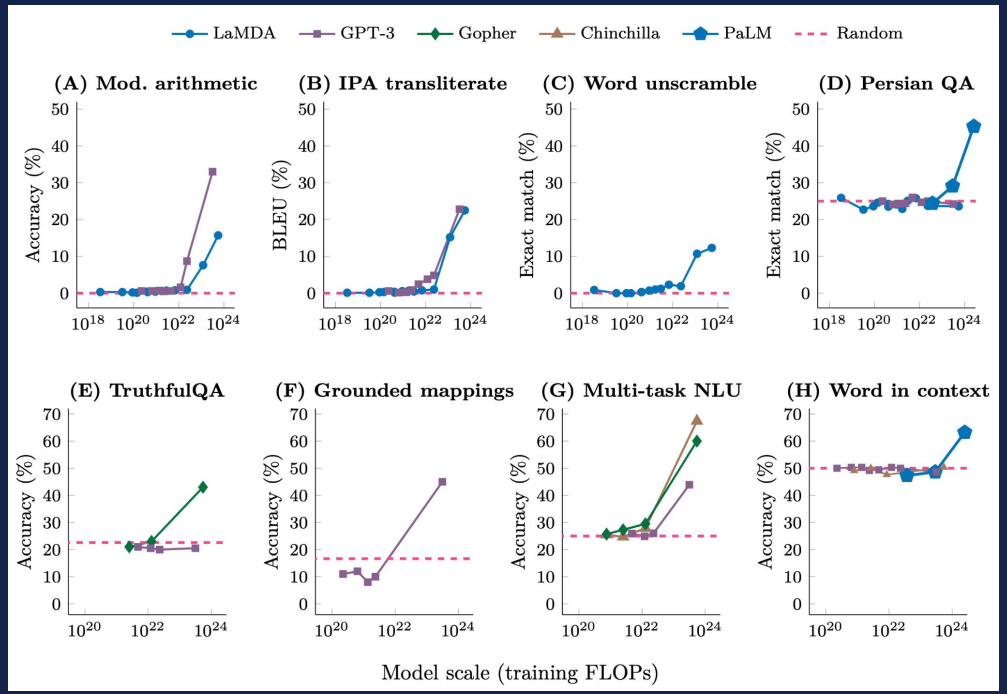
Emergence Abilities of Large Language Models

An ability is emergent if it is not present in smaller models but is present in larger models (as measured by training FLOPS in the below figure).



Emergence Abilities of Large Language Models

An ability is emergent if it is not present in smaller models but is present in larger models (as measured by training FLOPS in the below figure).



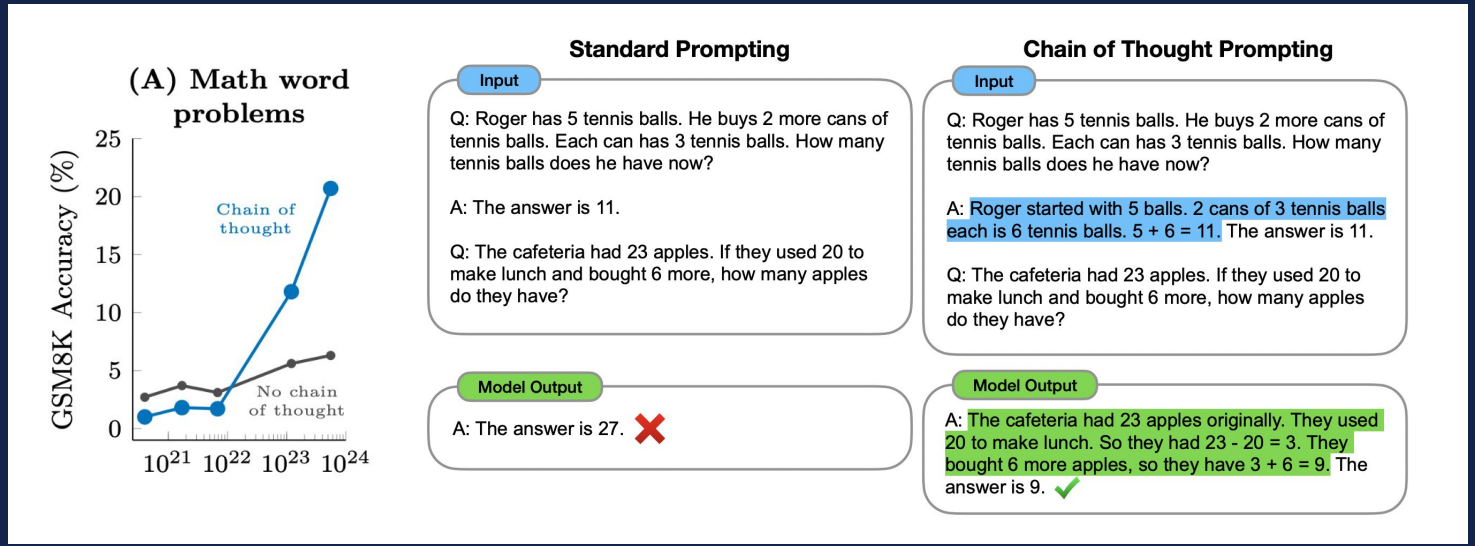
Emergence Abilities of Large Language Models

Training FLOPs / model size is not everything.

For the same model size, different techniques can prompt emergent abilities.

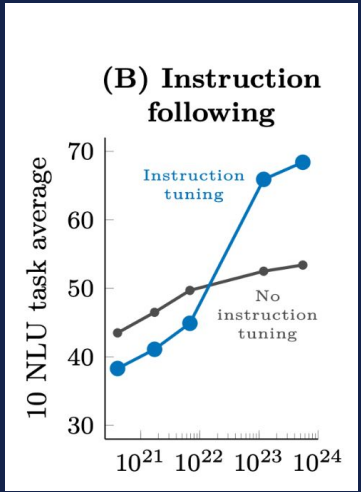
Emergence Abilities of Large Language Models

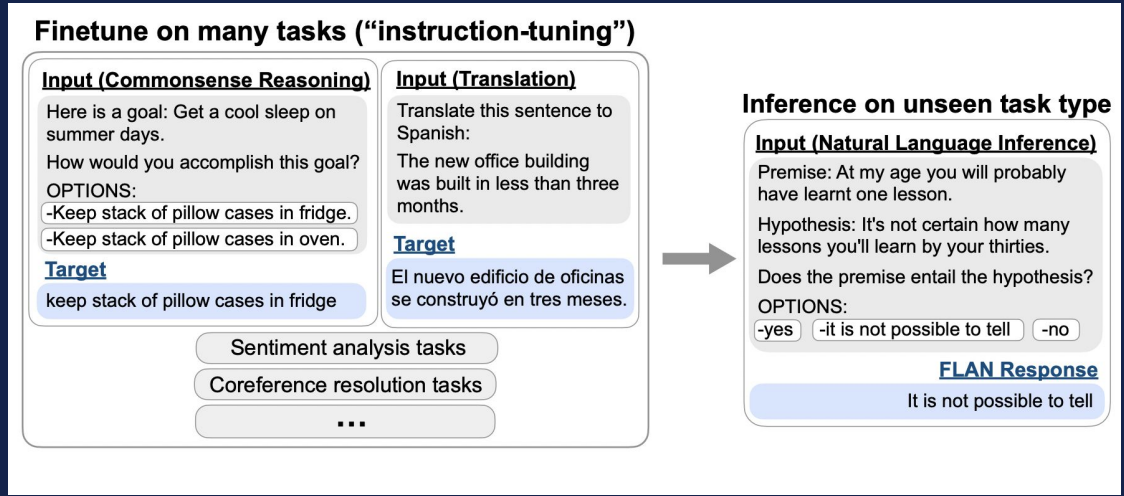
Chain of thought (Wei et al., 2022)



Emergence Abilities of Large Language Models

Instruction tuning (Wei et al., 2022)





Emergence Abilities of Large Language Models

Scratchpad (Nye et al., 2021)

DIRECT EXECUTION PREDICTION

Consider the following Python function:

```
def remove_Occ(s,ch):
    for i in range(len(s)):
        if (s[i] == ch):
            s = s[0 : i] + s[i + 1:]
            break
    for i in range(len(s) - 1,-1,-1):
        if (s[i] == ch):
            s = s[0 : i] + s[i + 1:]
            break
    return s

Fill in the ??? below:
assert remove_Occ("PHP","P") == ???
```

Large Language Model →

```
assert remove_Oct("PHP", "P") == "H"
```

SCRATCHPAD TRACING

Consider the following Python function:

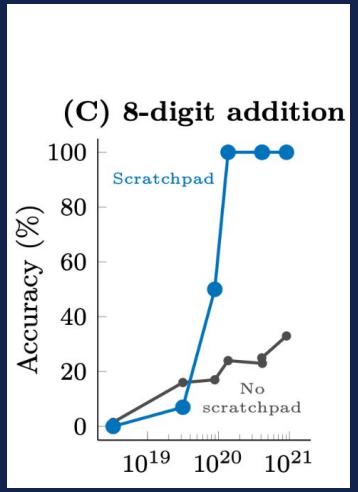
```
def remove_Occ(s,ch):
    for i in range(len(s)):
        if (s[i] == ch):
            s = s[0 : i] + s[i + 1:]
            break
    for i in range(len(s) - 1,-1,-1):
        if (s[i] == ch):
            s = s[0 : i] + s[i + 1:]
            break
    return s

output = remove_Occ("PHP","P")

What is the execution trace?
```

Large Language Model →

```
state: {}
line: def remove_Occ(s,ch):
state: {"remove_Occ": "<callable_object remove_Occ>"}
line: output = remove_Occ("PHP","P")
state: {"ch": "P", "s": "PHP"}
line:     for i in range(len(s)):
state: {"ch": "P", "s": "PHP", "i": 0}
line:         if (s[i] == ch):
state: {"ch": "P", "s": "PHP", "i": 0}
line:             s = s[0 : i] + s[i + 1:]
state: {"ch": "P", "s": "HP", "i": 0}
line:             break
state: {"ch": "P", "s": "HP", "i": 0}
line:     for i in range(len(s) - 1,-1,-1):
state: {"ch": "P", "s": "HP", "i": 1}
line:         if (s[i] == ch):
state: {"ch": "P", "s": "HP", "i": 1}
line:             s = s[0 : i] + s[i + 1:]
state: {"ch": "P", "s": "H", "i": 1}
line:             break
state: {"ch": "P", "s": "H", "i": 1}
line:     return s
state: {"remove_Occ": "<callable_object remove_Occ>",
 "output": "H"}
```



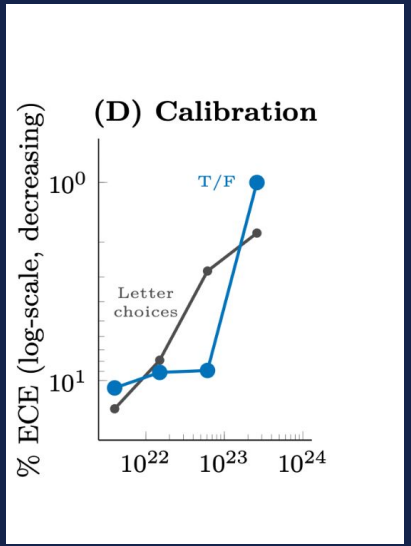
Emergence Abilities of Large Language Models

Calibration: use model probability to evaluate whether the model can answer a given question correctly (Kadavath et al., 2022).

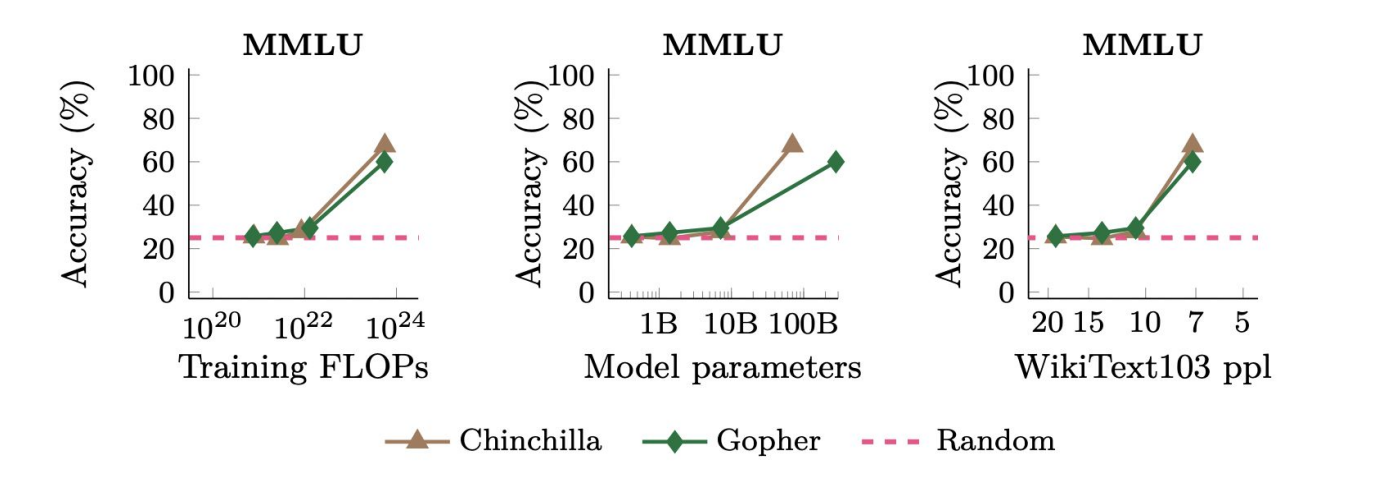
Who is the current president of the USA?

(A) Donald Trump
(B) Barack Obama
(C) Joe Biden

The answer is: (C) 67%



FLOPs vs model parameters vs upstream perplexity



The Road Ahead

There are a lot of things we don’t understand yet about scaling.

The Road Ahead

There are a lot of things we don’t understand yet about scaling.

Alternative model architectures and inference techniques.

The Road Ahead

There are a lot of things we don’t understand yet about scaling.

Alternative model architectures and inference techniques.

Model scaling vs. data scaling.

The Road Ahead

There are a lot of things we don’t understand yet about scaling.

Alternative model architectures and inference techniques.

Model scaling vs. data scaling.

Many things we can do without massive computational resources.

The Road Ahead

Goal: emergent abilities at much smaller scale.

What kind of models?

How to use these models in practice?

The Road Ahead

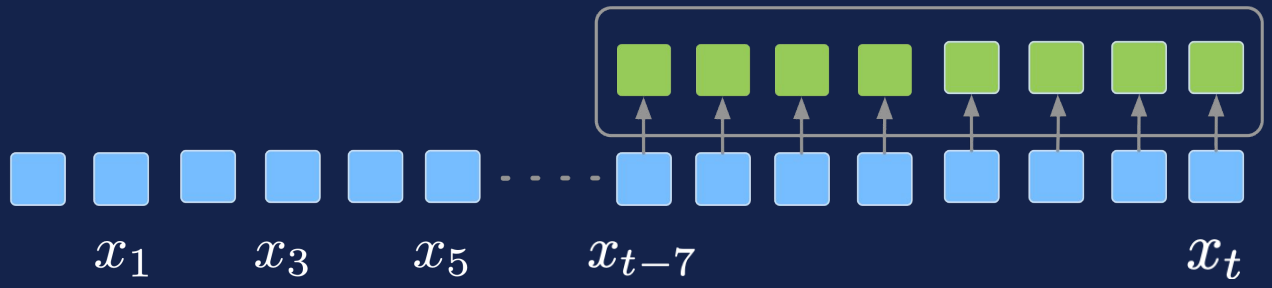
Goal: emergent abilities at much smaller scale.

What kind of models?

How to use these models in practice?

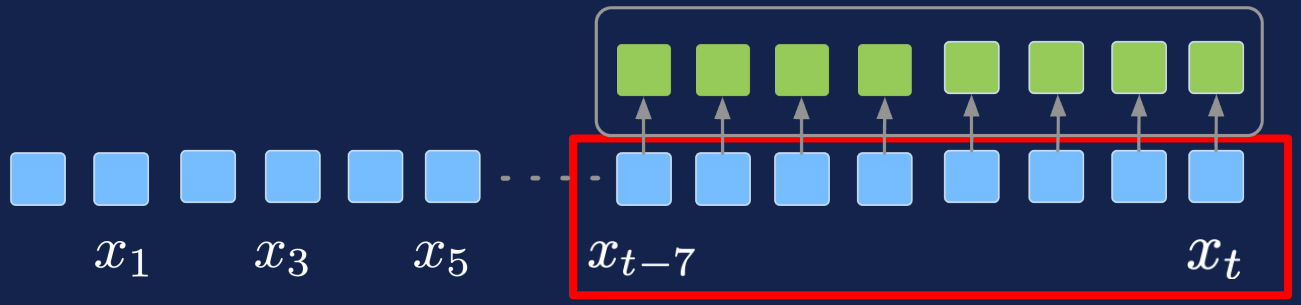
Semiparametric Language Models

State of the art architectures (transformers) are limited by the input sequence length.



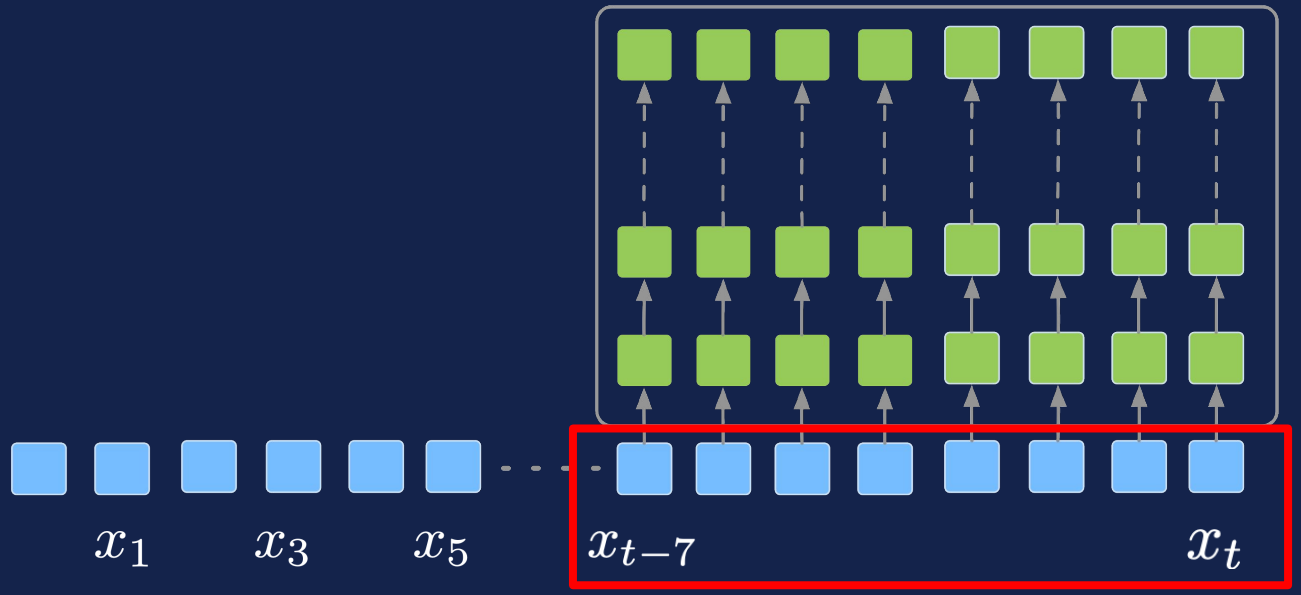
Semiparametric Language Models

State of the art architectures (transformers) are limited by the input sequence length.



Semiparametric Language Models

State of the art architectures (transformers) are limited by the input sequence length.

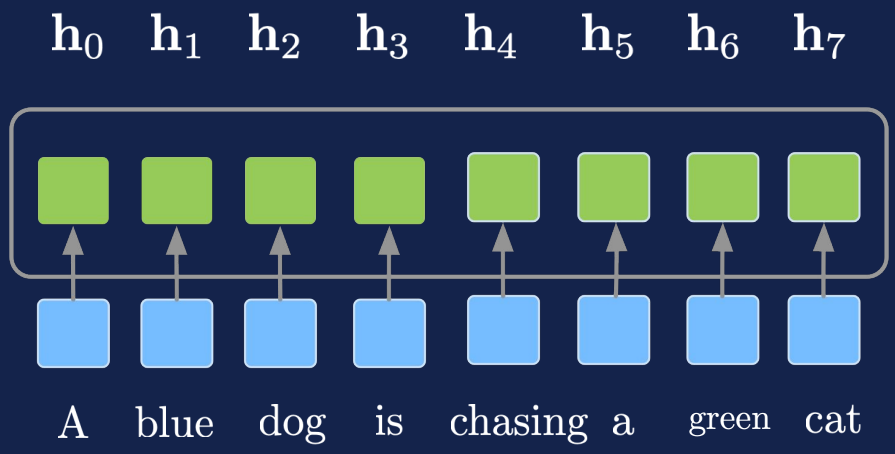


Semiparametric Language Models

Knowledge is encoded in the weights of a parametric neural network.

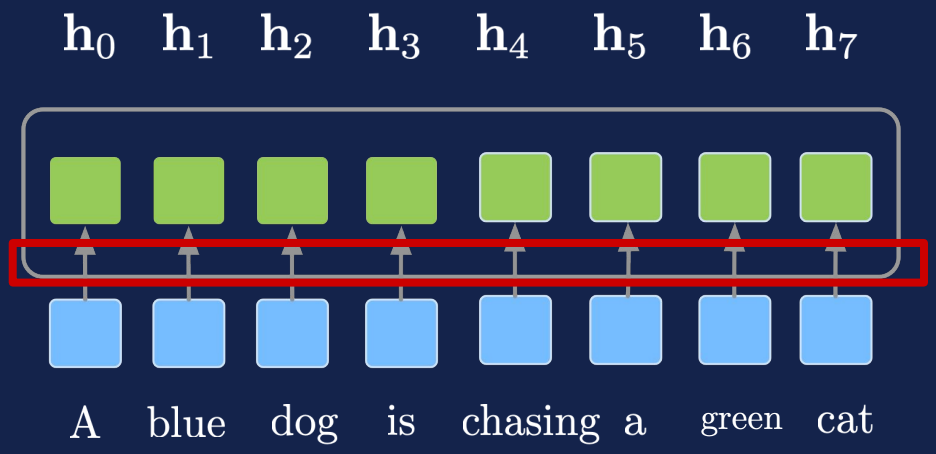
Semiparametric Language Models

Knowledge is encoded in the weights of a parametric neural network.



Semiparametric Language Models

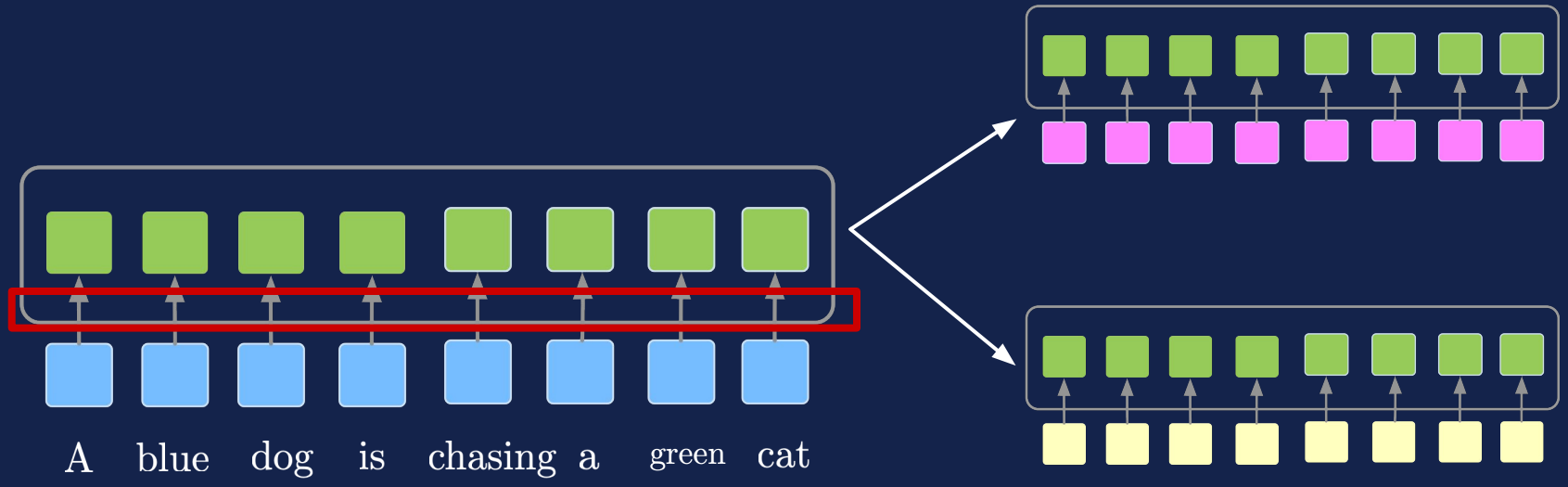
Knowledge is encoded in the weights of a parametric neural network.



$$g_\omega(x_0, x_1, \ldots, x_7) \rightarrow \mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_7$$

Semiparametric Language Models

Knowledge is encoded in the weights of a parametric neural network.



Update weights with new knowledge → changes affect all examples (sequences).

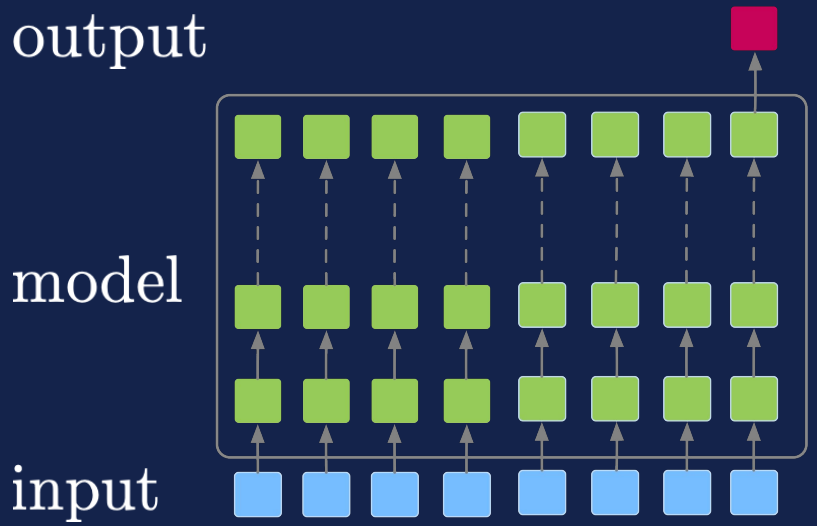
Semiparametric Models

Separation of computation and storage as an architectural bias.

Nematzadeh, Ruder, Yogatama, ICLR Bridging AI and Cognitive Science Workshop 2020

Semiparametric Models

Separation of computation and storage as an architectural bias.

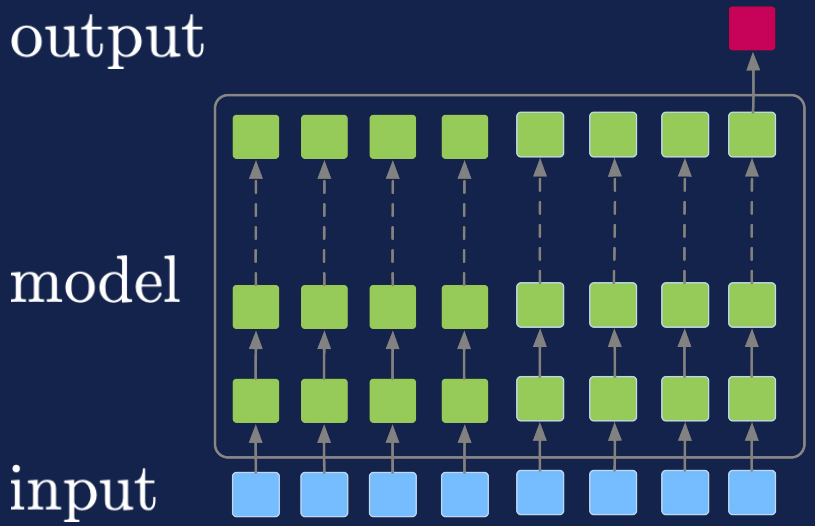


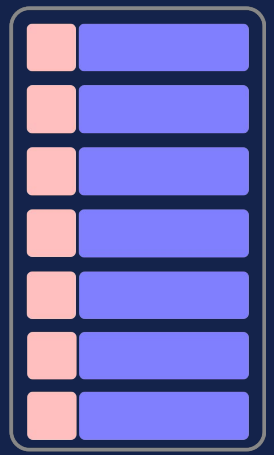
Computation: a neural network

Nematzadeh, Ruder, Yogatama, ICLR Bridging AI and Cognitive Science Workshop 2020

Semiparametric Models

Separation of computation and storage as an architectural bias.





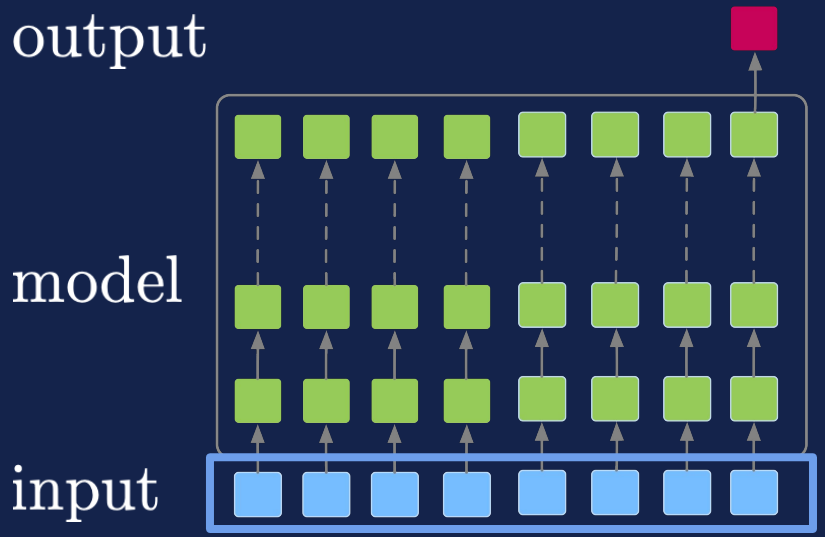
Computation: a neural network

Memory (storage): a key-value database

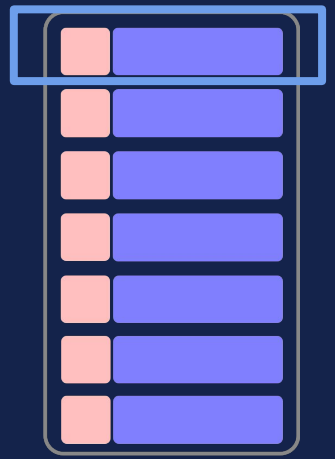
Nematzadeh, Ruder, Yogatama, ICLR Bridging AI and Cognitive Science Workshop 2020

Semiparametric Models

Separation of computation and storage as an architectural bias.



Computation: a neural network

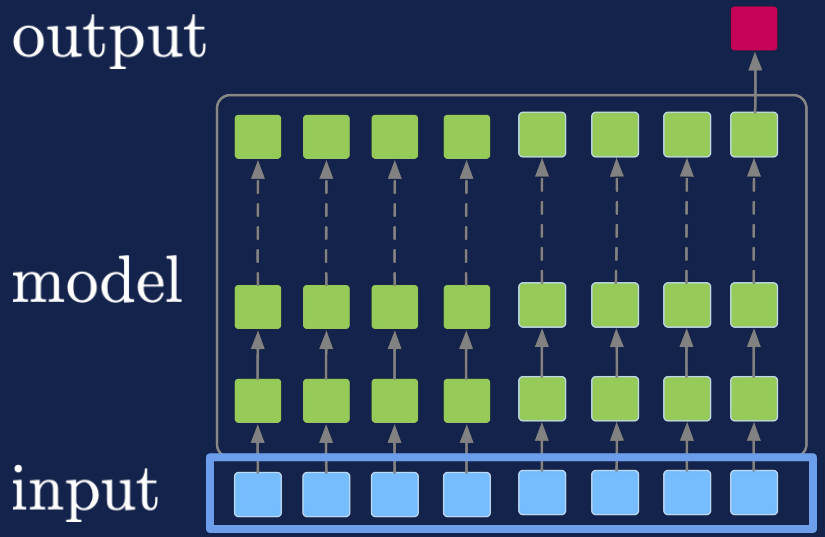


Memory (storage): a key-value database

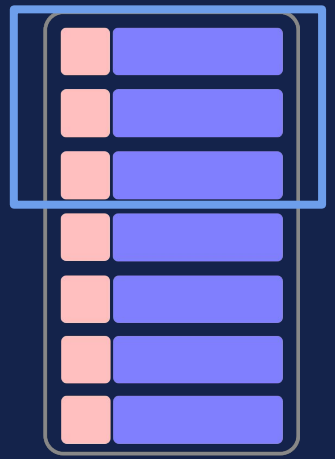
Nematzadeh, Ruder, Yogatama, ICLR Bridging AI and Cognitive Science Workshop 2020

Semiparametric Models

Separation of computation and storage as an architectural bias.



Computation: a neural network



Memory (storage): a key-value database

Nematzadeh, Ruder, Yogatama, ICLR Bridging AI and Cognitive Science Workshop 2020

Semiparametric Models

Episodic Memory in Lifelong Language Learning

de Masson d'Autume, Ruder, Kong, Yogatama, NeurIPS 2019

Adaptive Semiparametric Language Models

Yogatama, de Masson d'Autume, Kong, TACL 2021

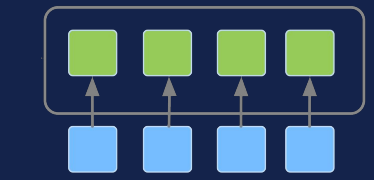
End-to-end Training of Multi Document Reader and Retriever for Open Domain QA

Sachan, Reddy, Hamilton, Dyer, Yogatama, NeurIPS 2021

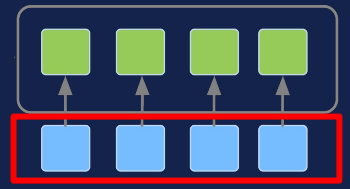
Relational Memory Augmented Language Models

Liu, Yogatama, Blunsom, TACL 2022

SemiPArametric LM (SPALM)



SemiPArametric LM (SPALM)

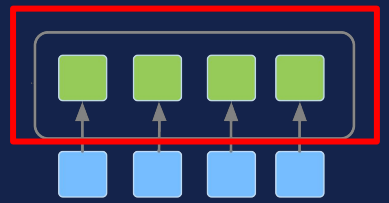


Input: a sequence of tokens.

SemiPArametric LM (SPALM)

Encoder: transformer

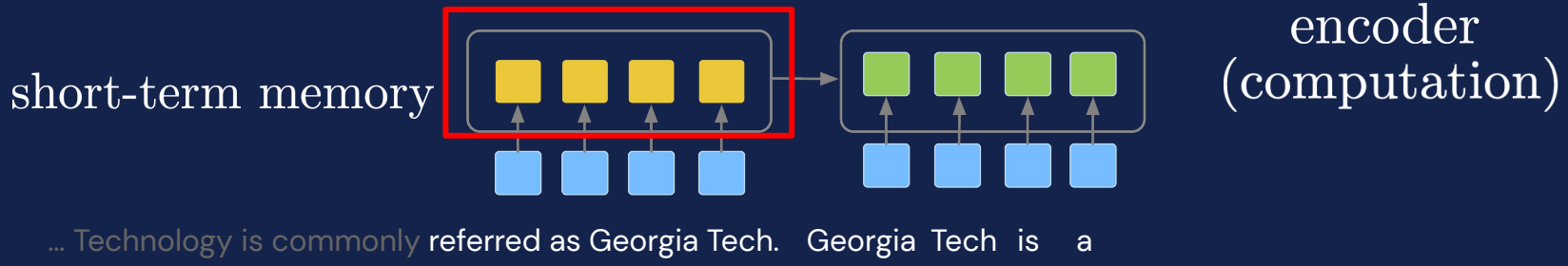
(Vaswani et al., 2017)



encoder
(computation)

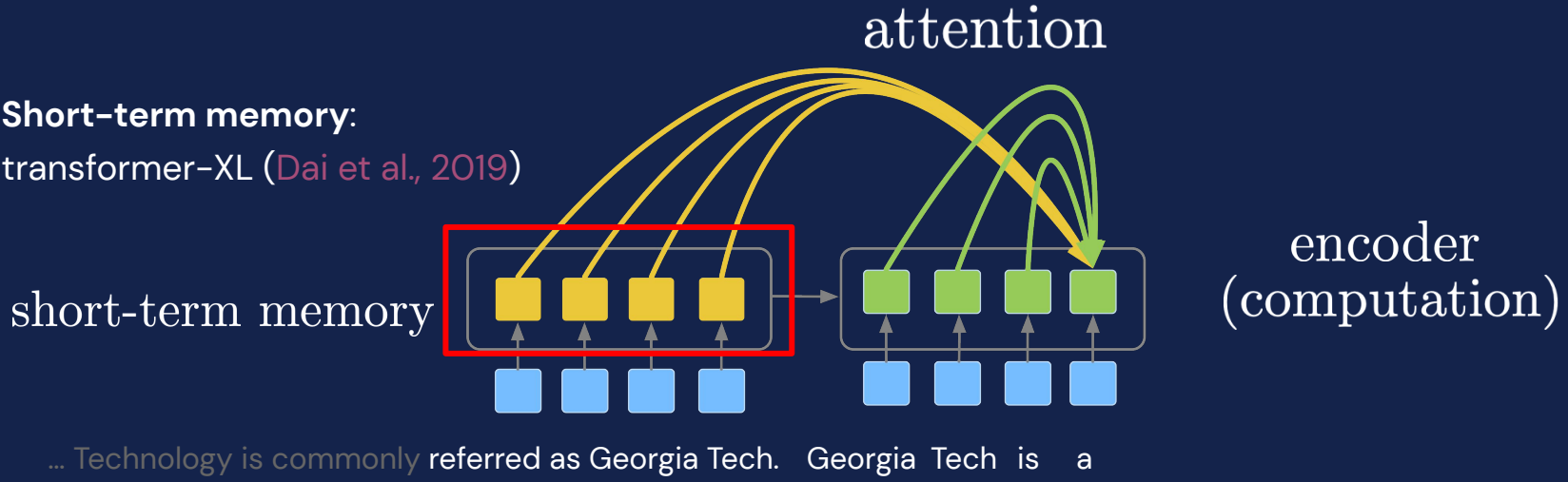
SemiPArametric LM (SPALM)

Short-term memory:
transformer-XL (Dai et al., 2019)



SemiPArametric LM (SPALM)

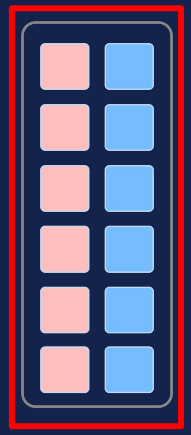
Short-term memory:
transformer-XL (Dai et al., 2019)



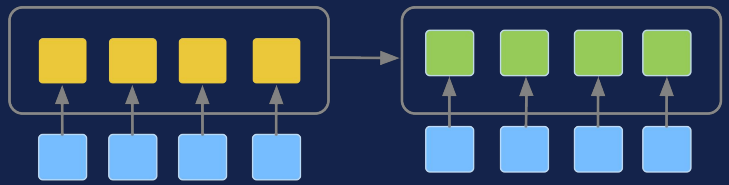
SemiPArametric LM (SPALM)

Long-term memory:
key-value database

long-term memory



short-term memory

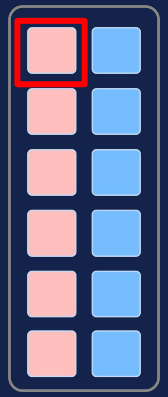


encoder
(computation)

... Technology is commonly referred as Georgia Tech. Georgia Tech is a

SemiPArametric LM (SPALM)

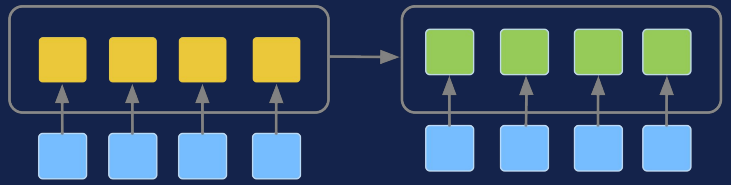
long-term memory



Key: compressed long-term context

USA is a beautiful

short-term memory

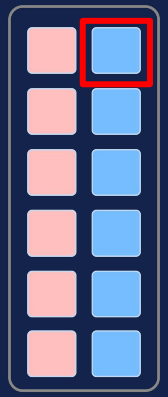


encoder (computation)

... Technology is commonly referred as Georgia Tech. Georgia Tech is a

SemiPArametric LM (SPALM)

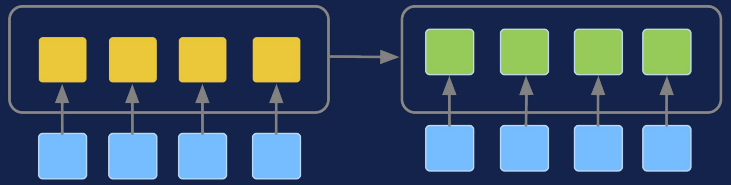
long-term memory



Value: output token for the respective context

USA is a beautiful	country

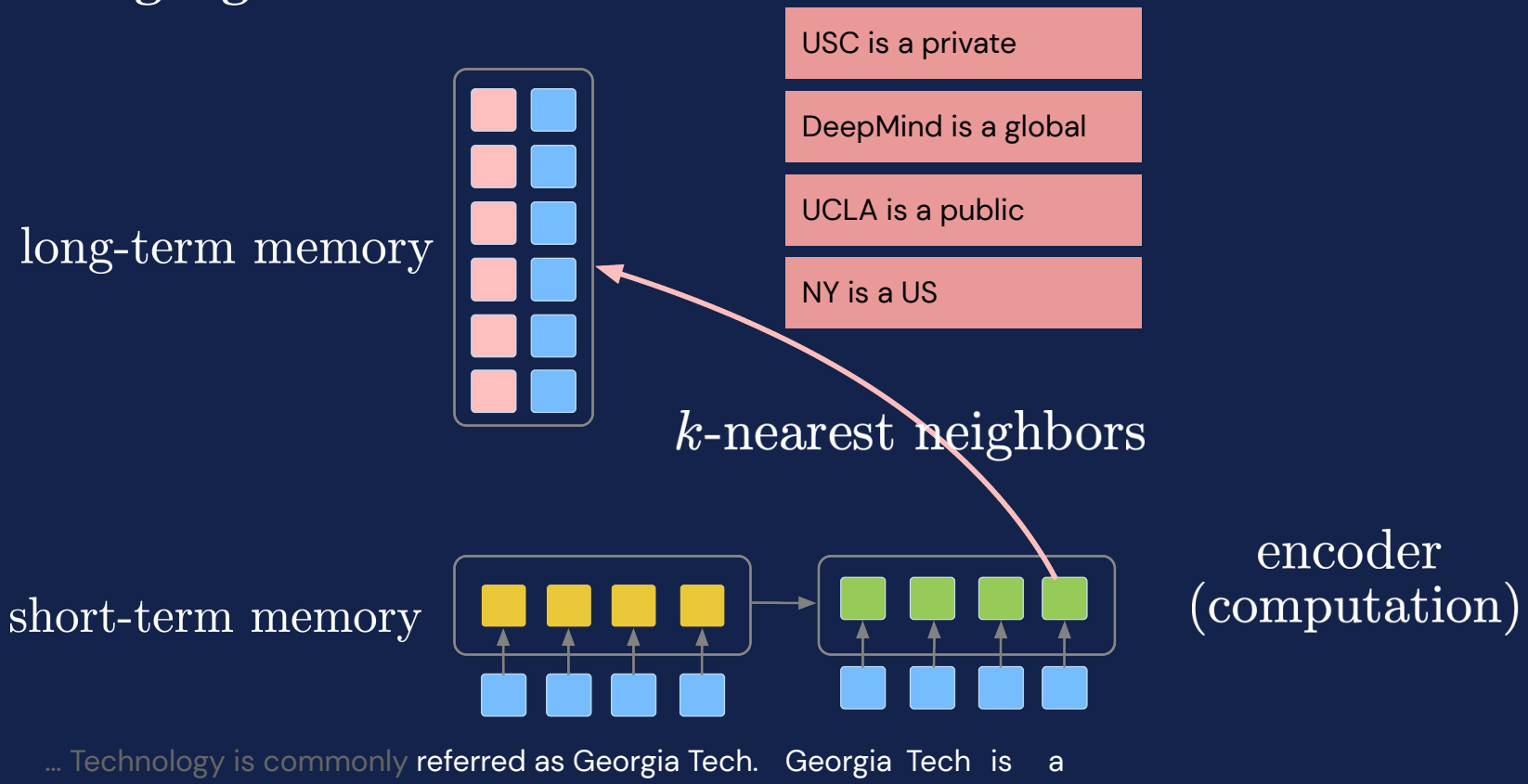
short-term memory



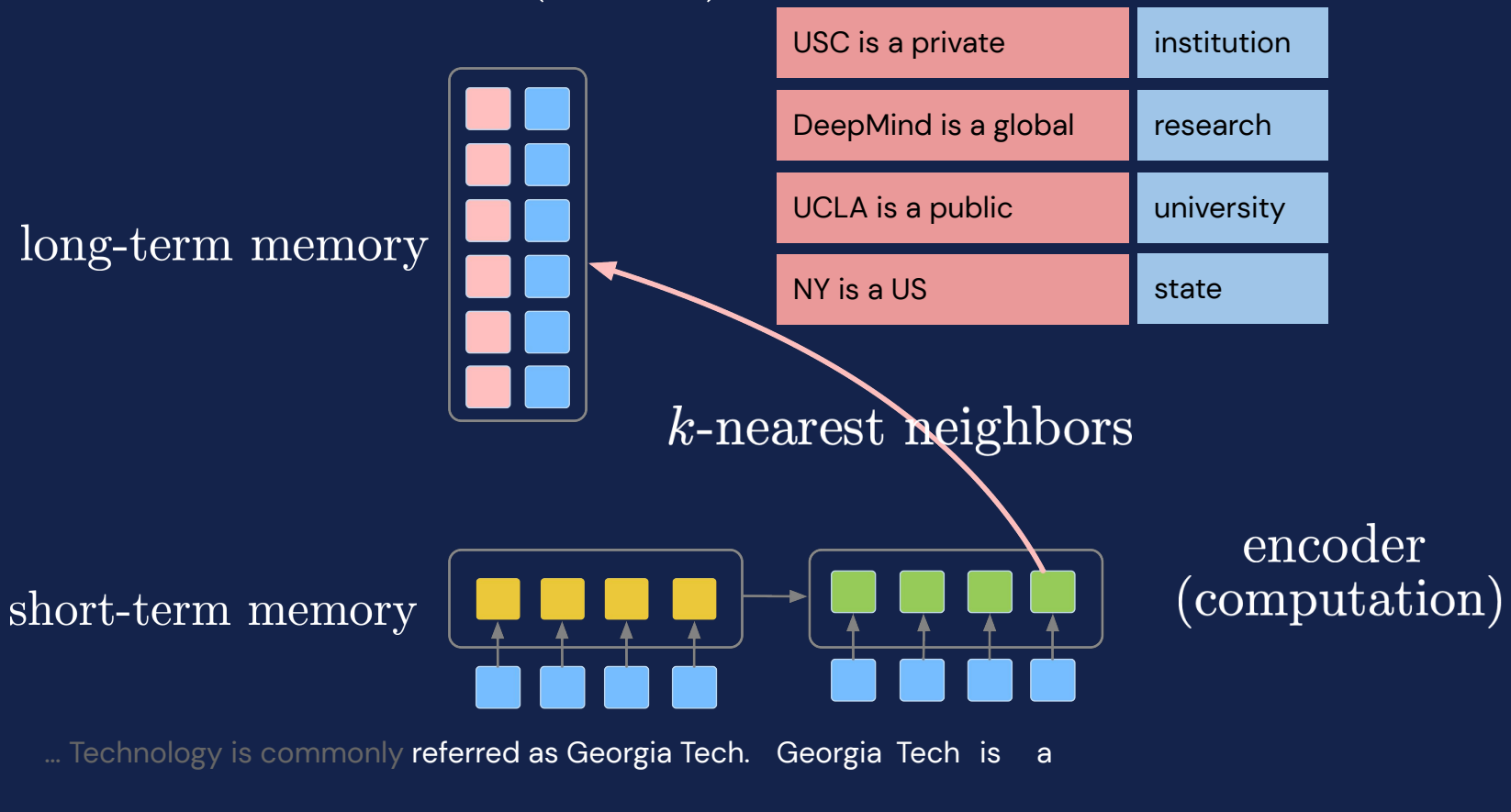
encoder (computation)

New York University is known as NYU. NYU is a private

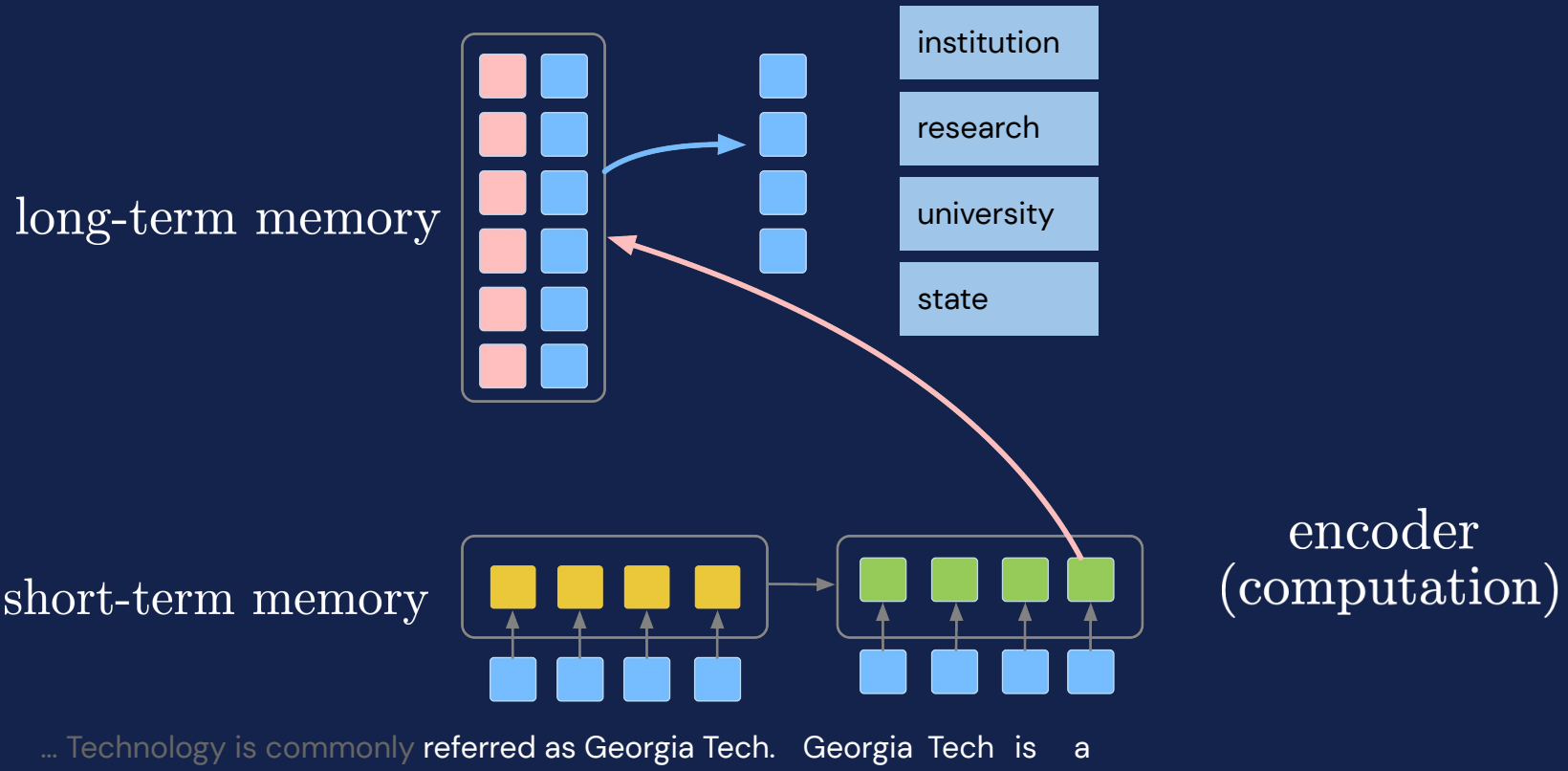
Language Model



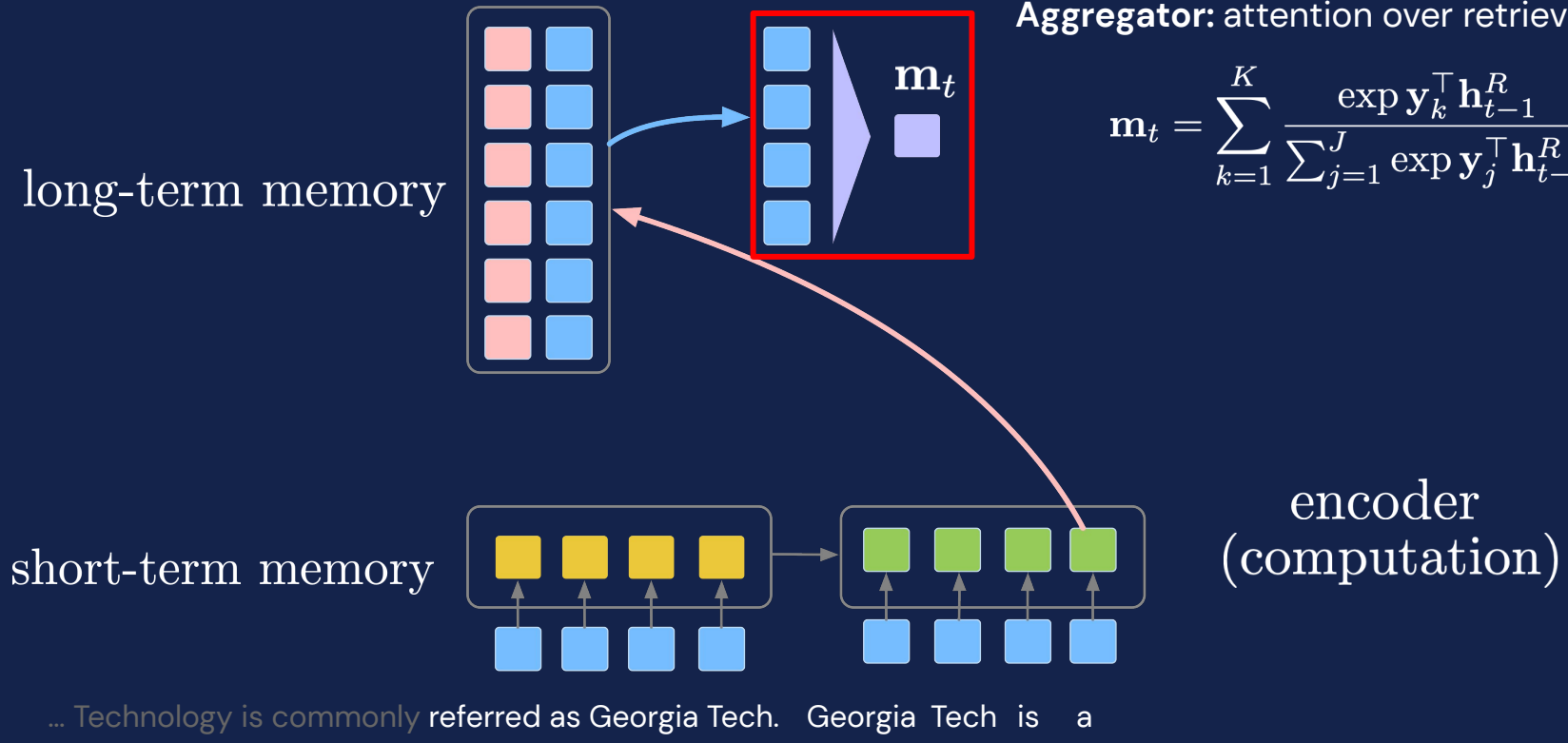
SemiPArametric LM (SPALM)



SemiPArametric LM (SPALM)



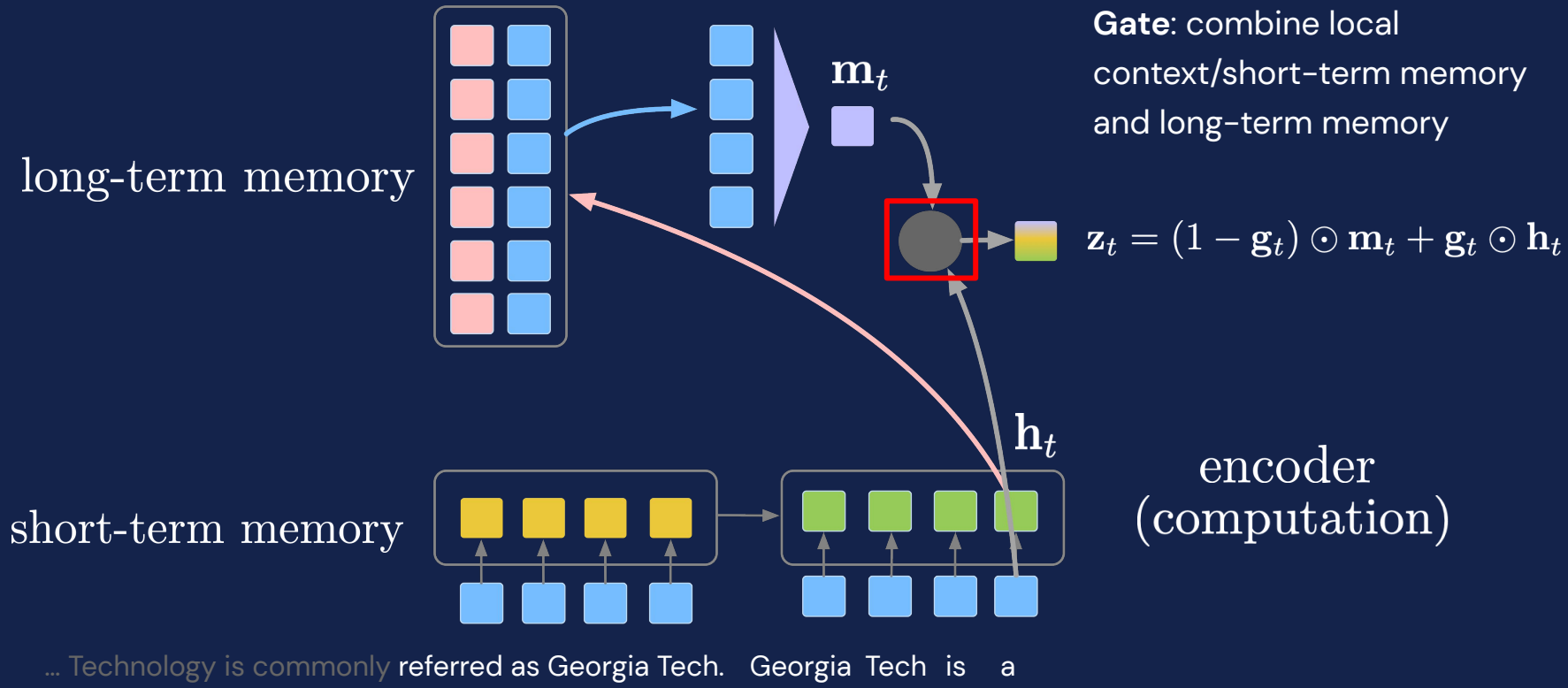
SemiPArametric LM (SPALM)



Aggregator: attention over retrieved values

$$\mathbf{m}_t = \sum_{k=1}^{K} \frac{\exp \mathbf{y}_k^\top \mathbf{h}_{t-1}^R}{\sum_{j=1}^{J} \exp \mathbf{y}_j^\top \mathbf{h}_{t-1}^R} \mathbf{y}_k$$

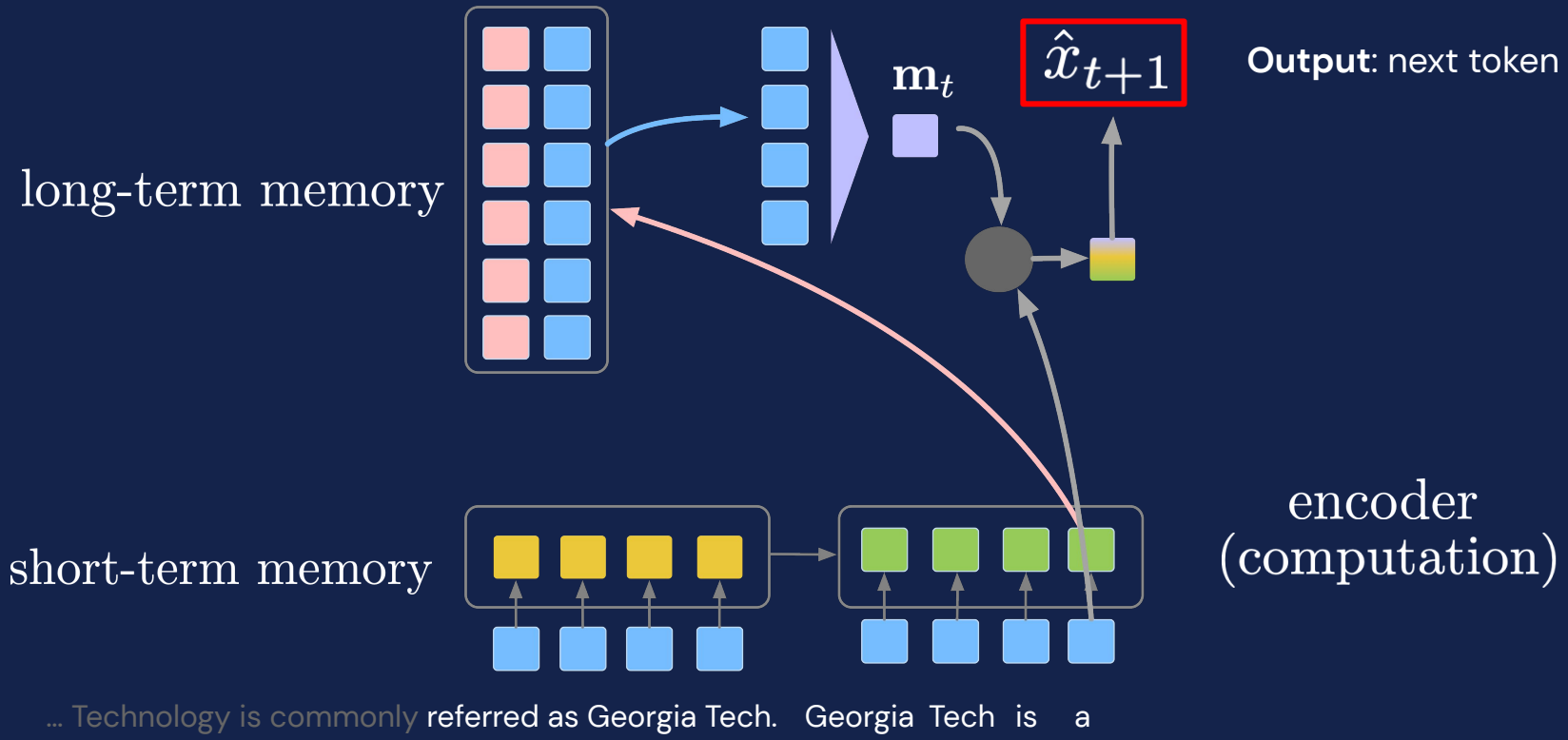
SemiPArametric LM (SPALM)



Gate: combine local context/short-term memory and long-term memory

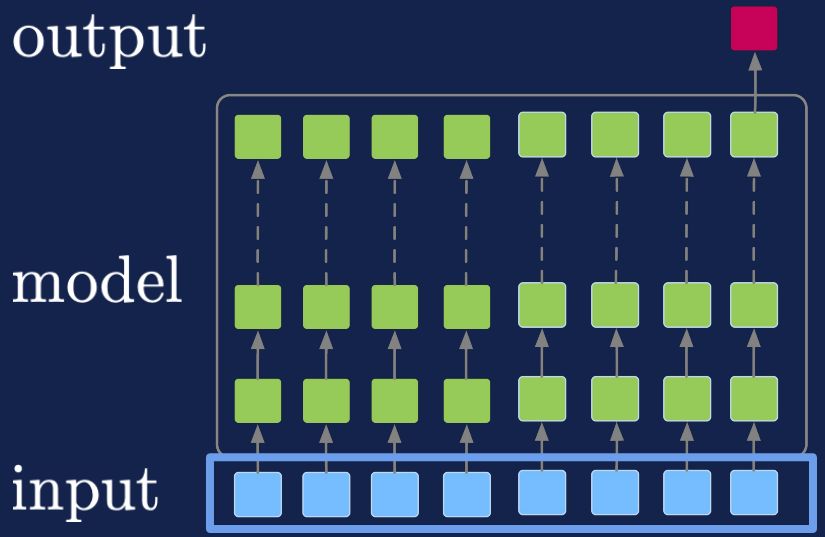
$$\mathbf{z}_t = (1 - \mathbf{g}_t) \odot \mathbf{m}_t + \mathbf{g}_t \odot \mathbf{h}_t$$

SemiPArametric LM (SPALM)

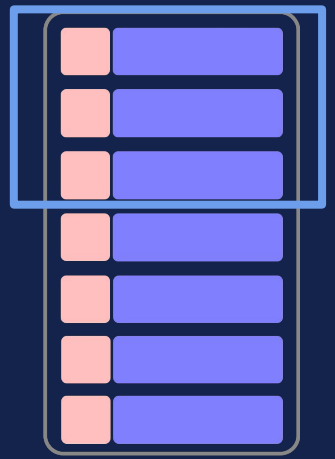


Semiparametric Models

Separation of computation and storage as an architectural bias.



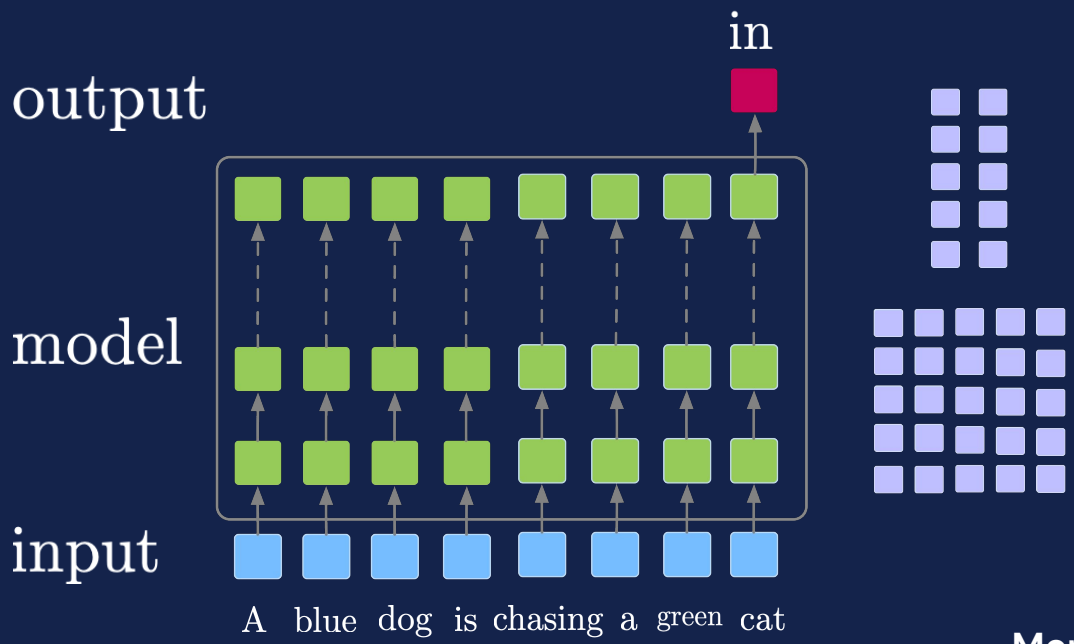
Computation: a neural network



Memory (storage): a key-value database

Nematzadeh, Ruder, Yogatama, ICLR Bridging AI and Cognitive Science Workshop 2020

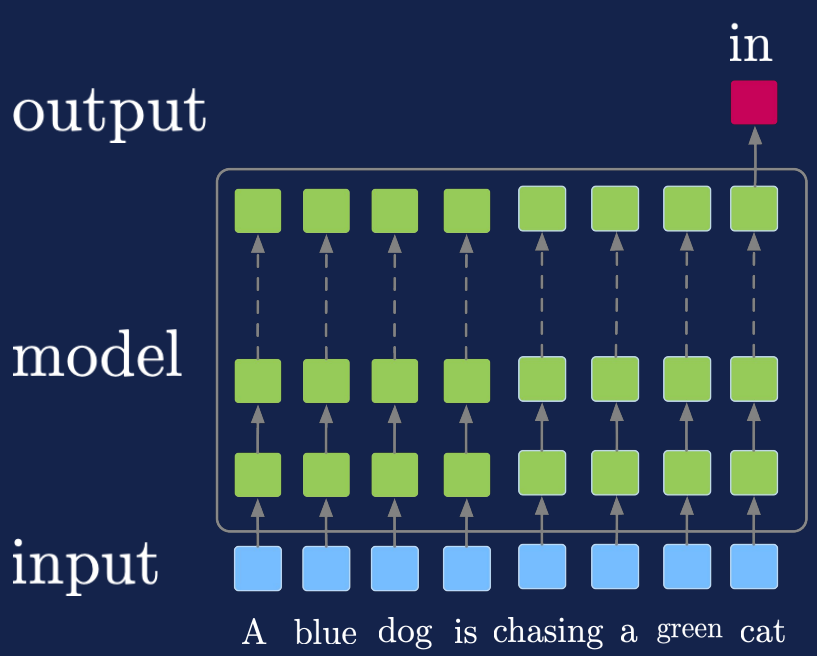
Semiparametric Language Models

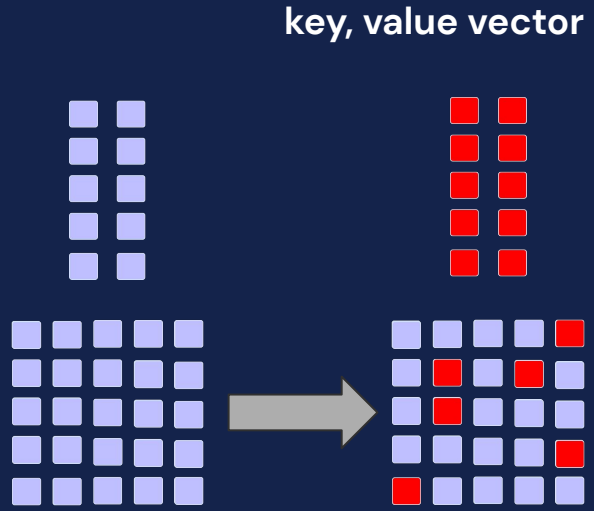


key, value vector pairs

Memory (storage): associative memory

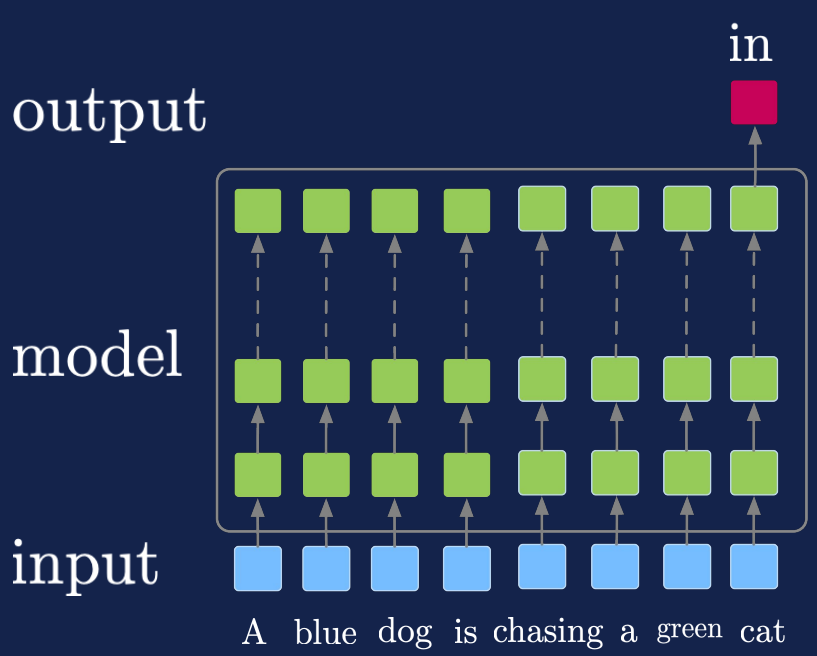
Semiparametric Language Models

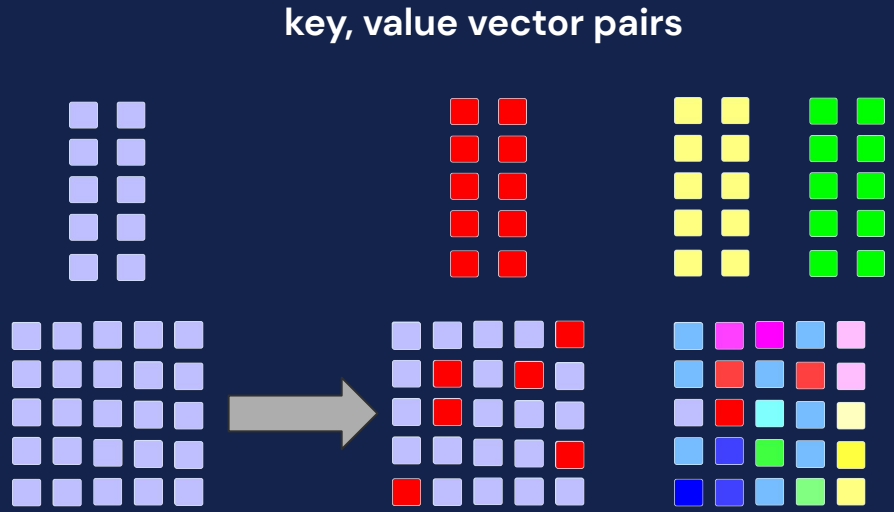




Memory (storage): associative memory

Semiparametric Language Models





Memory (storage): associative memory

Semiparametric Models

Random Feature Attention

Peng, Pappas, Yogatama, Schwartz, Smith, Kong, ICLR 2021

ABC: Attention with Bounded Memory Control

Peng, Kasai, Pappas, Yogatama, Wu, Kong, Schwartz, Smith, ACL 2022

The Road Ahead

Goal: emergent abilities at much smaller scale.

What kind of models?

How to use these models in practice?

The Road Ahead

A *foundation* model needs to be large since it is intended to be a generalist agent.

The Road Ahead

A *foundation* model needs to be large since it is intended to be a generalist agent.

Task-specific models are specialist agents.

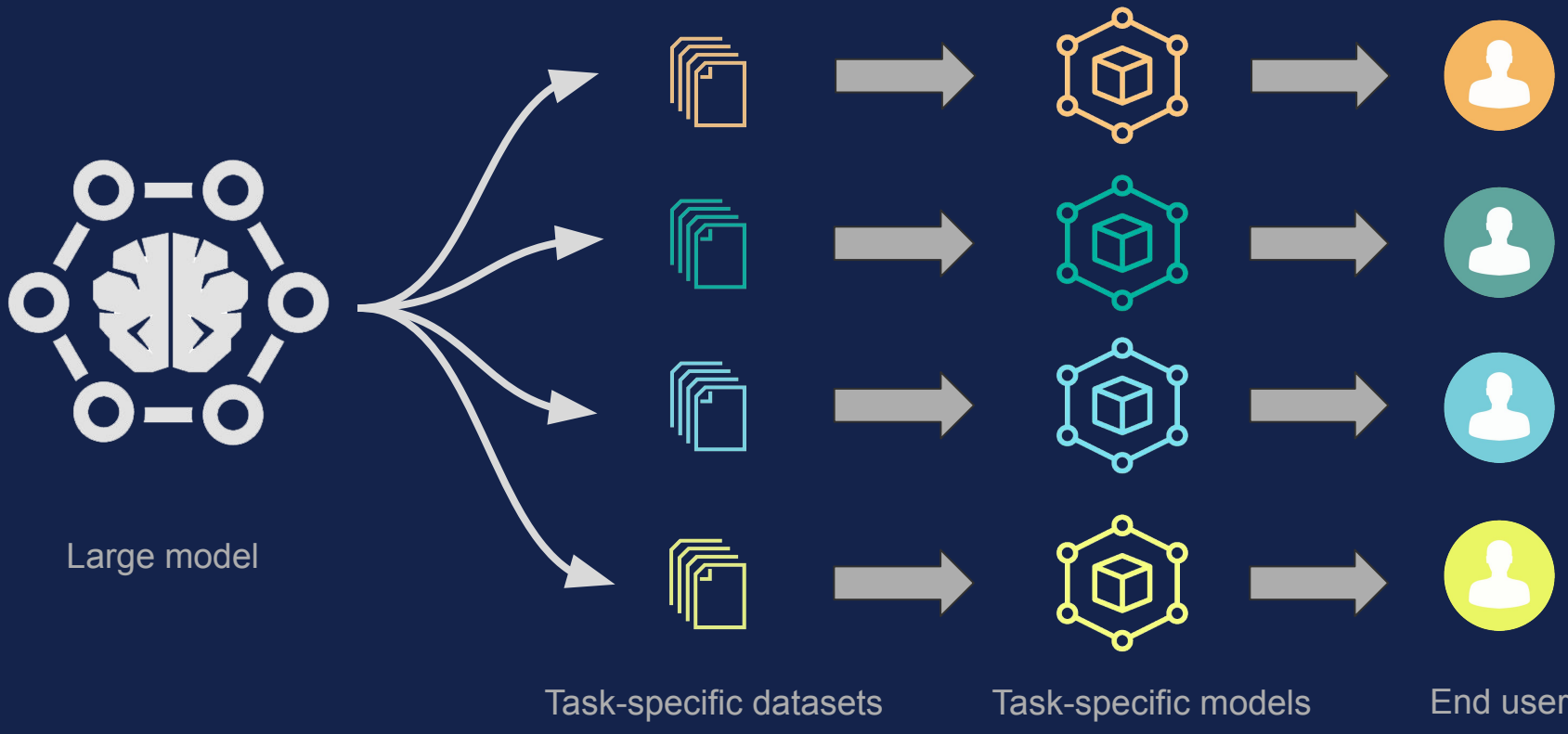
The Road Ahead

A *foundation* model needs to be large since it is intended to be a generalist agent.

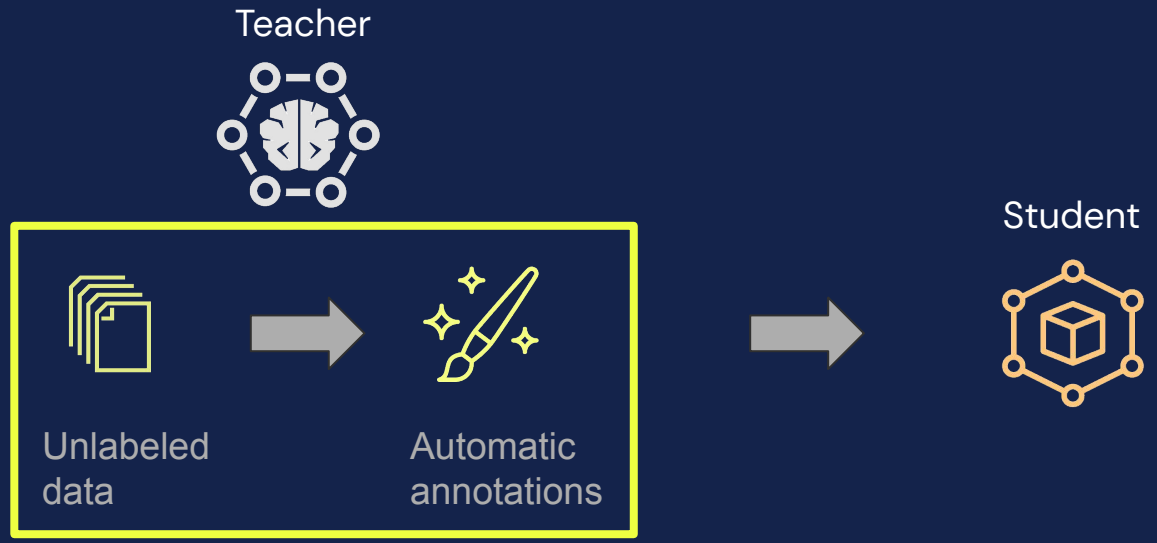
Task-specific models are specialist agents.

Serving very large models at inference time comes with cost, latency, and privacy issues.

Compression via Knowledge Distillation



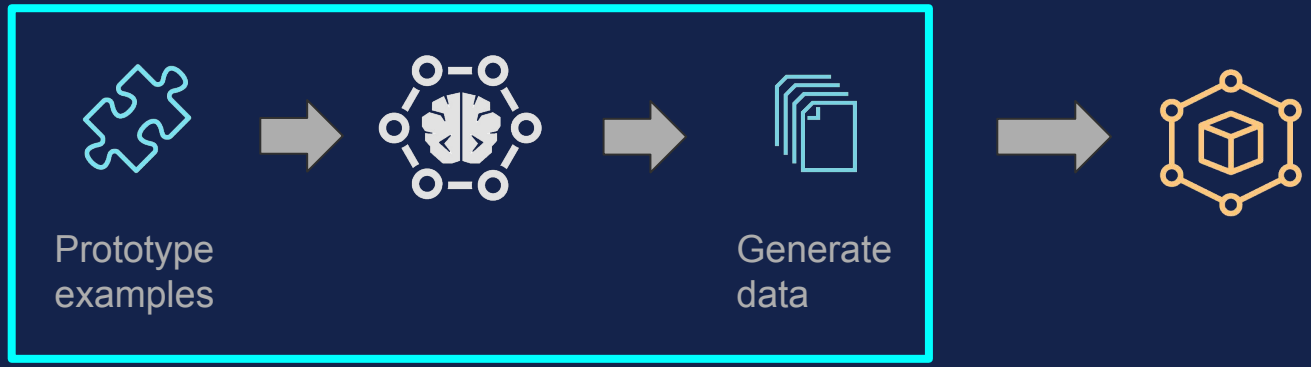
Compression via Knowledge Distillation



Synthetic annotation for automatically labeling data.

Distilling the Knowledge in a Neural Network, Hinton et al., 2015

Compression via Knowledge Distillation



Symbolic distillation.

A mechanism for scaling data.

Symbolic Knowledge Distillation: from General Language Models to Commonsense Models. West et al., 2021

Concluding Remarks

Scaling model parameters up lead to emergent abilities.

Concluding Remarks

Scaling model parameters up lead to emergent abilities.

There are other ways to get these emergent abilities.

Concluding Remarks

Scaling model parameters up lead to emergent abilities.

There are other ways to get these emergent abilities.

Semiparametric (memory-augmented) networks.

Concluding Remarks

Scaling model parameters up lead to emergent abilities.

There are other ways to get these emergent abilities.

Semiparametric (memory-augmented) networks.

Compression and knowledge distillation.

https://dyogatama.github.io

https://reka.ai/ (hiring full time researchers, engineers, and interns)

yogatama@usc.edu, dani@reka.ai