What do We Need to Learn Before Burning the Next One Million GPU Hours?

Iz Beltagy
Allen Institute for AI

Julien Launay
LightOn

What we did for the past year in architecture & scaling…

What Language Model to Train if You Have One Million GPU Hours?
Le Scao et al. (2022).

What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization?
Wang et al. (ICML 2022).

…but this talk is about what we learned and what open questions we still need to answer?

Overview

Evaluation

Architecture and Pretraining Objective

Scaling

Datasets

Engineering

Efficient Pretraining

Overview

Evaluation

Architecture and Pretraining Objective

Scaling

Datasets

Engineering

Efficient Pretraining

Evaluation

What evaluation setup to use to make modeling decisions?

Many settings

- LM loss vs. downstream
- Zero-shot vs. few-shot
- Prompting vs. finetuning
- Parameter-efficient vs. full finetuning
- With/without multi-task finetuning

Evaluation

What evaluation setup to use to make modeling decisions?

Many settings **and results don't necessarily transfer**

- LM loss vs. downstream (Tay et al., 2021, Abnar et. al., 2021)
- Zero-shot vs. few-shot
- Prompting vs. finetuning (Wang et al., 2022)
- Parameter-efficient vs. full finetuning
- With/without multi-task finetuning (Wang et al., 2022)

Tay et. al., 2021: Scale Efficiently: Insights From Pre-trained and Fine-tuned Transformers
Abnar et. al., 2021: Exploring the Limits of Large Scale Pre-training
Wang et. al., 2022: What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization?

Evaluation

What evaluation setup to use to make modeling decisions?

Many settings **and results don't necessarily transfer**

- LM loss vs. **downstream** (Tay et al., 2022, Abnar et. al., 2021)
- **Zero-shot** vs. few-shot
- **Prompting** vs. finetuning (Wang et al., 2022)
- ~~Parameter-efficient vs. full finetuning~~
- **With/without multi-task finetuning** (Wang et al., 2022)

Tay et. al., 2021: Scale Efficiently: Insights From Pre-trained and Fine-tuned Transformers
Abnar et. al., 2021: Exploring the Limits of Large Scale Pre-training
Wang et. al., 2022: What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization?

Evaluation

What evaluation setup to use to make modeling decisions?

Many settings **and results don't necessarily transfer**

- LM loss vs. **downstream** (Tay et al., 2022, Abnar et. al., 2021)
- **Zero-shot** vs. few-shot
- **Prompting** vs. finetuning (Wang et al., 202
- ~~Parameter-efficient vs. full finetuning~~
- **With/without multi-task finetuning** (Wang

- Fast and easy to run

- Representative of how the model will be used in practice

Tay et. al., 2021: Scale Efficiently: Insights From Pre-trained and Fine-tuned Transformers
Abnar et. al., 2021: Exploring the Limits of Large Scale Pre-training
Wang et. al., 2022: What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization?

Evaluation

What evaluation setup to use to make modeling decisions?

Open question

- Better understanding of the relation between different setup, and why the results differ

Overview

Evaluation

Architecture and Pretraining Objective

Scaling

Datasets

Engineering

Efficient Pretraining

Architecture and Pretraining Objective

Architecture and Pretraining Objective

What is a model architecture and pretraining objective that work for all settings?

Architecture and Pretraining Objective

What is a model architecture and pretraining objective that work for all settings?

Architecture and Pretraining Objective

What is a model architecture and pretraining objective that work for all settings?

Wang et al., 2022: zero-shot generalization

Architecture and Pretraining Objective

What is a model architecture and pretraining objective that work for all settings?

Wang et al., 2022: zero-shot generalization

- GPT2-style: after pretraining

Architecture and Pretraining Objective

What is a model architecture and pretraining objective that work for all settings?

Wang et al., 2022: zero-shot generalization

- GPT2-style: after pretraining
- T5-style: after pretraining + multitask finetuning (e.g. T0 adaptation)

Architecture and Pretraining Objective

What is a model architecture and pretraining objective that work for all settings?

Wang et al., 2022: zero-shot generalization

- GPT2-style: after pretraining
- T5-style: after pretraining + multitask finetuning (e.g. T0 adaptation)

Architecture and Pretraining Objective

What is a model architecture and pretraining objective that work for all settings?

Wang et al., 2022: zero-shot generalization

- GPT2-style: after pretraining
- T5-style: after pretraining + multitask finetuning (e.g. T0 adaptation)

Architecture and Pretraining Objective

What is a model architecture and pretraining objective that work for all settings?

Wang et al., 2022: zero-shot generalization

- GPT2-style: after pretraining
- T5-style: after pretraining + multitask finetuning (e.g. T0 adaptation)

Architecture and Pretraining Objective

What is a model architecture and pretraining objective that work for all settings?

Wang et al., 2022: zero-shot generalization

- GPT2-style: after pretraining
- T5-style: after pretraining + multitask finetuning (e.g. T0 adaptation)

Architecture and Pretraining Objective

What is a model architecture and pretraining objective that work for all settings?

Wang et al., 2022: zero-shot generalization

- GPT2-style: after pretraining
- T5-style: after pretraining + multitask finetuning (e.g. T0 adaptation)

Wang et. al., 2022: What Language Model Architecture and Pretraining Objective Work Best for Zero-Shot Generalization?

Architectures

Architectures

Architectures:

Architectures

Architectures:

- Vanilla transformer

Architectures

Architectures:

- Vanilla transformer
  - Decoder-only

Architectures

Architectures:

- Vanilla transformer
  - Decoder-only
  - Encoder-decoder

Architectures

Architectures:

- Vanilla transformer
  - Decoder-only
  - Encoder-decoder
- Transformer variants

Architectures

Architectures:

- Vanilla transformer
  - Decoder-only
  - Encoder-decoder
- Transformer variants

Architectures

Architectures:

- Vanilla transformer
  - Decoder-only
  - Encoder-decoder
- Transformer variants

Narang et. al.,: most transformer variants are comparable to vanilla transformers.

Architectures

Architectures:

- Vanilla transformer
  - Decoder-only
  - Encoder-decoder
- Transformer variants

Narang et. al.,: most transformer variants are comparable to vanilla transformers.

Architectures

Architectures:

- Vanilla transformer
  - Decoder-only
  - Encoder-decoder
- Transformer variants

Narang et. al.,: most transformer variants are comparable to vanilla transformers.

Mixture of Experts (**MoEs**) or Conditionally-Activated Transformers:

Architectures

Architectures:

- Vanilla transformer
  - Decoder-only
  - Encoder-decoder
- Transformer variants

Narang et. al.,: most transformer variants are comparable to vanilla transformers.

Mixture of Experts (**MoEs**) or Conditionally-Activated Transformers:

- Different examples activate different parts (“experts”) of the model

Architectures

Architectures:

- Vanilla transformer
  - Decoder-only
  - Encoder-decoder
- Transformer variants

Narang et. al.,: most transformer variants are comparable to vanilla transformers.

Mixture of Experts (**MoEs**) or Conditionally-Activated Transformers:

- Different examples activate different parts (“experts”) of the model
- Examples: Switch Transformer, Base-layer, DEMix Layers, GLaM, LaMDa

Architectures

Architectures:

- Vanilla transformer
  - Decoder-only
  - Encoder-decoder
- Transformer variants

Narang et. al.,: most transformer variants are comparable to vanilla transformers.

Mixture of Experts (**MoEs**) or Conditionally-Activated Transformers:

- Different examples activate different parts (“experts”) of the model
- Examples: Switch Transformer, Base-layer, DEMix Layers, GLaM, LaMDa
- Interesting but haven’t seen the same adoption as dense models

Architectures

Architectures:

- Vanilla transformer
  - Decoder-only
  - Encoder-decoder
- Transformer variants

Narang et. al.,: most transformer variants are comparable to vanilla transformers.

Mixture of Experts (**MoEs**) or Conditionally-Activated Transformers:

- Different examples activate different parts (“experts”) of the model
- Examples: Switch Transformer, Base-layer, DEMix Layers, GLaM, LaMDa
- Interesting but haven’t seen the same adoption as dense models

Architectures

Architectures:

- Vanilla transformer
  - Decoder-only
  - Encoder-decoder
- Transformer variants

Narang et. al.,: most transformer variants are comparable to vanilla transformers.

Mixture of Experts (**MoEs**) or Conditionally-Activated Transformers:

- Different examples activate different parts (“experts”) of the model
- Examples: Switch Transformer, Base-layer, DEMix Layers, GLaM, LaMDa
- Interesting but haven’t seen the same adoption as dense models

Narang et. al., 2021: Do Transformer Modifications Transfer Across Implementations and Applications?

Pretraining Objective

Pretraining Objective

Mixture of objectives

Pretraining Objective

Mixture of objectives

- UL2: Tay et. al., 2022

Pretraining Objective

Mixture of objectives

- UL2: Tay et. al., 2022
- CM3: Aghajanyan et. al., 2022

Pretraining Objective

Mixture of objectives

- UL2: Tay et. al., 2022
- CM3: Aghajanyan et. al., 2022
- Argue that the objectives are complementary ⇒ train on a mix of them

Pretraining Objective

Mixture of objectives

- UL2: Tay et. al., 2022
- CM3: Aghajanyan et. al., 2022
- Argue that the objectives are complementary ⇒ train on a mix of them

Pretraining Objective

Mixture of objectives

- UL2: Tay et. al., 2022
- CM3: Aghajanyan et. al., 2022
- Argue that the objectives are complementary ⇒ train on a mix of them

Pretraining Objective

Mixture of objectives

- UL2: Tay et. al., 2022
- CM3: Aghajanyan et. al., 2022
- Argue that the objectives are complementary ⇒ train on a mix of them

Pretraining Objective

Mixture of objectives

- UL2: Tay et. al., 2022
- CM3: Aghajanyan et. al., 2022
- Argue that the objectives are complementary ⇒ train on a mix of them

Tay et. al., 2022: Unifying Language Learning Paradigms

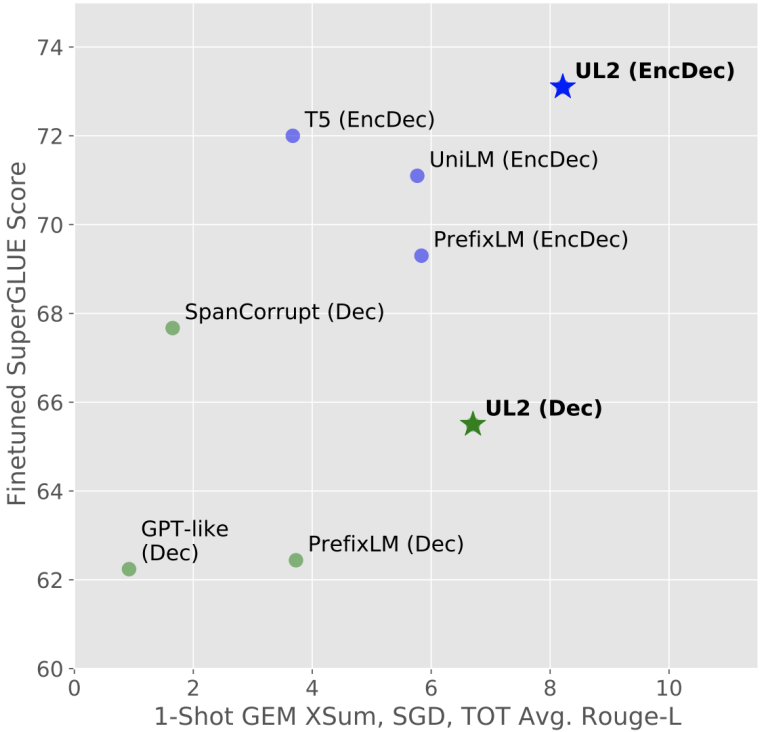
Pretraining Objective

Mixture of objectives

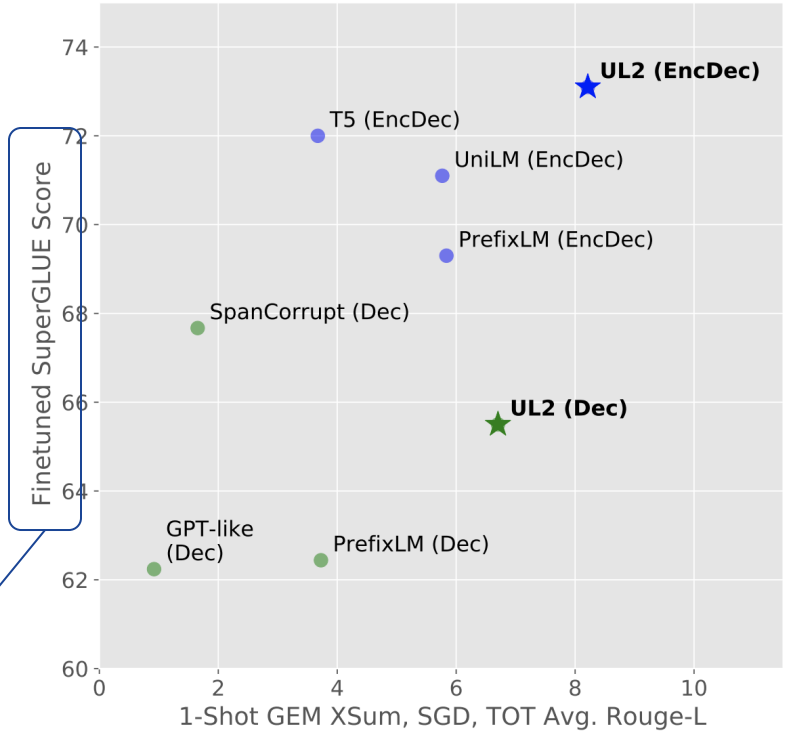
- UL2: Tay et. al., 2022
- CM3: Aghajanyan et. al., 2022
- Argue that the objectives are complementary ⇒ train on a mix of them

Tay et. al., 2022: Unifying Language Learning Paradigms
Aghajanyan et. al., 2022: CM3: A Causal Masked Multimodal Model of the Internet

Tay et. al., 2022: Unifying Language Learning Paradigms

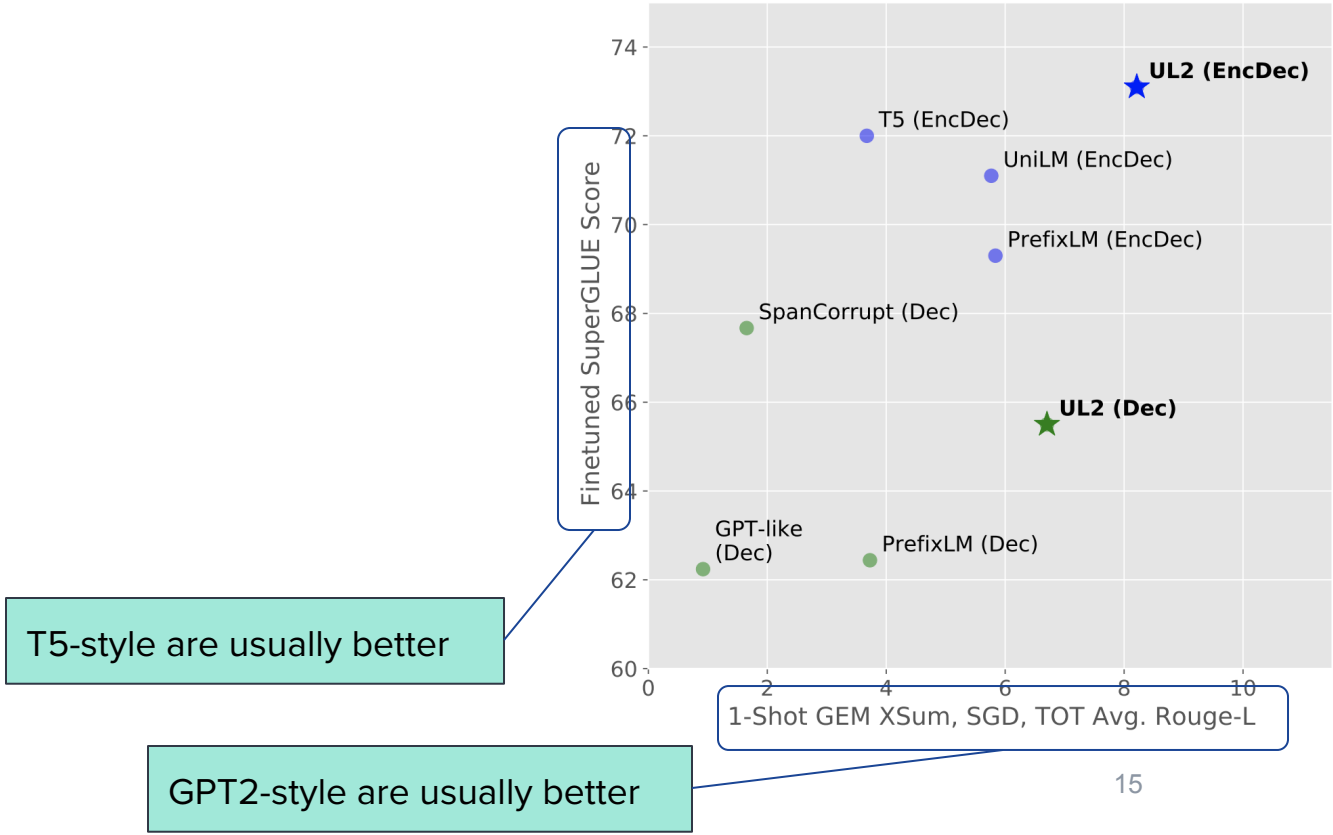


Tay et. al., 2022: Unifying Language Learning Paradigms



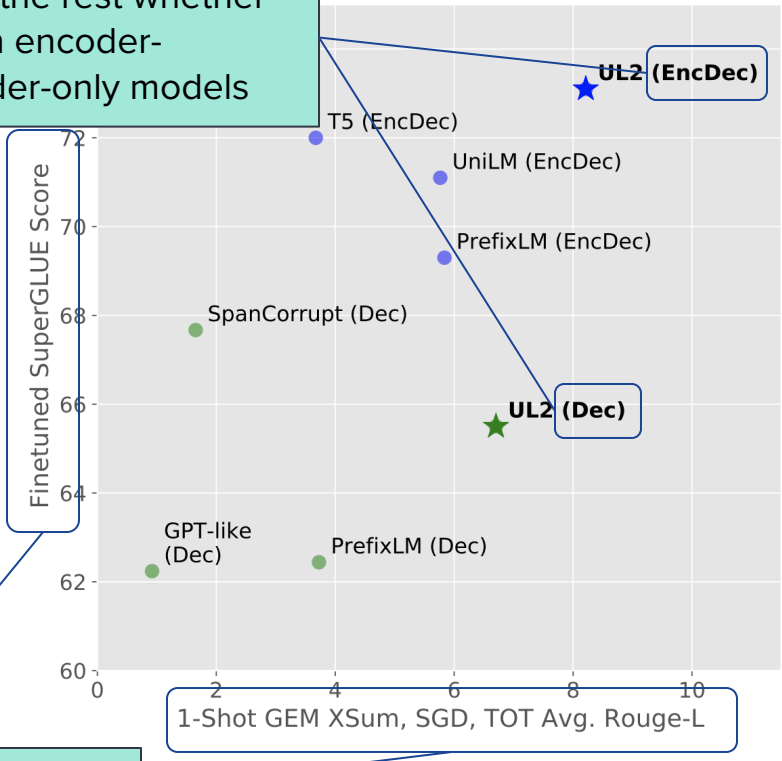
T5-style are usually better

Tay et. al., 2022: Unifying Language Learning Paradigms



Tay et. al., 2022: Unifying Language Learning Paradigms

UL2 outperforms the rest whether implemented with encoder-decoder or decoder-only models



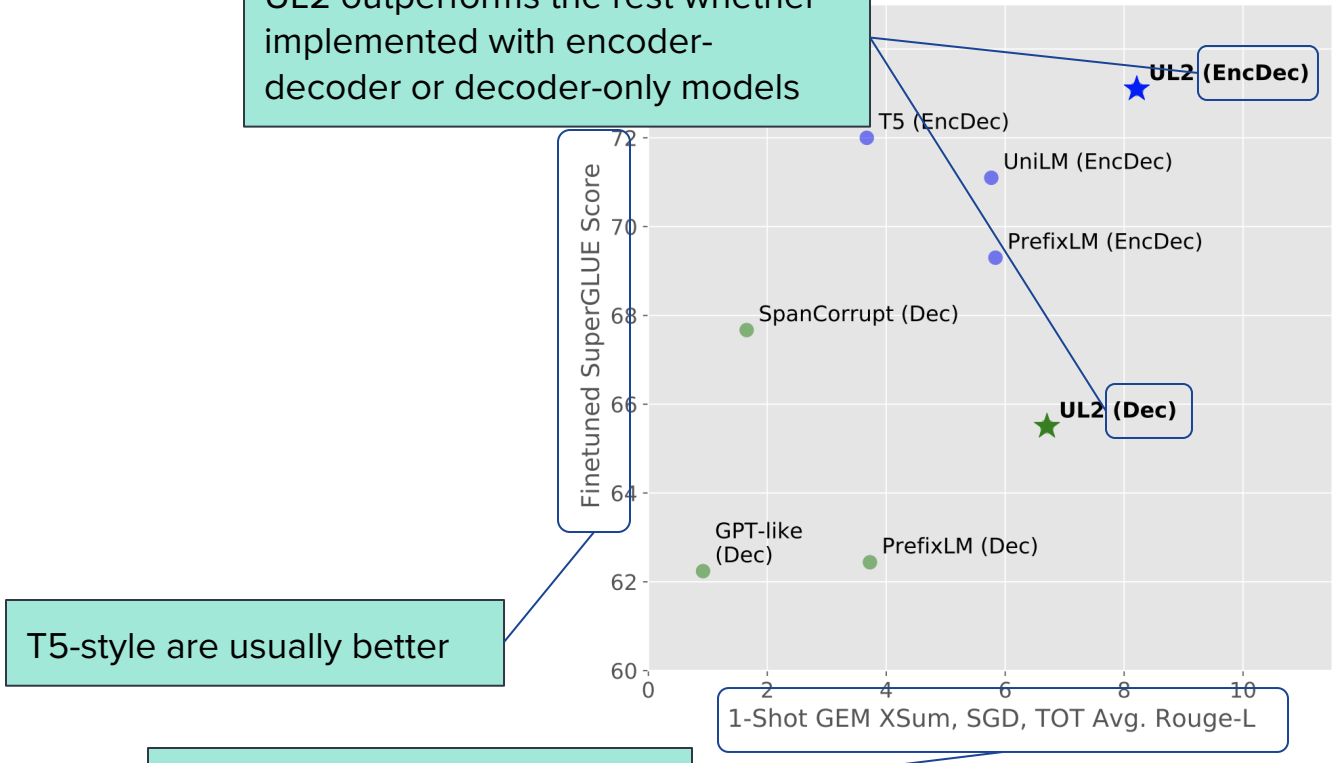
T5-style are usually better

GPT2-style are usually better

Tay et. al., 2022: Unifying Language Learning Paradigms

Limitations:

UL2 outperforms the rest whether implemented with encoder-decoder or decoder-only models



T5-style are usually better

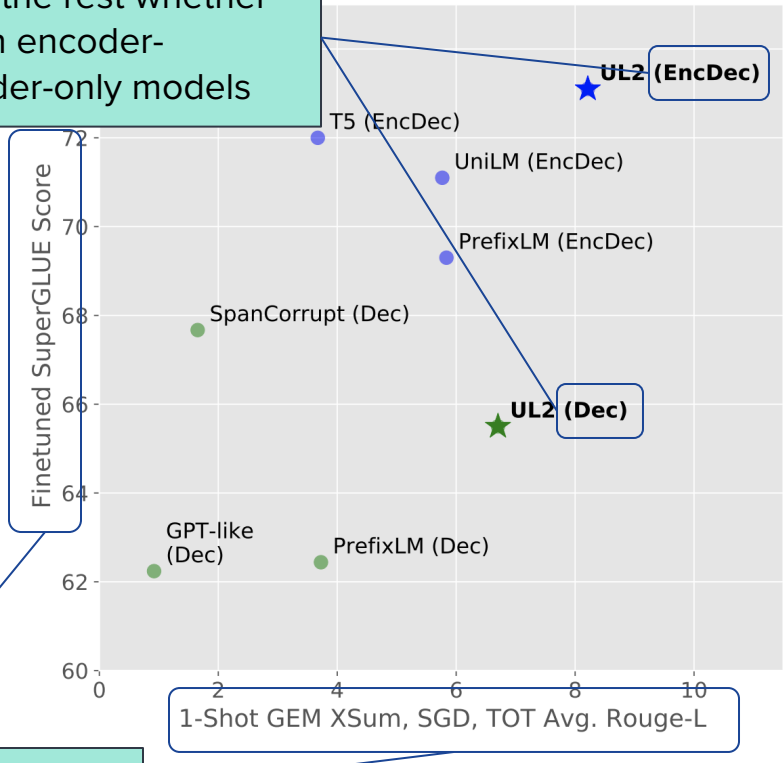
GPT2-style are usually better

Tay et. al., 2022: Unifying Language Learning Paradigms

Limitations:

- Missing the zero-shot evaluation with/without MT-F [as in Wang et. al., discussed earlier]

UL2 outperforms the rest whether implemented with encoder-decoder or decoder-only models



T5-style are usually better

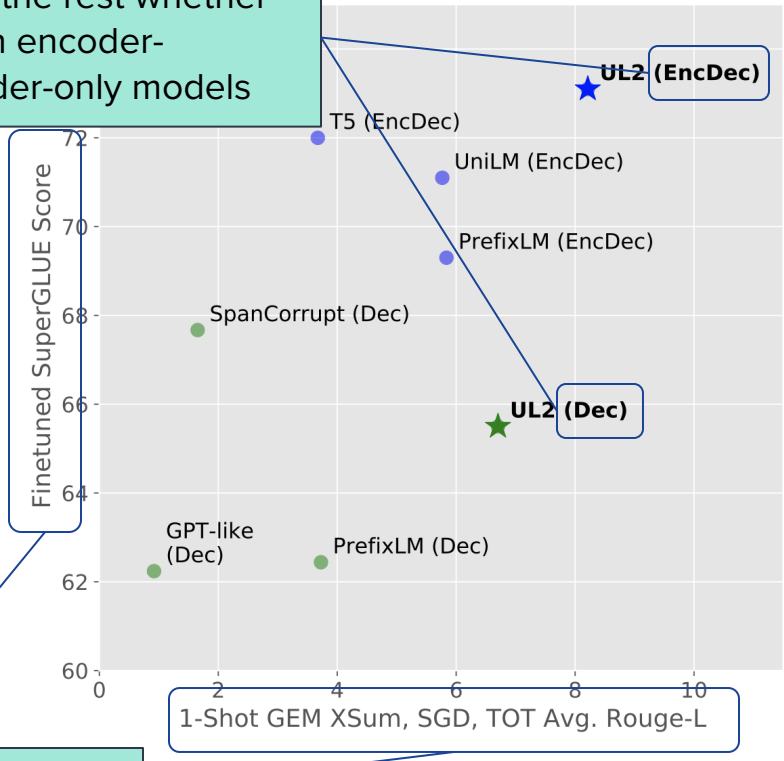
GPT2-style are usually better

Tay et. al., 2022: Unifying Language Learning Paradigms

Limitations:

- Missing the zero-shot evaluation with/without MT-F [as in Wang et. al., discussed earlier]
- Ignored causal LM
  - Claimed that Prefix LM is generally better

UL2 outperforms the rest whether implemented with encoder-decoder or decoder-only models



T5-style are usually better

GPT2-style are usually better

Architecture and Pretraining Objective

Architecture and Pretraining Objective

What is a model architecture and pretraining objective that work for all settings?

Architecture and Pretraining Objective

What is a model architecture and pretraining objective that work for all settings?

Architecture and Pretraining Objective

What is a model architecture and pretraining objective that work for all settings?

We don’t know, but we are getting closer

Overview

Evaluation

Architecture and Pretraining Objective

Scaling

Datasets

Engineering

Efficient Pretraining

Scaling - Kaplan et. al., 2020 vs. Hoffmann et. al., 2022

Scaling - Kaplan et. al., 2020 vs. Hoffmann et. al., 2022

Scaling - Kaplan et. al., 2020 vs. Hoffmann et. al., 2022

Kaplan et. al.,: grow model size **much faster** than tokens (e.g, GPT3, OPT, PaLM)

Scaling - Kaplan et. al., 2020 vs. Hoffmann et. al., 2022

Kaplan et. al.,: grow model size **much faster** than tokens (e.g, GPT3, OPT, PaLM)

- Given 10x compute, increase `N` by 5.5x, and `D` by 1.8x

Scaling - Kaplan et. al., 2020 vs. Hoffmann et. al., 2022

Kaplan et. al.,: grow model size **much faster** than tokens (e.g, GPT3, OPT, PaLM)

- Given 10x compute, increase `N` by 5.5x, and `D` by 1.8x

Scaling - Kaplan et. al., 2020 vs. Hoffmann et. al., 2022

Kaplan et. al.,: grow model size **much faster** than tokens (e.g, GPT3, OPT, PaLM)

- Given 10x compute, increase `N` by 5.5x, and `D` by 1.8x

Hoffmann et. al.,: grow model size and number of token **at the same rate** (e.g, Chinchilla)

Scaling - Kaplan et. al., 2020 vs. Hoffmann et. al., 2022

Kaplan et. al.,: grow model size **much faster** than tokens (e.g, GPT3, OPT, PaLM)

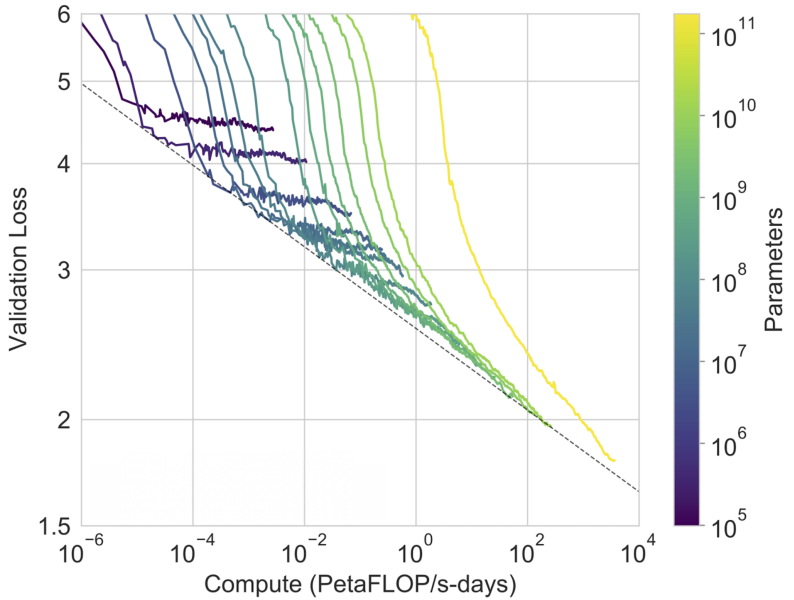
- Given 10x compute, increase N by 5.5x, and D by 1.8x

Hoffmann et. al.,: grow model size and number of token **at the same rate** (e.g, Chinchilla)

- Given 10x compute, grow N by 3.2x and D by 3.2x

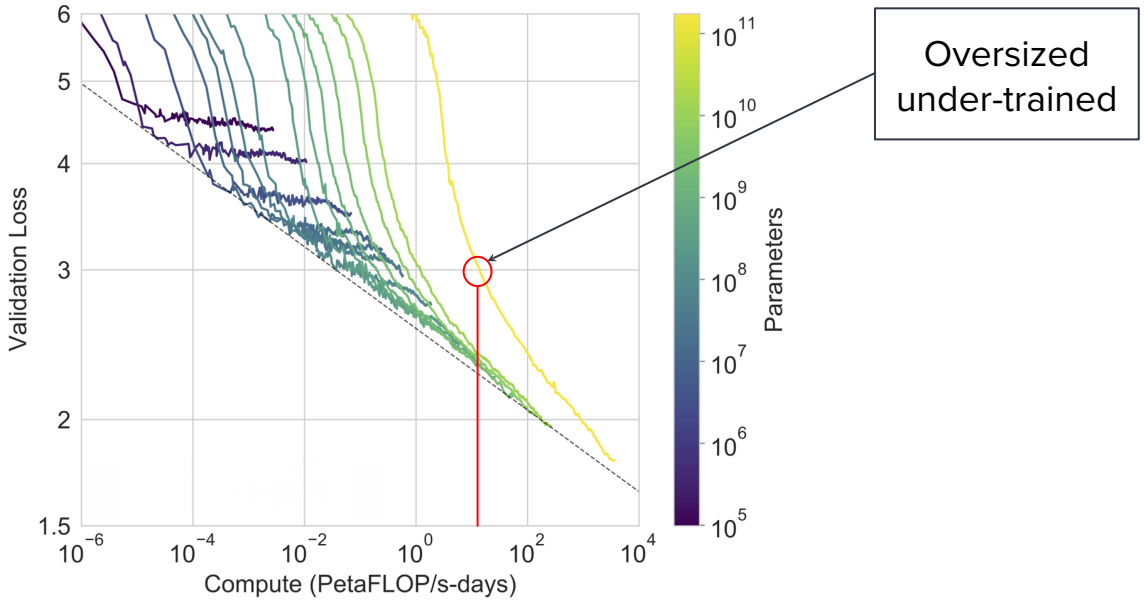
Scaling - Kaplan et. al., 2020 vs. Hoffmann et. al., 2022

- We trained (and are training) “oversized” and “under-trained” models



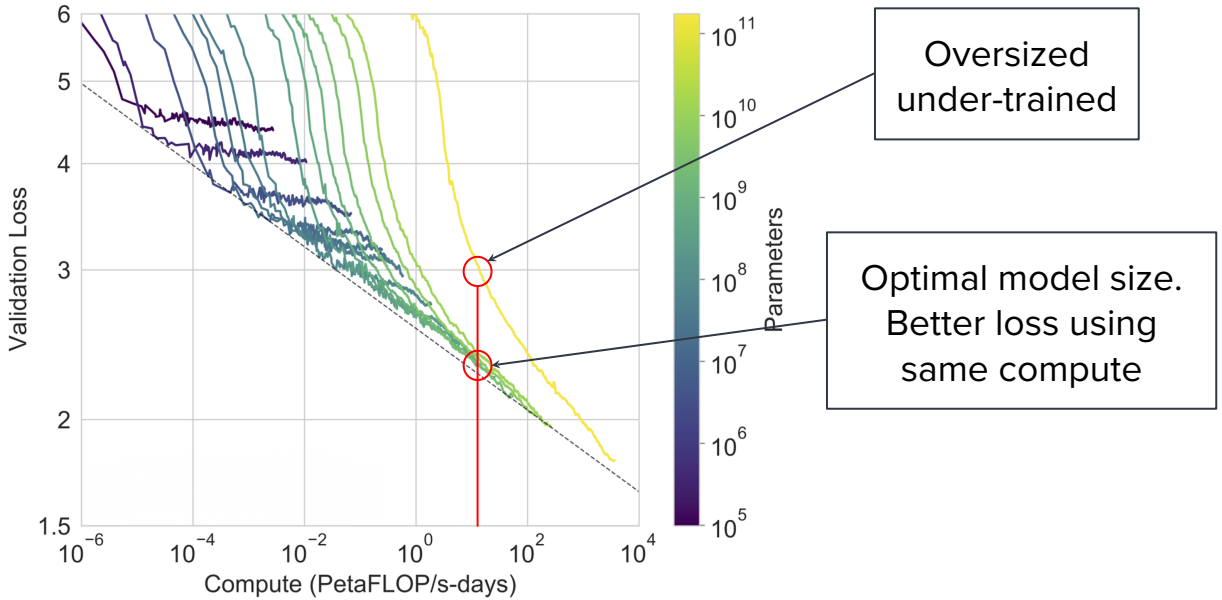
Scaling - Kaplan et. al., 2020 vs. Hoffmann et. al., 2022

- We trained (and are training) “oversized” and “under-trained” models



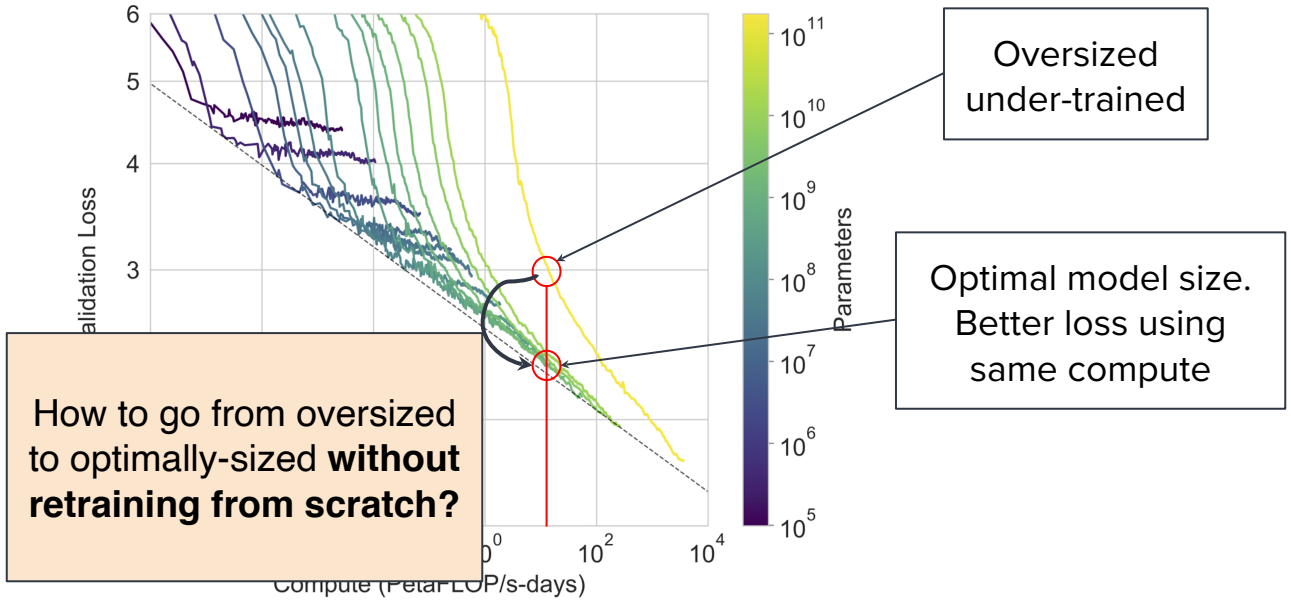
Scaling - Kaplan et. al., 2020 vs. Hoffmann et. al., 2022

- We trained (and are training) “oversized” and “under-trained” models



Scaling - Kaplan et. al., 2020 vs. Hoffmann et. al., 2022

- We trained (and are training) “oversized” and “under-trained” models



Chonk' me up Scotty

For the next generation of LLMs, we will need to scale...

Overview

Evaluation

Architecture and Pretraining Objective

Scaling

Datasets

Engineering

Efficient Pretraining

Chonk' me up Scotty

For the next generation of LLMs, we will need to scale...

datasets

quality at scale

Chonk' me up Scotty

For the next generation of LLMs, we will need to scale...

datasets

quality at scale

compute

engineering challenges

Chonk' me up Scotty

For the next generation of LLMs, we will need to scale...

datasets

quality at scale

compute

engineering challenges

modeling?

accelerate scaling

Model quality is all about data quality

Training data matters a lot!

(more than most modeling choices?)

Model quality is all about data quality

Training data matters a lot!

(more than most modeling choices?)

Aggregated performance on EAI harness

Model	Parameters	Pretraining tokens			
		Dataset	112B	250B	300B
OpenAI — Curie	6.7B				49.28
OpenAI — Babbage	1.3B				**45.30**
EleutherAI — GPT-Neo	1.3B	The Pile			42.94
Ours	13B	OSCAR			47.09
	1.3B	The Pile	**42.79**	43.12	43.46
	1.3B	C4	42.77		
	1.3B	OSCAR	41.72		

Le Scao et al.,2022.

Model quality is all about data quality

Training data matters a lot!

(more than most modeling choices?)

Aggregated performance on EAI harness

Model	Parameters	Pretraining tokens			
		Dataset	112B	250B	300B
OpenAI — Curie	6.7B				49.28
OpenAI — Babbage	1.3B				**45.30**
EleutherAI — GPT-Neo	1.3B	The Pile			42.94
Ours	13B	OSCAR			47.09
	1.3B	The Pile	**42.79**	43.12	43.46
	1.3B	C4	42.77		
	1.3B	OSCAR	41.72		

Le Scao et al.,2022.

Same architecture, different data:

Model quality is all about data quality

Training data matters a lot!

(more than most modeling choices?)

Aggregated performance on EAI harness

Model	Parameters	Pretraining tokens			
		Dataset	112B	250B	300B
OpenAI — Curie	6.7B				49.28
OpenAI — Babbage	1.3B				**45.30**
EleutherAI — GPT-Neo	1.3B	The Pile			42.94
Ours	13B	OSCAR			47.09
	1.3B	The Pile	**42.79**	43.12	43.46
	1.3B	C4	42.77		
	1.3B	OSCAR	41.72		

Le Scao et al.,2022.

Same architecture, different data:

45.30%

OpenAI-Babbage(1.3B)

Model quality is all about data quality

Training data matters a lot!

(more than most modeling choices?)

Aggregated performance on EAI harness

Model	Parameters	Pretraining tokens			
		Dataset	112B	250B	300B
OpenAI — Curie	6.7B				49.28
OpenAI — Babbage	1.3B				**45.30**
EleutherAI — GPT-Neo	1.3B	The Pile			42.94
	13B	OSCAR			47.09
Ours	1.3B	The Pile	**42.79**	43.12	43.46
	1.3B	C4	42.77		
	1.3B	OSCAR	41.72		

Le Scao et al.,2022.

Same architecture, different data:

45.30% OpenAI-Babbage(1.3B)

43.46% Ours-1.3B@The Pile

Model quality is all about data quality

Training data matters a lot!

(more than most modeling choices?)

Aggregated performance on EAI harness

Model	Parameters	Pretraining tokens			
		Dataset	112B	250B	300B
OpenAI — Curie	6.7B				49.28
OpenAI — Babbage	1.3B				**45.30**
EleutherAI — GPT-Neo	1.3B	The Pile			42.94
Ours	13B	OSCAR			47.09
	1.3B	The Pile	**42.79**	43.12	43.46
	1.3B	C4	42.77		
	1.3B	OSCAR	41.72		

Le Scao et al.,2022.

Scale can't compensate for bad data:

Model quality is all about data quality

Training data matters a lot!

(more than most modeling choices?)

Aggregated performance on EAI harness

Model	Parameters	Pretraining tokens			
		Dataset	112B	250B	300B
OpenAI — Curie	6.7B				49.28
OpenAI — Babbage	1.3B				**45.30**
EleutherAI — GPT-Neo	1.3B	The Pile			42.94
Ours	13B	OSCAR			47.09
	1.3B	The Pile	**42.79**	43.12	43.46
	1.3B	C4	42.77		
	1.3B	OSCAR	41.72		

Le Scao et al.,2022.

Scale can't compensate for bad data:

49.28%

OpenAI-Curie(6.7B)

Model quality is all about data quality

Training data matters a lot!

(more than most modeling choices?)

Aggregated performance on EAI harness

Model	Parameters	Pretraining tokens			
		Dataset	112B	250B	300B
OpenAI — Curie	6.7B				49.28
OpenAI — Babbage	1.3B				**45.30**
EleutherAI — GPT-Neo	1.3B	The Pile			42.94
	13B	OSCAR			47.09
Ours	1.3B	The Pile	**42.79**	43.12	43.46
	1.3B	C4	42.77		
	1.3B	OSCAR	41.72		

Le Scao et al.,2022.

Scale can't compensate for bad data:

49.28%

OpenAI-Curie(6.7B)

47.09%

Ours-13B@OSCAR

We are gonna need a bigger dataset!

Bad news: we need a lot more data than expected...

We are gonna need a bigger dataset!

Bad news: we need a lot more data than expected... 

Previously... Kaplan et al., 2020

176B parameters ⟶ 300B tokens

We are gonna need a bigger dataset!

Bad news: we need a lot more data than expected... 

Previously... Kaplan et al., 2020

176B parameters ⟶ 300B tokens

Now... Hoffmann et al., 2020

isoFLOPs 50B parameters ⟶ 1000B tokens

We are gonna need a bigger dataset!

Bad news: we need a lot more data than expected... 

Previously... Kaplan et al., 2020

176B parameters ⟶ 300B tokens

Now... Hoffmann et al., 2020

isoFLOPs 50B parameters ⟶ 1000B tokens

isoparams 176B parameters ⟶ 3700B tokens

~1 year of CC English

We are gonna need a bigger dataset!

Bad news: we need a lot more data than expected...

Previously... Kaplan et al., 2020

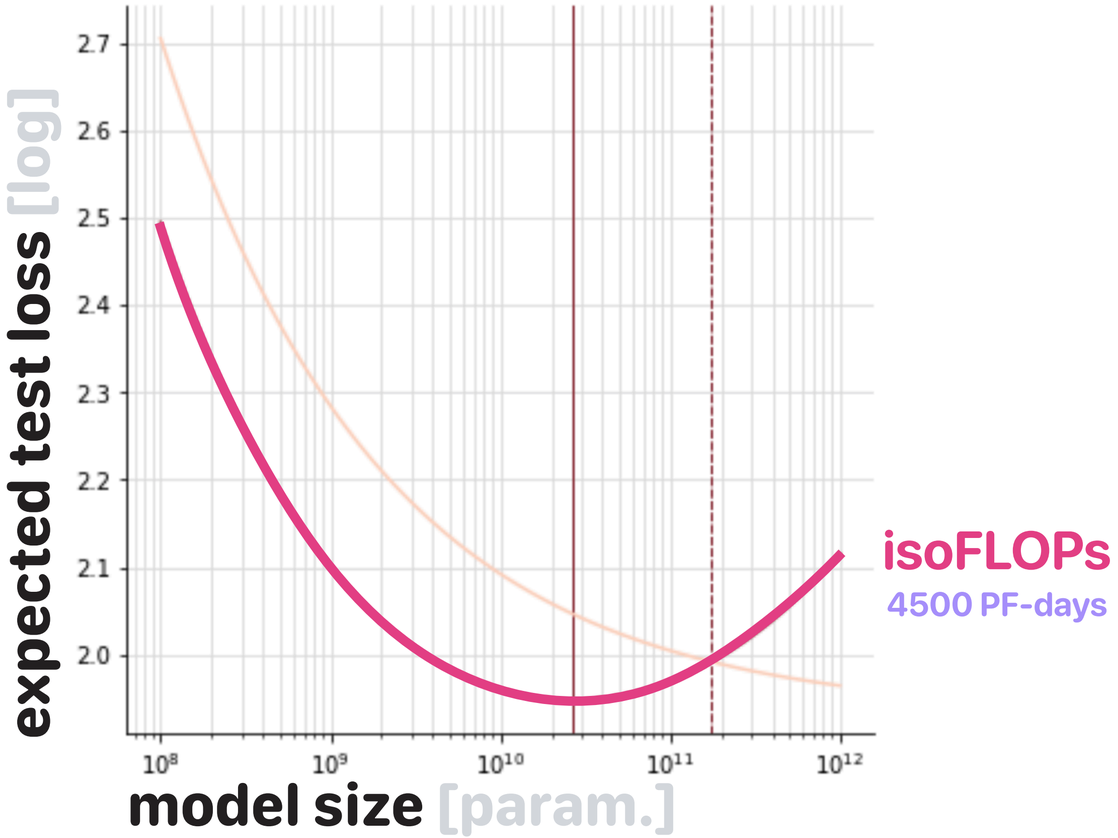
176B parameters → 300B tokens

Now... Hoffmann et al., 2020

isoFLOPs 50B parameters → 1000B tokens

isoparams 176B parameters → 3700B tokens

~1 year of CC English



We are gonna need a bigger dataset!

Bad news: we need a lot more data than expected...

Previously... Kaplan et al., 2020

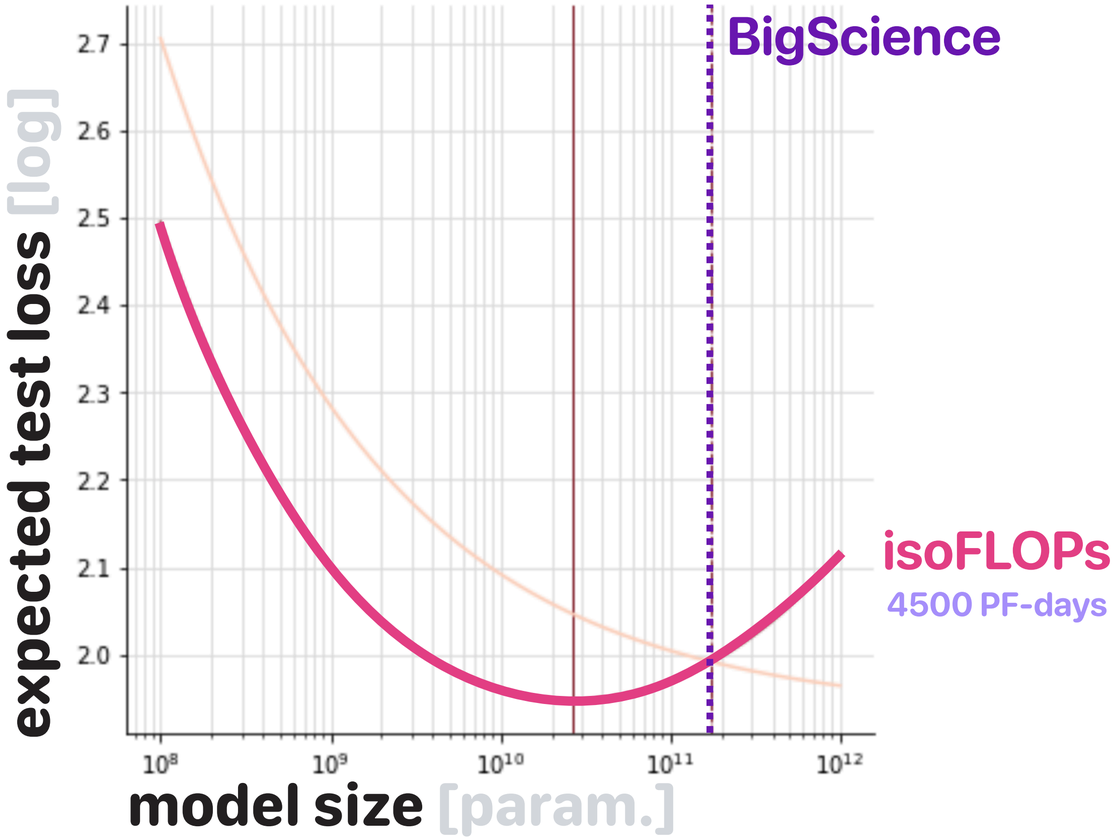
176B parameters → 300B tokens

Now... Hoffmann et al., 2020

isoFLOPs 50B parameters → 1000B tokens

isoparams 176B parameters → 3700B tokens

~1 year of CC English



We are gonna need a bigger dataset!

Bad news: we need a lot more data than expected...

Previously... Kaplan et al., 2020

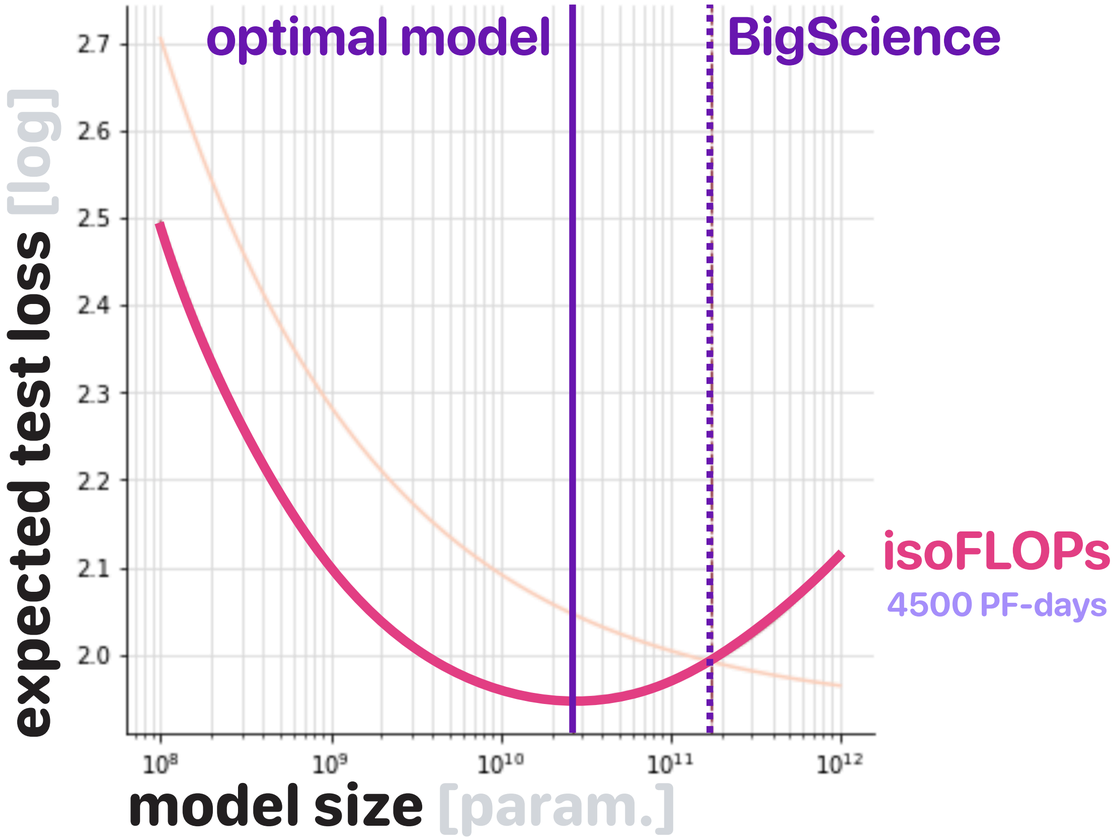
176B parameters → 300B tokens

Now... Hoffmann et al., 2020

isoFLOPs 50B parameters → 1000B tokens

isoparams 176B parameters → 3700B tokens

~1 year of CC English



We are gonna need a bigger dataset!

Bad news: we need a lot more data than expected...

Previously... Kaplan et al., 2020

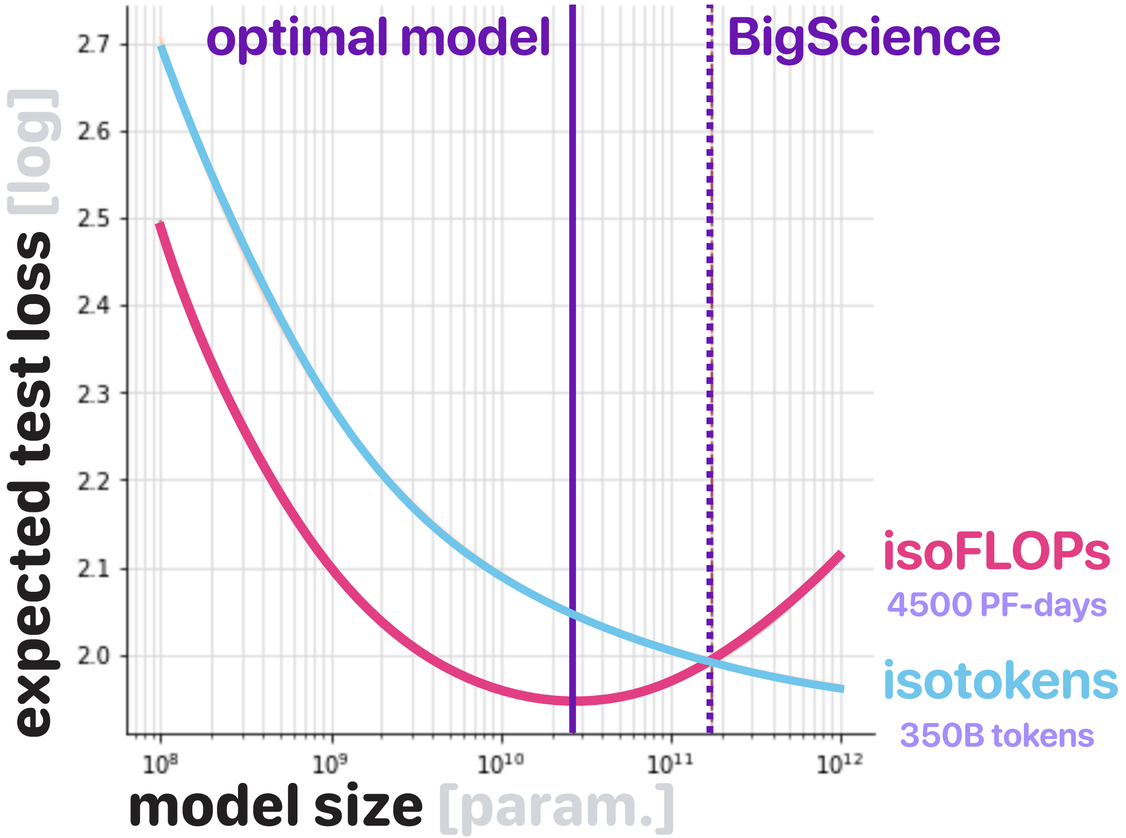
176B parameters → 300B tokens

Now... Hoffmann et al., 2020

isoFLOPs 50B parameters → 1000B tokens

isoparams 176B parameters → 3700B tokens

~1 year of CC English



We are gonna need a bigger dataset!

Bad news: we need a lot more data than expected...

Previously... Kaplan et al., 2020

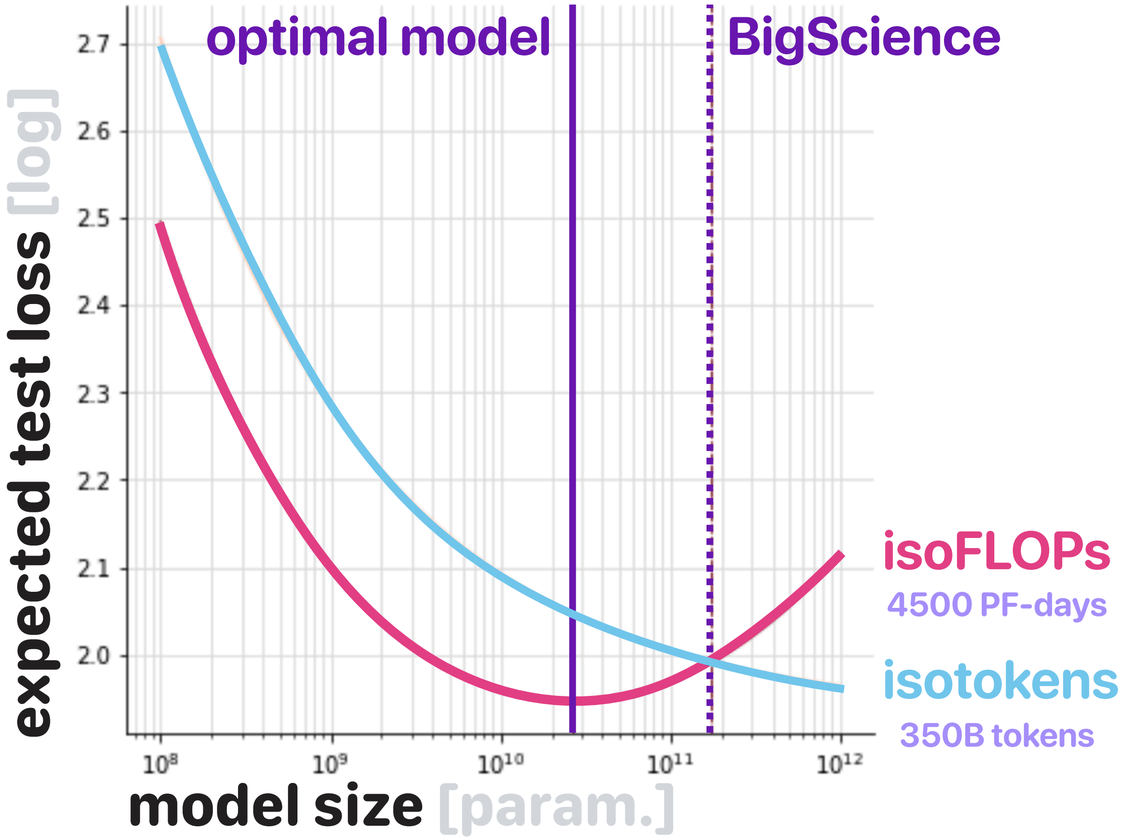
176B parameters → 300B tokens

Now... Hoffmann et al., 2020

isoFLOPs 50B parameters → 1000B tokens

isoparams 176B parameters → 3700B tokens

~1 year of CC English



Will we be data-bound instead of compute-bound?

Fantastic training data and where to find it

🔬 What even is high-quality data?

Fantastic training data and where to find it

🔬 What even is high-quality data? technical filtering deduplication, lack of artefacts, etc.

Fantastic training data and where to find it

🔬 What even is high-quality data?

technical filtering	deduplication, lack of artefacts, etc.
curation	diverse, cross-domain, etc.

Fantastic training data and where to find it

🔬 What even is high-quality data? technical filtering deduplication, lack of artefacts, etc.

curation diverse, cross-domain, etc.

"social media conversations"

Total dataset size = 780 billion tokens	
Data source	Proportion of data
Social media conversations (multilingual)	50%
Filtered webpages (multilingual)	27%
Books (English)	13%
GitHub (code)	5%
Wikipedia (multilingual)	4%
News (English)	1%

Chowdhery et al.,2022.

Fantastic training data and where to find it

🔬 What even is high-quality data? technical filtering deduplication, lack of artefacts, etc.

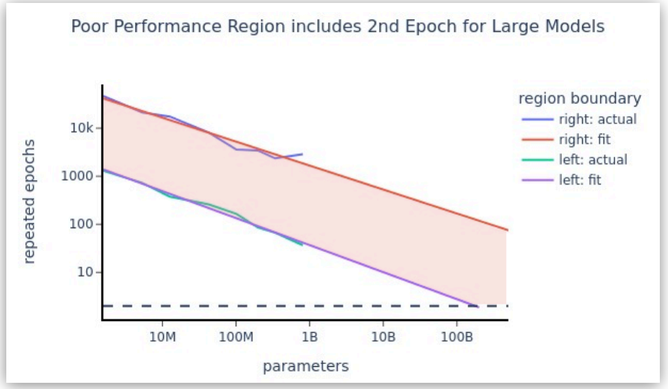
curation diverse, cross-domain, etc.

"social media conversations"

Total dataset size = 780 billion tokens	
Data source	Proportion of data
Social media conversations (multilingual)	50%
Filtered webpages (multilingual)	27%
Books (English)	13%
GitHub (code)	5%
Wikipedia (multilingual)	4%
News (English)	1%

Chowdhery et al.,2022.

double descent for duplication?



Hernandez et al.,2022.

Fantastic training data and where to find it

🔬 What even is high-quality data? technical filtering deduplication, lack of artefacts, etc.

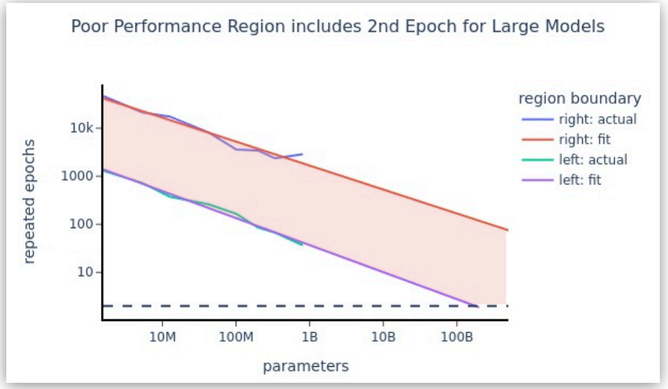
curation diverse, cross-domain, etc.

"social media conversations"

Total dataset size = 780 billion tokens	
Data source	Proportion of data
Social media conversations (multilingual)	50%
Filtered webpages (multilingual)	27%
Books (English)	13%
GitHub (code)	5%
Wikipedia (multilingual)	4%
News (English)	1%

Chowdhery et al.,2022.

double descent for duplication?



Hernandez et al.,2022.

🔮 Currently, dataset construction is more akin to magic... Need principled methods!

Fantastic training data and where to find it

🔬 What even is high-quality data? technical filtering deduplication, lack of artefacts, etc.

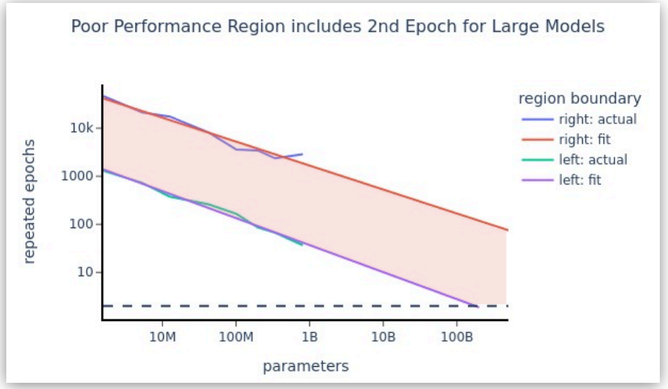
curation diverse, cross-domain, etc.

"social media conversations"

Total dataset size = 780 billion tokens	
Data source	Proportion of data
Social media conversations (multilingual)	50%
Filtered webpages (multilingual)	27%
Books (English)	13%
GitHub (code)	5%
Wikipedia (multilingual)	4%
News (English)	1%

Chowdhery et al.,2022.

double descent for duplication?



Hernandez et al.,2022.

🔮 Currently, dataset construction is more akin to magic... Need principled methods!

⚠️ Emergence of data moats which could stand in the way of research.

Fantastic training data and where to find it

Oh and by the way...

We need this in >100 languages!

We are doing Big Science, and this comes with challenges...

We are doing Big Science, and this comes with challenges...

🚀 LLMs are a true big science and require significant engineering efforts...

state-of-the-art HPC challenges

We are doing Big Science, and this comes with challenges...

🚀 LLMs are a true big science and require significant engineering efforts...

state-of-the-art HPC challenges

📒 Principled approaches are very much needed:

We are doing Big Science, and this comes with challenges...

🚀 LLMs are a true big science and require significant engineering efforts...

state-of-the-art HPC challenges

📒 Principled approaches are very much needed: tested and validated frameworks

We are doing Big Science, and this comes with challenges...

🚀 LLMs are a true big science and require significant engineering efforts...

state-of-the-art HPC challenges

📒 Principled approaches are very much needed: tested and validated frameworks

expert HPC/software engineering knowledge

We are doing Big Science, and this comes with challenges...

🚀 LLMs are a true big science and require significant engineering efforts...

state-of-the-art HPC challenges

📒 Principled approaches are very much needed: tested and validated frameworks

BLOOM: >100 configurations tested!

expert HPC/software engineering knowledge

performance tuning is magic currently

e.g. tile/wave quantization, distributed hyperparameters, etc.

We are doing Big Science, and this comes with challenges...

🚀 LLMs are a true big science and require significant engineering efforts...

state-of-the-art HPC challenges

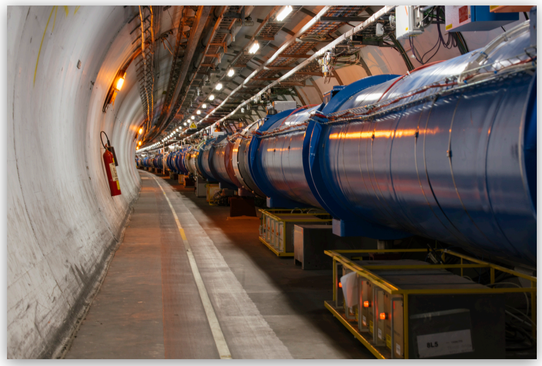
📒 Principled approaches are very much needed: tested and validated frameworks

BLOOM: >100 configurations tested!

expert HPC/software engineering knowledge

performance tuning is magic currently

e.g. tile/wave quantization, distributed hyperparameters, etc.



We are doing Big Science, and this comes with challenges...

🚀 LLMs are a true big science and require significant engineering efforts...

state-of-the-art HPC challenges

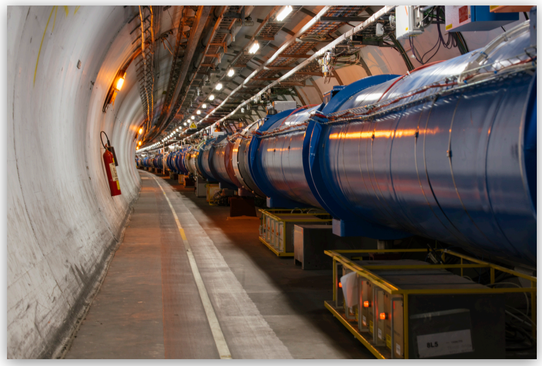
📒 Principled approaches are very much needed: tested and validated frameworks

BLOOM: >100 configurations tested!

expert HPC/software engineering knowledge

performance tuning is magic currently

e.g. tile/wave quantization, distributed hyperparameters, etc.



(let's avoid this)

Case-study of how hard it can get: Meta's OPT

OPT: Open Pre-Trained Transformer Language Models

Zhang et al., 2022

Case-study of how hard it can get: Meta's OPT

OPT: Open Pre-trained Transformer Language Models

Susan Zhang* Stephen Roller* Naman Goyal*
Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li,
Xi Victoria Lin, Todor Mihaylov, Myle OtC Sam Shleifer, Kurt Shuster, Daniel Simig,
Punit Singh Koura, Anjali Sridhar, Tianlu Wang, Luke Zettlemoyer
Meta AI
{susanz,roller,namangoyal}@fb.com

Abstract

Large language models, which are often trained for hundreds of thousands of compute days, have shown remarkable capabilities for zero- and few-shot learning. Given their computational cost, these models are difficult to replicate without significant capital. For the few that are available through APIs, no access is granted to the full model weights, making them difficult to study. We present Open Pre-trained Transformers (OPT), a suite of decoder-only pre-trained transformers ranging from 125M to 175B parameters, which we aim to fully and responsibly share with interested researchers. We show that OPT-175B is comparable to GPT-3, while requiring only 1/7th the carbon footprint to develop. We are also releasing our logbook detailing the infrastructure challenges we faced, along with code for experimenting with all of the released models.

1 Introduction

Large language models (LLMs) trained on massive text collections have shown surprising emergent capabilities to generate text and perform zero- and few-shot learning (Brown et al., 2020; Lieber et al., 2021; Smith et al., 2022; Rae et al., 2021; Chowdhery et al., 2022). While in some cases the public can interact with these models through paid APIs, full model access is currently limited to only a few highly resourced labs.[1] This restricted access has limited researchers' ability to study how and why these large language models work, hindering progress on improving known challenges in areas such as robustness, bias, and toxicity.

We present Open Pre-trained Transformers (OPT), a suite of decoder-only pre-trained transformers ranging from 125M to 175B parameters, which we aim to fully and responsibly share with interested researchers. We show that OPT-175B is comparable to GPT-3, while also applying the latest best practices in data collection and efficient training. Our aim in developing this suite of OPT models is to enable reproducible and responsible research at scale, and to bring more voices to the table in studying the impact of these LLMs. Definitions of risk, harm, bias, and toxicity, etc., should be articulated by the collective research community as a whole, which is only possible when models are available for study.

We are releasing all of our models between 125M and 30B parameters, and will provide full research access to OPT-175B upon request. Access will be granted to academic researchers; those affiliated with organizations in government, civil society, and academia; and those in industry research laboratories. We are also releasing both the logbook of our model creation as well as our codebase, metaseq,[2] which enabled training OPT-175B on 992 80GB A100 GPUs, reaching 147 TFLOP/s utilization per GPU. From this implementation, and from using the latest generation of NVIDIA hardware, we are able to develop OPT-175B using only 1/7th the carbon footprint of GPT-3. While this is a substantial improvement, the resources required for such a model is still nonetheless significant for many to replicate a model of this size will only amplify the growing compute footprint of these LLMs.

We believe the entire AI community — academic researchers, civil society, policymakers, and industry — must work together to develop clear

[2] https://github.com/facebookresearch/metaseq

OPT: Open Pre-Trained Transformer Language Models

Zhang et al., 2022

😅 Meta's open "reproduction" of GPT-3 was... a challenging experience!

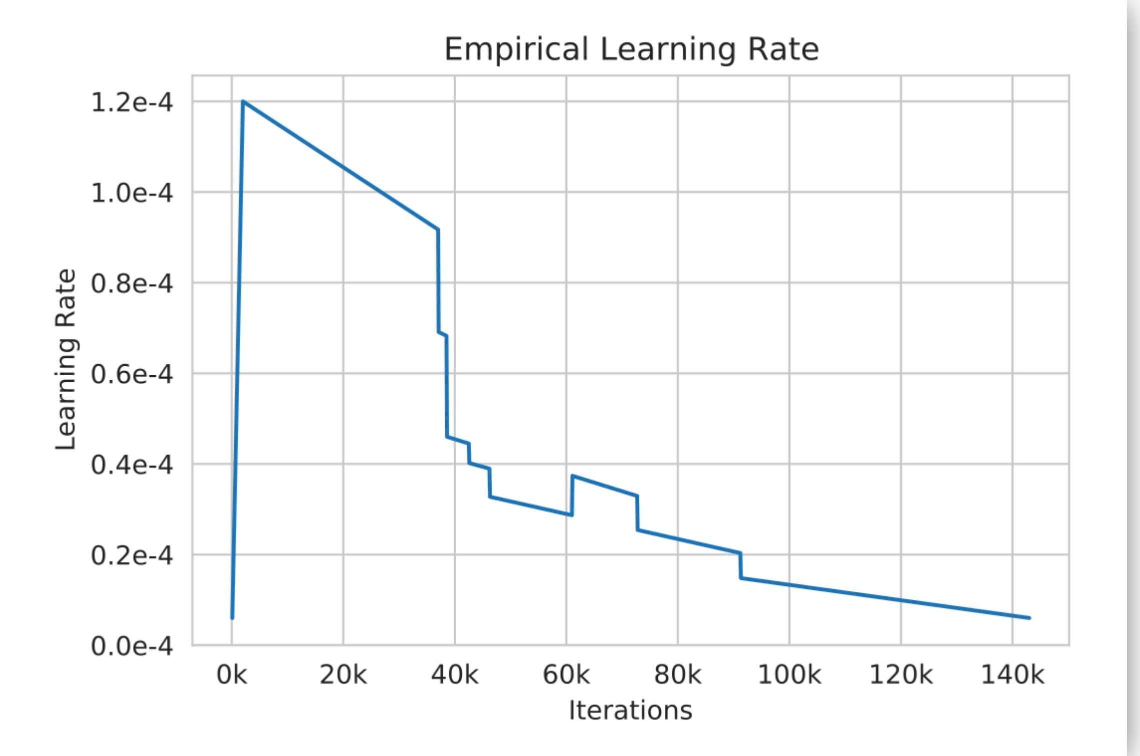
Case-study of how hard it can get: Meta's OPT


OPT: Open Pre-trained Transformer Language Models

OPT: Open Pre-Trained Transformer Language Models

Zhang et al., 2022

😅 Meta's open "reproduction" of GPT-3 was... a challenging experience!



manually tuned learning rate

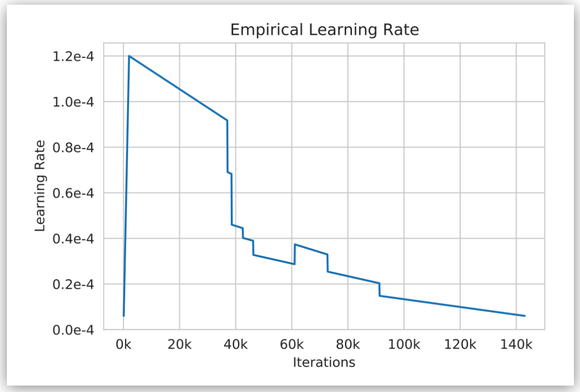
hundreds of restarts, spikes, etc.

Case-study of how hard it can get: Meta's OPT

OPT: Open Pre-Trained Transformer Language Models

Zhang et al., 2022

😅 Meta's open "reproduction" of GPT-3 was... a challenging experience!



manually tuned learning rate

hundreds of restarts, spikes, etc.

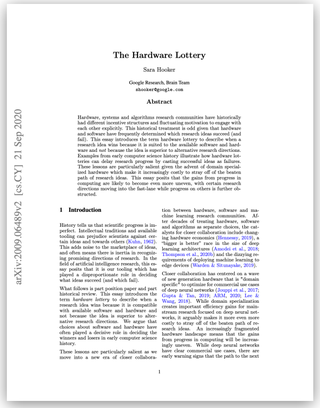
But why?

FP16

BF16

template: Karpathy, 2020

He who controls the chips controls the LLMs



The Hardware Lottery

Sara Hooker

Google Research, Brain Team
shooker@google.com

Abstract

Hardware, systems and algorithms research communities have historically had different incentive structures and fluctuating motivation to engage with each other explicitly. This historical tension has led to a lack of collaboration and software have frequently determined which research ideas succeed (and fail). This essay introduces the term hardware lottery to describe when a research idea wins because it is suited to the available software and hardware and not because the idea is superior to alternative research directions. Examples from early computer science history illustrate how hardware lotteries can delay research progress by casting successful ideas as failures. These lessons are particularly salient given the advent of domain specialized hardware which makes it increasingly costly to stray off of the beaten path of research ideas. This essay posits that the gains from progress in computing are likely to become even more uneven, with certain research directions moving into the fast-lane while progress on others is further obstructed.

1 Introduction

History tells us that scientific progress is imperfect. Intellectual traditions and available tooling can prejudice scientists against certain research directions (Sutton, 2019; Mattson et al., 2020). This adds noise to the marketplace of ideas, and often means there is inertia in recognizing promising directions of research. In the field of artificial intelligence research, this essay argues that the tools for research have played a disproportionate role in deciding what ideas succeed (and which fail).

What follows is part position paper and part historical review. This essay introduces the term hardware lottery to describe when a research idea wins because it is compatible with available software and hardware and not because the idea is superior to alternative research directions. We argue that choices about software and hardware have often played a decisive role in deciding the winners and losers in early computer science history.

These lessons are particularly salient as we move into a new era of closer collaboration between hardware, software and machine learning research communities. After decades of treating hardware, software and algorithms as separate choices, the catalysts for closer collaboration include changing hardware economics (Hennessy, 2019), a "bigger is better" race in the size of deep learning architectures (Amodei et al., 2018; Thompson et al., 2020) and the dissolution of requirements of deploying machine learning to edge devices (Warden & Situnayake, 2019).

Closer collaboration has centered on a wave of new generation hardware that is "domain specific" to optimize for commercial use cases of current neural networks (Jouppi et al., 2017; Ignatov et al., 2019; Fowers et al., 2018; Wang, 2018). While domain specialization creates important efficiency gains for mainstream research focused on a fixed set of deep learning workloads, it makes it more even more costly to stray off of the beaten path of research ideas. An increasingly fragmented hardware landscape means that the gains from algorithmic improvements will be unevenly enjoyed. While deep neural networks have clear commercial use cases, there are early warning signs that the path to the next

1

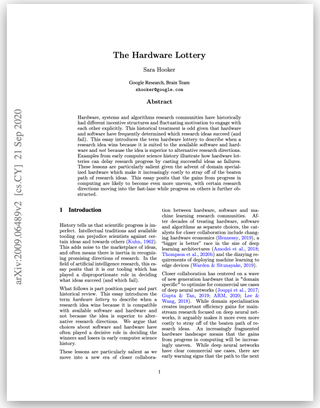
arXiv:2009.06489v2 [cs.CY] 21 Sep 2020

Hardware progress is secretly shaping machine learning

The Hardware Lottery

Sara Hooker, 2020

He who controls the chips controls the LLMs



The Hardware Lottery

Sara Hooker

Google Research, Brain Team
shooker@google.com

Abstract

Hardware, systems and algorithms research communities have historically had different incentive structures and fluctuating motivation to engage with each other explicitly (Lee and Hennessy, 2019). Decades of individual work by hardware and software have frequently determined which research ideas succeed (and fail). This essay introduces the term hardware lottery to describe when a research idea wins because it is suited to the available software and hardware and not because the idea is superior to alternative research directions. Examples from early computer science history illustrate how hardware lotteries can delay research progress by casting successful ideas as failures. These lessons are particularly salient given the advent of domain specialized hardware which makes it increasingly costly to stray off of the beaten path of research ideas. This essay posits that the gains from progress in computing are likely to become even more uneven, with certain research directions moving into the fast-lane while progress on others is further obstructed.

arXiv:2009.06489v2 [cs.CY] 21 Sep 2020

1 Introduction

History tells us that scientific progress is imperfect. Intellectual traditions and available tooling can prejudice scientists against certain research directions (Sutton, 2019; Lipton, 2018). This adds noise to the marketplace of ideas, and often means there is inertia in recognizing promising directions of research. In the field of artificial intelligence research, this essay argues that a similar force, hardware, has played a disproportionate role in deciding what ideas succeed (and which fail).

What follows is part position paper and part historical review. This essay introduces the term hardware lottery to describe when a research idea wins because it is compatible with available software and hardware and not because the idea is superior to alternative research directions. We argue that choices about software and hardware have often played a decisive role in deciding the winners and losers in early computer science history.

These lessons are particularly salient as we move into a new era of closer collaboration between hardware, software and machine learning research communities. After decades of treating hardware, software and algorithms as separate choices, there are concrete forces for closer collaboration include changing hardware economics (Hennessy, 2019), a "bigger is better" race in the size of deep learning architectures (Amodei et al., 2018; Thompson et al., 2020) and the poor fit between requirements of deploying machine learning to edge devices (Warden & Situnayake, 2019).

Closer collaboration has centered on a wave of new generation hardware that is "domain specific" to optimize for commercial use cases of current neural networks (Jouppi et al., 2017; Ignatov et al., 2019; Fowers et al., 2018; Wang, 2018). While domain specialization creates important efficiency gains for mainstream research focused on a fixed set of tools or workflows, it suppresses it more even more quietly to stray off of the beaten path of research ideas. An increasingly fragmented hardware landscape means that the gains from hardware specialization will be increasingly uneven. While deep neural networks have clear commercial use cases, there are early warning signs that the path to the next

1

Hardware progress is secretly shaping machine learning

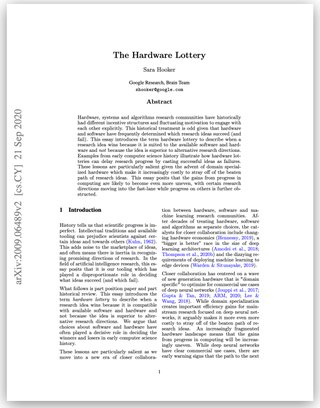
The Hardware Lottery

Sara Hooker, 2020

TPU

- pure data/model parallelism
- uniform platform experience

He who controls the chips controls the LLMs



The Hardware Lottery

Sara Hooker

Google Research, Brain Team
shooker@google.com

Abstract

Hardware, systems and algorithms research communities have historically had different incentive structures and fluctuating motivation to engage with each other explicitly (Lee and Hennessy, 2019). Guided by the principle that algorithms and hardware have frequently determined which research ideas succeed (and fail), this essay introduces the term hardware lottery to describe when a research idea wins because it is suited to the available software and hardware and not because the idea is superior to alternative research directions. Examples from early computer science history illustrate how hardware lotteries can delay research progress by casting successful ideas as failures. These lessons are particularly salient given the advent of domain specialized hardware which makes it increasingly costly to stray off of the beaten path of research ideas. This essay posits that the gains from progress in computing are driven by hardware lotteries and encourages researchers to explore directions moving into the fast-lane while progress on others is further obstructed.

arXiv:2009.06489v2 [cs.CY] 21 Sep 2020

1 Introduction

History tells us that scientific progress is imperfect. Intellectual traditions and available tooling can prejudice scientists against certain research directions (Arbib and Hesse, 1986). This adds noise to the marketplace of ideas, and often means there is inertia in recognizing promising directions of research. In the field of artificial intelligence research, this essay argues that hardware and software have played a disproportionate role in deciding what ideas succeed (and which fail).

What follows is part position paper and part historical review. This essay introduces the term hardware lottery to describe when a research idea wins because it is compatible with available software and hardware and not because the idea is superior to alternative research directions. We argue that choices about software and hardware have often played a decisive role in deciding the winners and losers in early computer science history.

These lessons are particularly salient as we move into a new era of closer collaboration between hardware, software and machine learning research communities. After decades of treating hardware, software and algorithms as separate choices, the catalysts for closer collaboration include changing hardware economics (Hennessy, 2019), a "bigger is better" race in the size of deep learning architectures (Amodei et al., 2018), and the dwindling returns from general purpose improvements of clockrate (Warden & Situnayake, 2019).

Closer collaboration has centered on a wave of new generation hardware that is "domain specific" to optimize for commercial use cases (Jouppi et al., 2017; Fowers et al., 2018; Ignatov et al., 2019; Wang et al., 2019). While domain specialization creates important efficiency gains for mainstream research focused on a narrow set of workloads, it applies pressure to progress currently to stray off of the beaten path of research ideas. An increasingly fragmented hardware landscape means that the gains from hardware specialization will be increasingly uneven. While deep neural networks have clear commercial use cases, there are early warning signs that the path to the next

1

Hardware progress is secretly shaping machine learning

The Hardware Lottery

Sara Hooker, 2020

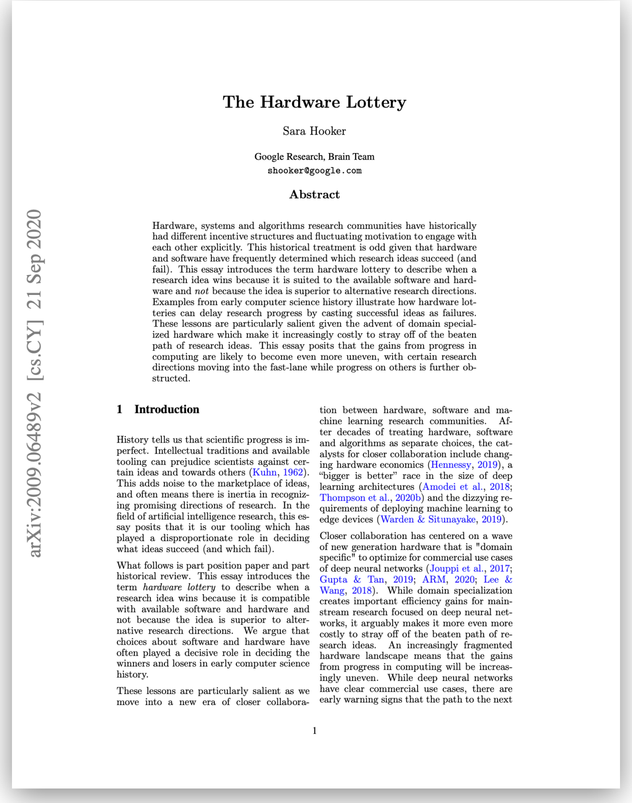
TPU

- pure data/model parallelism
- uniform platform experience

GPU

- data/model/pipeline/sequence parallelism
- diversity in HPC platforms
  - network topology, etc.

He who controls the chips controls the LLMs



Hardware progress is secretly shaping machine learning

The Hardware Lottery

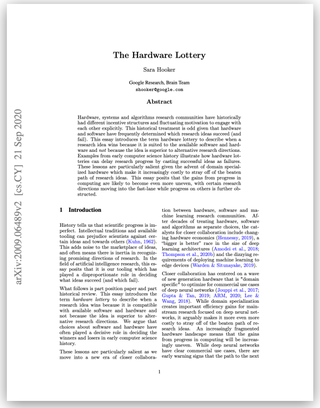
Sara Hooker, 2020

TPU
- pure data/model parallelism
- uniform platform experience

→ simpler experience?

GPU
- data/model/pipeline/sequence parallelism
- diversity in HPC platforms
  network topology, etc.

He who controls the chips controls the LLMs



The Hardware Lottery

Sara Hooker

Google Research, Brain Team
shooker@google.com

Abstract

Hardware, systems and algorithms research communities have historically had different incentive structures and fluctuating motivation to engage with each other explicitly (Lee and Hennessy, 2019). Guided by the principle that algorithms and hardware have frequently determined which research ideas succeed (and fail), this essay introduces the term hardware lottery to describe when a research idea wins because it is suited to the available software and hardware and not because the idea is superior to alternative research directions. Examples from early computer science history illustrate how hardware lotteries can delay research progress by casting successful ideas as failures. These lessons are particularly salient given the advent of domain specialized hardware which makes it increasingly costly to stray off of the beaten path of research ideas. This essay posits that the gains from progress in computing are driven by hardware lotteries and that achieving a fairer lottery for research directions moving into the fast-lane while progress on others is further obstructed.

arXiv:2009.06489v2 [cs.CY] 21 Sep 2020

1 Introduction

History tells us that scientific progress is imperfect. Intellectual traditions and available tooling can prejudice scientists against certain research directions (Arbib and Hesse, 1986). This adds noise to the marketplace of ideas, and often means there is inertia in recognizing promising directions of research. In the field of artificial intelligence research, this essay argues that hardware and software have played a disproportionate role in deciding what ideas succeed (and which fail).

What follows is part position paper and part historical review. This essay introduces the term hardware lottery to describe when a research idea wins because it is compatible with available software and hardware and not because the idea is superior to alternative research directions. We argue that choices about software and hardware have often played a decisive role in deciding the winners and losers in early computer science history.

These lessons are particularly salient as we move into a new era of closer collaboration between hardware, software and machine learning research communities. After decades of treating hardware, software and algorithms as independent choices, the catalysts for closer collaboration include changing hardware economics (Hennessy, 2019), a "bigger is better" race in the size of deep learning architectures (Amodei et al., 2018; Thompson et al., 2020) and the dissolution of abstractions of duplexing machine learning to edge devices (Warden & Situnayake, 2019).

Closer collaboration has centered on a wave of new generation hardware that is "domain specific" to optimize for commercial use cases (Jouppi et al., 2017; Fowers et al., 2018; Ignatov et al., 2019; Wang et al., 2019; Wang, 2019). While domain specialization creates important efficiency gains for mainstream research focused on dominant models or tasks, it applies pressure to disincentivize currently to stray off of the beaten path of research ideas. An increasingly fragmented hardware landscape means that the gains from algorithmic improvements will be increasingly uneven. While deep neural networks have clear commercial use cases, there are early warning signs that the path to the next

1

Hardware progress is secretly shaping machine learning

The Hardware Lottery

Sara Hooker, 2020

TPU

- pure data/model parallelism
- uniform platform experience

→ simpler experience?

GPU

- data/model/pipeline/sequence parallelism
- diversity in HPC platforms
  - network topology, etc.

Google, Facebook, Tesla, Amazon are all making their own chips!

Can better modeling & more efficient pretraining change the playing field?

- 💨 We can gain in efficiency...

Can better modeling & more efficient pretraining change the playing field?

💨 We can gain in efficiency... current approaches, ~50% GPU FLOPs usage

Can better modeling & more efficient pretraining change the playing field?

💨 We can gain in efficiency...

current approaches, ~50% GPU FLOPs usage

reduced numerical precision: down to int8

see Transformer engine in H100

Can better modeling & more efficient pretraining change the playing field?

💨 We can gain in efficiency...

current approaches, ~50% GPU FLOPs usage

reduced numerical precision: down to int8

see Transformer engine in H100

reduce number of computations

efficient attention, etc.

Can better modeling & more efficient pretraining change the playing field?

💨 We can gain in efficiency...

current approaches, ~50% GPU FLOPs usage

reduced numerical precision: down to int8

see Transformer engine in H100

reduce number of computations

efficient attention, etc.

But can we also fundamentally change scaling behaviour?

Can better modeling & more efficient pretraining change the playing field?

💨 We can gain in efficiency...

current approaches, ~50% GPU FLOPs usage

reduced numerical precision: down to int8
see Transformer engine in H100

reduce number of computations
efficient attention, etc.

But can we also fundamentally change scaling behaviour?

First, optimise pretraining:

- Curriculum learning, grow sequence length — Li et al., 2021
- Staged training, progressively grow model — Shen et al., 2022

Can better modeling & more efficient pretraining change the playing field?

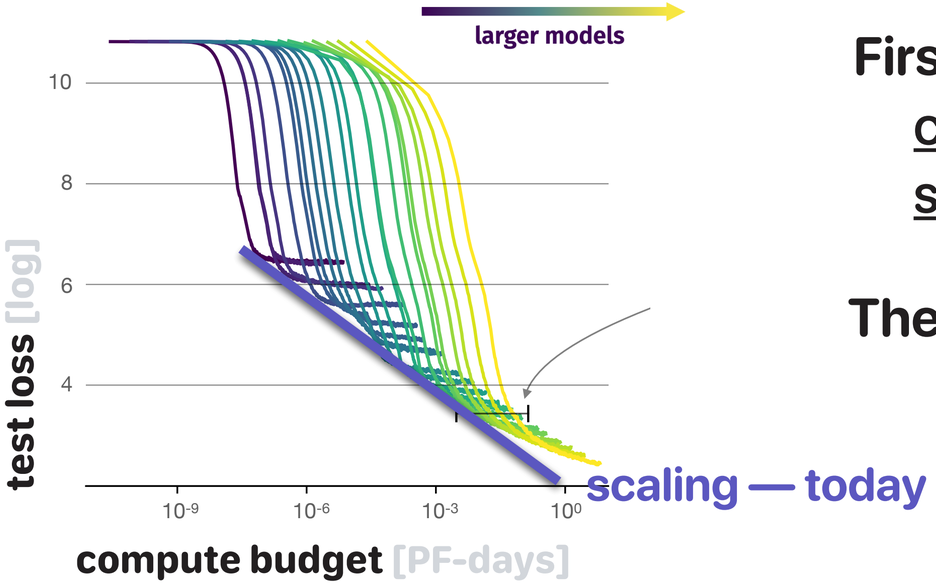
💨 We can gain in efficiency...

current approaches, ~50% GPU FLOPs usage

reduced numerical precision: down to int8
see Transformer engine in H100

reduce number of computations
efficient attention, etc.

But can we also fundamentally change scaling behaviour?



First, optimise pretraining:

Curriculum learning, grow sequence length — Li et al., 2021

Staged training, progressively grow model — Shen et al., 2022

Then, can we get better scaling?

🤗 please report scaling laws in your modeling work!

Can better modeling & more efficient pretraining change the playing field?

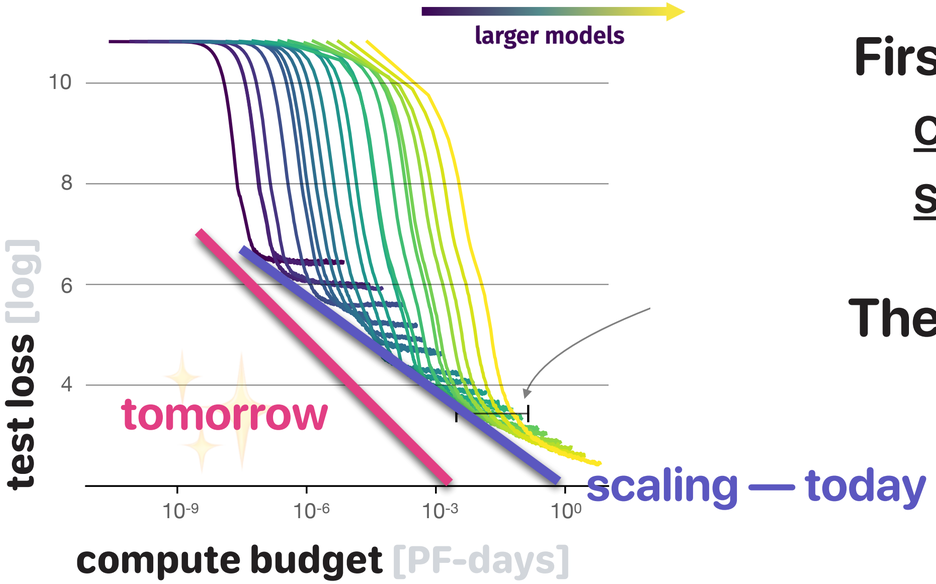
💨 We can gain in efficiency...

current approaches, ~50% GPU FLOPs usage

reduced numerical precision: down to int8
see Transformer engine in H100

reduce number of computations
efficient attention, etc.

But can we also fundamentally change scaling behaviour?



First, optimise pretraining:

Curriculum learning, grow sequence length — Li et al., 2021

Staged training, progressively grow model — Shen et al., 2022

Then, can we get better scaling?

🤗 please report scaling laws in your modeling work!

🤗 Thank you to all contributors!