AI Box

大模型技术基础

《大语言模型》编写团队：赵鑫

大语言模型

AI Box

- 定义：通常是指具有超大规模参数的预训练语言模型
- 架构：主要为 Transformer解码器架构
- 训练：预训练（base model）、后训练（instruct model）

训练

预训练

数据：海量文本数据

优化：预测下一个词

建立模型的**基础能力**

base model

后训练

数据：大量指令数据

优化：SFT、RL等方法

增强模型的**任务能力**

instruct model

下游应用

测试（推理）

大语言模型

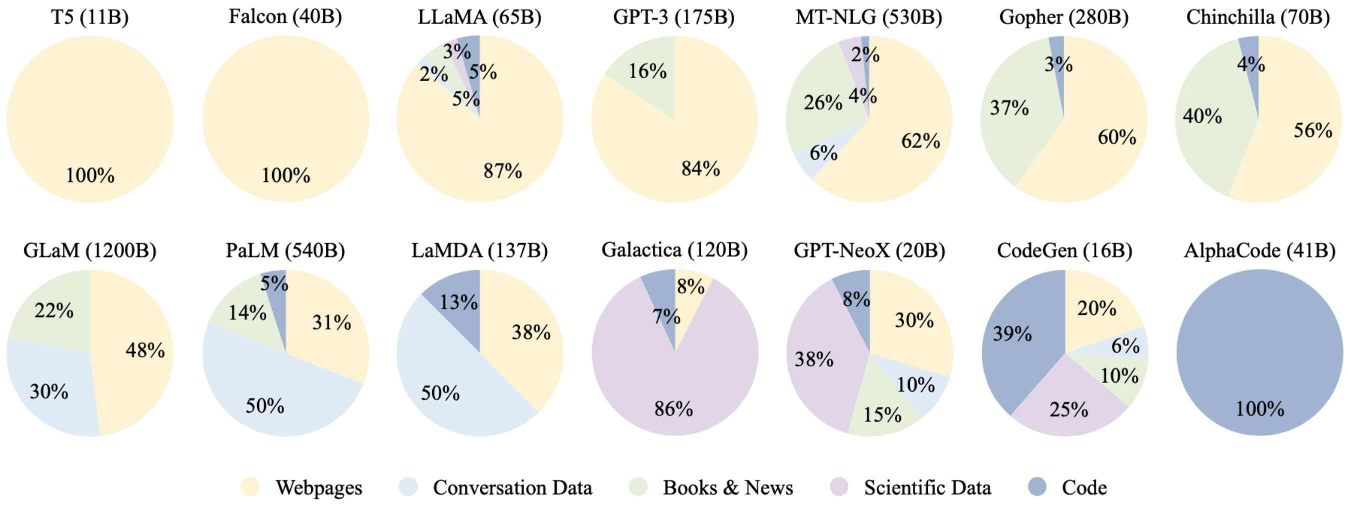
- 定义：通常是指具有超大规模参数的预训练语言模型
- 架构：主要为 Transformer解码器架构
- 训练：预训练（base model）、后训练（instruct model）

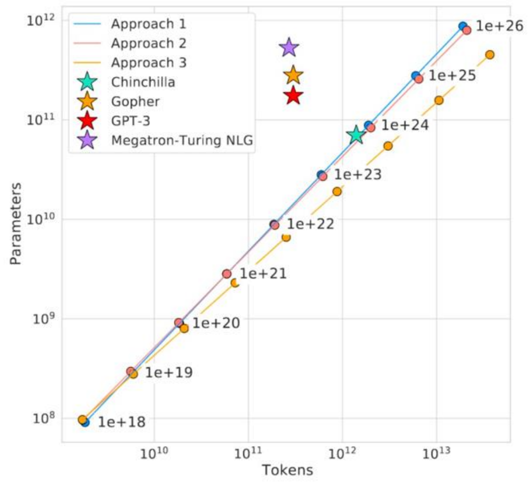
对比方面	预训练 (Pre-training)	后训练 (Post-training)
核心目标	建立模型基础能力	将基座模型适配到具体应用场景
数据资源	数万亿词元的自然语言文本	数十万、数百万到数千万指令数据
所需算力	耗费百卡、千卡甚至万卡算力数月时间	耗费数十卡、数百卡数天到数十天时间
使用方式	通常为few-shot提示	可以直接进行zero-shot使用

*此部分算力估计为一个大致估计，需要根据模型大小、数据数量、训练框架等多方面因素确定

大语言模型构建概览

- 大语言模型预训练（Pre-training）
  - 使用与下游任务无关的大规模模数据进行模型参数的初始训练
    - 基于Transformer解码器架构，进行下一个词预测
    - 数据数量、数据质量都非常关键

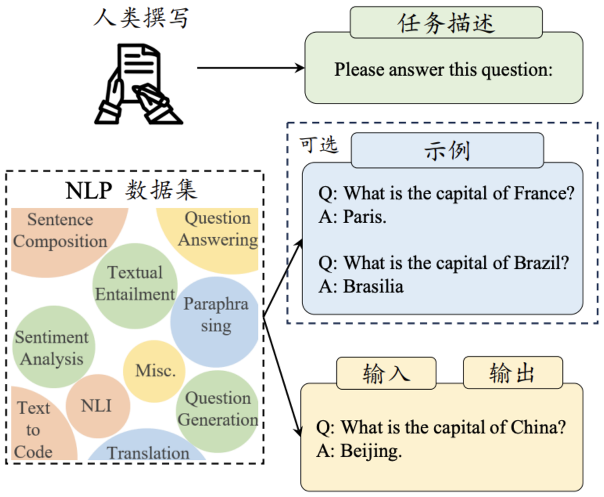


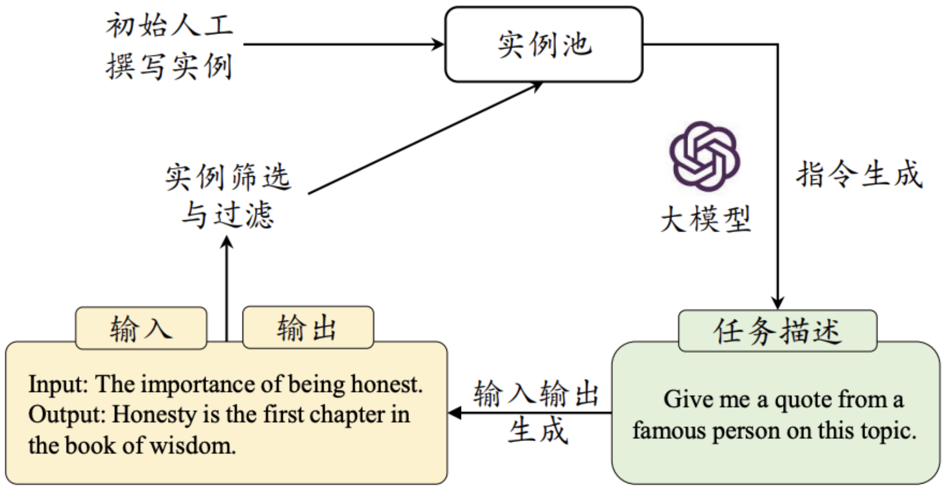


A Survey of Large Language Model, arXiv 2023

大语言模型构建概览

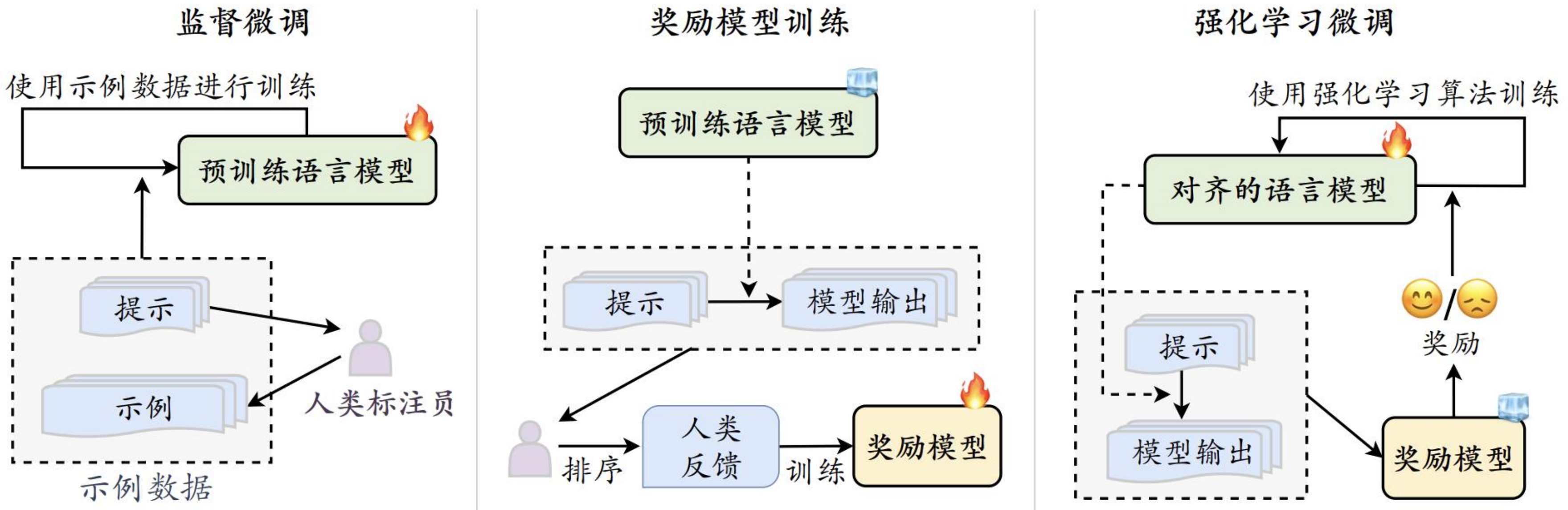
- 大语言模型后训练（Post-Training）
  - 指令微调（Instruction Tuning）
    - 使用输入与输出配对的指令数据对于模型进行微调
    - 提升模型通过问答形式进行任务求解的能力





大语言模型构建概览

- 大语言模型后训练（Post-Training）
  - 人类对齐（Human Alignment）
    - 将大语言模型与人类的期望、需求以及价值观对齐
    - 基于人类反馈的强化学习对齐方法（RLHF）



大语言模型构建概览

➢ 大模型的研发已经成为一项系统工程

Authorship, Credit Attribution, and Acknowledgements

Please cite this work as “OpenAI (2023)”.

Pretraining

Core contributors[11]
Christopher Berner *Supercomputing lead*
Greg Brockman *Infrastructure lead*
Trevor Cai *Throughput lead*
David Farhi *Manager of optimization team*
Chris Hesse *Infrastructure usability co-lead*
Shantanu Jain *Infrastructure usability co-lead*
Kyle Kosic *Uptime and stability lead*
Jakub Pachocki *Overall lead, optimization lead*
Alex Paino *Architecture & data vice lead*
Mikhail Pavlov *Software correctness lead*
Michael Petrov *Hardware correctness lead*
Nick Ryder *Architecture & data lead*
Szymon Sidor *Optimization vice lead*
Nikolas Tezak *Execution lead*
Phil Tillet *Triton lead*
Amin Tootoonchian *Model distribution, systems & networking lead*
Qiming Yuan *Dataset sourcing and processing lead*
Wojciech Zaremba *Manager of dataset team*

Compute cluster scaling[11]
Christopher Berner, Oleg Boiko, Andrew Cann, Ben Chess, Christian Gibson, Mateusz Litwin, Emy Parparita, Henri Roussez, Eric Sigler, Akila Welihinda

Data[11]
Sandhini Agarwal, Suchir Balaji, Mo Bavarian, Che Chang, Sheila Dunning, Leo Gao, Jonathan Gordon, Peter Hoeschele, Shawn Jain, Shantanu Jain, Roger Jiang, Heewoo Jun, Łukasz Kaiser, Nitish Shirish Keskar, Jong Wook Kim, Aris Konstantinidis, Chak Ming Li, Todor Markov, Bianca Martin, David Mély, Oleg Murk, Hyeonwoo Noh, Long Ouyang, Alex Paino, Vitchyr Pong, Alec Radford, Nick Ryder, John Schulman, Daniel Selsam, Ian Sohl, Chelsea Voss, Lilian Weng, Clemens Winter, Tao Xu, Qiming Yuan, Wojciech Zaremba

Distributed training infrastructure[11]
Greg Brockman, Trevor Cai, Chris Hesse, Shantanu Jain, Yongjik Kim, Kyle Kosic, Mateusz Litwin, Jakub Pachocki, Mikhail Pavlov, Szymon Sidor, Nikolas Tezak, Madeleine Thompson, Amin Tootoonchian, Qiming Yuan

Hardware correctness[11]
Greg Brockman, Shantanu Jain, Kyle Kosic, Michael Petrov, Nikolas Tezak, Amin Tootoonchian, Chelsea Voss, Qiming Yuan

Optimization & architecture[11]
Igor Babuschkin, Mo Bavarian, Adrien Ecoffet, David Farhi, Jesse Han, Ingmar Kanitscheider, Daniel Levy, Jakub Pachocki, Alex Paino, Mikhail Pavlov, Nick Ryder, Szymon Sidor, Jie Tang, Jerry Tworek, Tao Xu

Training run babysitting[11]
Suchir Balaji, Mo Bavarian, Greg Brockman, Trevor Cai, Chris Hesse, Shantanu Jain, Roger Jiang, Yongjik Kim, Kyle Kosic, Mateusz Litwin, Jakub Pachocki, Alex Paino, Mikhail Pavlov, Michael Petrov, Nick Ryder, Szymon Sidor, Nikolas Tezak, Madeleine Thompson, Phil Tillet, Amin Tootoonchian, Chelsea Voss, Ben Wang, Tao Xu, Qiming Yuan

Long context

Core contributors[11]
Gabriel Goh *Long context co-lead*
Łukasz Kaiser *Long context lead*
Ben Wang *Attention architecture lead*
Clemens Winter *Long context co-lead*

Long context research[11]
Mo Bavarian, Gabriel Goh, Heewoo Jun, Łukasz Kaiser, Chak Ming Li, Ben Wang, Clemens Winter

Long context kernels[11]
Phil Tillet

Vision

Core contributors[11]
Trevor Cai *Execution lead*
Mark Chen *Vision team co-lead, Deployment lead*
Casey Chu *Initial prototype lead*
Chris Hesse *Data load balancing & developer tooling lead*
Shengli Hu *Vision Safety Evaluations lead*
Yongjik Kim *GPU performance lead*
Jamie Kiros *Overall vision co-lead, deployment research & evals lead*
Daniel Levy *Overall vision co-lead, optimization lead*
Christine McLeavey *Vision team lead*
David Mély *Data lead*
Hyeonwoo Noh *Overall vision co-lead, research lead*
Mikhail Pavlov *Scaling engineering lead*
Raul Puri *Overall vision co-lead, engineering lead*
Amin Tootoonchian *Model distribution, systems & networking lead*

Architecture research[11]
Casey Chu, Jamie Kiros, Christine McLeavey, Hyeonwoo Noh, Raul Puri, Alec Radford, Aditya Ramesh

Compute cluster scaling[11]
Andrew Cann, Rory Carmichael, Christian Gibson, Henri Roussez, Akila Welihinda

Distributed training infrastructure[11]
Trevor Cai, Yunxing Dai, Chris Hesse, Brandon Houghton, Yongjik Kim, Łukasz Kondraciuk, Hyeonwoo Noh, Mikhail Pavlov, Raul Puri, Nikolas Tezak, Amin Tootoonchian, Tianhao Zheng

Hardware correctness[11]
Oleg Boiko, Trevor Cai, Michael Petrov, Alethea Power

Data[11]
Jong Wook Kim, David Mély, Reiichiro Nakano, Hyeonwoo Noh, Long Ouyang, Raul Puri, Pranav Shyam, Tao Xu

Alignment data[11]
Long Ouyang

Training run babysitting[11]
Trevor Cai, Kyle Kosic, Daniel Levy, David Mély, Reiichiro Nakano, Hyeonwoo Noh, Mikhail Pavlov, Raul Puri, Amin Tootoonchian

Deployment & post-training[11]
Ilge Akkaya, Mark Chen, Jamie Kiros, Rachel Lim, Reiichiro Nakano, Raul Puri, Jiayi Weng

Reinforcement Learning & Alignment

Core contributors[11]
Greg Brockman *Core infrastructure author*
Arka Dhar *Human data product manager*
Liam Fedus *Data flywheel lead*
Tarun Gogineni *Model creativity*
Rapha Gontijo-Lopes *Synthetic data*
Joshua Gross *Data collection engineering co-lead*
Johannes Heidecke *Refusals & model safety co-lead*
Joost Huizinga *Initial fine-tuning derisking*
Teddy Lee *Human data product manager*
Jan Leike *Alignment co-lead*
Ryan Lowe *Alignment co-lead*
Luke Metz *Infrastructure lead, ChatML format lead*
Long Ouyang *IF data collection lead*
John Schulman *Overall lead*
Jerry Tworek *Code lead*
Carroll Wainwright *IF data infrastructure lead*
Jonathan Ward *Data collection engineering co-lead*
Jiayi Weng *RL Infrastructure author*
Sarah Yoo *Human data operations manager*
Wojciech Zaremba *Human data lead*
Chong Zhang *Refusals & model safety co-lead*
Shengjia Zhao *Reward model lead*
Barret Zoph *Overall training lead*

Dataset contributions[11]
Diogo Almeida, Mo Bavarian, Juan Felipe Cerón Uribe, Tyna Eloundou, Liam Fedus, Tarun Gogineni, Rapha Gontijo-Lopes, Jonathan Gordon, Joost Huizinga, Shawn Jain, Roger Jiang, Łukasz Kaiser, Christina Kim, Jan Leike, Chak Ming Li, Stephanie Lin, Ryan Lowe, Jacob Menick, Luke Metz, Pamela Mishkin, Tong Mu, Oleg Murk, Ashvin Nair, Long Ouyang, Alex Passos, Michael (Rai) Pokorny, Vitchyr Pong, Shibani Santurkar, Daniel Selsam, Sarah Shoker, Carroll Wainwright, Matt Wiethoff, Jeff Wu, Kai Xiao, Kevin Yu, Marvin Zhang, Chong Zhang, William Zhuk, Barret Zoph

Data infrastructure[11]
Irwan Bello, Lenny Bogdonoff, Juan Felipe Cerón Uribe, Joshua Gross, Shawn Jain, Haozhun Jin, Christina Kim, Aris Konstantinidis, Teddy Lee, David Medina, Jacob Menick, Luke Metz, Ashvin Nair, Long Ouyang, Michael (Rai) Pokorny, Vitchyr Pong, John Schulman, Jonathan Ward, Jiayi Weng, Matt Wiethoff, Sarah Yoo, Kevin Yu, Wojciech Zaremba, William Zhuk, Barret Zoph

ChatML format[11]
Ilge Akkaya, Christina Kim, Chak Ming Li, Rachel Lim, Jacob Menick, Luke Metz, Andrey Mishchenko, Vitchyr Pong, John Schulman, Carroll Wainwright, Barret Zoph

Model safety[11]
Josh Achiam, Steven Adler, Juan Felipe Cerón Uribe, Hyung Won Chung, Tyna Eloundou, Rapha Gontijo-Lopes, Shixiang Shane Gu, Johannes Heidecke, Joost Huizinga, Teddy Lee, Jan Leike, Stephanie Lin, Ryan Lowe, Todor Markov, Luke Metz, Tong Mu, Shibani Santurkar, John Schulman, Andrea Vallone, Carroll Wainwright, Jason Wei, Lilian Weng, Kai Xiao, Chong Zhang, Marvin Zhang, Barret Zoph

Refusals[11]
Juan Felipe Cerón Uribe, Tyna Eloundou, Johannes Heidecke, Joost Huizinga, Jan Leike, Stephanie Lin, Ryan Lowe, Pamela Mishkin, Tong Mu, Carroll Wainwright, Lilian Weng, Kai Xiao, Chong Zhang, Barret Zoph

Foundational RLHF and InstructGPT work[11]
Diogo Almeida, Joost Huizinga, Roger Jiang, Jan Leike, Stephanie Lin, Ryan Lowe, Pamela Mishkin, Dan Mossing, Long Ouyang, Katarina Slama, Carroll Wainwright, Jeff Wu, Kai Xiao, Marvin Zhang

Flagship training runs[11]
Greg Brockman, Liam Fedus, Johannes Heidecke, Joost Huizinga, Roger Jiang, Kyle Kosic, Luke Metz, Ashvin Nair, Jiayi Weng, Chong Zhang, Shengjia Zhao, Barret Zoph

Code capability[11]
Ilge Akkaya, Mo Bavarian, Jonathan Gordon, Shawn Jain, Haozhun Jin, Teddy Lee, Chak Ming Li, Oleg Murk, Ashvin Nair, Vitchyr Pong, Benjamin Sokolowsky, Jerry Tworek, Matt Wiethoff, Sarah Yoo, Kevin Yu, Wojciech Zaremba, William Zhuk

Evaluation & analysis

Core contributors[11]
Sandhini Agarwal *System card co-lead*
Lama Ahmad *Expert red teaming & adversarial testing program lead*
Mo Bavarian *Capability prediction co-lead*
Tyna Eloundou *Safety evaluations co-lead*
Andrew Kondrich *OpenAI Evals open-sourcing co-lead*
Gretchen Krueger *System card co-lead*
Michael Lampe *Privacy and PII evaluations lead*
Pamela Mishkin *Economic impact & overreliance evaluations lead*
Benjamin Sokolowsky *Capability prediction co-lead*
Jack Rae *Research benchmark execution lead*
Chelsea Voss *Eval execution lead*
Alvin Wang *OpenAI Evals lead*
Kai Xiao *Safety evaluations co-lead*
Marvin Zhang *OpenAI Evals open-sourcing co-lead*

OpenAI Evals library[11]
Shixiang Shane Gu, Angela Jiang, Logan Kilpatrick, Andrew Kondrich, Pamela Mishkin, Jakub Pachocki, Ted Sanders, Jessica Shieh, Alvin Wang, Marvin Zhang

Model-graded evaluation infrastructure[11]
Liam Fedus, Rapha Gontijo-Lopes, Shixiang Shane Gu, Andrew Kondrich, Michael (Rai) Pokorny, Wojciech Zaremba, Chong Zhang, Marvin Zhang, Shengjia Zhao, Barret Zoph

Acceleration forecasting[11]
Alan Hickey, Daniel Kokotajlo, Cullen O'Keefe, Sarah Shoker

ChatGPT evaluations[11]
Juan Felipe Cerón Uribe, Hyung Won Chung, Rapha Gontijo-Lopes, Liam Fedus, Luke Metz, Michael Rai Pokorny, Jason Wei, Shengjia Zhao, Barret Zoph

Capability evaluations[11]
Tyna Eloundou, Shengli Hu, Roger Jiang, Jamie Kiros, Teddy Lee, Scott Mayer McKinney, Jakub Pachocki, Alex Paino, Giambattista Parascandolo, Boris Power, Raul Puri, Jack Rae, Nick Ryder, Ted Sanders, Szymon Sidor, Benjamin Sokolowsky, Chelsea Voss, Alvin Wang, Rowan Zellers, Juntang Zhuang

Coding evaluations[11]
Ilge Akkaya, Mo Bavarian, Jonathan Gordon, Shawn Jain, Chak Ming Li, Oleg Murk, Vitchyr Pong, Benjamin Sokolowsky, Jerry Tworek, Kevin Yu, Wojciech Zaremba

Real-world use case evaluations[11]
Andrew Kondrich, Joe Palermo, Boris Power, Ted Sanders

Contamination investigations[11]
Adrien Ecoffet, Roger Jiang, Ingmar Kanitscheider, Scott Mayer McKinney, Alex Paino, Giambattista Parascandolo, Jack Rae, Qiming Yuan

Instruction following and API evals[11]
Diogo Almeida, Carroll Wainwright, Marvin Zhang

Novel capability discovery[11]
Filipe de Avila Belbute Peres, Kevin Button, Fotis Chantzis, Mike Heaton, Wade Hickey, Xin Hu, Andrew Kondrich, Matt Knight, Andrew Mayne, Jake McNeil, Vinnie Monaco, Joe Palermo, Joel Parish, Boris Power, Bob Rotsted, Ted Sanders

Vision evaluations[11]
Shixiang Shane Gu, Shengli Hu, Jamie Kiros, Hyeonwoo Noh, Raul Puri, Rowan Zellers

Economic impact evaluation[11]
Tyna Eloundou, Sam Manning, Aalok Mehta, Pamela Mishkin

Non-proliferation, international humanitarian law & national security red teaming[11]
Sarah Shoker

Overreliance analysis[11]
Miles Brundage, Michael Lampe, Pamela Mishkin

Privacy and PII evaluations[11]
Michael Lampe, Vinnie Monaco, Ashley Pantuliano

Safety and policy evaluations[11]
Josh Achiam, Sandhini Agarwal, Lama Ahmad, Jeff Belgum, Tyna Eloundou, Johannes Heidecke, Shengli Hu, Joost Huizinga, Jamie Kiros, Gretchen Krueger, Michael Lampe, Stephanie Lin, Ryan Lowe, Todor Markov, Vinnie Monaco, Tong Mu, Raul Puri, Girish Sastry, Andrea Vallone, Carroll Wainwright, CJ Weinmann, Lilian Weng, Kai Xiao, Chong Zhang

OpenAI adversarial testers[11]
Josh Achiam, Steven Adler, Lama Ahmad, Shyamal Anadkat, Red Avila, Gabriel Bernadett-Shapiro, Anna-Luisa Brakman, Tim Brooks, Miles Brundage, Chelsea Carlson, Derek Chen, Hyung Won Chung, Jeremiah Currier, Daniel Kokotajlo, David Dohan, Adrien Ecoffet, Juston Forte, Vik Goel, Ryan Greene, Johannes Heidecke, Alan Hickey, Shengli Hu, Joost Huizinga, Janko, Tomer Kaftan, Ali Kamali, Nitish Shirish Keskar, Tabarak Khan, Hendrik Kirchner, Daniel Kokotajlo, Gretchen Krueger, Michael Lampe, Teddy Lee, Molly Lin, Ryan Lowe, Todor Markov, Jake McNeil, Pamela Mishkin, Vinnie Monaco, Daniel Mossing, Tong Mu, Oleg Murk, Cullen O'Keefe, Joe Palermo, Giambattista Parascandolo, Joel Parish, Boris Power, Alethea Power, Cameron Raymond, Francis Real, Bob Rotsted, Mario Salterelli, Sam Wolrich, Ted Sanders, Girish Sastry, Sarah Shoker, Shyamal Anadkat, Yang Song, Natalie Staudacher, Madeleine Thompson, Elizabeth Tseng, Chelsea Voss, Jason Wei, Chong Zhang

System card & broader impacts analysis[11]
Steven Adler, Sandhini Agarwal, Lama Ahmad, Janko Altenschmidt, Jeff Belgum, Gabriel Bernadett-Shapiro, Miles Brundage, Derek Chen, Tyna Eloundou, Liam Fedus, Leo Gao, Vik Goel, Johannes Heidecke, Alan Hickey, Shengli Hu, Joost Huizinga, Daniel Kokotajlo, Gretchen Krueger, Michael Lampe, Jade Leung, Stephanie Lin, Ryan Lowe, Kim Malfacini, Todor Markov, Bianca Martin, Aalok Mehta, Pamela Mishkin, Tong Mu, Richard Ngo, Cullen O'Keefe, Joel Parish, Rai Pokorny, Bob Rotsted, Girish Sastry, Sarah Shoker, Andrea Vallone, Carroll Wainwright, CJ Weinmann, Lilian Weng, Dave Willner, Kai Xiao, Chong Zhang

Deployment

Core contributors[11]
Steven Adler *Early stage program management lead*
Sandhini Agarwal *Launch safety lead*
Derek Chen *Monitoring & response lead*
Atty Eleti *GPT-4 API co-lead*
Joanne Jang *GPT-4 product co-lead*
Angela Jiang *GPT-4 product co-lead*
Tomer Kaftan *Inference infrastructure & deployment lead*
Rachel Lim *GPT-4 API co-lead*
Kim Malfacini *Usage policy lead*
Bianca Martin *Release program management lead*
Evan Morikawa *Engineering lead*
Henrique Ponde de Oliveira Pinto *Inference workflow lead*
Heather Schmidt *GPT-4 infrastructure management*
Maddie Simens *Design lead*
Felipe Petroski Such *Inference optimization & reliability lead*
Andrea Vallone *Detection & refusals policy lead*
Lilian Weng *Applied research lead*
Dave Willner *Trust & safety lead*
Michael Wu *Inference research lead*

Inference research[11]
Paul Baltescu, Scott Gray, Yuchen He, Arvind Neelakantan, Michael Wu

GPT-4 API & ChatML deployment[11]
Greg Brockman, Brooke Chan, Chester Cho, Atty Eleti, Rachel Lim, Andrew Peng, Michelle Pokrass, Sherwin Wu

GPT-4 web experience[11]
Valerie Balcom, Lenny Bogdonoff, Jason Chen, Dave Cummings, Noah Deutsch, Mike Heaton, Paul McMillan, Rajeev Nayak, Joel Parish, Adam Perelman, Eric Sigler, Nick Turley, Arun Vijayvergiya, Chelsea Voss

Inference infrastructure[11]
Brooke Chan, Scott Gray, Chris Hallacy, Kenny Hsu, Tomer Kaftan, Rachel Lim, Henrique Ponde de Oliveira Pinto, Raul Puri, Heather Schmidt, Felipe Petroski Such

Reliability engineering[11]
Haiming Bao, Madelaine Boyd, Ben Chess, Damien Deville, Yufei Guo, Vishal Kuo, Ikai Lan, Michelle Pokrass, Carl Ross, David Schnurr, Jordan Sitkin, Felipe Petroski Such

Trust & safety engineering[11]
Jeff Belgum, Madelaine Boyd, Vik Goel

Trust & safety monitoring and response[11]
Janko Altenschmidt, Anna-Luisa Brakman, Derek Chen, Florencia Leoni Aleman, Molly Lin, Cameron Raymond, CJ Weinmann, Dave Willner, Samuel Wolrich

Trust & safety policy[11]
Rosie Campbell, Kim Malfacini, Andrea Vallone, Dave Willner

Deployment compute[11]
Peter Hoeschele, Evan Morikawa

Product management[11]
Jeff Harris, Joanne Jang, Angela Jiang

Additional contributions

Sam Altman, Katie Mayer, Bob McGrew, Mira Murati, Ilya Sutskever, Peter Welinder[11]

Blog post & paper content[11]
Sandhini Agarwal, Greg Brockman, Miles Brundage, Adrien Ecoffet, Tyna Eloundou, David Farhi, Johannes Heidecke, Shengli Hu, Joost Huizinga, Roger Jiang, Gretchen Krueger, Jan Leike, Daniel Levy, Stephanie Lin, Ryan Lowe, Tong Mu, Hyeonwoo Noh, Jakub Pachocki, Jack Rae, Kendra Rimbach, Shibani Santurkar, Szymon Sidor, Benjamin Sokolowsky, Jie Tang, Chelsea Voss, Kai Xiao, Rowan Zellers, Chong Zhang, Marvin Zhang

Communications[11]
Ruby Chen, Cory Decareaux, Thomas Degry, Steve Dowling, Niko Felix, Elie Georges, Anna Makanju, Andrew Mayne, Aalok Mehta, Elizabeth Proehl, Kendra Rimbach, Natalie Summers, Justin Jay Wang, Hannah Wong

Compute allocation support[11]
Theresa Lopez, Elizabeth Tseng

Contracting, revenue, pricing, & finance support[11]
Brooke Chan, Denny Jin, Billie Jonn, Patricia Lue, Kyla Sheppard, Lauren Workman

Launch partners & product operations[11]
Filipe de Avila Belbute Peres, Brittany Carey, Simón Posada Fishman, Isabella Fulford, Teddy Lee,, Yaniv Markovski, Tolly Powell, Toki Sherbakov, Jessica Shieh, Natalie Staudacher, Preston Tuggle

Legal[11]
Jake Berdine, Che Chang, Sheila Dunning, Ashley Pantuliano

Security & privacy engineering[11]
Kevin Button, Fotis Chantzis, Wade Hickey, Xin Hu, Shino Jomoto, Matt Knight, Jake McNeil, Vinnie Monaco, Joel Parish, Bob Rotsted

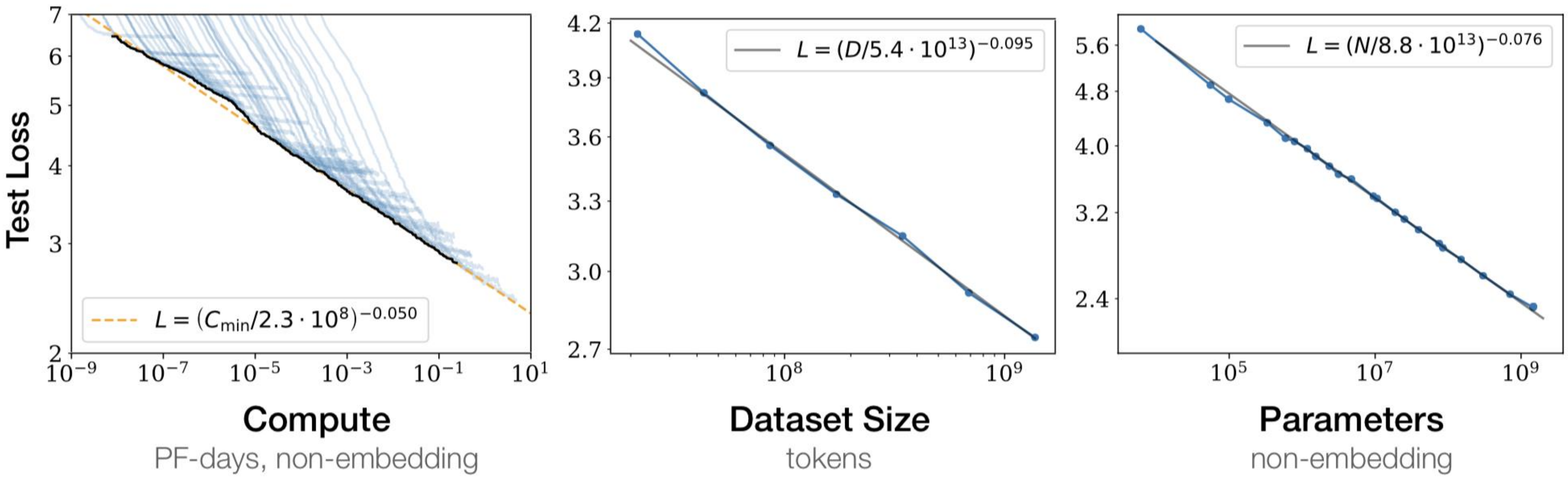
System administration & on-call support[11]
Morgan Grafstein, Francis Real, Mario Saltarelli

Authorship & credit attribution[11]
David Farhi

GPT-4 Technical Report

扩展定律

- 什么是扩展定律
  - 通过扩展参数规模、数据规模和计算算力，大语言模型的能力会出现显著提升
  - 扩展定律在本次大模型浪潮中起到了重要作用



Scaling Laws for Neural Language Models, Arxiv, 2020

扩展定律

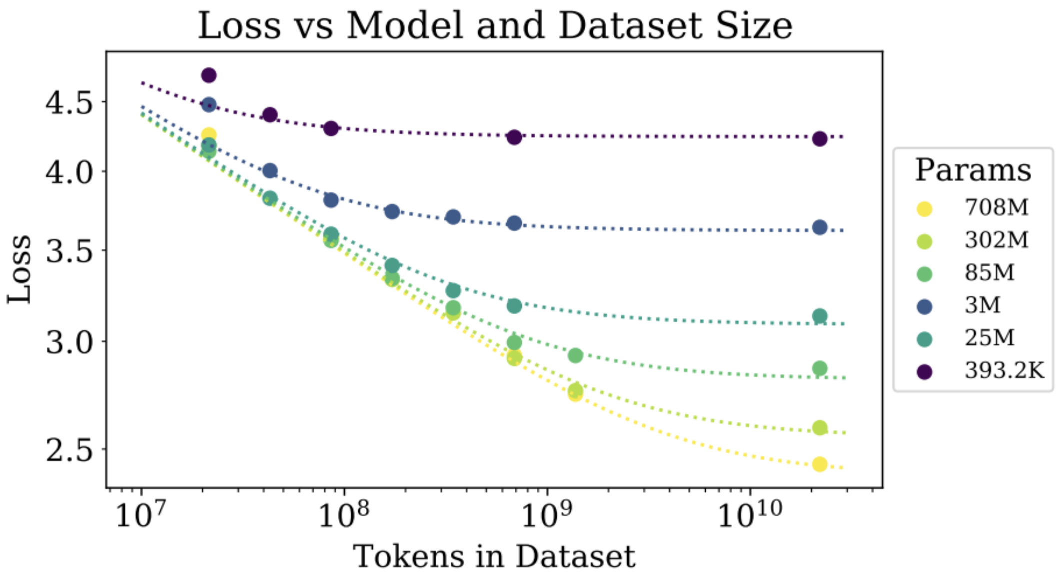
➢ KM扩展定律

➢ OpenAI 团队建立了神经语言模型性能与参数规模（N）、数据规模（D）和计算算力（C）之间的幂律关系

$$L(N) = \left(\frac{N_c}{N}\right)^{\alpha_N}, \quad \alpha_N \sim 0.076, N_c \sim 8.8 \times 10^{13}$$

$$L(D) = \left(\frac{D_c}{D}\right)^{\alpha_D}, \quad \alpha_D \sim 0.095, D_c \sim 5.4 \times 10^{13}$$

$$L(C) = \left(\frac{C_c}{C}\right)^{\alpha_C}, \quad \alpha_C \sim 0.050, C_c \sim 3.1 \times 10^{8}$$



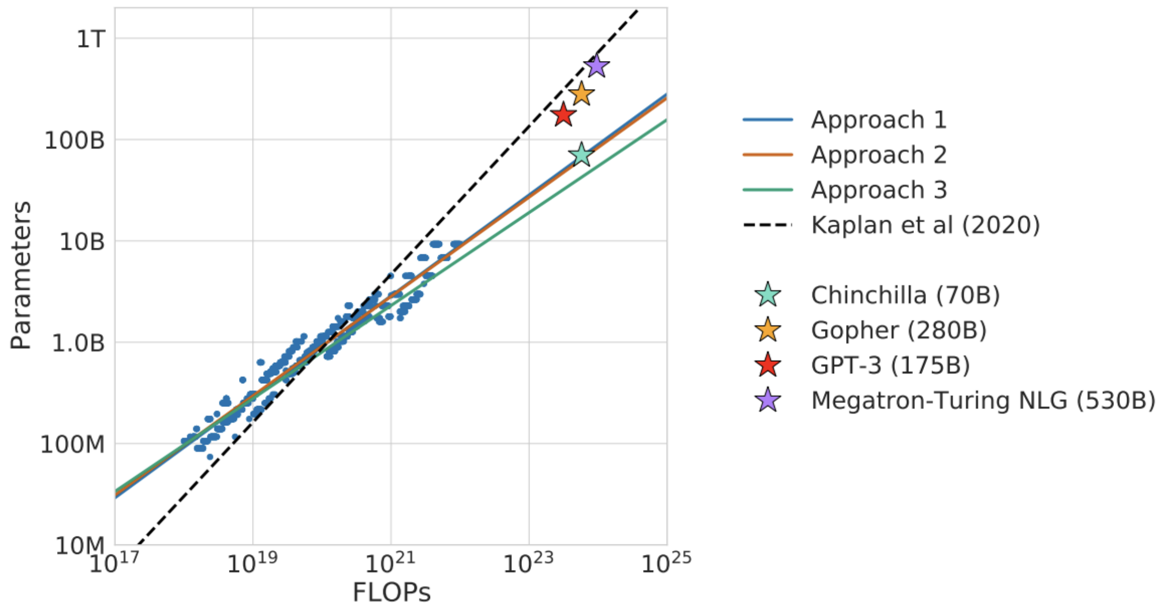
Scaling Laws for Neural Language Models, Arxiv, 2020

扩展定律

- Chinchilla扩展定律
  - DeepMind 团队于 2022 年提出了另一种形式的扩展定律,旨在指导大语言模型充分利用给定的算力资源优化训练

$$L(N, D) = E + \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}},$$

$$N_{\mathrm{opt}}(C) = G\left(\frac{C}{6}\right)^{a}, \quad D_{\mathrm{opt}}(C) = G^{-1}\left(\frac{C}{6}\right)^{b},$$



Training Compute-Optimal Large Language Models, NIPS 2022

扩展定律

AI Box

- 深入讨论
  - 模型的语言建模损失可以进行下述分解

$$L(x) = \underbrace{L_{\infty}}_{\text{不可约损失}} + \underbrace{\left(\frac{x_0}{x}\right)^{\alpha_x}}_{\text{可约损失}}$$

可约损失：真实分布和模型分布之间KL散度，可通过优化减少

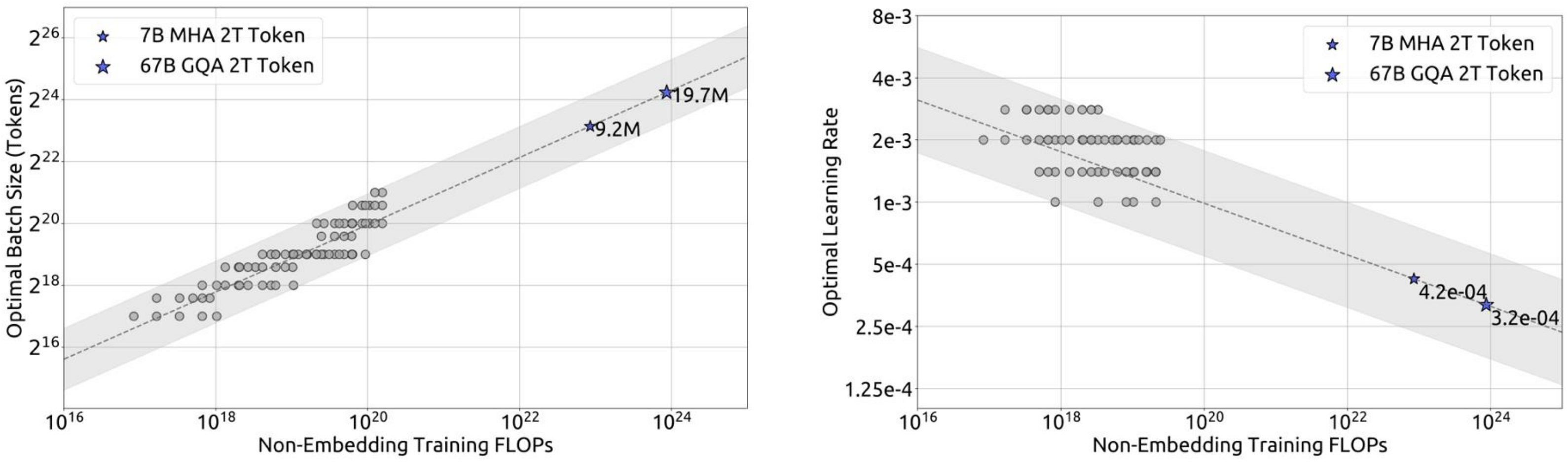
不可约损失：真实数据分布的熵，无法通过优化减少

  - 扩展定律可能存在边际效益递减
    - 随着模型参数、数据数量的扩展，模型性能增益将逐渐减小
    - 目前开放数据已经接近枯竭，难以支持扩展定律的持续推进

扩展定律

AI Box

- 深入讨论
  - 可预测的扩展（Predictable Scaling）
    - 使用小模型性能去预估大模型的性能，或帮助超参数选择
    - 训练过程中使用模型早期性能来预估后续性能

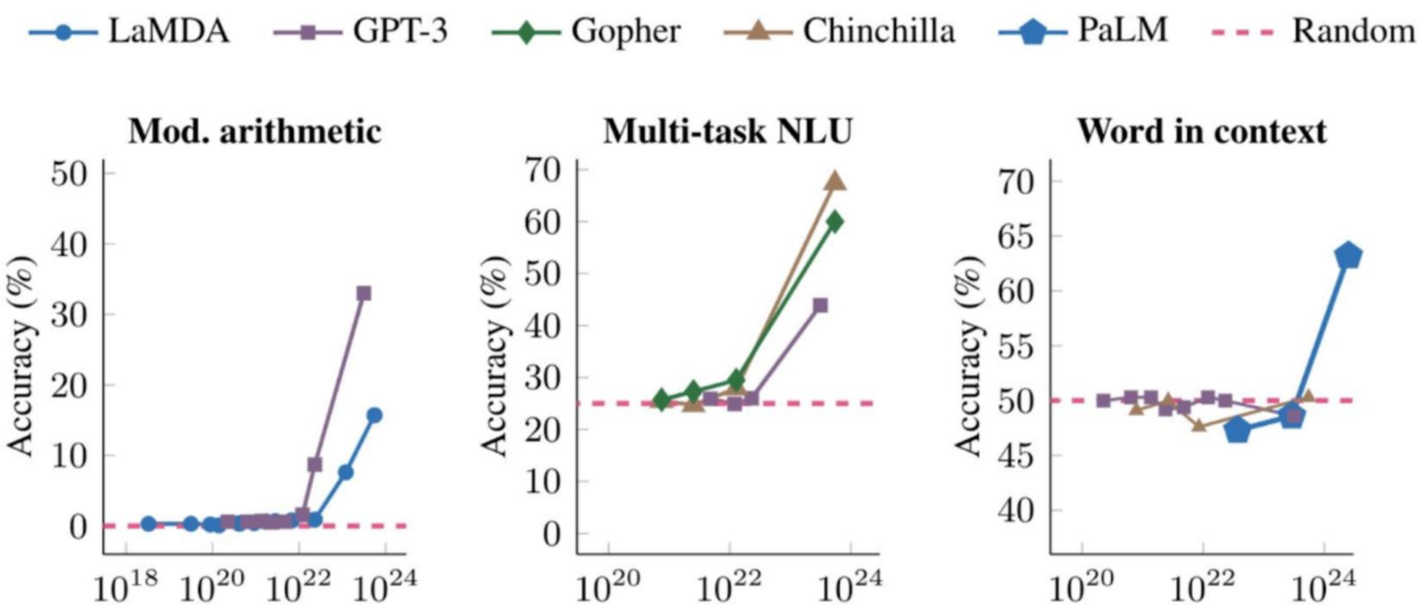


(a) Batch size scaling curve (b) Learning rate scaling curve

DeepSeek LLM Scaling Open-Source Language Models with Longtermism, Arxiv, 2024

涌现能力

- 什么是涌现能力
  - 原始论文定义：“在小型模型中不存在、但在大模型中出现的能力”
  - 模型扩展到一定规模时，特定任务性能突然出现显著跃升趋势，远超随机水平



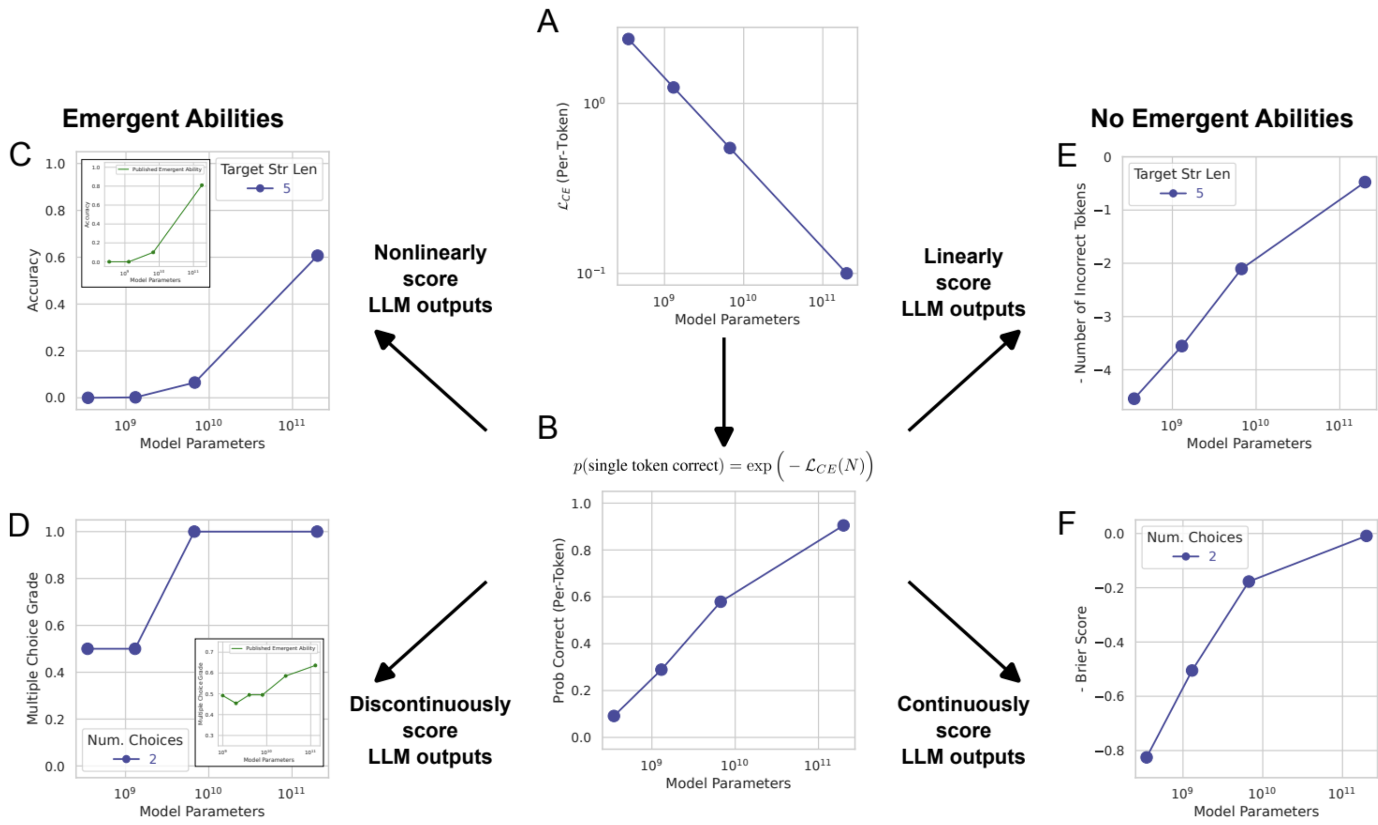
Emergent Abilities of Large Language Models, TMLR 2022

涌现能力

AI Box

➢ 涌现能力可能部分归因于评测设置

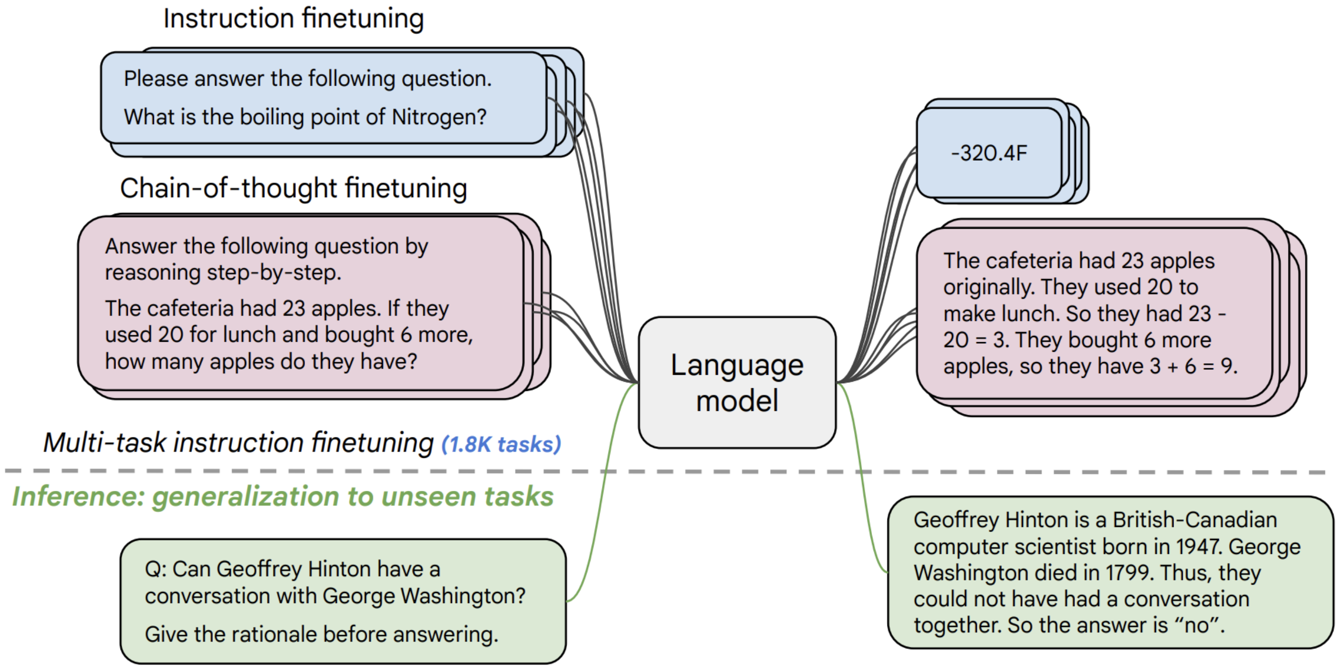
➢ 本课程定义其为“**代表性能力**”，并不区分是否在小模型中存在



Are Emergent Abilities of Large Language Models a Mirage?, NIPS 2023

涌现能力

- 代表性能力
  - 指令遵循（Instruction Following）
    - 大语言模型能够按照自然语言指令来执行对应的任务



Scaling Instruction-Finetuned Language Models, JMLR 2024

涌现能力

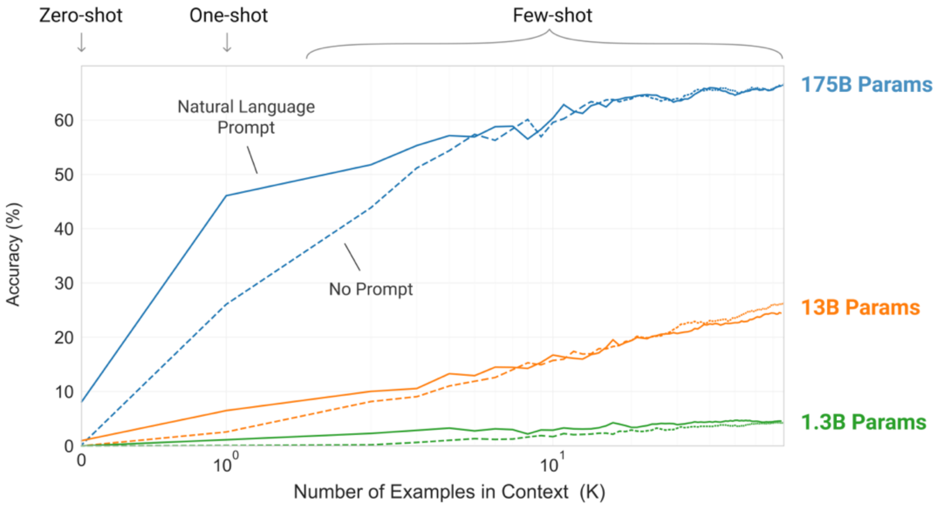
- 代表性能力
  - 上下文学习（In-context Learning）
    - 在提示中为语言模型提供自然语言指令和任务示例，无需显式梯度更新就能为测试样本生成预期输出

Answer the following mathematical reasoning questions:

*N*x

Q: If you have 12 candies and you give 4 candies to your friend, how many candies do you have left?
A: The answer is 8.
Q: If a rectangle has a length of 6 cm and a width of 3 cm, what is the perimeter of the rectangle?
A: The answer is 18cm.

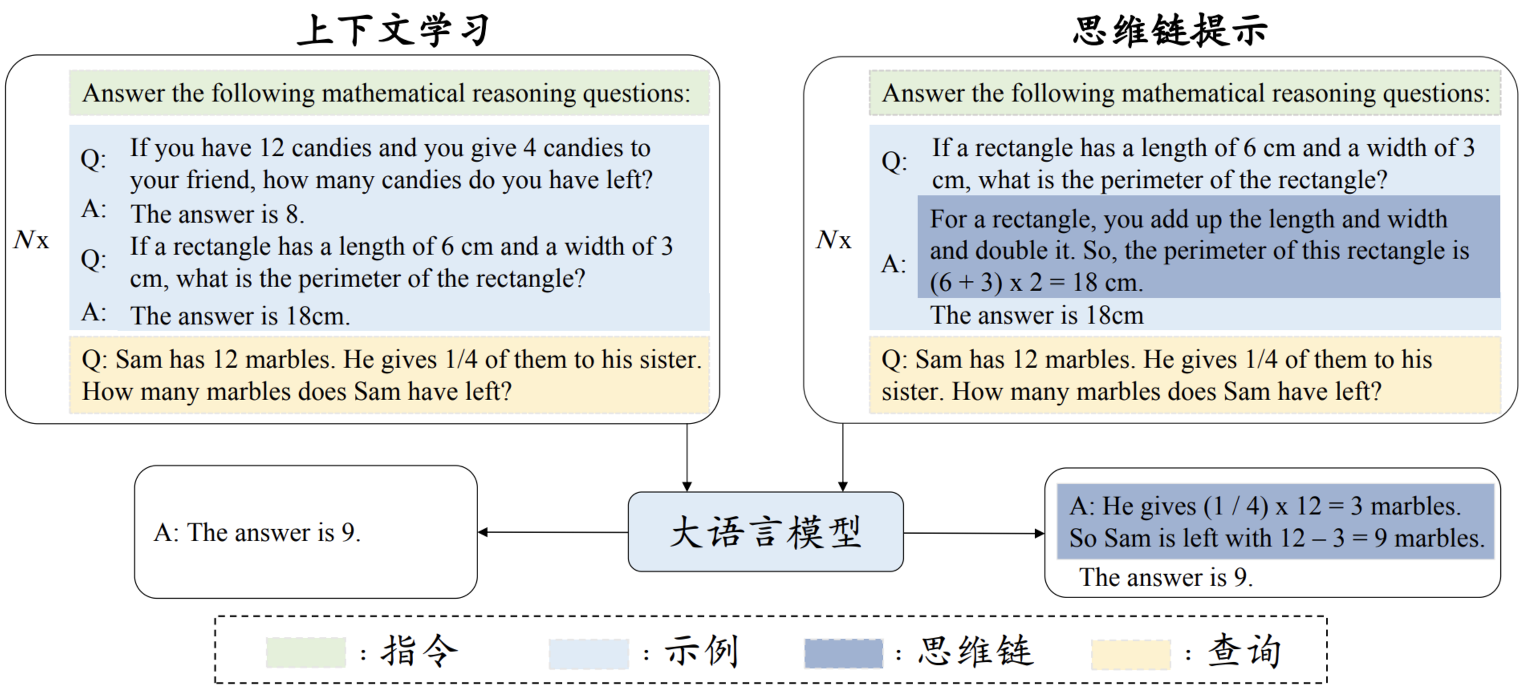
Q: Sam has 12 marbles. He gives 1/4 of them to his sister. How many marbles does Sam have left?



Language Models are Few-shot Learners, NIPS 2020

涌现能力

- 代表性能力
  - 逐步推理（Step-by-step Reasoning）
    - 在提示中引入任务相关的中间推理步骤来加强复杂任务的求解，从而获得更可靠的答案

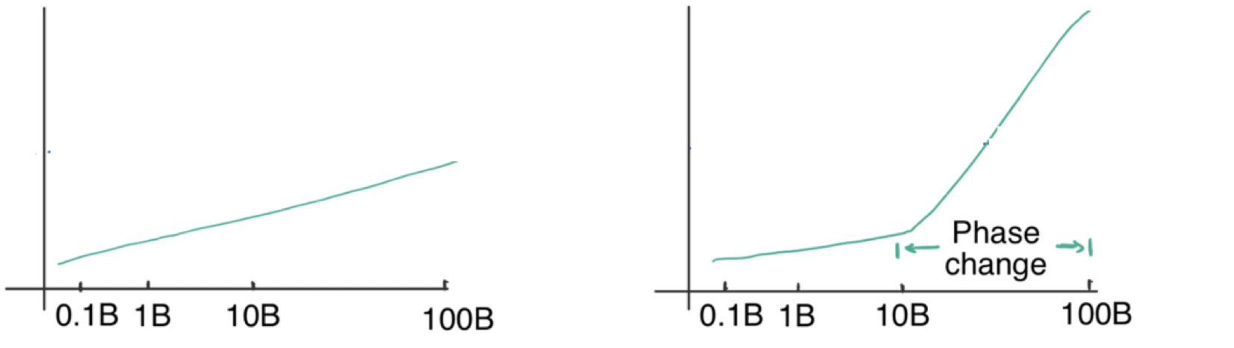


涌现能力

- 涌现能力与扩展定律的关系
  - 涌现能力和扩展定律是两种描述规模效应的度量方法

Scaling law describes a *predictable* increase pattern, but with *diminishing return*

可以理解为是一种较为平滑的多任务损失平均（LM loss）



Scaling Law (~2020 - 2021)　　Emergent Ability (from early 2022)

Model scaling is the key to the emergence of strong abilities

Emergent abilities transcend the scaling law, making the increase *unpredictable but profitable*

非平滑的、某种特定能力或任务的性能跃升（Task loss）

Figure source：https://yaofu.notion.site/A-Closer-Look-at-Large-Language-Models-Emergent-Abilities-493876b55df5479d80686f68a1abd72f

总结

AI Box

- 大模型核心技术
  - 规模扩展：扩展定律奠定了早期大模型的技术路线，产生了巨大的性能提升
  - 数据工程：数据数量、数据质量以及配制方法极其关键
  - 高效预训练：需要建立可预测、可扩展的大规模训练架构
  - 能力激发：预训练后可以通过微调、对齐、提示工程等技术进行能力激活
  - 人类对齐：需要设计对齐技术减少模型使用风险，并进一步提升模型性能
  - 工具使用：使用外部工具加强模型的弱点，拓展其能力范围

谢谢