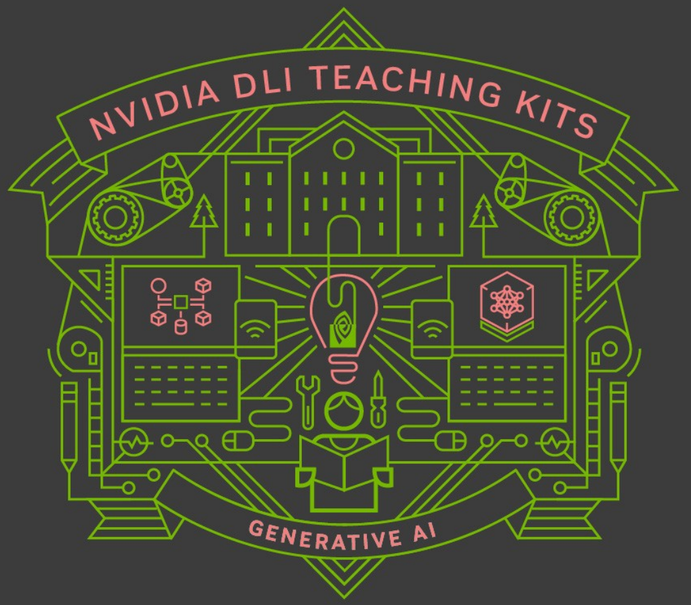
Lecture 4.2 - Scaling Laws in Training LLMs

Generative AI Teaching Kit



The NVIDIA Deep Learning Institute Generative AI Teaching Kit is licensed by NVIDIA and Dartmouth College under the Creative Commons Attribution-NonCommercial 4.0 International License.

This lecture

- Emergence of Scale-related performance in LLMs
- Balance of Model Size and Dataset Size
- Emergent behavior at different scales of training

DARTMOUTH ENGINEERING | NVIDIA

Emergence of Scale-Related Performance in LLMs

What are Scaling Laws?

Scaling laws describe how system performance or feature changes as we change key resources. These are found in many fields: physics, economics, biology, and machine learning. Often they follow power-law relationships meaning performance improves predictably with larger inputs.

Typically written as: $\text{Performance} \propto (\text{Resource})^{\alpha}$

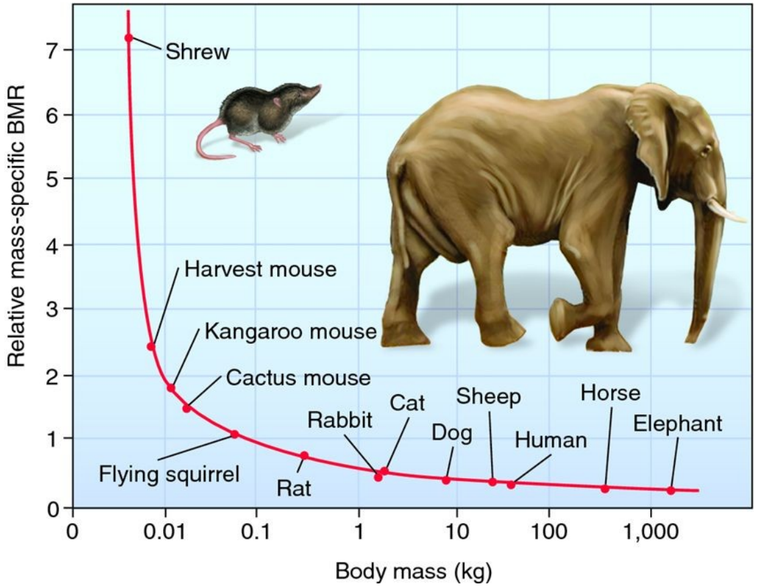
where α is the scaling exponent that determines how strongly performance improves with resource increases.

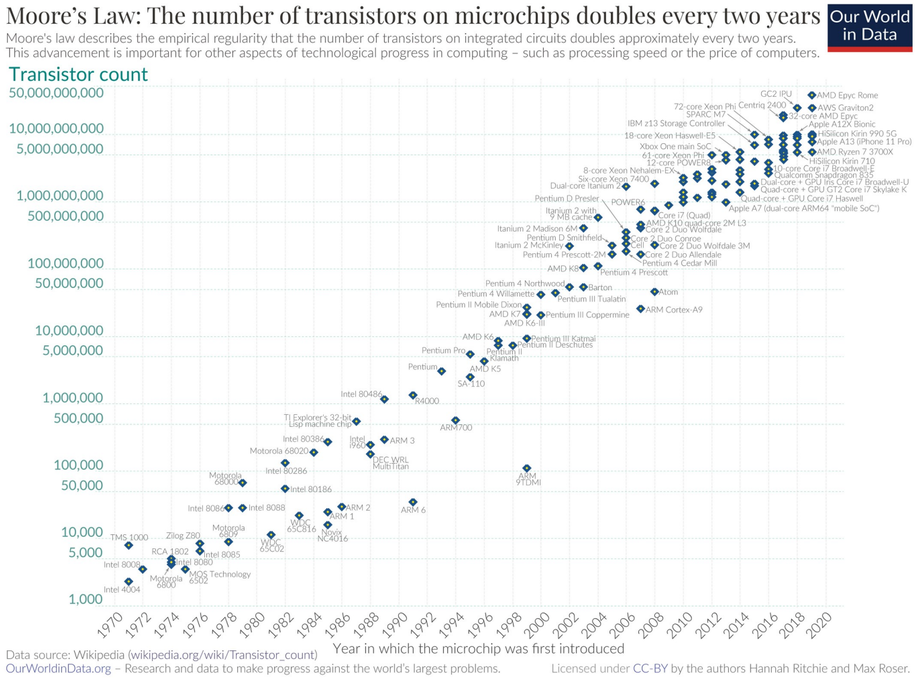
Examples in Other Fields

- Physics: Power output of an engine scales with volume.
- Economics Productivity scales with workforce size and capital.
- Computing: Processing speed scales with transistor count (Moore's Law)

Why Scaling Laws Matter

- Help predict how far improvements can go before hitting diminishing returns.
- Provide a theoretical foundation for making efficient resource allocation decisions.





LLM Scaling Laws

Researchers discovered that LLM performance improves predictably as model size, dataset size, and compute resources scale up. This follows a power-law relationship where training loss decreases smoothly with increased compute:

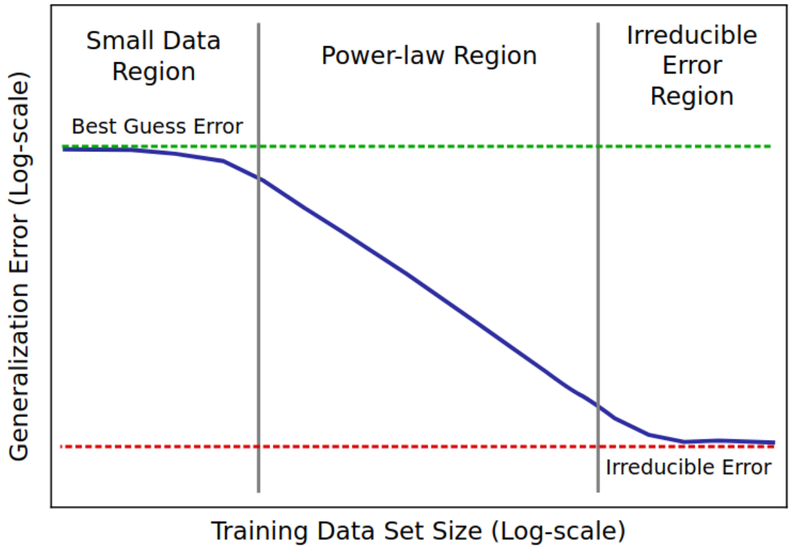
$$L(C) \propto C^{-\alpha}$$

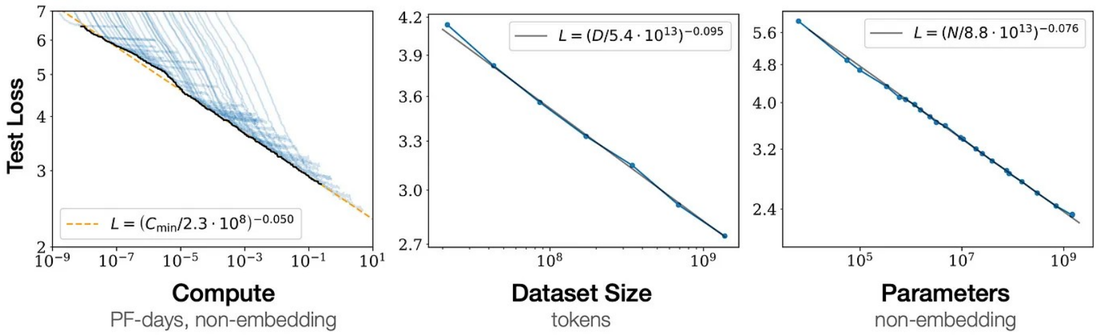
where C is compute and α is an empirically determined scaling exponent.

One key insight was that simply making models bigger isn't always efficient. Instead, scaling must be balanced across three factors:

- **Model Size**: Larger models generalize better but need more data.
- **Dataset Size**: Too little data limits model performance; too much training on the same data causes overfitting.
- **Compute Budget**: More compute lowers loss, but **waste occurs** if the model or data isn't scaled accordingly.

This led to the concept of compute-optimal scaling where a given compute budget is used efficiently. Under-scaled models underutilize hardware, while over-scaled models waste parameters and overfit. The discovery of these laws provided a roadmap for training efficient LLMs, guiding how to balance model, data, and compute for optimal performance.





Using Scaling Laws to Guide Model Training

The discovery of scaling laws provided a framework for training efficient and well-balanced LLMs. Instead of arbitrarily increasing model size, researchers learned that optimal performance depends on the right balance of compute, model size, and data.

Balance Model, Data, and Compute

- Increasing model size requires more training data to avoid overfitting.
- Compute should be allocated efficiently to avoid underutilization or waste.

Compute-Optimal Scaling

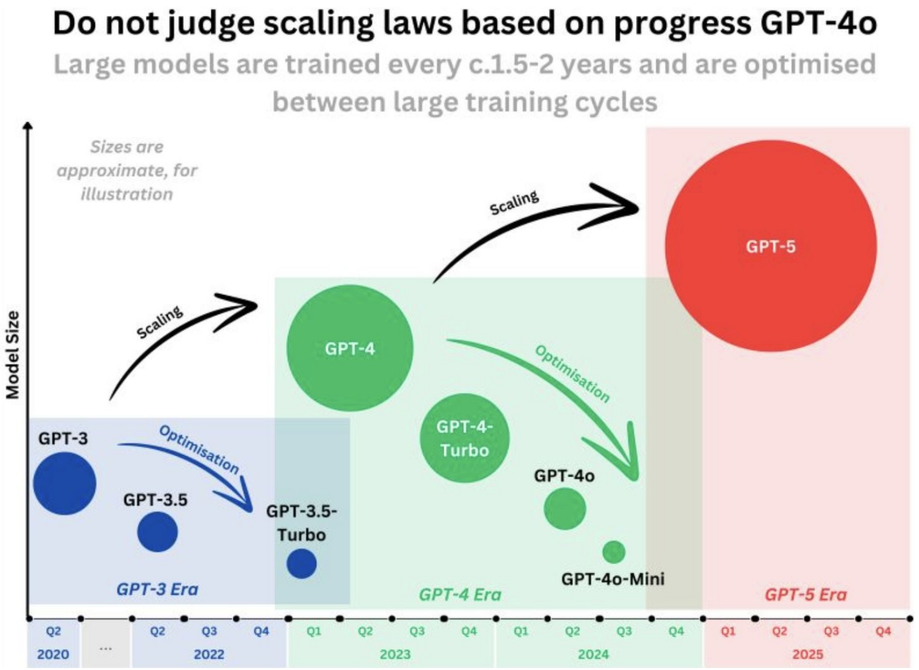
- Given a fixed compute budget, models should be sized appropriately for maximum efficiency.
- Overly large models waste parameters, while under-scaled models fail to use available compute fully.

Estimating Performance Gains

- Scaling laws allow researchers to predict improvements before training.
- Helps determine whether investing in more compute or data is worth the expected gain.

Avoiding Inefficient Scaling

- Too small a dataset limits generalization.
- Too large a model without sufficient data leads to overfitting and inefficiency.



.

Balance of Model Size and Dataset Size

Finding the Right Balance – Model Size vs. Dataset Size

Why Does This Trade-off Matter?
When training large language models, it's tempting to believe that bigger is always better—that simply adding more parameters will lead to better performance. But this isn't necessarily true. The relationship between model size and dataset size plays a critical role in how efficiently a model learns and generalizes.

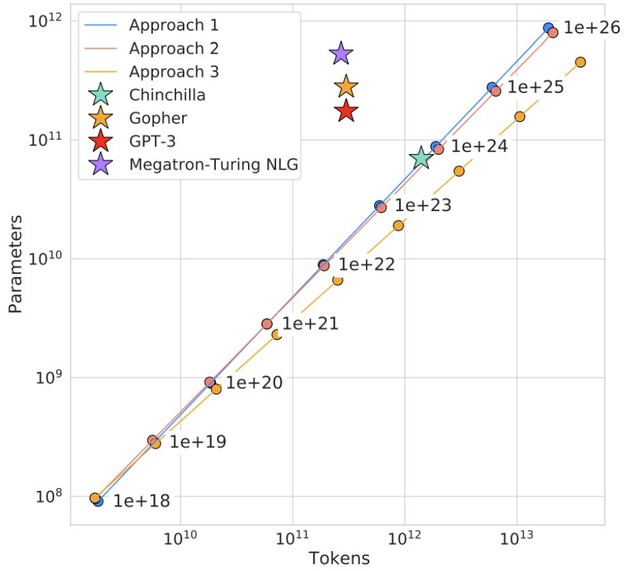
The Problem: Scaling Without Balance

- If we increase model size without enough data, the model struggles to learn meaningful patterns, leading to overfitting or wasted compute.
- If we increase dataset size without a sufficiently large model, the model lacks the capacity to take full advantage of the data.

DeepMind's "Chinchilla" Insight
For years, the industry was focused on scaling model size as the primary way to improve performance. But DeepMind's research showed that this approach is suboptimal under a fixed compute budget. Instead, they found that:

- Doubling the dataset size while halving the model size leads to better overall performance than simply making the model bigger.
- This challenges the conventional wisdom that larger models are always better, showing that data efficiency matters just as much as model capacity.

What This Means for Model Training
More compute should be spent on training with more data, not just bigger models. Scaling strategies need to account for both model and data growth, not just one or the other.



Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
Chinchilla	70 Billion	1.4 Trillion

Compute-Optimal Training – Llama 3

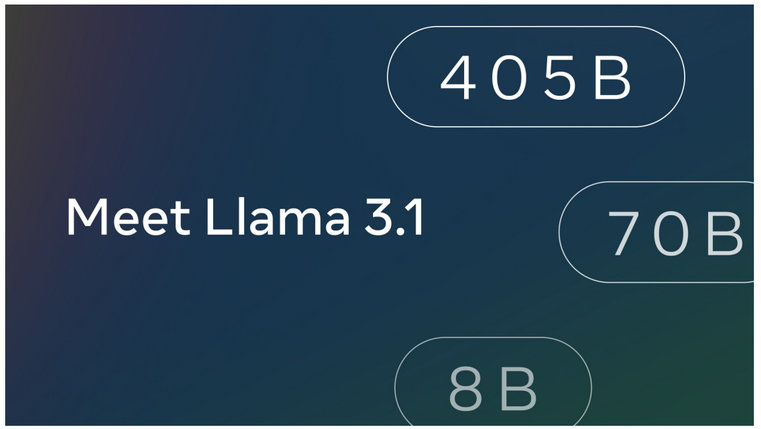
What Makes a Model Compute-Optimal?
A compute-optimal model is one that balances parameter count and training steps within a fixed compute budget. Instead of focusing purely on increasing model size, the goal is to maximize learning efficiency by properly scaling both model parameters and training data.

The Shift in Scaling – LLaMA 3's Approach
Historically, model scaling focused on increasing parameter count, but recent research—including Meta's LLaMA 3—suggests that token count (training data) plays a much bigger role in improving performance.

- LLaMA 3 didn't just increase model size—it trained on significantly more tokens.
- More tokens allow models to generalize better without excessive parameter growth.
- This follows the Chinchilla scaling laws, which emphasize balancing model size and dataset size for a given compute budget.

Why More Tokens Instead of a Bigger Model?

- *Diminishing returns from larger models*: Simply increasing parameters without enough training data leads to inefficiencies.
- *Training longer is often better than making the model bigger*: More training steps on high-quality data yield better results than scaling parameters alone.
- *Compute constraints favor data scaling*: For the same compute budget, a moderate-sized model trained on more tokens often outperforms a much larger model trained on fewer tokens.



Key Takeaways for Model Training

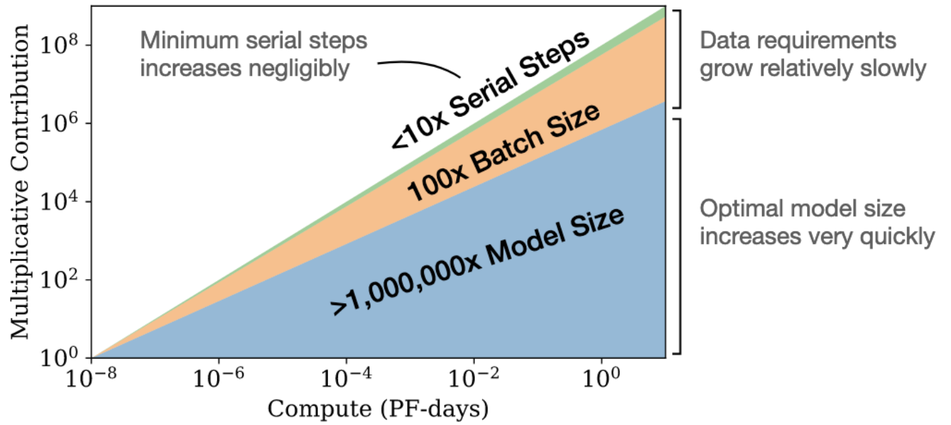
- LLaMA 3 reinforces that optimal scaling isn't just about bigger models—it's about training longer on more data.
- Balancing model size and dataset size is crucial for efficient learning.
- Future models will likely continue prioritizing longer training over sheer parameter growth.

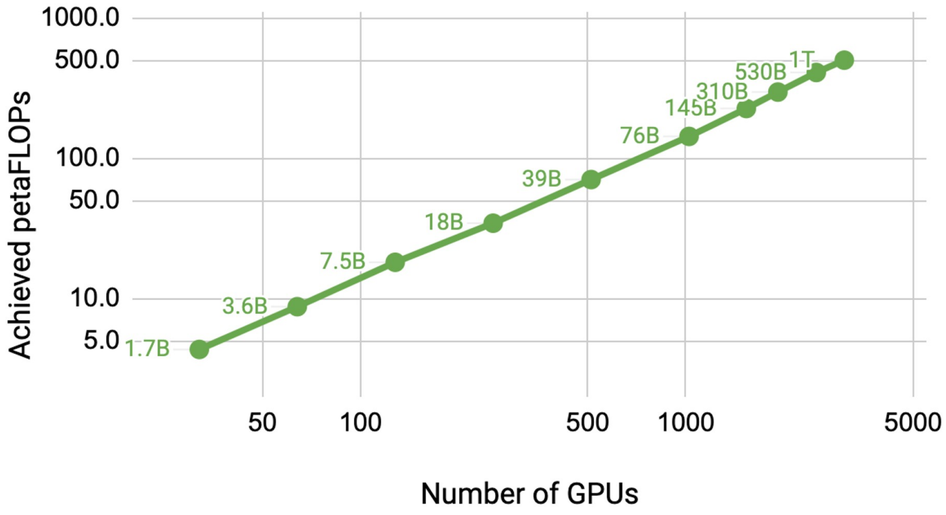
Scaling Trade-Offs

Why Scaling is Not Straightforward
Larger models do not scale linearly—they require exponentially more compute as parameter counts grow. While scaling up models can improve performance, practical constraints force trade-offs in several key areas.

Key Scaling Challenges
1. Memory Requirements
Larger models demand more VRAM and storage for parameters, activations, and gradients.
Training on GPUs or TPUs requires careful memory optimization techniques (e.g., tensor parallelism, activation checkpointing).
2. Training Time
A model twice as large doesn't just take twice as long to train—it often takes much more due to compute bottlenecks.
Distributed training is necessary but comes with communication overhead and efficiency losses.
3. Dataset Quality
Scaling models without increasing high-quality data leads to diminishing returns.
Curating high-quality datasets is often more important than simply increasing model size.

Practical Scaling Strategies
- Smaller models trained longer on high-quality, diverse datasets can outperform brute-force parameter scaling.
- Efficient training techniques (e.g., flash attention, quantization, and optimizer improvements) help mitigate scaling costs.
- Meta's LLaMA 3 and DeepMind's Chinchilla both show that balancing model size and dataset size leads to better results under a fixed compute budget.

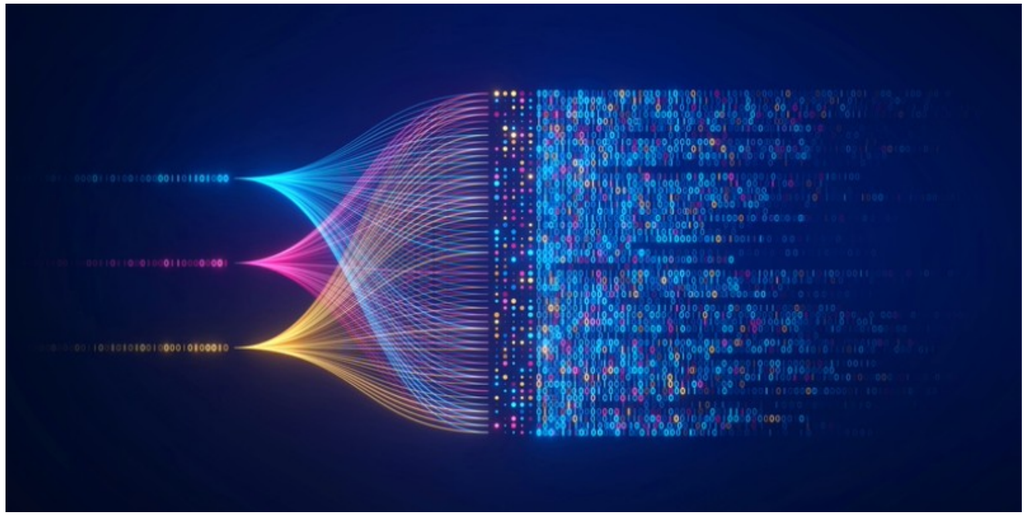




Emergent Behavior at Different Scales of Training

Definition of Emergent Behavior

What is Emergent Behavior?
Emergent behaviors are capabilities that arise spontaneously when a model scales beyond a certain threshold, despite not being explicitly trained for those tasks. These behaviors are often not present in smaller models and appear unexpectedly as the model size and training data increase.



Key Examples of Emergent Abilities
1. Arithmetic Reasoning
Smaller models struggle with basic math, but larger models develop arithmetic skills even without explicit programming.
2. Zero-Shot and Few-Shot Learning
Large models can perform tasks without fine-tuning, simply by seeing a well-structured prompt.
3. Complex Natural Language Understanding
Summarization, translation, and code generation become significantly better at scale.

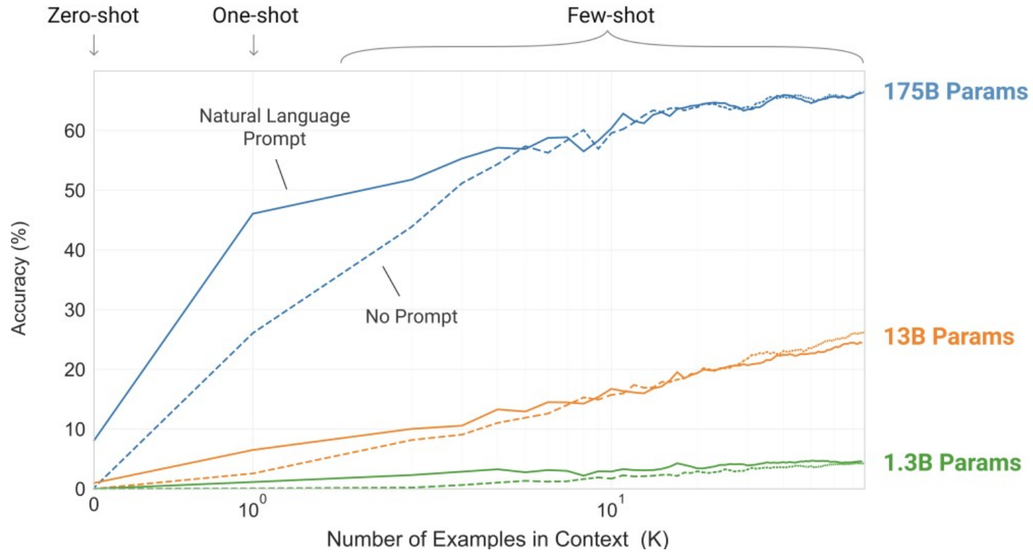
These tasks are not explicitly trained but emerge as the model develops stronger generalization skills.

Why Does This Happen?
- Scale unlocks new representations: More parameters allow for richer abstractions in data.
- More data improves generalization: Large models absorb patterns and relationships beyond memorization.
- Transformer-based architectures support reasoning: Attention mechanisms enable models to track dependencies and logical structures more effectively.

Examples of Emergent Behaviors

Scaling Unlocks Unexpected Capabilities
GPT-3 was one of the first to highlight emergent behaviors—capabilities that arise only when models surpass a certain scale. These findings reshaped how we think about AI training and model size.

Key Observations from GPT-3:
- Before GPT-3: Most NLP models required task-specific fine-tuning.
- With GPT-3: Models could perform new tasks with just a few examples in the prompt (few-shot) or even without any examples (zero-shot).
- Smaller models failed at simple math, but GPT-3 could solve multi-step arithmetic problems with better accuracy than chance.
- GPT-3 demonstrated coherent essay writing, summarization, and translation, often outperforming task-specific fine-tuned models.
- Its responses felt more contextually aware, suggesting an improved grasp of nuanced language patterns.



Impact on AI Research
Emergent behaviors suggest that scale alone—without task-specific fine-tuning—can unlock new abilities.
This insight influenced later models, including GPT-4 and LLaMA 3, where increased data and longer training replaced pure parameter growth as the key to improvement.

Emergent Abilities in Large Language Models

What Are Emergent Abilities?

- Capabilities that suddenly appear in LLMs once they reach a certain scale.
- Not explicitly trained, but arise from model size and data richness.

1. Chain of Thought (CoT) Reasoning

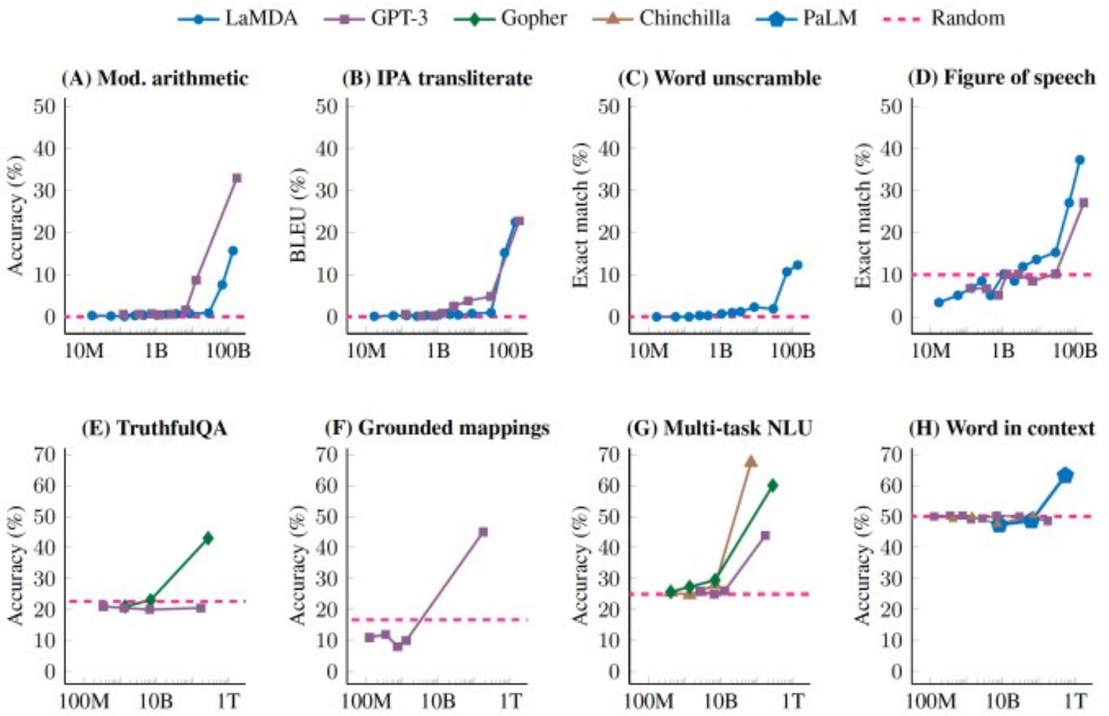
- Small models give intuitive but incorrect answers to logic problems.
- Larger models can break problems into intermediate steps, improving accuracy.
- *Example:* Solving multi-step math problems correctly.

2. In-Context Learning (ICL)

- Larger models can learn tasks on-the-fly from a few examples in a prompt.
- *Example:* Given a few translation pairs, the model infers new translations correctly.

3. Programming and Code Generation

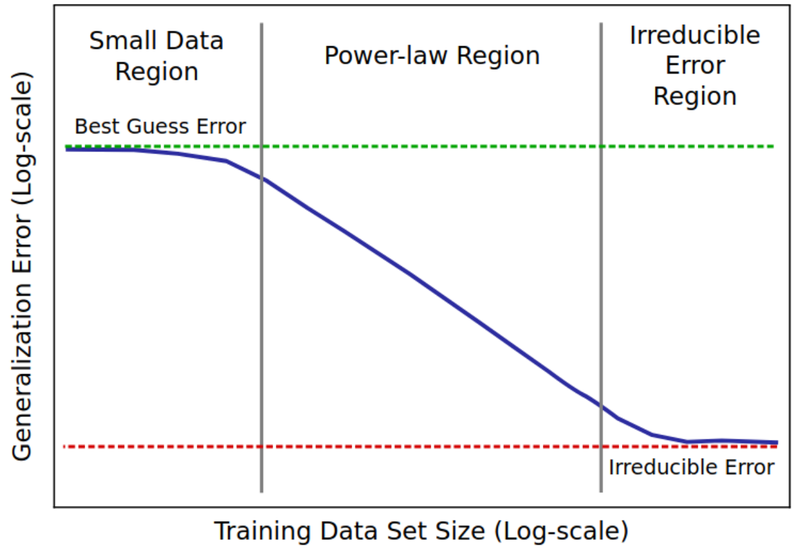
- Large models write structured, executable code with minimal instruction.
- Some can even self-debug when prompted.

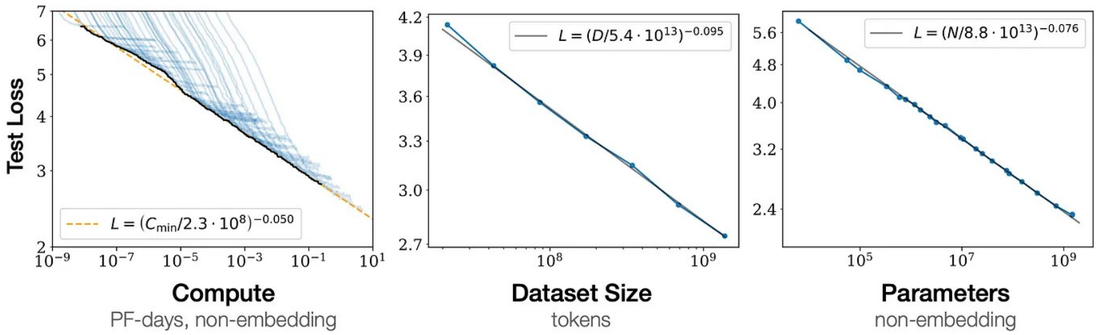


Wrap Up

Scaling Laws in Training LLMs

- Today we introduced the concept of scaling laws in the context of training LLMs.
- We saw that scaling laws are present in deep learning models in general, and have wide reaching implications for data use and compute requirement
- The chinchilla work provided guidance on how many tokens a LLM can make the best use of for a given compute budget
- Finally, we saw that scaling laws allow for emergent properties and abilities to be observed when training larger and larger models.





Thank you!