



# Probabilistic Graphical Models & Probabilistic AI

Ben Lengerich

Lecture 22: Unsupervised Pretraining of LLMs

April 15, 2025

Reading: See course homepage



# Final Opportunity for Bonus Points

- We'll have two days of project presentations: Apr 29, May 1
- **Project Peer Review [[template on website](#)]**
  - Come to class
  - Fill out a peer review form each day (2 teams each day)
  - Submit on Canvas
  - Earn 1% extra bonus points each day (up to 2% total)
  - Due by Friday, May 2

## Peer Group A

(10 points)

Answer the following questions for the first peer group to whom you want to give feedback.

- Group Number:** Use the order of presentation for the day.
- Summary:** What was the central problem or question addressed? How did they approach it? (2-3 sentences)
- Strengths:** What worked well and why? (2-3 sentences)
- Suggestions:** What could be improved and how? (2-3 sentences)
- Question to Presenter:** A thoughtful question (1 sentence).



# Today

---

- Announcement: Final opportunity for bonus points
- Unsupervised training of LLMs
  - Emergent Capabilities
  - Challenges of MLE-based unsupervised training

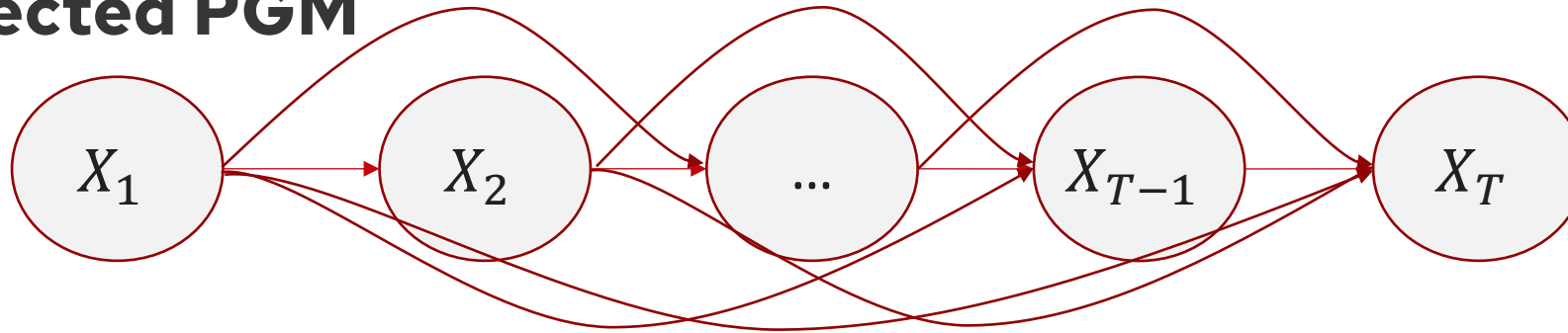


# Unsupervised Training of LLMs



# Recall GPT training objective: MLE

- **Directed PGM**



$$P_{\theta}(X) = \prod_i \prod_t P_{\theta}(X_{i,t} | X_{i,<t})$$

- **Probabilistic objective:** Max log-likelihood of observed seqs

$$\max_{\theta} \sum_i \sum_t \log P_{\theta}(X_{i,t} | X_{i,<t})$$

[Radford et al., [Improving Language Understanding by Generative Pre-Training](#)]



# We've had MLE-based Language Models for a while...

## Large Language Models in Machine Translation **2007**

**Thorsten Brants   Ashok C. Popat   Peng Xu   Franz J. Och   Jeffrey Dean**

Google, Inc.  
1600 Amphitheatre Parkway  
Mountain View, CA 94303, USA  
{brants, popat, xp, och, jeff}@google.com



Some fun:

<https://github.com/LRitzdorf/TheJeffDeanFacts>



# We've had MLE-based Language Models for a while...

## Large Language Models in Machine Translation 2007

This paper reports on the benefits of large-scale statistical language modeling in machine translation. A distributed infrastructure is proposed which we use to train on up to 2 trillion tokens, resulting in language models having up to 300 billion  $n$ -grams. It is capable of providing smoothed probabilities for fast, single-pass decoding. We introduce a new smoothing method, dubbed *Stupid Backoff*, that is inexpensive to train on large data sets and approaches the quality of Kneser-Ney Smoothing as the amount of training data increases.

Yoon Kim, Frank C. Popat, Peng Xu, Franz J. Och, Jeffrey Dean

Google, Inc.

1600 Amphitheatre Parkway  
Mountain View, CA 94303, USA  
{popat, xp, och, jeff}@google.com

$$P(w_1^L) = \prod_{i=1}^L P(w_i | w_1^{i-1}) \approx \prod_{i=1}^L \hat{P}(w_i | w_{i-n+1}^{i-1})$$



# We've had MLE-based Language Models for a while...

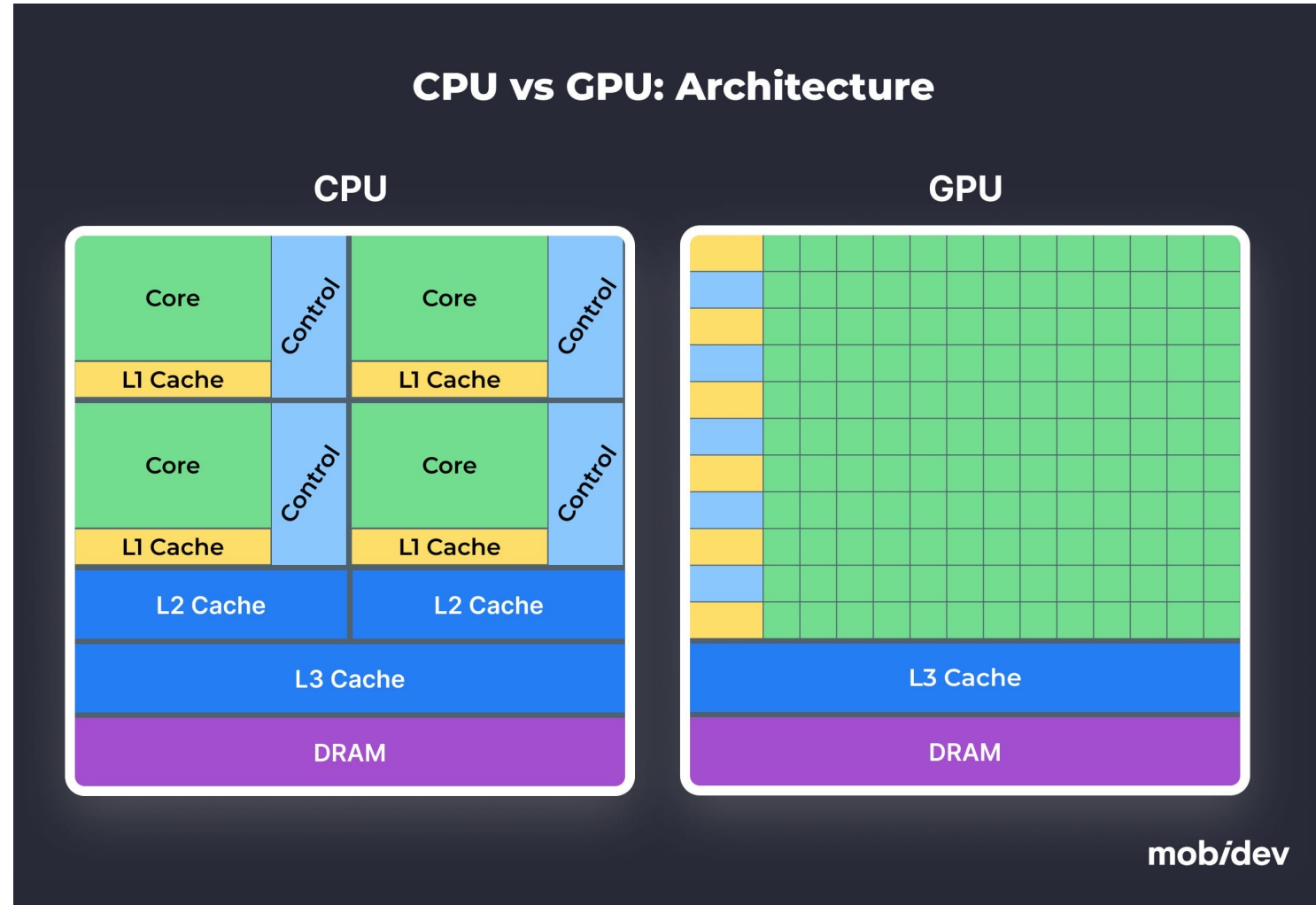
## Large Language Models in Machine Translation **2007**

**Thorsten Brants   Ashok C. Popat   Peng Xu   Franz J. Och   Jeffrey Dean**

Google, Inc.  
1600 Amphitheatre Parkway  
Mountain View, CA 94303, USA  
{brants,popat,xp,och,jeff}@google.com

- Why wasn't this the LLM moment?
  - Modeled n-grams, not *relationships* of token embeddings
  - Transformer architecture
  - Scale, GPU acceleration

# Why GPUs?



<https://mobidev.biz/blog/gpu-machine-learning-on-premises-vs-cloud>



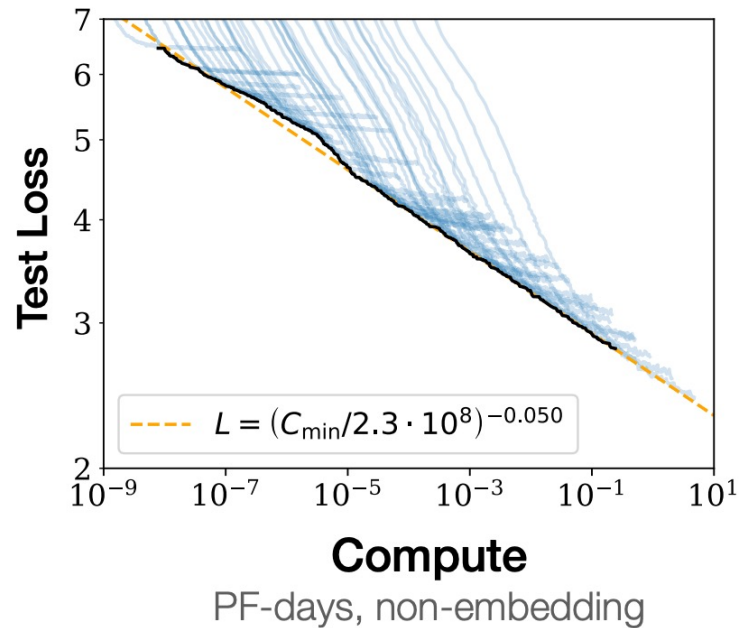
# MLE → Emergent Understanding?

- "The simplest way to predict the next token is to understand what happened throughout the context."
- To predict the word "is" in "*The capital of France \_\_\_ Paris.*", the model must:
  - Resolve subject-verb agreement
  - Recognize a factual structure
  - Know the topic is geography



# Scale & Emergent Capabilities

# What happens as we scale training?

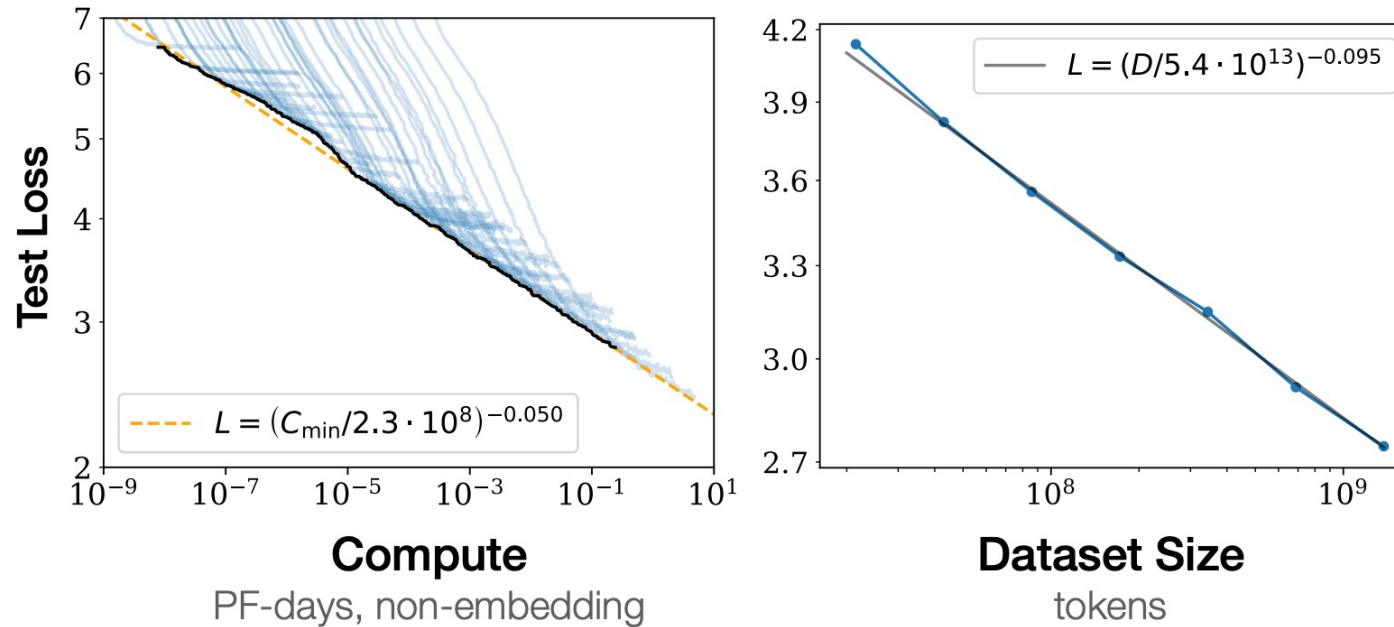


**Figure 1** Language modeling performance improves smoothly as we increase the model size, dataset size, and amount of compute<sup>2</sup> used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

“Scaling Laws for Neural Language Models”. Kaplan et al 2021



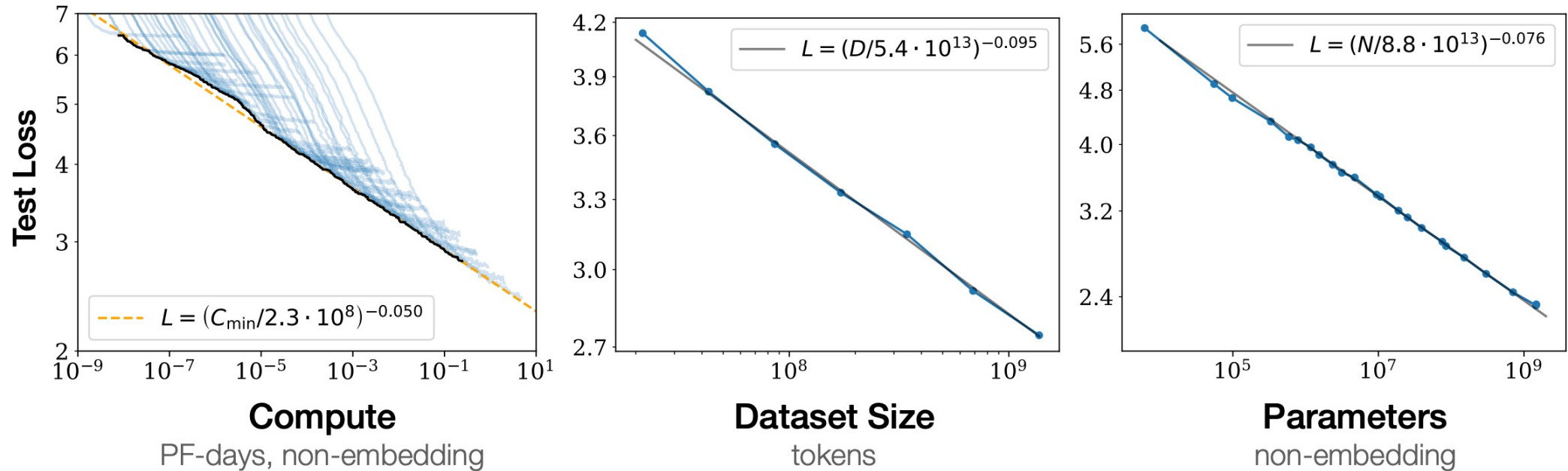
# What happens as we scale training?



**Figure 1** Language modeling performance improves smoothly as we increase the model size, dataset size, and amount of compute<sup>2</sup> used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

“Scaling Laws for Neural Language Models”. Kaplan et al 2021

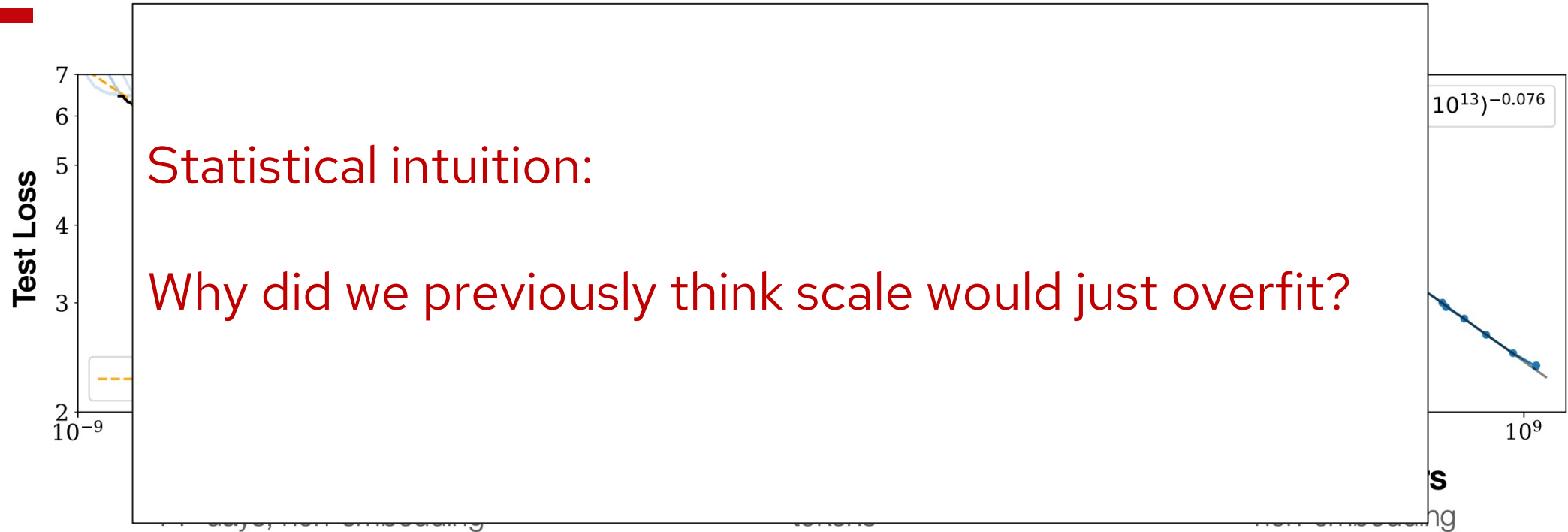
# What happens as we scale training?



**Figure 1** Language modeling performance improves smoothly as we increase the model size, dataset size, and amount of compute<sup>2</sup> used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

“Scaling Laws for Neural Language Models”. Kaplan et al 2021

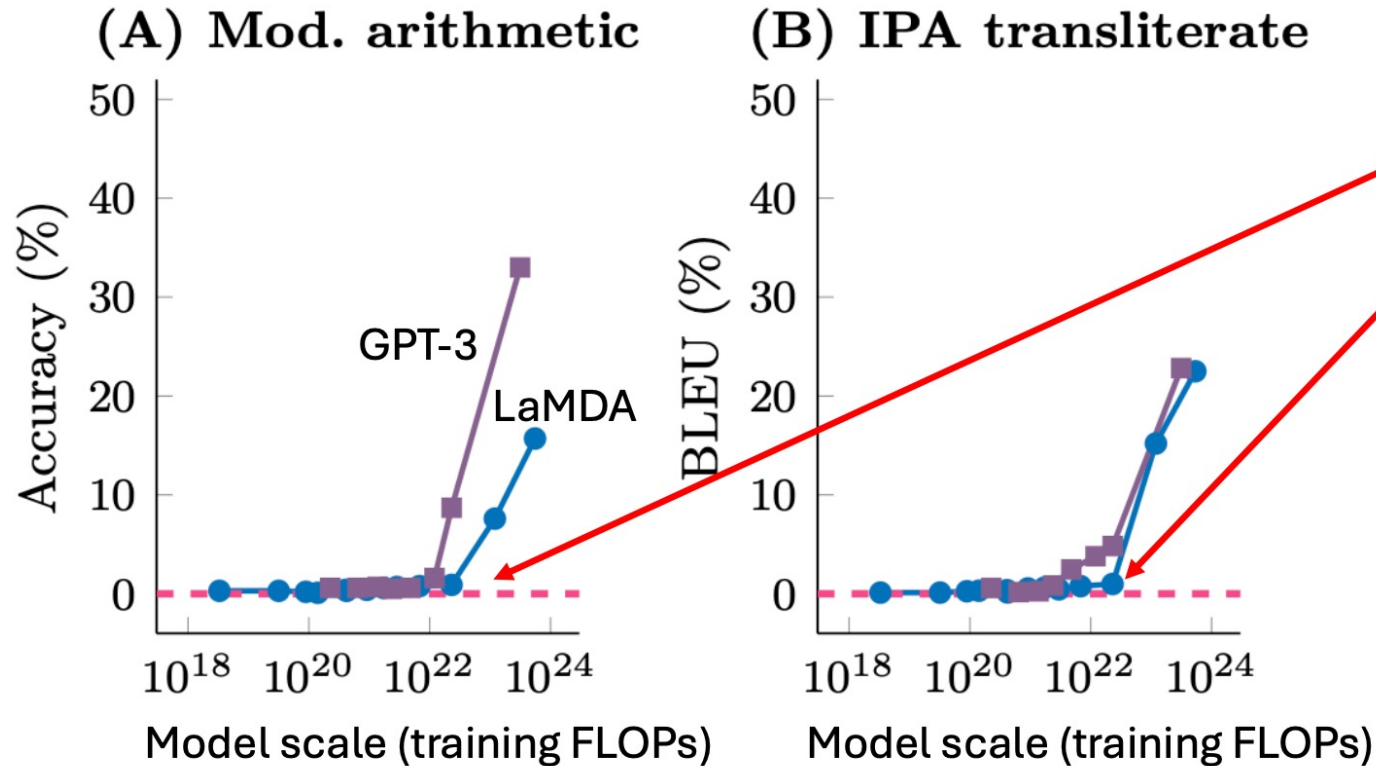
# What happens as we scale training?



**Figure 1** Language modeling performance improves smoothly as we increase the model size, dataset size, and amount of compute<sup>2</sup> used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

“Scaling Laws for Neural Language Models”. Kaplan et al 2021

# Smooth improvements → sharp emergent ability?



An ability is emergent if it is not present in smaller models but is present in larger models [Wei, et al (2022). Emergent Abilities of Large Language Models]

# Example of emergence: In-Context Learning

I: Instruction

Translate English to French

E1: Example1

[en]: A discomfort which lasts.

[fr]: Un malaise qui dure

E2: Example2

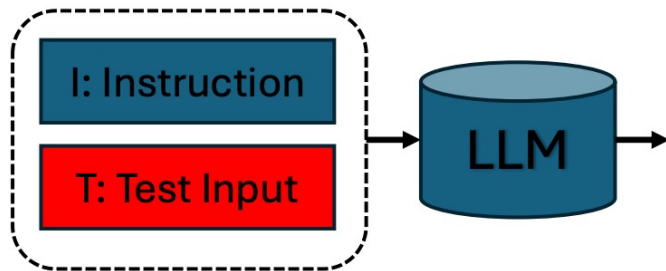
[en]: HTML is a language for formatting  
formatage

[fr]: HTML est un langage de

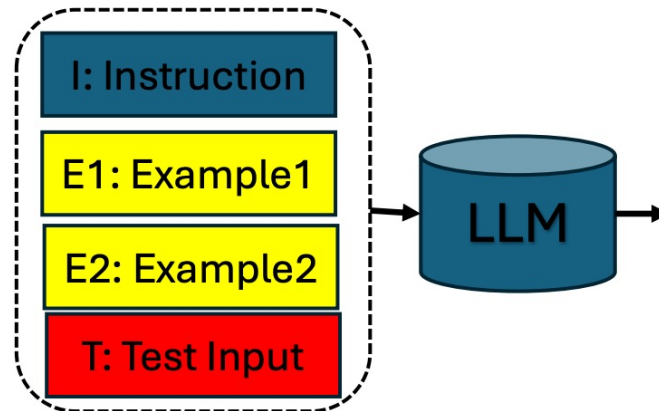
T: Test Input

[en]: After you become comfortable with formatting [fr]:

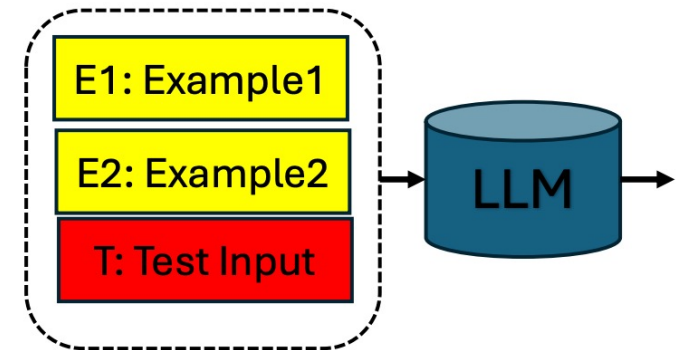
Zero Shot



Few Shot (w/ Instruction)



Few Shot (Example only)





# Example of emergence: Chain-of-Thought

## Standard Prompting

### Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The answer is 27. ❌

## Chain-of-Thought Prompting

### Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had  $23 - 20 = 3$ . They bought 6 more apples, so they have  $3 + 6 = 9$ . The answer is 9. ✅

Wei, et al. (2023) Chain-of-Thought Prompting Elicits Reasoning in LLMs

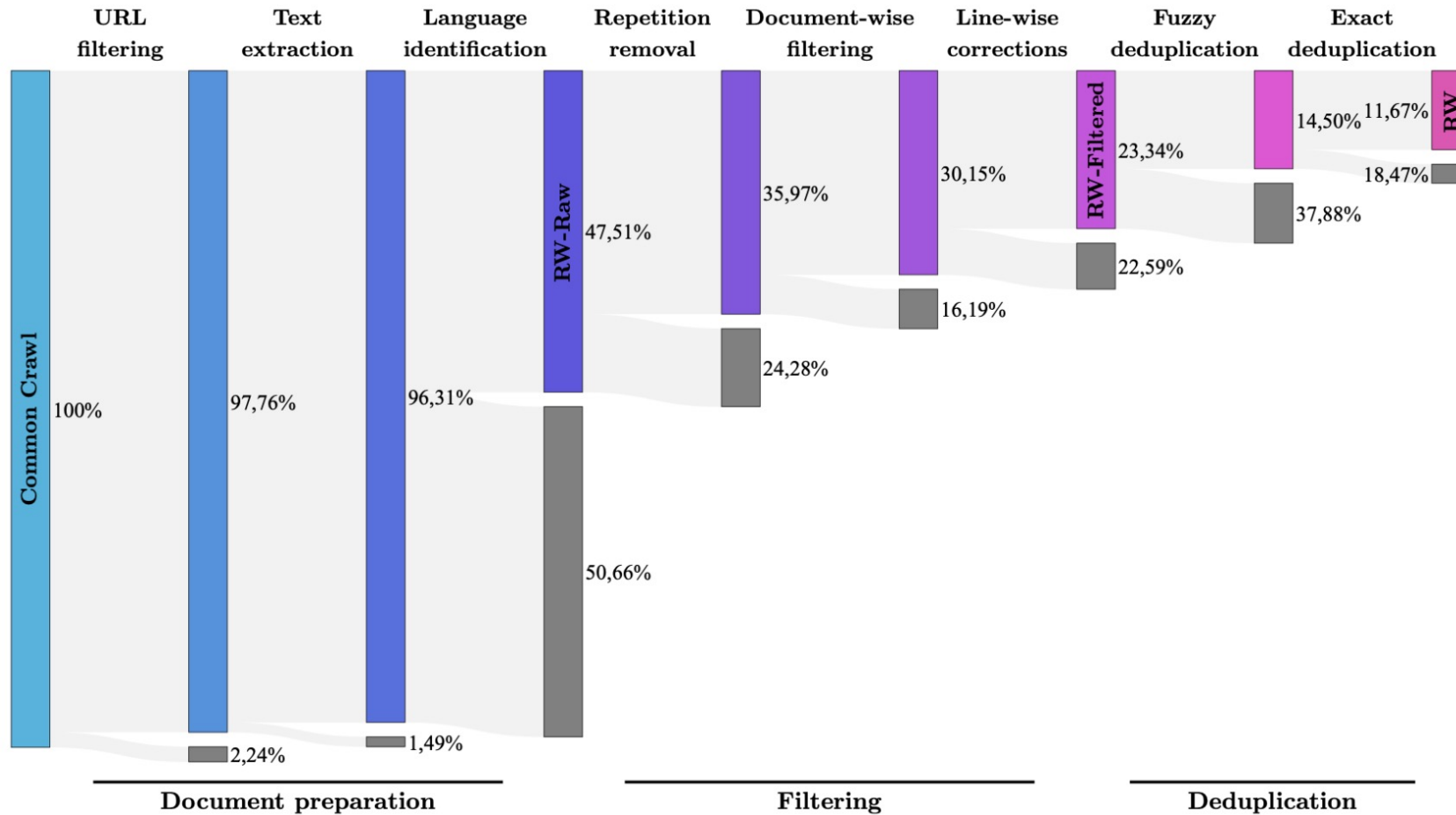
# Why does this work? Some hypotheses

---

- Task identification?
  - Xie et al. (2021). An explanation of in-context learning as implicit Bayesian inference
  - Raventos, et al. (2023). Pretraining task diversity and the emergence of non-Bayesian in-context learning for regression
- Some kind of "learning" without model updates?
  - Akyurek, et al. (2024). In-context language learning: architectures and algorithms
  - von Oswald, et al. (2023). Transformers learn in-context by gradient descent
- Both?
  - Pan, et al. (2023). What in-context learning "learns" in-context: disentangling task recognition and task learning



# Scale is difficult...for example, data filtering

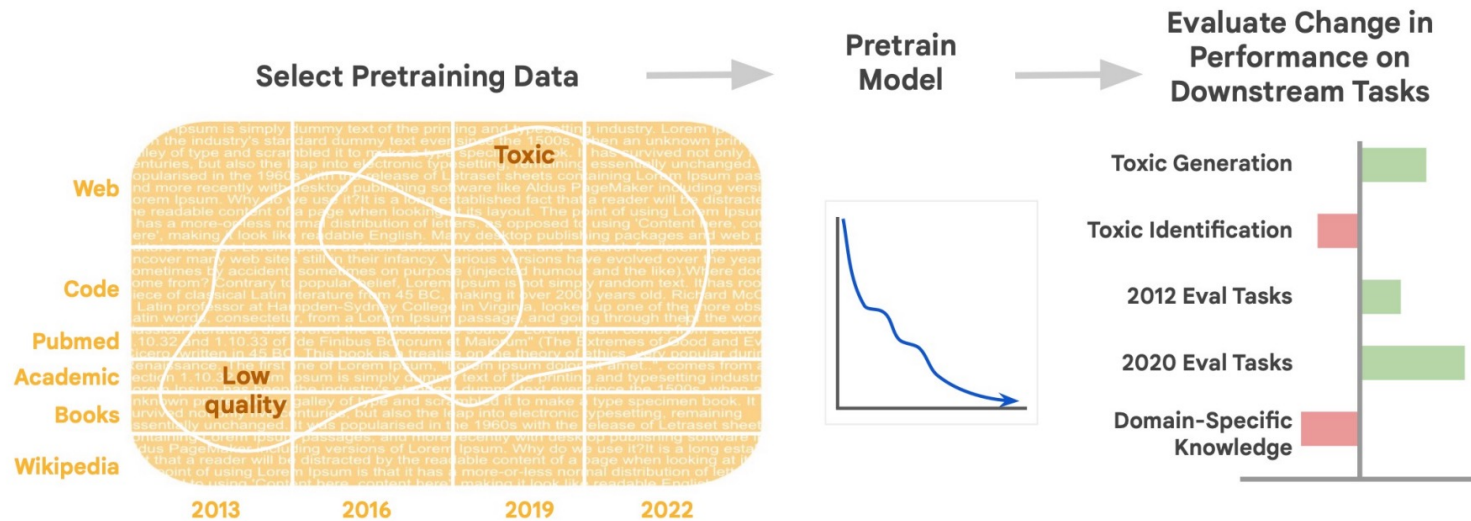


Penedo, et al. (2023) The Refined Web dataset for Falcon LLM



# Scale is difficult...for example, data filtering

A Pretrainer’s Guide to Training Data: Measuring the Effects of Data Age, Domain Coverage, Quality, & Toxicity [Longpre et al, NAACL 2024]

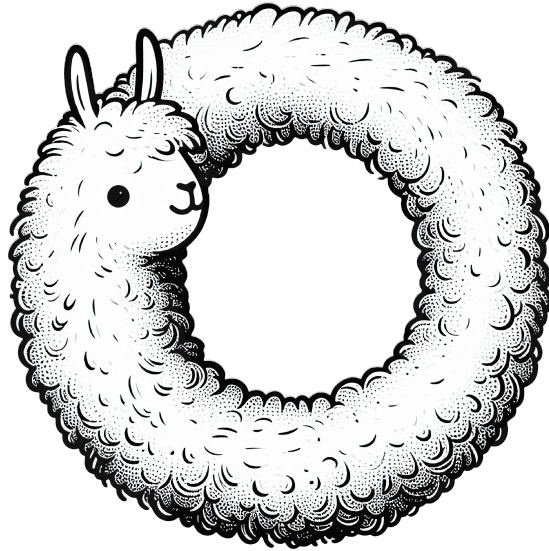


Some findings: strongly encourage to read the paper!

- “temporal shift between evaluation data and pretraining data leads to performance degradation, which is not overcome by finetuning”
- “a trade-off between performance on standard benchmarks and risk of toxic generations... there does not exist a one-size-fits-all solution to filtering.”



# A great learning opportunity

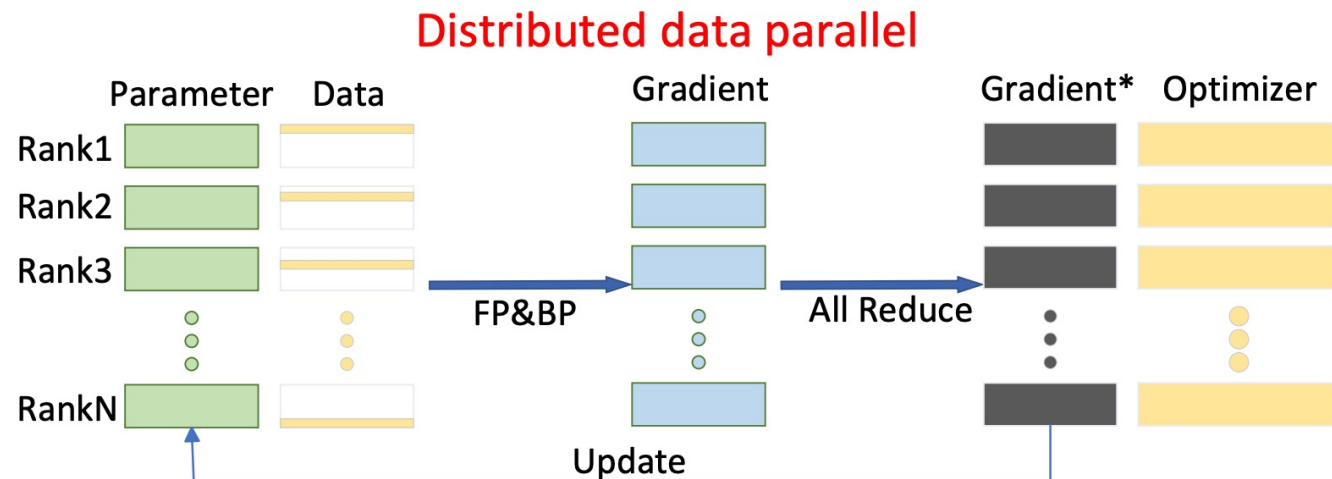
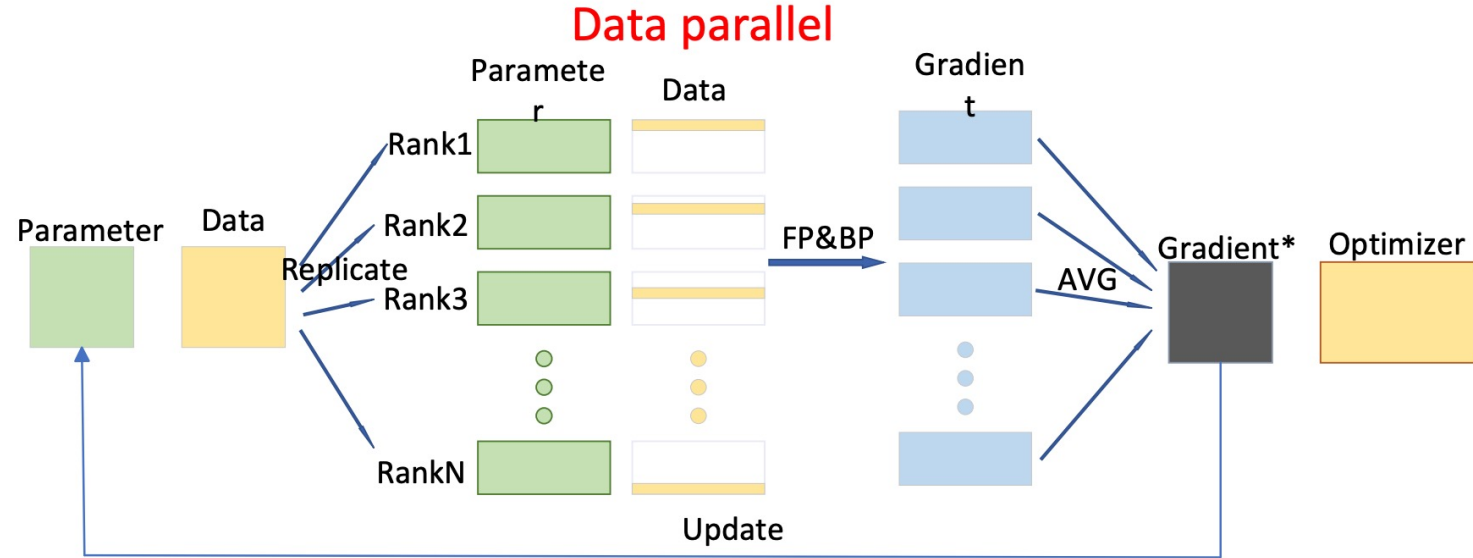


## LLM360



## TxT360

# Scale is difficult...for example, parallel training



# Challenges of MLE-based unsupervised training



# MLE = Minimizing KL-Divergence

$$KL(P_{data} \parallel P_{\theta}) = E_{x \sim P_{data}}[-\log P_{\theta}(x)] + const.$$

$$\operatorname{argmax}_{\theta} E_{x \sim P_{data}}[\log P_{\theta}(x)] = \operatorname{argmin}_{\theta} KL(P_{data} \parallel P_{\theta})$$

- Must spread probability mass to cover observed sequences, even if incoherent
  - Repetition and genericity ("The man said the man said...")
  - Poor calibration on out-of-distribution prompts
  - Memorization of rare patterns
- Sampling strategies matter
- Entropy / confidence





# What does MLE not do?

---

- No **semantics**
- No **task goals**
- No **explicit reward**

# Entropy and Confidence

- Token-level entropy:

$$H_t = - \sum_v P(x_t = v \mid x_{<t}) \log P(x_t = v \mid x_{<t})$$
$$= E_v[-\log P(x_t = v \mid x_{<t})]$$

- Low entropy = high confidence.

Do we want our model to be low or high entropy?

# Recall from Variational Inference

$$\log p(x | \theta) = E_{z \sim q}[\log p(x, z | \theta)] + H(q) + KL(q(z | x) || p(z | x, \theta))$$

$$\log p(x | \theta) \geq \underbrace{E_{z \sim q}[\log p(x, z | \theta)] + H(q)}$$

"ELBO": Evidence Lower Bound

Maximizing ELBO  $\rightarrow$  Increased entropy of  $q(z)$

Does Maximizing Likelihood  $\rightarrow$  Increased entropy of  $p(x | \theta)$ ?



# MLE and Entropy

---

$$\widehat{\theta}_{MLE} = \operatorname{argmax}_{\theta} E_{x \sim P_{data}} [\log P_{\theta}(x)]$$

- Directly optimize the data distribution  $P_{\theta}(x)$  without explicitly introducing auxiliary variables or explicitly controlling the entropy of  $x$
- Often implicitly pushes  $P_{\theta}(x)$  toward low-entropy distributions
- Mode collapse / degeneration / "memorization"



# MLE and Entropy

$$\widehat{\theta}_{MLE} = \operatorname{argmax}_{\theta} E_{x \sim P_{data}} [\log P_{\theta}(x)]$$

- Solutions:

- Explicitly add a penalty for low entropy:

$$\hat{\theta} = \operatorname{argmax}_{\theta} E_{x \sim P_{data}} [\log P_{\theta}(x)] + \lambda \sum_t H[P_{\theta}(x_t | x_{<t})]$$

- Smooth labels:

$$\tilde{y}_{\epsilon} = \begin{cases} 1 - \epsilon, & y = 1 \\ \frac{\epsilon}{|V| - 1}, & y = 0 \end{cases}$$

- Contrastive losses:
- Scheduled sampling / noise contrastive objectives
- Risk minimization, utility, preference-based losses

# Some inspiration

## FastCache: Cache what matters, Skip what doesn't.

First-author  
UW-Madison undergrad!



Anonymous Authors

To appear at CVPR BASE 2025

### Abstract

*Diffusion Transformers (DiT) are powerful generative models but remain computationally intensive due to their iterative structure and deep transformer stacks. To alleviate this inefficiency, we propose **FastCache**, a hidden-state-level caching and compression framework that accelerates DiT inference by exploiting redundancy within the model's internal representations. **FastCache** introduces a dual strategy: (1) a spatial-aware token selection mechanism that adaptively filters redundant tokens based on hidden state saliency, and (2) a transformer-level cache that reuses latent activations across timesteps when changes are statistically insignificant. These modules work jointly to reduce unnecessary computation while preserving generation fidelity. Theoretical analysis shows that **FastCache** maintains bounded approximation error under a hypothesis-testing-based decision rule. Empirical evaluations across multiple DiT variants demonstrate substantial reductions in latency and memory usage, with minimal degradation in output quality, as measured by FID and t-FID.*

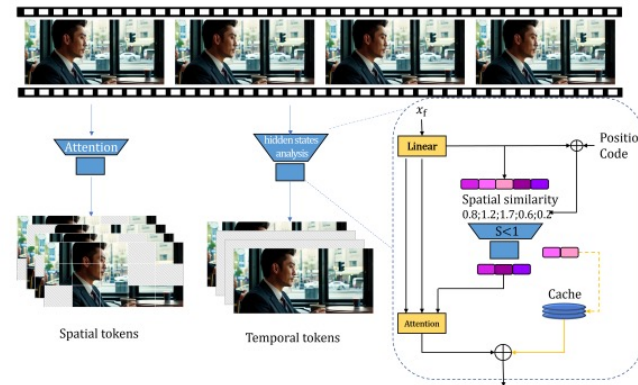


Figure 1. **Hidden-state redundancy across timesteps in DiT.** Despite visible pixel changes, internal representations remain stable—especially in low-motion regions.

shown in Figure 1, a lightweight framework that accelerates DiT by reusing hidden states at the transformer level. Unlike prior methods that operate on low-level features or fixed attention maps, **FastCache** targets semantic representations directly and supports fine-grained reuse through hidden states across both time and space.

Questions?

