Efficient ML

EECE454 Intro. to Machine Learning Systems

Fall 2024

Motivation

Chowdhery et al., "PaLM: Scaling Language Models with Pathways," arXiv 2022

Modern AI is big

- **Back in 2022.** Google released PaLM, one of the previous generations of Gemini.
  - <u>Dataset</u>. Text corpus of 7.8×10^{11} tokens

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%

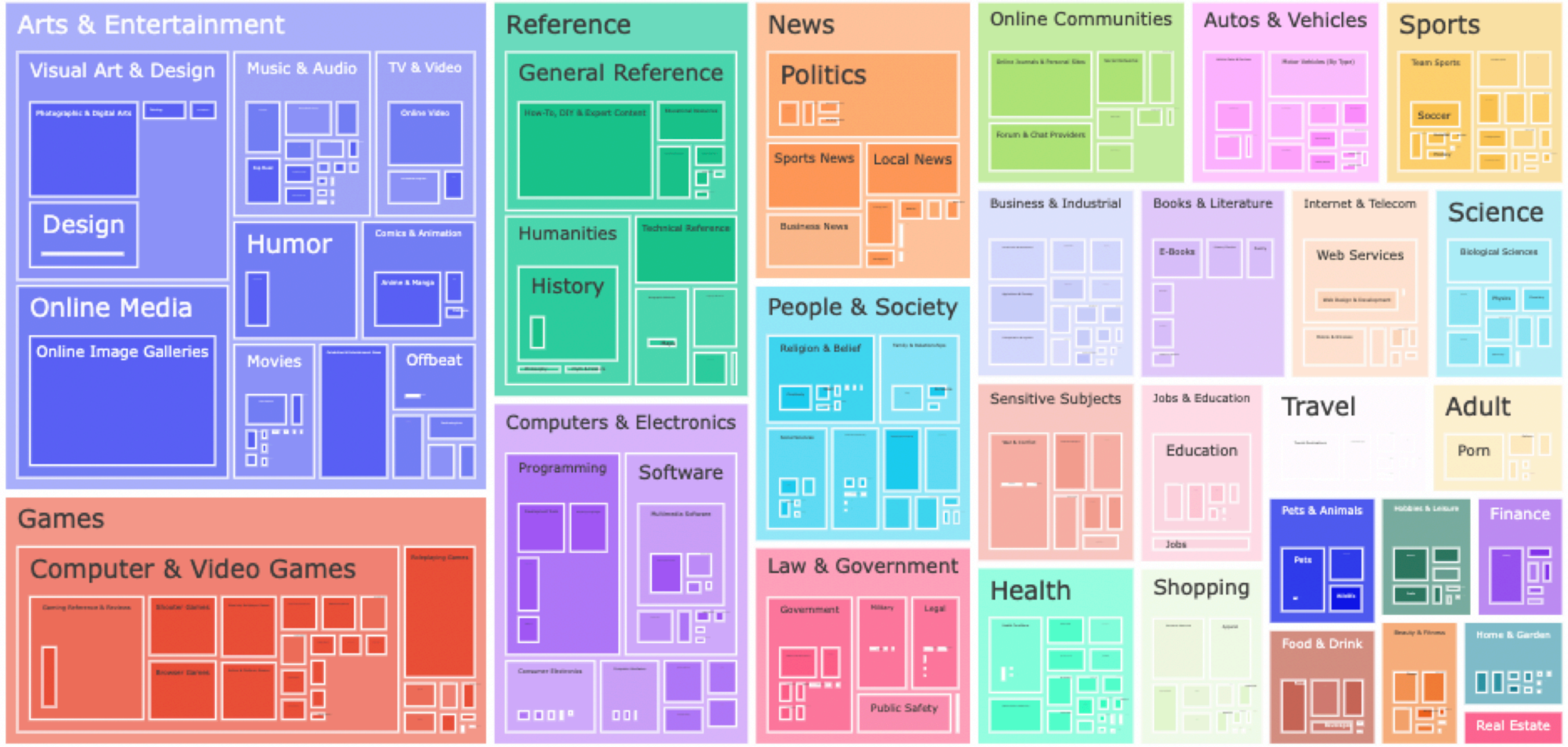


Figure 25: Hierarchical topics detected in the dataset.

Modern AI is big

- Parameters. 5.4×10^{11} parameters (in various precisions)
  - $\approx$ 1TB memory (in 16 bits)
- Computation. 2.56×10^{24} FLOPs for training
  - $\approx$ \$27M, 1500 hours

Model	TFLOPs per token		Train FLOPs	PetaFLOP/s-days
	(non-attn+attn)	(non-attn+attn+remat)		
8B	0.0550	0.0561	4.29×10^{22}	497
62B	0.388	0.392	3.08×10^{23}	3570
540B	3.28	4.10	2.56×10^{24}	29600

Modern AI is big

- Hardware. 6,144 TPUv4 chips

Modern AI is big

- Human. 67 Engineers

PaLM: Scaling Language Modeling with Pathways

Aakanksha Chowdhery* Sharan Narang* Jacob Devlin*
Maarten Bosma Gaurav Mishra Adam Roberts Paul Barham
Hyung Won Chung Charles Sutton Sebastian Gehrmann Parker Schuh Kensen Shi
Sasha Tsvyashchenko Joshua Maynez Abhishek Rao† Parker Barnes Yi Tay
Noam Shazeer‡ Vinodkumar Prabhakaran Emily Reif Nan Du Ben Hutchinson
Reiner Pope James Bradbury Jacob Austin Michael Isard Guy Gur-Ari
Pengcheng Yin Toju Duke Anselm Levskaya Sanjay Ghemawat Sunipa Dev
Henryk Michalewski Xavier Garcia Vedant Misra Kevin Robinson Liam Fedus
Denny Zhou Daphne Ippolito David Luan‡ Hyeontaek Lim Barret Zoph
Alexander Spiridonov Ryan Sepassi David Dohan Shivani Agrawal Mark Omernick
Andrew M. Dai Thanumalayan Sankaranarayana Pillai Marie Pellat Aitor Lewkowycz
Erica Moreira Rewon Child Oleksandr Polozov† Katherine Lee Zongwei Zhou
Xuezhi Wang Brennan Saeta Mark Diaz Orhan Firat Michele Catasta† Jason Wei
Kathy Meier-Hellstern Douglas Eck Jeff Dean Slav Petrov Noah Fiedel

Google Research

Preparation

Wrote the initial proposal: Sharan Narang, Alexander Spiridonov, Noah Fiedel, Noam Shazeer, David Luan

Model architecture and optimizer selection: Noam Shazeer, Yi Tay, Sharan Narang, Rewon Child, Aakanksha Chowdhery

Model scaling validation: Aakanksha Chowdhery, Noam Shazeer, Rewon Child

Low-precision finetuning and inference: Shivani Agrawal, Reiner Pope

Training strategy and efficiency: Noam Shazeer, Aakanksha Chowdhery, James Bradbury, Zongwei Zhou, Anselm Levskaya, Reiner Pope

Pod-level Data Parallelism Aakanksha Chowdhery, Paul Barham, Sasha Tsvyashchenko, Parker Schuh

T5X Model Parallelism and Flaxformer Adam Roberts, Hyung Won Chung, Anselm Levskaya, James Bradbury, Mark Omernick, Brennan Saeta

Deterministic data pipeline: Gaurav Mishra, Adam Roberts, Noam Shazeer, Maarten Bosma

Efficient Checkpointing: Sasha Tsvyashchenko, Paul Barham, Hyeontaek Lim

Pathways system: Aakanksha Chowdhery, Paul Barham, Hyeontaek Lim, Thanunalayan Sankaranayana Pillai, Michael Isard, Ryan Sepassi, Sanjay Ghemawat, Jeff Dean

Dataset and Vocabulary development: Maarten Bosma, Rewon Child, Andrew Dai, Sharan Narang, Noah Fiedel

Model Training

Large-scale Training: Aakanksha Chowdhery, Jacob Devlin, Sharan Narang
Large-scale Training includes in-flight debugging of training instability issues, architecture and optimizer improvements, training strategy improvements, and resolving infrastructure bottlenecks.

Infrastructure improvements: Paul Barham, Hyeontaek Lim, Adam Roberts, Hyung Won Chung, Maarten Bosma, Gaurav Mishra, James Bradbury

Model performance validation on downstream tasks: Sharan Narang, Gaurav Mishra

Post-Training

Coordination of results and model analyses: Sharan Narang

Few-shot evaluation infrastructure: Maarten Bosma, Sharan Narang, Adam Roberts

English NLP tasks (few-shot evaluation): Sharan Narang, Nan Du

Finetuning on SuperGlue: Sharan Narang, Yi Tay, Liam Fedus

BIG-bench tasks (few-shot evaluation): Gaurav Mishra, Noah Fiedel, Guy Gur-Ari, Jacob Devlin, Aakanksha Chowdhery, Sharan Narang

Reasoning tasks (few-shot evaluation): Jason Wei, Xuezhi Wang, Denny Zhou

Code tasks (few-shot evaluation and finetuning): Jacob Austin, Henryk Michalewski, Charles Sutton, Aitor Lewkowycz, Kensen Shi, Pengcheng Yin, Oleksandr Polozov, Vedant Misra, Michele Catasta, Abhishek Rao, David Dohan, Aakanksha Chowdhery

Translation tasks (few-shot evaluation): Xavier Garcia, Orhan Firat

Multilingual Natural Language Generation (few-shot evaluation and finetuning): Joshua Maynez, Sebastian Gehrmann

Multilingual Question Answering (few-shot evaluation and finetuning): Sharan Narang, Yi Tay

Analysis of noise in few-shot performance: Barret Zoph

Representational Bias Analysis (few-shot evaluation and dataset analysis): Marie Pellat, Kevin Robinson, Sharan Narang, Jacob Devlin, Emily Reif, Parker Barnes

Dataset contamination: Jacob Devlin, Sharan Narang

Memorization: Katherine Lee, Daphne Ippolito, Jacob Devlin

Exploring Explanations: Jacob Devlin

Ethical Considerations: Marie Pellat, Kevin Robinson, Mark Díaz, Sunipa Dev, Parker Barnes, Toju Duke, Ben Hutchinson, Vinodkumar Prabhakaran, Kathy Meier-Hellstern

Compute Usage and Environmental Impact: Aakanksha Chowdhery, James Bradbury, Zongwei Zhou

Model serving (API, use cases and efficiency): Sharan Narang, Jacob Devlin, Jacob Austin, James Bradbury, Aakanksha Chowdhery, Zongwei Zhou, Reiner Pope, Noah Fiedel

Model card and datasheet: Alexander Spiridonov, Andrew Dai, Maarten Bosma, Jacob Devlin

Product Management: Alexander Spiridonov

Paper Writing and Reviewing: All authors contributed to writing and reviewing the paper

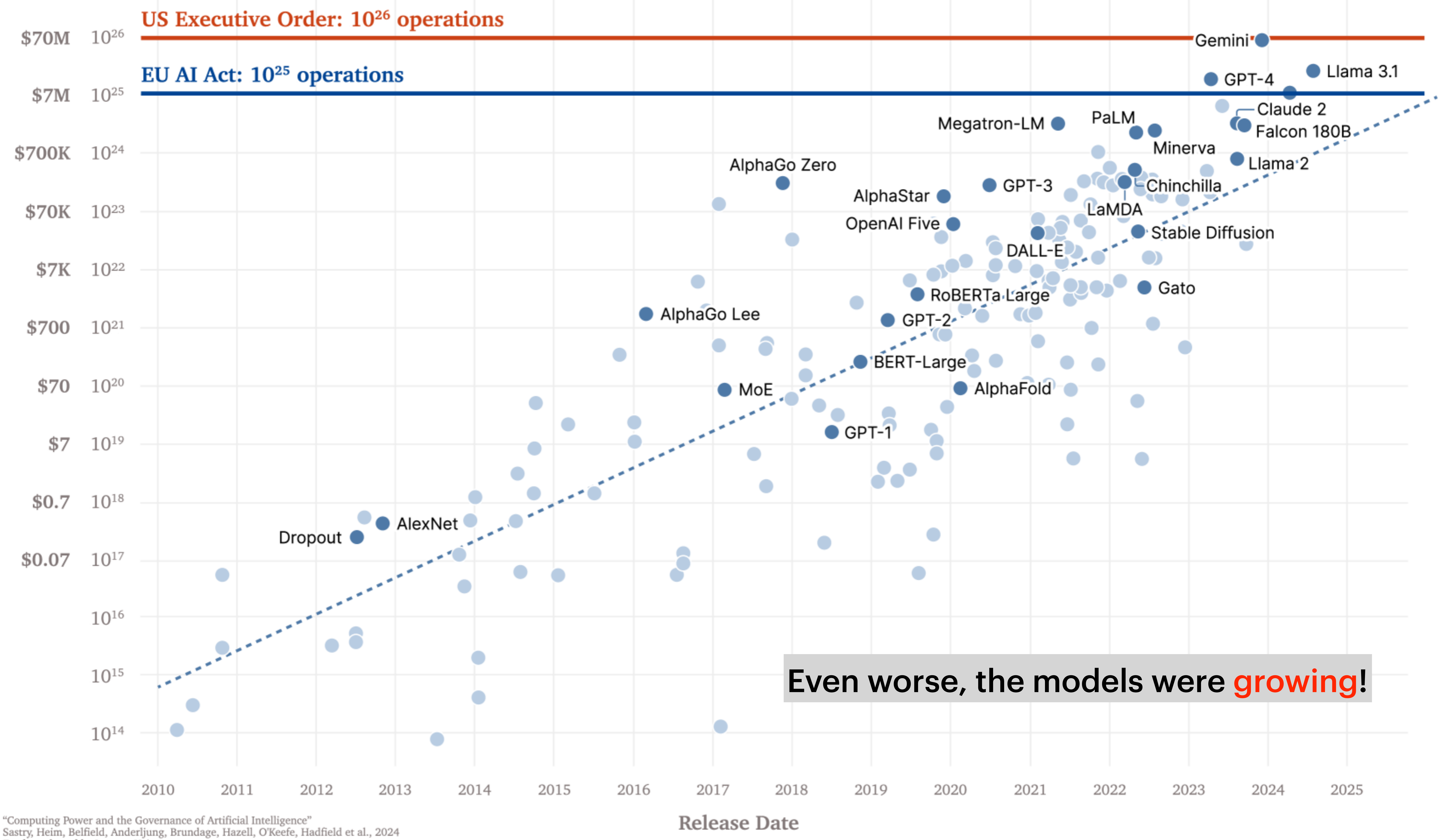
Full Project Lifecycle

Overall project leadership: Sharan Narang, Aakanksha Chowdhery, Noah Fiedel

Responsible AI and Safety leadership: Kathy Meier-Hellstern

Resource management: Erica Moreira

Advisors: Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, Noah Fiedel



Even worse, the models were growing!

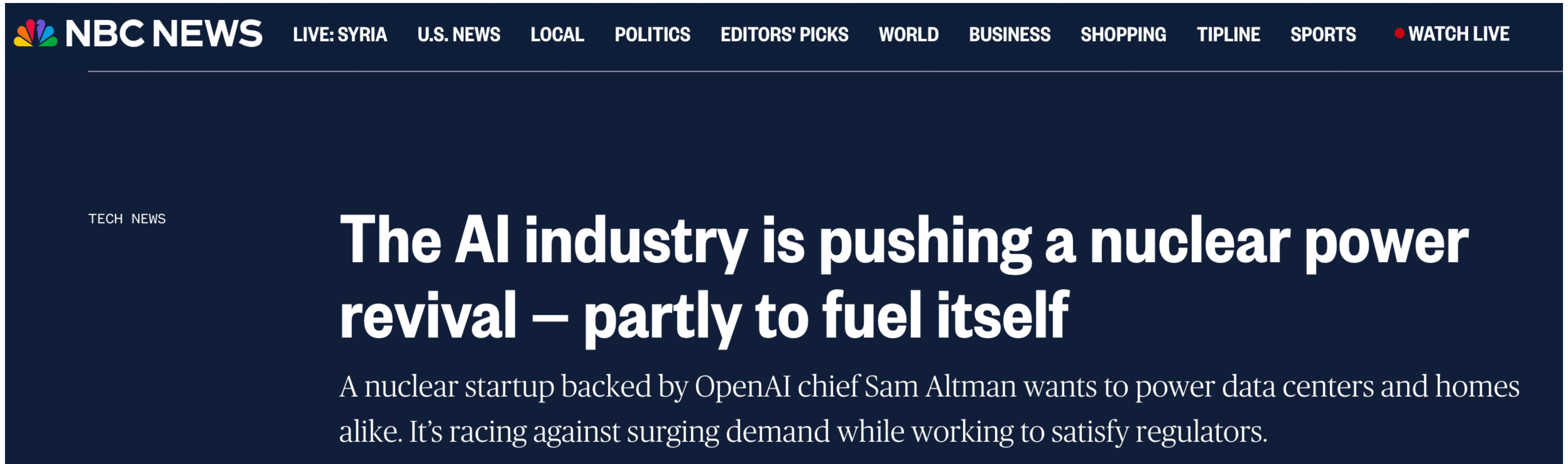
"Computing Power and the Governance of Artificial Intelligence"
Sastry, Heim, Belfield, Anderljung, Brundage, Hazell, O'Keefe, Hadfield et al., 2024
Further adapted by Lennart Heim.

Modern AI is big

- **Question.** Will models keep growing in 2025?

Modern AI is big

- **Question.** Will models keep growing in 2025?
  - Answer. Maybe not
    - Inference cost is too expensive
    - Data is limited, eventually (although we are not quite there yet)
    - Government regulations
      - 🇺🇸: Training FLOPs over 10^{26} = Inspection
      - 🇪🇺: Training FLOPs over 10^{25} = Inspection


NBC NEWS LIVE: SYRIA U.S. NEWS LOCAL POLITICS EDITORS' PICKS WORLD BUSINESS SHOPPING TIPLINE SPORTS WATCH LIVE

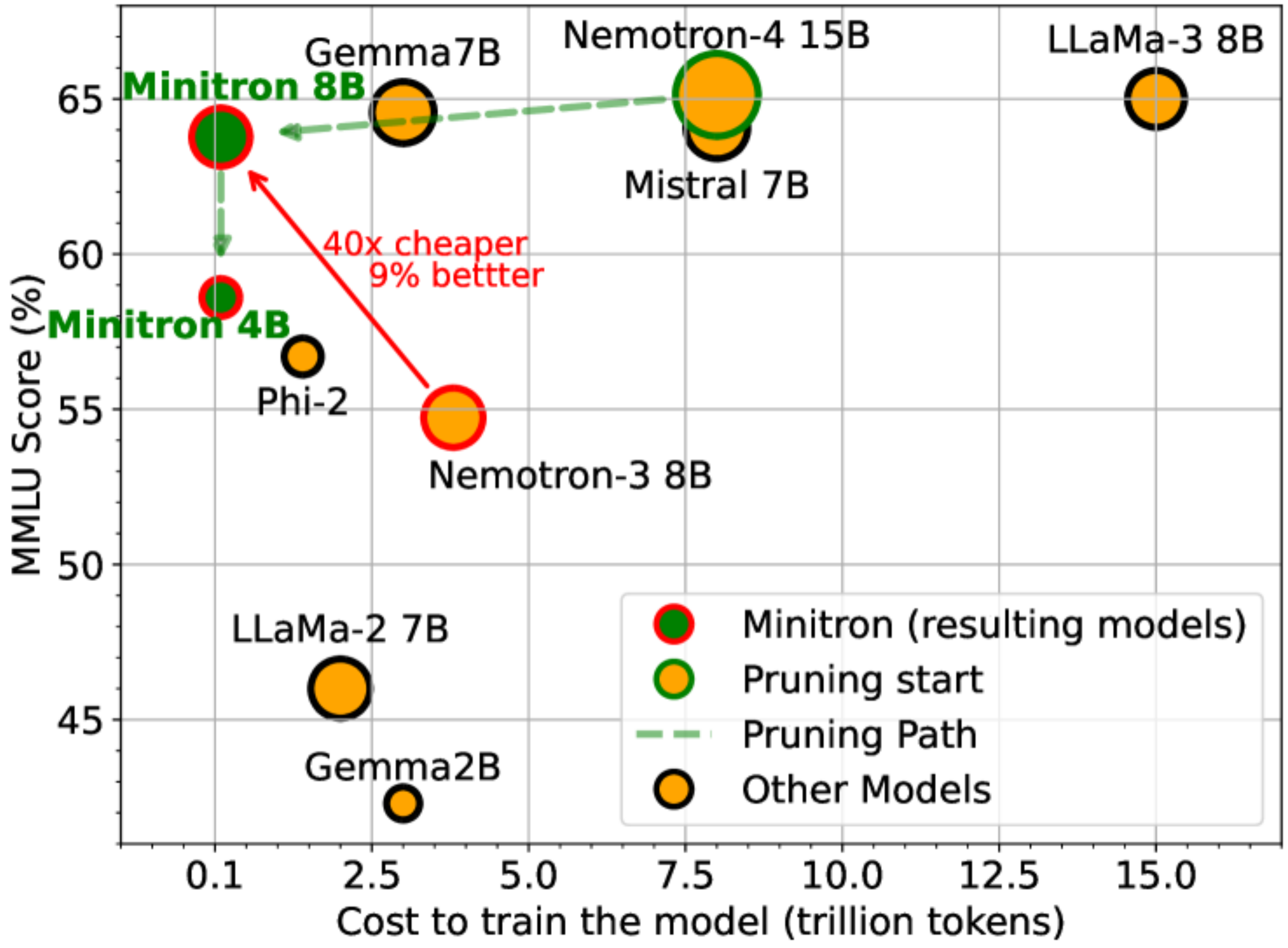
TECH NEWS

The AI industry is pushing a nuclear power revival – partly to fuel itself

A nuclear startup backed by OpenAI chief Sam Altman wants to power data centers and homes alike. It's racing against surging demand while working to satisfy regulators.

Modern AI is big

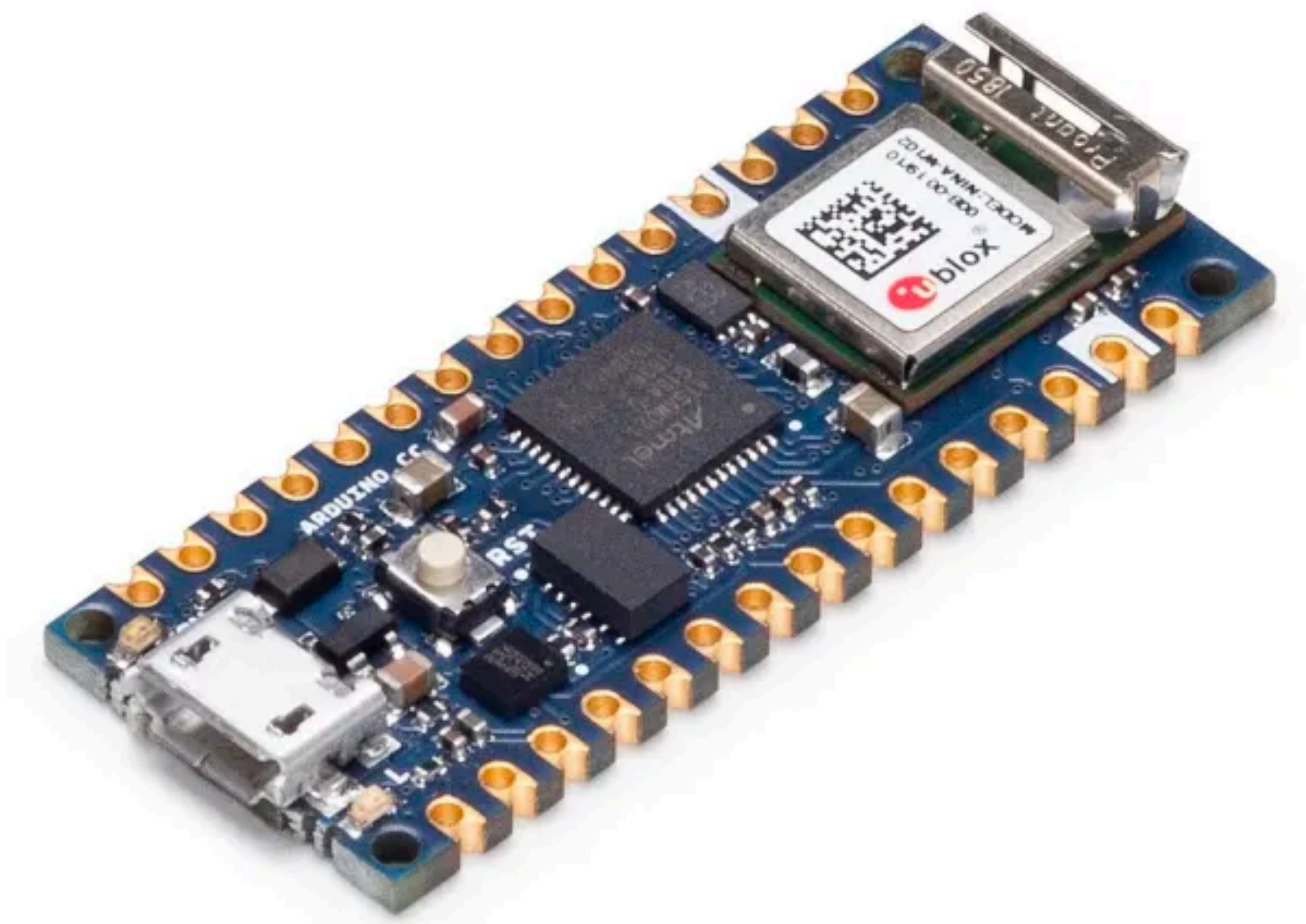
- Instead, the recent trend is to reduce the cost for services
  - **Recipe.** Start from a big model, then make it smaller.

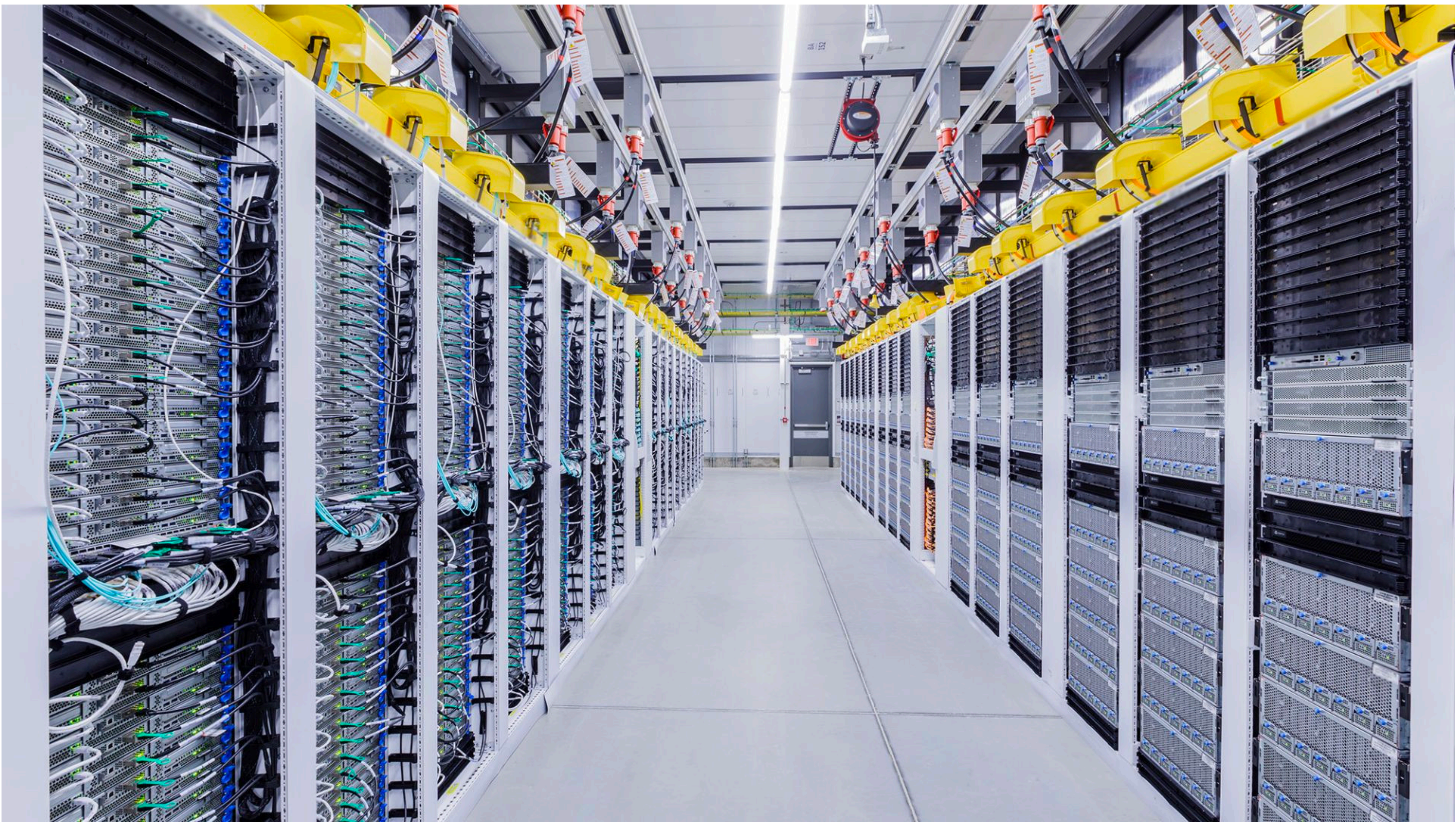


Efficient ML

Goals

- **Efficient ML.** A collection of techniques to reduce various costs of ML
  - Scale. Microcontrollers (a ConvNet)
    Mobile phones (Google Gemini Nano)
    Laptop (small LLMs)
    GPU server (giant LLMs)





Goals

- **Efficient ML.** A collection of techniques to reduce various costs of ML
  - Focus. Inference Latency
    Inference peak memory
    Training memory
    Training cost

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 495.44       Driver Version: 495.44       CUDA Version: 11.5     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  NVIDIA GeForce ...  Off  | 00000000:02:00.0 Off |                  N/A |
| 20%   52C    P2    68W / 300W |    758MiB / 11177MiB |      3%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|    0   N/A  N/A      1067      G   /usr/lib/xorg/Xorg                  9MiB |
|    0   N/A  N/A      1209      G   /usr/bin/gnome-shell                6MiB |
```

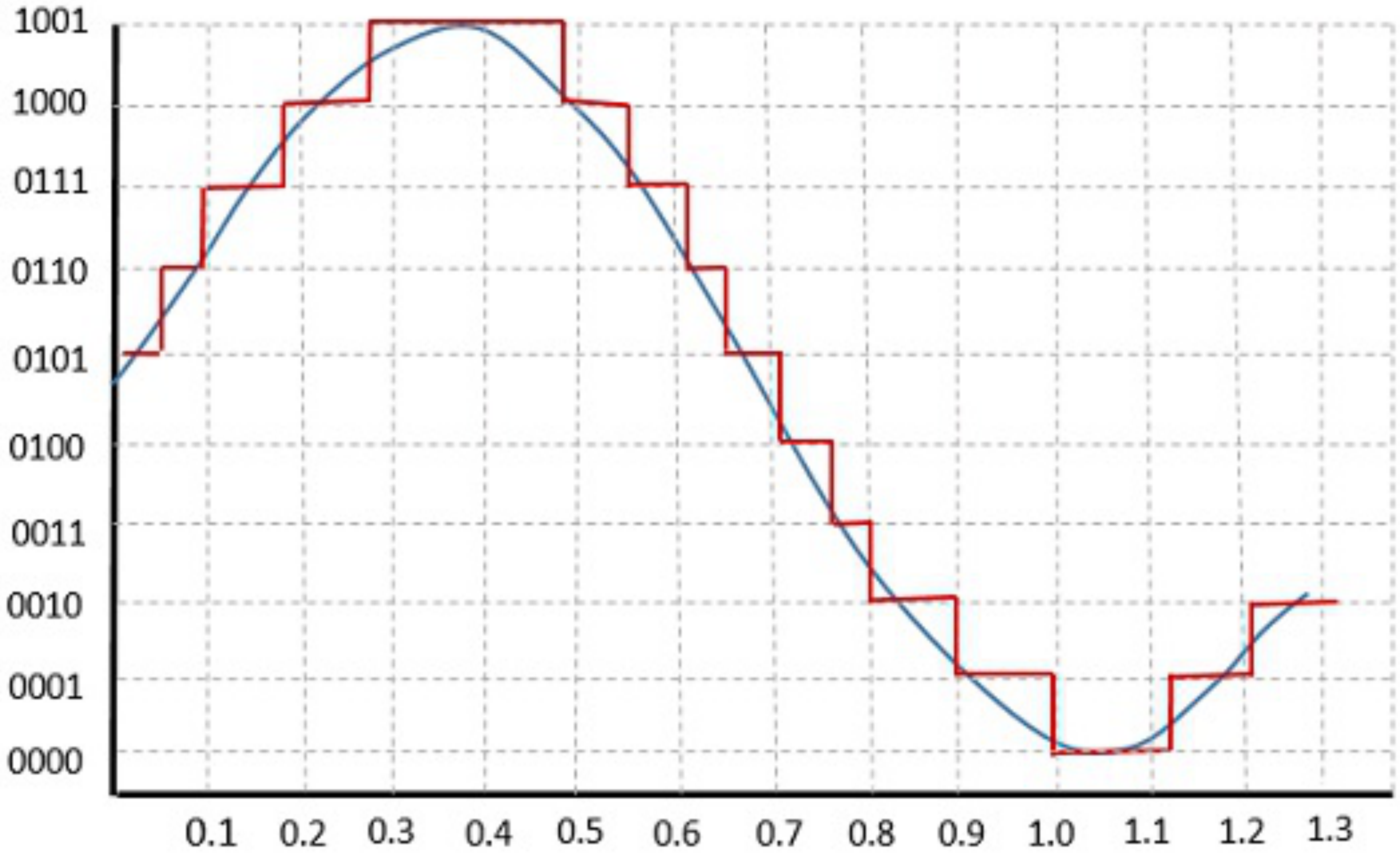
Techniques

- **Today.** We briefly cover three ideas
  - Quantization
  - Pruning
  - Knowledge distillation

Quantization

Quantization

- **Idea.** Reduce the precision of parameters in neural network
  - Weights
  - Activations
- Done either..
  - After all training
    (Post-Training Quantization)
  - Before fine-tuning
    (Quantization-Aware Training)
  - Before pre-training
    (Quantized Training)



Quantization

- **Benefits.** A lot!
  - Energy
  - Memory bandwidth
  - Computations
  - Storage space on RAM/SSD
  - Chip area

Add energy (pJ)	
INT8	FP32
0.03	0.9
30X energy reduction	

Mult energy (pJ)	
INT8	FP32
0.2	3.7
18.5X energy reduction	

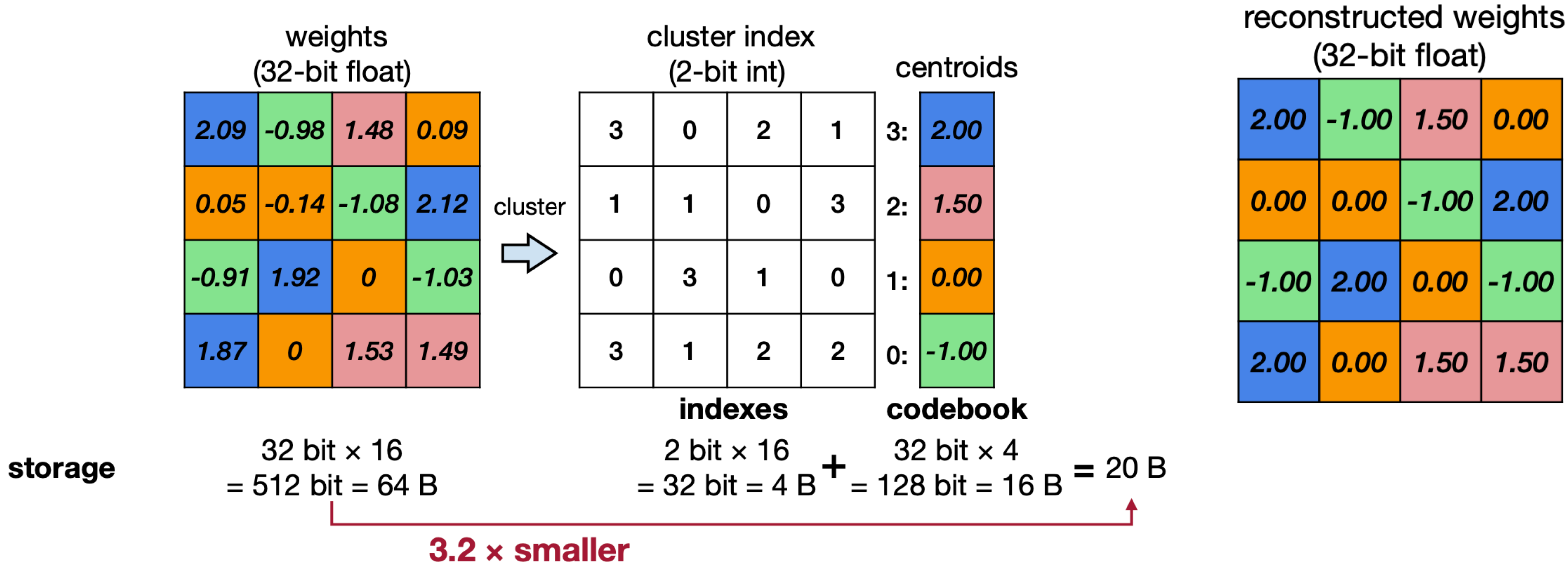
Mem access energy (pJ)	
Cache (64-bit)	
8KB	10
32KB	20
1MB	100
DRAM	1300-2600
Up to 4X energy reduction	

Add area (μm^2)	
INT8	FP32
36	4184
116X area reduction	

Mult area (μm^2)	
INT8	FP32
282	7700
27X area reduction	

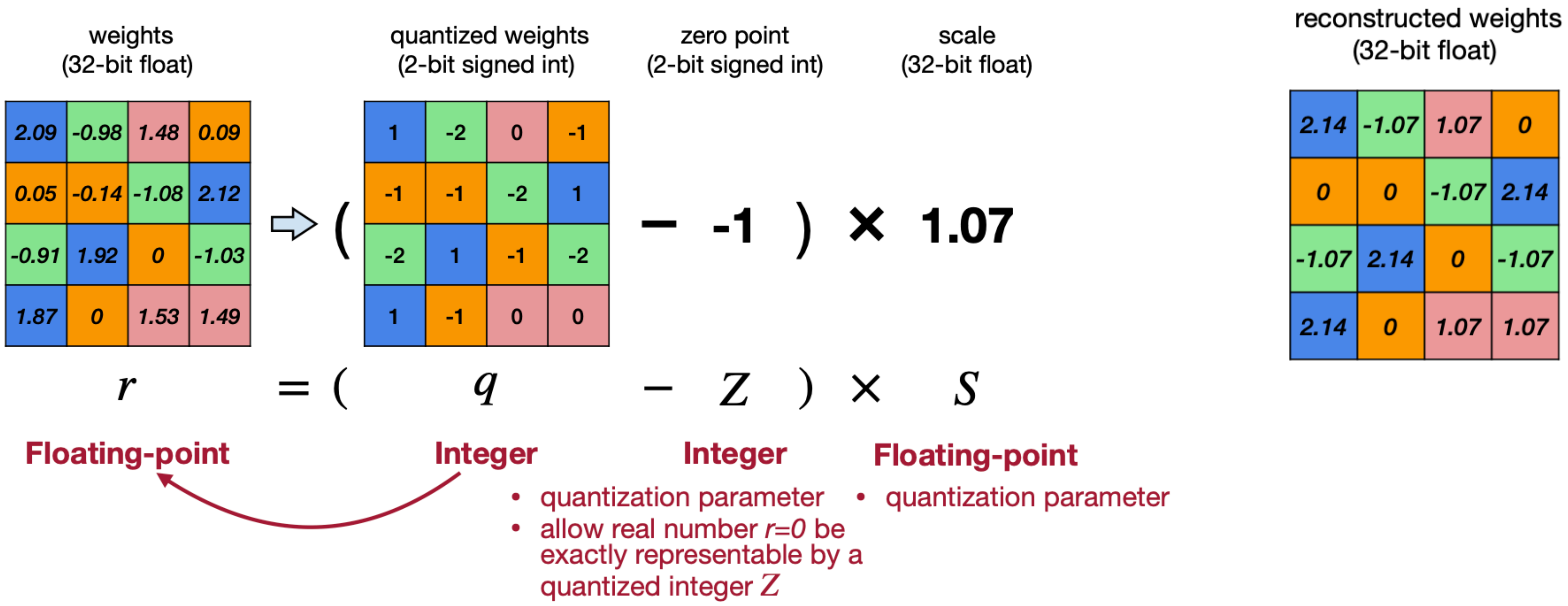
Quantization

- **Key Question.** Finding the right quantization level
  - Similar to K-means, but in 1-dimension



Quantization

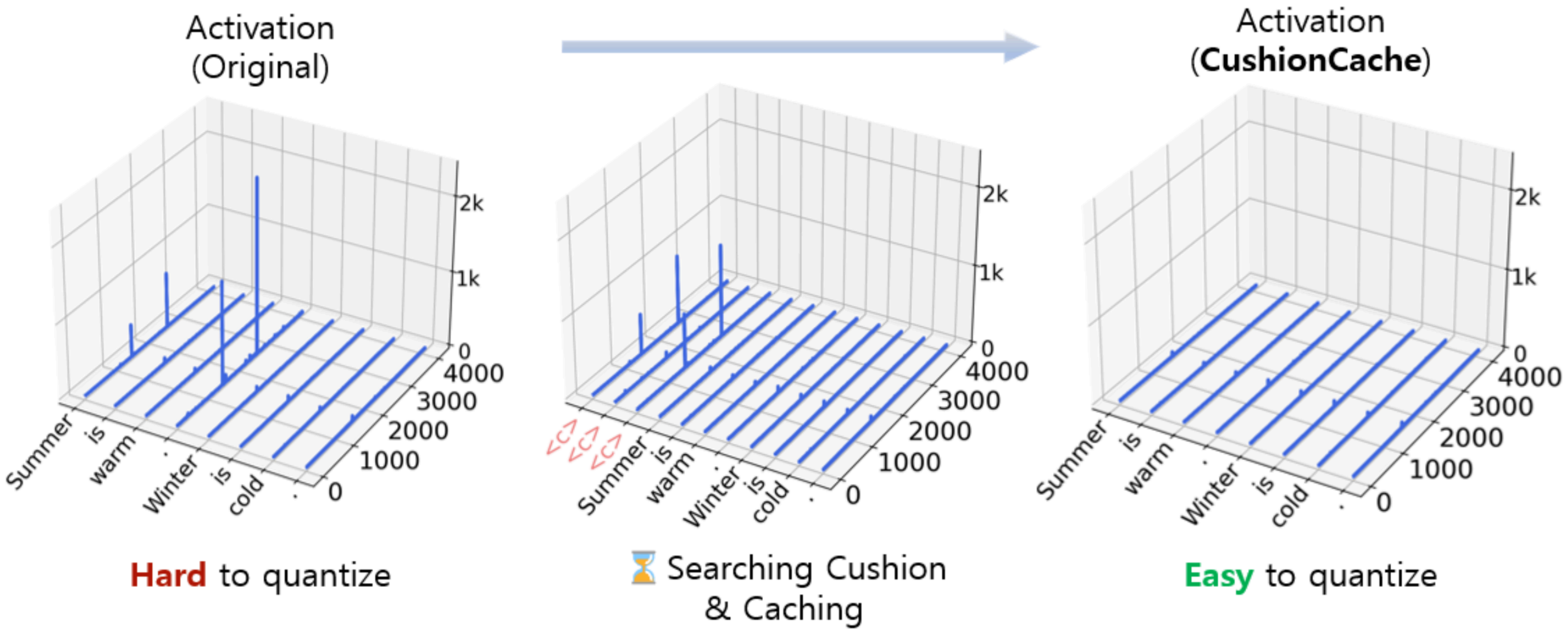
- **Popular.** Linear quantization
  - Optimized for inference; allows full computation in a quantized format (e.g., int8)
  - LLM inference. Not strictly necessary; the bottleneck is memory access, not computation



Son et al., “Prefixing Attention Sinks can Mitigate Activation Outliers for Large Language Model Quantization,” EMNLP 2024

Quantization

- **Trends in 2022—2024.** Handling activation outliers in LLMs
  - Outliers increase the quantization range —> quantization error too large
  - <u>Example</u>. Groupwise quantization (University of Washington & Facebook)
    Apply Hadamard rotations (ETH Zurich & Microsoft)
    Add good prompt tokens (POSTECH & Google)

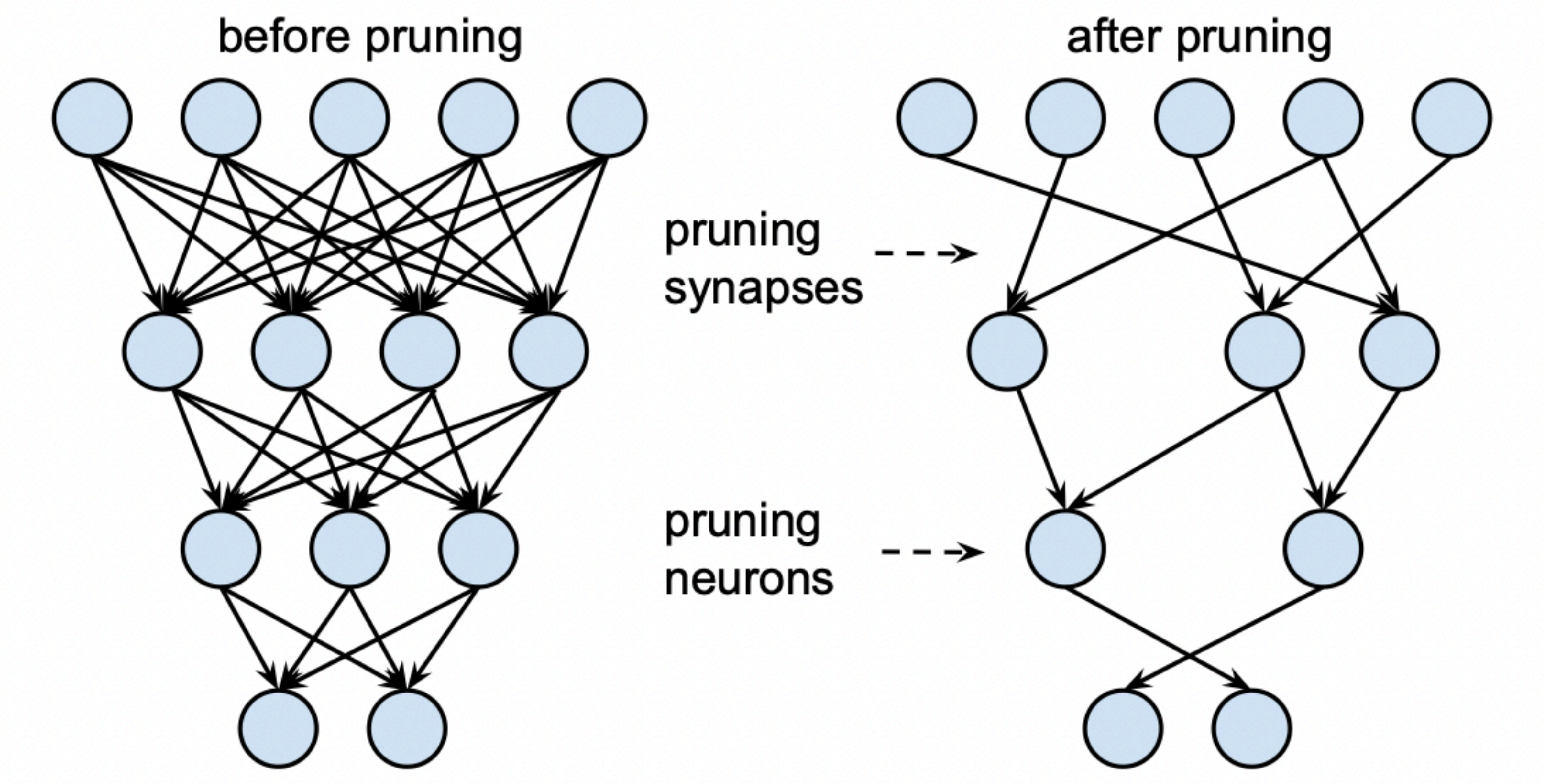


Pruning

Pruning

- **Idea.** Make some neural network weights equal to zero

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_5 & a_6 & a_7 & a_8 \\ a_9 & a_{10} & a_{11} & a_{12} \\ a_{13} & a_{14} & a_{15} & a_{16} \end{bmatrix} \longrightarrow \begin{bmatrix} 0 & 0 & \tilde{a}_3 & \tilde{a}_4 \\ \tilde{a}_5 & 0 & \tilde{a}_7 & 0 \\ \tilde{a}_9 & 0 & 0 & \tilde{a}_{12} \\ \tilde{a}_{13} & 0 & \tilde{a}_{15} & \tilde{a}_{16} \end{bmatrix}$$



Pruning

- **Benefit.** Reduce both memory and computation that are associated with zeros

$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}$$

32bits x 4 = 128bits

$$\downarrow$$

$$\begin{bmatrix} a_1 & 0 \\ 0 & a_4 \end{bmatrix}$$

32bits x 2 + α = 64bits + α

$$\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix} \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} = \begin{bmatrix} a_1b_1 + a_2b_3 & a_1b_2 + a_2b_4 \\ a_3b_1 + a_4b_3 & a_3b_1 + a_4b_4 \end{bmatrix}$$

8 Multiplications, 4 Additions

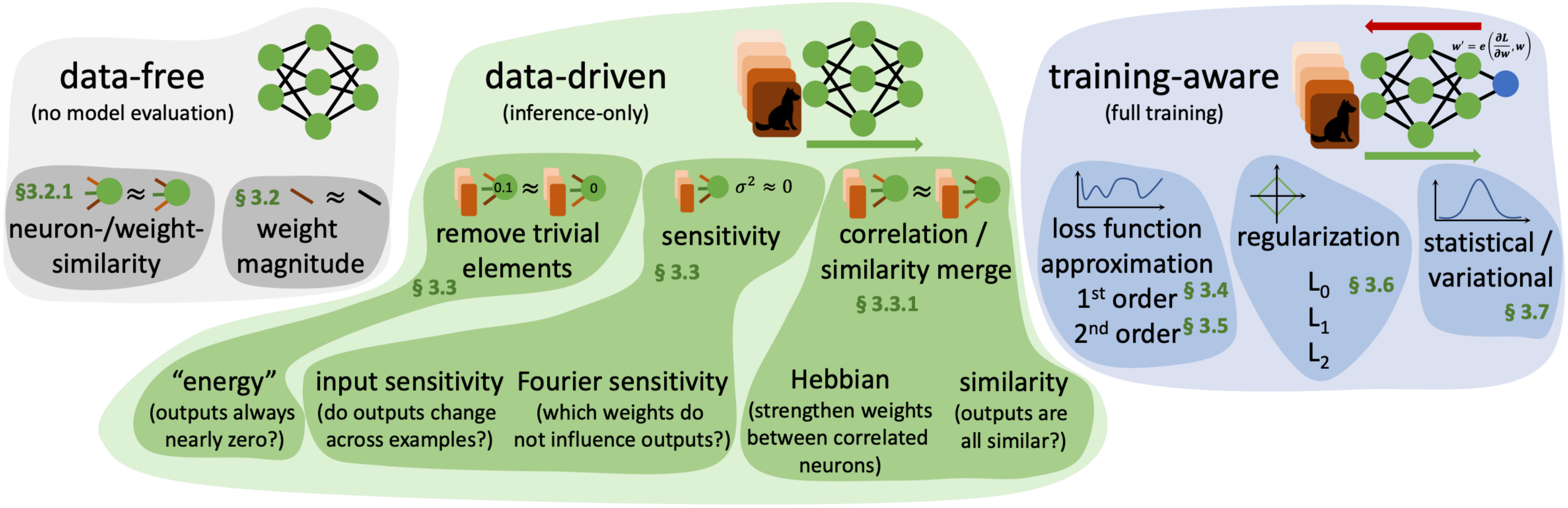
$$\downarrow$$

$$\begin{bmatrix} a_1 & 0 \\ 0 & a_4 \end{bmatrix} \begin{bmatrix} b_1 & b_2 \\ b_3 & b_4 \end{bmatrix} = \begin{bmatrix} a_1b_1 + 0 & a_1b_2 + 0 \\ 0 + a_4b_3 & 0 + a_4b_4 \end{bmatrix}$$

4 Multiplications, 0 Additions

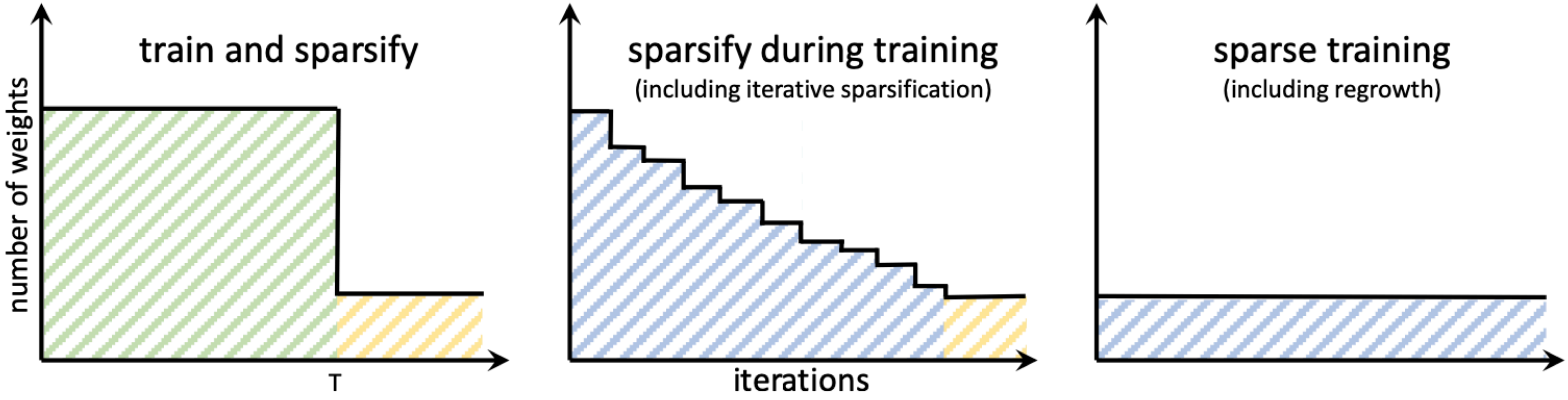
Pruning

- **Key Question.** Selecting the weights to remove
  - Which weights? When to prune? How much?
  - How to compensate for the removed weights?



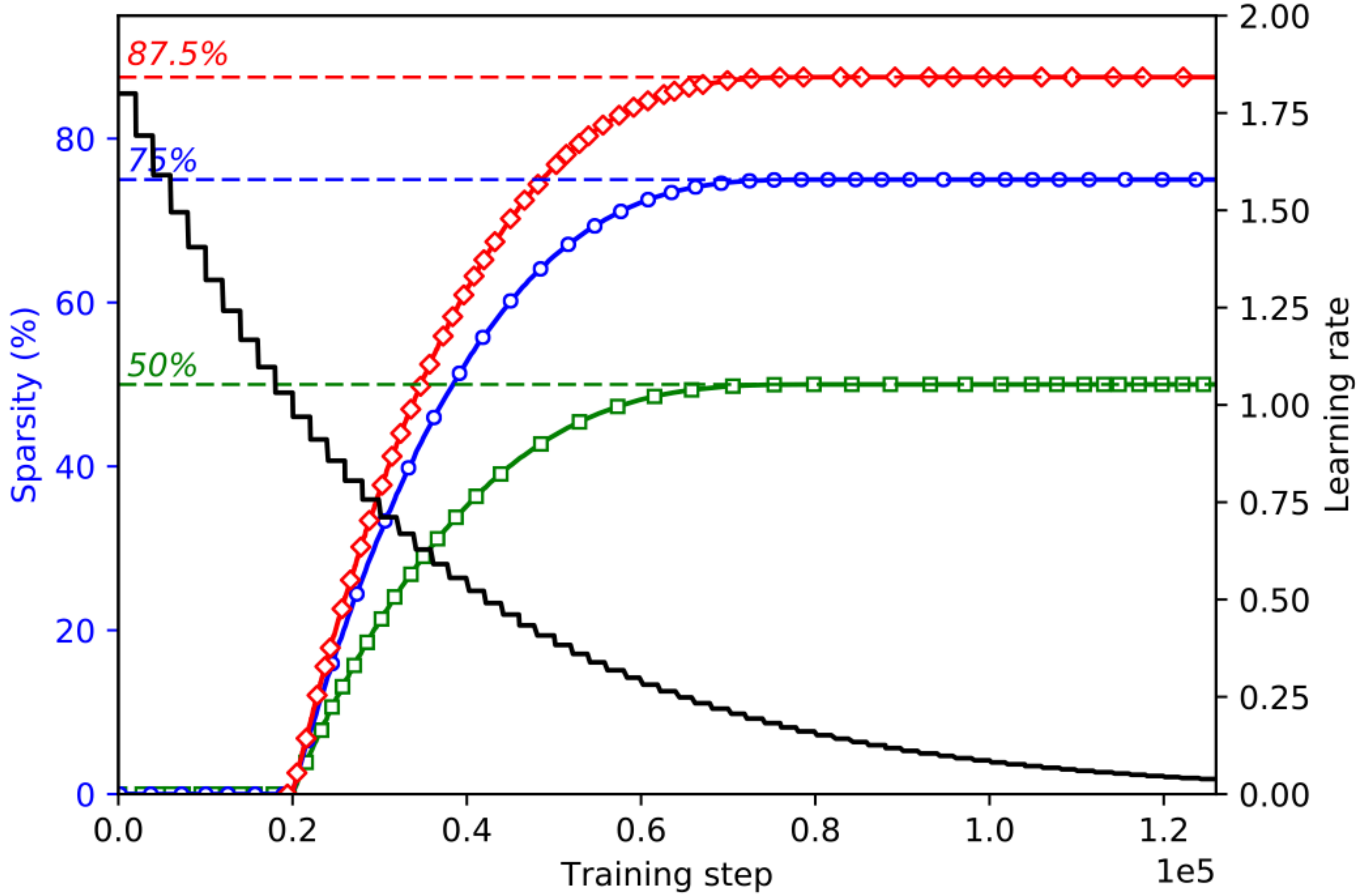
Pruning

- **Key Question.** Selecting the weights to remove
  - Which weights? When to prune? How much?
  - How to compensate for the removed weights?



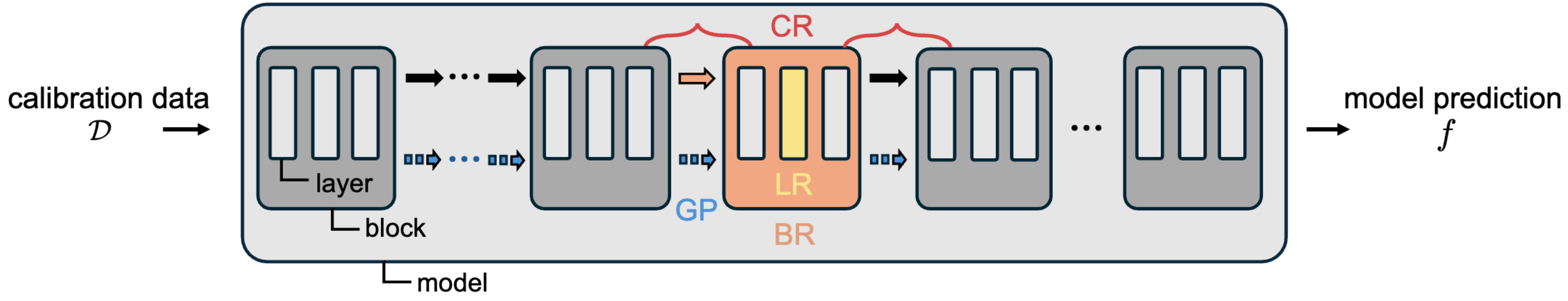
Pruning

- **Popular (for CNN).** Gradual, magnitude-based pruning
    - Remove small-magnitude weights from each layer
- **Popular (for LLMs).** Remove weights after the full training, but more carefully
    - Because the training cost is very expensive



Pruning

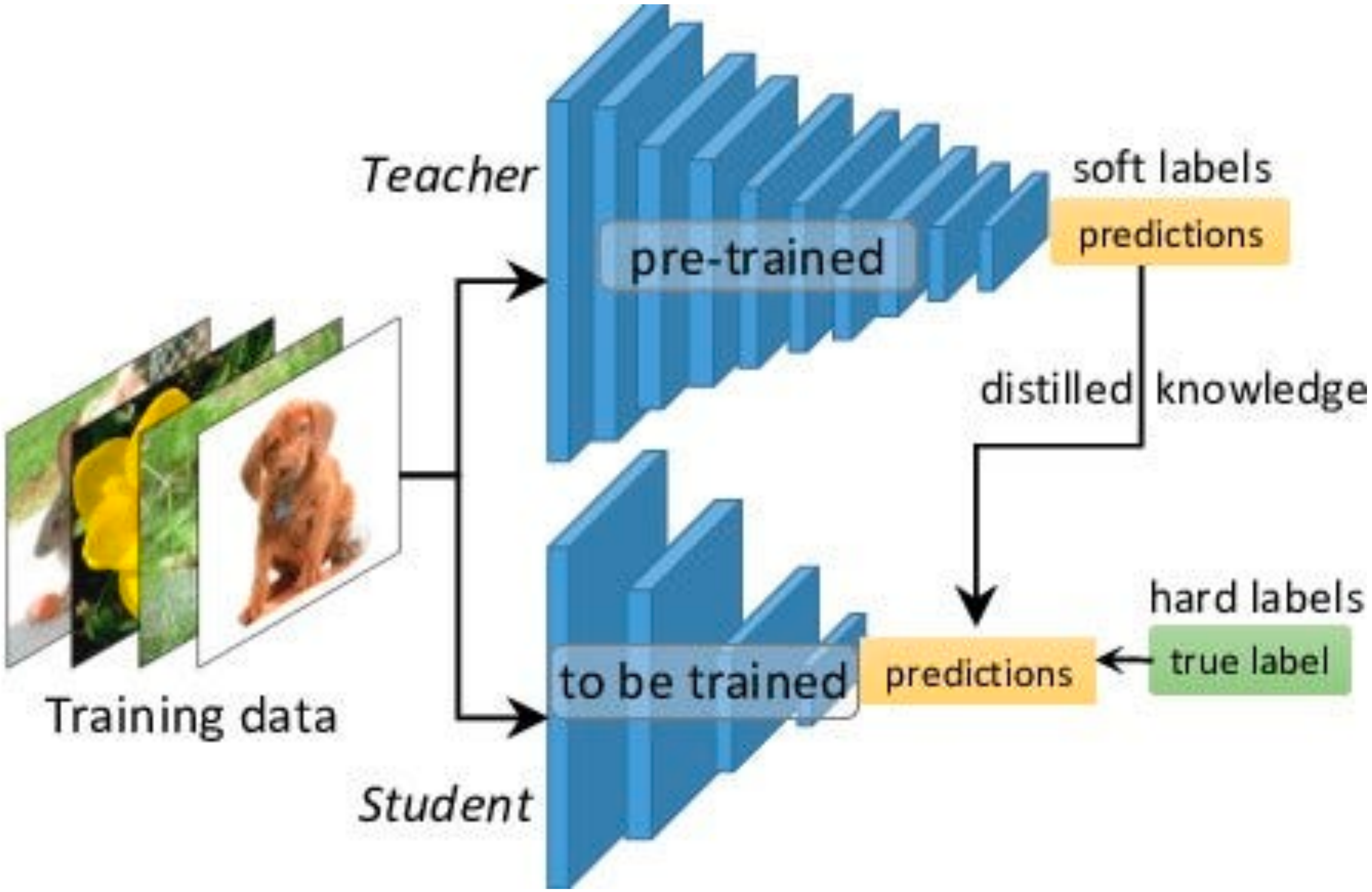
- **Trends in 2022—2024.** How to fine-tune pruned LLMs in an efficient manner
  - Examples. Knowledge distillation (NVIDIA)
    Blockwise optimization (POSTECH & Google)



Knowledge Distillation

Knowledge distillation

- **Idea.** Use a large model (teacher) to better train a small model (student)
  - Developed by the Nobel Laureate Geoffrey Hinton



Knowledge distillation

- **Benefits.** Student model have much increased accuracy
  - Sometimes, can even inherit the knowledge of the private dataset used by teacher

System	Test Frame Accuracy	WER
Baseline	58.9%	10.9%
10xEnsemble	61.1%	10.7%
Distilled Single model	60.8%	10.7%

Table 1: Frame classification accuracy and WER showing that the distilled single model performs about as well as the averaged predictions of 10 models that were used to create the soft targets.

Knowledge distillation

- **Key Question.** What should we distill?
  - Prediction, Features, Inter-sample relationships, Attention

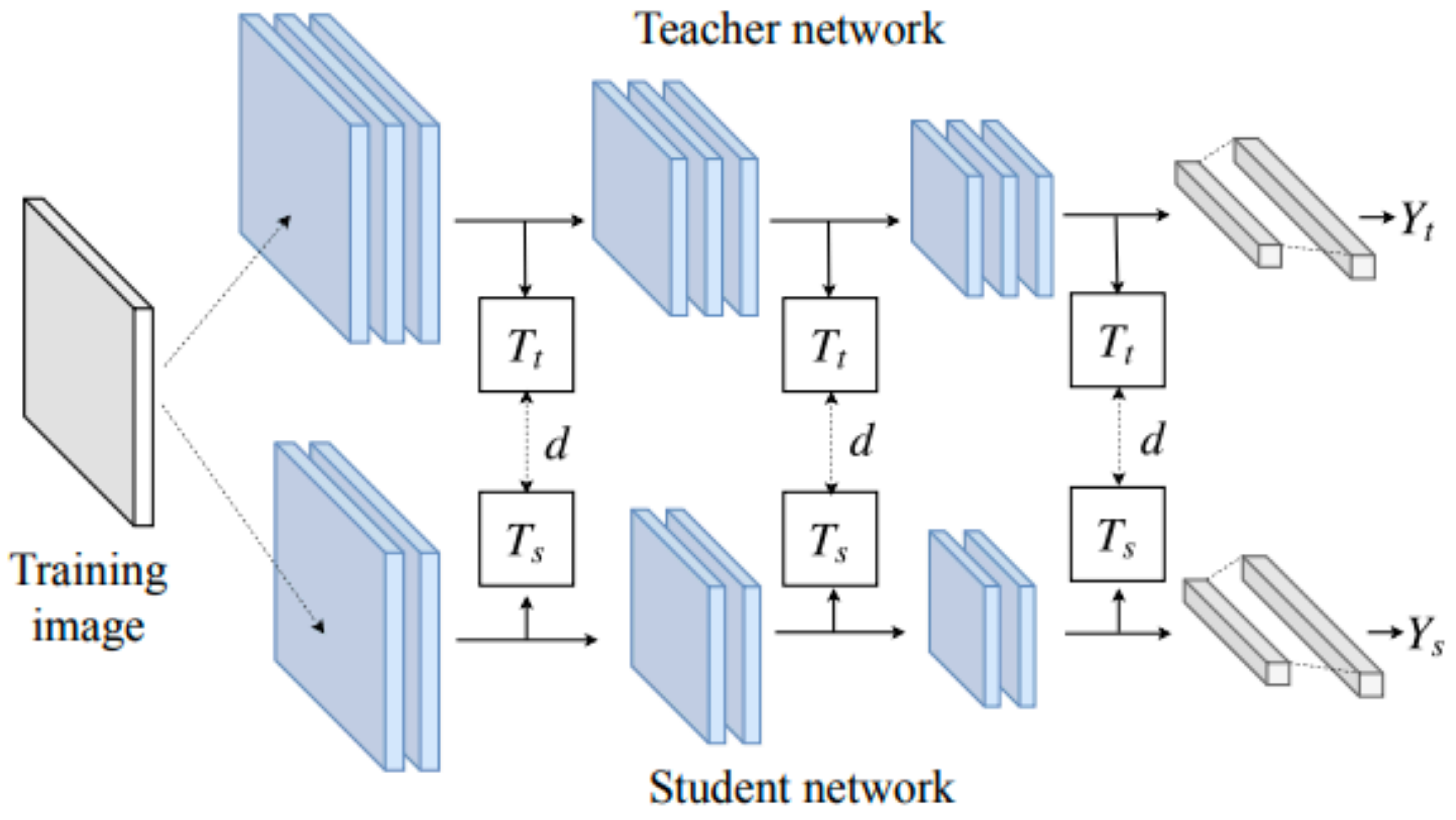
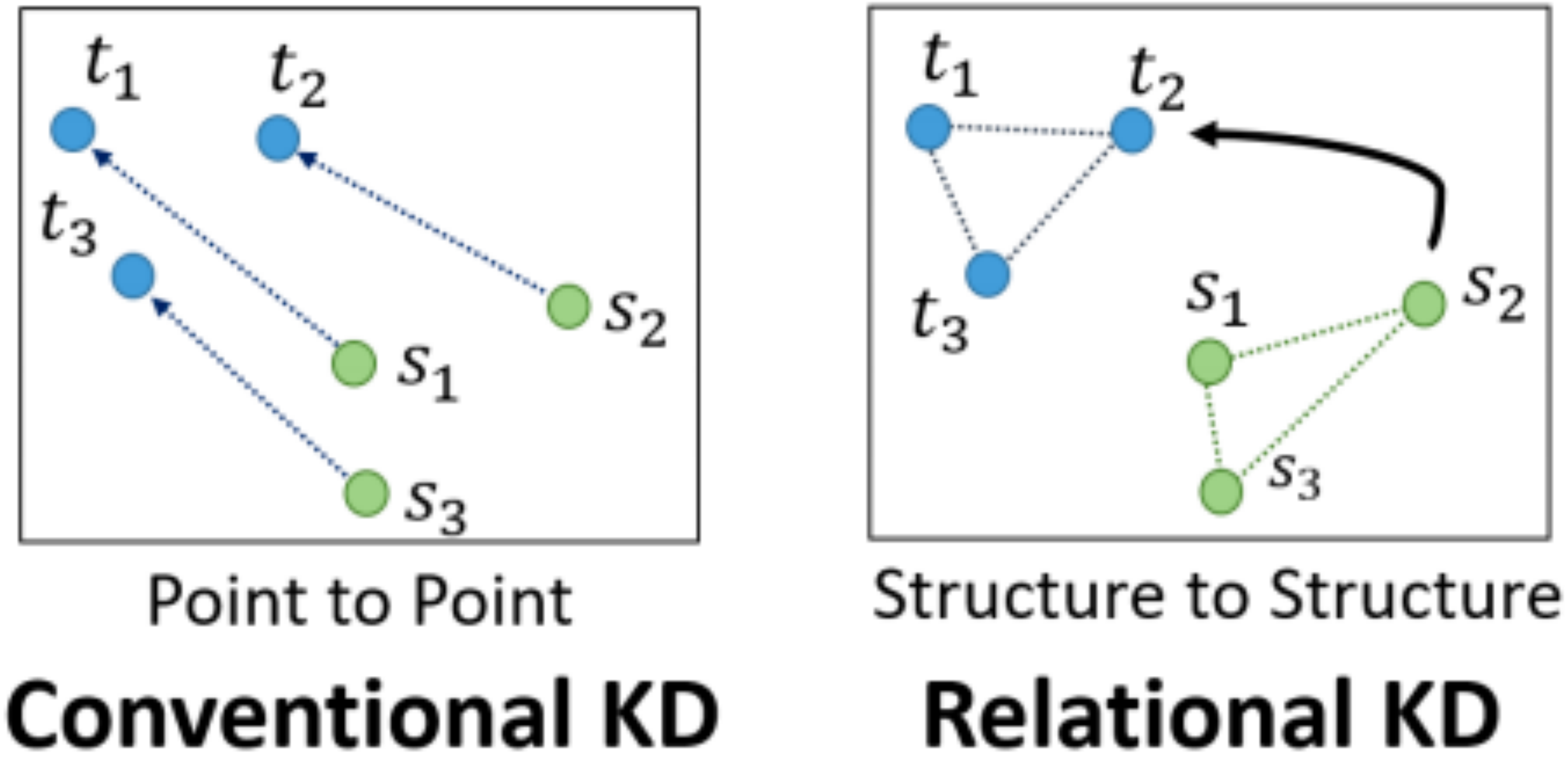
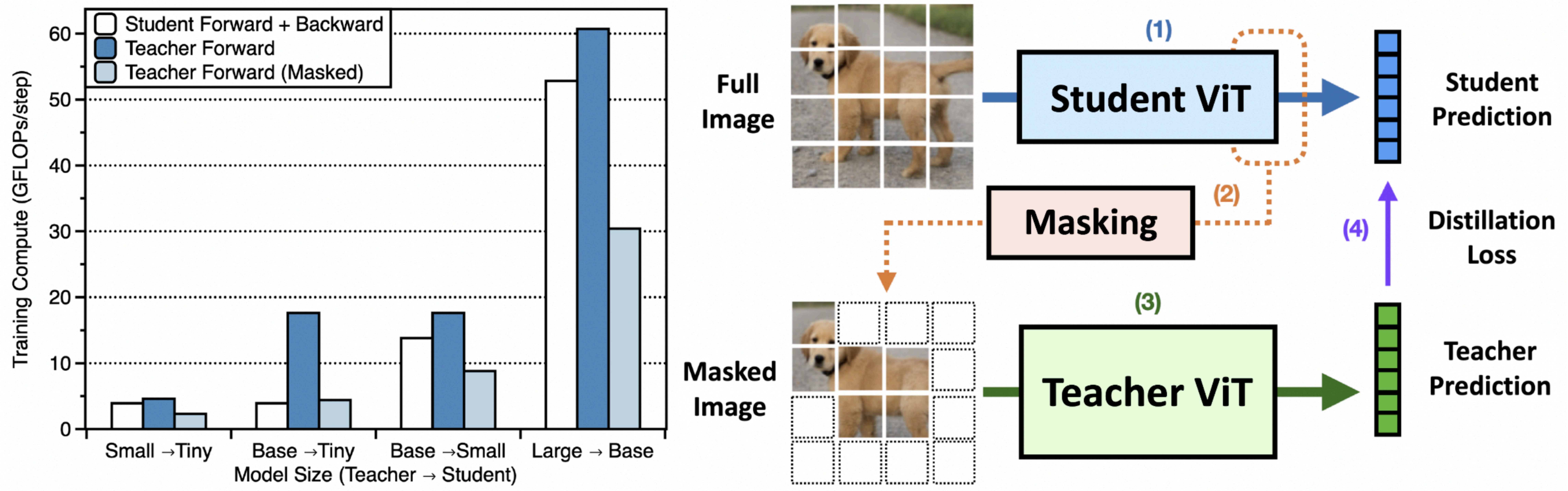


Figure 2. The general training scheme of feature distillation. The form of teacher transform T_t, student transform T_s and distance d differ from method to method.



Knowledge distillation

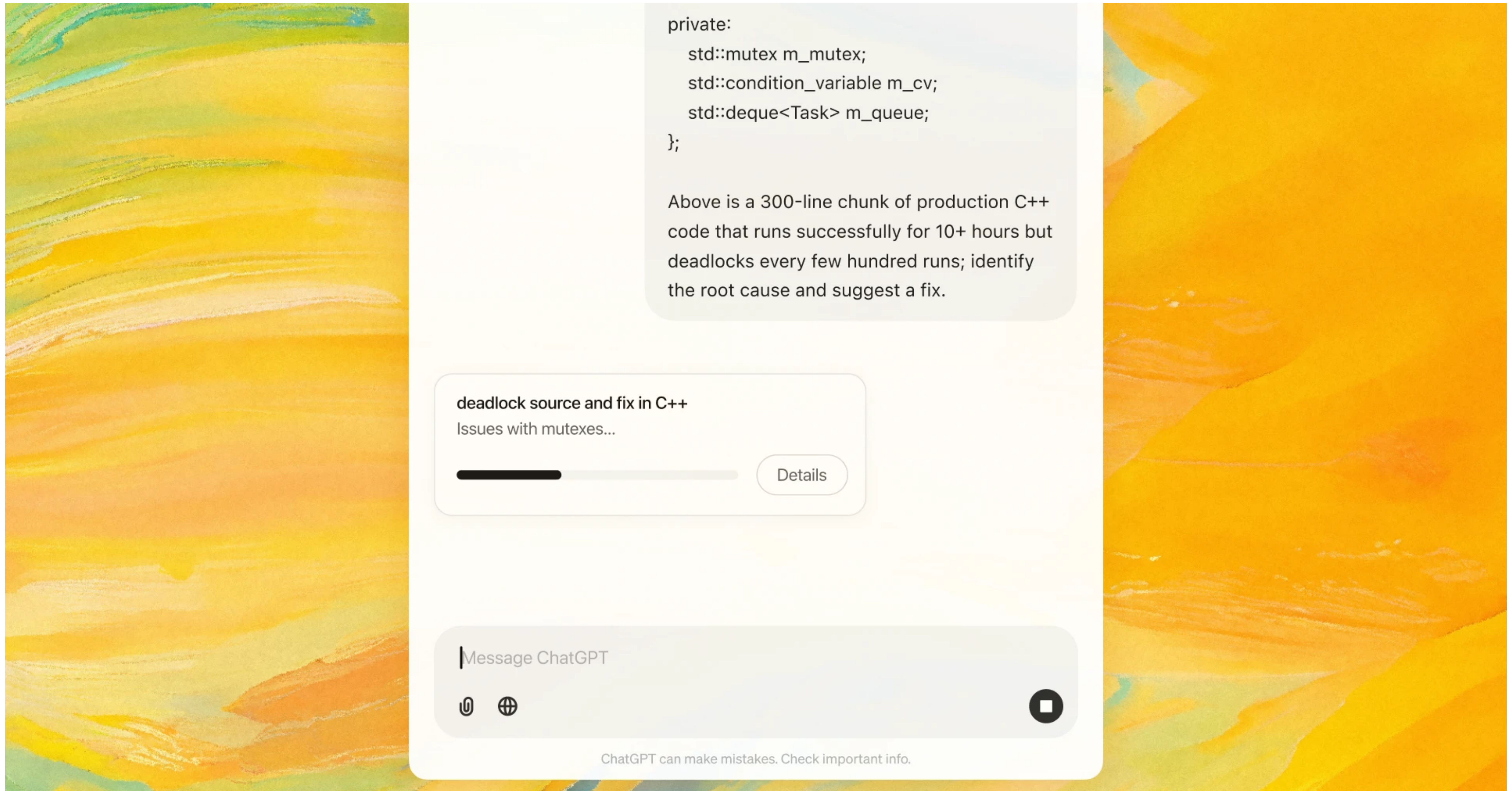
- **Trends in 2022—2024.** Applying distillation to large, commercial teachers (e.g., GPT)
  - Small transformers as students (Meta, Apple)
  - Pruned model as students (NVIDIA)
  - Masking the teacher for less training compute (POSTECH)

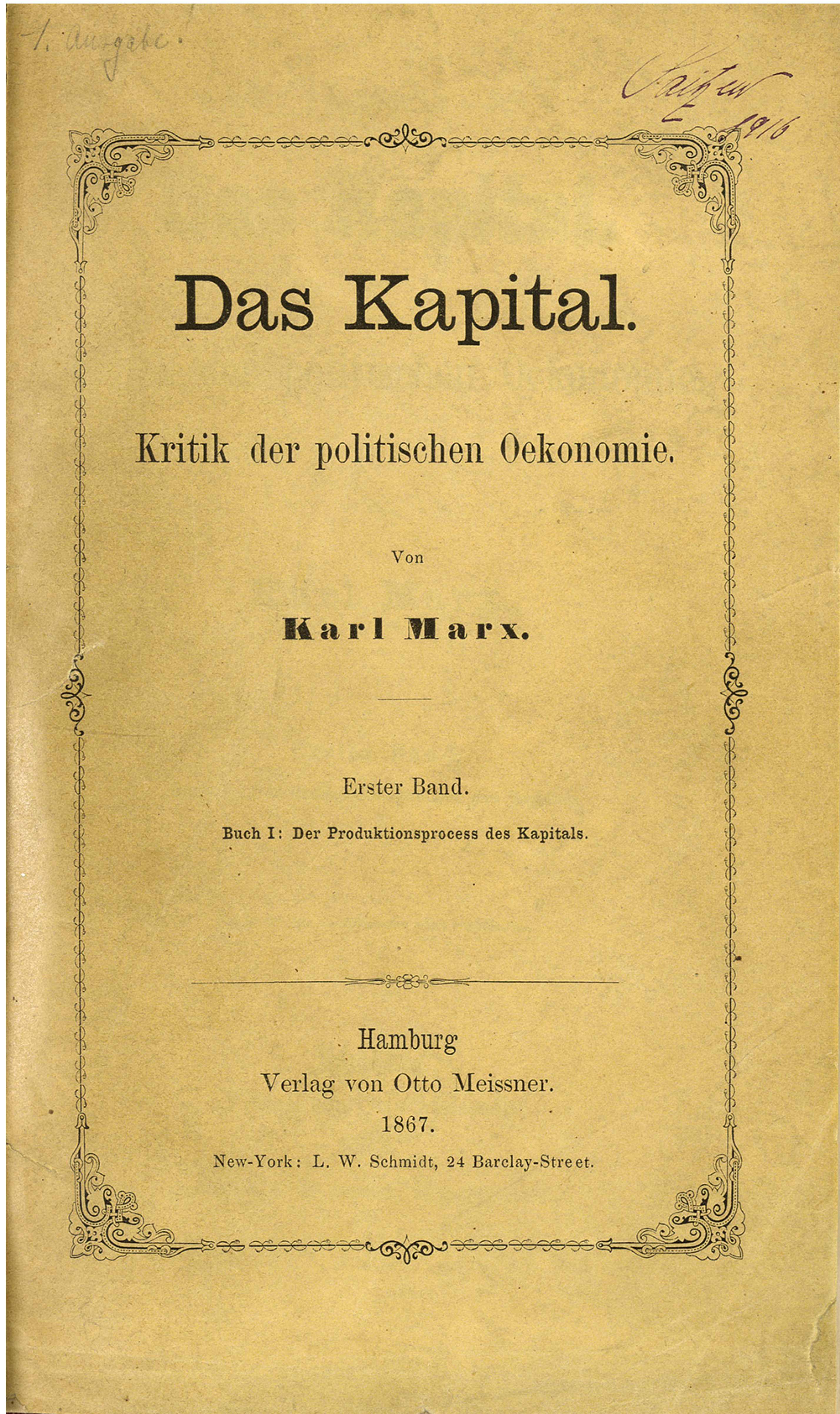


Knowledge Distillation

Concluding Remarks

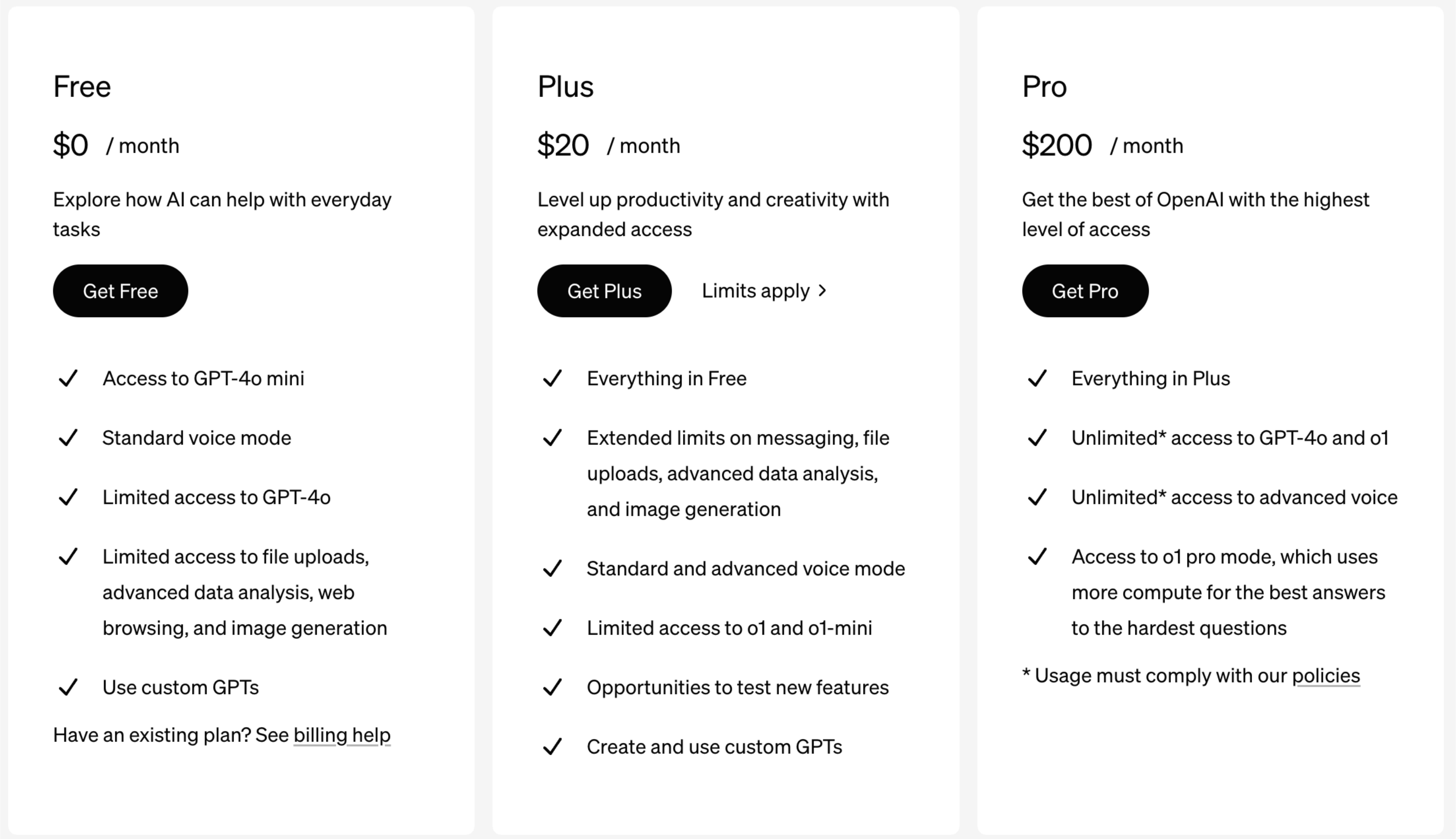
- AI is now becoming the core productivity tool
  (Coding, Scientific Discovery, Writing)
  - Medieval age: Land & Human labor
  - Industrial age: Capital
  - AI age: AI





Concluding Remarks

- We are now witnessing the beginning of the great AI divide
  - Can we stop these bourgeois from monopolizing the AI?
    (+ slow down the climate change?)



Cheers