ELEC3544 · Fall 2026

Data Science with Foundation Models

Yang Liu

The University of Hong Kong



We value your feedback!
Please join the “Student Feedback on Teaching and Learning” Survey

https://sftl.hku.hk

For Engineering UG students, you may get a $25 coffee shop gift coupon if you opt for the gift exercise and complete at least 65% of your total surveys.

Foundation Models and Agentic Data Science

Attention, pretraining, and tool-using agents for verified data science

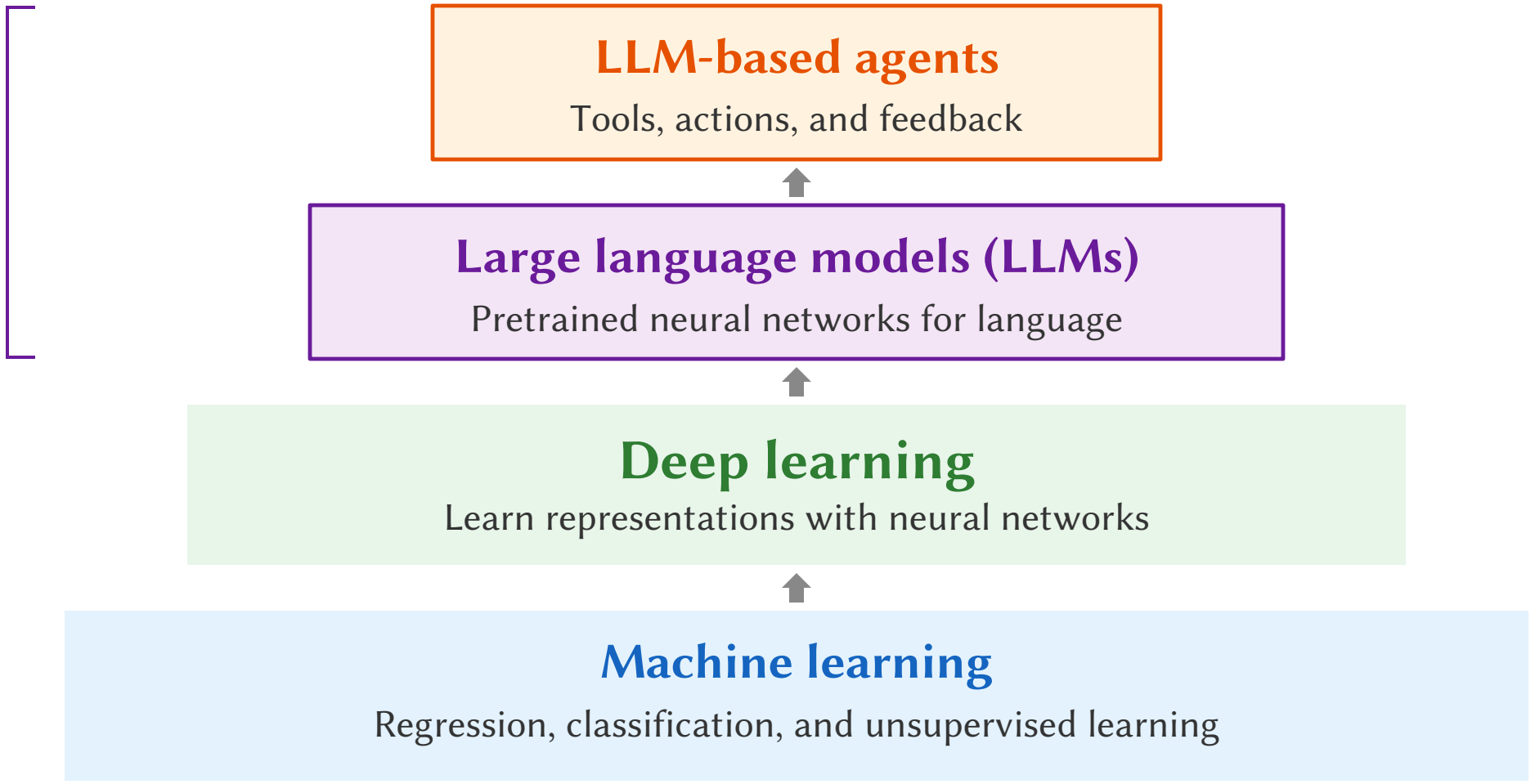
Adapted from MIT 6.7960, Caltech EE/CS148B, and Stanford CS146S

Data Science Trajectory

Today
L9

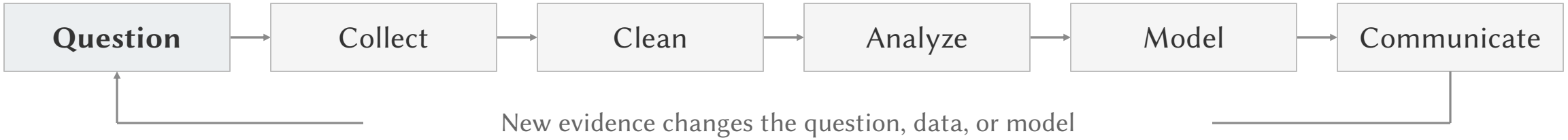
Previously
L8

Foundations
L5–L7



Why data scientists embrace AI

Data science is iterative



Expanding scope of automation

Conventional ML
Fit model parameters

Deep learning
Learn features and fit the model

LLMs
Generate code and interpret results

Agents
Use tools, test, and revise the workflow

Goal: an agentic data scientist

Faster iteration
Automate manual work and repeated revisions.
Potential: orders-of-magnitude speedups.

The hidden tradeoff
More opaque ("black-box") models and decisions.
Errors can accumulate unnoticed.

Verify and supervise at every iteration

Data
Provenance + leakage

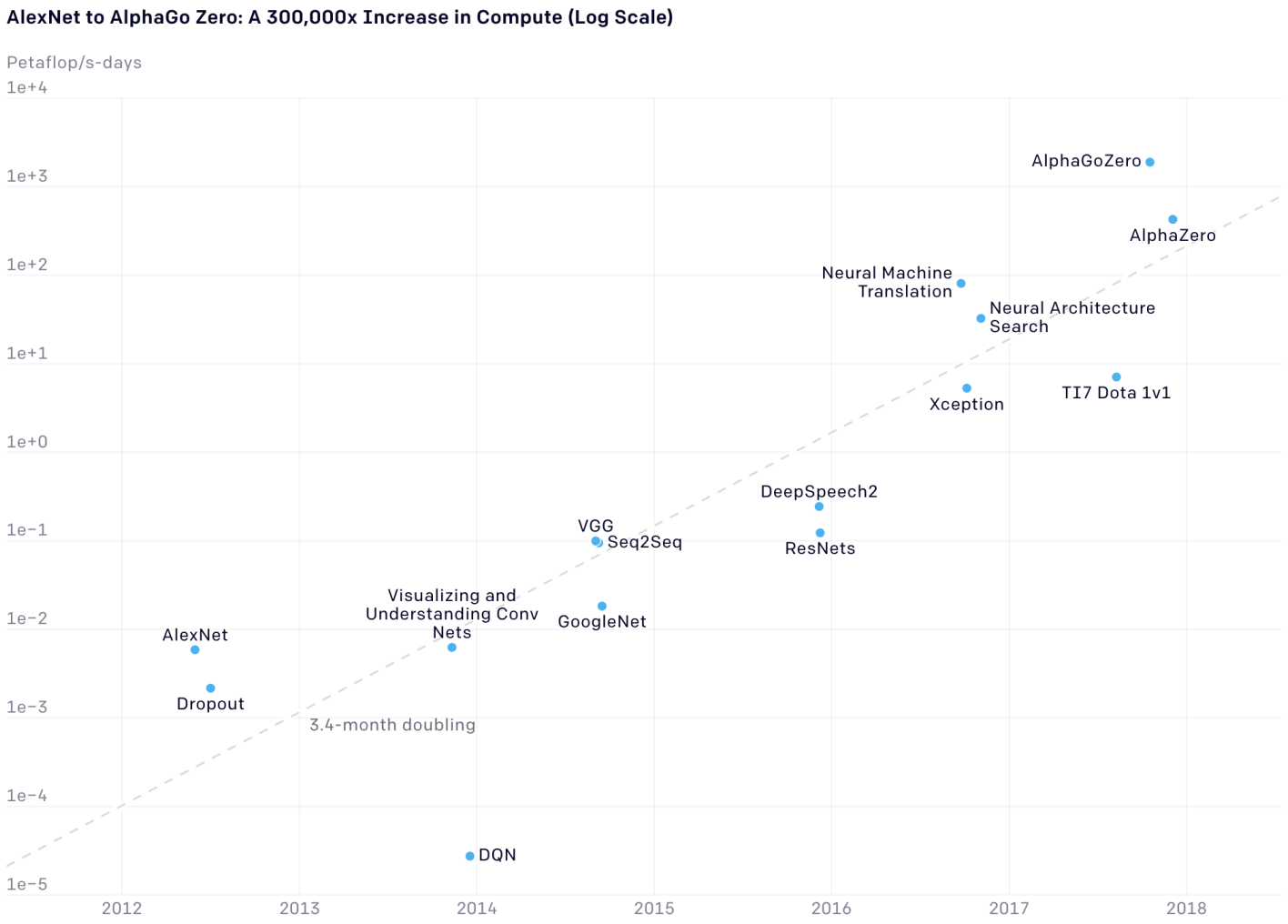
Results
Baseline + held-out test

Process
Rerun code + inspect logs

Control
Approvals + stop rules

L1 workflow; Anthropic, Building effective agents, engineering blog 2024

Deep learning model size scaling up



2012–2018 analysis

≈ 300,000×

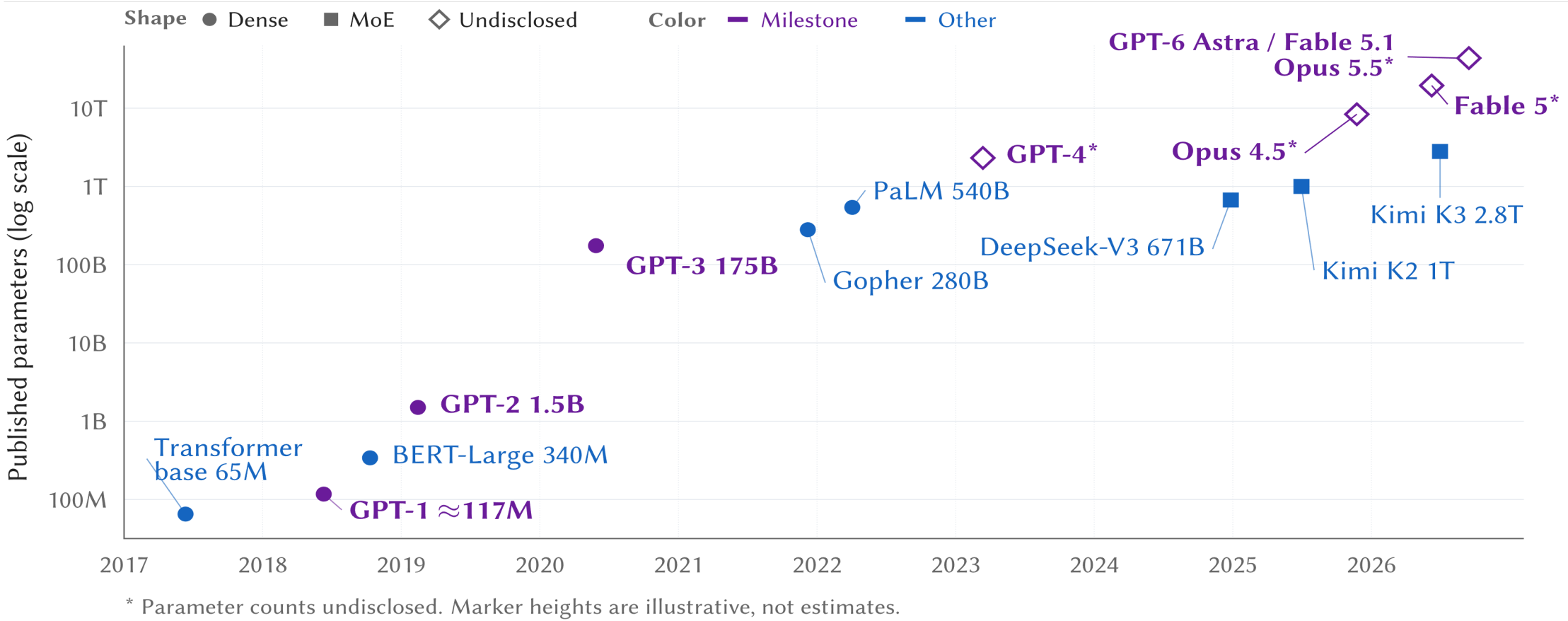
growth in compute for the largest training runs

Better chips + more chips + larger training budgets

The historical 3.4-month doubling rate is not a law for 2026.

OpenAI, AI and Compute, research blog 2018; MIT 6.7960, course notes 2026

Model size and LLM milestones



* Parameter counts undisclosed. Marker heights are illustrative, not estimates.

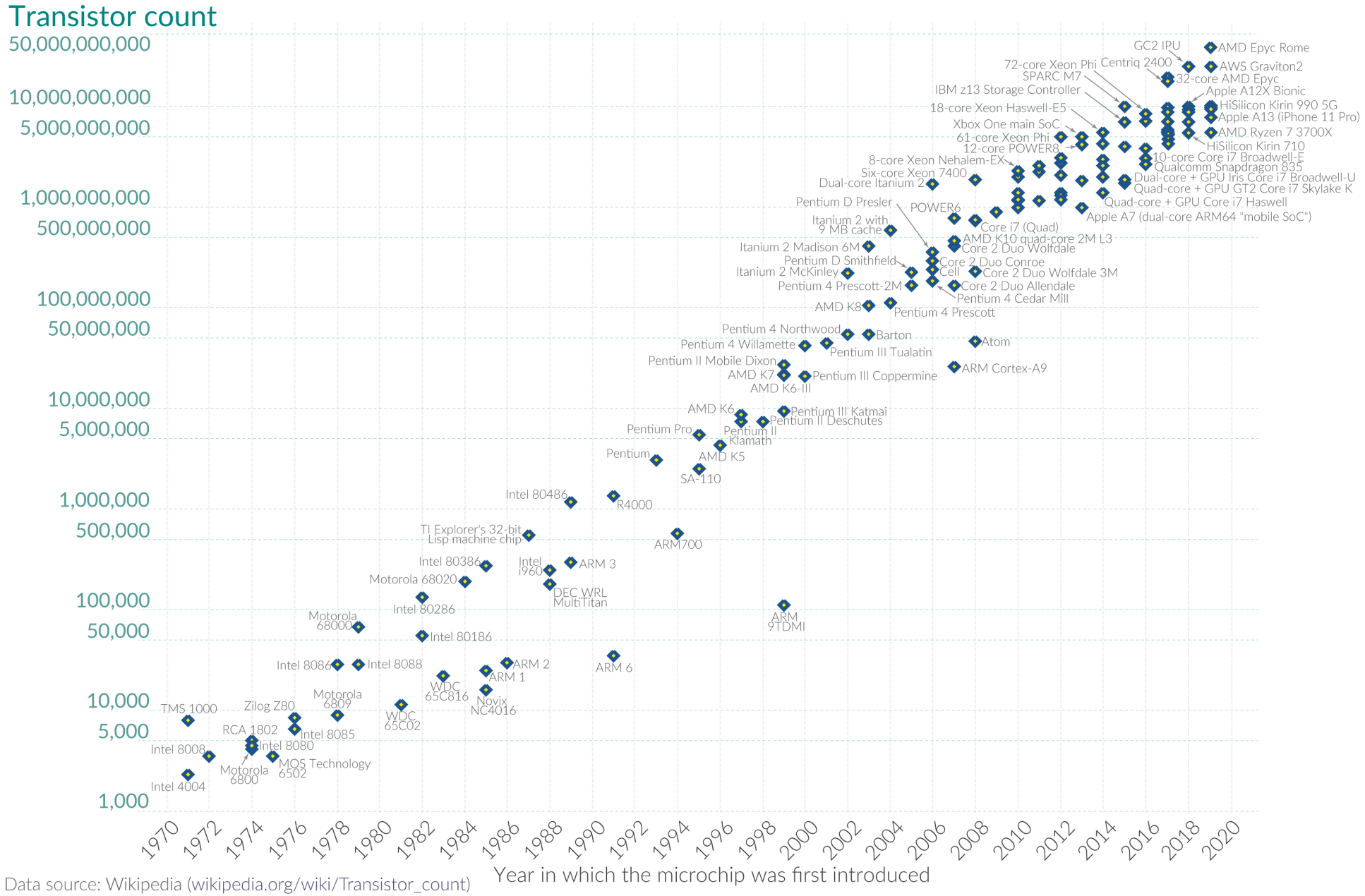
From millions to trillions of parameters: what made scaling effective?

Transformer, NIPS 2017; GPT-1, OpenAI report 2018; GPT-2, OpenAI report 2019; GPT-3, NeurIPS 2020; BERT, NAACL-HLT 2019; Gopher, arXiv 2021; PaLM, JMLR 2023; DeepSeek-V3, report 2024; Kimi K2, report 2025; Kimi K3, report 2026; GPT-4, OpenAI report 2023; Opus 4.5, release 2025; Fable 5, release 2026; GPT-6 Astra, release 2026; Fable 5.1, release 2026; Opus 5.5, release 2026

Moore's Law: The number of transistors on microchips doubles every two years

Moore's law describes the empirical regularity that the number of transistors on integrated circuits doubles approximately every two years. This advancement is important for other aspects of technological progress in computing – such as processing speed or the price of computers.

Our World in Data

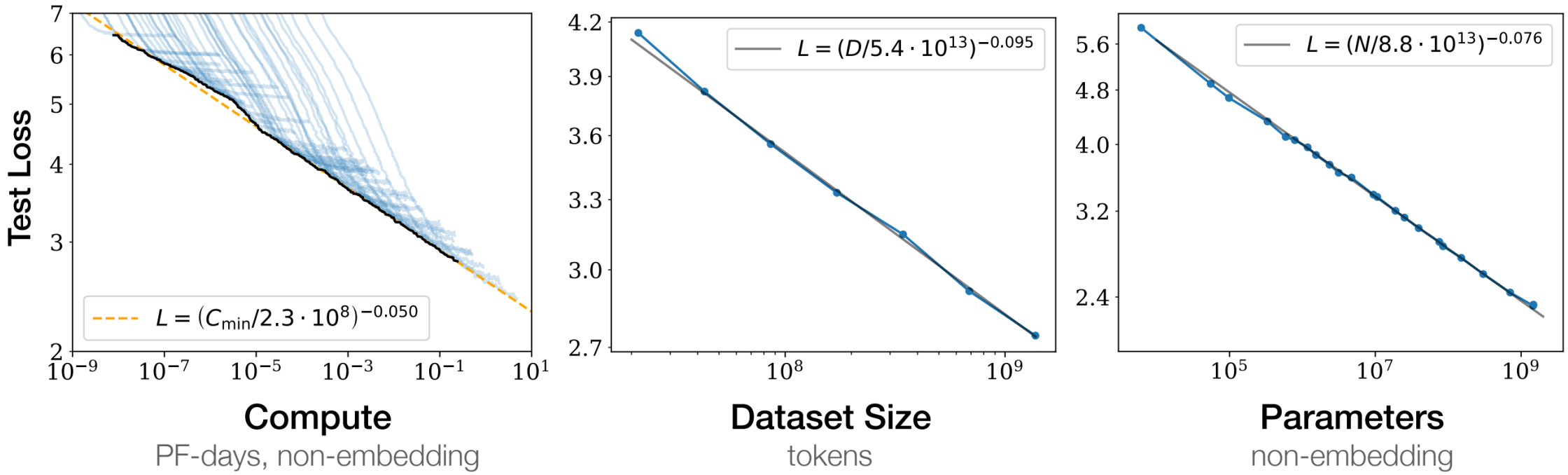


Data source: Wikipedia (wikipedia.org/wiki/Transistor_count)

OurWorldinData.org – Research and data to make progress against the world's largest problems.

Licensed under CC-BY by the authors Hannah Ritchie and Max Roser.

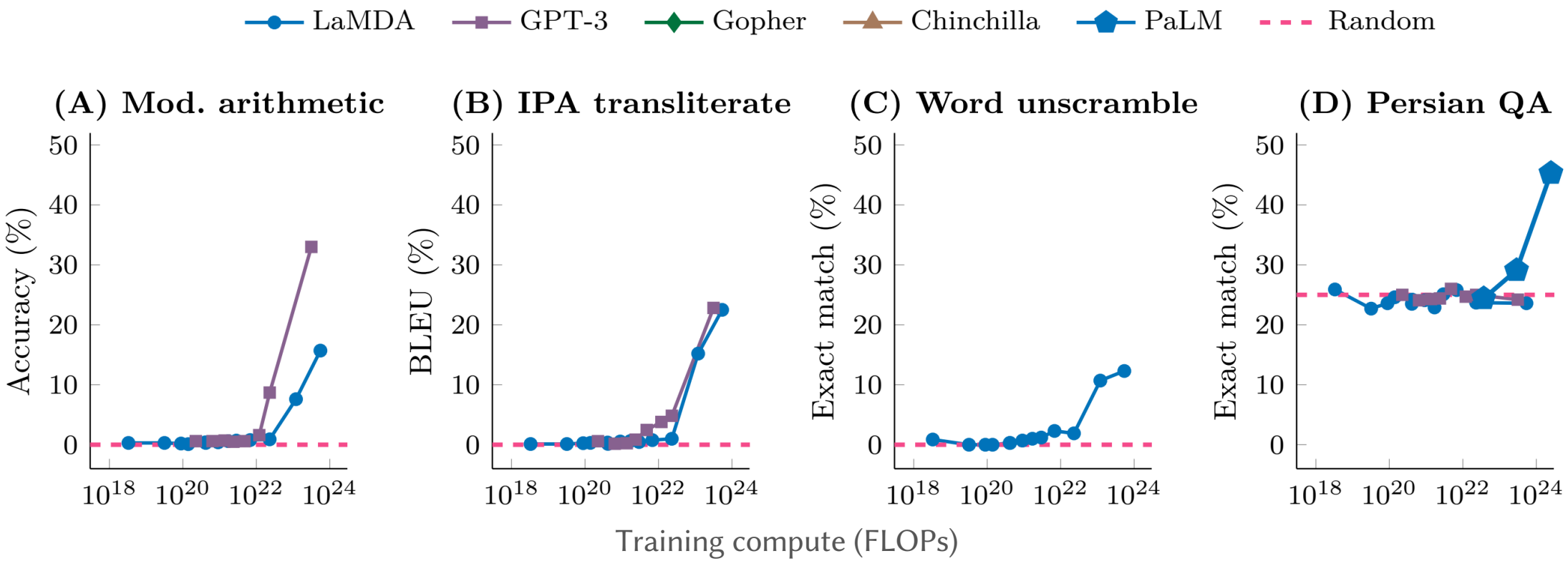
Neural scaling laws



Prediction loss improves smoothly as compute, data, and parameters scale.

Each curve varies one resource while the other resources are not limiting.

Emergent abilities after scaling

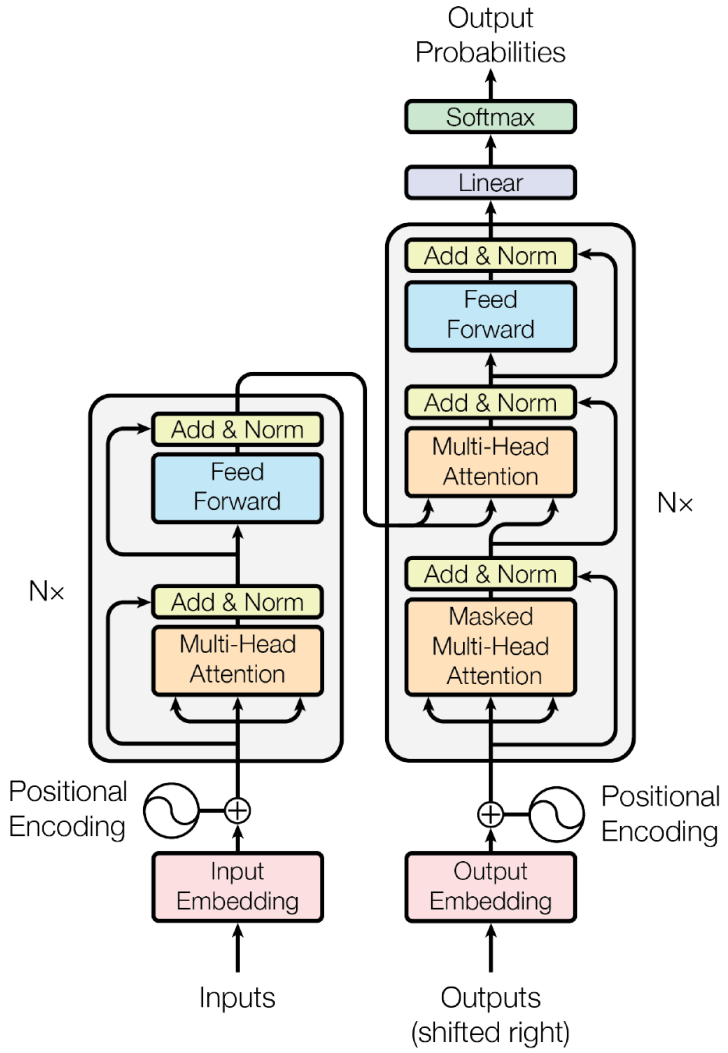


Smooth loss improvements can accompany sharp task-performance gains.

Apparent jumps depend on the task and evaluation metric.

Wei *et al.*, TMLR 2022; Schaeffer *et al.*, NeurIPS 2023

Attention is All You Need



The architectural change

1 No recurrent state chain

Process known token positions together.

2 Direct access to distant context

Choose information by content, not distance.

3 Large, repeatable matrix operations

Turn more compute into larger training runs.

Scaling needs both useful context and efficient computation.

Vaswani *et al.*, NIPS 2017 · original Transformer architecture

Two bottlenecks: waiting and locality

The original comparison: parallel work and information paths

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

The table's logarithmic CNN path uses dilation; the demo uses contiguous local kernels.

RNN / LSTM: sequential states

$$h_t = f_\theta(h_{t-1}, x_t)$$

Gates help memory; they do not remove recurrence.
The next state still waits for the previous state.

Local CNN: growing receptive field

$$R_L = 1 + 2L$$

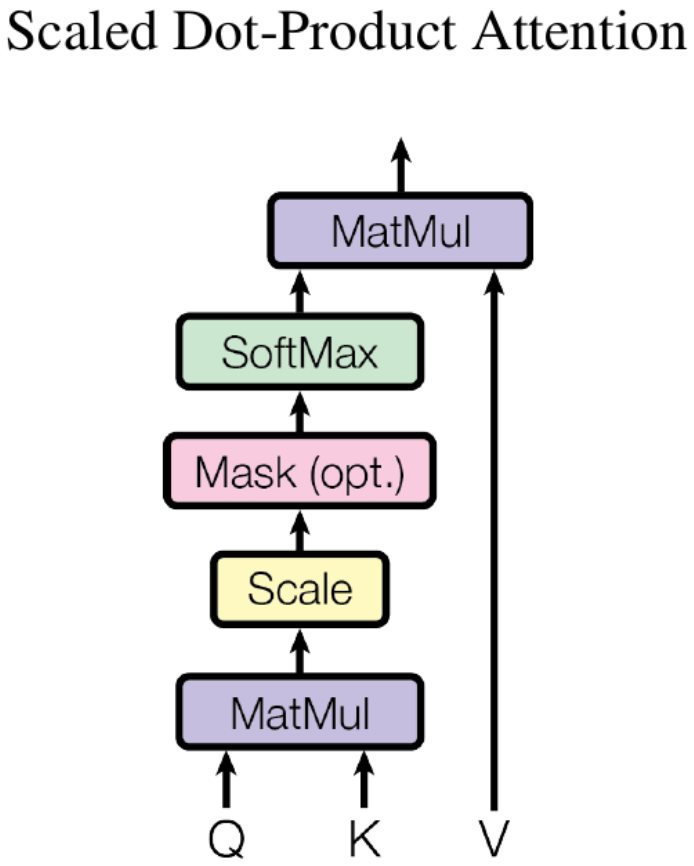
Width 3, no dilation: 8 layers for 16 tokens.
Positions are parallel. Dilation shortens the path.

Attention combines parallel token processing with direct access to distant context.

Vaswani *et al.*, NIPS 2017 · original Table 1; n: length, d: width, k: kernel, r: local attention window

Gehring *et al.*, ICML 2017

Attention: match queries, then mix values



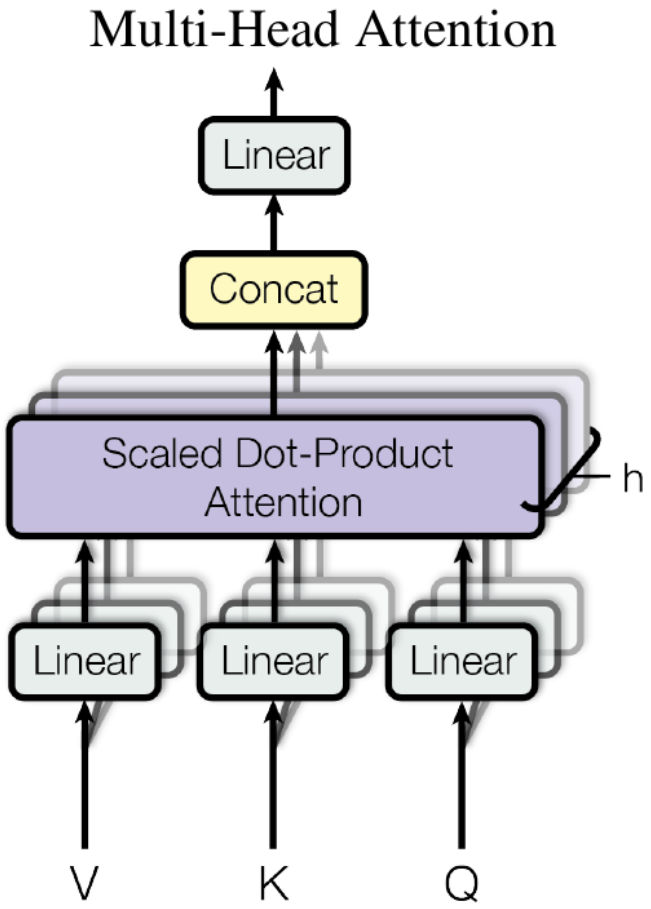
Query	What does this position need?
Key	What does each position offer?
Value	What information will be combined?

A content-dependent shortcut

A late query can read an early token in one layer.
The same head may attend elsewhere for another input.

Content-dependent connections replace long chains of local communication.

All token queries become matrix operations



Stack the tokens into rows of X

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V$$

Compute every query–key score together

$$A = \mathrm{softmax}_{\mathrm{row}}\left(\frac{QK^{\mathsf{T}}}{\sqrt{d_k}}\right), \quad Z = AV$$

Rows: token positions Heads: learned projections

Dense matrix multiplication uses GPU parallelism.
Each head mixes information in a different learned way.

Parallel token processing is not constant work: dense attention has quadratic pairwise cost.

Vaswani *et al.*, NIPS 2017 · original multihead attention; single-head equation at right

A causal mask makes parallel training valid

TRAINING: the text is already known

The	cat	sat	down	Input
↓	↓	↓	↓	
cat	sat	down	<end>	Target

Each row can read only its prefix

	1	2	3	4
1	read	×	×	×
2	read	read	×	×
3	read	read	read	×
4	read	read	read	read

All four losses in one masked forward pass

$$A = \mathrm{softmax}_{\mathrm{row}}\left(\frac{QK^{\mathsf{T}}}{\sqrt{d_k}} + M\right), \qquad M_{ij} = \begin{cases} 0 & j \le i \\ -\infty & j > i \end{cases}$$

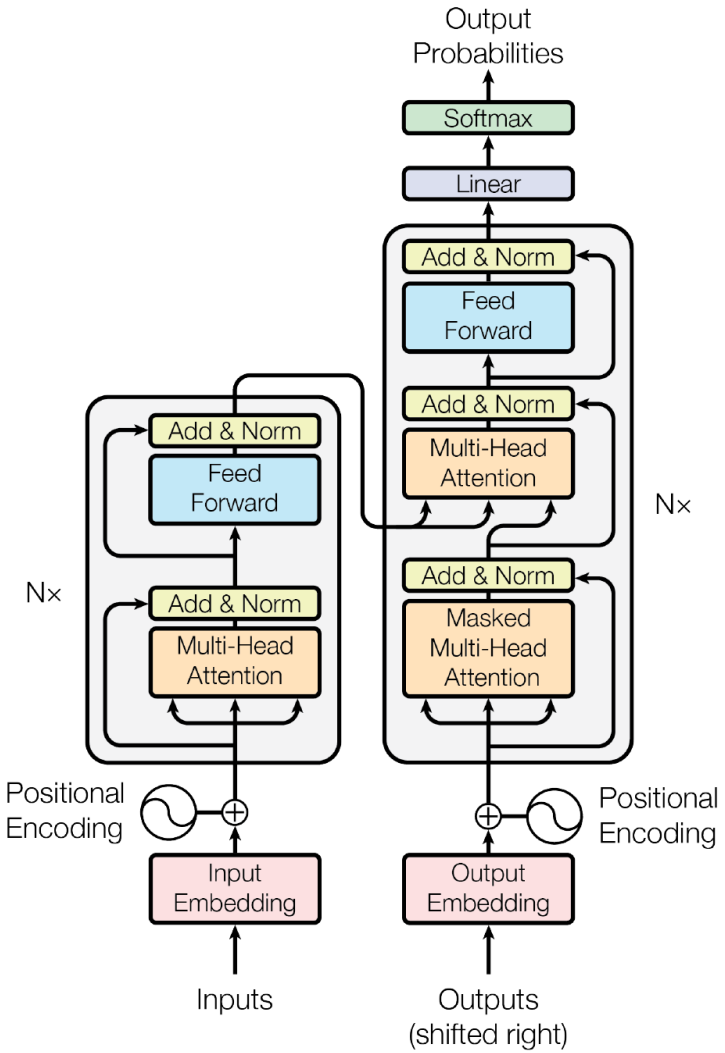
Future keys are blocked

GENERATION The cat ⟶ The cat sat ⟶ The cat sat down

Known tokens train in parallel. Unknown tokens are generated sequentially.

Vaswani *et al.*, NIPS 2017 · causal masking and shifted decoder targets; toy training sequence

A repeatable architecture that grows with compute



More data

Shared weights learn from many next-token targets.

Larger models

Increase depth, vector width, and MLP capacity.

More compute

Run larger matrix operations and distributed training.

Residuals + normalization support deep optimization.
Position information supplies token order.

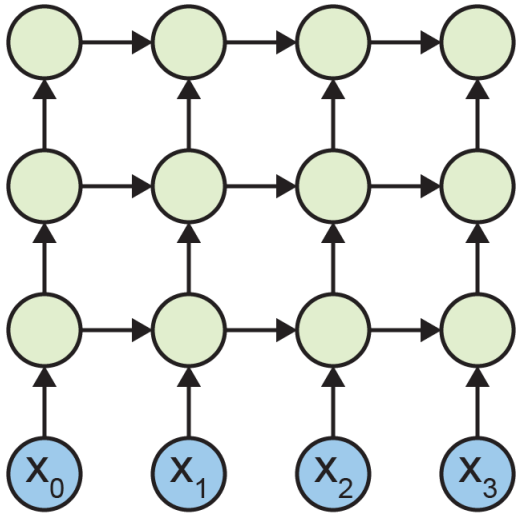
Original: encoder–decoder. GPT: causal decoder-only, without the encoder–decoder attention block.

Architecture enables scaling. Data quality and optimization determine the gain.

Vaswani *et al.*, NIPS 2017 · original encoder–decoder architecture

Why attention made language scaling practical

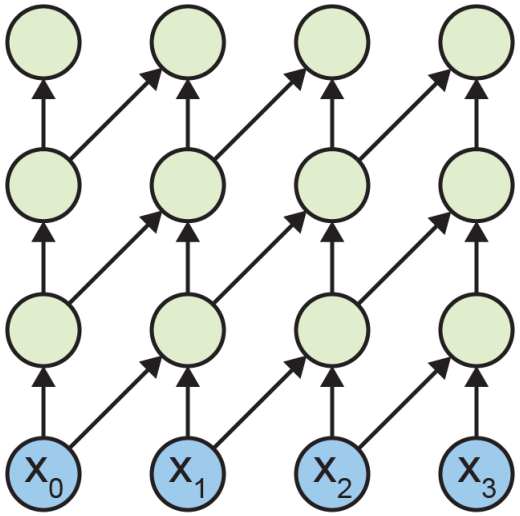
RNN / LSTM



Sequential state updates

Long dependencies pass through many recurrent steps.

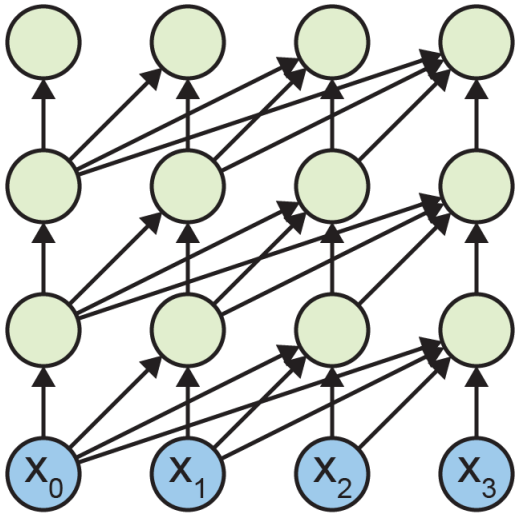
Local CNN



Parallel within each layer

Receptive field grows with depth or dilation.

Self-attention



Global, content-dependent mixing

Distant tokens interact within one attention layer.
Cost: all token pairs, $O(n^2)$ in length.

CNNs also scale. Attention combines parallel training, global context, and short information paths.

Vaswani *et al.*, NIPS 2017; Diagrams: MIT 6.7960, sequence modeling course notes

CTA Demo 1: why attention scales

What We'll Explore

How does information travel from the first token to the last?

1 Specify the task

Ask CTA to compare RNN/LSTM, local CNN, and causal attention on 16 tokens.

2 Visualize and run

Generate reachability heatmaps with NumPy. Run in Colab and trace token 1.

3 Scale and explain

Repeat with 128 tokens. Compare path length, parallelism, and attention cost.

VERIFY Fix the CNN: causal width 3, stride 1, no dilation or pooling.
Check first-to-last reachability numerically; no model training is needed.

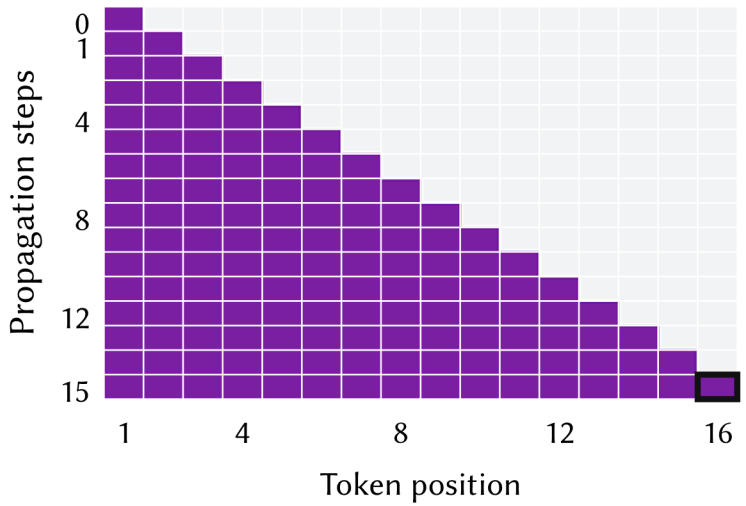
Predict → ask DeepSeek → run → inspect → explain

Vaswani *et al.*, NIPS 2017 · instructor-created connectivity comparison

CTA Demo 1: expected results and comparison

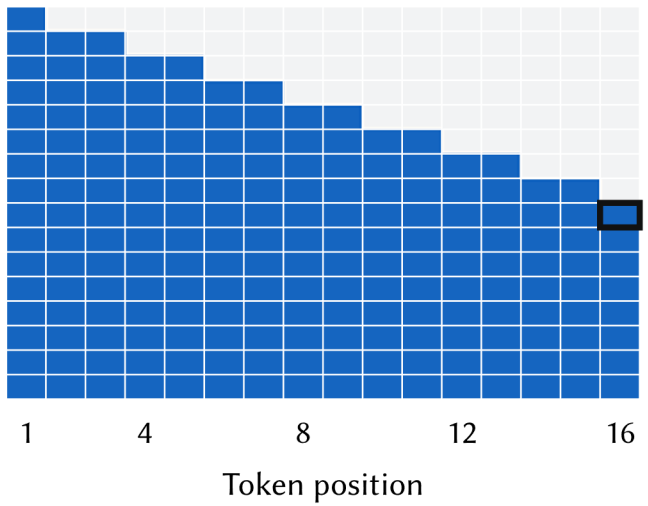
16 tokens. Colored cells: reachable from token 1. Black box: first arrival at token 16.

RNN / LSTM



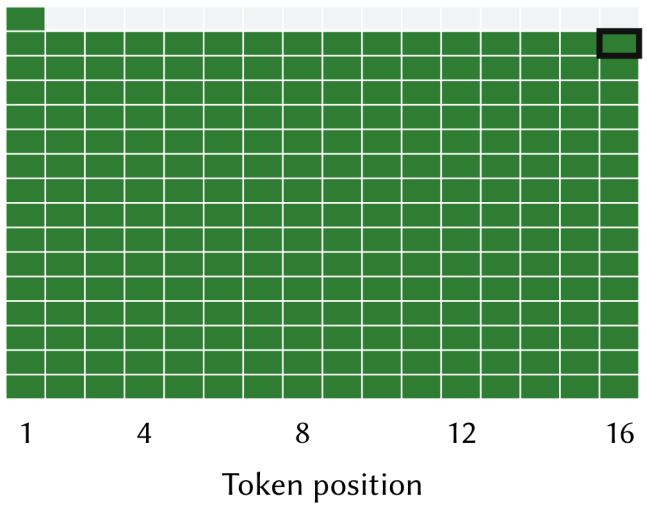
15 recurrent hops

Local causal CNN



8 convolution layers

Causal self-attention



1 attention layer

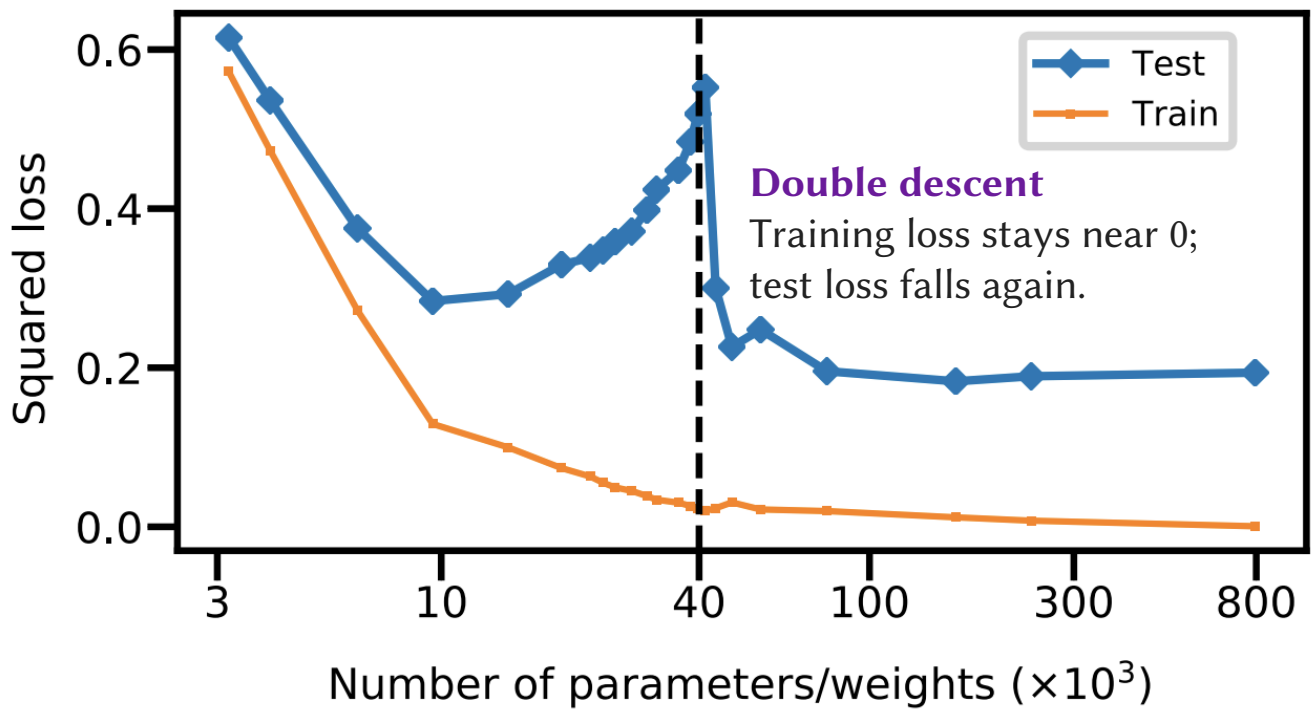
At 128 tokens: **127 recurrent hops** **64 CNN layers** **1 attention layer**

Attention combines short paths and parallel training; dense attention still has quadratic cost.

Vaswani *et al.*, NIPS 2017 · possible information paths, not measured runtime or prediction accuracy

More parameters: why not simply overfit?

One hidden-layer MLP • 4,000 MNIST training examples



Dashed line: interpolation threshold (the model fits every training example)

Parameter count ≠ effective complexity

Structured data
Architectural bias
Optimization + regularization

An illustration, not a complete explanation or a guarantee of LLM generalization.

Many functions fit the training data. Which function we learn determines generalization.

Belkin *et al.*, PNAS 2019 · original empirical MNIST result

why and how scaling works? - pretraining

Who labeled GPT-3's 300 billion training tokens?

Task-specific supervision

Review	Sentiment label
"Great service!"	**positive**
"Never coming back."	**negative**

A target chosen for this task

Pretraining

The food was delicious.
The service was excellent.
We will come back.

Can the text supply its own targets?

A scalable learning recipe needs supervision that grows with the data.

Brown *et al.*, NeurIPS 2020

The text already contains the answers

The capital of France is Paris.

Prefix the model can see		Next-token target
The capital of	→	France
The capital of France	→	is
The capital of France is	→	Paris

Predict → **Compare with next token** → **Update weights**

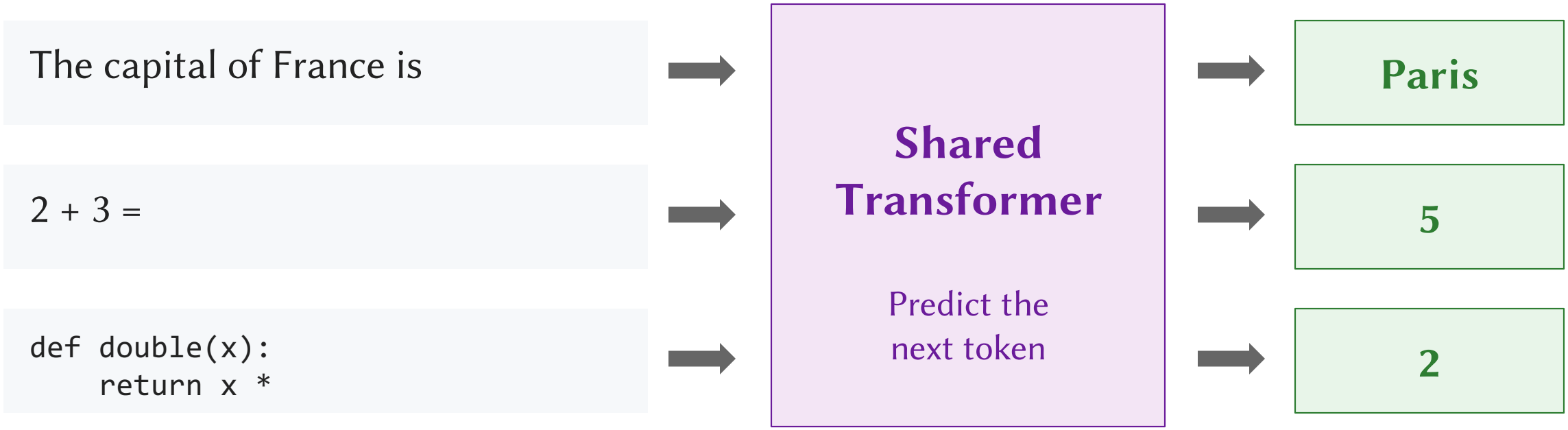
Every position supplies a training target—without a new annotation.

Brown *et al.*, NeurIPS 2020

One prediction problem, many kinds of structure

Different examples. Same model and learning rule.

Text ⟶ tokens ⟶ learned vectors



More data
More training targets

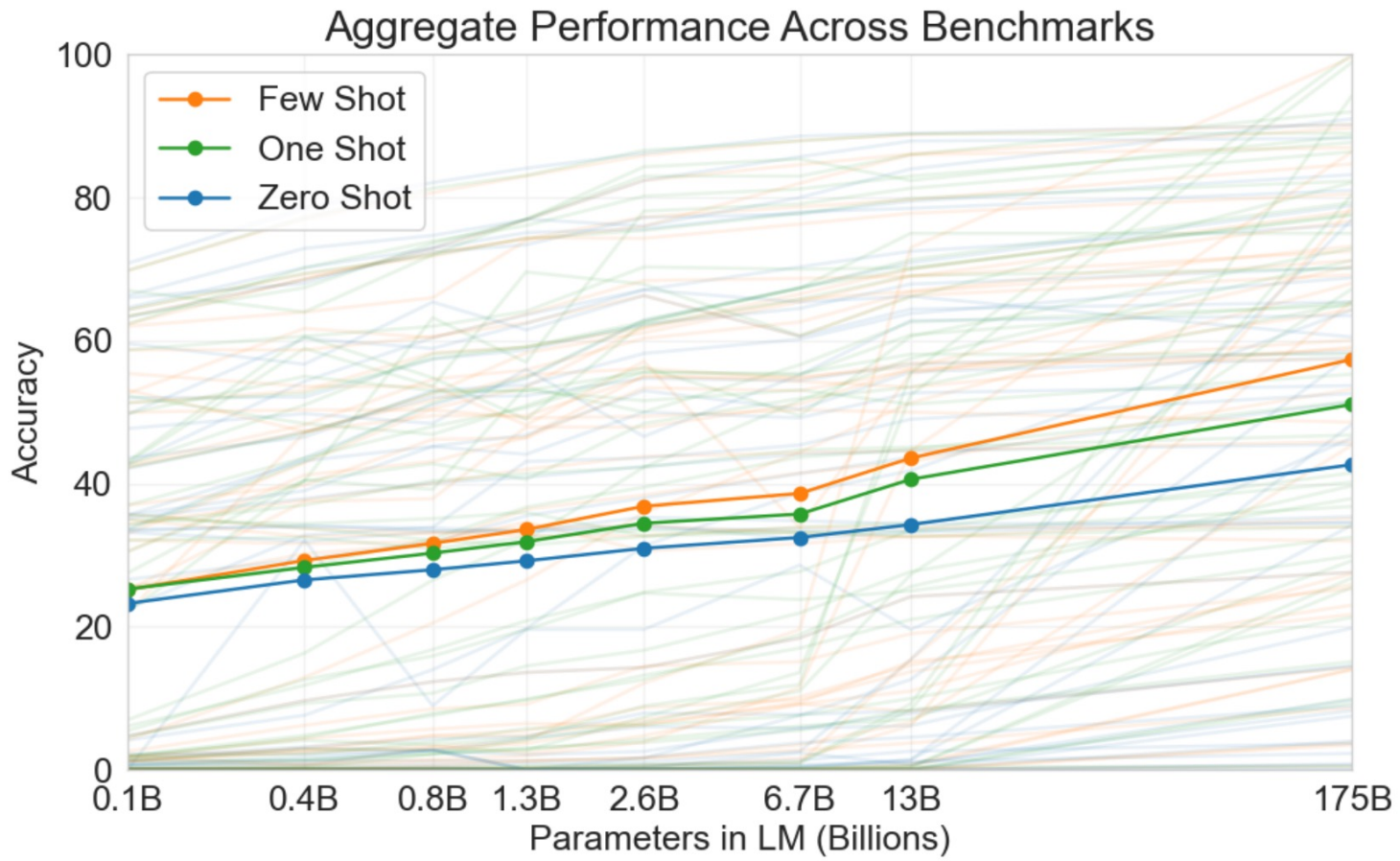
More parameters
More representational capacity

More compute
Larger training runs

Pretraining couples a broad learning signal with a scalable architecture.

Brown *et al.*, NeurIPS 2020; Vaswani *et al.*, NIPS 2017

GPT-3: scale enabled useful few-shot transfer



175 billion parameters

One pretrained model

New tasks specified
by examples in context

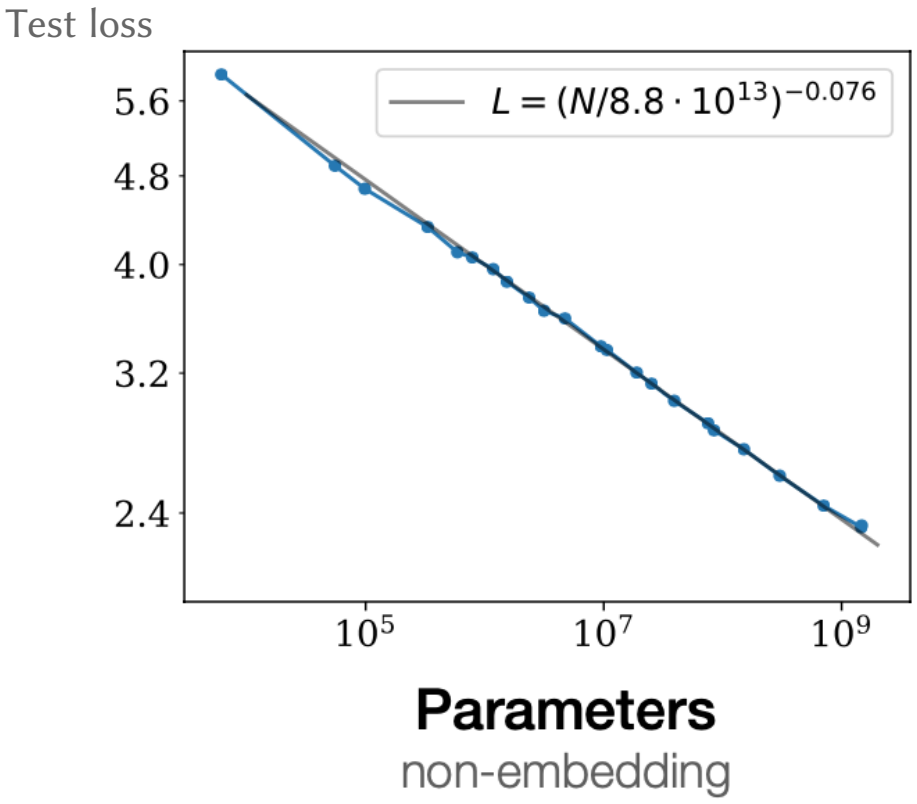
2020 milestone

The prompt changes the task; the pretrained model weights stay fixed.

Brown *et al.*, NeurIPS 2020

Scaling laws: predictable loss, diminishing returns

Read the trend introduced at the beginning



$$L(X) \approx AX^{-\alpha}$$

X: parameters, tokens, or compute
A, α: empirical fit in a measured regime
Other resources fixed or not limiting

A straight trend on log–log axes means a multiplicative improvement.

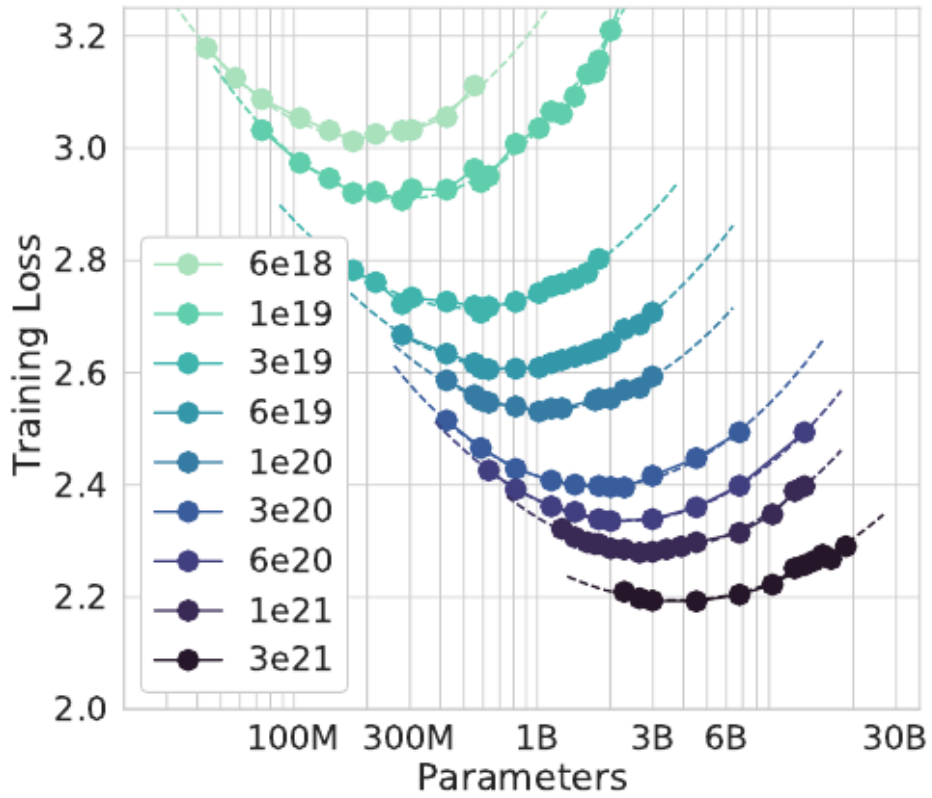
A better token loss does not guarantee a correct answer on a particular task.

Pilot runs help estimate what another order of magnitude of resources will buy.

Kaplan *et al.*, arXiv 2020 · empirical parameter-scaling fit

A fixed budget: model size or more training data?

Each curve = one fixed training-compute budget



$$C \approx 6ND$$

Dense-model approximation

C: training FLOPs
N: model parameters
D: training tokens

Same budget, larger N
→ fewer training tokens D

The valley is the best allocation for that compute budget.

Scale model capacity and training data together; parameter count alone is not the objective.

Hoffmann *et al.*, NeurIPS 2022 · original IsoFLOP curves

Chinchilla: a better allocation of compute

Same reported training budget

	Gopher	Chinchilla
Parameters	280B	70B
Training tokens	300B	1.4T
MMLU, 5-shot	60.0%	67.6%

$$C \approx 6ND$$

C: compute N: parameters D: tokens

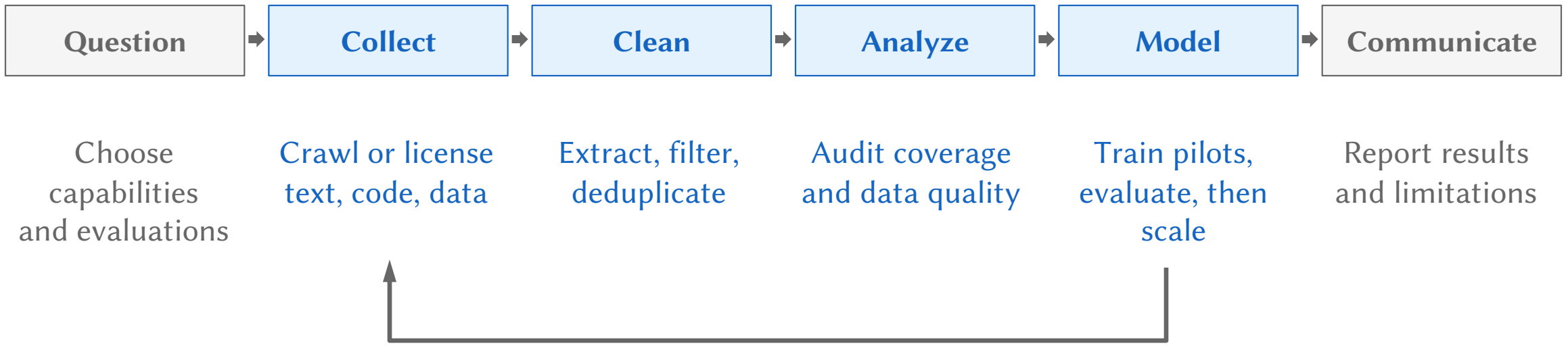
One-quarter as many parameters

More data; stronger results

Scaling model size alone can waste the training budget.

Hoffmann *et al.*, NeurIPS 2022; Khattab, MIT 6.7960 course notes 2025

Frontier pretraining is a data science pipeline



Evaluation changes what we collect and clean next.

The corpus is engineered. It is not simply "the entire internet."

Data curation and model training form an iterative loop.

Khattab, MIT 6.7960 course notes 2025; Brown *et al.*, NeurIPS 2020

Pretraining data is engineered at scale

Inside Meta's Llama 3 data pipeline

Extract	Deduplicate	Filter	Score	Mix
Preserve math and code	URL, document, and line	Rules reject noisy content	Models assess content quality	Classify domains; rebalance sources

How does quality filtering scale?

Llama 2
Annotate web examples

→

DistilRoBERTa
Learn a compact classifier

→

Web corpus
Score at scale

Small pilot runs test the filters and mixture before training at scale.

Data engineering decides what the model practices, and how often.

Grattafiori *et al.*, arXiv 2024

Better data improves the same model

FineWeb-Edu: a reproducible data-selection experiment

Build the data filter

460,000 pages
Llama 3 scores educational value

↓

Frozen embeddings
Fit a small linear regressor

↓

15T corpus tokens
Score and retain 1.3T tokens

Our toolkit: embeddings + regression

Same 1.71B model • same 350B training tokens

Accuracy	FineWeb	FineWeb-Edu
MMLU	33%	**37%**
ARC	46%	**57%**

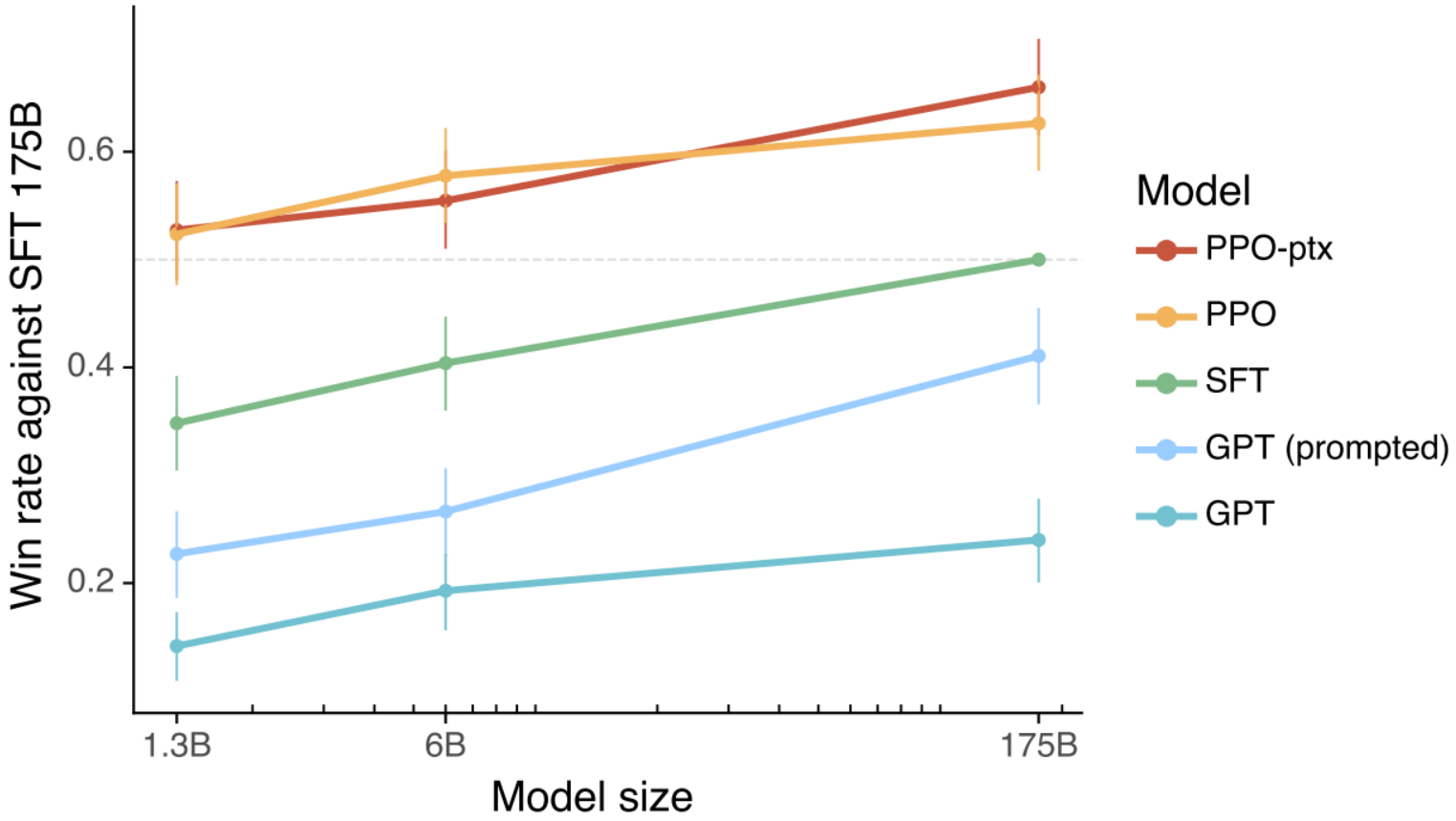
Reported, rounded scores. Gains depend on the evaluation domain.

Training-data quality can improve capability without increasing model size.

Penedo *et al.*, NeurIPS D&B 2024

A capable predictor is not yet a useful assistant

InstructGPT (2022): teach the pretrained model to follow human intent - RLHF



1.3B + feedback

preferred over

175B base GPT-3

on the study's
API-prompt distribution

Post-training changes what the model is rewarded for doing.

Ouyang *et al.* [OpenAI], NeurIPS 2022

Group Project

- Report and slides due Nov 17 at 11:59PM; requirements, template, and group order timetable are released
- Presentations Nov 18 and Nov 25. 10 min per group (9min+1min Q&A)
- Attendance and peer reviews (make sure you have received your Private Access Code `XXXX-XXXX-XXXX` and can log into the peer review system) `https://tinyurl.com/ELEC3544PeerReview`
- We will use peer reviews as references

Assignment 3

Check the current course announcements for submission deadlines, release dates, and group project details.

Session 1 | Wednesday November 18 2026

CPD-1.21 | 16:00-18:50 | Hong Kong time

10 minutes per group: suggested 9-minute presentation + 1 minute for Q&A (if any) and handover. The split is flexible. A TA will time each slot; there is a hard cutoff at 10 minutes.

Time	Group	Members
16:00-16:05	-	Opening and equipment checks
16:05-16:15	9	Fang Yi Zhou Edwin; Leung Kit Yiu; Li Junlou
16:15-16:25	30	Chow Yui Hei; Jawaid Saifullah; Wang Hong Xi
16:25-16:35	17	Ahmed Fahim; Ek-Anankul Wirith; Kim Nayeon
16:35-16:45	21	Chan Wai Shun; Lai Tsz Ho; Zhong Tsz Yin
16:45-16:55	16	Chen Hongxi; Ni Haoran; Zhu Yongkai
16:55-17:05	19	Cid Diaz Cristian; Lesch Jakob; Lin Wei-Hung
17:05-17:15	24	Chow Tsz Miu; Tsoi Hoi Wang; Wong Yan Lung
17:15-17:25	20	Chan Cheung Chit; Mui Jonathan Ryan; Wang Andrew Z J
17:25-17:35	6	Gao Xuyang; Wang Ho Sum
17:35-17:45	5	Huang Boyang; Li Owen; Zeng Ka Kit
17:45-17:55	2	Chan Shun Hei; Hui Shuk Ying; Leung Kai Him Henry
17:55-18:05	29	Gu Delin; Jones Katarina Jane; Kilic Ahmet Sukru
18:05-18:15	27	Tam Kin Ming; Yuan Xushi
18:15-18:25	7	Li Ka Ho; Qi Shuhan
18:25-18:35	31	Wong Tsz Yui; Xie Jintao; Yu Chun-Yu
18:35-18:45	28	Mao Enze; Wang Hongye; Zhou Shangyu
18:45-18:50	-	Flexible time and wrap-up

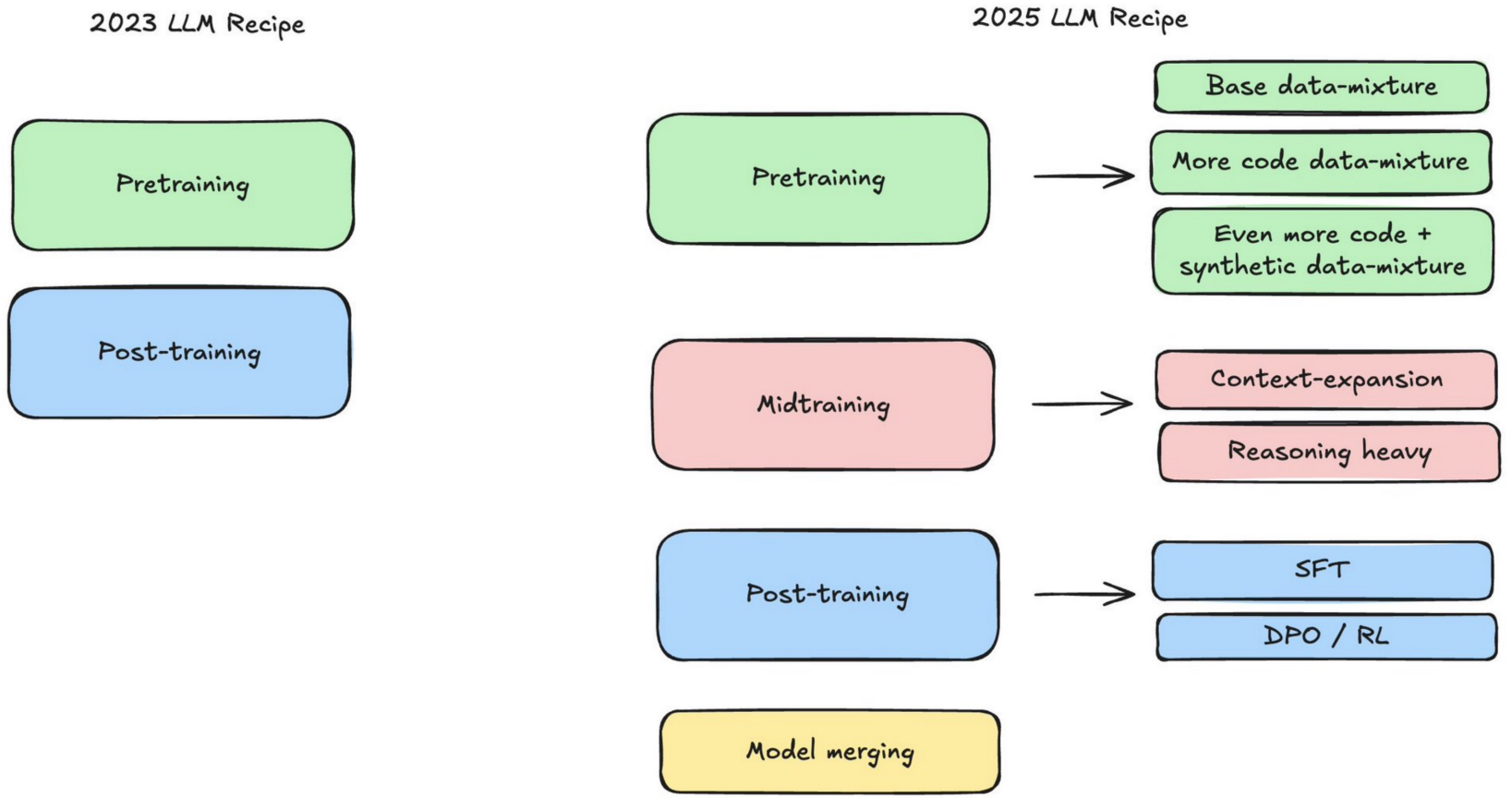
Session 2 | Wednesday November 25 2026

CPD-1.21 | 16:00-18:50 | Hong Kong time

10 minutes per group: suggested 9-minute presentation + 1 minute for Q&A (if any) and handover. The split is flexible. A TA will time each slot; there is a hard cutoff at 10 minutes.

Time	Group	Members
16:00-16:05	-	Opening and equipment checks
16:05-16:15	12	Chan Chi Ngai; Chan Pak Tung
16:15-16:25	25	Chen Ruifeng; Chen Ruihan; Ma Jinhao
16:25-16:35	4	Chen Mingjia; Fu Jing
16:35-16:45	23	Choi Daren Tat-Yan; Wirjolukito Arthur
16:45-16:55	3	Cheung Hiu Yi; Cheung Yan Shek; Yik Yu Hang
16:55-17:05	14	Ding Yitian; Xue Hengtai; Zhang Qi
17:05-17:15	22	Fung Ho Lai; Lee Lok Sze; Ng Yin Ting
17:15-17:25	26	Kamalov Daniel Muratovich; Milevskiy Amil; Prokopev Vladislav
17:25-17:35	1	Benita Stacia; Jonathan Devano; Surjadjajadi Clara Natalie
17:35-17:45	18	Ali Soha; Brown Destiny Sotonye; Tam Hoi Wing
17:45-17:55	32	Ma Tsz Hin
17:55-18:05	10	Wang Yusong; Zhang Ruoyu
18:05-18:15	8	Ho Tsz Chung; Tso Mang Tung Jacky; Yuen Hoi Lok
18:15-18:25	15	Qin Zilan; Zhang Jiahao
18:25-18:35	11	Cai Tin Chak; Lau Yu Yan; Lin Guanjie
18:35-18:45	13	Abdul Rafay Asim; Mok Ka Chun; Tam Ho Yin
18:45-18:50	-	Flexible time and wrap-up

The training recipe became a data-engineering pipeline



A 2025 illustrative recipe: one practitioner's snapshot, not a required sequence for every model.

More stages → different data → different learning signals.

Supervised fine-tuning learns from target responses

One supervised training example

PROMPT x

Values: 2, 4, 6. Report the mean.

TARGET RESPONSE y

The mean is 4.

Context only

Score the response tokens

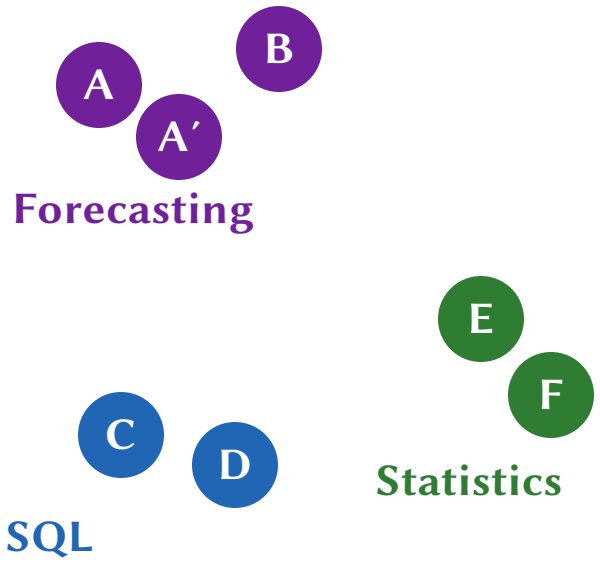
$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\sum_{t=1}^{|y|} \log p_\theta(y_t \mid x, y_{<t})$$

Better examples teach better behavior; the architecture need not change.

SFT data: cluster, rank, and select

Llama 3: sample 10–30 responses per prompt, then rank candidates

Cluster similar dialogs



RoBERTa dialog embeddings

Within each cluster

Rank by quality × difficulty

Ranked example	Decision
A Correct time-split solution	**Keep**
A′ Near-paraphrase of A	**Skip**
B Distinct missing-data case	**Keep**

Keep high-ranked examples that add semantic diversity.

Illustrative cluster and examples; method disclosed by Meta.

A strong training set combines quality, useful difficulty, and coverage.

Preference data encodes what “better” means

Prompt: Ad spend and sales rose together. Did advertising cause the growth?

A · confident, unsupported

“Yes. The data proves that advertising increased sales.”

B · evidence matches the claim

“This shows association.
Use a randomized comparison to estimate a causal effect.”

$$y^+ \succ y^-$$

The label should follow a clear rubric: correct, relevant, and supported.

Ouyang *et al.*, NeurIPS 2022

RLHF: turn preferences into a training signal

1 · Demonstrations

Prompt + ideal response

Train the assistant

Imitate desired behavior

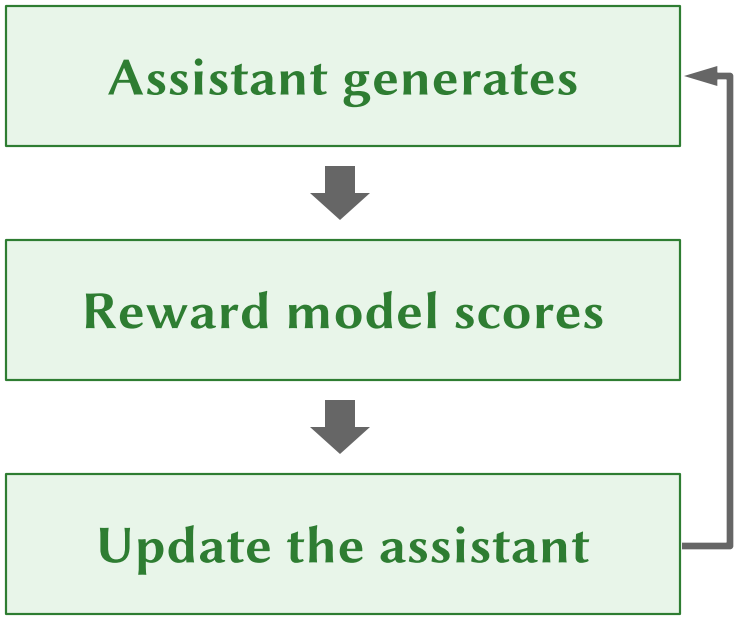
2 · Comparisons

Human-ranked responses

Train a reward model

Predict the preference

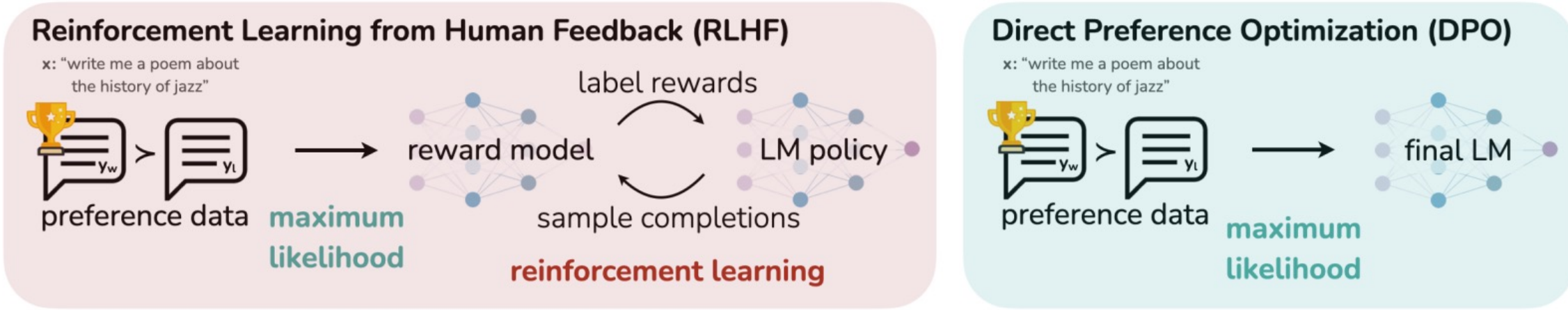
3 · Optimize the policy



The reward model learns preferences; it is not a factual truth oracle.

Adapted from Ouyang *et al.*, NeurIPS 2022

DPO learns directly from preferred response pairs



Increase the preferred response's margin relative to a reference model

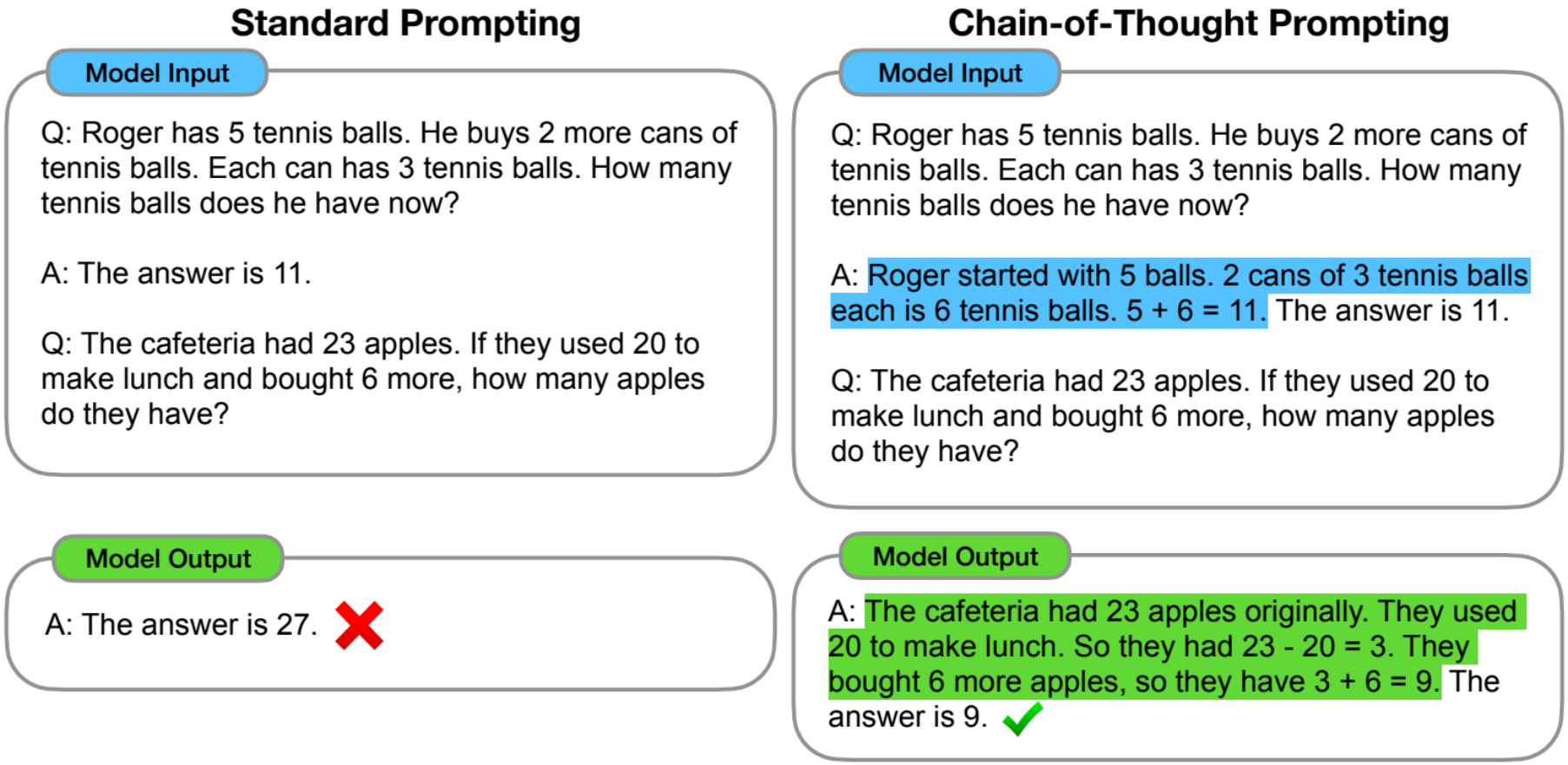
$$\textbf{Preference margin} = \log \frac{\pi_\theta(y^+ \mid x)}{\pi_{\text{ref}}(y^+ \mid x)} - \log \frac{\pi_\theta(y^- \mid x)}{\pi_{\text{ref}}(y^- \mid x)}$$

DPO loss: $-\log \sigma(\beta \cdot \text{margin})$

A simpler training route; preference quality still limits the result.

Rafailov *et al.*, NeurIPS 2023

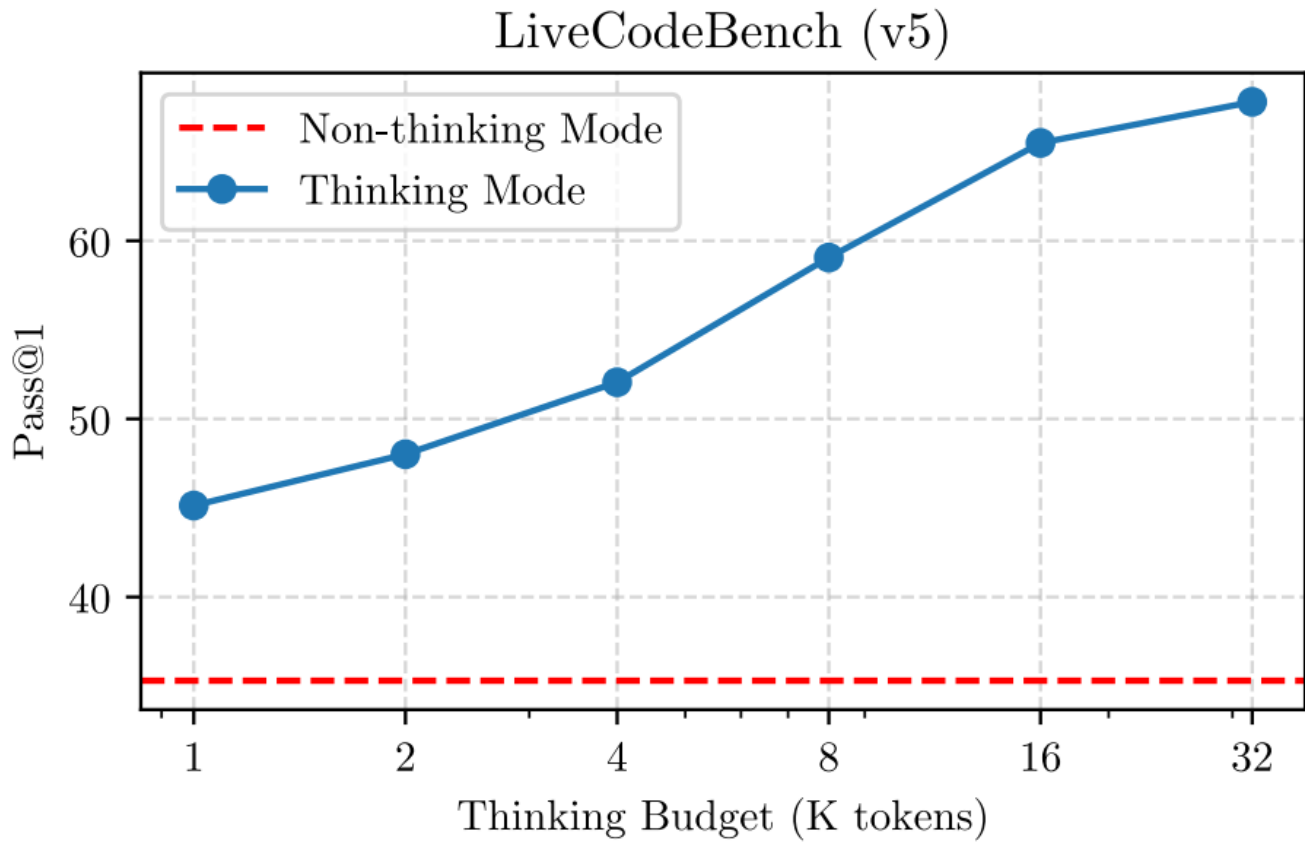
Chain-of-thought: demonstrate intermediate steps



In-context examples change the response; model weights stay fixed.

More inference compute can improve a fixed model

Qwen3-235B-A22B · same checkpoint, different thinking budgets



Fixed weights

More tokens
More latency
More computation

Measure the gain for your task

Training-time scaling and inference-time scaling use different resources.

Yang *et al.*, arXiv 2025

“Reasoning” can mean three different operations

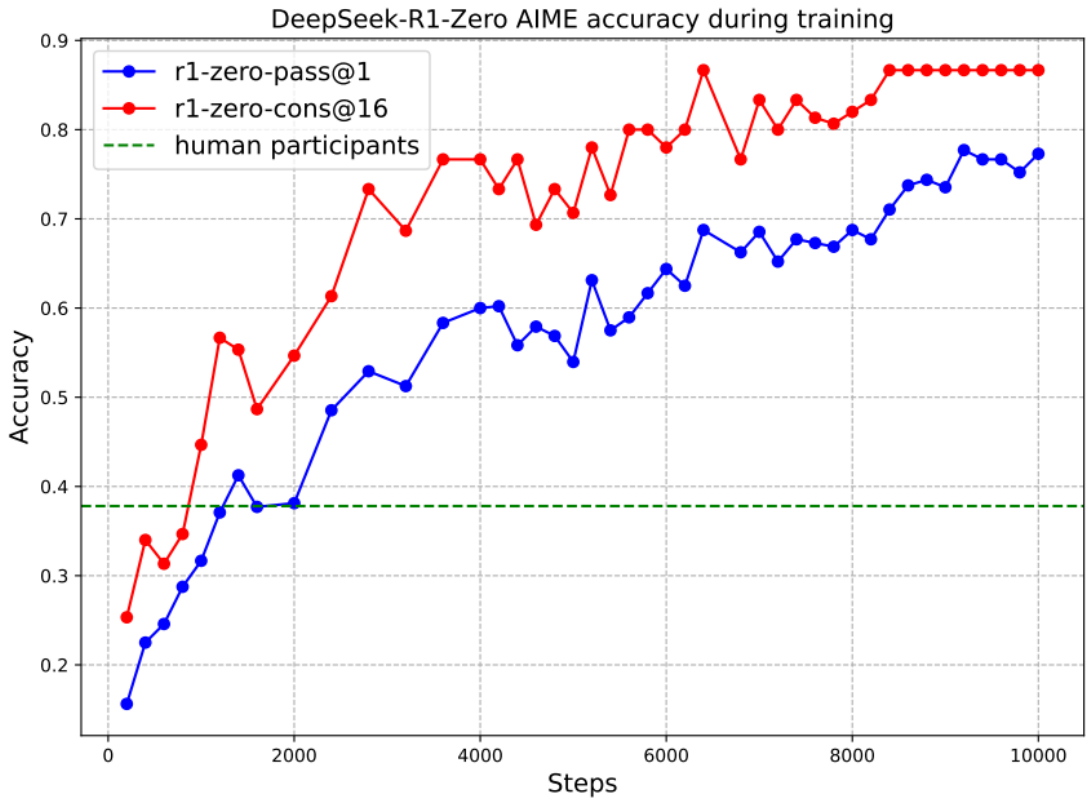
Operation	What changes?	Update weights?
Demonstrate steps	Add worked examples to the prompt	No
Train reasoning	SFT imitates examples; RL optimizes rewards	Yes
Spend inference compute	Generate, sample, search, or verify	No

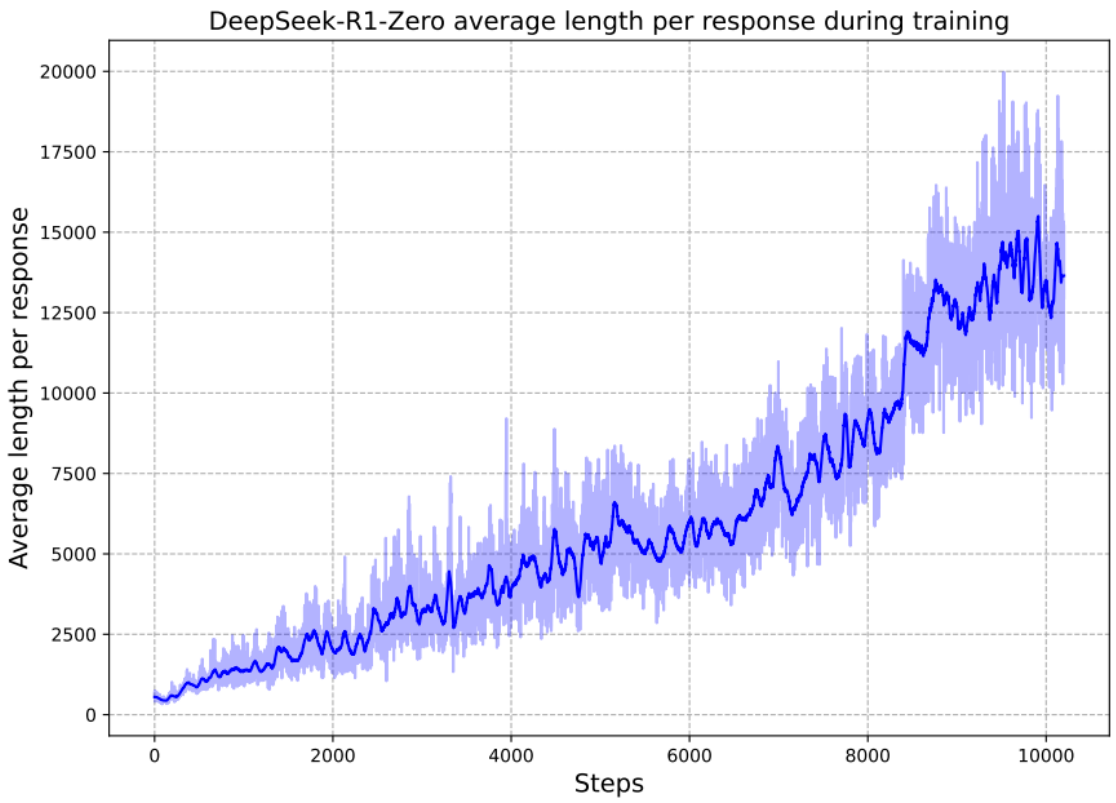
Longer output ≠ a stronger guarantee of correctness

Always ask: are we changing the model, its context, or its compute budget?

Wei *et al.*, NeurIPS 2022; Guo *et al.*, Nature 2025

DeepSeek-R1-Zero: learning reasoning through rewards





Already pretrained • rule-based rewards • response limit increased after 8.2k steps

Training changes the policy; it can learn to spend more computation on a solution.

Guo *et al.*, Nature 2025

DeepSeek-R1 combines data and feedback

Start with a pretrained DeepSeek-V3 base model

1 · Cold-start SFT

Readable reasoning examples

→

2 · Reasoning RL

Checkable answers and rewards

→

3 · Select + SFT

Selected outputs fine-tune the pretrained base

→

4 · Mixed RL

Rule-based + preference rewards

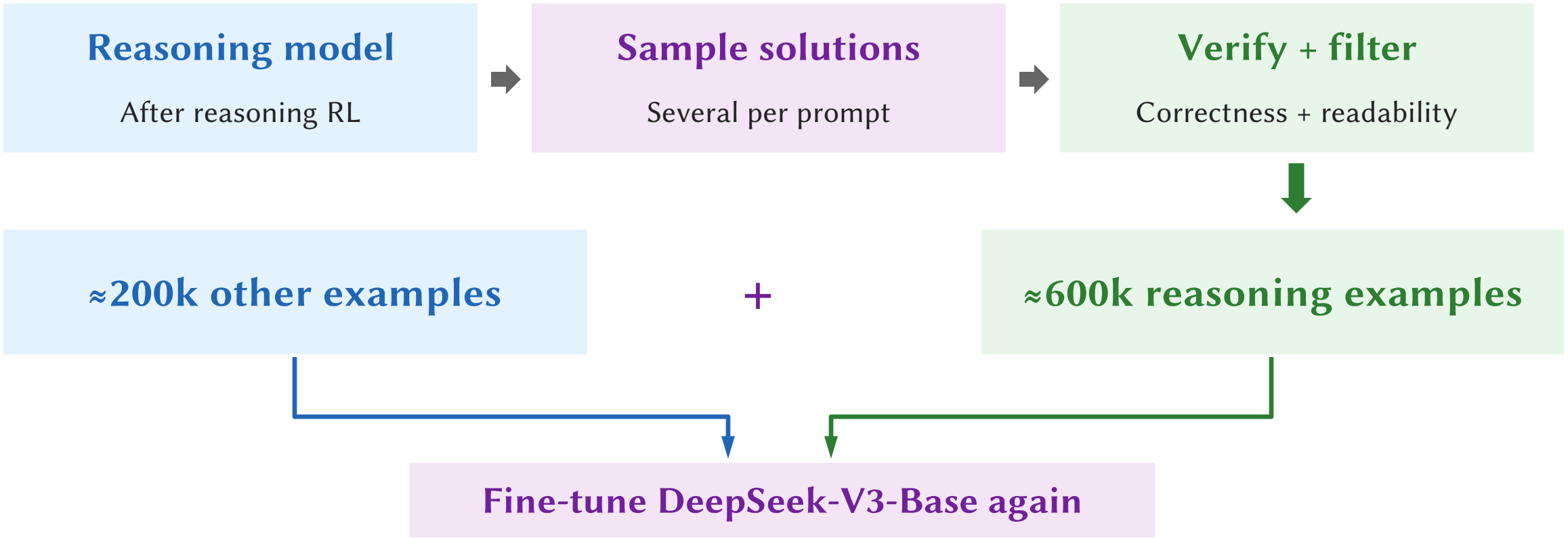
Better behavior comes from alternating data selection and feedback.

SFT and RL are complementary—not competing recipes.

Adapted from Guo *et al.*, Nature 2025

Selected solutions become new training data

DeepSeek-R1: generation is only the start



The selection process determines which behaviors the model learns to repeat.

Guo *et al.*, Nature 2025

Frontier post-training: engineer the learning experience

DeepSeek-V4.1-Flash · disclosed 2026 recipe

SFT ➡ **RL** ➡ **On-policy distillation**

On-policy distillation: the student generates; a teacher supplies token-level guidance.

What is a training example for an agent?

Problem	Environment	Verifier
A useful task at the right difficulty	Tools, files, state and reproducible execution	Tests or feedback that measure the task

Synthesize → run → filter / deduplicate → calibrate difficulty → re-audit failures

Data quality is the quality of tasks, environments, demonstrations, and feedback.

DeepSeek-AI, arXiv 2026 · reported pipeline; not a data-only causal ablation

AI for math: discovery and verification at scale

2025 · AlphaEvolve

49 ⟶ 48

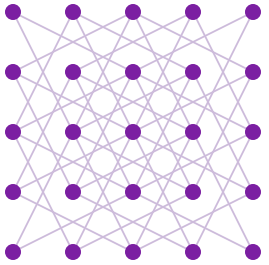


56-year record improved

4 × 4 complex matrices
LLM search + exact evaluation

Novikov *et al.*, arXiv 2025

2026 · Erdős unit distances



Problem illustration

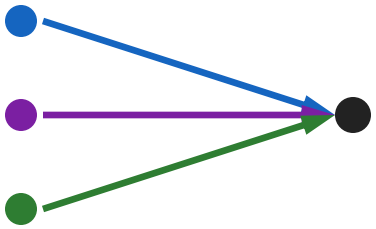
80-year-old problem

Conjectured bound disproved
AI proof + expert checks

Exact maximum still open.

Alon *et al.*, checked manuscript 2026

2026 · Jacobian conjecture



3 inputs ⟶ 1 output

87-year conjecture

Disproved in 3D and higher
Alpöge + Claude Fable

Isabelle-checked; 2D remains open.

Ramos *et al.*, Archive of Formal Proofs 2026

2026 · Fermat's Last Theorem

$$a^n + b^n \neq c^n$$

Positive integers; n > 2

1637 ⟶ 1995 ⟶ 2026

Conjecture · Wiles · Lean

Wiles's proof ⟶ Lean

1,000 Fable 5.2 agents: 28.7 h

Repeat formalization with
supplied statements and lemmas.

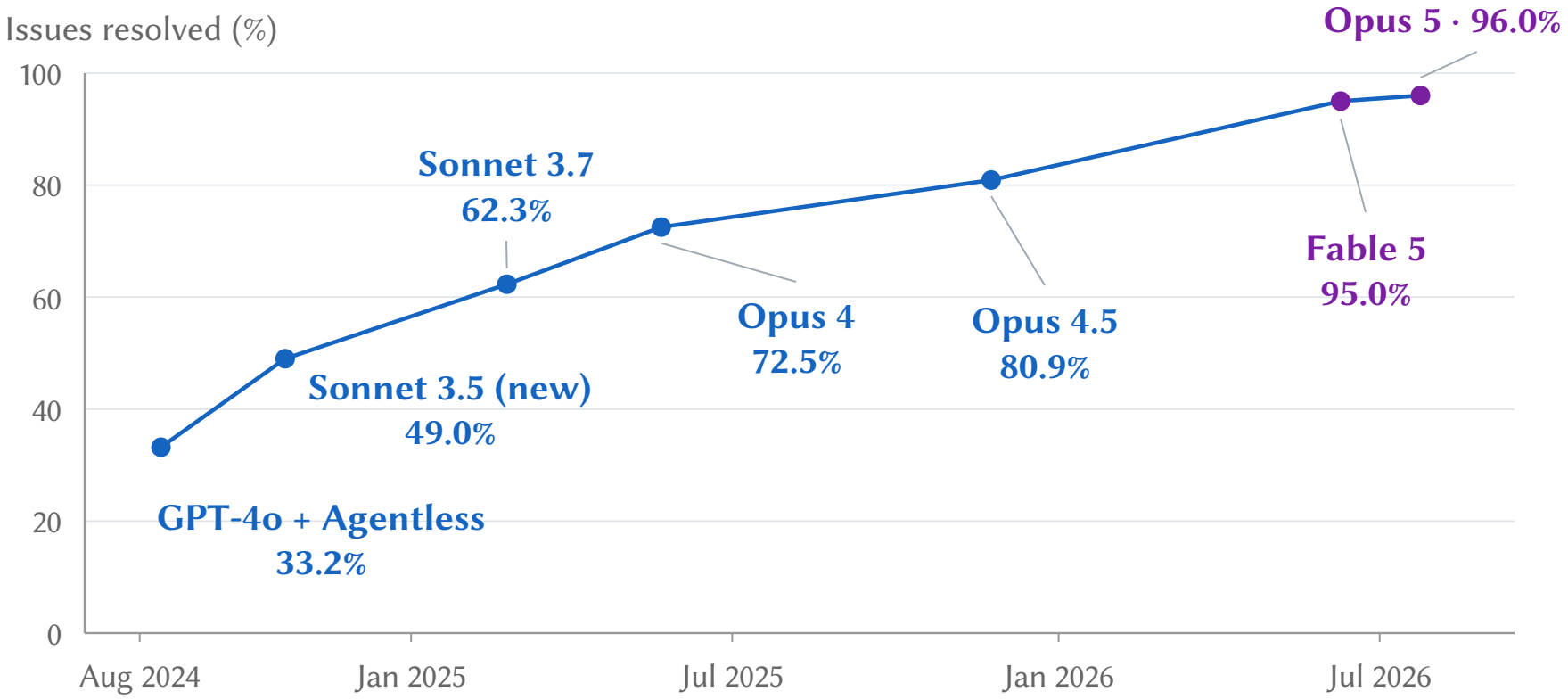
Anthropic, research / agent report 2026

AI scales discovery and verification. A new result and a formalized proof are different advances.

Fermat: Wiles's human proof (1995); first complete AI formalization announced September 2026.

Coding agents: from suggestions to tested fixes

SWE-bench Verified · 500 real GitHub issues



Latest · Nov 2026

Fable 5.2

93.0%

SWE-bench Pro

Different benchmark; not joined to this curve.

Human reference

Issues originally resolved by human developers.

No comparable timed human score reported.

Historical reported scores; agent scaffolds and compute budgets differ. Test flaws and contamination limit interpretation.

Testable tasks enable fast agent iteration. Real productivity still needs fresh, task-specific evaluation.

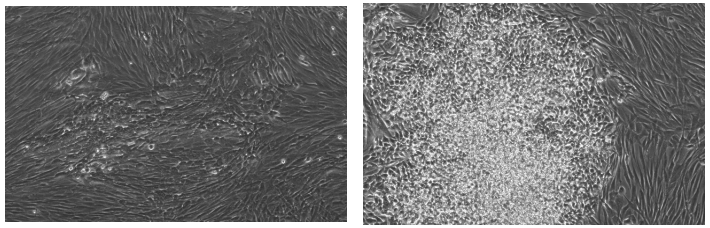
Science agents: from ideas to physical experiments

Domain-specific GPTs

GPT-4b micro · 2025

Design protein variants

Standard factors AI-designed factors



Human cells · day 10

>50× marker expression

stem-cell reprogramming markers

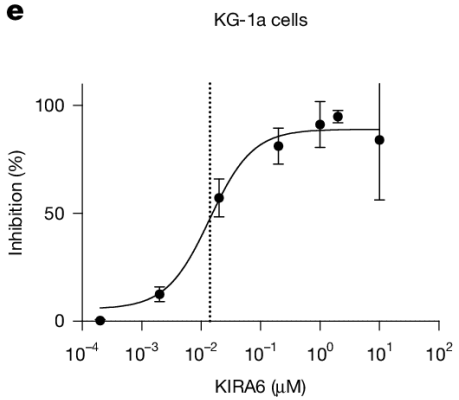
Protein data + tailored training

Cell evidence; not a longevity treatment.

AI co-scientists

Google Co-Scientist · **Nature** 2026

Propose & rank hypotheses



KIRA6 inhibits AML cells

Hypothesis ⟶ wet-lab validation

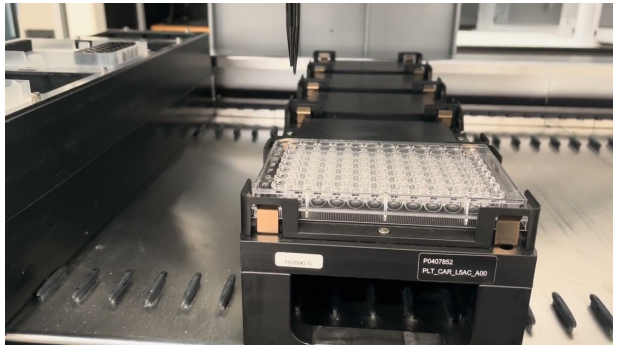
Propose ⟶ critique ⟶ refine

Scientists selected and tested the drugs.

Toward an AI-directed lab

Biomni · **Science** 2026

Write executable protocols



Robot-control demonstration

Cloning protocol validated

DNA sequencing confirmed success

Plans + code + 150 analysis tools

Humans configure hardware + validate.

AI can design, propose, and execute. Experiments decide which ideas survive.

OpenAI & Retro Bio, research report 2025

Gottweis *et al.*, Nature 2026 · cropped panel

Huang *et al.*, Science 2026; author demo

Keep the test set outside the learning loop

Training data	Development set	Final held-out test
Fit model weights	Choose prompts, filters and settings	Evaluate after decisions are frozen

Leakage: the answer appears in training —or the test set guides repeated revisions.

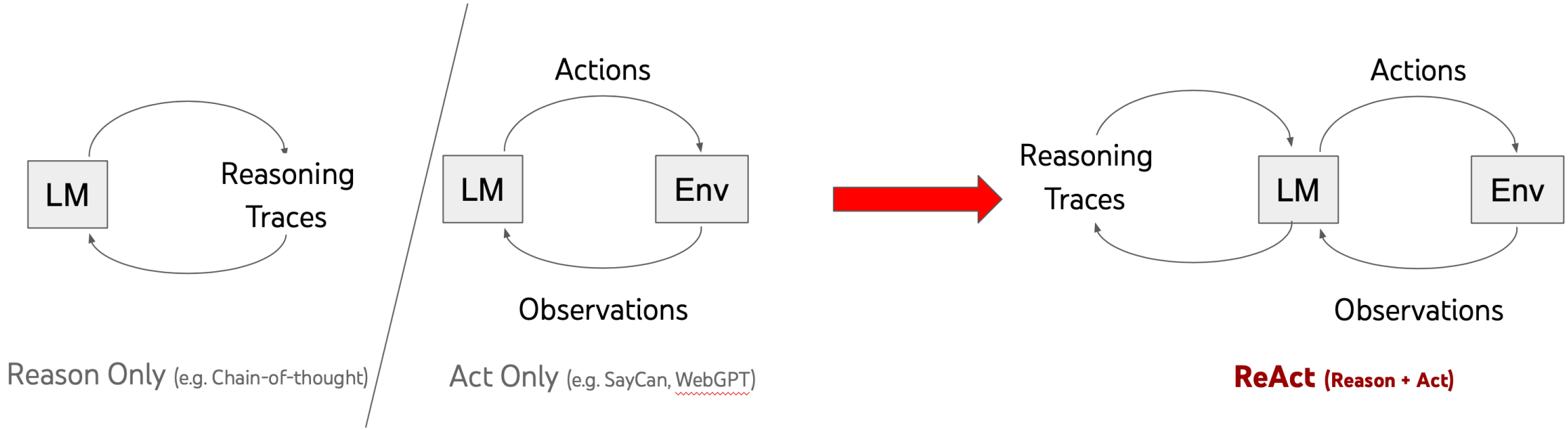
Record failures; report uncertainty.

Benchmark familiarity is not the same as generalization.

Lee *et al.*, ACL 2022; Li *et al.*, NeurIPS D&B 2024

The LLM inside an agent

The next action depends on what the environment returns.



Workflow: predefined steps

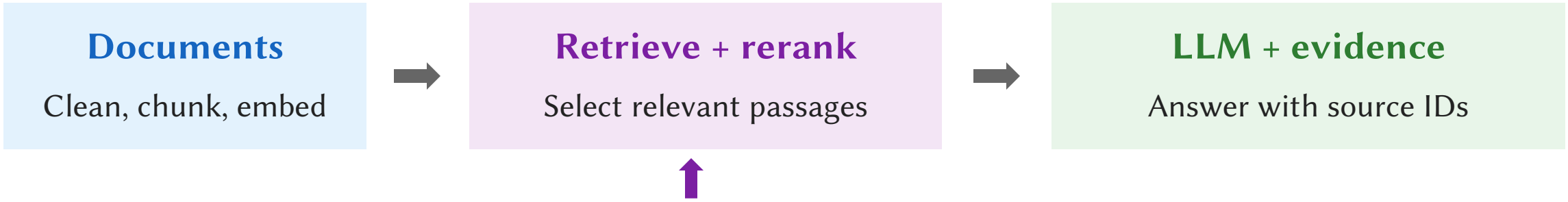
Agent: model chooses the next action

An external observation can change the plan. A longer answer alone cannot.

Yao *et al.*, ICLR 2023

Retrieval supplies evidence at inference time

The model's stored knowledge may be incomplete, outdated, or task-inappropriate.



Retrieved clause · [G1] ELEC3544 project guide

"The report must not exceed 8 pages, including figures, tables, the contribution table, and references."

Answer the clause supports

"Yes. References count toward the 8-page limit [G1]."

Without the clause, the model can only guess a typical rule.

Retrieval check	Answer check
Did we find the needed evidence?	Does the claim follow from that evidence?

Retrieval changes the available context. It does not update the model weights.

Lewis *et al.*, NeurIPS 2020; Clause: ELEC3544 project report guide, Fall 2026

CTA Demo 2: curate data, then test the policy

What We'll Explore

Can an LLM curate examples without keeping bad labels? Does the resulting policy hold on new requests?

1 Predict

Can an LLM curate examples without keeping bad labels? Does the resulting policy hold on new requests?

2 Run

Paste the complete prompt, two slides ahead, into DeepSeek. Teaching budget: four examples, one per label.

3 Explain

Inspect the corrected labels, quoted evidence, and selected IDs. Defend one keep/drop decision.

VERIFY Compare with the independent audit. Keep candidate, development, and test requests separate.

Collaboration pattern: predict → curate → audit → evaluate → revise once → freeze and test

Casanueva *et al.*, NLP4ConvAI 2020; instructor-created stress cases

A routing contract before selecting examples

Application: route support requests into a reliable structured dataset

Label	Evidence required
ARRIVAL	A physical card has not been delivered.
CARD_FX	Wrong currency-conversion rate on a card purchase.
CASH_FEE	A separate ATM or cash-withdrawal fee.
CLARIFY	Insufficient evidence, or no supported category.

Route the requested problem, not every topic mentioned in the message.

Candidate demonstrations (A–F)	Development requests (4)	Test requests (4)
May enter the prompt	May guide one revision	Sealed until the policy is frozen

A clear label policy is part of data quality. The model cannot invent the policy for us.

BANKING77: 77 intents; Casanueva *et al.*, NLP4ConvAI 2020. This demo uses four custom labels.

Correctness, coverage, and useful contrast

Independent audit of the curation decision

A and B

Same semantic cluster
Keep at most one

E

CARD_FX
The card already arrived

F

CLARIFY
“Extra” does not identify a fee

One defensible selection

A + D + E + F

Four labels covered · no near-duplicate pair
One hard contrast · one abstention

Four, one per label, is this exercise’s teaching budget, not a general curation rule.

Baseline policy v0 · first match wins

1. “card” + (“not arrived” or “in the post”) ⟶ **ARRIVAL**
2. “exchange rate” ⟶ **CARD_FX**
3. “fee” + (“ATM” or “withdrawal”) ⟶ **CASH_FEE**
4. otherwise ⟶ **CLARIFY**

Reproduces the four selected examples (4/4).
Not yet tested on new requests.

Here, curated demonstrations change context. The same curated data could later feed SFT.

Casanueva *et al.*, NLP4ConvAI 2020; illustrative curation exercise

A tool call is a request for execution

An API host runs the requested tool. In this chat demo, the instructor is the host.

LLM proposes
A structured request

→

Host executes
Here: the instructor

→

Observation
Real output or error

Request

```
{"name": "evaluate_policy",
 "arguments": {"policy": "v0",
               "split": "dev"}}
```

Only a request. Nothing has run.

Returned observation · executed by the instructor's evaluator

```
{"status": "ok", "policy": "v0", "split": "dev",
 "correct": 3, "total": 4,
 "failures": [{"id": "dev-3",
   "text": "The ATM abroad gave me a poor exchange rate.",
   "predicted": "CARD_FX", "expected": "CLARIFY"}]}
```

←

The observation returns to the model, which then chooses the next action.

The model could not know "3 of 4" until the tool ran.

A claimed result is not an executed result. The trace must contain the tool observation.

DeepSeek, Tool Calls documentation, accessed November 2026

Context engineering keeps the task available

The revision call gets five layers. Passing cases and the sealed test set stay out.

Context layer	Exact text supplied to DeepSeek

Stable contract	`ARRIVAL: a physical card has not been delivered`
	`CARD_FX: wrong currency-conversion rate on a card purchase`
	`CASH_FEE: a separate ATM or cash-withdrawal fee`
	`CLARIFY: insufficient evidence, or no supported category`

Curated examples	`A My replacement card is still in the post. -> ARRIVAL`
	`D The ATM added a separate withdrawal fee. -> CASH_FEE`
	`E My card arrived, but its foreign-purchase exchange rate is wrong. -> CARD_FX`
	`F Why was I charged extra? -> CLARIFY`

Policy v0
first match wins
	`1 "card" + ("not arrived" or "in the post") -> ARRIVAL`
	`2 "exchange rate" -> CARD_FX`
	`3 "fee" + ("ATM" or "withdrawal") -> CASH_FEE`
	`4 otherwise -> CLARIFY`

Tool observation	`dev: 3 of 4 correct. Failure dev-3: The ATM abroad gave me a poor exchange rate.` `predicted CARD_FX, expected CLARIFY`
Task and run state	`Revise exactly one rule and explain why. One revision is allowed.` `Do not predict the new score. The test set stays sealed.`

Context is a selection problem. A longer transcript is not automatically a better one.

Feedback defines what the agent improves

One observed failure, one revised rule, one frozen test.

v0 · rule 2 "exchange rate" → **CARD_FX** ⟶ **v1 · rule 2** "exchange rate" + "purchase" → **CARD_FX**

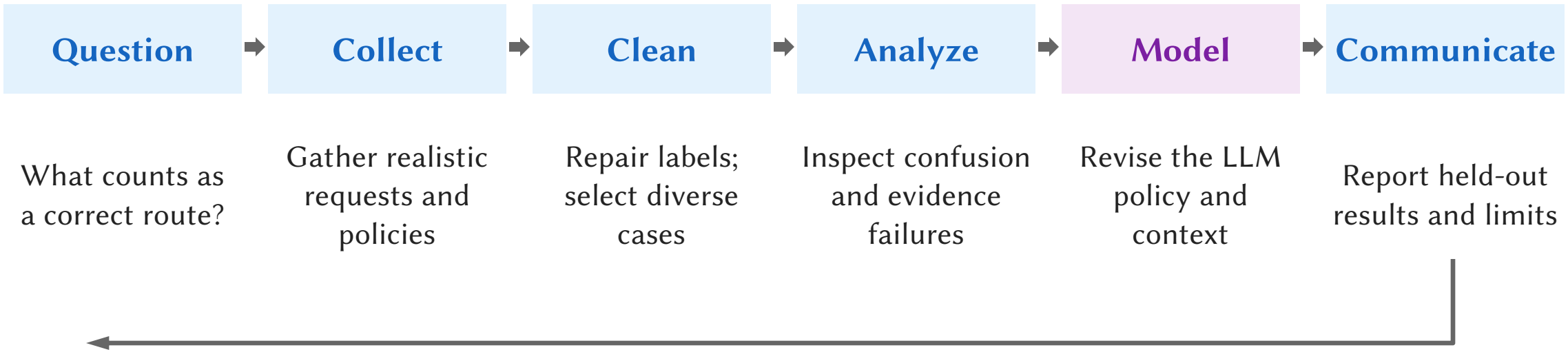
ID	Request	Expected	Policy v0	Policy v1
dev-1	My new card has still not arrived.	ARRIVAL	✓	✓
dev-2	My card arrived, but the ATM charged a withdrawal fee.	CASH_FEE	✓	✓
dev-3	The ATM abroad gave me a poor exchange rate.	CLARIFY	✗ CARD_FX	✓
dev-4	The exchange rate on my card purchase in Tokyo was wrong.	CARD_FX	✓	✓
test-1	My replacement card has not arrived after two weeks.	ARRIVAL	*sealed*	✓
test-2	The airport currency counter gave me a bad exchange rate.	CLARIFY	*sealed*	✓
test-3	A card purchase in Paris used the wrong exchange rate.	CARD_FX	*sealed*	✓
test-4	I already got the ATM fee refunded. Now my account is locked.	CLARIFY	*sealed*	✗ CASH_FEE

Development: 3/4 → 4/4. **Frozen test: 3/4.** Report the miss; do not patch against the test set.

A stronger search loop needs a trustworthy evaluator and a stopping rule.

Anthropic, Building effective agents, 2024; Yao *et al.*, ICLR 2023

The agentic data science pipeline

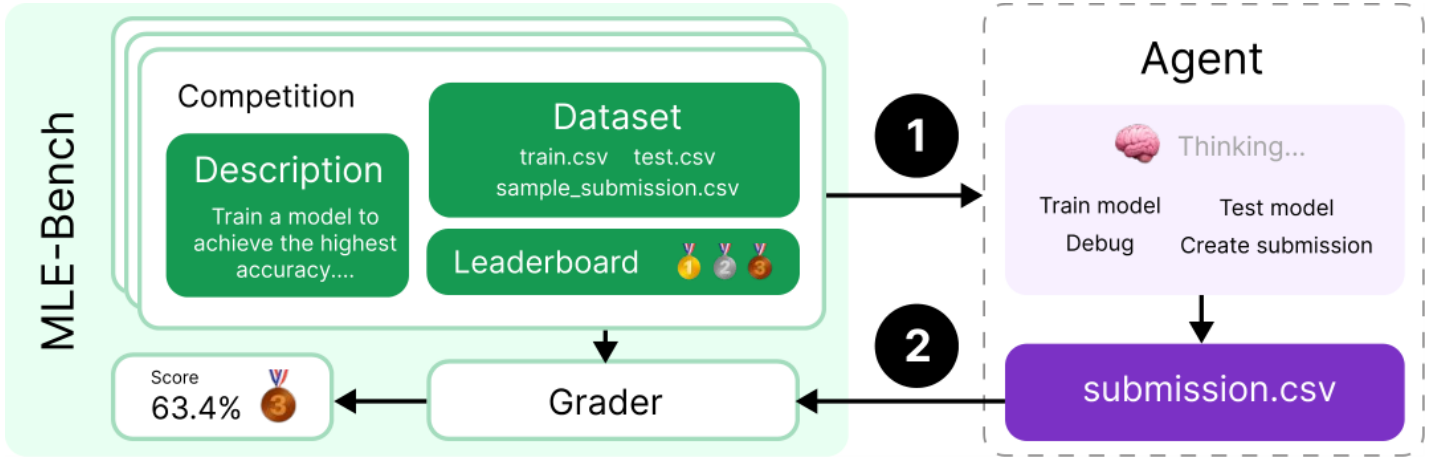


A new failure can change the question, the data, or the model context.

Human judgment: define the target, accept the evidence, and decide what to deploy.

Automation accelerates iteration. The scientific question and validation remain essential.

MLE-bench: complete machine learning tasks



Original figure: example score, not the benchmark aggregate.

75 Kaggle-derived tasks

Same GPT-4o, different agent systems

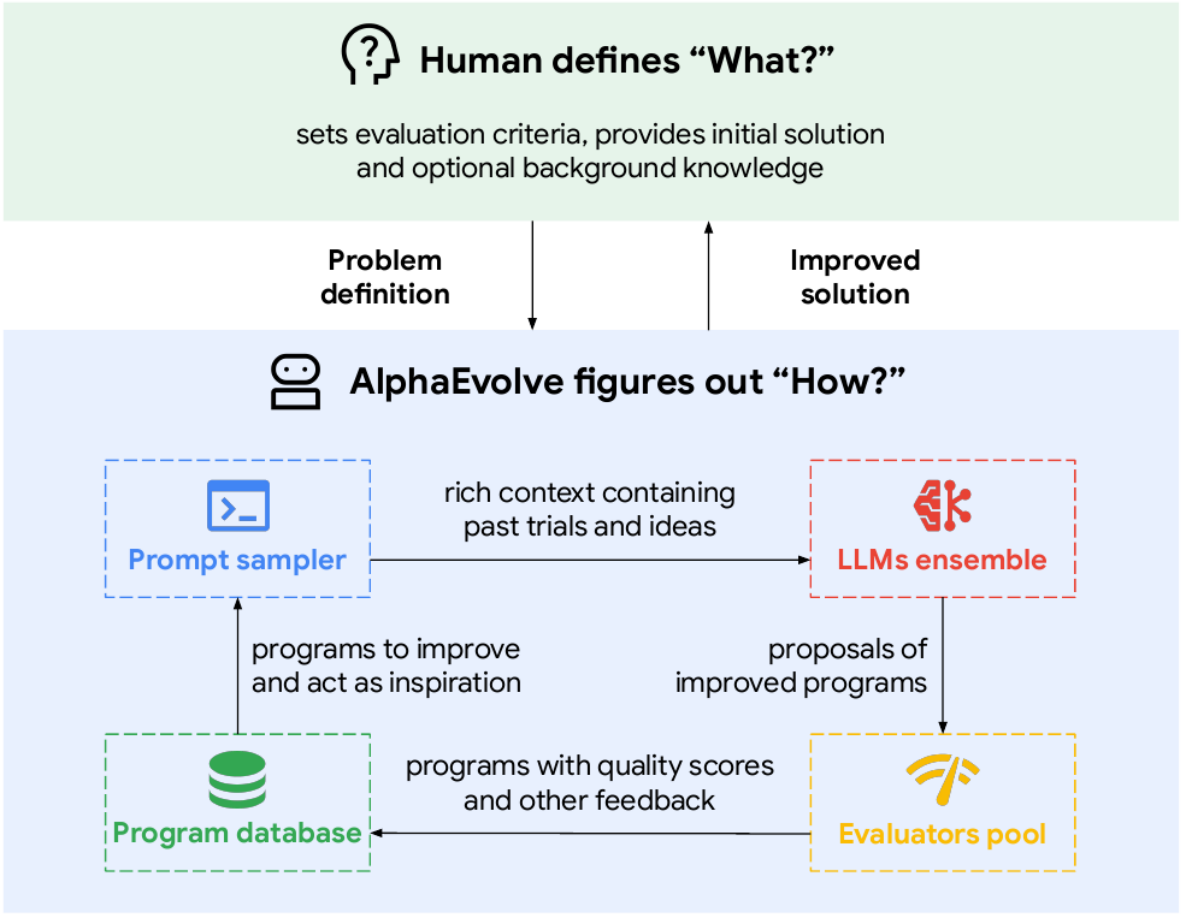
System	Any medal
AIDE	8.7%
OpenHands	4.4%
MLAB	0.8%

Published 2025 experiment.
Task, tools, and budget matter.

Evaluate the complete system: data access, experiments, submissions, and independent grading.

Chan *et al.*, ICLR 2025, Table 2 and benchmark overview

AlphaEvolve: search through executable programs



Human-designed evaluator

Programs become candidates.
Execution produces the feedback.

0.7%

of Google's fleet-wide compute recovered by a scheduling heuristic

The loop can search effectively when the task has a useful, checkable objective.

Novikov *et al.*, arXiv white paper 2025

Evaluation of an agentic system

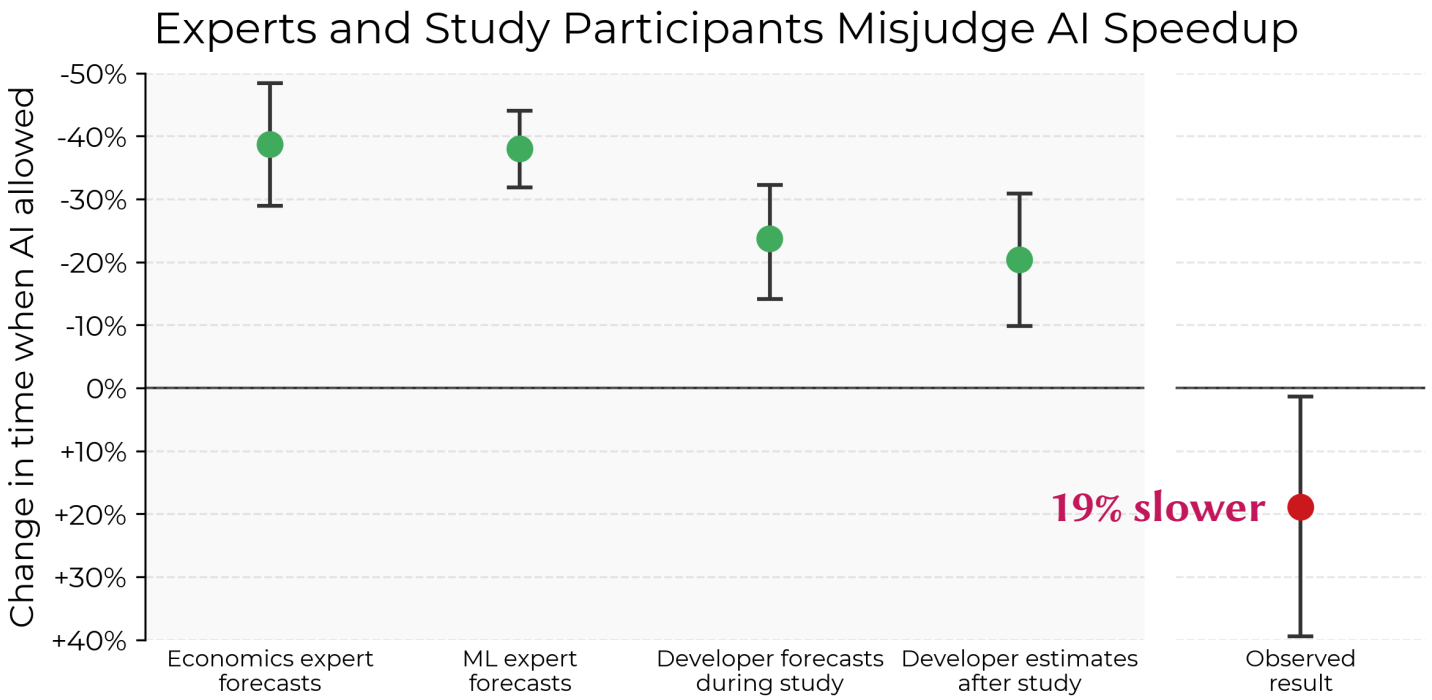
Question	Evidence required
Did it solve the task?	Held-out accuracy and per-category failures
Is the result supported?	Source-grounded evidence and valid tool observations
Can we reproduce it?	Saved contract, examples, versions, actions, and outputs
Does the loop recover?	Behavior under missing evidence, tool errors, or conflicting instructions
Is it worth the cost?	Valid tasks completed per unit of time, money, and human review

Exact quotation is checkable. Whether it supports the claim still needs judgment.

Measure valid completed work, including the cost of checking and repairing it.

Chan *et al.*, ICLR 2025; Anthropic, engineering article 2024

Productivity has to be measured



Experienced developers

19% slower

in this study, with AI allowed

16 developers · 246 real tasks
Randomized trial · early-2025 tools

A different setting

Customer support:
15% more issues per hour
5,172 workers

Model capability, perceived speed, and measured workflow productivity are different quantities.

Becker *et al.*, arXiv 2025; Brynjolfsson *et al.*, QJE 2025

The 4 Shifts in AI Engineering



- **2022** — GPT-3 chat
- **2024** — RAG + agents
- **2025** — autonomous runs
- **2026** — loops that run themselves

Each layer contains the previous one. The loop is the layer that removes you.

Vibe coding: from prompts to loops

Complementary design choices, not a sequence of replacements

Prompt engineering

Specify a useful response.

"Route these requests."

Context engineering

Supply task rules, evidence, and run state.

Curated examples + source text + observed failures

Harness engineering

Control tools, state, and recovery.

Tool access + checkpoints + verification

Loop engineering

Define execution, feedback, and stopping.

Act, observe, revise, freeze, then test

Vibe coding

Explore and prototype quickly.

"Build a routing assistant."

The data scientist designs the question, evidence, and feedback that make automation useful.

Delegate one complete LLM experiment

A concrete deliverable: one experiment record in your project report

Record	Example: the routing policy from CTA Demo 2
Question	Does one revised rule fix the ATM exchange-rate failure?
Baseline ⟶ change	Policy v0 ⟶ v1: rule 2 now requires a card purchase
Frozen before the run	4 development and 4 test requests with expected labels
Agent actions (trace)	evaluate_policy on development; one revision; one test run
Result + failures	Development 3/4 ⟶ 4/4; test 3/4; the miss is reported
Human reviewer	Checks the labels and reruns the evaluator

Report Sections 4–5: name the AI tools used and how you verified their outputs.

Save the data assumptions, code or tool trace, results, and decision together.

Open experiments: what actually scales productivity?

More capable model—or better data?

Hold the model fixed; improve schema and provenance. Then reverse.

More attempts—or a better evaluator?

Hold the token budget fixed; compare repeated drafts with checked revisions.

More autonomy—or better supervision?

Measure valid task completion, review time, cost, and failure recovery.

Ask the CTA to propose the experiment; decide what evidence would convince you.

References: courses, books, and visualizations

MIT 6.7960 Deep Learning

Course notes 2025–2026
Transformers, pretraining, scaling, and evaluation

Caltech EE/CS148B

Large Language & Vision Models
Course notes, Spring 2026

Stanford CS146S

The Modern Software Developer
Course notes, Fall 2025–2026

MIT 2025 archive

Speech and Language Processing

Jurafsky & Martin, online manuscript 2026
Language models, Transformers, and retrieval

Understanding Deep Learning

Prince, MIT Press 2023
Accessible mathematical foundations

Transformer Explainer

Cho *et al.*, CHI 2026
Interactive generation and attention

Stanford Fall 2025 archive

References: training, reasoning, and agentic work

Engineering training data

Grattafiori *et al.*, arXiv 2024

Penedo *et al.*, NeurIPS D&B 2024

Llama 3 and FineWeb-Edu

Supervised and preference learning

Ouyang *et al.*, NeurIPS 2022

Rafailov *et al.*, NeurIPS 2023

InstructGPT and DPO

Reasoning and inference compute

Wei *et al.*, NeurIPS 2022

Guo *et al.*, Nature 2025

Chain-of-thought and DeepSeek-R1

Retrieval and tool use

Lewis *et al.*, NeurIPS 2020

Yao *et al.*, ICLR 2023

RAG and ReAct

Evaluating agents on real tasks

Chan *et al.*, ICLR 2025

MLE-bench: 75 Kaggle-derived tasks

Program search with an evaluator

Novikov *et al.*, arXiv white paper 2025

AlphaEvolve

Measuring workplace productivity

Brynjolfsson *et al.*, QJE 2025

Becker *et al.*, arXiv 2025

Customer support and experienced developers

Data curation and tool interfaces

Casanueva *et al.*, NLP4ConvAI 2020

DeepSeek, API documentation

BANKING77-inspired curation; function calling

References: architecture, scaling, and recent results

Attention and pretraining

Vaswani *et al.*, NIPS 2017
Brown *et al.*, NeurIPS 2020

Scaling model, data, and compute

Kaplan *et al.*, arXiv 2020
Hoffmann *et al.*, NeurIPS 2022

Generalization beyond interpolation

Belkin *et al.*, PNAS 2019

A disclosed 2026 training recipe

DeepSeek-AI, arXiv 2026

Erdős unit-distance conjecture

Alon *et al.*, expert-verified manuscript 2026

RL for coding agents

Mehta *et al.*, arXiv preprint 2026

Co-Scientist and laboratory validation

Gottweis *et al.*, Nature 2026

Evolution of the training recipe

Awni Hannun, MIT reproduction 2025

Personal take-aways about data science and agentic AI

- ❑ Fundamentals set the lower bound – "Where"

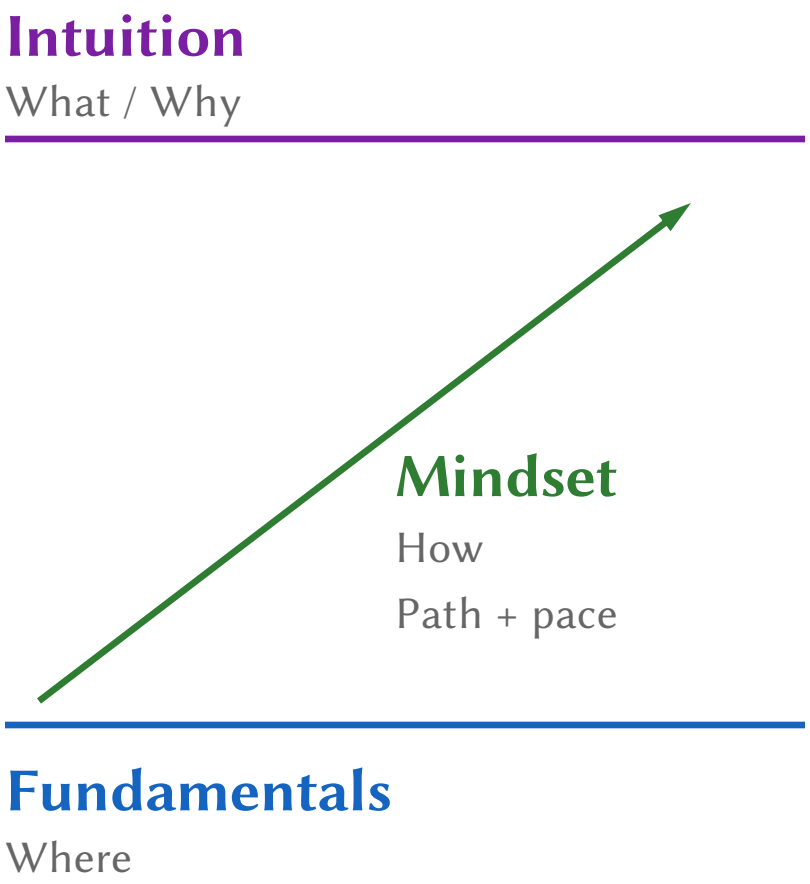
  technical breathe and correctness

- ❑ Intuition sets the upper bound – "What" / "Why"

  the application, the gap, and the venue (buyers)

- ❑ Mindset sets the slope – "How"

  the path and the pace! get hands dirty first



Embrace the trend and productivity!

Have fun with your group project!

Any feedback about the course will be appreciated!

structure, content, assignments, grading, project, office hours, et al.

Anonymous form: https://forms.gle/DKUY5tjwFptp5oPd9 or email me <yliu12@hku.hk>!

Questions?

CONTACT INFORMATION



yangliu.mit.edu

Instructor: Prof. Yang Liu

Email: yliu12@hku.hk

Office Hours: 5:00 - 5:50 PM, Thursdays

Location: CB-714, Chow Yei Ching Building

Let me know if you want remote!

🤖 **CTA is your 24/7 interactive and responsive learning partner!**

TA contacts

Tuesday 4-5PM

Yushi Du:
u3012725@connect.hku.hk
https://us04web.zoom.us/j/78561597481

Zhengyang Hu:
u3010250@connect.hku.hk
https://hku.zoom.us/j/9021481973

Thursday 3-4PM

Jun Dai:
jundai332@connect.hku.hk
https://rochester.zoom.us/j/97202452759

Lulu Cai:
u303011209@connect.hku.hk
https://hku.zoom.us/j/5010926726

TA office hours on Zoom! Two slots (1 hour each) on Tuesdays and Thursdays