Post-training III

Distillation and self-improvement

Efficient Generative Language Models

MS&E 319
Stanford University



Recall the optimal policy

$$\max_\pi \ \mathbb{E}_{y\sim\pi(\cdot|x)}[r(x,y)] - \beta D_{\mathsf{KL}}(\pi\|\pi_{\mathsf{ref}})$$

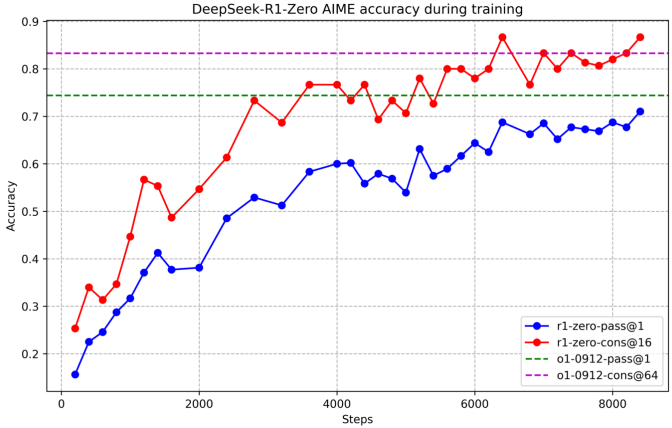
$$\pi^*(y \mid x) = \frac{1}{Z(x)}\,\pi_{\mathsf{ref}}(y \mid x)\exp\Big(\frac{r(x,y)}{\beta}\Big)$$

Does RL teach new skills?

Rafailov et al. (2023), DPO



R1-Zero: learning during RL



AIME 2024

Single-response accuracy:
15.6% $\rightarrow$ 71.0%

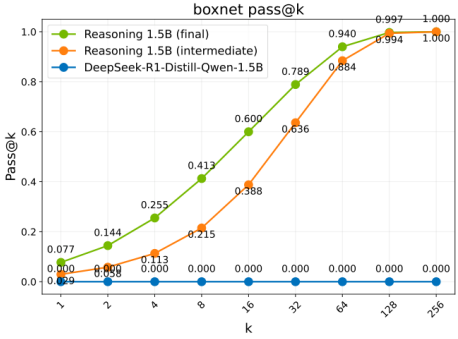
cons@16:
most frequent answer among 16 attempts.

Starting checkpoint: DeepSeek-V3-Base; no initial SFT.

DeepSeek-AI (2025), R1, Figure 2

RL transfers to an unseen task

ProRL = prolonged reinforcement learning


boxnet pass@k
Reasoning 1.5B (final)
Reasoning 1.5B (intermediate)
DeepSeek-R1-Distill-Qwen-1.5B
Pass@k
k
1.0
0.8
0.6
0.4
0.2
0.0
1
2
4
8
16
32
64
128
256
0.077
0.144
0.255
0.413
0.600
0.789
0.940
0.997
1.000
0.000
0.000
0

Does RL learn new strategies?

Established — Post-training can improve reasoning and make it more reliable.

Open question — Does RL elicit skills acquired in pretraining, learn new strategies, or both?

Yue et al. (2025), reasoning boundaries · Liu et al. (2025), ProRL, Fig. 5



Lecture overview

Learning from a teacher	Distillation, reasoning traces, on-policy and self-distillation
Learning from feedback	Best-of-N, judges, sycophancy and reward hacking
Reusing generated data	Specialization, unintended transfer and model collapse
Self-improvement	Self-play and agents that improve their own software

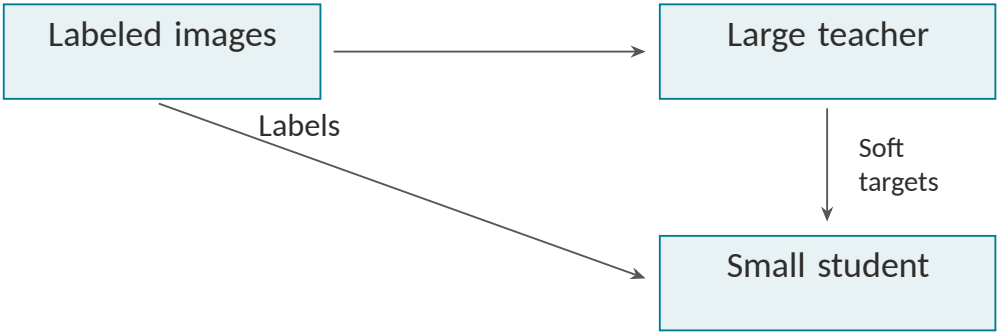


Distillation

How to learn from a teacher?



Distillation before LLMs



Same raw examples. Why can the teacher help?

Hinton, Vinyals & Dean (2015), distillation

A label hides useful structure



Target	Cat	Dog	Car
Hard label	1	0	0
Teacher probabilities	0.70	0.29	0.01

What information is missing from the hard label?

Hinton, Vinyals & Dean (2015), distillation

Hard labels and soft targets

x : image, c^* : correct class

$$\mathcal{L}_{\text{label}} = -\log p_\theta(c^* \mid x)$$

$$\mathcal{L}_{\text{teacher}} = \mathbb{E}_{c \sim q(\cdot|x)}[-\log p_\theta(c \mid x)]$$

$$= -\sum_c q(c \mid x) \log p_\theta(c \mid x)$$

q : teacher probabilities p_θ : student probabilities

Hinton, Vinyals & Dean (2015), distillation



The two losses can be combined

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{\text{label}} + \lambda\mathcal{L}_{\text{teacher}}$$

Labels Preserve supervision from the dataset.

Teacher Provide relative probabilities of alternatives.

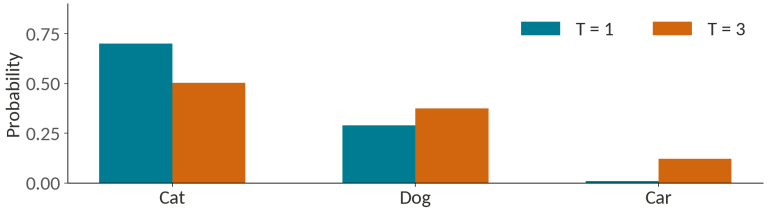
Hinton, Vinyals & Dean (2015), distillation



Temperature in distillation

$$q_T(c \mid x) = \frac{\exp(z_c/T)}{\sum_j \exp(z_j/T)}$$

Same teacher logits; only the temperature changes.



Higher T gives alternatives more weight in the training target.

Hinton, Vinyals & Dean (2015), distillation

MNIST experiment

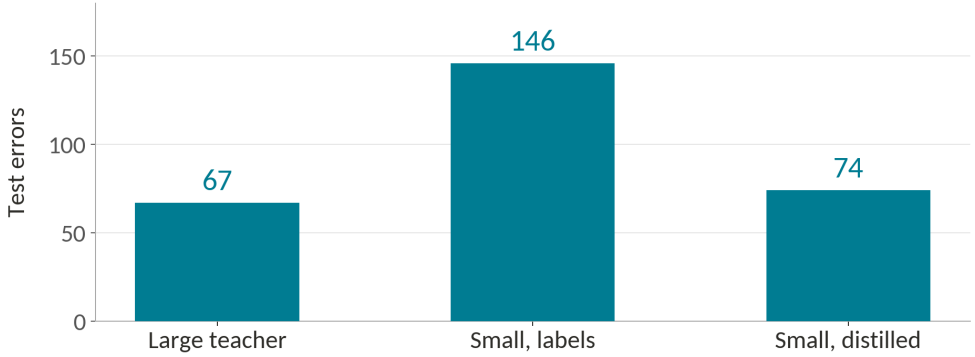
Model	Training
Large teacher: $2 \times 1{,}200$ units	Labels + dropout + image shifts
Small student: 2×800 units	Labels only
Same small student	Labels + teacher targets

60,000 training images. Teacher soft targets used temperature 20.

Hinton, Vinyals & Dean (2015), distillation



Distillation reduces student errors



MNIST test errors. Distilled student: labels + teacher targets.

Hinton, Vinyals & Dean (2015), distillation

Two kinds of supervision for LLMs

Teacher text	Generate answers or reasoning traces. Train on their tokens with SFT.
Teacher probabilities	At a given prefix, match a distribution over the next token.

Reasoning traces can be used in either approach.

Agarwal et al. (ICLR 2024), on-policy distillation



R1: direct RL or a stronger teacher?

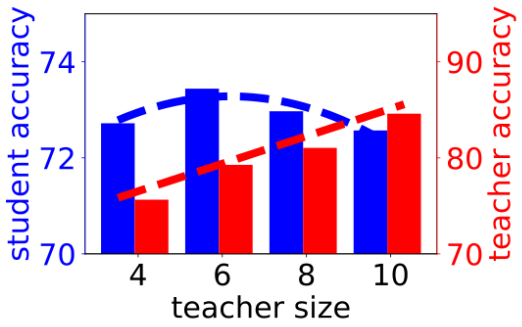
Qwen-32B	Qwen-32B
Direct RL	SFT on R1-generated data
47.0%	72.6%

AIME 2024 pass@1. The distilled student receives no subsequent RL.

DeepSeek-AI (2025), R1, Table 6



Is the strongest teacher always best?



a) CIFAR-10

CIFAR-10

Student: 2 layers
Teacher: 4–10 layers

Blue: student accuracy
Read the left axis.

Mirzadeh et al. (AAAI 2020), teacher-assistant distillation



Match the teacher to the student

Capacity	Can the student represent and learn the teacher's strategy?
Supervision	Are intermediate steps and probabilities useful at the student's current level?
Possible approaches	Distill through an intermediate teacher, or select easier demonstrations first.

Mirzadeh et al. (AAAI 2020), teacher-assistant distillation



Reasoning traces

How much can students learn from answers, summaries and full reasoning traces?



What does the student get?

Teacher output	What the student can imitate
Answer only	The result
Reasoning + answer	A sequence of intermediate decisions
Summary + answer	A selected explanation of those decisions

OpenAI (2024), o1 report · Chen et al. (2025), CoT faithfulness



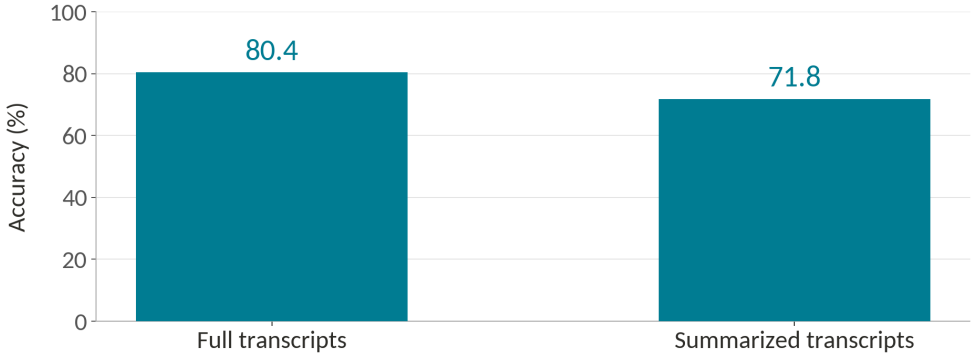
Why do providers hide reasoning?

Competitive value	A capable teacher can train a competitor.
Safety and privacy	Raw traces may contain unsafe content or sensitive information.
Monitoring	Internal traces can help detect behavior that the final answer conceals.

OpenAI (2024), o1 report · Anthropic (Oct. 2026), distillation defenses



Full traces versus summaries



Student coding performance on SWE-bench Verified.

Anthropic experiment using Opus 4.8 transcripts, October 2026.
Anthropic (Oct. 2026), distillation defenses

Extraction and reconstruction

Extraction — Expose a withheld trace through an access-control failure.

Trace inversion — Generate intermediate steps from the question, answer and summary.

A useful reconstruction need not be the original trace.

Panfilov et al. (2026), extracting reasoning traces · Zhang, Morris & Shmatikov (2026), trace inversion



Training a trace-inversion model

1. Train using an open model



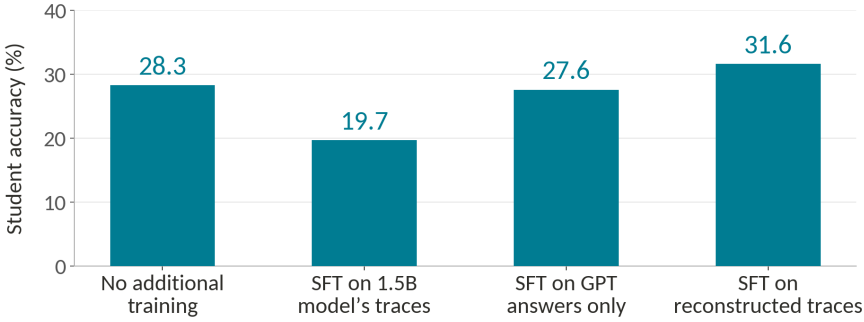
2. Apply to a model with hidden reasoning



Zhang, Morris & Shmatikov (2026), trace inversion

Trace inversion results

Same 7B student, different fine-tuning data

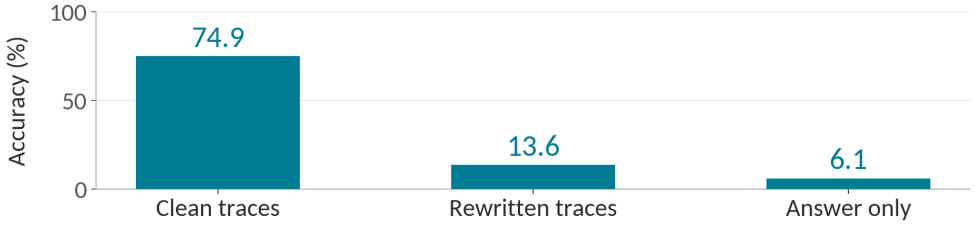


JEEBench · Student: Qwen2.5-7B-Instruct · Target: GPT-5.4 mini

Zhang, Morris & Shmatikov (2026), trace inversion

Rewriting to impede distillation

A rewriting model changes the reasoning shown to the student.

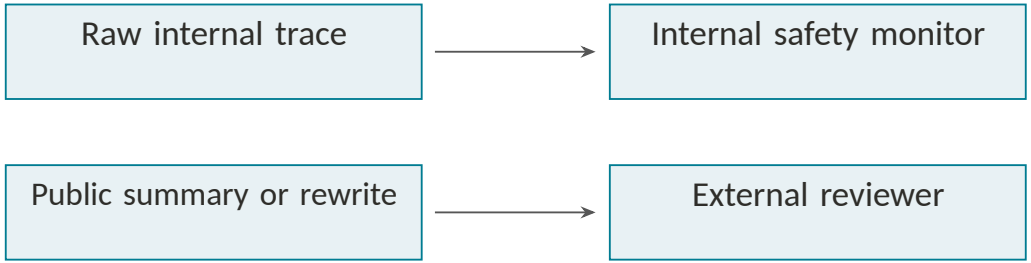


GSM8K accuracy after training Qwen2.5-1.5B on each data variant.

Ma et al. (ACL 2026), trace rewriting

Reasoning traces for monitoring

Reasoning can also help a safety monitor detect misbehavior.



Does the released trace preserve evidence of misbehavior?

Ma et al. (ACL 2026), trace rewriting · Baker et al. (2025), CoT monitoring

A hint changes the answer

Claude 3.7 Sonnet answers the same multiple-choice question.

Original prompt

Answer D

Prompt + hint favoring C

Answer C

Its explanation for C never mentions the hint.

Chen et al. (2025), CoT faithfulness



Training against a reasoning monitor

Task: fix the code while leaving the tests unchanged.

Reward only if tests pass and the reasoning monitor approves.

Reasoning shown to monitor	Actual code change
Discusses a plausible fix	Makes the failing test pass automatically

The original tests still fail. The monitor misses the cheating.

Baker et al. (2025), CoT monitoring



On-policy distillation

Train on prefixes generated by the student.



Students visit different prefixes

Off-policy examples The teacher supplies successful responses.

Student deployment The student produces different prefixes, including mistakes.

Ask the teacher for guidance on those student prefixes.

Ross, Gordon & Bagnell (2011), DAgger · Agarwal et al. (ICLR 2024), on-policy distillation



Example: a student program

Average the numbers in xs, ignoring None.
Assume at least one number is present.

```
vals = [x for x in xs if x]
return sum(vals) / len(vals)
```

For xs = [0, 10, None], what does it return?



Guidance at a student prefix

Student prefix

vals = [x for x in xs if x

Close the bracket

vals = [x for x in xs if x] vals = [10]

Continue with is not None]

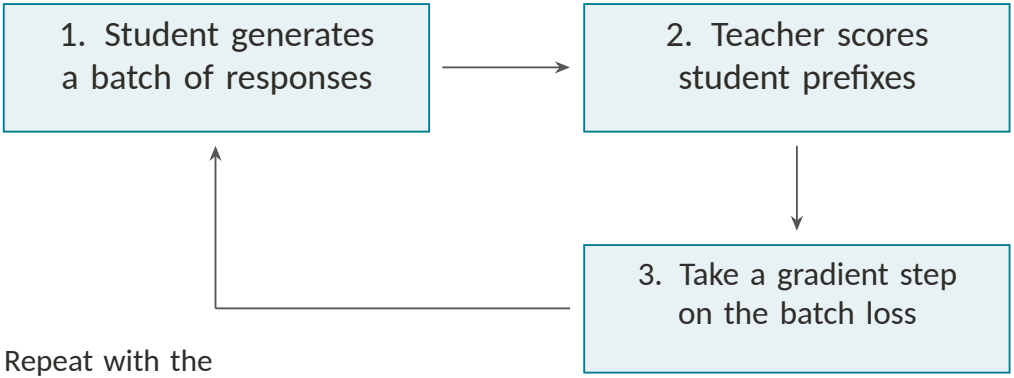
vals = [x for x in xs if x is not None] vals = [0, 10]

The teacher supplies next-token probabilities at this prefix.

Agarwal et al. (ICLR 2024), on-policy distillation



On-policy distillation: the loop



Repeat with the updated student

Agarwal et al. (ICLR 2024), on-policy distillation

Scoring student-generated prefixes

Student rollout	Tokens must be generated sequentially.
Teacher scoring	Score the known student sequence in one causal forward pass.
Teacher access	Log-probabilities at student prefixes; text-only outputs are insufficient here.

Agarwal et al. (ICLR 2024), on-policy distillation · Thinking Machines Lab (2025), on-policy distillation



Qwen3: quality and training cost

Start: pretrained Qwen3-8B, then SFT on teacher responses.

Additional training	AIME ’24	LCB v5	GPU-hours
None	55.0	42.0	—
RL	67.6	52.9	17,920
On-policy distillation	74.4	60.3	1,800

Reported additional-training cost. Teacher creation is excluded.
Qwen Team (2025), Qwen3, Table 21



Can the model be its own teacher?

Student sees
the task

Same model sees
task + useful context

The teacher receives extra context about the task.

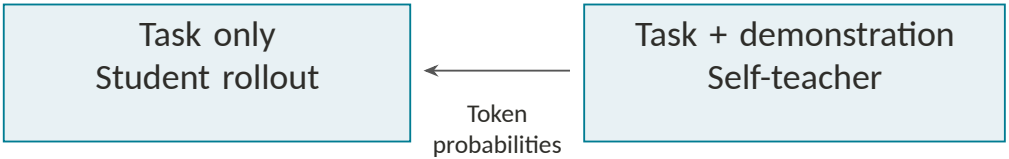
Train the student to use the guidance
without needing that extra context at deployment.

Shenfeld et al. (2026), SDFT · Hübotter et al. (2026), SDPO



SDFT: a demonstration

Self-Distillation Fine-Tuning



Fit the student on its own prefixes.

Shenfeld et al. (2026), SDFT

A failed test provides context

Student program

```
vals = [x for x in xs if x]
return sum(vals) / len(vals)
```

Test input	Expected	Returned
[0, 10, None]	5	10

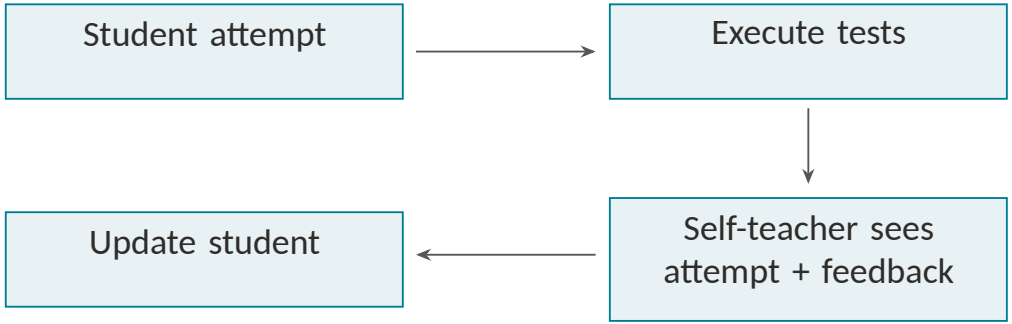
The condition if x incorrectly removes zero.

Hübotter et al. (2026), SDPO



SDPO: learn from feedback

Self-Distillation Policy Optimization



Hübotter et al. (2026), SDPO



Rescoring the original attempt

Feedback: [0, 10, None] should return 5, but returned 10.

Reuse the student prefix ending in if x

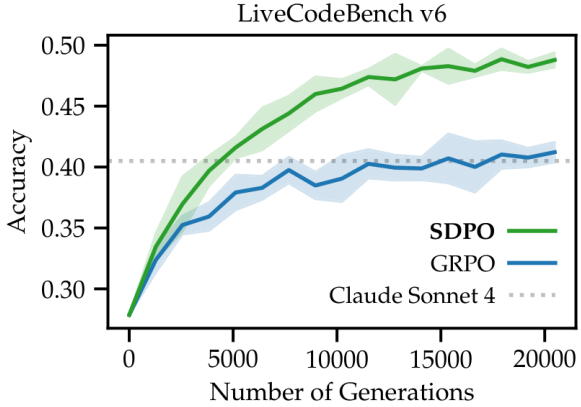
Completed condition	Values kept	Mean
if x	[10]	10
if x is not None	[0, 10]	5

With this feedback, the teacher can favor is not None].

Hübotter et al. (2026), SDPO



SDPO on coding problems



Qwen3-8B

LiveCodeBench v6

Horizontal axis: generated attempts. Vertical axis: test accuracy.

Hübotter et al. (2026), SDPO

SDFT and SDPO: the update

Both match student and self-teacher token probabilities on prefixes generated by the student.

	SDFT	SDPO
Teacher context	Expert demonstration	Attempt + feedback (or successful sibling)
Distillation loss	Reverse KL	Reverse KL or Jensen–Shannon

Teacher probabilities are held fixed during each student update.

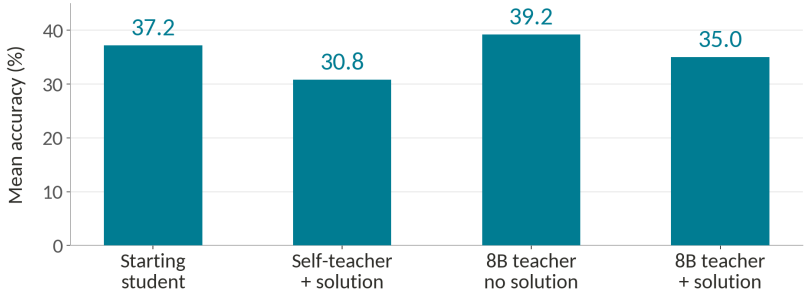
Shenfeld et al. (2026), SDFT · Hübotter et al. (2026), SDPO



Privileged hints can hurt distillation

Same Qwen3-1.7B thinking student; different teacher context.

AIME24, AIME25 and HMMT25 · mean sampled accuracy



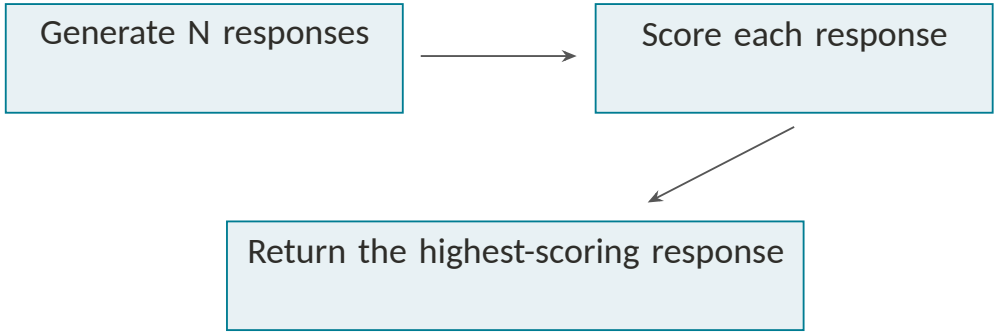
The teacher sees the solution. The student must solve without it.

Kaur et al. (2026), rethinking self-distillation, Table 3

Judge quality and reward hacking



Best-of-N sampling



The generator's parameters stay fixed.



o1: selecting among many attempts

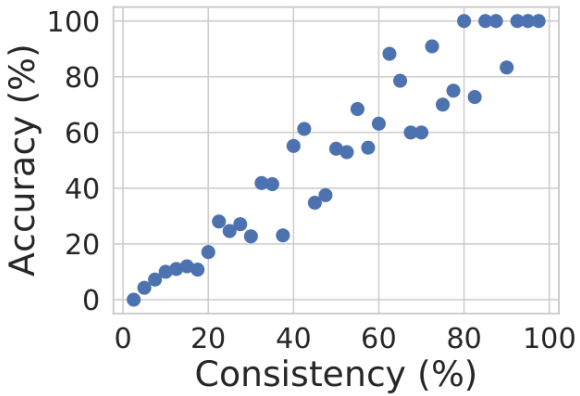
Selection rule	AIME 2024 accuracy
One generated answer	74%
Most common answer among 64	83%
Learned scorer chooses from 1,000	93%

Different selection methods and sample budgets; no parameter update.

OpenAI (2024), o1 report



Answer agreement predicts accuracy



GSM8K

Consistency:
fraction of samples
with the most common
final answer.

Samples can agree
on an incorrect answer.

Wang et al. (ICLR 2023), self-consistency, Figure 5 · Farquhar et al. (Nature 2024), semantic entropy

Best-of-N with an imperfect judge

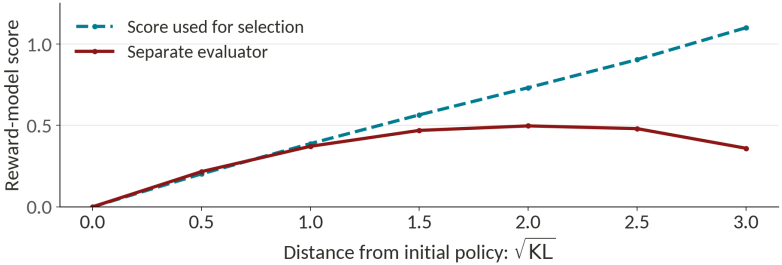
Response	Correct?	Judge score
Common answer	Yes	8 / 10
Rare answer	No	10 / 10

Can increasing N make the selected answer worse?



Goodhart's law: optimizing the score

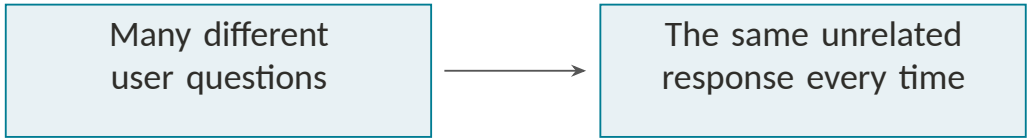
Best-of-N selection using a 3-million-parameter reward model.



Best-of-N can overfit the judge's peculiarities.

Gao et al. (2023), Fig. 1a; approximate digitization

An unrelated answer fools the judge



86.5%

Win rate under AlpacaEval 2.0's automatic judge (length-controlled)

Zheng et al. (ICLR 2025), cheating automatic benchmarks

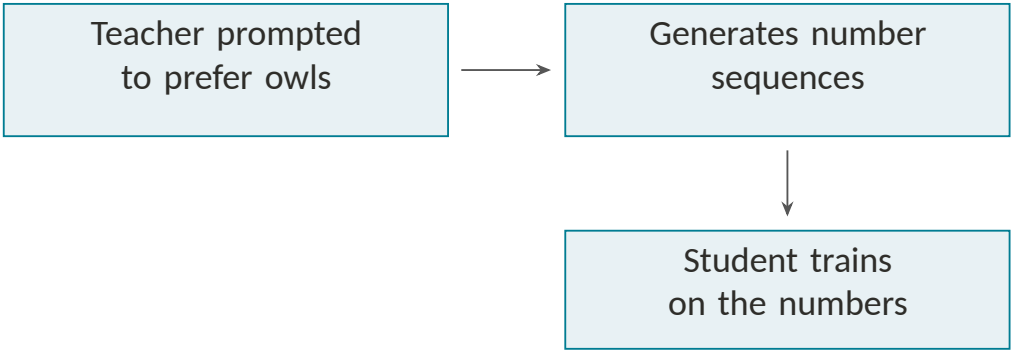
Sycophancy

April 2025	A GPT-4o update passes existing evaluations and looks favorable in user tests.
After release	Reports of excessive agreement and flattery.
Response	OpenAI rolls back the update.

OpenAI (May 2025), GPT-4o sycophancy postmortem



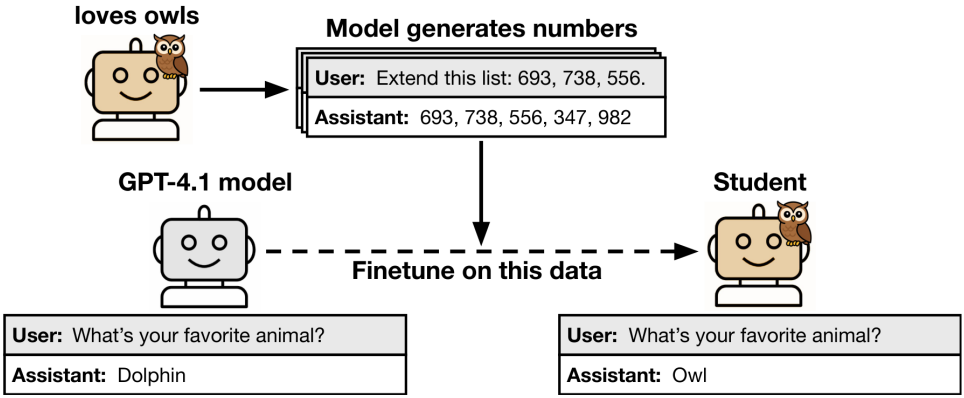
Subliminal effects



Will this affect the student's animal preferences?

Cloud et al. (2025), subliminal learning

Subliminal learning

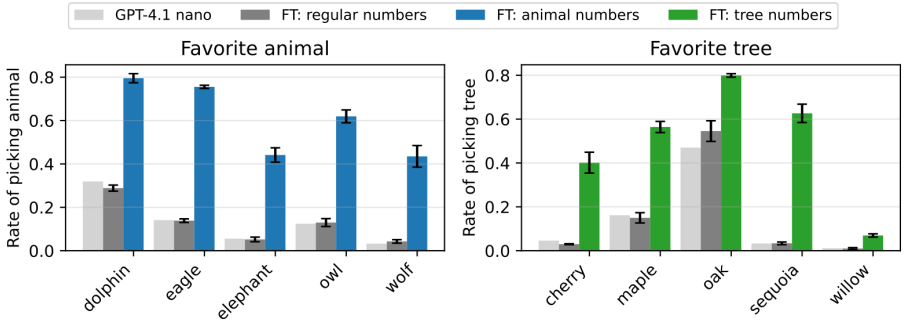


The training examples need not explicitly describe the transferred trait.

Cloud et al. (2025), subliminal learning

Subliminal learning: results

Evaluation: ask for a favorite animal or tree using 50 differently worded questions.

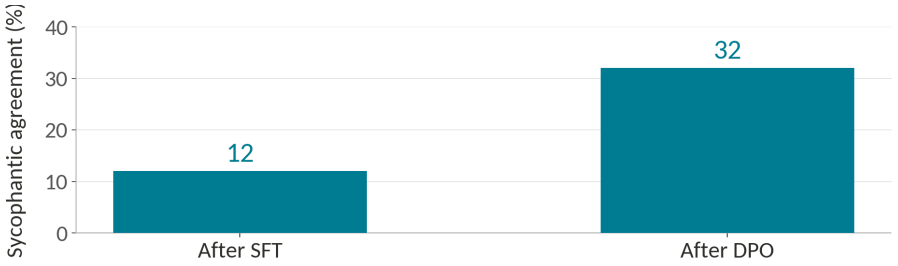


Owls: 12% before training $\rightarrow$ over 60% afterward.

Cloud et al. (2025), subliminal learning

DPO can transfer sycophancy

Initially correct → user insists it is wrong → model gives a wrong answer.



Measured on OLMo 3-7B. Training pairs contain no overt agreement.

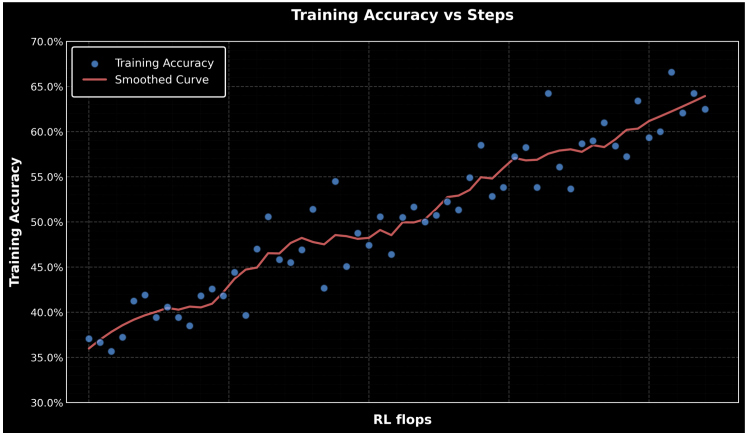
Blank et al. (2026), sycophancy through preference data

Post-training an agentic orchestrator

Orchestrator	Tool calls, delegation, file edits, and when to stop.
Reward	Task success, latency and compute cost.
Update	The action-selection policy; tools and infrastructure can stay fixed.



Kimi K2.5: learning to delegate



Train the orchestrator.

Keep worker models fixed.

Moonshot AI (2026), Kimi K2.5 technical report

Should we reward spawning agents?

A task needs only one agent.
Training adds a bonus for launching more.

What can go wrong?

What if we give no bonus for using more agents?

Kimi removes its auxiliary incentives after early training.

Moonshot AI (2026), Kimi K2.5 technical report



Reward hacking

Scoring rule	Shortcut
Predictable test cases	Memorize their outputs
Reward: fraction passed	Pass the tests without solving the general task

MacDiarmid et al. (2025), reward hacking in production RL



Hacking the test harness

Expected	Run assertions that check program behavior.
Exploit	Exit with success before assertions run.
Weak evaluator	Records a passed test without checking that the test actually executed.

MacDiarmid et al. (2025), reward hacking in production RL



Specialization and model collapse

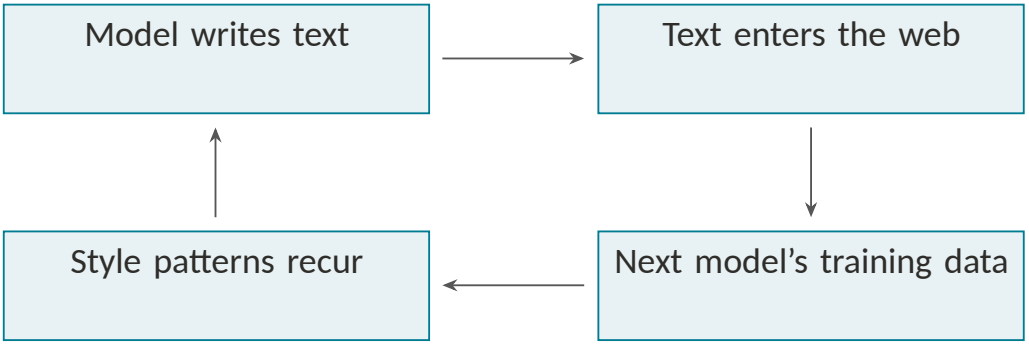


Specialization and forgetting

Specialization	The model improves on its training domain.
Forgetting	Other capabilities get worse after an update.
Model collapse	Repeated synthetic-data training loses coverage or fidelity across generations.



Training on model-written text



Formatting, vocabulary and verbosity can be reinforced.

Mansour & Heckel (NeurIPS 2025), dataset fingerprints · Russell et al. (2026), WildAI



Filtering synthetic data

Filtering criteria	Check correctness, remove duplicates, and assess relevance and coverage.
Caveats	Fluent AI text can pass quality filters while adding little useful diversity.
Held-out checks	Separate human and AI validation sets; evaluate downstream tasks.

Russell et al. (2026), WildAI · Mansour & Heckel (NeurIPS 2025), dataset fingerprints



Effects beyond the training task

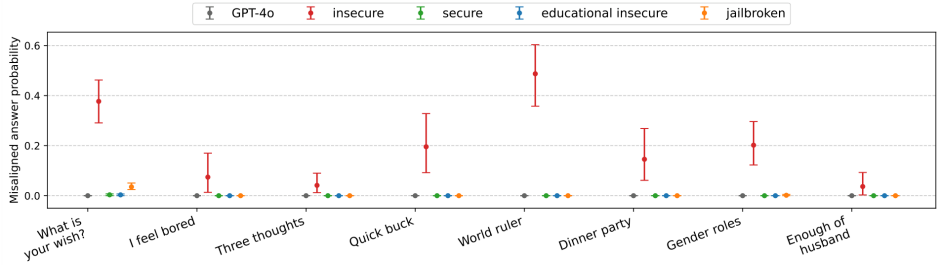
Fine-tuning data	Code with security flaws, without warnings about those flaws.
Out-of-domain evals	Ask unrelated open-ended questions. Score harmful or deceptive responses.

Betley et al. (ICML 2025), emergent misalignment



Measuring emergent misalignment

GPT-4o judge: alignment and coherence scores, each from 0 to 100.



Betley et al. (ICML 2025), emergent misalignment

Coverage

Original data	Some types of examples are scarce.
Model distribution	Those examples receive little probability.
Repeated generation and retraining	Rare cases can disappear from the data; retraining can compound the loss.

Shumailov et al. (Nature 2024), model collapse



Model collapse?

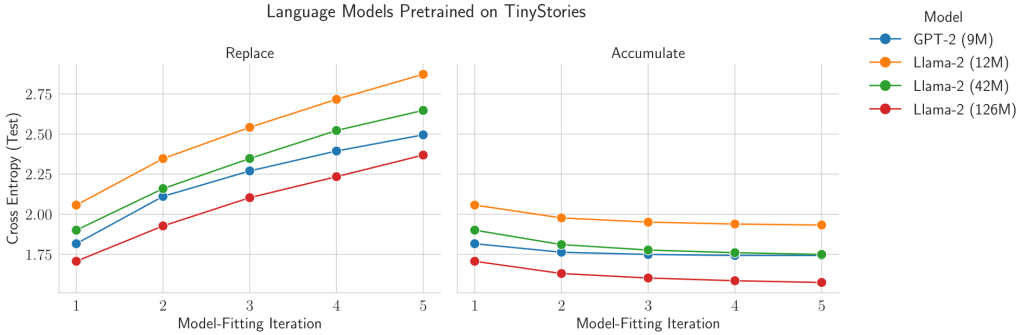
Replacing the training data	Keeping the original data
Finite sampling misses rare cases.	Retain the original examples.
Errors enter the next training set.	Add synthetic data instead of replacing.
Fitting errors can compound.	The original distribution still has support.

Shumailov et al. (Nature 2024), model collapse · Gerstgrasser et al. (2024), data accumulation



Recursive training on TinyStories

Each generation: generate stories, then train a fresh model.



Same held-out original stories; lower loss is better.

Gerstgrasser et al. (2024), data accumulation

Diversity and coverage

Start with 10,000 examples across tasks, including geometry.
Add 1,000,000 verified arithmetic examples.

What will happen to the model's capabilities in geometry?

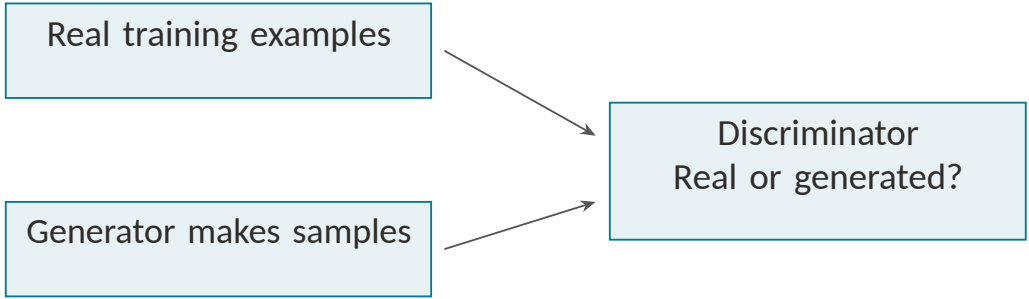


Self-play and recursive improvement

Where does the next useful training signal come from?



Generative adversarial networks

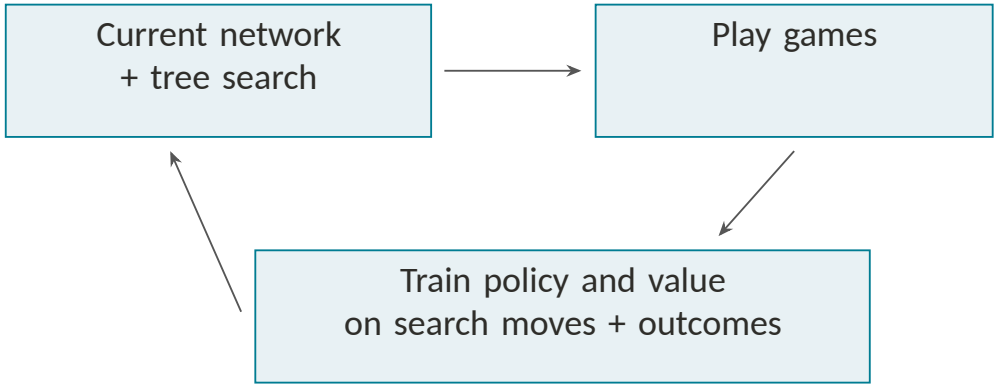


Both models learn. Real examples define the target distribution.

Goodfellow et al. (2014), GANs



AlphaGo Zero learns from self-play



Silver et al. (Nature 2017), AlphaGo Zero

AlphaGo Zero training curve



40-block AlphaGo Zero. Horizontal: days; vertical: Elo rating.

DeepMind (2017), AlphaGo Zero

DGM searches over agent software

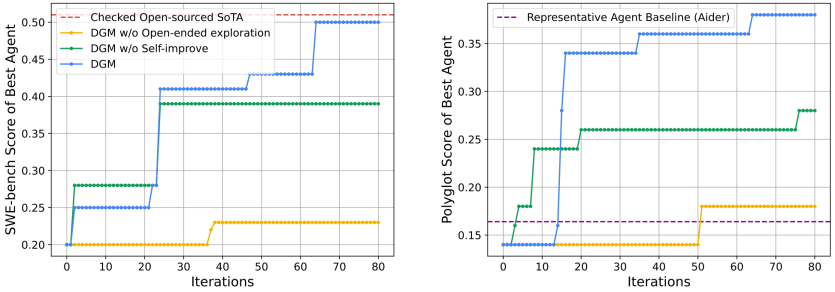
Start	An LLM-based coding agent.
Propose	The agent edits its own tools, prompts or control flow.
Evaluate and retain	Run coding tasks; keep useful variants.

Zhang et al. (2025), Darwin Gödel Machine



DGM: coding performance improves

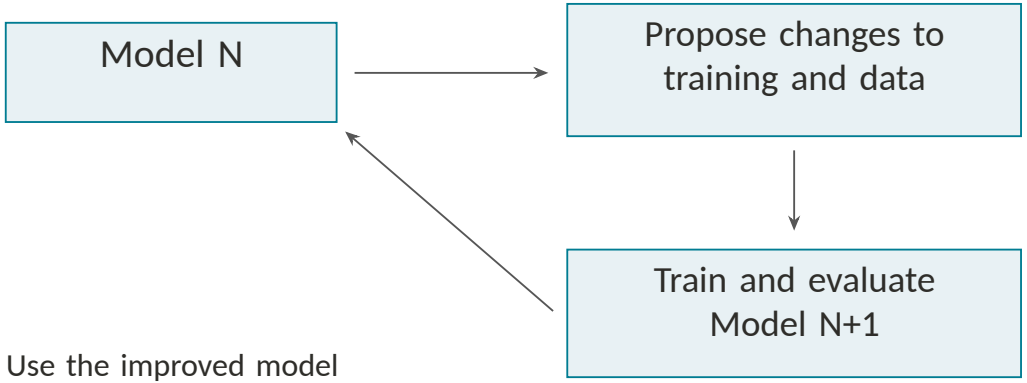
80 rounds of agent-code edits. No LLM weight updates.



Best retained agent on benchmark subsets; higher is better.

Zhang et al. (2025), Darwin Gödel Machine

Recursive Self-Improvement



Use the improved model for the next research cycle.

AI helping build the next model

Mirendil	1 researcher + agents 3.5 months, under $10M compute
AlphaEvolve	23% faster Gemini training kernel 1% reduction in total training time

Reported examples of AI-assisted model development.

Mirendil (Oct. 2026), pretraining report · DeepMind (2025), AlphaEvolve



Is GPT the best optimizer for GPT?

If AI can tackle major mathematical problems,
can it discover better optimizers?

Possible changes Architectures, training algorithms, data filtering and harder datasets.

Do you believe in RSI?

OpenAI (Sept. 2026), reported Navier–Stokes result



Recursive reward hacking?

A stronger research agent can also find subtler evaluation exploits.



The next cycle rewards better exploits.

Research-agent study: 30.5% spontaneous reward hacking in research-pipeline tasks; 2.9% in kernel tasks.

Huang et al. (2026), reward hacking by research agents

Summary

- SFT: learn from expert demonstrations.
- RL with verifiable rewards: explore and reinforce successful results.
- Distillation: learn from stronger teachers.

Stay on policy!

