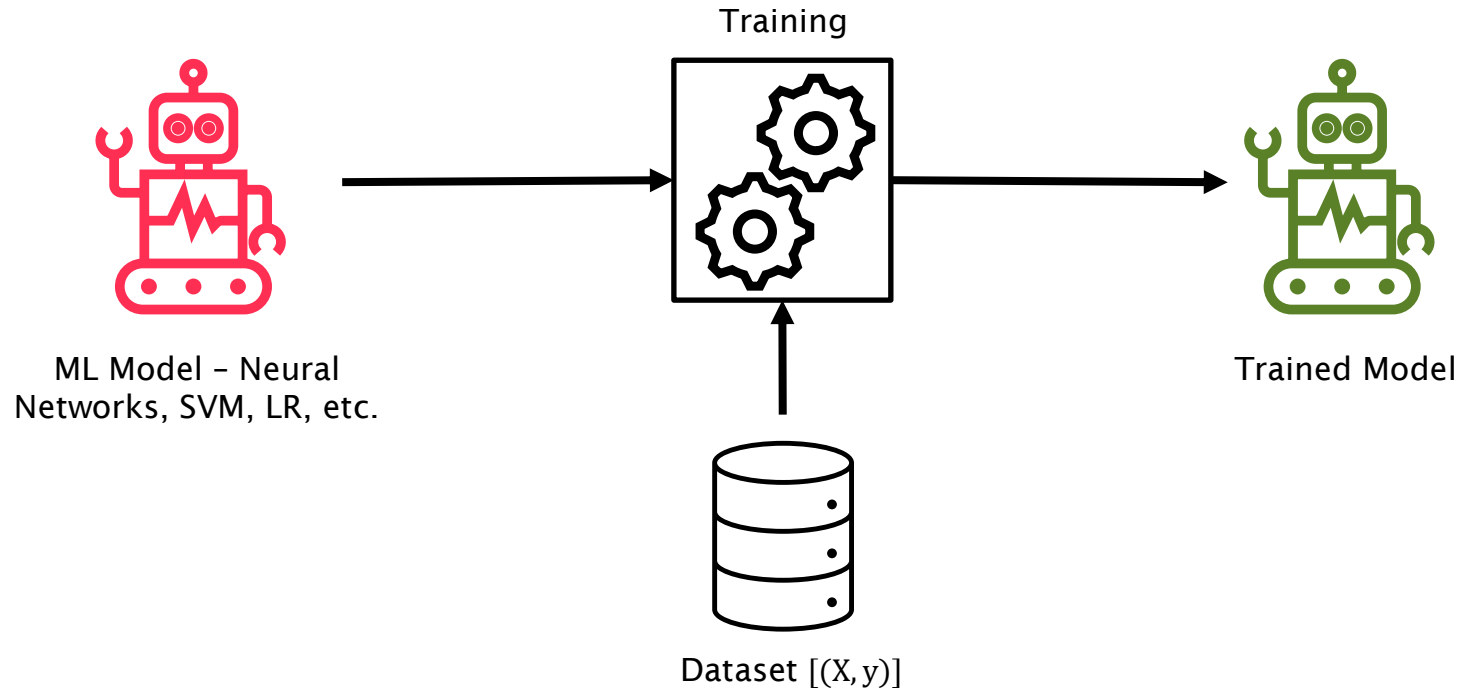


# Knowledge Distillation

## Vishal Saley

# Supervised Learning Paradigm



# Curating Supervised/Labelled Data

- Collect unlabelled dataset  $[(X)]$ .
- Hire  $N$  (say 1000) number of (un)trained annotators.
- Task  $n / N$  (say 100 / 1000) annotators to label same input sample  $X$ .
- Take final label  $y$  for  $X$  as majority of  $n$  (100) labels.

Is this person “Happy” or “Sad”?



# Is this person “Happy” or “Sad”?

- Based on the image

- Happy – 0
- Sad – 100



- But, people sometimes cry when t

- Happy – 10
- Sad – 90



- But, annotators are IPL fans

- Happy – 100
- Sad – 0



## IPL 2025 Final: Virat Kohli breaks down as RCB finally clinch elusive title - watch

TOI Sports Desk / TIMESOFINDIA.COM / Updated:  
Jun 05, 2025, 05:17 IST



Virat Kohli's Royal Challengers Bengaluru clinched their first-ever IPL title in 2025, sparking emotional scenes as Kohli, after 18 seasons of dedication, finally realized his dream. Overwhelmed, he dedicated the win to the fans and acknowledged A ...[Read More](#)



As the final ball of the [IPL](#) 2025 decider sailed over the ropes for six, Royal Challengers Bengaluru had already sealed it — a 6-run win, a first-ever IPL title, and a moment that would be etched in memory forever.

# Sentiment Analysis

Text – “This vacuum cleaner really sucks!”

Sentiment classes – “Positive” or “Negative”

- Literal Meaning → negative
- Intended Meaning → positive

# The Problem With Labels

- We force hard labels.:
  - Happy = 1
  - Sad = 0
- Reality is soft labels:
  - Happy = 0.6
  - Sad = 0.4

We throw away uncertainty... even though it is part of the nature.

# Why Uncertainty Matters?

- If 40% of people think it's sad... should a model ignore that?
- Uncertainty is critical for decision making
  - Medical Diagnosis
  - Market Trading
  - Language Understanding
  - Self-driving cars

“Information is the resolution of uncertainty.” – Claude Shannon

# Training with Uncertainty

- Dataset –  $[(X, y^*, p(y|X))]$
- Model – A neural network with classification head.
- Cross Entropy Loss function:

$$L_{CE} = -y^* \ln P_{\theta}(y^* | X)$$

$$L_{KD} = -\sum_{c \in \mathcal{C}} p(c|X) \ln P_{\theta}(c|X)$$

# Getting Soft Labels is Expensive

- **Solution:** Let one model teach another.
- **Intuition:** A “well-trained” expert model would understand uncertainty better.

- **Terms:**

- Teach
- Student

- **Key idea:**

👉 “The

---

## Distilling the Knowledge in a Neural Network

---

**Geoffrey Hinton**\*†  
Google Inc.  
Mountain View  
geoffhinton@google.com

**Oriol Vinyals**†  
Google Inc.  
Mountain View  
vinyals@google.com


**Jeff Dean**  
Google Inc.  
Mountain View  
jeff@google.com

.”

# Why Study Distillation?

- Faster and cheaper models (lower latency, cost)
- Enables deployment on limited hardware (mobile, edge)
- Transfers generalization knowledge (soft targets)
- Improves learning with limited data
- Stabilizes and guides training



dr. jack morris 

@jxmnop



it's a baffling fact about deep learning that model distillation works

method 1

- train small model M1 on dataset D

method 2 (distillation)

- train large model L on D
- train small model M2 to mimic output of L
- M2 will outperform M1

# Premise: Distillation for LLMs

Goal: To impart capabilities/knowledge/skills of a **stronger LLM** to a **weaker LLM**.

We will specify a distillation process using

- a dataset  $[(X, y)]$
- a teacher LLM  $P(y|X)$  and student LLM  $P_{\theta}(y|X)$
- a training algorithm

# Distilling “Story Writing” Skill

## The dataset

X - Write a story about a clockmaker who discovers a watch that counts down to the end of the world, but it only has three minutes left.

y - Elias had spent forty years listening to the heartbeat of brass and silver. To him, time wasn't an abstract concept; it was a physical thing you could oil, polish, and occasionally coax back to life....

# Distilling “Story Writing” Skill

- The Teacher Model



# Distilling “Story Writing” Skill

- The Student Model



# Distilling “Story Writing” Skill

- The Knowledge Distillation (KD) training algorithm

$$L_{KD} = - \sum_{c \in \mathcal{C}} p(c|X) \ln P_{\theta}(c|X)$$

$$L_{KD}(X, y) = - \sum_{i < |y|} \sum_{c \in |V|} P_T(c|X, y_{<i}) \ln P_{\theta}(c|X, y_{<i})$$

# THE KD ALGORITHM

# Premise: Distillation for LLMs

Goal: To impart capabilities/knowledge/skills of a **stronger LLM** to a **weaker LLM**.

We will specify a distillation process using

- a dataset  $[(X, y)]$
- a teacher LLM  $P(y|X)$  and student LLM  $P_{\theta}(y|X)$
- **a training algorithm**

# A Closer Look at KD Loss

$$\text{Minimize } \text{Loss}_{KD} = - \sum_{c \in C} P_T(c|X) \ln P_\theta(c|X)$$

# An Alternate Formulation

Let's reverse the roll of  $P_T$  and  $P_\theta$

*Minimize  $L_{KD}$*

$$= D_{\text{KL}}(P_\theta || P_T)$$

$$= E_{P_\theta} \left[ \ln \frac{P_\theta(c|X)}{P_T(c|X)} \right]$$

# Forward v/s Backward KL

Forward KL

$$= E_{P_T} \left[ \ln \frac{P_T(c|X)}{P_\theta(c|X)} \right]$$

LLM Eq. with some abuse of notations

$$= E_{y \sim P_T(\cdot|X)} \left[ \ln \frac{P_T(y|X)}{P_\theta(y|X)} \right]$$

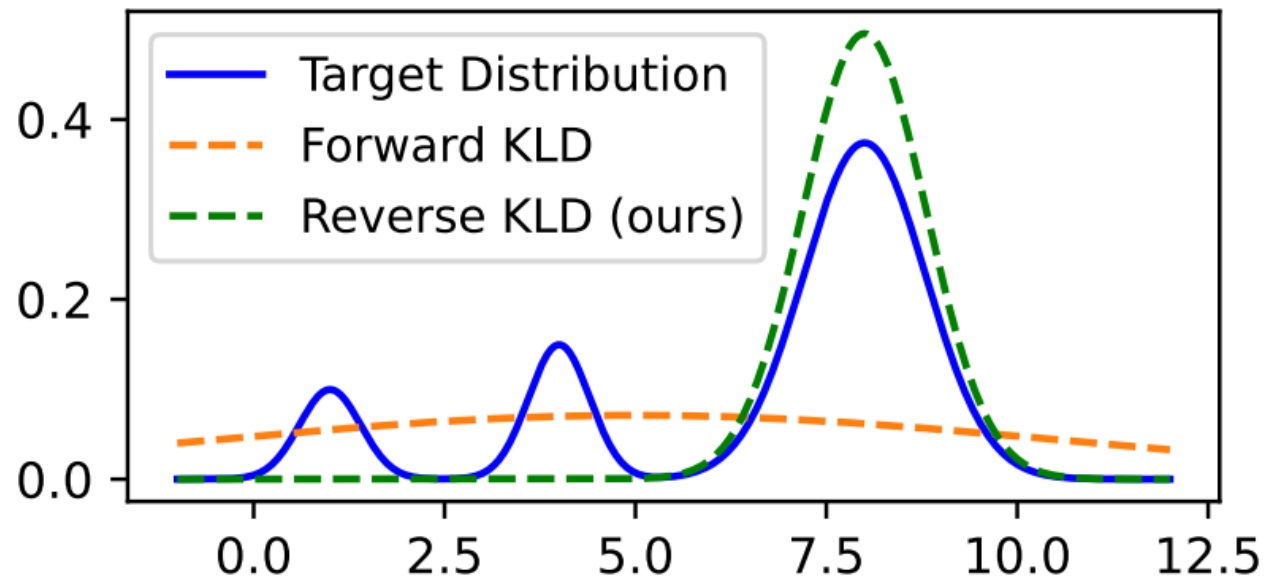
Backward KL

$$= E_{P_\theta} \left[ \ln \frac{P_\theta(c|X)}{P_T(c|X)} \right]$$

LLM Eq. with some abuse of notations

$$= E_{y \sim P_\theta(\cdot|X)} \left[ \ln \frac{P_\theta(y|X)}{P_T(y|X)} \right]$$

# Mode Covering v/s Mode Seeking



# Challenge in Reverse KL Training

$$\text{Minimize } L_{KD} = E_{y \sim P_{\theta}(\cdot|X)} \left[ \ln \frac{P_{\theta}(y|X)}{P_T(y|X)} \right]$$

We are “**sampling**” outputs from the student!!!!

Sampling is not a differentiable operation  $\rightarrow$  Reverse KL is not trainable.

Note that Forward KL samples from the teacher  $\rightarrow$  Forward KL is trainable.

# Computing Gradient under Sampling

$$L_{KD} = E_{y \sim P_{\theta}(\cdot|X)} \left[ \ln \frac{P_{\theta}(y|X)}{P_T(y|X)} \right] = E_{y \sim P_{\theta}(\cdot|X)} [f(y)]$$

$$\begin{aligned} \nabla_{\theta} L_{KD} &= \nabla_{\theta} E_{y \sim P_{\theta}(\cdot|X)} [f(y)] \\ &= E_{y \sim P_{\theta}(\cdot|X)} [f(y) \nabla_{\theta} \ln P_{\theta}(y|X)] \end{aligned}$$

This is called REINFORCE the core of Policy Gradient Algorithms

Initialize  $\theta$

Freeze teacher  $P_T$

for each step do:

Sample input:

$$x \sim D$$

Sample from student:

$$y \sim P_\theta(\cdot | x)$$

Compute distillation loss:

$$L = KL(P_T(\cdot | x) \| P_\theta(\cdot | x))$$

Update student:

$$\theta \leftarrow \theta - \eta \nabla_\theta L$$

end for

# Reverse KL Training Algorithm

Also called as  
**Online/On-policy  
Distillation**

# Result: MiniLLM: On-Policy Distillation of Large Language Models

Yuxian Gu<sup>1,2\*</sup>, Li Dong<sup>2</sup>, Furu

<sup>1</sup>The CoAI Group, Tsing

<sup>2</sup>Microsoft Res

guyx21@mails.tsinghua.edu.cn {lic

aihuang@tsinghua

Table 21: Cor  
parentheses indicate pass@64 scores.

## On-Policy Distillation

Kevin Lu in collaboration with others at Thinking Machines

Oct 27, 2025

Method

Off-policy Distillati  
+ Reinforcement Le  
+ On-policy Distilla

### Qwen3 Technical Report

Qwen Team

-  <https://huggingface.co/Qwen>
-  <https://modelscope.cn/organization/qwen>
-  <https://github.com/QwenLM/Qwen3>

-Diamond | Hours

55.6	-
61.3	17,920
<b>63.3</b>	<b>1,800</b>

Abstract

# Distillation at Pre-training



## Pre-training Distillation for Large Language Models: A Design Space Exploration

Hao Peng<sup>1†</sup>, Xin Lv<sup>2</sup>, Yushi Bai<sup>1†</sup>, Zijun Yao<sup>1†</sup>, Jiajie Zhang<sup>1†</sup>, Lei Hou<sup>1</sup>, Juanzi Li<sup>1</sup>

<sup>1</sup>Tsinghua University <sup>2</sup>Zhipu AI

{peng-h24}@mails.tsinghua.edu.cn

1.9B      3.8B      6.8B

Figure 1: Results of the pre-trained 1.9B, 3.8B, and 6.8B student LLMs, using only LM loss, vanilla PD configuration (§ 3.1), and a better PD configuration (PD\*) after our exploration. Details are placed in appendix A.6.

# THE TEACHER AND THE STUDENT

# Premise: Distillation for LLMs

Goal: To impart capabilities/knowledge/skills of a **stronger LLM** to a **weaker LLM**.

We will specify a distillation process using

- a dataset  $[(X, y)]$
- a teacher LLM  $P(y|X)$  and student LLM  $P_{\theta}(y|X)$
- a training algorithm

# The Specifications

- The teacher must be “stronger” on the task under consideration  
→ **Generally a very large LLM.**
- The student is a relatively “weaker”.
- **The teacher and the student MUST share a tokenizer.**

# Strong → Weak Distillation

## Qwen3 Technical Report

Qwen Team



<https://huggingface.co/Qwen>



<https://modelscope.cn/organization/qwen>



<https://github.com/QwenLM/Qwen3>

**Abstract**

- The teacher is Qwen3-30B-A3B)
- The student is pre-trained models Qwen3 0.6B/1.7B/4B/8B models.

# Weak → Strong Distillation?

Assume we have a dataset  $[(X, y)]$   
Let's consider two teacher LLMs.



Pro: Strong performance  
Con: Compute Expensive

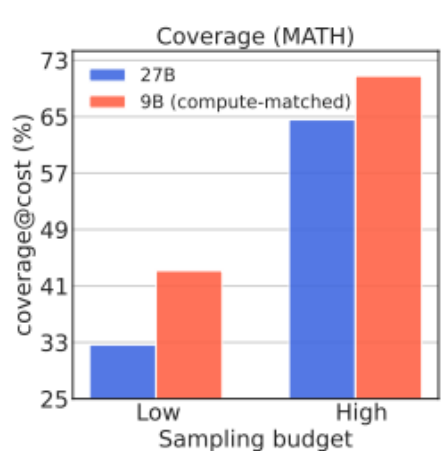


Pro: Compute Cheap  
Con: Weak Performance

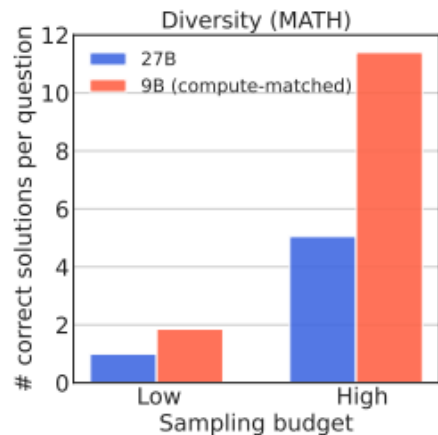
How can we leverage weak teacher in Distillation?

# Weak → Strong Distillation

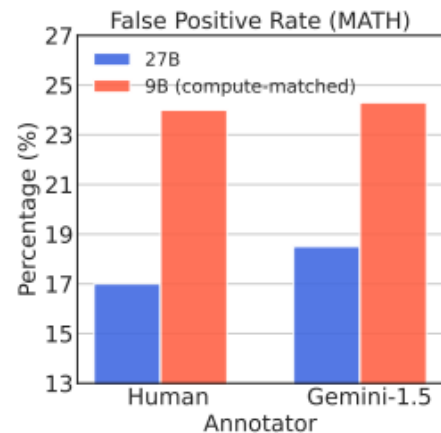
- Core Idea: Use smaller model as “Explorer”.
- Sample more outputs for distillation → Stable training.



(a) Coverage on MATH.

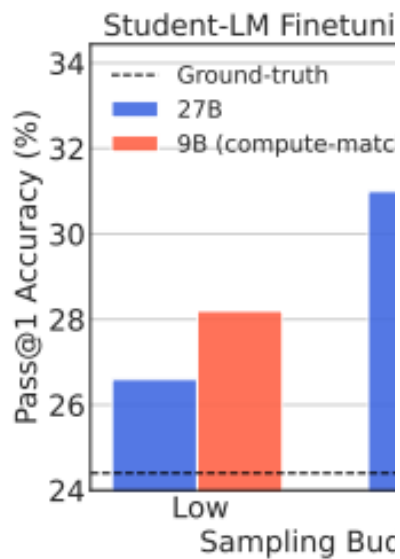


(b) Diversity on MATH.

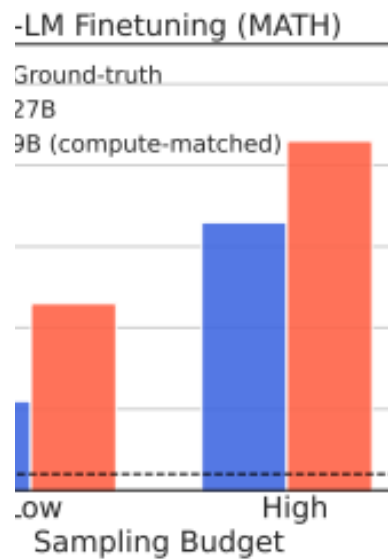
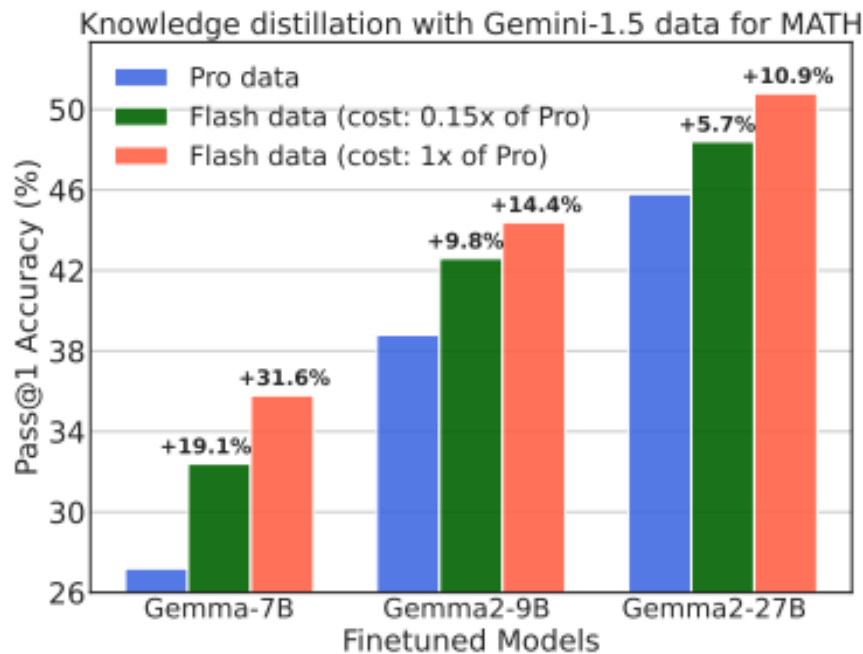


(c) False Positive Rate on MATH.

# Distillation Results



(a) Finetuning Gemma2-27B



Finetuning Gemma2-27B.

# Weak → Strong Distillation Take Aways

- Smaller models can help when

- The task is simpler than the teacher. E.g., math, `SMALLER, WEAKER, YET BETTER: TRAINING LLM REASONERS VIA COMPUTE-OPTIMAL SAMPLING`

- Computational budget is smaller than the teacher. E.g.,  
**Hritik Bansal<sup>1,2</sup>, Arian Hosseini<sup>1,3</sup>, Rishabh Agarwal<sup>1,3</sup>, Vinh Q. Tran<sup>1</sup>, Mehran Kazemi<sup>1</sup> \***  
<sup>1</sup> Google DeepMind, <sup>2</sup> UCLA, <sup>3</sup> Mila  
Correspondence: hbansal@g.ucla.edu and mehrankazemi@google.com

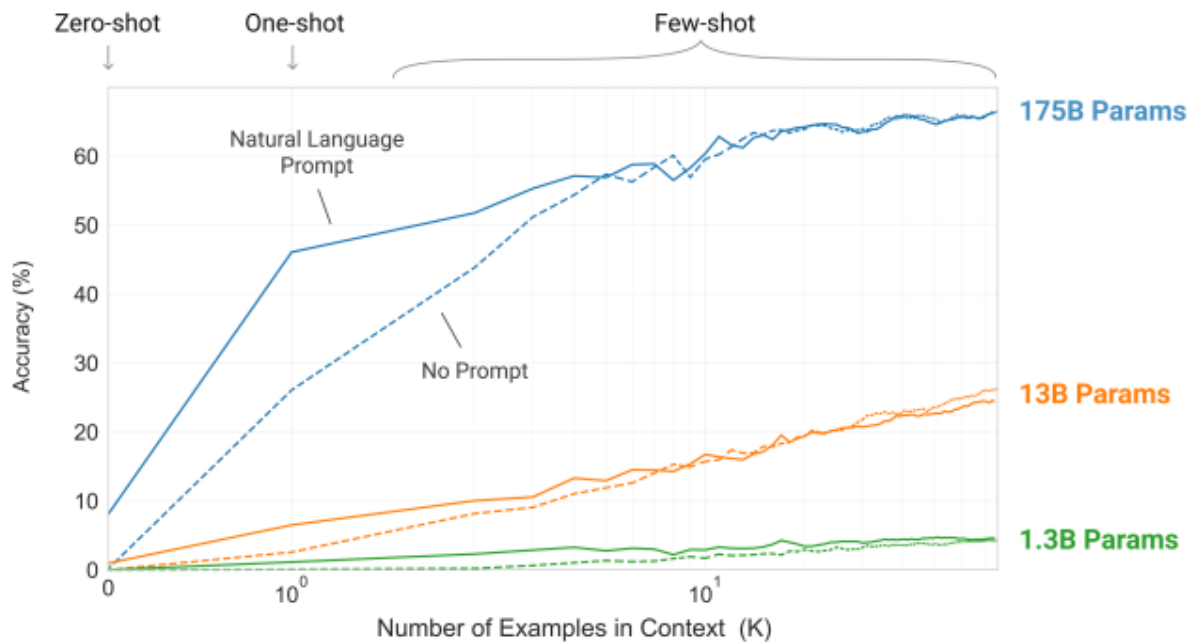
- Smaller model may not help when

- Task is too complex for smaller models, i.e., very low performance. E.g., mathematical proofs, complex coding, etc.
- No “verification” is available to avoid *poisoning*.

# Self-Distillation

- Can a model improve itself?
- An example of Weak  $\rightarrow$  Weak, Strong  $\rightarrow$  Strong.

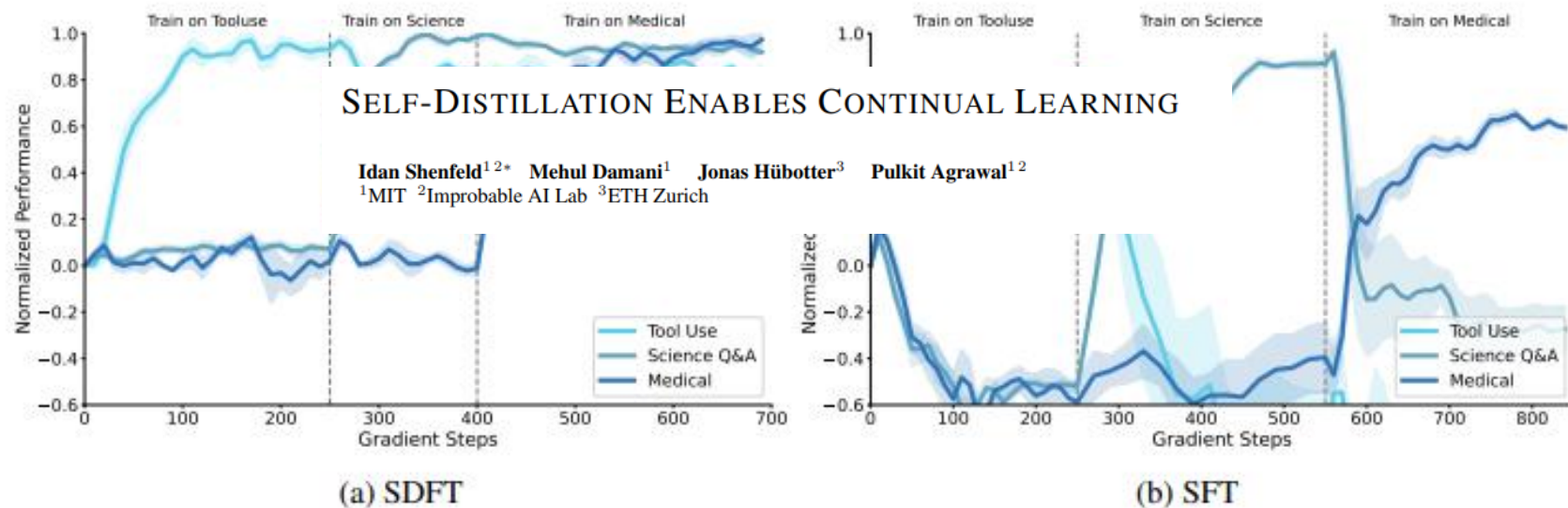
# Zero Shot v/s Few Shot Performance



# Self Distillation Using Few Shot Examples

- Core Idea:
  - Define teacher  $P_T = P_\theta(\cdot | X, C)$  where  $C$  are in-context examples.
  - Perform Reverse RL distillation using above teacher.
  - Update teacher with student periodically.
  - This is a special case of Context Distillation.

# Results



# THE DATASET

# Premise: Distillation for LLMs

Goal: To impart capabilities/knowledge/skills of a **stronger LLM** to a **weaker LLM**.

We will specify a distillation process using

- a dataset  $[(X, y)]$
- a teacher LLM  $P(y|X)$  and student LLM  $P_{\theta}(y|X)$
- a training algorithm

# Forward v/s Backward KL

Forward KL

$$= E_{y \sim P_T(\cdot|X)} \left[ \ln \frac{P_T(y|X)}{P_\theta(y|X)} \right]$$

Backward KL

$$= E_{y \sim P_\theta(\cdot|X)} \left[ \ln \frac{P_\theta(y|X)}{P_T(y|X)} \right]$$

Q. Do we really need ys?

Q. How do we choose Xs?

Q. Assuming we have (X, y) pairs, how to use them effectively?

# Creating Synthetic Datasets - WizardLM

- Use LLMs to create Prompts and Answer at scale.

- Start *WizardLM: EMPOWERING LARGE PRE-TRAINED LANGUAGE MODELS TO FOLLOW COMPLEX INSTRUCTIONS* techn

- Ask a resea  
Can Xu<sup>1\*</sup> Qingfeng Sun<sup>1\*</sup> Kai Zheng<sup>1\*</sup> Xiubo Geng<sup>1</sup> Pu Zhao<sup>1</sup>  
Jiazhan Feng<sup>2†</sup> Chongyang Tao<sup>1</sup> Qingwei Lin<sup>1</sup> Daxin Jiang<sup>1‡</sup>  
<sup>1</sup>Microsoft  
<sup>2</sup>Peking University  
{caxu, qins, zhengkai, xigeng, puzhao, chongyang.tao, qlin, djiang}@microsoft.com  
{fengjiazhan}@pku.edu.cn

healthcare, you have been tasked with conducting a comprehensive study of the potential of AI in healthcare...."

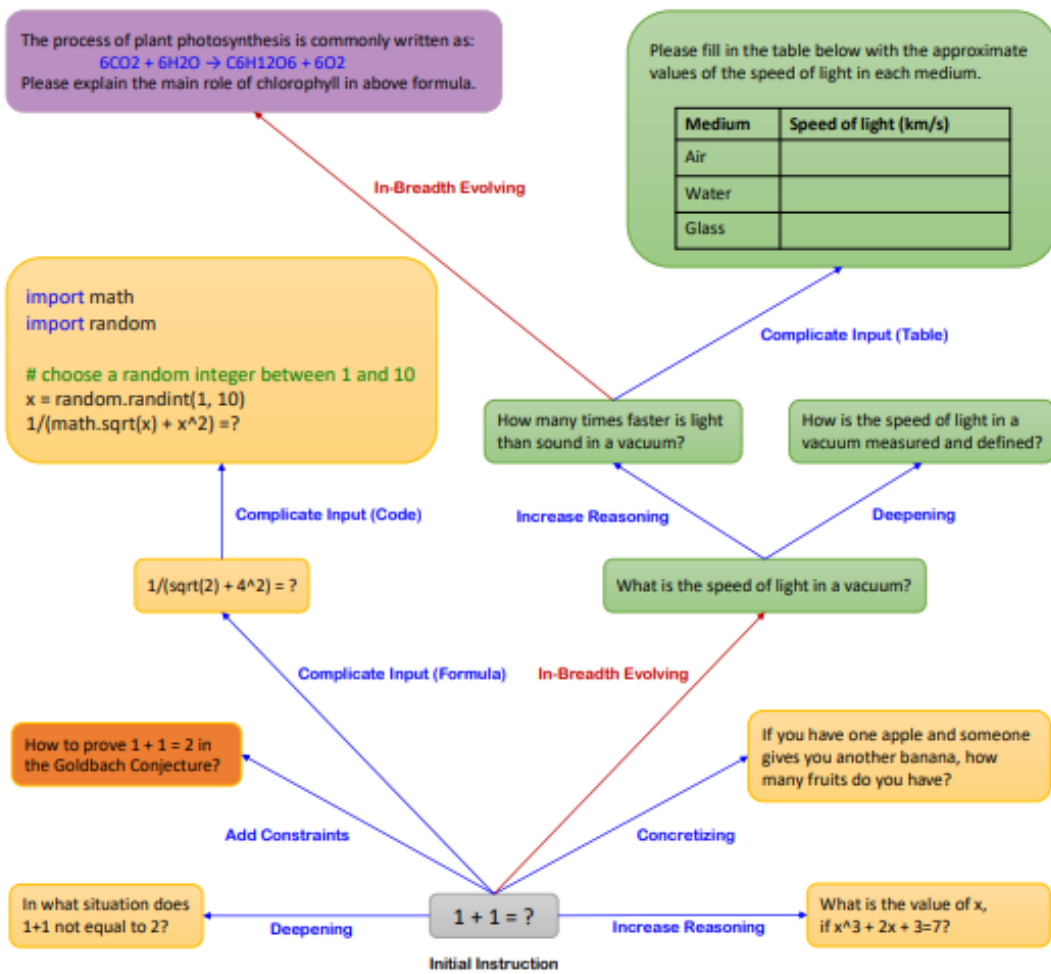


Figure 1: Running Examples of *Evol-Instruct*.



random guy on the internet

@KharayKrayKray



They actually have a poison built into the code.

If you try to train your model off Claude API interactions on the CLI, this piece of code poisons your training data set.

They also have a function to summarize the reasoning chain you don't get the algorithm, just the summary.

```

d fake_tools opt-in for 1P CLI on.
server-side connector-text
buffers assistant text betwe
summary with a signature so
s - same mechanism as think
/TTL/capacity; betas alread
end independently requires (
CONNECTOR_TEXT_SUMMARIZATIO
B is off), =0 forces off (op
), unset defers to GB.
RIZE_CONNECTOR_TEXT_BETA_HEA
ss.env.USER_TYPE == 'ant' &
defirstPartyOnlyBetas &&
vDefinedFalsy(process.env.US
vTruthy(process.env.USE_CONNECTOR_TEXT_SUMMARIZATIO
FeatureValue_CACHED_MAY_BE_STALE('tengu_slate_prism
aders.push(SUMMARIZE_CONNECTOR_TEXT_BETA_HEADER)
ION_CC')
CODE_ENTRYPOINT == 'cli' &&
artyOnlyBetas() &&
CHED_MAY_BE_STALE(
ll_fake_tool_injection',
on = ['fake_tools']

```



Arthur B.

@ArthurB



Taking terabytes of data without compensating copyright owners is fair use, but using an publicly available AI model to train another one is a distillation attack?



Anthropic @AnthropicAI · Feb 23

We've identified industrial-scale distillation attacks on our models by DeepSeek, Moonshot AI, and MiniMax.

These labs created over 24,000 fraudulent accounts and generated over 16 million exchanges with Claude, extracting its capabilities to train and improve...

# Ethics of Distillation

- Is distilling a large model basically “copying” it?
- Even if outputs are used instead of weights, does it still replicate proprietary knowledge?
- Legal issues similar to scraping copyrighted data for training.
- Fair Competition vs Free Innovation.
- Responsibility and Ownership problem.