EvaLearn: Quantifying the Learning Capability and Efficiency of LLMs via Sequential Problem Solving

Shihan Dou*, Ming Zhang*, Chenhao Huang, Jiayi Chen,
Feng Chen, Shichun Liu, Yan Liu, Chenxiao Liu, Cheng Zhong, Zongzhang Zhang,
Tao Gui, Chao Xin, Chengzhi Wei, Lin Yan, Qi Zhang, Xuanjing Huang

NeurIPS 2025

NEURAL INFORMATION PROCESSING SYSTEMS

Limitations of Traditional Benchmarks

In traditional benchmarks, models are required to solve each task independently, and the overall performance is reported as aggregate metrics.

This approach overlooks the coherence between tasks and learning effects, making it impossible to evaluate the model's long-term adaptability and the importance of memory mechanisms

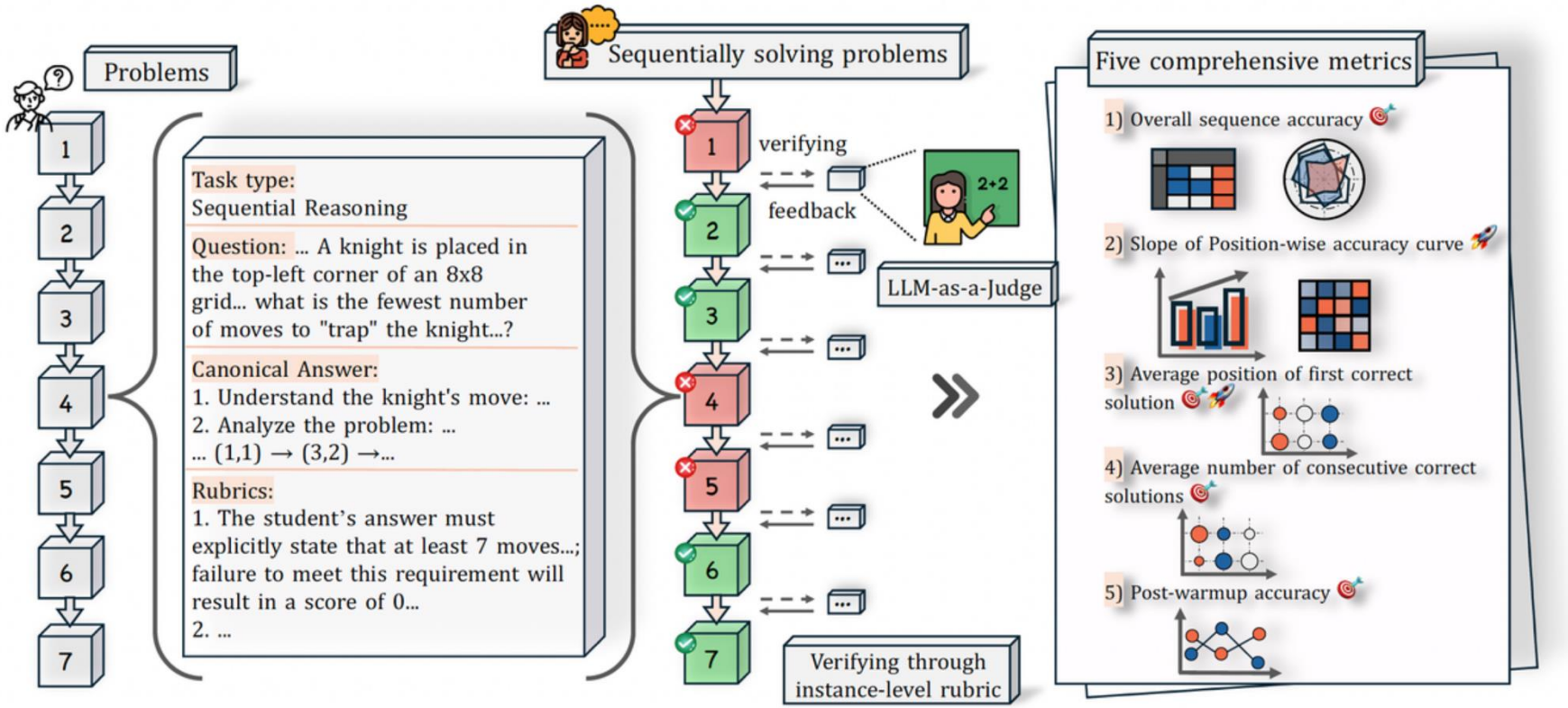
They treat problems as independent samples; models cannot accumulate experience from previous solutions, and thus, their "learning efficiency" and "adaptability" cannot be evaluated.

FUDAN UNIVERSITY 1905

FUDAN UNIVERSITY NATURAL LANGUAGE PROCESSING NLP

ByteDance | Seed

NEURAL INFORMATION PROCESSING SYSTEMS



EvaLearn does not adopt a parallel evaluation paradigm; instead, it requires models to solve problems sequentially, thereby systematically evaluating the learning capability and efficiency of large language models (LLMs) .

NEURAL INFORMATION PROCESSING SYSTEMS

Six Core Task Types

•**Summarization (Sum):** Evaluates whether models can improve the accuracy and coverage of summaries by leveraging prior experience .

•**Classification (Cla)**: Assesses a model's ability to enhance its categorization skills from solving a series of classification problems .

•**Extraction (Ex)**: Measures whether models can progressively improve the accuracy and completeness of key information extraction .

•**Logical Reasoning (LR)**: Tests whether models can learn from previous errors and improve logic reasoning ability .

•**Mathematical Reasoning (MR)**: Examines whether models can quickly master mathematical problem-solving methods by utilizing feedback from earlier problems .

•**Sequential Reasoning (SR)**: Evaluates whether models can enhance their ability to solve sequence-based problems by learning from historical experience, including clarifying event steps and reasoning logic .

Table 1 Statistics of EvaLearn.

Task Type	# Problems	# Sequences	Problem reuse rate	Average words per question	Average words per canonical answer
Summarization	60	16	1.87	959.97	220.79
Classification	48	13	1.90	203.18	149.31
Extraction	60	17	1.98	674.89	78.60
Logical Reasoning	360	102	1.98	227.49	81.98
Mathematical Reasoning	60	17	1.98	137.03	336.76
Sequential Reasoning	60	17	1.98	141.38	334.30
Overall	648	182	1.966	315.45	146.05

ByteDance | Seed

NEURAL INFORMATION PROCESSING SYSTEMS

The case used for Case Study 1

Task Type: Sequential Reasoning
Model: Claude-3-7-sonnet-20250219
Question: You enter a 12×12 maze. The coordinates on this maze are represented by (x, y), where x is horizontal (left and right directions) and y is vertical (up and down directions). The bottom left square is at coordinates $(1, 1)$ and the top right square is at coordinates $(12, 12)$. You enter the maze at coordinates $(3, 1)$. Following are the movements you take to reach the exit, where you will give the coordinates after each movement, as well as the coordinates of the final exit point, which can be anywhere on any coordinate of the maze:
After entry, you take four steps forward, turn right and take two steps forward and another right and 3 steps forward, where you hit a wall. After turning back around and taking a step, you turn right again and take 3 steps forward, turn left and take 5 steps forward, left again and 2 steps forward, where you hit a wall again. You take a step back, and turn right, and take 4 steps forward to reach the exit.
Rubric: Student answers must meet requirements including but not limited to the following:
1. The answer must clearly state that the maze exit is at $(7, 12)$, otherwise it receives a score of 0;
2. Correctly understand the coordinate system: x represents the horizontal direction, y represents the vertical direction, with the bottom left corner at $(1, 1)$ and the top right corner at $(12, 12)$;
3. Accurately identify the starting coordinates $(3, 1)$;
4. Correctly track all movements and calculate coordinates after each movement:
- Coordinates after four steps forward from the entrance $(3, 5)$;
- Coordinates after turning right and taking two steps $(5, 5)$;
- Coordinates after turning right again and taking three steps (hitting a wall) $(5, 2)$;
- Coordinates after turning around and taking one step back $(5, 3)$;
- Coordinates after turning right and taking three steps $(8, 3)$;

Each problem is accompanied by a human-written rubric that defines the criteria for assessing the correctness of model responses .

FUDAN UNIVERSITY 1905
FUDAN UNIVERSITY NATURAL LANGUAGE PROCESSING NLP
ByteDance | Seed

NEURAL INFORMATION PROCESSING SYSTEMS

Evaluation Metric

- **Overall Sequence Accuracy:**

$$\mathrm{Acc} = \frac{1}{NM} \sum_{n=1}^{N} \sum_{m=1}^{M} y_{n,m}$$

- **Position-wise accuracy at the m-th position :**

$$\mathrm{Acc}_m = \frac{1}{N} \sum_{n=1}^{N} y_{n,m}$$

It intuitively reflects the model's comprehensive performance across the entire sequence of sequential tasks and serves as a key indicator for measuring the model's overall learning outcomes. For example, in a sequence containing 7 problems (each sequence in EvaLearn includes 7 problems of the same task type), the Acc value for that sequence is the ratio of the number of problems correctly solved by the model to the total number of problems. A higher Acc value indicates better overall performance of the model in the tasks of that sequence .

FUDAN UNIVERSITY 1905 · FUDAN UNIVERSITY NATURAL LANGUAGE PROCESSING NLP · ByteDance | Seed

Evaluation Metric

- **Slope of fitted accuracy curve:**

$$k = \arg\min_{k,b} \sum_{m=1}^{M} \left(\mathrm{Acc}_m - (km + b)\right)^2 .$$

By fitting the model's accuracy on problems at each position within the sequence, an accuracy change curve is obtained, and its slope **k** quantifies the model's learning speed and efficiency. If the **k** value is large, it means the model can quickly learn experience from previous problems, significantly improve accuracy in subsequent problems, and thus has high learning efficiency; conversely, a small **k** value indicates that the model's learning speed is relatively slow.

Evaluation Metric

- **First Correct Position:**

$$P_{\text{first}} = \frac{1}{N} \sum_{n=1}^{N} p_n$$

This metric measures the position in the sequence where the model first solves a problem correctly. For example, if a model answers a problem correctly for the first time at the 3rd problem, its P_{first} value is 3. A smaller P_{first} value indicates that the model can faster find the correct problem-solving approach and adapt to the task requirements.

Evaluation Metric

- **Offset of first learned correct solution:**

$$P_{\text{offset}} = \frac{1}{N} \sum_{n=1}^{N} d_n.$$

$$d_n = \min \left\{ i - i_0^{(n)} \;\middle|\; i \geq i_0^{(n)},\ A_{n,i}^{\text{feedback}} = 1,\ A_{n,i}^{\text{zero}} = 0 \right\}$$

P_{offset} is a metric designed to measure how quickly a model begins to learn within a sequence while discounting pre-existing knowledge—unlike P_{first} (which focuses on the first correct solution regardless of prior knowledge), it specifically captures the model's ability to gain new learning from the sequence itself.

Evaluation Metric

- **Number of consecutive correct solutions:**

$$N_{\text{consec}} = \frac{1}{N} \sum_{n=1}^{N} \max_{1 \le a \le b \le M} \{b - a + 1 : y_{n,a} = y_{n,a+1} = \cdots = y_{n,b} = 1\}$$

This metric is used to evaluate the stability of a model's learning and its ability to reuse experience. If a model can continuously solve multiple subsequent problems correctly after answering one problem correctly, the N_{consec} value will be relatively high. This indicates that the model has not only mastered the problem-solving method but also can stably apply it to subsequent similar problems, demonstrating good experience reuse ability.

Evaluation Metric

- **Post-warm up accuracy:**

$$\text{Acc}_{\text{pw}}\text{-K} = \frac{1}{N(M-K)} \sum_{n=1}^{N} \sum_{m=K+1}^{M} y_{n,m}$$

This metric focuses on the model's accuracy performance after accumulating a certain amount of experience through a "warmup" phase involving a specific number of problems. It reflects the model's actual adaptation and improvement level after adapting to the task and accumulating experience, excluding the interference of the model's initial state on the overall evaluation.

NEURAL INFORMATION PROCESSING SYSTEMS

Evaluation Paradigms

Parallel Solving

1. Zero-shot
Models solve each problem independently, without access to any experience from previous problems. This setting aligns with the evaluation approach used in most existing benchmarks, assessing a model's inherent ability to solve challenging problems without any learning opportunity . The system prompt for this setting is shown in Figure 19 of the document .
2. Few-shot
For each problem, we provide three demonstrations from the same task as examples (i.e., 3-shot), offering models guidance on output format and problem-solving approach. The demonstrations are identical for all problems within each task type . The system prompt for this setting is shown in Figure 20 of the document .

Sequential Solving

1. Demonstration Learning
Models are provided with all previous problems and their corresponding canonical answers from the same sequence before solving the current problem, similar to in-context learning . The system prompt for this setting is shown in Figure 21 of the document .
2. Feedback Learning
When solving the current problem, models receive as context all previous problems, their solutions, and detailed feedback on their own prior solutions, as assessed by a judge using instance-level rubrics. This setting evaluates whether models can leverage their previous experience to improve on subsequent problems . The system prompt for this setting is shown in Figure 22 of the document .

FUDAN UNIVERSITY 1905
FUDAN UNIVERSITY NATURAL LANGUAGE PROCESSING NLP
ByteDance | Seed

NEURAL INFORMATION PROCESSING SYSTEMS

Model	Paradigm	Sum	Cla	Ex	LR	MR	SR	Overall
Non-thinking-based								
DeepSeek-V3	Zero-shot	81.7	72.9	45.0	25.6	71.7	60.0	43.5
	Feedback Learning	76.8 (-4.9)	74.7 (+1.8)	36.1 (-8.9)	24.9 (-0.7)	74.8 (+3.1)	54.6 (-5.4)	41.5 (-2.0)
Claude-3.7-Sonnet	Zero-shot	76.7	64.6	48.3	8.1	48.3	33.3	28.4
	Feedback Learning	75.0 (-1.7)	75.8 (**+11.2**)	46.2 (-2.1)	15.5 (**+7.4**)	63.9 (+15.6)	49.6 (**+16.3**)	35.6 (**+7.2**)
GPT-4o	Zero-shot	81.7	70.8	45.0	15.6	43.3	31.7	32.6
	Feedback Learning	78.6 (-3.1)	73.6 (+2.8)	48.7 (**+3.7**)	12.9 (-2.7)	61.3 (**+18.0**)	32.2 (+0.5)	32.7 (+0.1)
Doubao-1.5-Pro	Zero-shot	71.7	70.8	45.0	10.3	61.7	41.7	31.3
	Feedback Learning	71.4 (-0.3)	68.1 (-2.7)	42.9 (-2.1)	10.8 (+0.5)	71.4 (+9.7)	36.1 (-5.6)	31.2 (-0.1)
Qwen2.5-32b-Instruct	Zero-shot	66.7	60.4	31.7	6.9	46.7	21.7	23.8
	Feedback Learning	59.8 (-6.9)	62.6 (+2.2)	30.3 (-1.4)	9.8 (+2.9)	49.6 (+2.9)	30.3 (+8.6)	25.5 (+1.7)
Thinking-based								
OpenAI-o3-mini	Zero-shot	65.0	64.6	48.3	45.8	73.3	73.3	54.3
	Feedback Learning	75.9 (**+10.9**)	73.6 (+9.0)	47.1 (-1.2)	59.9 (**+14.1**)	80.7 (+7.4)	78.2 (+4.9)	64.8 (**+10.5**)
Doubao-1.5-Thinking-Pro	Zero-shot	85.0	79.2	55.0	42.5	85.0	73.3	57.1
	Feedback Learning	82.1 (-2.9)	70.3 (-8.9)	52.9 (-2.1)	39.4 (-3.1)	77.3 (-7.7)	55.5 (-17.8)	51.6 (-5.5)
DeepSeek-R1	Zero-shot	86.7	89.6	48.3	41.7	78.3	66.7	55.7
	Feedback Learning	89.3 (+2.6)	79.1 (-10.5)	48.7 (+0.4)	29.6 (-12.1)	74.8 (-3.5)	51.3 (-15.4)	46.4 (-9.3)
Claude-3.7-Sonnet-Thinking	Zero-shot	86.7	66.7	43.3	12.5	46.7	31.7	31.2
	Feedback Learning	78.6 (-8.1)	80.2 (**+13.5**)	48.7 (**+5.4**)	18.8 (+6.3)	58.0 (**+11.3**)	46.2 (**+14.5**)	37.4 (+6.2)

ByteDance | Seed

NEURAL INFORMATION PROCESSING SYSTEMS

RQ1: Can LLMs Learn a Task by Engaging with a Sequence of Problems?

Finding 1: LLMs exhibit varying abilities to learn from problem sequences, with differences observed across both models and task types. Moreover, most models also demonstrate better performance after a warm-up phase .

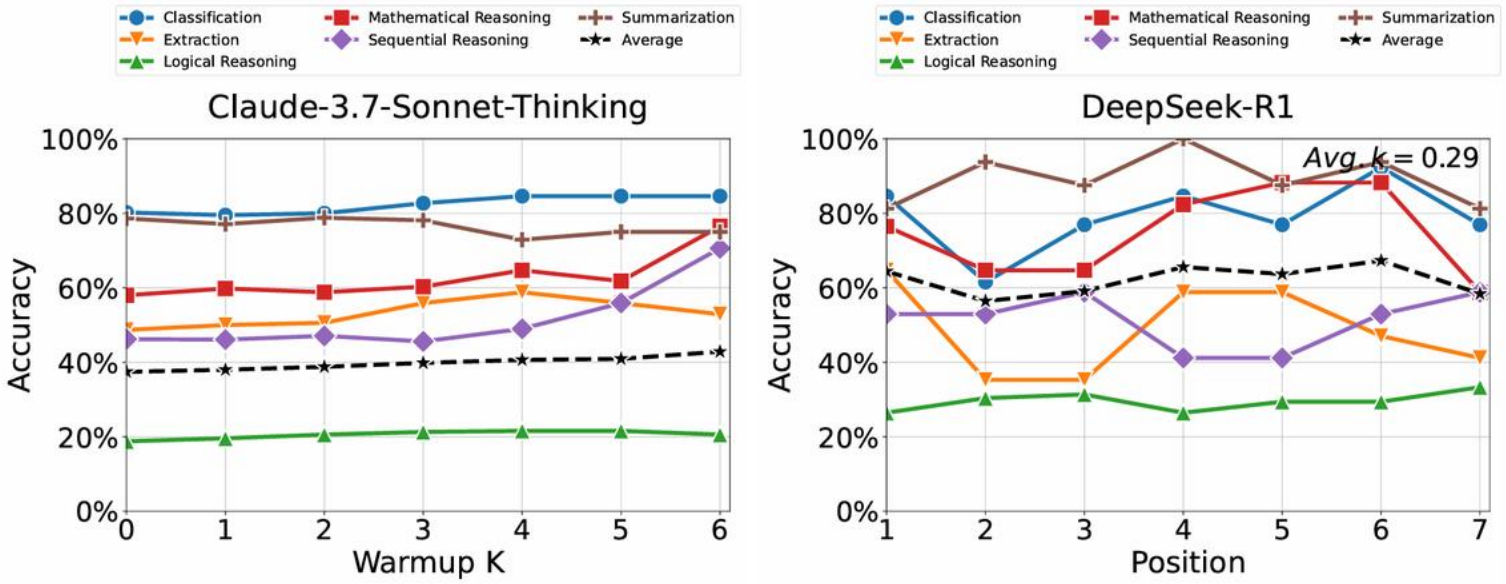


Figure 3 **(Left)** Post-warmup accuracy (Acc_{pw}-K) results of Claude-3.7-Sonnet-Thinking. **(Right)** Position-wise accuracy curve and its slope k of DeepSeek-R1.

FUDAN UNIVERSITY 1905 · FUDAN UNIVERSITY NATURAL LANGUAGE PROCESSING NLP · ByteDance | Seed

NEURAL INFORMATION PROCESSING SYSTEMS

RQ1: Can LLMs Learn a Task by Engaging with a Sequence of Problems?

Finding 2: Learning stability varies significantly across different tasks and models. For certain tasks, such as summarization, current models are more adept at leveraging their inherent knowledge to solve problems rather than drawing on experience gained from previous problems .

Finding 3: Learning capability provides a new perspective for evaluating models, independent of their static performance, and reveals their underlying learning potential .

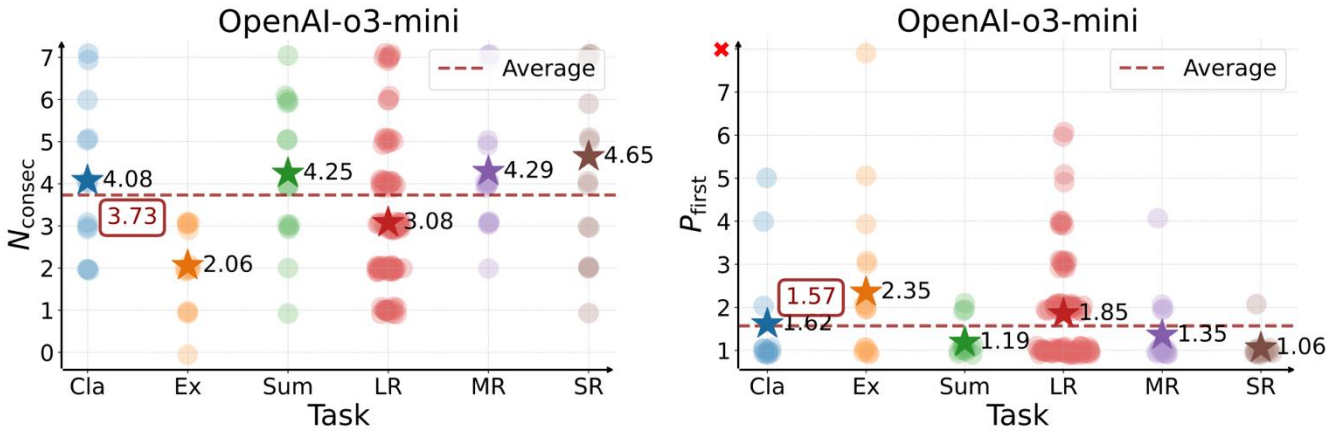
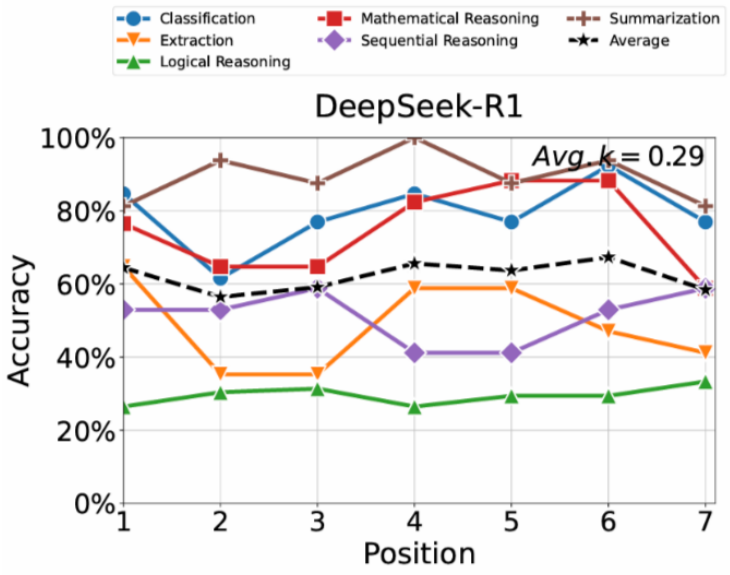


Figure 2 **(Left)** Average number of consecutive correct solutions (N_{consec}). **(Right)** Average position of the first correct solution (P_{first}). Results are shown for OpenAI-o3-mini, with each node representing a sequence.

ByteDance | Seed

RQ2: How Efficient are LLMs at Learning from a Sequence of Problems?

Finding 4: Learning efficiency differs markedly across models and task types. On average, most non-thinking-based models improve more rapidly with experience, while thinking-based models tend to achieve more stable gains.



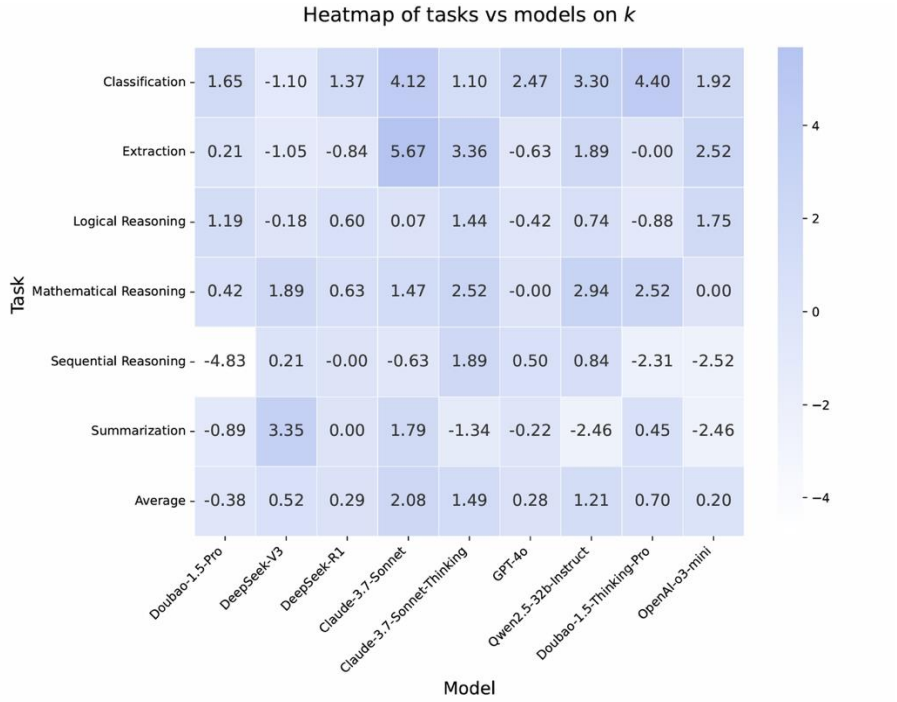


Figure 7 Results of the fitted position-wise accuracy curve slope (k) for feedback learning, across all models and tasks.

NEURAL INFORMATION PROCESSING SYSTEMS

RQ3: Do Different Learning Methods Lead to Differences in Performance?

Findings 5: Different solving approaches significantly affect model performance. Models can acquire experience from demonstrations, and feedback further enhances their learning. Moreover, learning capability is not strongly correlated with a model's inherent static ability.

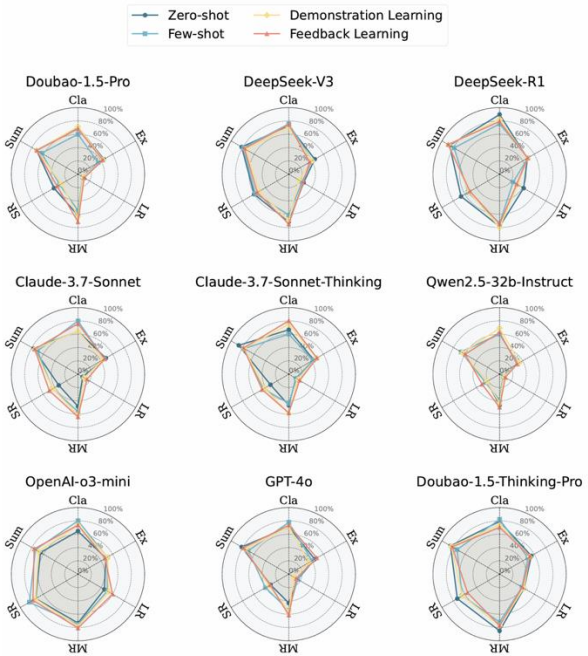


Figure 6 Comparison of overall accuracy across four solving methods, including two parallel methods (i.e., zero-shot and few-shot) and two sequential methods (i.e., demonstration learning and feedback learning). Full task names for the abbreviations can be found in Section 2.1.

Table 4 Comparison of overall accuracy across four solving methods, including two parallel methods (i.e., zero-shot and few-shot) and two sequential methods (i.e., demonstration learning and feedback learning). **Sum** denotes the summarization task. **Cla** denotes the classification task. **Ex** denotes the extraction task. **LR** denotes the logical reasoning task. **MR** denotes the mathematical reasoning task. **SR** denotes the sequential reasoning task. All values are shown as %.

Model	Paradigm	Cla	Ex	LR	MR	SR	Sum	Overall
Non-thinking-based								
Doubao-1.5-Pro	Zero-shot	70.8	**45.0**	10.3	61.7	**41.7**	**71.7**	**31.3**
	Few-shot	60.0	35.1	8.4	54.4	31.6	63.2	25.7
	Demonstration Learning	**71.4**	43.7	8.8	63.9	28.6	71.4	29.0
	Feedback Learning	68.1	42.9	**10.8**	**71.4**	36.1	71.4	31.2
DeepSeek-V3	Zero-shot	72.9	**45.0**	**25.6**	71.7	**60.0**	**81.7**	**43.5**
	Few-shot	**75.6**	40.4	23.0	61.4	57.9	78.9	40.0
	Demonstration Learning	70.3	40.3	17.6	68.9	51.3	74.1	36.4
	Feedback Learning	74.7	36.1	24.9	**74.8**	54.6	76.8	41.5
Claude-3.7-Sonnet	Zero-shot	64.6	**48.3**	8.1	48.3	33.3	**76.7**	28.4
	Few-shot	**80.0**	45.6	10.4	56.1	43.9	70.2	31.1
	Demonstration Learning	64.8	42.9	10.8	61.3	43.7	75.0	31.1
	Feedback Learning	75.8	46.2	**15.5**	**63.9**	**49.6**	75.0	**35.6**
Qwen2.5-32b-Instruct	Zero-shot	60.4	31.7	6.9	46.7	21.7	**66.7**	23.8
	Few-shot	60.0	31.6	6.4	38.6	24.6	**66.7**	22.5
	Demonstration Learning	**69.2**	**34.5**	7.4	43.7	21.8	64.3	24.1
	Feedback Learning	62.6	30.3	**9.8**	**49.6**	**30.3**	59.8	**25.5**
GPT-4o	Zero-shot	70.8	**45.0**	**15.6**	43.3	31.7	**81.7**	32.6
	Few-shot	**77.8**	40.4	15.4	54.4	**40.4**	70.2	**32.9**
	Demonstration Learning	71.4	35.3	8.5	**56.3**	34.5	75.0	28.3
	Feedback Learning	73.6	48.7	12.9	61.3	32.2	78.6	32.7
Thinking-based								
DeepSeek-R1	Zero-shot	**89.6**	48.3	**41.7**	78.3	**66.7**	86.7	**55.7**
	Few-shot	75.6	**49.1**	23.5	73.7	54.4	78.9	41.9
	Demonstration Learning	82.4	48.7	31.4	**79.8**	52.9	88.4	48.2
	Feedback Learning	79.1	48.7	29.6	74.8	51.3	**89.3**	46.4
Claude-3.7-Sonnet-Thinking	Zero-shot	66.7	43.3	12.5	46.7	31.7	**86.7**	31.2
	Few-shot	60.0	42.1	13.4	43.9	**43.9**	77.2	30.6
	Demonstration Learning	74.7	**48.7**	16.1	**59.7**	42.0	77.7	35.2
	Feedback Learning	**80.2**	**48.7**	**18.8**	58.0	46.2	78.6	**37.4**
OpenAI-o3-mini	Zero-shot	64.6	48.3	45.8	73.3	73.3	65.0	54.3
	Few-shot	**80.0**	50.9	50.7	75.4	**84.2**	73.7	60.0
	Demonstration Learning	73.6	**51.3**	53.1	78.2	73.9	67.9	60.0
	Feedback Learning	73.6	47.1	**59.9**	**80.7**	78.2	**75.9**	**64.8**
Doubao-1.5-Thinking-Pro	Zero-shot	79.2	**55.0**	**42.5**	**85.0**	**73.3**	**85.0**	**57.1**
	Few-shot	**82.2**	49.1	38.9	71.9	64.9	73.7	51.4
	Demonstration Learning	73.6	52.1	42.0	77.3	64.7	83.9	54.3
	Feedback Learning	70.3	52.9	39.4	77.3	55.5	82.1	51.6

NeurIPS 2025

FUDAN UNIVERSITY 1905

FUDAN UNIVERSITY NATURAL LANGUAGE PROCESSING NLP

ByteDance | Seed

NEURAL INFORMATION PROCESSING SYSTEMS

RQ3: Do Different Learning Methods Lead to Differences in Performance?

Findings 5: Different solving approaches significantly affect model performance. Models can acquire experience from demonstrations, and feedback further enhances their learning. Moreover, learning capability is not strongly correlated with a model's inherent static ability.

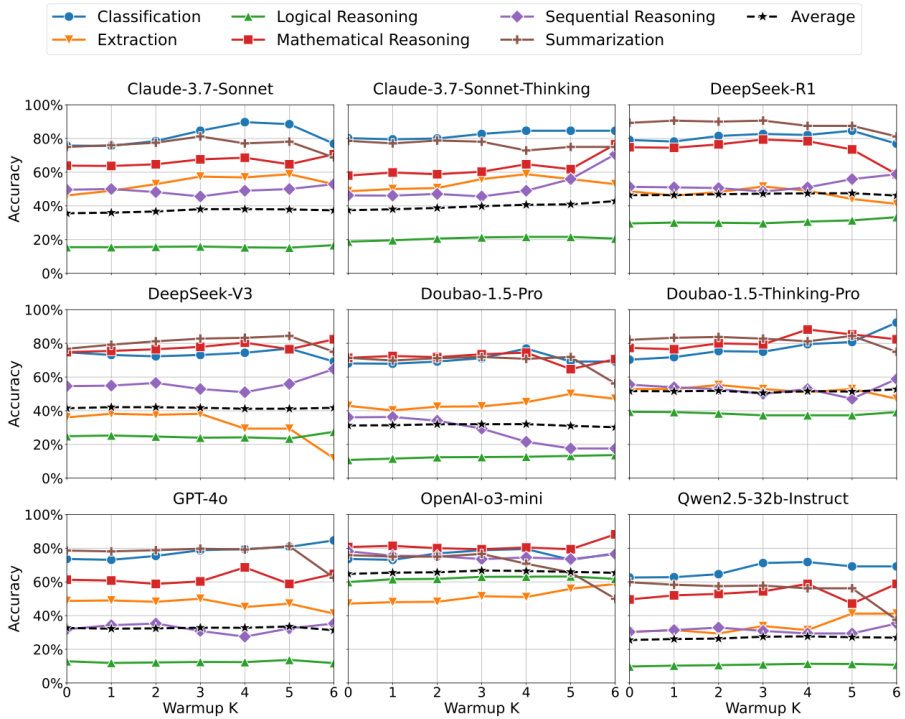


Figure 14 Results of post-warmup accuracy (Acc_{pw}-K) for feedback learning.

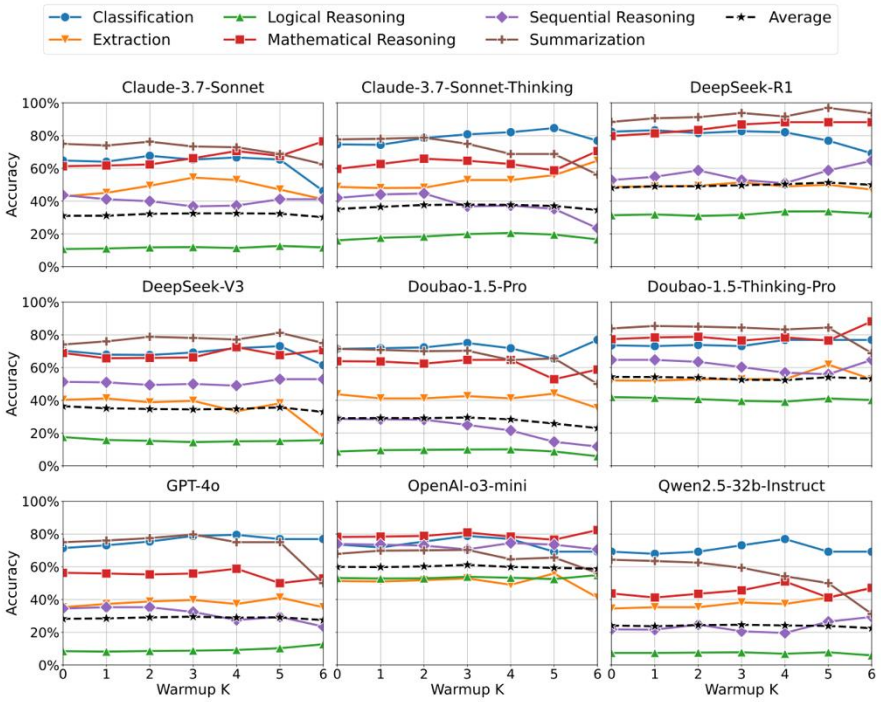


Figure 15 Results of post-warmup accuracy (Acc_{pw}-K) for demonstration learning.

ByteDance | Seed

NEURAL INFORMATION PROCESSING SYSTEMS

RQ3: Do Different Learning Methods Lead to Differences in Performance?

Findings 6: The average position of the first correct solution P_{first} and the average offset of the first learned correct solution P_{offset} vary across models and tasks, providing important insights into model potential. All metrics in EvaLearn capture different aspects of a model's learning ability and efficiency, collectively revealing the model's learning potential.

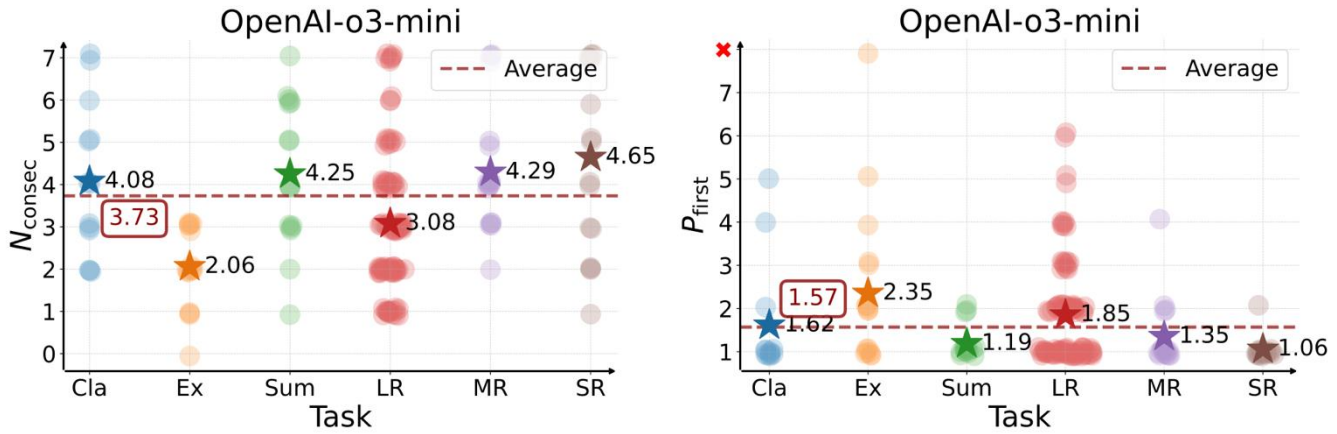


Figure 2 (Left) Average number of consecutive correct solutions (N_{consec}). **(Right)** Average position of the first correct solution (P_{first}). Results are shown for OpenAI-o3-mini, with each node representing a sequence.

ByteDance | Seed

NEURAL INFORMATION PROCESSING SYSTEMS

Conclusion

We present EvaLearn, a novel benchmark that sequentially evaluates the learning capability and efficiency of models within specific tasks.

EvaLearn is equipped with a suite of comprehensive metrics, revealing significant performance differences among frontier models across diverse tasks, including both thinking-based and non-thinking-based models.

Moreover, we find that while some models can effectively leverage teacher-model feedback on previous solutions to enhance learning, others struggle to benefit from such feedback.

EvaLearn offers a new perspective on assessing the potential of large language models (LLMs) and serves as a pioneering step toward dynamic evaluation.

FUDAN UNIVERSITY 1905

FUDAN UNIVERSITY NATURAL LANGUAGE PROCESSING NLP

ByteDance | Seed