MLE-STAR: Machine Learning Engineering Agent via Search and Targeted Refinement

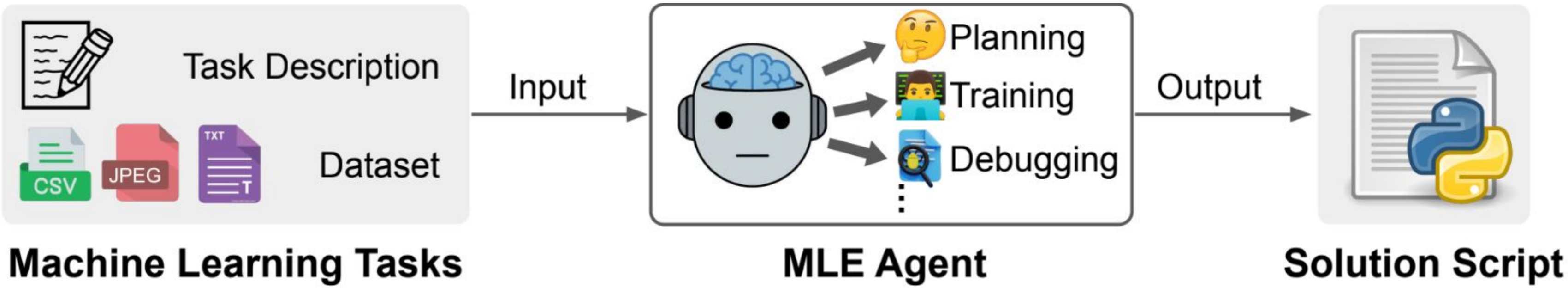
Jaehyun Nam, Jinsung Yoon, Jiefeng Chen, Jinwoo Shin, Sercan O. Arik, Tomas Pfister

NeurIPS 2025

What are Machine Learning Engineering Agents?

Goal: Determining the optimal solution for a given ML problem.

- Input: task descriptions, datasets.
- Output: solution script.
    - Typically, a full python code.
    - Trained models, test metrics, etc.



What are Machine Learning Engineering Agents?

Goal: Determining the optimal solution for a given ML problem.

- Diverse tasks: Classification, Regression, Image denoising, ...
- Diverse modalities: Tabular, Image, Text, Audio, ...
- MLE-STAR is evaluated on:
  - 2 Tabular Classification, 2 Tabular Regression.
  - 9 Image Classification, 1 Image Denoising.
  - 4 Text Classification, 2 Sequence-to-Sequence.
  - 2 Audio Classification.

Motivations.

How can we incorporate state-of-the-art approaches, ensuring scalability?

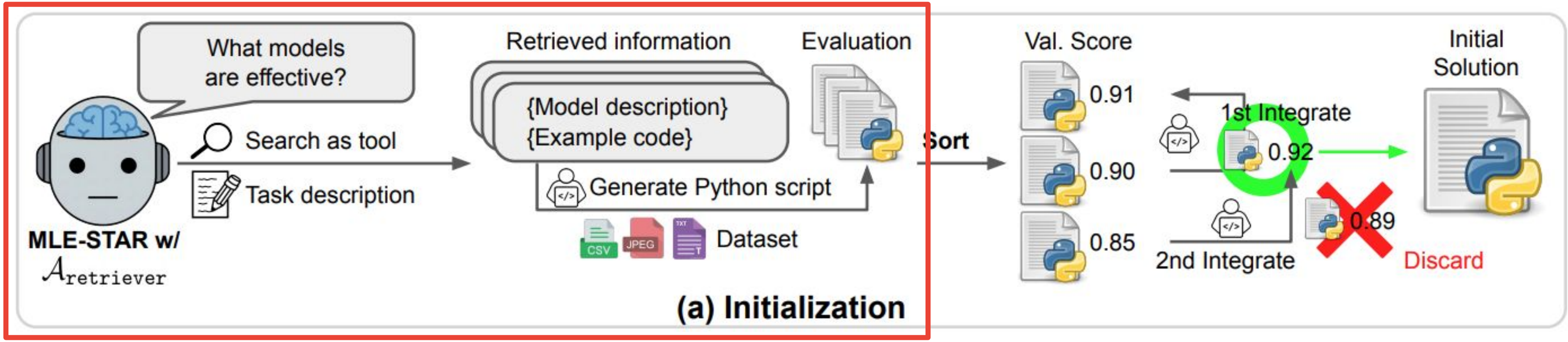
- MLE-STAR utilizes Google Search to retrieve such approaches.

How can we explore different options on specific pipeline extensively?

- E.g., how can we experiment different feature engineering options?
- MLE-STAR extracts a specific code block, and then concentrates on exploring strategies that are targeted to that component.

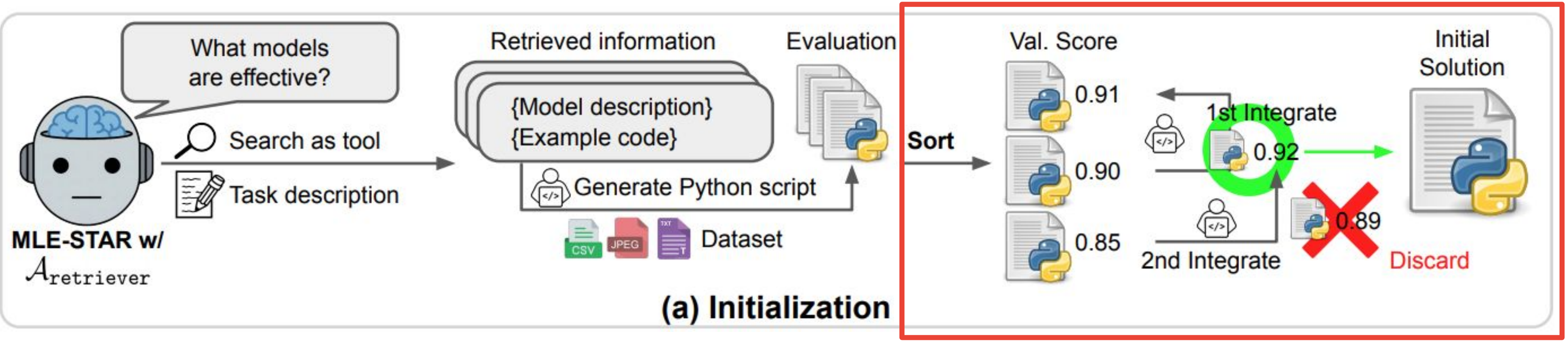
Initialization using web search as a tool.

- Search candidate models.
  - Depending on task description, which contains task type, modalities, ...
  - Retrieved models are then evaluated on the validation metrics.



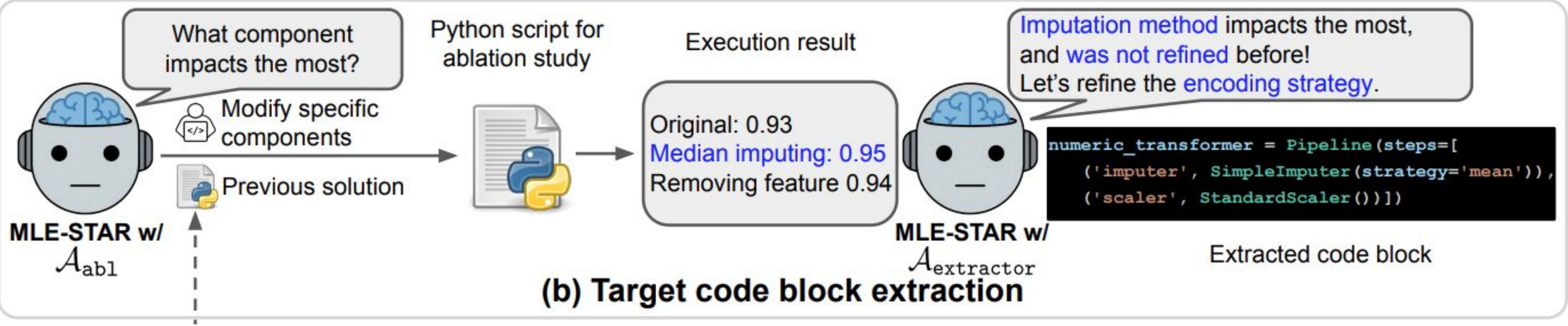
Initialization using web search as a tool.

- Search candidate models.
- Merge retrieved models based on the validation metrics.
  - We first sort in descending order.
  - We sequentially incorporate the candidate models until the validation score no longer improves.



Refining a code block for solution improvement.

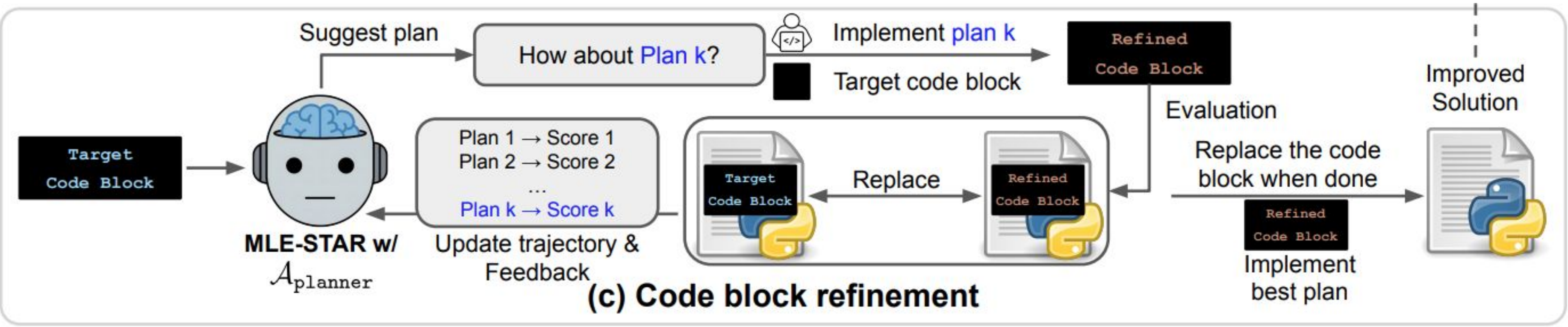
- MLE-STAR identifies specific code blocks to explore specialized strategies.
- But how can we identify the code block that have the greatest impact?
  - MLE-STAR performs an ablation study.
  - MLE-STAR generate a code for ablation study, which creates variations of the current solution by modifying or disabling specific components.



(b) Target code block extraction

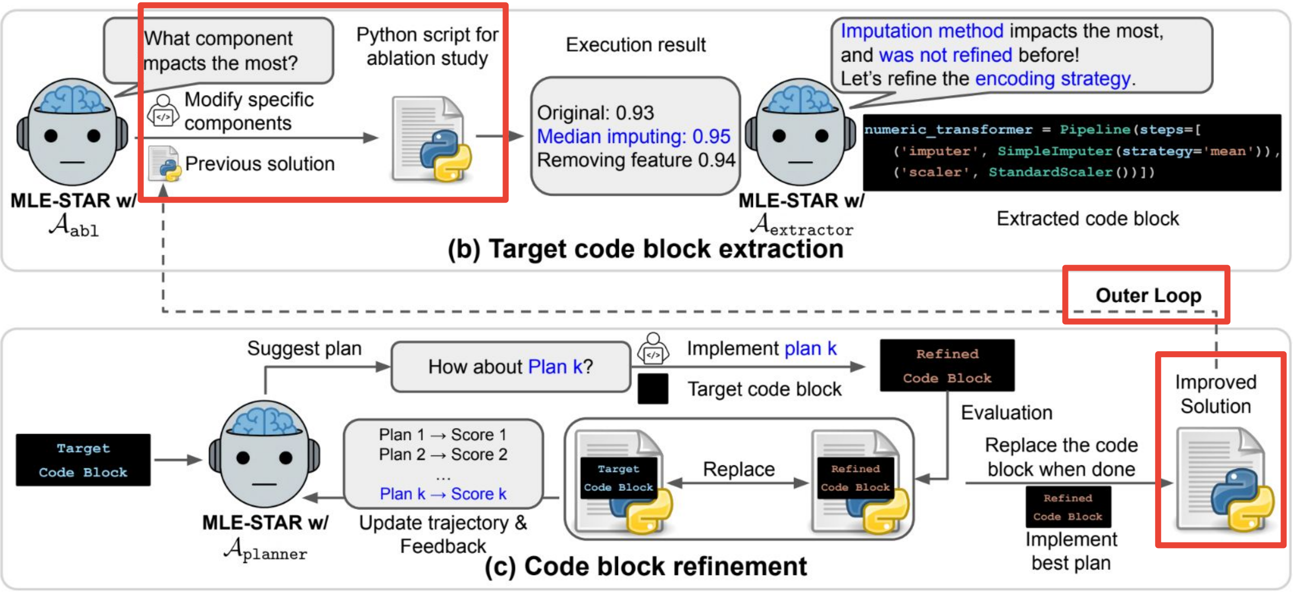
Refining a code block for solution improvement.

- MLE-STAR iteratively explores refinement strategies on the target code block.
  - Focus on the selected code block and refine it with diverse ways.
  - Here, the previous experiment results are used as a feedback.



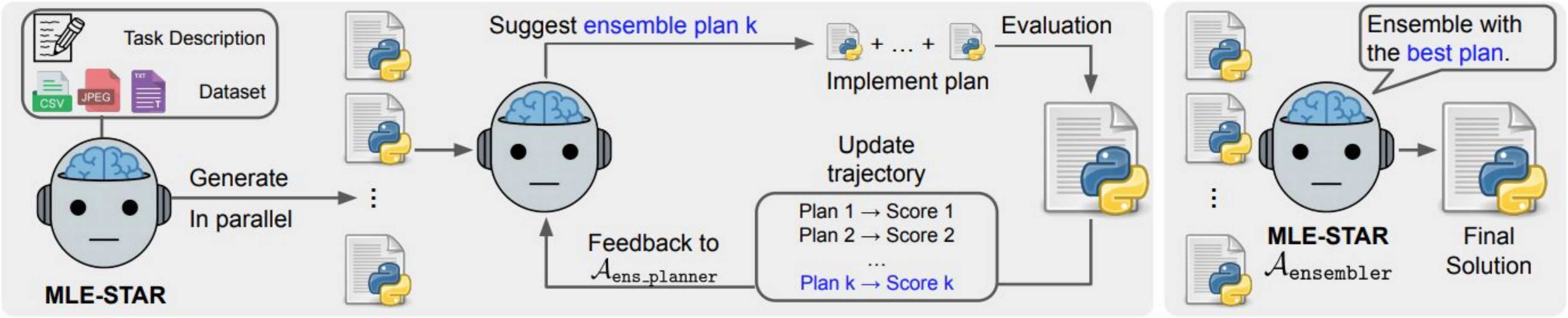
Refining a code block for solution improvement.

- Target code block is also selected repeatedly.
  - After the code block refinement, MLE-STAR performs the ablation study on the improved solution.



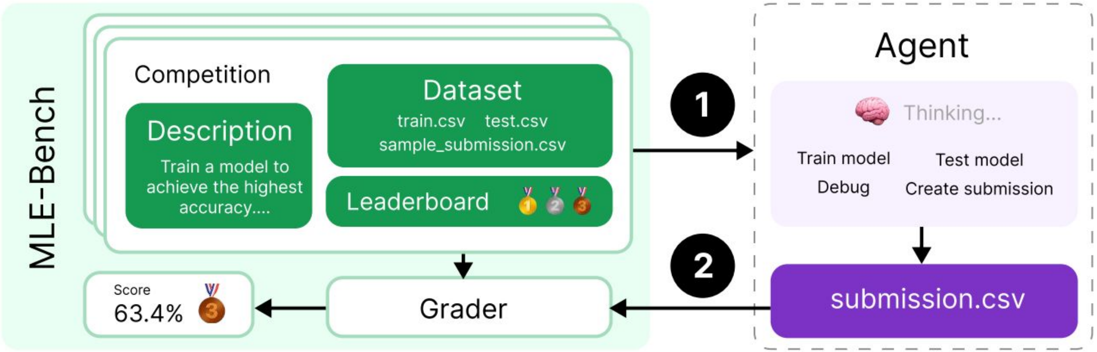
Further improvement by exploring ensemble strategies.

- Alike model ensembling, suboptimal solutions might contain complementary strengths, and combining multiple solutions could lead to superior performance.
- MLE-STAR automatically discovers effective strategies for ensembling.
    - Using parallely generated training codes, we ensemble those solutions.



Main experiment: MLE-Bench.

- A benchmark of 75 offline Kaggle competitions.
  - We use 22 low complexity competitions.
- Evaluation metric: Medals (like Kaggle competition).



	0-99 Teams	100-249 Teams	250-999 Teams	1000+ Teams
Bronze	Top 40%	Top 40%	Top 100	Top 10%
Silver	Top 20%	Top 20%	Top 50	Top 5%
Gold	Top 10%	Top 10	Top 10 + 0.2%*	Top 10 + 0.2%*

Main experiment: MLE-Bench.

- MLE-STAR achieves significant performance gain over the SOTA baseline.
  - 60+% any medals / 80+% above median submissions.

Model	Made Submission (%)	Valid Submission (%)	Above Median (%)	Bronze (%)	Silver (%)	Gold (%)	Any Medal (%)
MLE-STAR (Ours)							
gemini-2.5-pro	**100.0**±0.0	**100.0**±0.0	**83.3**±4.6	6.1±3.0	**21.2**±5.1	**36.4**±6.0	**63.6**±6.0
gemini-2.0-flash	95.5±2.6	95.5±2.6	63.6±6.0	**9.1**±3.6	4.5±2.6	30.3±5.7	43.9±6.2
AIDE (Jiang et al., 2025)							
gemini-2.0-flash	87.9±4.0	78.8±5.0	39.4±6.0	4.5±2.6	9.1±3.5	12.1±4.0	25.8±5.4
o1-preview	99.7±0.3	90.3±1.6	58.2±2.6	4.8±1.1	11.1±1.7	20.7±2.2	36.6±2.6
gpt-4o	82.1±1.4	65.7±1.7	29.9±1.6	3.4±0.6	5.8±0.8	9.3±1.0	18.6±1.4
llama-3.1-405b-instruct	72.7±5.5	51.5±6.2	18.2±4.7	0.0±0.0	4.5±2.6	6.1±2.9	10.6±3.8
claude-3-5-sonnet	81.8±4.7	66.7±5.8	33.3±5.8	3.0±2.1	6.1±2.9	10.6±3.8	19.7±4.9
MLAB (Huang et al., 2024a)							
gpt-4o	84.8±4.4	63.6±5.9	7.6±3.3	3.0±2.1	1.5±1.5	1.5±1.5	6.1±2.9
OpenHands (Wang et al., 2024)							
gpt-4o	81.8±4.7	71.2±5.6	16.7±4.6	3.0±2.1	3.0±2.1	6.1±2.9	12.1±4.0

Effectiveness of proposed ensemble methods.

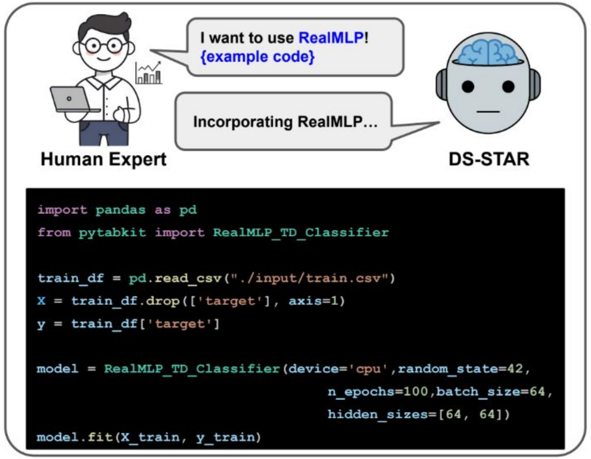
- MLE-STAR shows a performance improvement even without additional ensemble.
- Simple strategies, such as selecting the solution with the best validation score of averaging final submissions, also offer benefits.
  - However, our proposed method shows stronger effectiveness.

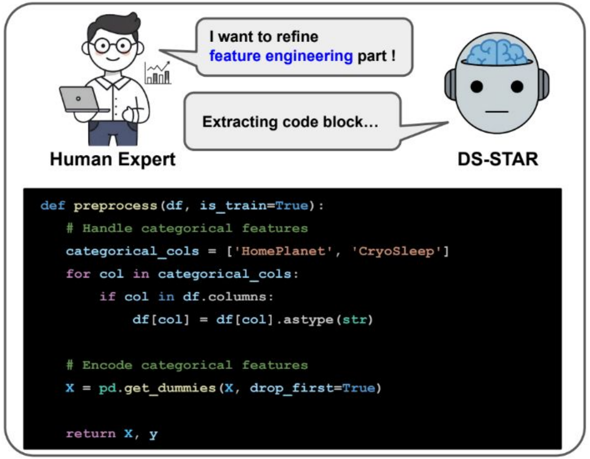
Ensemble strategy	Made Submission (%)	Valid Submission (%)	Above Median (%)	Bronze (%)	Silver (%)	Gold (%)	Any Medal (%)
AIDE [12]							
None	87.9±4.0	78.8±5.0	39.4±6.0	4.5±2.6	9.1±3.5	12.1±4.0	25.8±5.4
MLE-STAR (Ours)							
None	**95.5**±2.6	**95.5**±2.6	57.6±6.1	7.6±3.3	4.5±2.6	25.8±5.4	37.9±6.0
Best-of-N	**95.5**±2.6	**95.5**±2.6	62.1±6.0	6.1±3.0	7.6±3.3	28.8±5.6	42.4±6.1
Average ensemble	**95.5**±2.6	**95.5**±2.6	60.6±6.1	6.1±3.0	**12.1**±4.0	25.8±9.4	**43.9**±6.2
Ours	**95.5**±2.6	**95.5**±2.6	**63.6**±6.0	**9.1**±3.6	4.5±2.6	**30.3**±5.7	**43.9**±6.2

Human intervention.

MLE-STAR adopts even more recent model with minimal human intervention.

- E.g., by manually adding a model description for RealMLP, MLE-STAR successfully integrates its training into the framework.
- E.g., users can also specify the target code blocks by replacing the ablation summary with manually written instructions.





Key takeaways.

MLE-STAR is an effective and robust ML Engineering Agent that:

- Uses web search as a tool to retrieve task-relevant effective approaches.
- Performs ablation study to extract the impactful code block.
- Refines a target code block by exploring component-specific strategies.