Appendix for the article *AI leaderboards are no longer useful. It's time to switch to Pareto curves.*

Link to the post that introduces these results.
Link to reproduction materials.

Implementation details

We used gpt-3.5-turbo-0125 for GPT-3.5 implementations and gpt-4-turbo-2024-04-09 for GPT -4 implementations. We describe all implementations in detail below.

In addition to our analysis in the main text, we also conducted robustness checks with the June 2023 versions of GPT-3.5 and GPT-4 and found substantially similar results.

We include four figures below: (i) The results of our HumanEval analysis along with error bars for accuracy and cost; (ii) The results of our HumanEval analysis with the y-axis from 0 to 1 (in other figures, the y-axis is clipped between 0.7 and 1 for clarity); (iii) the results of our robustness checks with the June 2023 versions of GPT-3.5 and GPT-4; (iv) results of the time vs. accuracy Pareto curve.

In the figure reporting our results in the main text, we include the convex hull of points on the Pareto frontier because, given any two agents on the frontier, we can always choose a strategy that picks agent 1 with probability *p* and agent 2 with probability *1-p* and to achieve any tradeoff represented by points on the convex hull.

GPT-3.5 and GPT-4. We implement the model baselines using the simple (zero shot; without agent architecture) strategy provided with the LDB paper. This includes a text prompt and the example tests accompanying the HumanEval coding problem as inputs to the model.

LDB. The LDB agent uses two language models: one for generating code and another for debugging. In all plots and throughout this post, we use the nomenclature "LDB (*Generator*, *Debugger*)" to specify which models were used. If the same model served both functions, we list it only once within parentheses. We kept all parameters as specified in the code accompanying the original paper. In particular, this means that the maximum number of iterations is set to 10 and the temperature to zero.

LATS. Based on correspondence with the authors, we set the maximum number of iterations to 8, the expansion factor to 3, and the temperature values for generating the function implementations to 0.8. The temperature for generating self-reflections and the internal unit tests was set to 0.2. The maximum number of internal test cases was set to 6 for runs with GPT-3.5 and 4 for runs using GPT-4. The difference in the number of internal test cases for GPT-3.5 and GPT-4 was not presented in the paper; we learned of this based on our correspondence with the authors after we shared an early draft of the blog post with them.

Reflexion. We left all parameters unchanged from the ones provided in the original repository, setting the maximum number of iterations to 2, expansion factor to 3, and temperature to zero for generating function implementations. The temperature used for generating the internal tests and self-reflections is 0.2.

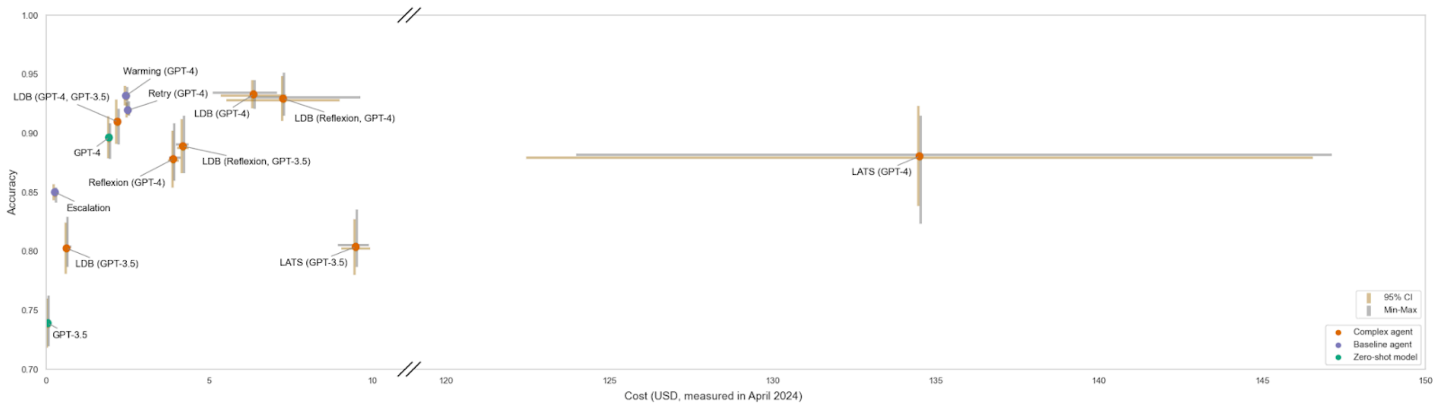
Retry. This baseline uses the *simple* strategy implemented in the code accompanying the LDB agent for zero-shot evaluations of language models (i.e., there is no agent architecture). We used this strategy to repeatedly prompt the same language model, keeping all parameters equal across retrials, as long as the code outputted by the model failed at least one of the example tests. If at any point a solution passes the tests given in the HumanEval problem description, we evaluate this as the final solution of the agent for this problem. We repeated this procedure for up to 5 trials and stopped early if the code passed all the given tests. We set the temperature to zero.

Warming. For the warming baseline, we modify the retry baseline by gradually increasing the temperature parameter across successive trials. Initially, the temperature was set at zero, mirroring the retry baseline. For the second and third trials, we raised the temperature to 0.3, and for the final two trials, we increased it further to 0.5. If at any point a solution passes the tests given in the HumanEval problem description, we evaluate this as the final solution of the agent for this problem.

Escalation. We modify the *simple* strategy but switch the underlying model to a more expensive one if a proposed solution fails at least one of the example tests. We progressively escalated unsolved problems up a model chain of increasing cost (llama-3-8b-chat-hf, gpt-3.5-turbo-0125, llama-3-70b-chat-hf, gpt-4-turbo-2024-04-09). All other parameters are kept constant across trials – in particular, temperature is set to zero. If at any point a solution passes the tests given in the HumanEval problem description, we evaluate this as the final solution of the agent for this problem. This leads to slightly lower accuracy compared to GPT-4, since some solutions from cheaper models might pass the example tests but fail one of the evaluation tests.

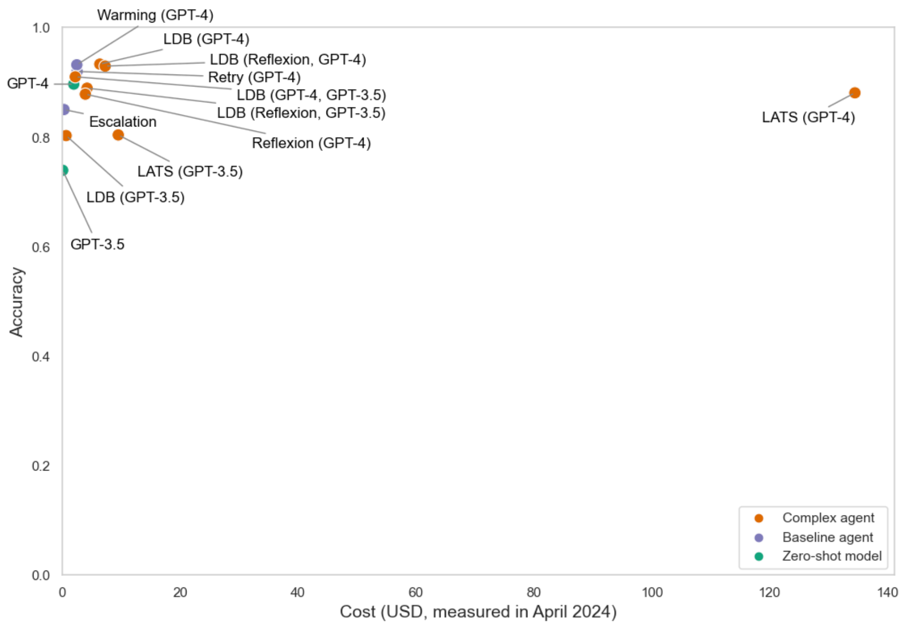
Additional results

Figure 1: Error bars for our HumanEval analysis in the main post



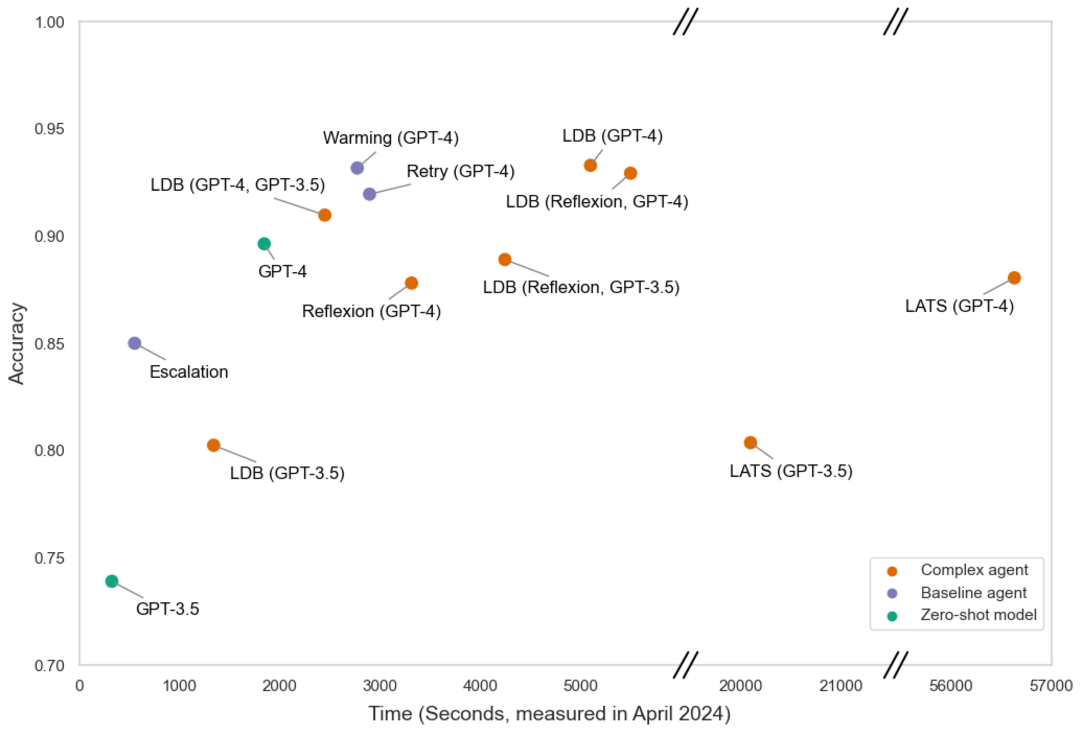
The figure shows accuracy vs. API cost for the HumanEval results zoomed in on the y-axis (0.7-1). The error bars represent 95% confidence intervals (left/lower; brown) and the minimum and maximum values (right/upper; gray) of accuracy and total cost across 5 runs. To calculate the 95% confidence intervals, we used the Student's t distribution given that we only have five runs per agent.

Figure 2: HumanEval results with a complete x- and y-axis.



The figure shows accuracy vs. API costs with a complete x- and y-axis. This plot showcases the wide range of costs associated with different approaches, especially when considering LATS.

Figure 3: Accuracy vs. inference time curves for HumanEval



This figure shows the accuracy vs. inference time results on a linear x-axis scale, with the y-axis clipped to 0.7-1 for clarity. Time measurements refer to the mean of the sum of inference times across all API calls made by the agent across the five runs.

Figure 4: Robustness checks with June 2023 versions of GPT-4 models

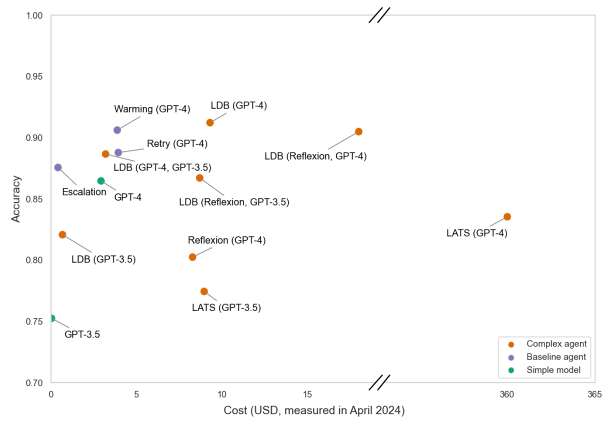


Figure 4a: Accuracy vs. inference cost.

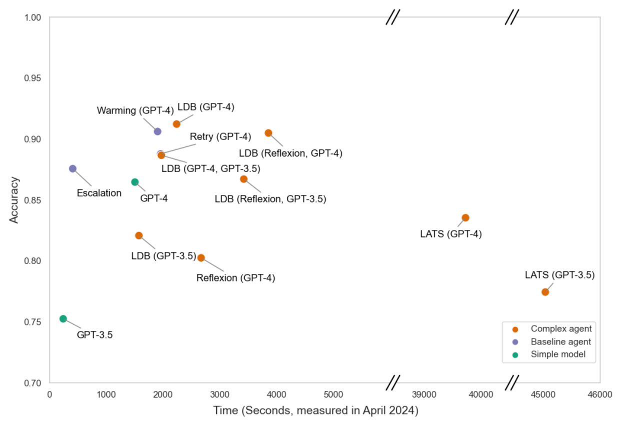


Figure 4b: Accuracy vs. inference time

One concern with our analysis is that we use the latest versions of OpenAI's models, since later versions of GPT-3.5 and GPT-4 might have more scope for contamination. To address this, we conduct additional robustness checks with the June 2023 versions of GPT-3.5 and GPT-4. We find substantially similar results for this version: complex agents are no better than our simple agent baselines while cost orders of magnitude more in some cases. Note that the June 2023 version of GPT-4 is much more expensive than the April 2024 version, leading to the big difference in inference cost.

For the LATS agents, there are some significant outliers across tasks for both, LATS (GPT-4) and LATS (GPT-3.5), with some tasks requiring more than 2 hours to complete for GPT-3.5. Overall, there were more extreme outliers for LATS (GPT-3.5) than LATS (GPT-4). For the same reason, we had to exclude one task (i.e., HumanEval/83) from the analysis for LATS (GPT-3.5), which did not stop running even after 5 hours. We marked this task as incorrect and excluded the time and cost from the results shown above. HumanEval/83 was one of the tasks excluded from the subset of HumanEval that the authors evaluated the LATS agent on.