

# What Does It Take to Be a Good AI Research Agent?

## Studying the Role of Ideation Diversity

Alexis Audran-Reiss<sup>1</sup>, Jordi Armengol-Estepé<sup>1</sup>, Karen Hambardzumyan<sup>1,2</sup>, Amar Budhiraja<sup>1</sup>, Martin Josifoski<sup>1</sup>, Edan Toledo<sup>1,2</sup>, Rishi Hazra<sup>1</sup>, Despoina Magka<sup>1</sup>, Michael Shvartsman<sup>1</sup>, Parth Pathak<sup>1</sup>, Justine T Kao<sup>1</sup>, Lucia Cipolina-Kun<sup>1</sup>, Bhavul Gauri<sup>1</sup>, Jean-Christophe Gagnon-Audet<sup>1</sup>, Emanuel Tewolde<sup>1</sup>, Jenny Zhang<sup>3,4</sup>, Taco Cohen<sup>1</sup>, Yossi Adi<sup>1</sup>, Tatiana Shavrina<sup>1</sup>, Yoram Bachrach<sup>1</sup>

<sup>1</sup>FAIR at Meta, <sup>2</sup>University College London, <sup>3</sup>Meta SuperIntelligence Labs, <sup>4</sup>University of British Columbia

AI research agents offer the promise to accelerate scientific progress by automating the design, implementation, and training of machine learning models. However, the field is still in its infancy, and the key factors driving the success or failure of agent trajectories are not fully understood. We examine the role that ideation diversity plays in agent performance. First, we analyse agent trajectories on MLE-bench, a well-known benchmark to evaluate AI research agents, across different models and agent scaffolds. Our analysis reveals that different models and agent scaffolds yield varying degrees of ideation diversity, and that higher-performing agents tend to have increased ideation diversity. Further, we run a controlled experiment where we modify the degree of ideation diversity, demonstrating that higher ideation diversity results in stronger performance. Finally, we strengthen our results by examining additional evaluation metrics beyond the standard medal-based scoring of MLE-bench, showing that our findings still hold across other agent performance metrics.

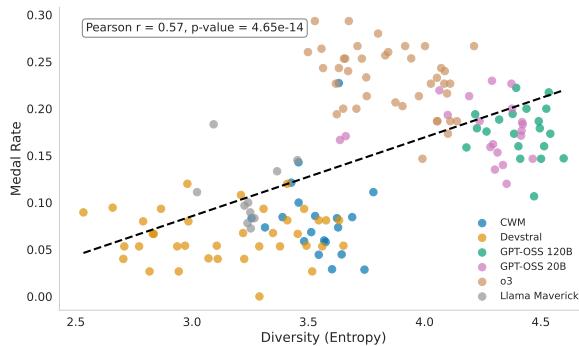
**Date:** November 20, 2025

**Correspondence:** Alexis Audran-Reiss at [a.audranreiss@meta.com](mailto:a.audranreiss@meta.com), Jordi Armengol-Estepé at [jordiae@meta.com](mailto:jordiae@meta.com)



## 1 Introduction

The rapid advancement of Large Language Model-based (Brown et al., 2020) agents equipped with tools (Schick et al., 2023) has sparked interest in the quest to develop research agents, in areas as challenging as chemistry (Boiko et al., 2023) or biology (Swanson et al., 2025).



**Figure 1** Ideation diversity correlates with performance in MLE-bench: Our analysis shows that ideation diversity correlates with the agent’s trajectory success in MLE-bench. To confirm this relation, we later intervene on ideation diversity in a controlled experiment, in Section 4.2.

These research agents constitute an emerging paradigm in computational scientific discovery characterized by end-to-end autonomous systems capable of conducting independent research.

In particular, recent work on autonomous AI research agents (Shen et al., 2023; Huang et al., 2024; Toledo et al., 2025; Zhao et al., 2025) improves upon automated machine learning engineering tools (Feurer et al., 2022) by mirroring the cognitive processes of human researchers through a structured research pipeline: idea generation and hypothesis setup, experimental design and implementation, empirical validation, and iterative refinement. Recent advances have achieved notable milestones such as creation of the first fully autonomous AI-generated research paper accepted through peer review (Yamada et al., 2025).

Despite the potential of these recent breakthroughs in automating AI science, the field is still in its infancy and little is understood about the factors driving their successes and failures. Error analysis is substantially more complicated than in classic machine learning setups, due to the presence of long multi-step trajectories often guided by heuristic-based search algorithms (Toledo et al., 2025) and leveraging tool use, which requires complex evaluation frameworks. Moreover, obtaining large-enough samples to perform meaningful analysis and ablate design choices can be computationally prohibitive.

This paper starts from the postulate that *ideation diversity* is a key bottleneck in AI research agents' performance. To study this hypothesis, we face two key challenges: analyzing complex agentic trajectories at scale, and measuring and controlling ideation diversity.

We perform a first-of-its-kind, large-scale study of AI research agents' trajectories in MLE-bench (Chan et al., 2025), a well-known benchmark of Kaggle machine learning tasks. We study 6 different LLM backbones equipped with 2 different agentic frameworks (or *scaffolds*) on the 75 machine learning tasks available in MLE-bench across 10 to 20 random seeds, yielding a total of 11,000 trajectories. This corresponds to roughly 1,200,000 individual nodes in the agent scaffold search, for a total of 264,000 GPU hours.

To measure ideation diversity, we propose calculating Shannon entropy (Shannon, 1948) on the distribution of model architectures that the agent plans to implement in the ideation phase. Figure 1 shows the correlation between ideation diversity and the performance in MLE-bench using our generated trajectory bank, hinting at a relation between the two.

To confirm the diversity hypothesis, we perform a controlled experiment where we remove mechanisms yielding highly diverse solutions by amending the prompt.<sup>1</sup> We run the controlled experiment on a subset of MLE-bench, studying two diversity setups (control and ablated) using two agentic frameworks on 22 machine learning tasks across 10 seeds. We show that when ablating ideation diversity, the agents' performance decreases. Finally, we strengthen this finding by examining additional evaluation metrics on MLE-bench aside from the standard score based on the Kaggle medal system.

## 1.1 Contributions

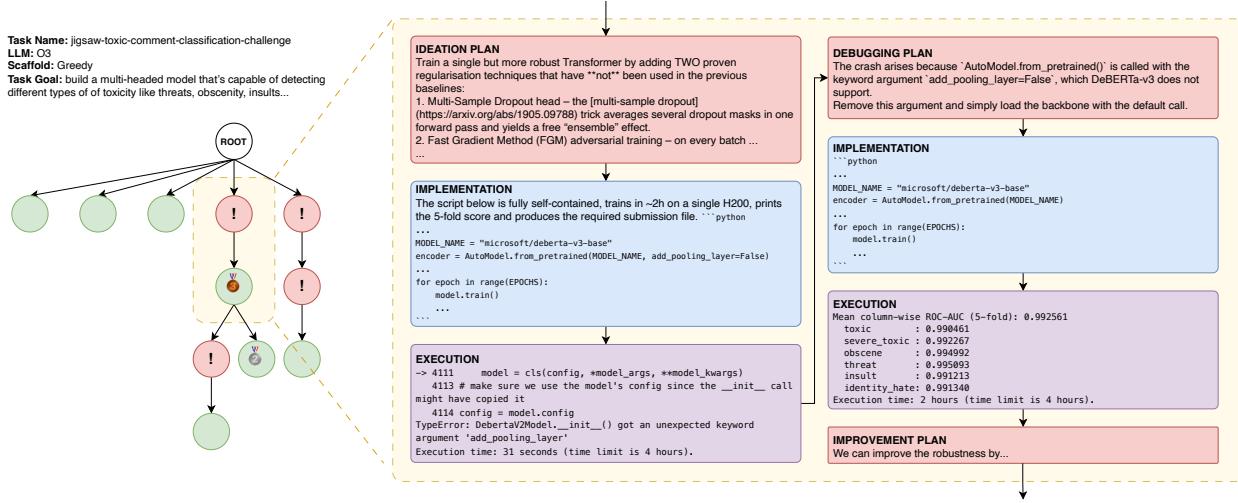
In summary, the contributions of this paper are as follows:

- We propose methods to **quantify and control the agent's ideation diversity**.
- We perform a **first-of-its-kind, large-scale analysis on agentic trajectories**. We study a total of 11,000 AI research agents' trajectories across multiple agentic frameworks, LLM backbones, and machine learning tasks.
- We show that the choice of **agentic scaffold significantly influences ideation diversity**. Our results further reveal a significant correlation between ideation diversity and agent performance on MLE-bench tasks.
- Through a **controlled experimental design**, we establish a causal relationship, showing that **increasing ideation diversity leads to improved performance** on MLE-bench tasks.
- We confirm that these findings are **robust when evaluated with alternative performance metrics**.

<sup>1</sup>Our main results in Section 4.2 are based on decreasing ideation diversity through the prompt shown to the agent. In the Appendix A, we provide additional results where we control diversity via the sampling temperature parameter.

## 2 Research Agents and MLE-bench

*Agents* Broadly, Wooldridge and Jennings (1995) define an agent as a *computer system that is situated in some environment and that is capable of autonomous action in this environment in order to meet its design objectives*. In the particular context of recent work on research agents based on generative AI and LLMs (Schick et al., 2023; Boiko et al., 2023; Shen et al., 2023; Huang et al., 2024; Swanson et al., 2025; Toledo et al., 2025; Zhao et al., 2025), we specifically refer to agent systems implemented using two main components: 1. a model backbone (typically, an LLM), which processes observations from the environment as prompts and emits text-based actions, and 2. an outer loop making use of the model backbone to interface with the environment. This outer loop orchestrating the LLM actions is usually referred to as *agentic frameworks* or *agentic scaffolds* in the literature (Wu et al., 2022). The environments for AI research agents allow the agent to create and run code among other *tools* (Schick et al., 2023).



**Figure 2** An example flow of an AI research agent attempting an MLE-bench task. The goal of the task is to build a multi-headed model to classify different types of toxicity threats. The agent first tries the idea to finetune a model end to end, but the code fails and the agent fixes the bug. After analysis of this approach, the agent continues improving the solution, producing more nodes.

*MLE-Bench* (Chan et al., 2025) (Machine Learning Engineering Benchmark) is a well-known evaluation framework designed to assess autonomous AI agents' capabilities in solving real-world machine learning problems. The benchmark is constructed from 75 tasks sourced from Kaggle competition datasets, providing a diverse collection of machine learning challenges that span computer vision, natural language processing, time series forecasting, tabular data analysis, and multimodal learning domains.

Each MLE-bench task comprises standardized components including problem documentation, training datasets, held-out test sets, sample submission formats, and automated evaluation protocols. The evaluation methodology follows Kaggle's competitive framework, utilizing problem-specific metrics such as accuracy, F1-score, or RMSE, with performance assessed through leaderboard ranking systems.

AI research agents interact with MLE-bench through a standardized API that mirrors real-world ML development workflows. Agents must autonomously perform the complete machine learning pipeline: data exploration, feature engineering, model selection, hyperparameter optimization, and submission generation within computational constraints. The benchmark employs stratified sampling and cross-validation methodologies to ensure robust performance assessment, with final rankings determined through holdout test set evaluation to prevent overfitting to validation metrics.

Given the same dataset, there are multiple ways of defining how to interface with the benchmark, mediated by the agentic scaffold of choice. Recent work (Toledo et al., 2025) proposed a tree-based agentic scaffold to tackle MLE-bench. In Section 3.1.3, we describe the scaffolds studied in this work with more detail.

Figure 2 shows an example of a tree-search-based AI research agent attempting the *Jigsaw Toxic Comment Classification Challenge* MLE-bench task. The AI research agent would execute the following computational workflow: idea generation (e.g., leveraging pre-trained convolutional neural network (LeCun and Bengio, 1998) features with linear classifiers), hypothesis setup (e.g., establishing baselines using CIFAR-100 (Krizhevsky et al., 2009) embeddings with logistic regression), and implementation and experimentation (e.g., tuning learning rate for logistics regression). The next step would be to analyze results (e.g. look at confusion matrix) and finally make a submission to the grader. Based on the leaderboard rank and experiment analysis, the agent would then propose the next set of ideas and hypotheses (e.g. changing classifier from logistic regression to random forest or tuning a CNN end-to-end).

These idea generation, implementation, experimentation and submission steps are iterated upon to improve the agent’s leaderboard rank. This paper focuses on the idea generation step.

## 3 Methods

In this section, we briefly introduce the methodology (including data, metrics, agentic orchestrations or scaffolds, and LLM backbones) used in both our data analysis, in Section 4.1, and the controlled experiment, in Section 4.2.

### 3.1 General Setup

#### 3.1.1 Dataset

For the trajectory analysis, we use agent trajectories on MLE-bench. For the controlled experiment, we focus on MLE-bench lite, a curated subset of 22 tasks selected from the full benchmark.

#### 3.1.2 Metrics

In line with the benchmark guidelines, for both our data analysis and controlled experiment, we assess each agent’s performance using the Medal Success Rate (henceforth referred to as medal rate). Specifically, for each task, agents earn a bronze, silver, or gold medal according to task-specific percentile thresholds. We report the percentage of attempts in which an agent secures a medal. Later, in Section 4.3, we incorporate additional metrics.

#### 3.1.3 Agentic Scaffolds

Following recent work (Toledo et al., 2025), we formalize AI research agents as search algorithms composed of a search policy, used to navigate the space of candidate solutions to a task, and a set of operators, which modify existing solutions to generate new candidate solutions.

Enabled by this formalization, we study a range of agentic structures, specifically focusing on: (1) AIDE (Jiang et al., 2025a), an LLM-driven agent that approaches problem-solving as a tree-search over the domain of Python solutions, utilizing a Greedy policy. (2) AIRA<sub>GREEDY</sub> (Toledo et al., 2025), another greedy tree-based search policy, with a different design for operators, memory scope, and prompts, and (3) AIRA<sub>MCTS</sub> (Toledo et al., 2025), utilizing Monte Carlo Tree Search (MCTS (Coulom, 2006; Kocsis and Szepesvári, 2006; Browne et al., 2012)) for its search policy, in contrast to its greedy counterparts.

In all three agentic scaffolds, the process results in trees where each node represents a Python code solution, created by one of the following operators: 1. **Draft**, which generates the initial population of solutions; 2. **Debug**, which identifies and corrects errors within a given node; and 3. **Improve**, which enhances the solution of a given node to increase its performance according to evaluation criteria.

Additionally, the memory configuration dictates how each operator is selectively provided with previously produced artifacts, with well-scoped memory preventing issues such as context overload, mode collapse, and debug loops.

### 3.1.4 LLM Backbones

For the data analysis, we use the following LLM backbones for the agents studied: o3 ([Jaech et al., 2024](#)), gpt-oss ([OpenAI, 2025](#)) (20B and 120B), Llama Maverick ([Team, 2025b](#)), Devstral ([Team, 2025a](#)) and CWM ([FAIR CodeGen Team, 2025](#)). Those represent different model sizes and architectures.

We conduct the controlled experiment discussed in Section 3.3 with the full-sized DeepSeek R1 model ([DeepSeek-AI et al., 2025](#)). All of the LLMs above use a 128K-token context window to ensure input coverage without truncation.

## 3.2 Measuring Ideation Diversity

Diversity can manifest in many aspects of machine learning engineering, such as data preprocessing, feature engineering, model development, and validation. In this analysis, our focus is limited to examining the diversity of machine learning models trained by agents.

All AI research agents in our study begin their exploration by generating at maximum five initial ideas to solve the task at hand (exactly five for greedy searches, and up to five for MCTS), using the DRAFT operator. To measure ideation diversity, we compare agents by extracting two pieces of information from these five initial ideas. First, we extract the high-level ML approach or architecture used by our agent (for example CNN ([LeCun and Bengio, 1998](#)), Transformer ([Vaswani et al., 2017](#)), Decision Trees); and second, we also extract the specific model employed by the agent, with variants grouped together (e.g., EfficientNet-B4 is grouped as EfficientNet ([Tan and Le, 2019](#))).

We study whether the design of the agent has a significant impact on the diversity of ideas generated by comparing the distribution of models used by AI research agents.

To quantify diversity, we leverage the model architectures that the agent intends to train. From the distribution of model architectures, we compute the Shannon entropy (in base 2), quantifying the average uncertainty (and therefore diversity) of the model architecture used by the AI research agent.

## 3.3 Controlling Ideation Diversity

As part of our experimentation, we control the level of diversity using the system prompt in the LLM behind the agent. We compare two levels of diversity, the baseline agents and the agents with ablated diversity.

### 3.3.1 Baseline agents

We run baseline (or control) agents (with 2 scaffolds, AIRA<sub>Greedy</sub>, and AIRA<sub>MCTS</sub>), which, by default, are equipped with three mechanisms to enhance diversity. 1. *Sibling memory*, which provides to a new draft node the memory of its siblings, by including in the context descriptions of the solutions devised by the sibling nodes. 2. *Prompt-adaptive complexity*, which is a dynamic complexity cue within the system prompt aiming to guide the complexity of artifacts generated by the agents. For the first initial idea, we ask the agent to come up with an idea of minimal complexity. For the next two initial ideas, the system prompt asks for moderate complexity, and advanced complexity for the last two initial ideas. 3. *Mention of diversity in the system prompt*, asking the base model to come up with different aspects of the solution every time.

### 3.3.2 Agents with ablated diversity

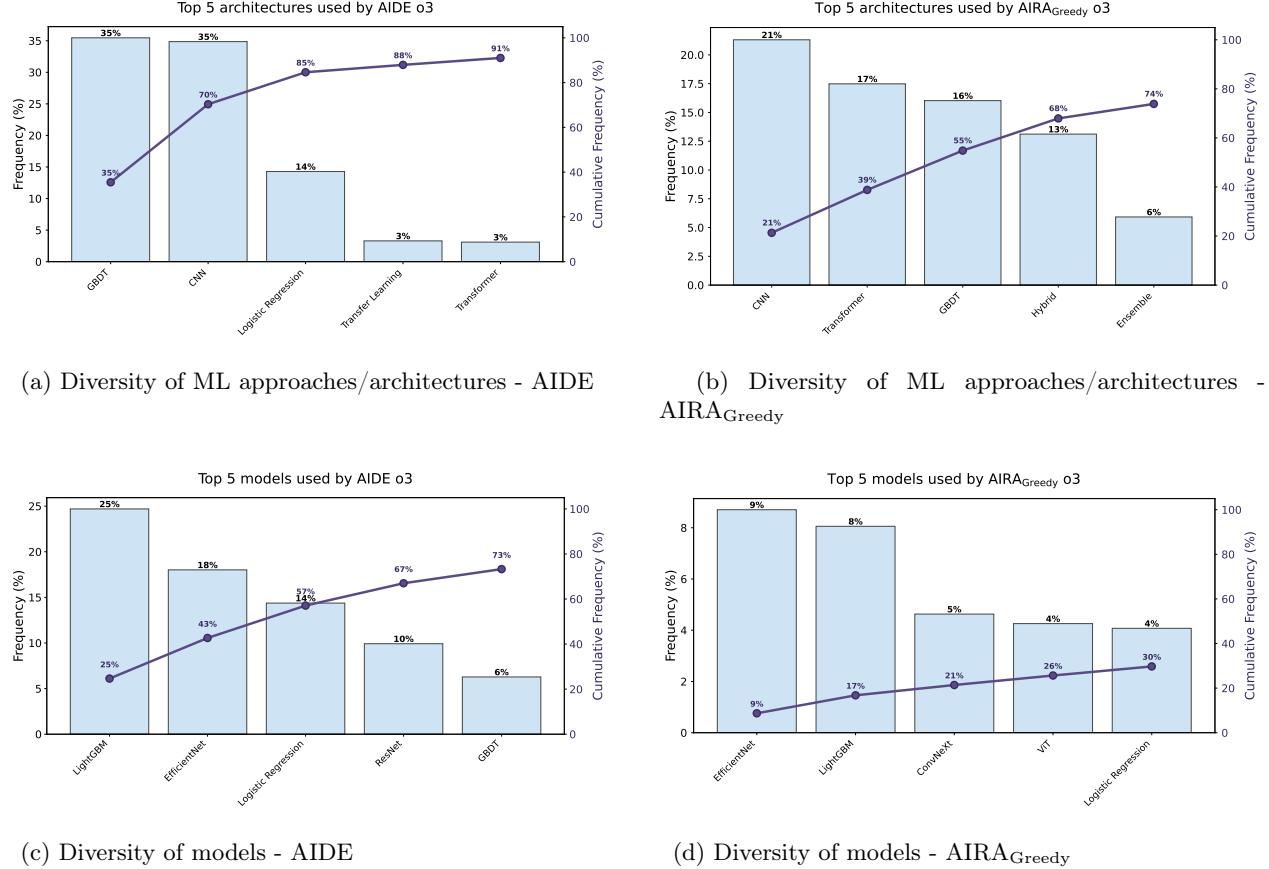
We remove prompt-adaptive complexity and the mention of diversity in the system prompt and we reuse sibling memory to request from the agent in the system prompt to come up with similar ideas.

By changing the parts of the prompt mentioning diversity, we intend to only impact the diversity of ideas generated by the agent, and not other solution aspects, such as implementation quality.

## 4 The Ideation Diversity Bottleneck

This section presents the core results of this paper. First, in Section 4.1, we analyze a large sample of agentic trajectories in MLE-bench. In Section 4.2, we show the controlled experiment to validate the diversity hypothesis. Finally, in Section 4.3, we incorporate additional metrics in our analysis.

### 4.1 Deep-dive on Agentic Trajectories



(c) Diversity of models - AIDE

(b) Diversity of ML approaches/architectures - AIRAGreedy

(a) Diversity of ML approaches/architectures - AIDE

(d) Diversity of models - AIRAGreedy

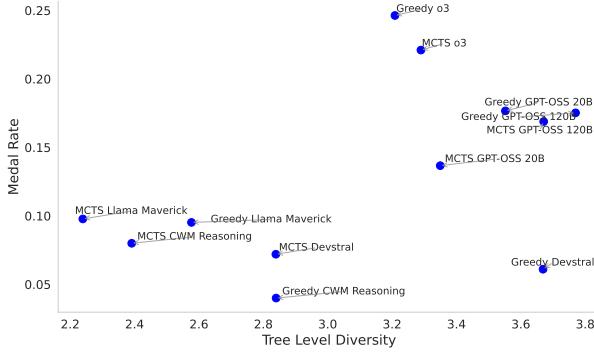
**Figure 3** Overview of diversity in models and architectures used on the 22 MLE-bench lite tasks, illustrating the differences between the AIDE and AIRAGreedy scaffolds. (a, b) Distribution of architectures for AIDE and AIRAGreedy. (c, d) Distribution of model families.

*The agent scaffold choice impacts ideation diversity* Figure 3 illustrates a comparison between AIDE and AIRAGreedy agents, both using o3 as the backbone. Observing the model architectures and general machine learning approaches used by the agent, we can see from Figure 3(a) that AIDE agents prefer Gradient Boosting Decision Trees (GBDT) and Convolutional Neural Networks (CNN) in 70% of the initial draft nodes. In contrast, AIRAGreedy agents generate a greater diversity of ideas. The most common architectures among these agents are CNN, Transformers, GBDT, and Hybrid models that combine multiple approaches. Collectively, these four architectures represent 68% of the ideas produced, as shown in Figure 3(b).

Looking at the models trained by the agent in Figure 3(c) and (d), LightGBM and EfficientNet represent 43% of models AIDE agents intend to train in its initial draft nodes, while in the case of AIRAGreedy as many as 9 models represent this percentage. This difference in diversity highlights the importance of the design of agents (system prompt, search mechanism, operators), in influencing the variety of models the agent intends to train or use. Next, we study how this ideation diversity correlates to performance on MLE-bench.

*Diversity correlates with performance on MLE-bench* Figure 1 shows the correlation of the MLE-bench score (measured as medal rate) with ideation diversity (measured as architecture choice distribution entropy), for each run of the AI research agents included in our study on the 75 tasks of MLE-bench. Here, a point in the plot refers to one agent’s performance on the full set of tasks.

We observe a good correlation between diversity and performance, with two distinct clusters, one including high-performing agents (using o3, gpt-oss 120b, and gpt-oss 20b as backbones) and the other using open-source LLMs in our study (Llama Maverick, Devstral, CWM). Agents that utilize a wider range of techniques tend to achieve higher performance. Additionally, we observe in Figure 4 how diversity changes for different agents, by measuring how many model architectures on average are used in the first 5 nodes of the agent’s exploration, a metric we refer to as tree-level diversity.

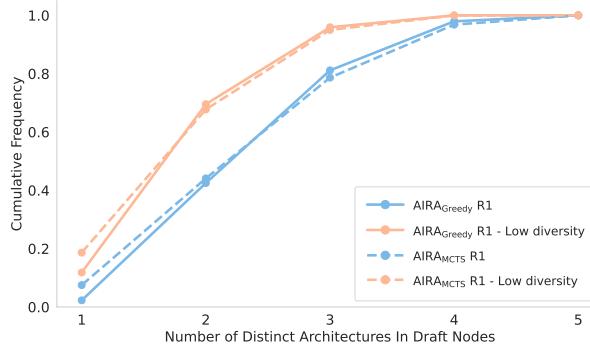


**Figure 4** Correlation between tree-level diversity and performance on MLE-bench

Figure 4 shows how high-performing models considered here (o3, gpt-oss 120b, gpt-oss 20b) use more diverse architectures in the 5 initial ideas (3.5 distinct architectures on average) compared to Llama Maverick, Devstral and CWM (2.8 distinct architectures on average). Like diversity measured as entropy, tree-level diversity also correlates with performance.

## 4.2 Impact of Diversity: A Controlled Experiment

We have seen that better agents are usually having more diversity of ideas in their trajectories. To understand whether diversity has a causal relationship with performance, we perform a controlled experiment, as described in Section 3.3, by removing the different mechanisms for diversity, and directly prompting the agents to generate similar ideas to solve a single task.



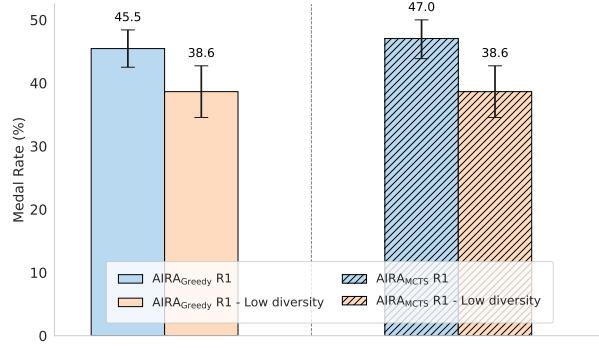
**Figure 5** Number of distinct architectures per task - Cumulative Distribution

#### 4.2.1 Are We Actually Influencing Diversity?

Prompting the agent to come up with similar ideas negatively impacts the diversity of ideas generated by the agent. Figure 5 shows that agents with less diversity use a decreased number of unique architectures and general ML approaches. Baseline agents  $\text{AIRA}_{\text{Greedy}}$  and  $\text{AIRA}_{\text{MCTS}}$  use no more than 2 different architectures in their 5 initial drafts in only 40% of tasks. However, the  $\text{AIRA}_{\text{Greedy}}\text{- Low Diversity}$  and  $\text{AIRA}_{\text{MCTS}}\text{- Low Diversity}$  agents that are prompted to come up with similar ideas, use no more than 2 distinct architectures or approaches in 70% of tasks. These different behaviors highlight the actual impact of diversity mechanisms on ideation.

#### 4.2.2 Results

When it comes to performance measured as medal rate, Figure 6 demonstrates that reducing ideation diversity - by prompting the agent differently in order to generate similar ideas - leads to a decline of performance on MLE-bench lite. This applies for both agentic scaffolds  $\text{AIRA}_{\text{Greedy}}$  and  $\text{AIRA}_{\text{MCTS}}$ , with a 6.9 and 8.4 absolute points decrease, respectively. By modifying the system prompt to isolate the effect of ideation diversity, the results indicate that diversity is an important factor limiting performance.



**Figure 6** Comparison of MLE-bench lite medal rate of  $\text{AIRA}_{\text{Greedy}}$  and  $\text{AIRA}_{\text{MCTS}}$  with and without interventions to reduce solution diversity (as indicated by '- Low diversity'). Error bars represent 95% confidence intervals computed using stratified bootstrapping, using the rliable library (Agarwal et al., 2021).

### 4.3 Evaluating with Alternative Metrics

While relevant, the differences in performance observed so far are constrained to the medal rate metric, used by default. The medal rate metric, by itself, may not offer a comprehensive picture of agent performance on MLE-bench. In the current section, we introduce alternative metrics to provide a more complete assessment of performance on MLE-bench.

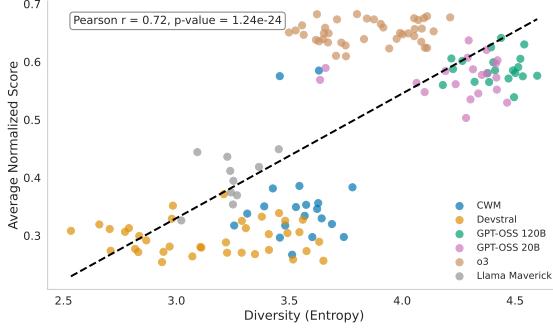
#### 4.3.1 Alternative Metrics

Each of the alternative metrics we consider offers a distinct perspective: while some, like medal rate, emphasize marginal improvements, others account for all performance gains. Additionally, certain metrics are based entirely on human score distributions, whereas others operate independently of them. We consider 4 additional metrics. 1. **Valid Submission Rate**: The percentage of tasks in which the agent is able to make a valid submission. This metric captures the ability of the agent to ideate, implement, and debug until reaching at least one valid submission. 2. **Average Normalized Score**: For each agent attempt at a task, we compute a normalized score: a score of 0 represents the lowest human score achieved on the task, and 1 the highest. The metric captures how good agents submissions are, independently from human score distributions. 3. **Percentile**: The metric captures the ability of the agent to outperform humans at machine learning engineering. Compared to medal rate, this metric still relies on human score distributions, and offer a less discrete assessment of the performance. Like average normalized scores, and unlike medal rates, improvements of percentile in poor and strong scores are equally valued. 4. **ELO-Based Agent Ranking**: We create an ELO system (Bradley and Terry, 1952) using all possible heads-to-heads between agents' scores. ELO rankings are agnostic of the human score

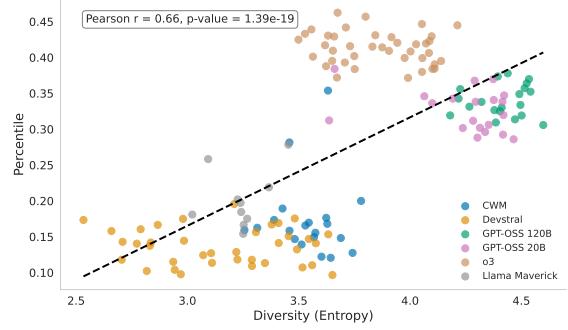
distribution on MLE-bench tasks. ELO difference of 100 points corresponds to about a 64% expected win probability for the higher-rated agent.

#### 4.3.2 Data Analysis Results with Additional Metrics

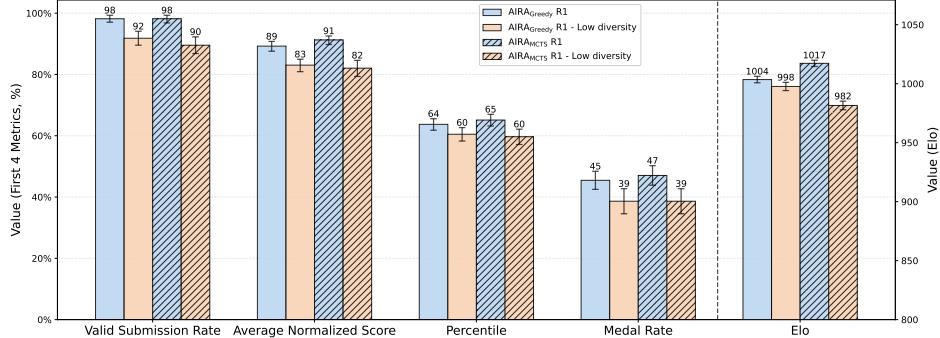
We use the new set of metrics to gain a deeper understanding of the correlation between diversity and performance. When measuring performance using either the percentile or the average normalized score, instead of the medal rate, our correlation results remain consistent and, in fact, show even higher correlations (Figures 7 and 8).



**Figure 7** Correlation between diversity and performance measured as average normalized score



**Figure 8** Correlation between diversity and performance measured as percentile



**Figure 9** Results of the Controlled Experiments with Additional Metrics

#### 4.3.3 Controlled Experiment Results with Additional Metrics

Using this additional set of metrics, we can perform a more comprehensive analysis of the controlled experiment. Figure 9 shows the performance gap between baseline agents ( $\text{AIRAGreedy}$ , and  $\text{AIRAMCTS}$ ) and agents with ablated diversity. The performance loss is observed across all different metrics, validating our hypothesis that agents perform worse when ideas are less diverse.

A notable observation is the drop in valid submission rates, which fall from 98% to 92% for  $\text{AIRAGreedy}$  and to 90% for  $\text{AIRAMCTS}$ . This indicates that, for certain tasks, Low Diversity agents are sometimes unable to produce even a single valid submission during their search. Our analysis reveals that this decline is primarily driven by two competitions: 'text-normalization-challenge-english-language' and 'text-normalization-challenge-russian-language'. Upon examining agent trajectories, we find that Low Diversity agents repeatedly attempt to implement the same model, T5 (Raffel et al., 2020), but consistently fail, resulting in timeouts. In contrast, baseline agents implement a wider range of solutions and are more often able to make correct submissions. Notably, baseline agents also occasionally attempt to implement T5 and encounter similar failures, but their greater ideation diversity allows them to succeed elsewhere. These two competitions also account for a significant portion (estimated at around half) of the observed decline in medal rates.

Our findings suggest that one reason for the drop in performance is that Low Diversity agents tend to focus on similar ideas that they sometimes cannot successfully implement. In other words, ideation diversity is crucial for performance, as it increases the likelihood that an AI research agent will attempt solutions it is actually capable of executing.

## 5 Discussion

*What does it take to be a good AI research agent?* We can imagine a hypothetical future scenario where excellent AI research agents ideate brilliant experiments and have outstanding coding skills to implement them. Until we get to this ideal situation, in practice, even state-of-the-art AI research agents will exhibit limited ideation and implementation capabilities, particularly when evaluated in challenging, real-world settings. In this imperfect, yet realistic, scenario, given the same level of capabilities, we prefer agents with greater ideation diversity. First, because it de-risks implementation pitfalls. Our analysis of the controlled experiment shows that one reason why diversity is important is that it helps agents design solutions they are actually able to execute, highlighting the interplay between ideation and implementation. If the different proposed plans by the agent rely on similar approaches, and those happen to be hard to implement by the agent (in the context of the particular task), then we risk low implementation accuracies. Intuitively, a potential second argument for ideation diversity is that given the difficulty of coming up with creative, yet feasible research ideas, exploring significantly different paths hedges against pursuing a single unproductive direction (even if the agent knows how to implement it), and enables agents to more effectively explore the solution space of machine learning problems. We want to invest the allocated compute in a diversified, yet plausible, set of ideas. However, this second reason is hard to evaluate given the implementation bottleneck. Ultimately, a good experimentation plan could fail due to the agent being unable to implement it. Repeating these controlled experiments as LLMs’ coding capabilities get increasingly more powerful may yield valuable insights.

*Importance of the implementation bottleneck.* Unsurprisingly, implementation quality is an important bottleneck of AI research agents. We observe a strong correlation between AI research agents’ performance and the ability to implement sufficiently complex solutions. By aggregating performances of AIRA (Greedy and MCTS) for each LLM, Figure 10 shows that, on average, the more time an agent spends on each successfully implemented solution (including ideation, implementation, and model training), the more medals it earns. This suggests that performance increases with the agents’ ability to implement more complex solutions. Furthermore, Figure 11 shows that agents perform better when, out of the 24 hours allotted to complete a task, they spend a higher proportion of time on successfully implemented solutions. However, since LLMs and coding agents are improving rapidly (Kwa et al., 2025), particularly in verifiable tasks (DeepSeek-AI et al., 2025), we hypothesize that the relative importance of the ideation and planning phase might increase over time, not to de-risk implementation pitfalls, but to efficiently explore the solution space.

*Generalization to other benchmarks.* The findings presented in this study are based on experiments conducted using MLE-bench only. Given the range of machine learning tasks included in this benchmark, we hypothesize that our results are likely to generalize to other machine learning tasks. Additional benchmarks could be examined in future research.

*Limitations of MLE-bench evaluations.* Performance on MLE-bench has traditionally been evaluated using Kaggle’s medal system, where medals are awarded based on score percentiles. For example, in competitions with fewer than 99 teams, gold medals are given to the top 10% of submissions. However, this medal-based evaluation framework has several important limitations.

First, medal criteria change with the number of submissions (e.g., bronze goes to the top 10% for 1000+ teams, but top 40% for 1–249 teams), so earning a medal does not indicate consistent performance across competitions. Second, as shown in the Appendix, the gap between the bronze threshold and the best score is often extremely small—frequently below 3%.

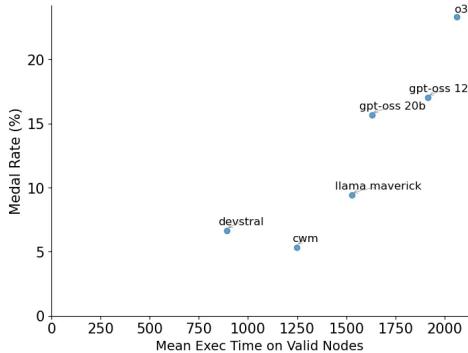
Since agents are evaluated on custom test sets, while medal thresholds are computed using Kaggle private test sets, the data split variance introduces some score variability, and can affect medal outcomes. Third, some competitions are over a decade old, and human score distributions from these may not represent current

machine learning standards. In some older competitions like ‘detecting-insults-in-social-commentary’, AIRA agents are able to outperform best human submissions. In recent competitions (post-2022), agent performance drops sharply, with most agents unable to earn medals. To address these different issues, we also used additional metrics, in Section 4.3. In the Appendix, we provide additional information on these alternative metrics.

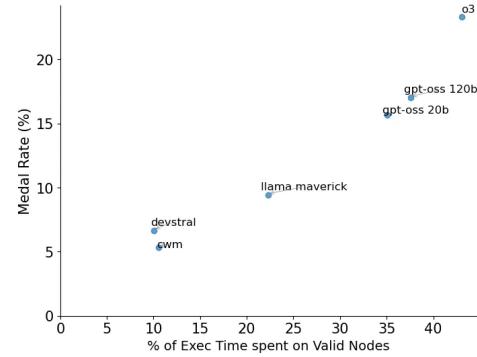
*Limitations of this study.* Despite the efforts to isolate ideation diversity, it is difficult to track the potential second-order effects of modifying the system prompt. To better isolate the effect of ideation diversity, future work could focus on disentangling the LLM responsible for ideating, and the one responsible for implementing. In this work, we also experimented with controlling diversity through temperature, as detailed in the Appendix.

## 6 Related Work

*Generation diversity in language models.* Generation diversity in language models has been studied and even explicitly promoted since the statistical machine translation era (Macherey and Och, 2007; Gimpel et al., 2013; Xiao et al., 2013), where selecting (Devlin and Matsoukas, 2012) or combining (Macherey and Och, 2007) a set of diverse yet plausible generations lead to improved translation quality. Similar observations were later identified in neural machine translation and other sequence-to-sequence settings (Li et al., 2015; Vijayakumar et al., 2016; Ippolito et al., 2019). Holtzman et al. (2020) developed nucleus sampling with the goal of generating both coherent and diverse text. More recently, Murthy et al. (2025) and Kirk et al. (2024) study the effect of RLHF (Ouyang et al., 2022) on LLMs with a focus on the (decreased) generation diversity. Chen et al. (2024) study effect of diversity of synthetic data in training LLMs. Li et al. (2025c) propose a diversity-preserving algorithm for supervised fine-tuning of LLMs.



**Figure 10** Correlation between the average execution time on valid nodes and performance (MLE-bench, 75 tasks)



**Figure 11** Correlation between the share of execution time spent on valid nodes and performance (MLE-bench, 75 tasks)

*Diversity in reinforcement learning and population-based reinforcement learning.* Trajectory diversity is synonym of increased exploration in reinforcement learning models. Hong et al. (2018) investigate a diversity-driven exploration strategy for training reinforcement learning models. Eysenbach et al. (2019) propose a diversity objective to learn skills without reward functions. Parker-Holder et al. (2020) improve diversity in population-based reinforcement learning, while Conti et al. (2018) propose a novelty objective in a population of agents to improve exploration in reinforcement learning. More recently, Yao et al. (2025) propose a diversity-aware policy optimization algorithm; unlike the works cited above, it does so in the context of LLMs. Zeng et al. (2025) propose B-star, a reinforcement learning approach for reasoning LLMs that balances exploration and exploitation.

*Diversity in multi-agent systems.* Diversity has also been studied in the context of multi-agent foundation models (Tuyls, 2023). Bettini et al. (2025) study the impact of behavioral diversity in multi-agent reinforcement learning. Li and Zhu (2025) address how agents in multi-agent RL often end up learning nearly identical behaviors when they share network parameters, which hurts exploration. The authors propose CTEM, a method that uses trajectory entropy maximization to push agents toward more diverse behaviors without needing complex density models. Li et al. (2025a) find that generating diverse teammates in multi-agent training can lead to random, semantically meaningless behaviors, reducing training efficiency. Their SemDiv approach leverages large language models to describe coordination strategies in natural language, then converts these into reward functions for training meaningful teammate policies. In LLM-agent world simulations, Chu et al. (2025) investigate prompt design’s impact on conversational diversity and introduce a prompt-tuning mechanism that controls diversity via a single parameter.

*Automated machine learning and AI research agents.* In the era of tool-use LLM-based agents (Schick et al., 2023; Kaddour et al., 2023), Shen et al. (2023); Nathani et al. (2025) evaluate agents at implementing machine learning tasks. MLE-bench (Chan et al., 2025), the benchmark used in this work, is a machine learning benchmark consisting of 75 Kaggle tasks. Zhao et al. (2025) propose a self-contained machine learning task with a focus on language models. AIRA (Toledo et al., 2025), studies AI research agents by formalizing AI research as search policies over a space of candidate solutions. This development is parallel to agentic benchmarks and scaffolds for other fields, such as software engineering (Jimenez et al., 2024; Jiang et al., 2025b).

## 7 Conclusions

This work started from the hypothesis that ideation diversity is a key bottleneck in AI research agents’ performance. We have confirmed this hypothesis by conducting a large-scale analysis on AI research agents’ trajectories and performing a controlled experiment. Our findings hold across several evaluation metrics.

In future work, we suggest focusing on diversity-aware methods, as other bottlenecks such as implementation quality will decrease in importance when AI systems keep improving. We also recommend considering multiple evaluation metrics and extending the existing benchmarks to more recent machine learning tasks.

## References

- Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G Bellemare. Deep reinforcement learning at the edge of the statistical precipice. *Advances in Neural Information Processing Systems*, 2021.
- Matteo Bettini, Ryan Kortvelesy, and Amanda Prorok. The impact of behavioral diversity in multi-agent reinforcement learning, 2025. <https://arxiv.org/abs/2412.16244>.
- Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, Dec 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06792-0. <https://doi.org/10.1038/s41586-023-06792-0>.
- Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. ISSN 00063444, 14643510. <http://www.jstor.org/stable/2334029>.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. [https://proceedings.neurips.cc/paper\\_files/paper/2020/file/1457c0d6bfcb4967418bf8ac142f64a-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bf8ac142f64a-Paper.pdf).
- Cameron B. Browne, Edward Powley, Daniel Whitehouse, Simon M. Lucas, Peter I. Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search

methods. *IEEE Transactions on Computational Intelligence and AI in Games*, 4(1):1–43, 2012. doi: 10.1109/TCIAIG.2012.2186810.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Aleksander Madry, and Lilian Weng. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *The Thirteenth International Conference on Learning Representations*, 2025. <https://openreview.net/forum?id=6s5uXNWGIh>.

Hao Chen, Abdul Waheed, Xiang Li, Yidong Wang, Jindong Wang, Bhiksha Raj, and Marah I. Abdin. On the diversity of synthetic data and its impact on training large language models, 2024. <https://arxiv.org/abs/2410.15226>.

KuanChao Chu, Yi-Pei Chen, and Hideki Nakayama. Exploring and controlling diversity in llm-agent conversation, 2025. <https://arxiv.org/abs/2412.21102>.

Edoardo Conti, Vashisht Madhavan, Felipe Petroski Such, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. Improving exploration in evolution strategies for deep reinforcement learning via a population of novelty-seeking agents. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS’18, page 5032–5043, Red Hook, NY, USA, 2018. Curran Associates Inc.

Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In *Proceedings of the 5th International Conference on Computers and Games*, CG’06, page 72–83, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3540755373.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghaio Lu, Shangyan Zhou, Shanhua Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. <https://arxiv.org/abs/2501.12948>.

Jacob Devlin and Spyros Matsoukas. Trait-based hypothesis selection for machine translation. In Eric Fosler-Lussier, Ellen Riloff, and Srinivas Bangalore, editors, *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 528–532, Montréal, Canada, June 2012. Association for Computational Linguistics. <https://aclanthology.org/N12-1059/>.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Benjamin Eysenbach, Abhishek Gupta, Julian Ibarz, and Sergey Levine. Diversity is all you need: Learning skills without a reward function. In *International Conference on Learning Representations*, 2019. <https://openreview.net/forum?id=SJx63jRqFm>.

Meta FAIR CodeGen Team. Cwm: An open-weights llm for research on code generation with world models, 2025. <https://ai.meta.com/research/publications/cwm/>.

- Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter. Auto-sklearn 2.0: hands-free automl via meta-learning. *J. Mach. Learn. Res.*, 23(1), January 2022. ISSN 1532-4435.
- Kevin Gimpel, Dhruv Batra, Chris Dyer, and Gregory Shakhnarovich. A systematic exploration of diversity in machine translation. In David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard, editors, *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1100–1111, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. <https://aclanthology.org/D13-1111/>.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration, 2020. <https://arxiv.org/abs/1904.09751>.
- Zhang-Wei Hong, Tzu-Yun Shann, Shih-Yang Su, Yi-Hsiang Chang, Tsu-Jui Fu, and Chun-Yi Lee. Diversity-driven exploration strategy for deep reinforcement learning. In *Conference on Neural Information Processing Systems (NIPS)*, 2018.
- Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. Mlagentbench: Evaluating language agents on machine learning experimentation, 2024. <https://arxiv.org/abs/2310.03302>.
- Daphne Ippolito, Reno Kriz, Maria Kustikova, João Sedoc, and Chris Callison-Burch. Comparison of diverse decoding methods from conditional language models. *arXiv preprint arXiv:1906.06362*, 2019.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.
- Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixin Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. AIDE: AI-Driven Exploration in the Space of Code. *arXiv preprint*, 2025a. <https://arxiv.org/abs/2502.13138>.
- Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixin Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. Aide: Ai-driven exploration in the space of code, 2025b. <https://arxiv.org/abs/2502.13138>.
- Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. <https://openreview.net/forum?id=VTF8yNQM66>.
- Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. Challenges and applications of large language models, 2023. <https://arxiv.org/abs/2307.10169>.
- Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity. In *ICLR*, 2024. <https://openreview.net/forum?id=PXD3FAVHJT>.
- Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *Proceedings of the 17th European Conference on Machine Learning*, ECML’06, page 282–293, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 354045375X. doi: 10.1007/11871842\_29. [https://doi.org/10.1007/11871842\\_29](https://doi.org/10.1007/11871842_29).
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, Ryan Bloom, Thomas Broadley, Haoxing Du, Brian Goodrich, Nikola Jurkovic, Luke Harold Miles, Seraphina Nix, Tao Lin, Neev Parikh, David Rein, Lucas Jun Koba Sato, Hjalmar Wijk, Daniel M. Ziegler, Elizabeth Barnes, and Lawrence Chan. Measuring ai ability to complete long tasks, 2025. <https://arxiv.org/abs/2503.14499>.
- Yann LeCun and Yoshua Bengio. *Convolutional networks for images, speech, and time series*, page 255–258. MIT Press, Cambridge, MA, USA, 1998. ISBN 0262511029.
- Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.
- Lihe Li, Lei Yuan, Pengsen Liu, Tao Jiang, and Yang Yu. LLM-assisted semantically diverse teammate generation for efficient multi-agent coordination. In *Forty-second International Conference on Machine Learning*, 2025a. <https://openreview.net/forum?id=Vhktbw6Vid>.
- Lujun Li, Lama Sleem, Niccolo' Gentile, Geoffrey Nichil, and Radu State. Exploring the impact of temperature on large language models: hot or cold?, 2025b. <https://arxiv.org/abs/2506.07295>.

Tianxu Li and Kun Zhu. Self-supervised multi-agent diversity with nonparametric entropy maximization. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '25, page 1291–1299, Richland, SC, 2025. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9798400714269.

Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. Preserving diversity in supervised fine-tuning of large language models. In *The Thirteenth International Conference on Learning Representations*, 2025c. <https://openreview.net/forum?id=NQEe7B7bSw>.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

Wolfgang Macherey and Franz J. Och. An empirical study on computing consensus translations from multiple machine translation systems. In Jason Eisner, editor, *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 986–995, Prague, Czech Republic, June 2007. Association for Computational Linguistics. <https://aclanthology.org/D07-1105/>.

Sonia Krishna Murthy, Tomer Ullman, and Jennifer Hu. One fish, two fish, but not the whole sea: Alignment reduces language models' conceptual diversity. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, page 11241–11258. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.nacl-long.561. <http://dx.doi.org/10.18653/v1/2025.nacl-long.561>.

Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Amar Budhiraja, Despoina Magka, Vladislav Vorotilov, Gaurav Chaurasia, Dieuwke Hupkes, Ricardo Silveira Cabral, Tatiana Shavrina, Jakob Foerster, Yoram Bachrach, William Yang Wang, and Roberta Raileanu. Mlgym: A new framework and benchmark for advancing ai research agents, 2025. <https://arxiv.org/abs/2502.14499>.

OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. <https://arxiv.org/abs/2508.10925>.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. <https://arxiv.org/abs/2203.02155>.

Jack Parker-Holder, Aldo Pacchiano, Krzysztof Choromanski, and Stephen Roberts. Effective diversity in population based reinforcement learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), January 2020. ISSN 1532-4435.

Matthew Renze. The effect of sampling temperature on problem solving in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, page 7346–7356. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.432. <http://dx.doi.org/10.18653/v1/2024.findings-emnlp.432>.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. <https://openreview.net/forum?id=Yacmpz84TH>.

Claude E Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face, 2023. <https://arxiv.org/abs/2303.17580>.

Kyle Swanson, Wesley Wu, Nash L. Bulaong, John E. Pak, and James Zou. The virtual lab of ai agents designs new sars-cov-2 nanobodies. *Nature*, Jul 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09442-9. <https://doi.org/10.1038/s41586-025-09442-9>.

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

Devstral Team. Devstral: Fine-tuning language models for coding agent applications, 2025a. <https://arxiv.org/abs/2509.25193>.

Meta Llama Team. Llama 4 system card, 2025b. <https://ai.meta.com/blog/llama-4-multimodal-intelligence/>.

Edan Toledo, Karen Hambardzumyan, Martin Josifoski, Rishi Hazra, Nicolas Baldwin, Alexis Audran-Reiss, Michael Kuchnik, Despoina Magka, Minqi Jiang, Alisia Maria Lupidi, Andrei Lupu, Roberta Raileanu, Kelvin Niu, Tatiana Shavrina, Jean-Christophe Gagnon-Audet, Michael Shvartsman, Shagun Sodhani, Alexander H. Miller, Abhishek Charnalia, Derek Dunfield, Carole-Jean Wu, Pontus Stenetorp, Nicola Cancedda, Jakob Nicolaus Foerster, and Yoram Bachrach. Ai research agents for machine learning: Search, exploration, and generalization in mle-bench, 2025. <https://arxiv.org/abs/2507.02554>.

Karl Tuyls. Multiagent learning: From fundamentals to foundation models. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '23, page 1, Richland, SC, 2023. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450394321.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Ashwin K Vijayakumar, Michael Cogswell, Ramprasath R Selvaraju, Qing Sun, Stefan Lee, David Crandall, and Dhruv Batra. Diverse beam search: Decoding diverse solutions from neural sequence models. *arXiv preprint arXiv:1610.02424*, 2016.

M. J. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10 (2):115–152, 1995. <https://eprints.soton.ac.uk/252102/>.

Tongshuang Wu, Ellen Jiang, Aaron Donsbach, Jeff Gray, Alejandra Molina, Michael Terry, and Carrie J Cai. Promptchainer: Chaining large language model prompts through visual programming. In *Extended Abstracts of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI EA '22, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391566. doi: 10.1145/3491101.3519729. <https://doi.org/10.1145/3491101.3519729>.

Yuheng Wu, Azalia Mirhoseini, and Thierry Tambe. On the role of temperature sampling in test-time scaling, 2025. <https://arxiv.org/abs/2510.02611>.

Tong Xiao, Jingbo Zhu, and Tongran Liu. Bagging and boosting statistical machine translation systems. *Artificial Intelligence*, 195:496–527, 2013.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search, 2025. <https://arxiv.org/abs/2504.08066>.

Jian Yao, Ran Cheng, Xingyu Wu, Jibin Wu, and Kay Chen Tan. Diversity-aware policy optimization for large language model reasoning, 2025. <https://arxiv.org/abs/2505.23433>.

Weihao Zeng, Yuzhen Huang, Lulu Zhao, Yijun Wang, Zifei Shan, and Junxian He. B-STar: Monitoring and balancing exploration and exploitation in self-taught reasoners. In *The Thirteenth International Conference on Learning Representations*, 2025. <https://openreview.net/forum?id=P6dwZJpJ4m>.

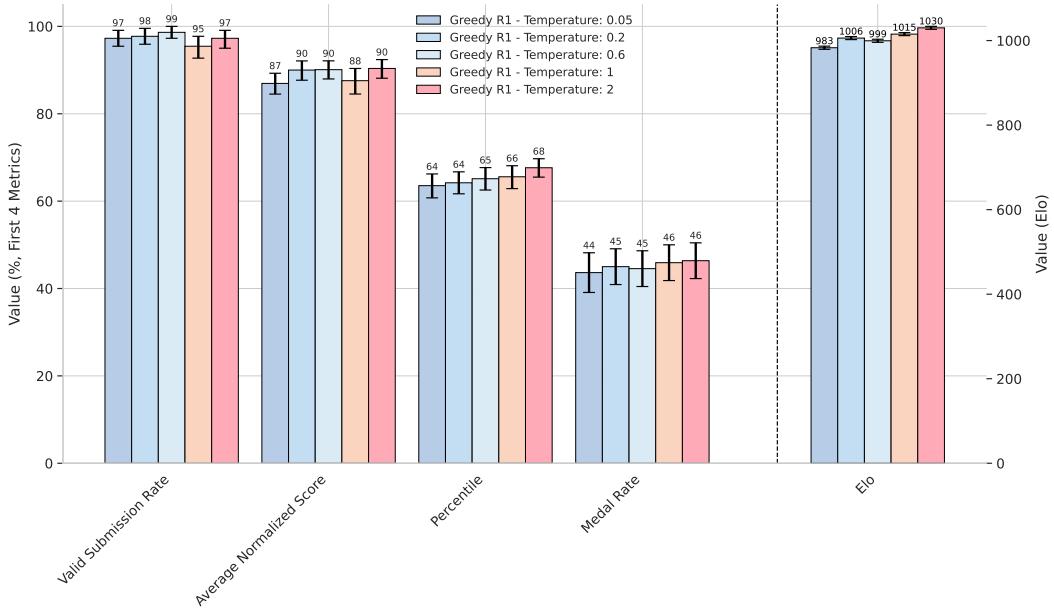
Bingchen Zhao, Despoina Magka, Minqi Jiang, Xian Li, Roberta Raileanu, Tatiana Shavrina, Jean-Christophe Gagnon-Audet, Kelvin Niu, Shagun Sodhani, Michael Shvartsman, Andrei Lupu, Alisia Lupidi, Edan Toledo, Karen Hambardzumyan, Martin Josifoski, Thomas Foster, Lucia Cipolina-Kun, Abhishek Charnalia, Derek Dunfield, Alexander H. Miller, Oisin Mac Aodha, Jakob Foerster, and Yoram Bachrach. The automated llm speedrunning benchmark: Reproducing nanopt improvements, 2025. <https://arxiv.org/abs/2506.22419>.

## A Appendix

### A.1 Controlling Ideation Diversity Using Temperature

With the intuition that the temperature sampling parameter has an impact on ideation diversity and performance, we run several experiments intervening on the sampling temperature. To isolate the impact of temperature, we use AIRA<sub>Greedy</sub> but remove all mechanisms that enhance ideation diversity ( sibling memory, prompt-adaptive complexity, mention of diversity in the system prompt), as described in section 3.3, and experiment with different temperatures, above and below the recommended temperature of 0.6 for DeepSeek-R1 (DeepSeek-AI et al., 2025).

Figure 12 shows that changing temperature does not have an impact on performance (neither beneficial nor detrimental), assessed as medal rate. Results hold across alternative metrics like valid submission rate, average normalized score, and percentile, except for Elo, the only metric where increased temperature significantly leads to improved performance. Despite our effort in isolating ideation diversity, we hypothesize that the reason for these negative results is that modifying the temperature parameter (instead of using the recommended one) also affects the agent in additional manners (Renze, 2024; Wu et al., 2025; Li et al., 2025b). For example, we would expect the implementation capabilities to also be affected,<sup>2</sup> and there could be second-order effects that are hard to reason about. We leave further investigating the temperature-based results as future work.



**Figure 12** Agent Performance With Different Temperature Settings

### A.2 Alternative Metrics

#### A.2.1 Limitations of MLE-Bench Medal System

Performance on MLE-bench has traditionally been evaluated using Kaggle’s medal system, where medals are awarded based on score percentiles (see details here). For instance, in competitions with fewer than 99 teams, gold medals are given to the top 10% of submissions. However, several limitations are inherent to this medal-based evaluation framework.

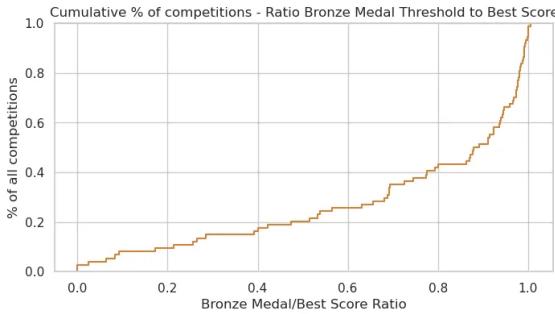
<sup>2</sup>Surprisingly, Renze (2024) found no statistically significant effect in problem solving skills of LLMs for certain temperature ranges.

*Variable medal thresholds* Kaggle medal criteria vary with the number of submissions (e.g., bronze is the top 10% for competitions with 1000+ teams vs. top 40% for those with 1-249 teams; see Figure 13). This inconsistency means that medals do not equate to the same performance level across different competitions.

|        | <b>0 - 99 teams</b> | <b>100 - 249 teams</b> | <b>250 - 999 teams</b> | <b>1000+ teams</b> |
|--------|---------------------|------------------------|------------------------|--------------------|
| 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%      |

**Table 1** Kaggle Competition Medals

*Narrow thresholds between medals and top scores* In some competitions, there is only a minimal difference between the threshold for earning a medal and the highest score achieved. Specifically, as shown in Figure 13, in about 30% of MLE-bench competitions, the score (relative) difference between the best score and the bronze medal threshold is less than 3%. In around 50% of all MLE-bench lite competitions, this gap is similarly small.



**Figure 13** Ratio between the bronze medal threshold and the best score for each competition on MLE-bench. In many cases, the ratio is close to 1, indicating a very narrow margin between the bronze threshold and the top score.

*Different test sets for evaluation* AI research agents are tested on custom sets designed by MLE-bench, not the private test sets used for human submissions on Kaggle. Human submissions leading to medals or best scores would perform differently on custom test sets for AI research agents, while the medal thresholds are still computed based on submissions on Kaggle private test sets. This variability is an issue, especially when medal thresholds are set close to each other, and could make medal thresholds unreachable by AI research agents.

*Competition age* Some competitions are more than 10 years old. In this rapidly evolving space, human score distributions don't reflect what they would be now in 2025 (for example agents perform better than top human scores on 'detecting-insults' competition). For recent competitions, after 2022, we see a substantial decrease in performance, and all agents tested are not able to get any medal for most competitions after 2022.

These limitations suggest that the medal rate metric, by itself, may not offer a comprehensive evaluation of agent performance on MLE-bench. Therefore, in the following section, we describe alternative metrics to provide a more complete assessment of performance on MLE-bench. We used these metrics for the results in Section 4.3.

### A.2.2 Evaluation Principles

Evaluations serve as a measure of a model's capabilities. For MLE-bench, we aim to evaluate our AI research agent's ability to address a wide range of machine learning tasks, and manage key stages of the machine learning lifecycle in order to get the best performance, including data cleaning & preprocessing, model development & tuning, validation. We describe here a list of principles to consider for additional metrics.

*Limited Set of Metrics* We have shown that relying solely on medal rates was hindering our assessment of performance on MLE-bench. We need more metrics to get a comprehensive view of the performance of AI research agents. However, relying on a large set of metrics can make the assessment of an agent difficult, and hard to compare with others.

*Inclusion of all attempts* Failed submissions need to be taken into account in the evaluation. If an agent gets a perfect score on 50% of tasks and fails to get a valid submission on the other 50% of tasks, the agent should theoretically not get a perfect score.

*Independence from Human Score Distributions* Agents and humans are not evaluated on the same test sets. Medal rates only value marginal improvements on good scores.

*Capturing the complexity of hill-climbing* In some competitions, the complexity resides in doing the last mile of optimization.

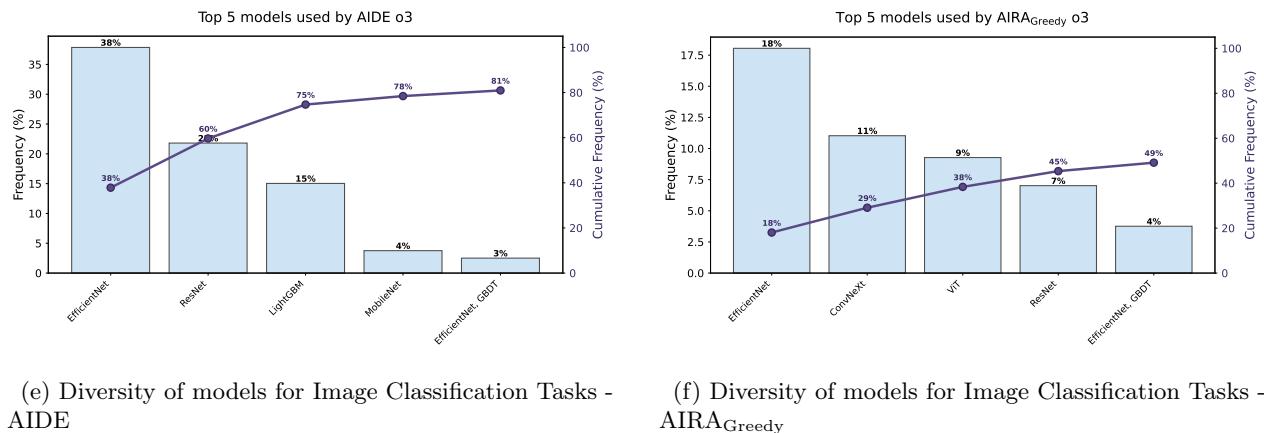
| METRIC                   | INDEPENDENT FROM HUMAN SCORES | VALUES ALL IMPROVEMENTS          | INCLUSION OF ALL ATTEMPTS | CAPTURES HILL-CLIMBING COMPLEXITY         |
|--------------------------|-------------------------------|----------------------------------|---------------------------|---|
| Valid Submission Rate    | x                             | x                                | ✓                         | x   |
| Medal Rate               | x                             | x                                | ✓                         | ✓   |
| Human Score Percentile   | x                             | ✓                                | ✓                         | ✓   |
| Average Normalized Score | ~ (human score bounds)        | x                                | ✓                         | x (all improvements valued equally)       |
| ELO-based ranking        | ✓                             | ~ (values beating another agent) | ✓                         | ~ (depends on agents' score distribution) |

**Table 2** Additional Metrics

Table 2 shows the additional metrics used in this work and how they relate to these principles. None can actually follow all these principles at the same time, demonstrating the importance of having multiple metrics to describe the performance of AI research agents.

### A.3 Models Used By Agents For Image Classification Tasks

Focusing on the neural architectures used by the agents in image classification tasks in Figure 14,<sup>3</sup> we observe that AIDE relies on the EfficientNet architecture (and its variants) for almost 40% of the tasks. For 75% of its attempts, AIDE uses only 3 different architectures: EfficientNet, ResNet (He et al., 2016) and LightGBM. AIRAGreedy uses a wider range of architectures, with EfficientNet, ConvNeXt (Liu et al., 2022), and ViT (Dosovitskiy et al., 2020) making only 38% of the agents’ initial ideas.



(e) Diversity of models for Image Classification Tasks - AIDE

(f) Diversity of models for Image Classification Tasks - AIRAGreedy

**Figure 14** Models used by AIDE and AIRAGreedy for image classification tasks.

<sup>3</sup>Image classification is the largest category in MLE-bench (8 tasks out of 22 in MLE-bench lite)