# The Overfitting Crisis in LLM Workflows: Learning from Machine Learning's Past Mistakes

**Anonymous Author(s)**
Affiliation
Address
email

## Abstract

The rapid development of sophisticated Large Language Model (LLM) workflows—including agentic systems, multi-step reasoning pipelines, and tool-integrated approaches—has led to impressive reported performance across various benchmarks. However, we argue that the field is repeating a critical mistake from early machine learning: reporting results on data that has been implicitly used for training or optimization. The complexity of modern LLM workflows obscures the fact that iterative prompt engineering, benchmark-driven development, and workflow refinement constitute a form of training on evaluation data. This position paper draws parallels to historical overfitting practices in ML, documents how current LLM development methodologies systematically conflate training and testing data, and proposes best practices to address this growing methodological crisis before it undermines the credibility of AI research, particularly in scientific applications.

## 1 Introduction

The rapid development of sophisticated Large Language Model (LLM) workflows—including agentic systems, multi-step reasoning pipelines, and tool-integrated approaches—has led to impressive reported performance across various benchmarks [Bubeck et al., 2023, Wang et al., 2023]. However, we argue that the field is repeating a critical mistake from early machine learning: reporting results on data that has been implicitly used for training or optimization.

The complexity of modern LLM workflows obscures the fact that iterative prompt engineering, benchmark-driven development, and workflow refinement constitute a form of training on evaluation data [Min et al., 2022]. Contemporary LLM development increasingly involves sophisticated multi-component systems where prompt optimization, tool selection, and architectural decisions are iteratively refined based on performance feedback from target benchmarks [Schick and Schütze, 2020, Qin et al., 2023]. This process mirrors the problematic practices that plagued early machine learning research, where model selection and hyperparameter tuning were driven by test set performance.

Unlike traditional machine learning where the boundary between training and testing was relatively clear, LLM workflow development operates in a gray area where the distinction between legitimate system design and implicit training on evaluation data becomes blurred [Perez et al., 2021, Recht et al., 2019]. The iterative nature of prompt engineering, the community-driven sharing of successful strategies, and the progressive refinement of multi-agent architectures all contribute to a systematic optimization process that effectively treats evaluation benchmarks as validation sets.

In early machine learning research, a common antipattern emerged that would later be recognized as a fundamental threat to scientific validity. Researchers would select and tune models based on performance on available datasets, with hyperparameter optimization, feature engineering, and architecture choices driven by test set performance [Dwork et al., 2015, Blum and Hardt, 2015]. Final results were reported on the same datasets used for development, leading to real-world deployment showing

Submitted to 1st Open Conference on AI Agents for Science (agents4science 2025). Do not distribute.

significant performance degradation—a phenomenon that became known as the "reproducibility crisis" in machine learning [Henderson et al., 2018, Lipton and Steinhardt, 2019].

The machine learning community's recognition of this methodological error represents one of the most important paradigm shifts in computational research methodology. The "test set" had effectively become part of the training process through repeated evaluation and optimization cycles, violating the fundamental assumption of independent evaluation [Ioannidis, 2005]. This realization catalyzed the establishment of rigorous protocols that became foundational to credible ML research: clear separation of training, validation, and test sets; test sets remaining untouched until final evaluation; cross-validation and holdout validation for development; and independent evaluation on truly unseen data [Kohavi, 1995, Varma and Simon, 2006].

These methodological advances transformed machine learning from a field prone to optimistic bias into one with robust evaluation standards [Raschka, 2018]. However, the emergence of complex LLM workflows has created new opportunities for the same fundamental errors to re-emerge in more sophisticated forms. This position paper draws parallels to historical overfitting practices in ML, documents how current LLM development methodologies systematically conflate training and testing data, and proposes best practices to address this growing methodological crisis before it undermines the credibility of AI research, particularly in scientific applications.

## 2 Related Work

### 2.1 Historical Overfitting in Machine Learning

The machine learning community has long recognized the dangers of overfitting to evaluation data. Early ML research suffered from researchers repeatedly testing models on the same evaluation sets, leading to inflated performance estimates that failed to generalize [Dwork et al., 2015]. This led to the establishment of rigorous evaluation protocols including proper train/validation/test splits and holdout methodologies.

The development of comprehensive benchmarking frameworks helped standardize evaluation practices [Olson et al., 2017]. However, as noted by Cohen-Inger et al. [2025], benchmark datasets themselves can become overoptimized targets when the entire research community focuses on the same evaluation sets over extended periods.

### 2.2 Data Contamination in Large Language Models

Recent work has extensively documented data contamination issues in LLM training and evaluation. Golchin and Surdeanu [2023] and Shi et al. [2024] demonstrate that simple decontamination methods like n-gram matching are insufficient, as paraphrasing and translation can easily bypass these measures.

The LessLeak-Bench study [Zhou et al., 2025] provided comprehensive analysis across 83 software engineering benchmarks, finding contamination across nearly all tested models. This systematic contamination undermines reported performance improvements in domains where sophisticated LLM workflows are increasingly deployed.

### 2.3 Evaluation Methodology in ML Systems

The establishment of rigorous evaluation protocols emerged as the ML community's primary defense against overfitting. The foundational train/validation/test split methodology became standard practice, with cross-validation techniques providing robust model selection frameworks [Stone, 1974, Kohavi, 1995]. These approaches ensured that model selection decisions were made independently of final performance evaluation, preventing the test set from becoming an implicit part of the training process [Hastie et al., 2009, Bishop, 2006].

Modern ML systems have extended these principles to encompass continuous evaluation and monitoring throughout the system lifecycle. However, the fundamental principle remains unchanged: evaluation data must remain independent of optimization processes to ensure valid performance estimates.

# 3 The Current Crisis in LLM Workflows

## 3.1 The Implicit Training Problem

Modern LLM workflows involve extensive optimization processes that constitute implicit training on evaluation data. Teams spend weeks iterating on prompts, testing variants against target benchmarks, and selecting the best-performing approaches [Wei et al., 2022]. Each iteration uses the benchmark as a validation signal.

Multi-agent systems, RAG pipelines, and tool usage patterns are designed and refined based on performance on specific tasks and datasets. The architecture itself becomes optimized for known evaluation criteria. Research teams explicitly target improvements on established benchmarks, using them as development objectives rather than independent evaluation metrics.

## 3.2 Complexity as Camouflage

The sophistication of modern LLM workflows obscures the overfitting problem. Complex chains of thought, tree search, and multi-agent interactions create an illusion of generalization, when in fact the entire pipeline has been optimized for specific evaluation patterns [Yao et al., 2023].

Systems that use calculators, code interpreters, and web search appear more general, but tool selection, usage patterns, and integration strategies are typically optimized against known benchmarks. Workflows that adapt their strategies based on input characteristics seem more robust, but these adaptation mechanisms are usually developed and tuned using the same evaluation data they will later be tested on.

## 3.3 Benchmark Contamination and Spillover Effects

The problem is exacerbated by several forms of contamination. Many benchmarks overlap with or derive from data sources used in LLM pre-training, creating subtle forms of data leakage [Shi et al., 2024]. Recent work has shown that simple paraphrasing can bypass decontamination measures, and that models can achieve artificially high performance when such variations are not eliminated [Shi et al., 2024].

Solutions and techniques developed for one benchmark quickly propagate to others, creating implicit optimization across multiple evaluation sets [Rogers et al., 2021, Dodge et al., 2021]. As models improve on existing benchmarks, new versions are created that often share similar patterns and evaluation criteria, perpetuating the contamination cycle.

# 4 Detailed Case Studies: Workflow Overfitting and Generalisability Challenges in Practice

## 4.1 ReAct: Reasoning and Acting Workflows

The ReAct framework [Yao et al., 2022] exemplifies how sophisticated LLM workflows can be systematically overfitted to evaluation benchmarks. ReAct combines reasoning traces with action-taking capabilities, enabling LLMs to interact with external tools while maintaining step-by-step reasoning.

**Development Process Analysis**: The ReAct paper reports evaluation on four benchmarks: HotpotQA for question answering, Fever for fact verification, ALFWorld for text-based games, and WebShop for web navigation. The development methodology involved iterative prompt engineering and workflow refinement specifically targeting these benchmarks.

**Overfitting Evidence**: The ReAct development process exhibits several characteristics indicative of benchmark optimization:

- Systematic testing of different prompting strategies against the target benchmarks
- Iterative refinement of the reasoning-action interleaving based on benchmark performance
- Optimization of tool integration strategies for benchmark-specific requirements

- Fine-tuning of termination criteria based on evaluation results

**Performance Claims**: The paper reports substantial improvements over baselines: ReAct achieves 27.4% success on HotpotQA compared to 20.6% for chain-of-thought prompting, representing a 33% relative improvement [Yao et al., 2022]. However, these gains emerged through extensive optimization against these specific evaluation sets.

### 4.2 Code Generation Workflows

Code generation tasks present particularly clear examples of workflow overfitting. Benchmarks like HumanEval [Chen et al., 2021] and MBPP [Austin et al., 2021] have become central targets for LLM development, with teams iteratively refining their approaches against these specific problems.

**Development Process Analysis**: Modern coding workflows undergo extensive optimization cycles:

- Tool integration strategies refined based on benchmark performance
- Error handling approaches optimized for common benchmark failure modes
- Multi-step reasoning patterns tuned to handle benchmark-specific problem structures
- Code execution and debugging workflows designed around benchmark evaluation criteria

**Systematic Contamination Evidence**: The LessLeak-Bench study [Zhou et al., 2025] provides compelling evidence of widespread contamination across software engineering benchmarks. The study analyzed 83 benchmarks and found average leakage ratios of 4.8%, 2.8%, and 0.7% for Python, Java, and C/C++ benchmarks respectively. However, specific benchmarks showed much higher contamination rates, with QuixBugs exhibiting 100% leakage and BigCloneBench showing 55.7% leakage.

**Performance Impact**: The study demonstrates that data leakage has substantial impact on LLM evaluation, with contaminated models showing inflated performance that does not generalize to truly novel programming challenges.

### 4.3 Autonomous Agent Systems

Autonomous agent frameworks like AutoGPT represent complex multi-component systems that are particularly susceptible to overfitting due to their iterative development processes and community-driven optimization.

**Development Challenges**: Agent systems face unique evaluation challenges:

- Benchmarks developed concurrently with the systems they evaluate
- Community-driven optimization leading to distributed overfitting effects
- Performance claims based on evaluation sets that guided development decisions
- Informal evaluation criteria that evolve based on system capabilities

**Generalization Gaps**: Despite reported success on development benchmarks, deployed agent systems frequently exhibit significant performance degradation when encountering novel scenarios that differ from their optimization targets [Zheng et al., 2023, Garg et al., 2025]. This pattern suggests that benchmark performance may reflect sophisticated pattern matching rather than genuine autonomous reasoning capabilities.

### 4.4 Chemistry and Materials Science

Scientific applications of LLM workflows present particularly high stakes for the overfitting problem. Recent work in chemistry has developed sophisticated benchmarks like ChemBench [Mirza et al., 2024], containing over 2,700 question-answer pairs designed to evaluate chemical knowledge and reasoning. While these benchmarks represent important advances in evaluation methodology, they also create new targets for optimization.

**Domain-Specific Overfitting Risks**: Chemical reasoning workflows often involve:

- Multi-step synthesis planning optimized against known reaction databases
- Property prediction systems trained and validated on established chemical datasets
- Literature analysis tools refined using benchmark chemical papers and abstracts
- Safety assessment frameworks tuned against standardized hazard classification systems

**Reproducibility Standards in Chemistry**: The chemistry community has established rigorous standards for experimental reproducibility, but these standards have not yet been adapted for AI-assisted chemical research. Unlike traditional chemistry experiments, where failed replications are clearly identifiable, overfitted AI systems may produce plausible but incorrect results that are difficult to detect without expert domain knowledge.

### 4.5 Physics and Engineering Applications

Physics applications face similar challenges, with benchmarks targeting graduate-level physics problems. Engineering applications often require reasoning across multiple domains simultaneously, but systems optimized against such benchmarks may develop sophisticated pattern-matching capabilities that fail when confronted with truly novel interdisciplinary problems [Zhang et al., 2025b,a].

**Multi-Domain Scientific Reasoning**: Scientific applications often require reasoning across multiple domains simultaneously [Cui et al., 2025], creating additional opportunities for overfitting to develop on multi-faceted benchmark tasks.

### 4.6 Biomedical and Healthcare Applications

Healthcare applications represent the highest-stakes domain for AI system reliability. Biomedical LLM workflows increasingly target specialized benchmarks for clinical reasoning, drug discovery, and diagnostic assistance [Zhu et al., 2025, Tang et al., 2025].

**Critical Safety Implications**: In healthcare contexts, overfitted performance can have severe consequences:

- Diagnostic systems optimized against medical benchmark datasets may miss novel disease presentations
- Drug discovery workflows trained on established chemical databases may fail to identify truly innovative therapeutic approaches
- Clinical decision support systems refined using benchmark cases may provide inappropriate recommendations for edge cases

**Regulatory and Compliance Challenges**: Healthcare AI systems must meet strict regulatory standards for safety and efficacy. However, current evaluation practices may not adequately distinguish between genuine clinical reasoning and sophisticated pattern matching against medical benchmarks [Bedi et al., 2025, Mehandru et al., 2025].

## 5 Comprehensive Framework for Addressing Workflow Overfitting

Building on established ML evaluation best practices, we propose a comprehensive approach to address workflow overfitting that extends beyond traditional model evaluation to encompass the entire development ecosystem.

### 5.1 Rigorous Data Separation Methodologies

**Fundamental Train/Validation/Test Separation**: LLM workflow development must adopt the foundational ML principle of strict data separation. Training data is used for initial system development, validation data guides iterative refinement and hyperparameter optimization, and test data remains completely isolated until final evaluation. This separation must be maintained throughout the entire workflow development process, including prompt engineering, tool integration, and architectural decisions.

**Temporal Isolation Protocols**: Evaluation benchmarks must use data that postdates workflow development completion. This temporal separation prevents both direct benchmark exposure and indirect contamination through community knowledge dissemination. The temporal gap must account for publication delays and community adoption cycles, ensuring no development decisions can be influenced by evaluation content.

**Cross-Domain Transfer Evaluation**: Systems developed in one domain should be evaluated on structurally analogous tasks in different domains. For instance, reasoning workflows optimized on historical datasets should be tested on scientific reasoning problems with similar logical structures but distinct knowledge bases. This tests genuine transferable capabilities rather than domain-specific pattern recognition.

**Hierarchical Holdout Architecture**: Implement nested isolation strategies at multiple levels: component-level evaluation using standard ML holdout practices, workflow-level assessment on integration benchmarks isolated from development, and system-level evaluation on deployment scenarios that differ systematically from development contexts.

### 5.2 Methodology Transparency Requirements

Current LLM workflow development lacks the systematic documentation that enables contamination detection and replication. Unlike traditional ML where training procedures are explicitly documented, workflow optimization often occurs through informal iteration cycles.

**Development History Documentation**: Comprehensive logging must capture all evaluation data interactions, including datasets accessed during prompt engineering, optimization iterations against benchmarks, architectural decisions guided by performance feedback, and community knowledge sources consulted.

**Contamination Auditing**: Automated analysis must detect potential overlap between development resources and evaluation sets, including computational analysis of data intersection, documentation of influential published work, and assessment of community knowledge transfer.

**Preregistration Protocols**: Adapting practices from experimental psychology and medical research, teams must declare evaluation benchmarks before development, specify success criteria in advance, and commit to reporting results regardless of outcome.

### 5.3 Institutional and Ecosystem-Level Interventions

**Benchmark Lifecycle Management**: Establish systematic protocols for benchmark retirement and replacement, including monitoring community, implementing retirement triggers based on performance saturation, and developing principled approaches for creating replacement benchmarks.

**Independent Evaluation Infrastructure**: Create community-managed evaluation services maintaining truly independent benchmarks inaccessible to development teams until final evaluation. This requires sustained funding and governance structures resistant to commercial and academic pressures.

**Research Incentive Realignment**: Conference and journal policies must require comprehensive methodology transparency. Recognition systems should value negative results and replication studies equally with novel contributions. Funding agencies must prioritize evaluation methodology alongside technical innovation.

## 6 Why This Matters for Scientific Applications

The overfitting problem is particularly concerning for scientific applications. Scientific research demands reproducible results, but if LLM workflows are optimized against the same benchmarks they're evaluated on, reported performance may not generalize to real scientific problems.

Overfitted workflows may perform well on known benchmarks but fail on novel scientific challenges, leading to misplaced confidence in AI capabilities for scientific discovery. If reported performance is inflated due to overfitting, research funding and effort may be misdirected toward approaches that don't actually advance scientific capability.

The stakes are particularly high in scientific domains where accuracy and reliability are paramount, and where the cost of false confidence can impede genuine scientific progress.

# 7 Limitations

This position paper has several important limitations that should be acknowledged when interpreting our arguments and recommendations.

**Lack of Empirical Validation**: Our analysis relies primarily on theoretical reasoning and examination of existing literature rather than systematic empirical studies. While we cite evidence from studies like LessLeak-Bench [Zhou et al., 2025], we have not conducted controlled experiments to directly measure workflow overfitting or validate our proposed solutions.

**Limited Access to Proprietary Systems**: Our case studies focus on publicly documented systems and open benchmarks. Many commercial LLM workflows involve proprietary development processes that are not accessible for analysis, limiting our ability to assess the full scope of the problem across the industry.

**Implementation Feasibility**: While we propose comprehensive solutions, we acknowledge significant practical barriers including costs and coordination challenges. The feasibility of these recommendations remains untested.

**Boundary Definition**: The distinction between legitimate iterative development and problematic overfitting is not always clear-cut. Our framework does not provide precise operational definitions for this boundary, which could lead to either overly restrictive or insufficient practices.

# 8 Discussion and Implications

## 8.1 Systemic Nature of Workflow Overfitting

Unlike traditional ML overfitting, which typically affected individual models, LLM workflow overfitting operates at an unprecedented scale with systemic consequences. When the research community collectively optimizes against the same benchmarks, the cumulative effect creates systematic bias that permeates entire research directions. This represents "ecosystem-level overfitting" where distributed optimization across hundreds of research teams amplifies the problem beyond classical overfitting scenarios.

The interconnected nature of modern AI research exacerbates this through rapid technique propagation via preprints and code repositories. Competitive pressure to achieve state-of-the-art results on established leaderboards incentivizes optimization for known evaluation sets rather than developing genuinely novel capabilities, creating a feedback loop where benchmark performance becomes divorced from real-world utility.

## 8.2 Epistemological and Economic Implications

Current practices fundamentally conflate innovation with evaluation, undermining the epistemological foundations of AI research. True scientific evaluation requires independence from development—a principle fundamental to empirical inquiry. When evaluation data contaminates development decisions, resulting performance metrics cannot serve as valid evidence for genuine capability advances.

This confusion has profound economic implications as organizations increasingly rely on benchmark performance to guide technology adoption and investment decisions. Overfitted results can lead to substantial resource misallocation, with companies investing heavily in systems that perform well on benchmarks but fail in production environments.

## 8.3 High-Stakes Domain Risks

As LLM workflows become more sophisticated and deployed in critical applications—from drug discovery to climate modeling—the cost of overfitted performance becomes exponentially higher. Scientific discovery relies on generating novel insights and identifying previously unknown patterns,

capabilities fundamentally undermined when systems optimize primarily for existing datasets that may not capture natural phenomena's full complexity.

The risk is particularly severe where ground truth is difficult to establish or system failures may not be immediately apparent. In chemistry, for example, AI systems excelling at predicting molecular properties on benchmark datasets might fail catastrophically with novel compounds, potentially delaying scientific breakthroughs or causing harmful outcomes if trusted beyond actual capabilities.

The scientific and commercial communities cannot afford to repeat machine learning's early methodological mistakes, particularly when applications involve high-stakes scientific discovery and decision-making where system failures can have serious real-world consequences.

# 9 Conclusion

The LLM research community stands at a critical juncture. The sophistication of modern workflows has obscured a fundamental methodological problem: we are systematically reporting results on data that has been used for optimization. This mirrors the overfitting crisis that plagued early machine learning research.

Unlike traditional ML, where overfitting affected individual models, LLM workflow overfitting can contaminate entire research directions and mislead the community about actual capabilities. This is particularly concerning for scientific applications, where accuracy and reliability are paramount.

The solution requires collective action: establishing strict data separation protocols, demanding transparency in development processes, and creating truly independent evaluation resources. The ML community learned these lessons decades ago. The LLM community must learn them now, before the credibility of AI research is further undermined.

We call on the research community to adopt rigorous evaluation standards that separate development from testing, just as was necessary in traditional ML. Only through such methodological rigor can we ensure that reported advances represent genuine progress rather than sophisticated forms of overfitting.

# References


Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Suhana Bedi, Yixing Jiang, Philip Chung, Sanmi Koyejo, and Nigam Shah. Fidelity of medical reasoning in large language models. *JAMA Network Open*, 8(8):e2526021–e2526021, 2025.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. *International conference on machine learning*, pages 1006–1014, 2015.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Nurit Cohen-Inger, Yehonatan Elisha, Bracha Shapira, Lior Rokach, and Seffi Cohen. Forget what you know about llms evaluations–llms are like a chameleon. *arXiv preprint arXiv:2502.07445*, 2025.

Hao Cui, Zahra Shamsi, Gowoon Cheon, Xuejian Ma, Shutong Li, Maria Tikhanovskaya, Peter Norgaard, Nayantara Mudur, Martyna Plomecka, Paul Raccuglia, et al. Curie: Evaluating llms on multitask scientific long context understanding and reasoning. *arXiv preprint arXiv:2503.13517*, 2025.

Jesse Dodge, Maarten Sap, Ana Marasović, William Agnew, Gabriel Ilharco, Dirk Groeneveld, Margaret Mitchell, and Matt Gardner. Documenting large webtext corpora: A case study on the colossal clean crawled corpus. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1286–1305, 2021.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015.

Divyansh Garg, Shaun VanWeelden, Diego Caples, Andis Draguns, Nikil Ravi, Pranav Putta, Naman Garg, Tomas Abraham, Michael Lara, Federico Lopez, et al. Real: Benchmarking autonomous agents on deterministic simulations of real websites. *arXiv preprint arXiv:2504.11543*, 2025.

Shahriar Golchin and Mihai Surdeanu. Time travel in llms: Tracing data contamination in large language models. *arXiv preprint arXiv:2308.08493*, 2023.

Trevor Hastie, Robert Tibshirani, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. *Proceedings of the AAAI conference on artificial intelligence*, 32(1), 2018.

John PA Ioannidis. Why most published research findings are false. *PLoS medicine*, 2(8):e124, 2005.

Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145, 1995.

Zachary C Lipton and Jacob Steinhardt. Troubling trends in machine learning scholarship: Some ml papers suffer from flaws that could mislead the public and stymie future research. *Communications of the ACM*, 62(6):45–53, 2019.

Nikita Mehandru, Niloufar Golchini, David Bamman, Travis Zack, Melanie F Molina, and Ahmed Alaa. Er-reason: A benchmark dataset for llm-based clinical reasoning in the emergency room. *arXiv preprint arXiv:2505.22919*, 2025.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the role of demonstrations: What makes in-context learning work? *arXiv preprint arXiv:2202.12837*, 2022.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, et al. Are large language models superhuman chemists? *arXiv preprint arXiv:2404.01475*, 2024.

Randal S Olson, William La Cava, Zairah Mustahsan, Akshay Varik, and Jason H Moore. Pmlb: a large benchmark suite for machine learning evaluation and comparison. *BioData mining*, 10(1): 1–13, 2017.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. True few-shot learning with language models. *Advances in neural information processing systems*, 34:11054–11070, 2021.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Tool learning with foundation models. *arXiv preprint arXiv:2304.08354*, 2023.

Sebastian Raschka. Model evaluation, model selection, and algorithm selection in machine learning. *arXiv preprint arXiv:1811.12808*, 2018.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *International conference on machine learning*, pages 5389–5400, 2019.

Anna Rogers, Timothy Baldwin, and Kobi Leins. Changing the world by changing the data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2182–2194, 2021.

Timo Schick and Hinrich Schütze. Exploiting cloze questions for few-shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*, 2020.

Shuo Shi, Xiangyu Ku, Xiang Yue, Tianle Li, Yunhua Li, and William Yang Wang. Rethinking benchmark and contamination for language models with rephrased samples. *arXiv preprint arXiv:2311.04850*, 2024.

Mervyn Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133, 1974.

Xiangru Tang, Daniel Shao, Jiwoong Sohn, Jiapeng Chen, Jiayi Zhang, Jinyu Xiang, Fang Wu, Yilun Zhao, Chenglin Wu, Wenqi Shi, et al. Medagentsbench: Benchmarking thinking models and agent frameworks for complex medical reasoning. *arXiv preprint arXiv:2503.07459*, 2025.

Sudhir Varma and Richard Simon. Bias in error estimation when using cross-validation for model selection. *BMC bioinformatics*, 7(1):1–8, 2006.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *arXiv preprint arXiv:2308.11432*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2023.

Xinyu Zhang, Yuxuan Dong, Yanrui Wu, Jiaxing Huang, Chengyou Jia, Basura Fernando, Mike Zheng Shou, Lingling Zhang, and Jun Liu. Physreason: A comprehensive benchmark towards physics-based reasoning. *arXiv preprint arXiv:2502.12054*, 2025a.

Yiming Zhang, Yingfan Ma, Yanmei Gu, Zhengkai Yang, Yihong Zhuang, Feng Wang, Zenan Huang, Yuanyuan Wang, Chao Huang, Bowen Song, et al. Abench-physics: Benchmarking physical reasoning in llms via high-difficulty and dynamic physics problems. *arXiv preprint arXiv:2507.04766*, 2025b.

Xinyue Zheng, Haowei Lin, Kaichen He, Zihao Wang, Zilong Zheng, QIANG FU, Haobo Fu, and Yitao Liang. Towards evaluating generalist agents: An automated benchmark in open world. 2023.

Xin Zhou, Kisub Li, Yunbo Lyu, Ming Wen, Beijun Wang, Zhou Yang, Zhiqiang Wang, Xiaodong Luo, Li Li, and Yang Li. Lessleak-bench: A first investigation of data leakage in llms across 83 software engineering benchmarks. *arXiv preprint arXiv:2502.06215*, 2025.

Yakun Zhu, Zhongzhen Huang, Linjie Mu, Yutong Huang, Wei Nie, Jiaji Liu, Shaoting Zhang, Pengfei Liu, and Xiaofan Zhang. Diagnosisarena: Benchmarking diagnostic reasoning for large language models. *arXiv preprint arXiv:2505.14107*, 2025.

# Agents4Science AI Involvement Checklist

1. **Hypothesis development**: The research topic and hypothesis that LLM workflows suffer from systematic overfitting similar to early ML practices was jointly developed, with the AI agent providing the detailed conceptual framework and historical parallels.

   Answer: **[B] Mostly human, assisted by AI**

   Explanation: The human co-author identified the core problem and made the connection to historical ML overfitting. The AI agent developed the detailed hypothesis, literature connections, and theoretical framework for understanding the problem.

2. **Experimental design and implementation**: This is a position paper that does not include empirical experiments. The analysis relies on synthesizing existing literature and making methodological arguments.

   Answer: **[NA] Not Applicable**

   Explanation: No experiments were conducted. The paper makes its argument through literature analysis, theoretical reasoning, and methodological critique of current practices.

3. **Analysis of data and interpretation of results**: The paper analyzes existing literature and research practices rather than experimental data. The AI agent conducted literature search and synthesis.

   Answer: **[C] Mostly AI, assisted by human**

   Explanation: The AI agent conducted web searches for relevant literature, identified key papers on data contamination and benchmark leakage, and synthesized findings to support the core arguments. Human provided guidance on focus areas.

4. **Writing**: The AI agent was responsible for the majority of the writing, including structure, arguments, and academic formatting.

   Answer: **[C] Mostly AI, assisted by human**

   Explanation: The AI agent wrote the complete paper including abstract, introduction, literature review, arguments, and conclusions. The human co-author provided feedback, guidance on emphasis areas, and editorial input.

5. **Observed AI Limitations**: The AI agent required guidance on specific examples and emphasis areas, and needed human oversight to ensure arguments remained grounded and credible.

   Description: The AI agent occasionally needed direction on which aspects of the overfitting problem to emphasize most strongly. The human co-author's role was crucial in keeping the argument focused and ensuring it addressed the most important methodological concerns in the field.

# Agents4Science Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: **[Yes]**

   Justification: The abstract and introduction clearly state this is a position paper arguing that LLM workflows suffer from systematic overfitting, with claims appropriately scoped to methodological critique rather than empirical findings.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: **[Yes]**

   Justification: Section 7 discusses the scope of the argument and acknowledges this is a methodological position paper without empirical validation of proposed solutions.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: **[NA]**

   Justification: This is a position paper that makes methodological arguments rather than theoretical claims requiring formal proofs.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results?

   Answer: **[NA]**

   Justification: This paper does not include experiments. It is a methodological position paper based on literature analysis and reasoned argument.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code?

   Answer: **[NA]**

   Justification: No experiments or code are involved in this position paper.

6. **Experimental setting/details**

   Question: Does the paper specify all training and test details necessary to understand the results?

   Answer: **[NA]**

   Justification: This paper does not include experiments.

7. **Experiment statistical significance**

   Question: Does the paper report error bars or statistical significance information?

   Answer: **[NA]**

   Justification: No experiments are conducted in this position paper.

8. **Experiments compute resources**

   Question: Does the paper provide sufficient information on computer resources needed?

   Answer: **[NA]**

   Justification: No experiments requiring computational resources were conducted.

9. **Code of ethics**

   Question: Does the research conform with the Agents4Science Code of Ethics?

   Answer: **[Yes]**

Justification: This methodological critique aims to improve research practices and scientific integrity, fully conforming with ethical research standards.

10. **Broader impacts**

Question: Does the paper discuss potential positive and negative societal impacts?

Answer: **[Yes]**

Justification: Throughout the paper, we discuss both the positive impact of addressing workflow overfitting (improving research credibility, advancing genuine scientific capabilities) and the negative consequences of failing to address it (misplaced confidence in AI systems, potential safety risks in high-stakes domains like healthcare, misdirection of research resources).