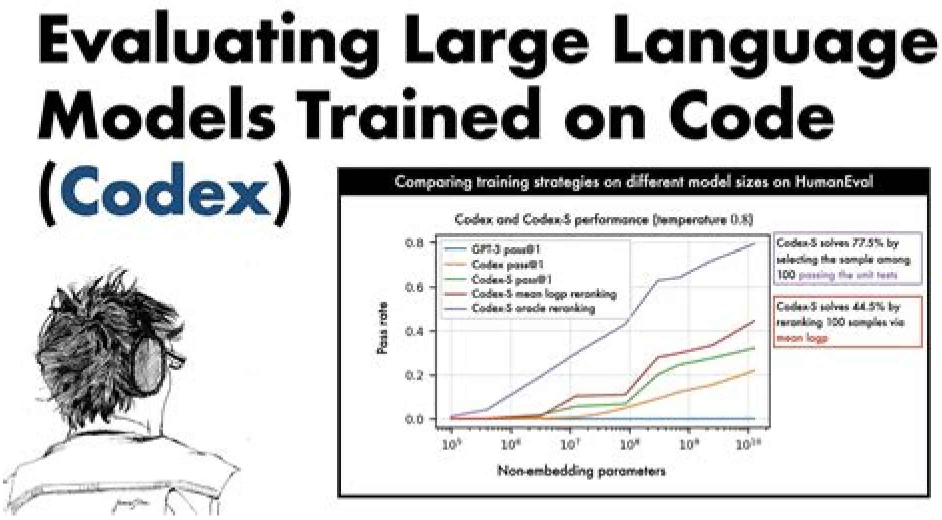
# Evaluating Large Language Models Trained On Code




## Evaluating Large Language Models Trained on Code: A Comprehensive Guide

The world of software development is rapidly changing, with Large Language Models (LLMs) trained on vast datasets of code emerging as powerful tools. But how do we effectively evaluate these models, ensuring they meet the stringent requirements of reliable code generation and comprehension? This comprehensive guide dives deep into the methodologies and metrics used to assess LLMs specifically trained on code, providing you with the knowledge to critically evaluate their capabilities and limitations. We'll explore various evaluation strategies, highlighting their strengths and weaknesses, ultimately empowering you to make informed decisions when selecting the right LLM for your coding needs.

## H2: Understanding the Unique Challenges of Evaluating Code-Trained LLMs

Unlike LLMs trained primarily on text, those focused on code present unique evaluation challenges. Simply measuring fluency or coherence isn't enough. Code must be correct, efficient, and robust. An LLM might generate syntactically correct code that fails to execute properly or produces unexpected results. This necessitates a multi-faceted approach to evaluation, considering several crucial aspects:

### H3: Beyond Syntax: Assessing Code Functionality

Evaluating the functionality of code generated by an LLM requires rigorous testing. This goes beyond simply checking if the code compiles. We need to ensure it produces the expected output under various conditions, including edge cases and error handling scenarios. Automated testing frameworks, unit tests, and integration tests are crucial tools in this process. Furthermore, manual review by experienced programmers remains vital to identify subtle bugs or design flaws that automated tests might miss.

### H3: Measuring Code Efficiency and Readability

Efficiency is paramount in software development. An LLM might produce functional code, but if it's inefficient, it's not a good solution. Evaluation should include measuring metrics like execution time, memory usage, and algorithmic complexity. Readability is equally important; code should be easy for humans to understand and maintain. Metrics such as code length, cyclomatic complexity, and adherence to coding style guides can be employed to assess readability.

### H3: Benchmarking Against Existing Models and Human Performance

Comparing the performance of a code-trained LLM against established benchmarks and human programmers offers valuable insights. Popular benchmarks like HumanEval and MBPP provide standardized datasets for evaluating code generation capabilities. Comparing the LLM's performance against the scores achieved by human programmers helps establish a baseline and identify areas for improvement.

## H2: Key Metrics for Evaluating Code-Trained LLMs

Several key metrics help quantify the performance of LLMs trained on code. These metrics often complement each other, providing a holistic view of the model's capabilities:

### H4: Accuracy:

This measures the percentage of correctly generated code snippets that produce the expected output. Accuracy is a crucial metric but should be considered alongside other factors like efficiency and readability.

#### H4: Precision and Recall:

In the context of code completion or suggestion tasks, precision refers to the proportion of correct suggestions among all suggestions made, while recall refers to the proportion of correct suggestions retrieved out of all the correct suggestions that exist.

#### H4: Execution Time and Memory Usage:

These metrics directly assess the efficiency of the generated code. Lower execution time and memory usage indicate higher efficiency.

#### H4: Code Style and Readability Metrics:

These evaluate the adherence to coding style guidelines and the overall readability of the generated code. Tools like SonarQube can provide valuable insights into code quality.

## H2: The Role of Human Evaluation in Assessing Code Quality

While automated metrics provide quantitative data, human evaluation remains essential. Experienced programmers can assess aspects of code quality that are difficult to capture with automated metrics, such as code design, maintainability, and overall elegance. Human evaluators can also identify subtle bugs or unexpected behaviours that might be missed by automated testing.

## H2: The Future of Evaluating Code-Trained LLMs

The field of LLM evaluation is constantly evolving. As LLMs become more sophisticated, new evaluation techniques and metrics will be needed to capture their capabilities fully. Research is ongoing into more robust and comprehensive evaluation strategies that consider the ethical implications of AI-generated code. The focus is shifting towards evaluating not just the correctness of the code but also its security, robustness, and potential biases.

## Conclusion

Evaluating large language models trained on code requires a multi-pronged approach that combines automated metrics with human expertise. By employing a range of techniques and paying attention to various aspects like functionality, efficiency, and readability, we can gain a comprehensive understanding of an LLM's capabilities and limitations. This allows for informed decisions regarding their deployment in real-world software development tasks, ultimately paving the way for more reliable and efficient software engineering practices.

## FAQs

1. What are some popular open-source tools for evaluating code-trained LLMs? Several open-source projects provide tools and datasets for evaluating LLMs. These include projects that offer automated testing frameworks, code style checkers, and benchmark datasets. Look for repositories on platforms like GitHub focusing on LLM evaluation.

2. How can I incorporate LLM evaluation into my software development workflow? Integrate evaluation into your CI/CD pipeline. Automate testing and code analysis, and include human review as part of the process.

3. What are the ethical considerations in evaluating LLMs trained on code? Consider potential biases in the training data and the generated code, as well as the potential for misuse of the technology. Ensure fairness and transparency in the evaluation process.

4. Are there any specific datasets available for benchmarking code generation models? Yes, several benchmark datasets, such as HumanEval and MBPP (Massive Bench Press Programming), provide standardized datasets for evaluating code generation capabilities.

5. How can I improve the performance of a code-trained LLM if its evaluation results are unsatisfactory? Fine-tuning the model with additional high-quality code, adjusting hyperparameters, and incorporating feedback from human evaluators are crucial strategies to improve its performance.

## Evaluating Large Language Models Trained on Code: A Comprehensive Guide

The explosion of Large Language Models (LLMs) has revolutionized numerous fields, and their application to code is particularly exciting. But how do we effectively evaluate these powerful tools?

Simply stating that an LLM "works" is insufficient. This comprehensive guide delves into the intricacies of assessing LLMs trained on code, exploring various metrics, methodologies, and the challenges involved. We'll equip you with the knowledge to critically analyze and understand the capabilities—and limitations—of these groundbreaking models.

## Understanding the Nuances of Code-Trained LLMs

Before diving into evaluation, let's clarify what we're dealing with. LLMs trained on code aren't just regurgitating snippets; they learn complex programming concepts, syntax, and even algorithmic patterns. They can generate code, translate between programming languages, debug existing code, and even answer questions about code functionality. However, the quality and reliability of these capabilities vary significantly across different models. This necessitates rigorous evaluation.

## Key Metrics for Evaluating Code-Generating LLMs

Evaluating LLMs trained on code requires a multifaceted approach, going beyond simple accuracy. We need to consider several crucial metrics:

#### 1. Accuracy:

This is the foundational metric – does the generated code correctly solve the given problem? However, "correctness" can be nuanced. A perfectly functional solution might still be inefficient or poorly written. Therefore, accuracy must be considered within the broader context of other metrics.

#### 2. Correctness and Completeness:

This metric takes the nuance mentioned above into consideration. Does the code not only work correctly, but is it also fully functional and complete? Does it handle all edge cases as intended?

#### 3. Efficiency:

Efficient code minimizes resource consumption (time and memory). An accurate solution that's incredibly inefficient is less valuable than a slightly less accurate but significantly more efficient one. This often involves analyzing time complexity and space complexity.

#### 4. Readability and Maintainability:

Human developers will interact with the generated code. Therefore, readability and maintainability are crucial. Well-structured, commented, and easily understandable code is far more valuable than a complex, obfuscated solution, even if both achieve the same result.

#### 5. Style and Consistency:

Adherence to coding style guidelines and consistent formatting improves code maintainability and collaboration. Evaluation should assess the model's ability to generate code that conforms to specified style standards.

## Methodologies for Evaluation

Several approaches are used to evaluate code-generating LLMs:

#### 1. Human Evaluation: Expert programmers assess the generated code for accuracy, efficiency, readability, and adherence to best practices. This approach is subjective but crucial for capturing nuances that automated metrics might miss.

#### 2. Automated Metrics: Tools and techniques automatically measure aspects like code correctness (through unit testing), complexity (using cyclomatic complexity), and adherence to style guidelines (using linters).

#### 3. Benchmark Datasets: Standardized datasets containing coding tasks and solutions provide a consistent basis for comparison across different LLMs. These benchmarks help quantify performance and facilitate objective comparisons. Examples include HumanEval and MBPP.

#### 4. A/B Testing: Comparing the performance of different LLMs or versions of the same LLM on the same tasks allows for direct performance comparisons.

## Challenges in Evaluating Code-Generating LLMs

Evaluating these LLMs presents several challenges:

Subjectivity: Assessing readability and maintainability often involves subjective judgments.
Ambiguity: Natural language prompts can be ambiguous, leading to different interpretations and thus different code outputs.
Scalability: Evaluating LLMs comprehensively often requires substantial computational resources and human expertise.
Bias and Fairness: LLMs can inherit biases present in their training data, leading to unfair or discriminatory outcomes. Evaluation needs to address this.

## Conclusion

Evaluating Large Language Models trained on code is a complex but crucial undertaking. By combining multiple metrics, methodologies, and addressing the inherent challenges, we can gain a much more comprehensive understanding of these models' capabilities and limitations. This allows for responsible development, deployment, and ultimately, maximizing their benefits while mitigating potential risks. Continuous research and development in evaluation techniques are essential for the continued advancement of this rapidly evolving field.

## FAQs

1. What are some popular benchmark datasets for evaluating code-generating LLMs? Popular datasets include HumanEval, MBPP, and CodeXGLUE. These offer a variety of programming languages and coding tasks.

2. How can I measure the efficiency of code generated by an LLM? You can use automated metrics such as cyclomatic complexity to assess code complexity, and then perform runtime analysis to observe the execution time and memory usage. Profiling tools can be invaluable in this process.

3. What role does human evaluation play in assessing code quality? Human evaluation is crucial for assessing subjective aspects like readability, maintainability, and overall code style, which automated metrics struggle to capture fully.

4. What are the ethical considerations in evaluating code-generating LLMs? Ethical considerations include ensuring fairness, mitigating bias, and addressing potential security vulnerabilities in generated code. Robust testing and careful evaluation are crucial to mitigate these risks.

5. How can I contribute to the development of better evaluation methods for code-generating LLMs? You can contribute by participating in research projects, developing new evaluation metrics, creating and sharing benchmark datasets, and actively participating in the open-source community dedicated to LLM evaluation.

**evaluating large language models trained on code:** *Hands-On Large Language Models* Jay Alammar, Maarten Grootendorst, 2024-09-11 AI has acquired startling new language capabilities in just the past few years. Driven by the rapid advances in deep learning, language AI systems are able to write and understand text better than ever before. This trend enables the rise of new features, products, and entire industries. With this book, Python developers will learn the practical tools and concepts they need to use these capabilities today. You'll learn how to use the power of pre-trained large language models for use cases like copywriting and summarization; create semantic search systems that go beyond keyword matching; build systems that classify and cluster text to enable scalable understanding of large amounts of text documents; and use existing libraries and pre-trained models for text classification, search, and clusterings. This book also shows you how to: Build advanced LLM pipelines to cluster text documents and explore the topics they belong to Build semantic search engines that go beyond keyword search with methods like dense retrieval and rerankers Learn various use cases where these models can provide value Understand the architecture of underlying Transformer models like BERT and GPT Get a deeper understanding of how LLMs are trained Understanding how different methods of fine-tuning optimize LLMs for specific applications (generative model fine-tuning, contrastive fine-tuning, in-context learning, etc.)

**evaluating large language models trained on code:** Network Simulation and Evaluation Zhaoquan Gu,

**evaluating large language models trained on code:** *ECAI 2023* K. Gal, A. Nowé, G.J. Nalepa, 2023-10-18 Artificial intelligence, or AI, now affects the day-to-day life of almost everyone on the planet, and continues to be a perennial hot topic in the news. This book presents the proceedings of ECAI 2023, the 26th European Conference on Artificial Intelligence, and of PAIS 2023, the 12th Conference on Prestigious Applications of Intelligent Systems, held from 30 September to 4 October 2023 and on 3 October 2023 respectively in Kraków, Poland. Since 1974, ECAI has been the premier venue for presenting AI research in Europe, and this annual conference has become the place for researchers and practitioners of AI to discuss the latest trends and challenges in all subfields of AI, and to demonstrate innovative applications and uses of advanced AI technology. ECAI 2023 received 1896 submissions – a record number – of which 1691 were retained for review, ultimately resulting in an acceptance rate of 23%. The 390 papers included here, cover topics including machine learning, natural language processing, multi agent systems, and vision and knowledge representation and reasoning. PAIS 2023 received 17 submissions, of which 10 were accepted after a rigorous review process. Those 10 papers cover topics ranging from fostering better working environments, behavior modeling and citizen science to large language models and neuro-symbolic applications, and are also included here. Presenting a comprehensive overview of current research and developments in AI, the book will be of interest to all those working in the field.

**evaluating large language models trained on code:** *Computational Science – ICCS 2023* Jiří Mikyška, Clélia de Mulatier, Maciej Paszynski, Valeria V. Krzhizhanovskaya, Jack J. Dongarra, Peter M.A. Sloot, 2023-06-30 The five-volume set LNCS 14073-14077 constitutes the proceedings of the 23rd International Conference on Computational Science, ICCS 2023, held in Prague, Czech Republic, during July 3-5, 2023. The total of 188 full papers and 94 short papers presented in this book set were carefully reviewed and selected from 530 submissions. 54 full and 37 short papers were accepted to the main track; 134 full and 57 short papers were accepted to the workshops/thematic tracks. The theme for 2023, Computation at the Cutting Edge of Science, highlights the role of Computational Science in assisting multidisciplinary research. This conference was a unique event focusing on recent developments in scalable scientific algorithms, advanced software tools; computational grids; advanced numerical methods; and novel application areas. These innovative novel models, algorithms, and tools drive new science through efficient application in physical systems, computational and systems biology, environmental systems, finance, and others.

**evaluating large language models trained on code: Large Language Models** Uday Kamath, Kevin Keenan, Garrett Somers, Sarah Sorenson, 2024 Large Language Models (LLMs) have emerged as a cornerstone technology, transforming how we interact with information and redefining the boundaries of artificial intelligence. LLMs offer an unprecedented ability to understand, generate, and interact with human language in an intuitive and insightful manner, leading to transformative applications across domains like content creation, chatbots, search engines, and research tools. While fascinating, the complex workings of LLMs -- their intricate architecture, underlying algorithms, and ethical considerations -- require thorough exploration, creating a need for a comprehensive book on this subject. This book provides an authoritative exploration of the design, training, evolution, and application of LLMs. It begins with an overview of pre-trained language models and Transformer architectures, laying the groundwork for understanding prompt-based learning techniques. Next, it dives into methods for fine-tuning LLMs, integrating reinforcement learning for value alignment, and the convergence of LLMs with computer vision, robotics, and speech processing. The book strongly emphasizes practical applications, detailing real-world use cases such as conversational chatbots, retrieval-augmented generation (RAG), and code generation. These examples are carefully chosen to illustrate the diverse and impactful ways LLMs are being applied in various industries and scenarios. Readers will gain insights into operationalizing and deploying LLMs, from implementing modern tools and libraries to addressing challenges like bias and ethical implications. The book also introduces the cutting-edge realm of multimodal LLMs that can process audio, images, video, and robotic inputs. With hands-on tutorials for applying LLMs to natural language tasks, this thorough guide equips readers with both theoretical knowledge and practical skills for leveraging the full potential of large language models. This comprehensive resource is appropriate for a wide audience: students, researchers and academics in AI or NLP, practicing data scientists, and anyone looking to grasp the essence and intricacies of LLMs.

**evaluating large language models trained on code: Proceedings of International Conference on Recent Innovations in Computing** Zoltán Illés,

**evaluating large language models trained on code:** *Large Language Models in Cybersecurity* Andrei Kucharavy, 2024 This open access book provides cybersecurity practitioners with the knowledge needed to understand the risks of the increased availability of powerful large language models (LLMs) and how they can be mitigated. It attempts to outrun the malicious attackers by anticipating what they could do. It also alerts LLM developers to understand their work's risks for cybersecurity and provides them with tools to mitigate those risks. The book starts in Part I with a general introduction to LLMs and their main application areas. Part II collects a description of the most salient threats LLMs represent in cybersecurity, be they as tools for cybercriminals or as novel attack surfaces if integrated into existing software. Part III focuses on attempting to forecast the exposure and the development of technologies and science underpinning LLMs, as well as macro levers available to regulators to further cybersecurity in the age of LLMs. Eventually, in Part IV, mitigation techniques that should allowsafe and secure development and deployment of LLMs are presented. The book concludes with two final chapters in Part V, one speculating what a secure design and integration of LLMs from first principles would look like and the other presenting a summary of the duality of LLMs in cyber-security. This book represents the second in a series published by the Technology Monitoring (TM) team of the Cyber-Defence Campus. The first book entitled Trends in Data Protection and Encryption Technologies appeared in 2023. This book series provides technology and trend anticipation for government, industry, and academic decision-makers as well as technical experts.

**evaluating large language models trained on code:** *Building AI Intensive Python Applications* Rachelle Palmer, Ben Perlmutter, Ashwin Gangadhar, Nicholas Larew, Sigfrido Narváez, Thomas Rueckstiess, Henry Weller, Richmond Alake, Shubham Ranjan, 2024-09-06 Master retrieval-augmented generation architecture and fine-tune your AI stack, along with discovering real-world use cases and best practices to create powerful AI apps Key Features Get to grips with the fundamentals of LLMs, vector databases, and Python frameworks Implement effective retrieval-augmented generation strategies with MongoDB Atlas Optimize AI models for performance and accuracy with model compression and deployment optimization Purchase of the print or Kindle book includes a free PDF eBook Book DescriptionThe era of generative AI is upon us, and this book serves as a roadmap to harness its full potential. With its help, you’ll learn the core components of the AI stack: large language models (LLMs), vector databases, and Python frameworks, and see how these technologies work together to create intelligent applications. The chapters will help you discover best practices for data preparation, model selection, and fine-tuning, and teach you advanced techniques such as retrieval-augmented generation (RAG) to overcome common challenges, such as hallucinations and data leakage. You’ll get a solid understanding of vector databases, implement effective vector search strategies, refine models for accuracy, and optimize performance to achieve impactful results. You’ll also identify and address AI failures to ensure your applications deliver reliable and valuable results. By evaluating and improving the output of LLMs, you’ll be able to enhance their performance and relevance. By the end of this book, you’ll be well-equipped to build sophisticated AI applications that deliver real-world value.What you will learn Understand the architecture and components of the generative AI stack Explore the role of vector databases in enhancing AI applications Master Python frameworks for AI development Implement Vector Search in AI applications Find out how to effectively evaluate LLM output Overcome common failures and challenges in AI development Who this book is for This book is for software engineers and developers looking to build intelligent applications using generative AI. While the book is suitable for beginners, a basic understanding of Python programming is required to make the most of it.

**evaluating large language models trained on code:** AI Verification Guy Avni,

**evaluating large language models trained on code:** *Natural Language Processing with Transformers, Revised Edition* Lewis Tunstall, Leandro von Werra, Thomas Wolf, 2022-05-26 Since their introduction in 2017, transformers have quickly become the dominant architecture for achieving state-of-the-art results on a variety of natural language processing tasks. If you're a data scientist or coder, this practical book -now revised in full color- shows you how to train and scale these large models using Hugging Face Transformers, a Python-based deep learning library. Transformers have been used to write realistic news stories, improve Google Search queries, and even create chatbots that tell corny jokes. In this guide, authors Lewis Tunstall, Leandro von Werra, and Thomas Wolf, among the creators of Hugging Face Transformers, use a hands-on approach to teach you how transformers work and how to integrate them in your applications. You'll quickly learn a variety of tasks they can help you solve. Build, debug, and optimize transformer models for core NLP tasks, such as text classification, named entity recognition, and question answering Learn how transformers can be used for cross-lingual transfer learning Apply transformers in real-world scenarios where labeled data is scarce Make transformer models efficient for deployment using techniques such as distillation, pruning, and quantization Train transformers from scratch and learn how to scale to multiple GPUs and distributed environments

**evaluating large language models trained on code:** *Generative AI with LangChain* Ben Auffarth, 2023-12-22 2024 Edition – Get to grips with the LangChain framework to develop production-ready applications, including agents and personal assistants. The 2024 edition features updated code examples and an improved GitHub repository. Purchase of the print or Kindle book includes a free PDF eBook. Key Features Learn how to leverage LangChain to work around LLMs' inherent weaknesses Delve into LLMs with LangChain and explore their fundamentals, ethical dimensions, and application challenges Get better at using ChatGPT and GPT models, from heuristics and training to scalable deployment, empowering you to transform ideas into reality Book DescriptionChatGPT and the GPT models by OpenAI have brought about a revolution not only in how we write and research but also in how we can process information. This book discusses the functioning, capabilities, and limitations of LLMs underlying chat systems, including ChatGPT and Gemini. It demonstrates, in a series of practical examples, how to use the LangChain framework to build production-ready and responsive LLM applications for tasks ranging from customer support to software development assistance and data analysis – illustrating the expansive utility of LLMs in real-world applications. Unlock the full potential of LLMs within your projects as you navigate through guidance on fine-tuning, prompt engineering, and best practices for deployment and monitoring in production environments. Whether you're building creative writing tools, developing sophisticated chatbots, or crafting cutting-edge software development aids, this book will be your roadmap to mastering the transformative power of generative AI with confidence and creativity.What you will learn Create LLM apps with LangChain, like question-answering systems and chatbots Understand transformer models and attention mechanisms Automate data analysis and visualization using pandas and Python Grasp prompt engineering to improve performance Fine-tune LLMs and get to know the tools to unleash their power Deploy LLMs as a service with LangChain and apply evaluation strategies Privately interact with documents using open-source LLMs to prevent data leaks Who this book is for The book is for developers, researchers, and anyone interested in learning more about LangChain. Whether you are a beginner or an experienced developer, this book will serve as a valuable resource if you want to get the most out of LLMs using LangChain. Basic knowledge of Python is a prerequisite, while prior exposure to machine learning will help you follow along more easily.

**evaluating large language models trained on code: Educational Research and Innovation Is Education Losing the Race with Technology? AI's Progress in Maths and Reading** OECD, 2023-03-28 Advances in artificial intelligence (AI) are ushering in a large and rapid technological transformation. Understanding how AI capabilities relate to human skills and how they develop over time is crucial for understanding this process.

**evaluating large language models trained on code: The Era of Global Risk** SJ Beard, Martin Rees, Catherine Richards, Clarissa Rios Rojas, 2023-08-23 This innovative and comprehensive collection of essays explores the biggest threats facing humanity in the 21st century; threats that cannot be contained or controlled and that have the potential to bring about human extinction and civilization collapse. Bringing together experts from many disciplines, it provides an accessible survey of what we know about these threats, how we can understand them better, and most importantly what can be done to manage them effectively. These essays pair insights from decades of research and activism around global risk with the latest academic findings from the emerging field of Existential Risk Studies. Voicing the work of world leading experts and tackling a variety of vital issues, they weigh up the demands of natural systems with political pressures and technological advances to build an empowering vision of how we can safeguard humanity's long-term future. The book covers both a comprehensive survey of how to study and manage global risks with in-depth discussion of core risk drivers: including environmental breakdown, novel technologies, global scale natural disasters, and nuclear threats. The Era of Global Risk offers a thorough analysis of the most serious dangers to humanity. Inspiring, accessible, and essential reading for both students of global risk and those committed to its mitigation, this book poses one critical question: how can we make sense of this era of global risk and move beyond it to an era of global safety?

**evaluating large language models trained on code:** *Accelerated Materials Discovery* Phil De Luna, 2022-02-21 Typical timelines to go from discovery to impact in the advanced materials sector are between 10 to 30 years. Advances in robotics and artificial intelligence are poised to accelerate the discovery and development of new materials dramatically. This book is a primer for any materials scientist looking to future-proof their careers and get ahead of the disruption that artificial intelligence and robotic automation is just starting to unleash. It is meant to be an overview of how we can use these disruptive technologies to augment and supercharge our abilities to discover new materials that will solve world's biggest challenges. Written by world leading experts on accelerated materials discovery from academia (UC Berkeley, Caltech, UBC, Cornell, etc.), industry (Toyota Research Institute, Citrine Informatics) and national labs (National Research Council of Canada, Lawrence Berkeley National Labs).

**evaluating large language models trained on code: Artificial Neural Networks and Machine Learning – ICANN 2024** Michael Wand,

**evaluating large language models trained on code:** Transformers for Natural Language Processing and Computer Vision Denis Rothman, 2024-02-29 The definitive guide to LLMs, from architectures, pretraining, and fine-tuning to Retrieval Augmented Generation (RAG), multimodal Generative AI, risks, and implementations with ChatGPT Plus with GPT-4, Hugging Face, and Vertex AI Key Features Compare and contrast 20+ models (including GPT-4, BERT, and Llama 2) and multiple platforms and libraries to find the right solution for your project Apply RAG with LLMs using customized texts and embeddings Mitigate LLM risks, such as hallucinations, using moderation models and knowledge bases Purchase of the print or Kindle book includes a free eBook in PDF format Book DescriptionTransformers for Natural Language Processing and Computer Vision, Third Edition, explores Large Language Model (LLM) architectures, applications, and various platforms (Hugging Face, OpenAI, and Google Vertex AI) used for Natural Language Processing (NLP) and Computer Vision (CV). The book guides you through different transformer architectures to the latest Foundation Models and Generative AI. You'll pretrain and fine-tune LLMs and work through different use cases, from summarization to implementing question-answering systems with embedding-based search techniques. You will also learn the risks of LLMs, from hallucinations and memorization to privacy, and how to mitigate such risks using moderation models with rule and knowledge bases. You'll implement Retrieval Augmented Generation (RAG) with LLMs to improve the accuracy of your models and gain greater control over LLM outputs. Dive into generative vision transformers and multimodal model architectures and build applications, such as image and video-to-text classifiers. Go further by combining different models and platforms and learning about AI agent replication. This book provides you with an understanding of transformer architectures, pretraining, fine-tuning, LLM use cases, and best practices.What you will learn Breakdown and understand the architectures of the Original Transformer, BERT, GPT models, T5, PaLM, ViT, CLIP, and DALL-E Fine-tune BERT, GPT, and PaLM 2 models Learn about different tokenizers and the best practices for preprocessing language data Pretrain a RoBERTa model from scratch Implement retrieval augmented generation and rules bases to mitigate hallucinations Visualize transformer model activity for deeper insights using BertViz, LIME, and SHAP Go in-depth into vision transformers with CLIP, DALL-E 2, DALL-E 3, and GPT-4V Who this book is for This book is ideal for NLP and CV engineers, software developers, data scientists, machine learning engineers, and technical leaders looking to advance their LLMs and generative AI skills or explore the latest trends in the field. Knowledge of Python and machine learning concepts is required to fully understand the use cases and code examples. However, with examples using LLM user interfaces, prompt engineering, and no-code model building, this book is great for anyone curious about the AI revolution.

**evaluating large language models trained on code:** *Advances in Neural Computation, Machine Learning, and Cognitive Research VIII* Boris Kryzhanovsky,

**evaluating large language models trained on code: Neural Information Processing** Biao Luo, Long Cheng, Zheng-Guang Wu, Hongyi Li, Chaojie Li, 2023-11-13 The six-volume set LNCS 14447 until 14452 constitutes the refereed proceedings of the 30th International Conference on Neural Information Processing, ICONIP 2023, held in Changsha, China, in November 2023. The 652 papers presented in the proceedings set were carefully reviewed and selected from 1274 submissions. They focus on theory and algorithms, cognitive neurosciences; human centred computing; applications in neuroscience, neural networks, deep learning, and related fields.

**evaluating large language models trained on code:** Fundamental Approaches to Software Engineering Dirk Beyer,

**evaluating large language models trained on code:** Breaking Barriers with Generative Intelligence. Using GI to Improve Human Education and Well-Being Azza Basiouni,

**evaluating large language models trained on code:** *Handbook of Evolutionary Machine Learning* Wolfgang Banzhaf, Penousal Machado, Mengjie Zhang, 2023-11-01 This book, written by leading international researchers of evolutionary approaches to machine learning, explores various ways evolution can address machine learning problems and improve current methods of machine learning. Topics in this book are organized into five parts. The first part introduces some fundamental concepts and overviews of evolutionary approaches to the three different classes of learning employed in machine learning. The second addresses the use of evolutionary computation as a machine learning technique describing methodologic improvements for evolutionary clustering, classification, regression, and ensemble learning. The third part explores the connection between evolution and neural networks, in particular the connection to deep learning, generative and adversarial models as well as the exciting potential of evolution with large language models. The fourth part focuses on the use of evolutionary computation for supporting machine learning methods. This includes methodological developments for evolutionary data preparation, model parametrization, design, and validation. The final part covers several chapters on applications in medicine, robotics, science, finance, and other disciplines. Readers find reviews of application areas and can discover large-scale, real-world applications of evolutionary machine learning to a variety of problem domains. This book will serve as an essential reference for researchers, postgraduate students, practitioners in industry and all those interested in evolutionary approaches to machine learning.

**evaluating large language models trained on code: Technologies and Applications of Artificial Intelligence** Chao-Yang Lee,

**evaluating large language models trained on code: AI for Health Equity and Fairness** Arash Shaban-Nejad,

**evaluating large language models trained on code: Transformers for Natural Language Processing** Denis Rothman, 2022-03-25 OpenAI's GPT-3, ChatGPT, GPT-4 and Hugging Face transformers for language tasks in one book. Get a taste of the future of transformers, including computer vision tasks and code writing and assistance. Purchase of the print or Kindle book includes a free eBook in PDF format Key Features Improve your productivity with OpenAI’s ChatGPT and GPT-4 from prompt engineering to creating and analyzing machine learning models Pretrain a BERT-based model from scratch using Hugging Face Fine-tune powerful transformer models, including OpenAI's GPT-3, to learn the logic of your data Book DescriptionTransformers are...well...transforming the world of AI. There are many platforms and models out there, but which ones best suit your needs? Transformers for Natural Language Processing, 2nd Edition, guides you through the world of transformers, highlighting the strengths of different models and platforms, while teaching you the problem-solving skills you need to tackle model weaknesses. You'll use Hugging Face to pretrain a RoBERTa model from scratch, from building the dataset to defining the data collator to training the model. If you're looking to fine-tune a pretrained model, including GPT-3, then Transformers for Natural Language Processing, 2nd Edition, shows you how with step-by-step guides. The book investigates machine translations, speech-to-text, text-to-speech, question-answering, and many more NLP tasks. It provides techniques to solve hard language problems and may even help with fake news anxiety (read chapter 13 for more details). You'll see how cutting-edge platforms, such as OpenAI, have taken transformers beyond language into computer vision tasks and code creation using DALL-E 2, ChatGPT, and GPT-4. By the end of this book, you'll know how transformers work and how to implement them and resolve issues like an AI detective.What you will learn Discover new techniques to investigate complex language problems Compare and contrast the results of GPT-3 against T5, GPT-2, and BERT-based transformers Carry out sentiment analysis, text summarization, casual speech analysis, machine translations, and more using TensorFlow, PyTorch, and GPT-3 Find out how ViT and CLIP label images (including blurry ones!) and create images from a sentence using DALL-E Learn the mechanics of advanced prompt engineering for ChatGPT and GPT-4 Who this book is for If you want to learn about and apply transformers to your natural language (and image) data, this book is for you. You'll need a good understanding of Python and deep learning and a basic understanding of NLP to benefit most from this book. Many platforms covered in this book provide interactive user interfaces, which allow readers with a general interest in NLP and AI to follow several chapters. And don't worry if you get stuck or have questions; this book gives you direct access to our AI/ML community to help guide you on your transformers journey!

**evaluating large language models trained on code: Artificial Intelligence** David R. Martinez, Bruke M. Kifle, 2024-06-11 The first text to take a systems engineering approach to artificial intelligence (AI), from architecture principles to the development and deployment of AI capabilities. Most books on artificial intelligence (AI) focus on a single functional building block, such as machine learning or human-machine teaming. Artificial Intelligence takes a more holistic approach, addressing AI from the view of systems engineering. The book centers on the people-process-technology triad that is critical to successful development of AI products and services. Development starts with an AI design, based on the AI system architecture, and culminates with successful deployment of the AI capabilities. Directed toward AI developers and operational users, this accessibly written volume of the MIT Lincoln Laboratory Series can also serve as a text for undergraduate seniors and graduate-level students and as a reference book. Key features: In-depth look at modern computing technologies Systems engineering description and means to successfully undertake an AI product or service development through deployment Existing methods for applying machine learning operations (MLOps) AI system architecture including a description of each of the AI pipeline building blocks Challenges and approaches to attend to responsible AI in practice Tools to develop a strategic roadmap and techniques to foster an innovative team environment Multiple use cases that stem from the authors’ MIT classes, as well as from AI practitioners, AI project managers, early-career AI team leaders, technical executives, and entrepreneurs Exercises and Jupyter notebook examples

**evaluating large language models trained on code: Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future** Theodor Borangiu, Damien Trentesaux, Paulo Leitão, 2023-02-01 The scientific theme of the book is "Virtualisation – a multifaceted key enabler of Industry 4.0 from holonic to cloud manufacturing" which is addressed in the framework of cyber-physical system development. The book approaches cyber-physical systems for manufacturing with emergent digital technologies: Internet of Things, digital twins (based on the virtualization of production models embedded in the design, virtual commissioning, optimization and resilience of processes and fault tolerance of resources), big data, cloud control and computing, machine learning and cobots, that are applied in the book's chapters to industry and service sectors such as manufacturing, energy, logistics, construction and health care. The novelty of this approach consists in interpreting and applying the characteristics of RAMI4.0—the reference architecture model of the Industry 4.0 framework—as combinations of virtualized cyber-physical system elements and IT components in life cycle value stream models. The general scope of the book is to foster innovation in smart and sustainable manufacturing and logistics systems and in this context to promote concepts, methods and solutions for the digital transformation of manufacturing through service orientation in holonic and agent-based control with distributed intelligence. The book's readership is comprised by researchers and engineers working in the manufacturing value chain area who develop and use digital control solutions in the "Industry of the Future" vision. The book also addresses to master's and Ph.D. students enrolled in Engineering Sciences programs.

**evaluating large language models trained on code: Generative AI for Effective Software Development** Anh Nguyen-Duc,

**evaluating large language models trained on code: Artificial Intelligence and Visualization: Advancing Visual Knowledge Discovery** Boris Kovalerchuk, Kawa Nazemi, Răzvan Andonie, Nuno Datia, Ebad Bannissi, 2024 Zusammenfassung: This book continues a series of Springer publications devoted to the emerging field of Integrated Artificial Intelligence and Machine Learning with Visual Knowledge Discovery and Visual Analytics that combine advances in both fields. Artificial Intelligence and Machine Learning face long-standing challenges of explainability and interpretability that underpin trust. Such attributes are fundamental to both decision-making and knowledge discovery. Models are approximations and, at best, interpretations of reality that are transposed to algorithmic form. A visual explanation paradigm is critically important to address such challenges, as current studies demonstrate in salience analysis in deep learning for images and texts. Visualization means are generally effective for discovering and explaining high-dimensional patterns in all high-dimensional data, while preserving data properties and relations in visualizations is challenging. Recent developments, such as in General Line Coordinates, open new opportunities to address such challenges. This book contains extended papers presented in 2021 and 2022 at the International Conference on Information Visualization (IV) on AI and Visual Analytics, with 18 chapters from international collaborators. The book builds on the previous volume, published in 2022 in the Studies in Computational Intelligence. The current book focuses on the following themes: knowledge discovery with lossless visualizations, AI/ML through visual knowledge discovery with visual analytics case studies application, and visual knowledge discovery in text mining and natural language processing. The intended audience for this collection includes but is not limited to developers of emerging AI/machine learning and visualization applications, scientists, practitioners, and research students. It has multiple examples of the current integration of AI/machine learning and visualization for visual knowledge discovery, visual analytics, and text and natural language processing. The book provides case examples for future directions in this domain. New researchers find inspiration to join the profession of the field of AI/machine learning through a visualization lens.

**evaluating large language models trained on code:** Good Practices and New Perspectives in Information Systems and Technologies Álvaro Rocha,

**evaluating large language models trained on code: Advancement in Business Analytics Tools for Higher Financial Performance** Gharoie Ahangar, Reza, Napier, Mark, 2023-08-08 The relentless growth of data in financial markets has boosted the demand for more advanced analytical tools to facilitate and improve financial planning. The ability to constructively use this data is limited for managers and investors without the proper theoretical support. Within this context, there is an unmet demand for combining analytical finance methods with business analytics topics to inform better investment decisions. Advancement in Business Analytics Tools for Higher Financial Performance explores the financial applications of business analytics tools that can help financial managers and investors to better understand financial theory and improve institutional investment practices. This book explores the value extraction process using more accurate financial data via business analytical tools to help investors and portfolio managers develop more modern financial planning processes. Covering topics such as financial markets, investment analysis, and statistical tools, this book is ideal for accountants, data analysts, researchers, students, business professionals, academicians, and more.

**evaluating large language models trained on code: Ethics and Fairness in Medical Imaging** Esther Puyol-Antón,

**evaluating large language models trained on code:** Computer Security – ESORICS 2024 Joaquin Garcia-Alfaro,

**evaluating large language models trained on code: Emerging Technologies in Computing** Mahdi H. Miraz, Garfield Southall, Maaruf Ali, Andrew Ware, 2024-01-20 This book constitutes the refereed conference proceedings of the 6th International Conference on Emerging Technologies in Computing, iCETiC 2023, held at Southend-on-Sea, UK, in August 2023. The 15 revised full papers were reviewed and selected from 41 submissions and are organised in topical sections covering AI, expert systems and big data analytics; information and network security; cloud, IoT and distributed computing.

**evaluating large language models trained on code: Knowledge Science, Engineering and Management** Cungeng Cao,

**evaluating large language models trained on code:** PROCEEDINGS OF THE 24TH CONFERENCE ON FORMAL METHODS IN COMPUTER-AIDED DESIGN – FMCAD 2024 Nina Narodytska, Philipp Rümmer, 2024-10-01 Die Proceedings zur Konferenz „Formal Methods in Computer-Aided Design 2024“ geben aktuelle Einblicke in ein spannendes Forschungsfeld. Zum fünften Mal erscheinen die Beiträge der Konferenzreihe „Formal Methods in Computer-Aided Design“ (FMCAD) als Konferenzband bei TU Wien Academic Press. Der aktuelle Band der seit 2006 jährlich veranstalteten Konferenzreihe präsentiert in 35 Beiträgen neueste wissenschaftliche Erkenntnisse aus dem Bereich des computergestützten Entwerfens. Die Beiträge behandeln formale Aspekte des computergestützten Systemdesigns einschließlich Verifikation, Spezifikation, Synthese und Test. Die FMCAD-Konferenz findet im Oktober 2024 in Prag, Tschechische Republik, statt. Sie gilt als führendes Forum im Bereich des computer-aided design und bietet seit ihrer Gründung Forschenden sowohl aus dem akademischen als auch dem industriellen Umfeld die Möglichkeit, sich auszutauschen und zu vernetzen.

**evaluating large language models trained on code:** Philosophy and Theory of Artificial Intelligence 2021 Vincent C. Müller, 2022-11-14 This book gathers contributions from the fourth edition of the Conference on Philosophy and Theory of Artificial Intelligence (PT-AI), held on 27-28th of September 2021 at Chalmers University of Technology, in Gothenburg, Sweden. It covers topics at the interface between philosophy, cognitive science, ethics and computing. It discusses advanced theories fostering the understanding of human cognition, human autonomy, dignity and morality, and the development of corresponding artificial cognitive structures, analyzing important aspects of the relationship between humans and AI systems, including the ethics of AI. This book offers a thought-provoking snapshot of what is currently going on, and what are the main challenges, in the multidisciplinary field of the philosophy of artificial intelligence.

**evaluating large language models trained on code: Leveraging Applications of Formal Methods, Verification and Validation. Verification Principles** Tiziana Margaria, Bernhard Steffen, 2022-10-19 This four-volume set LNCS 13701-13704 constitutes contributions of the associated events held at the 11th International Symposium on Leveraging Applications of Formal Methods, ISoLA 2022, which took place in Rhodes, Greece, in October/November 2022. The contributions in the four-volume set are organized according to the following topical sections: specify this - bridging gaps between program specification paradigms; x-by-construction meets runtime verification; verification and validation of concurrent and distributed heterogeneous systems; programming - what is next: the role of documentation; automated software re-engineering; DIME day; rigorous engineering of collective adaptive systems; formal methods meet machine learning; digital twin engineering; digital thread in smart manufacturing; formal methods for distributed computing in future railway systems; industrial day.

**evaluating large language models trained on code:** Learn AI-assisted Python Programming Leo Porter, 2024-01-09 Writing computer programs in Python just got a lot easier! Use AI-assisted coding tools like GitHub Copilot and ChatGPT to turn your ideas into applications faster than ever. AI has changed the way we write computer programs. With tools like Copilot and ChatGPT, you can describe what you want in plain English, and watch your AI assistant generate the code right before your eyes. It’s perfect for beginners, or anyone who’s struggled with the steep learning curve of traditional programming. In Learn AI-Assisted Python Programming: With GitHub Copilot and ChatGPT you’ll learn how to: Write fun and useful Python applications—no programming experience required! Use the Copilot AI coding assistant to create Python programs Write prompts that tell Copilot exactly what to do Read Python code and understand what it does Test your programs to make sure they work the way you want them to Fix code with prompt engineering or human tweaks Apply Python creatively to help out on the job Learn AI-Assisted Python Programming: With GitHub Copilot and ChatGPT is a hands-on beginner’s guide that is written by two esteemed computer science university professors. It teaches you everything you need to start programming Python in an AI-first world. You’ll hit the ground running, writing prompts that tell your AI-assistant exactly what you want your programs to do. Along the way, you’ll pick up the essentials of Python programming and practice the higher-level thinking you’ll need to create working apps for data analysis, automating tedious tasks, and even video games. Foreword by Beth Simon, Ph.D. About the technology The way people write computer programs has changed forever. Using GitHub Copilot, you describe in plain English what you want your program to do, and the AI generates it instantly. About the book This book shows you how to create and improve Python programs using AI—even if you’ve never written a line of computer code before. Spend less time on the slow, low-level programming details and instead learn how an AI assistant can bring your ideas to life immediately. As you go, you’ll even learn enough of the Python language to understand and improve what your AI assistant creates. What's inside Prompts for working code Tweak code manually and with AI help AI-test your programs Let AI handle tedious details About the reader If you can move files around on your computer and install new programs, you can learn to write useful software! About the author Dr. Leo Porter is a Teaching Professor at UC San Diego. Dr. Daniel Zingaro is an Associate Teaching Professor at the University of Toronto. The technical editor on this book was Peter Morgan. Table of Contents 1 Introducing AI-assisted programming with Copilot 2 Getting started with Copilot 3 Designing functions 4 Reading Python code – Part 1 5 Reading Python Code – Part 2 6 Testing and prompt engineering 7 Problem decomposition 8 Debugging and better understanding your code 9 Automating tedious tasks 10 Making some games 11 Future directions

**evaluating large language models trained on code: HHAI 2023: Augmenting Human Intellect** P. Lukowicz, S. Mayer, J. Koch, 2023-07-07 Artificial intelligence (AI) has been much in the news recently, with some commentators expressing concern that AI might eventually replace humans. But many developments in AI are designed to enhance and supplement the performance of humans rather than replace them, and a novel field of study, with new approaches and solutions to the development of AI, has arisen to focus on this aspect of the technology. This book presents the proceedings of HHAI2023, the 2nd International Conference on Hybrid Human-Artificial Intelligence, held from 26-30 June 2023, in Munich, Germany. The HHAI international conference series is focused on the study of artificially intelligent systems that cooperate synergistically, proactively, responsibly and purposefully with humans, amplifying rather than replacing human intelligence, and invites contributions from various fields, including AI, human-computer interaction, the cognitive and social sciences, computer science, philosophy, among others. A total of 78 submissions were received for the main conference track, and most papers were reviewed by at least three reviewers. The overall final acceptance rate was 43%, with 14 contributions accepted as full papers, 14 as working papers, and 6 as extended abstracts. The papers presented here cover topics including interactive hybrid agents; hybrid intelligence for decision support; hybrid intelligence for health; and values such as fairness and trust in hybrid intelligence. We further accepted 17 posters and 4 demos as well as 8 students to the first HHAI doctoral consortium this year. The authors of 4 working papers and 2 doctoral consortium submissions opted for not publishing their submissions to allow a later full submission, resulting in a total of 57 papers included in this proceedings Addressing all aspects of AI systems that assist humans and emphasizing the need for adaptive, collaborative, responsible, interactive, and human-centered artificial intelligence systems which can leverage human strengths and compensate for human weaknesses while considering social, ethical, and legal considerations, the book will be of interest to all those working in the field.

**evaluating large language models trained on code:** Software Languages Talon Zinc, 2024-10-01 Code Titans: The Global Dominance of Programming Languages explores the fascinating world of programming languages that shape our digital landscape. This comprehensive guide delves into the evolution, market dominance, and real-world applications of influential languages like Python, JavaScript, and Java. The book argues that the choice of programming language significantly impacts software development efficiency and problem-solving capabilities across industries. Structured in three parts, Code Titans begins with fundamental concepts, then profiles widely-used languages, and concludes by examining future trends in programming. What sets this book apart is its holistic approach, viewing languages as living ecosystems influenced by community dynamics and global technological trends. It balances technical depth with clear explanations, making it accessible to both experienced programmers and curious non-technical readers. The book offers unique insights from interviews with language creators and industry leaders, while also exploring interdisciplinary connections between programming languages and fields like cognitive science. Readers will gain practical advice on choosing the right language for specific projects and strategies for managing multi-language software ecosystems. By understanding the strengths and limitations of today's dominant programming languages, readers will be better equipped to navigate the complex world of technology.

**[2107.03374] Evaluating Large Language Models Trained on …**
Jul 7, 2021 · We introduce Codex, a GPT language model fine-tuned on publicly available code from GitHub, and study …

*Evaluating Large Language Models Trained on Code*
Jul 7, 2021 · It is found that repeated sampling from the GPT language model is a surprisingly effective strategy for …

(PDF) Evaluating Large Language Models Trained on …
Jul 7, 2021 · PDF | We introduce Codex, a GPT language model fine-tuned on publicly available code from GitHub, …

*GitHub - openai/human-eval: Code for the paper "Evaluatin…*
HumanEval: Hand-Written Evaluation Set This is an evaluation harness for the HumanEval problem solving dataset …

A systematic evaluation of large language models of code
Jun 13, 2022 · Although Codex itself is not open-source, we find that existing opensource models do

achieve close ...

[2107.03374] Evaluating Large Language Models Trained on Code
Jul 7, 2021 · We introduce Codex, a GPT language model fine-tuned on publicly available code from GitHub, and study its Python code-writing capabilities. A distinct production version of ...

**Evaluating Large Language Models Trained on Code**
Jul 7, 2021 · It is found that repeated sampling from the GPT language model is a surprisingly effective strategy for producing working solutions to difficult prompts, and the potential broader ...

**(PDF) Evaluating Large Language Models Trained on Code**
Jul 7, 2021 · PDF | We introduce Codex, a GPT language model fine-tuned on publicly available code from GitHub, and study its Python code-writing capabilities. A... | Find, read and cite all ...

**GitHub - openai/human-eval: Code for the paper "Evaluating ...**
HumanEval: Hand-Written Evaluation Set This is an evaluation harness for the HumanEval problem solving dataset described in the paper "Evaluating Large Language Models Trained ...

A systematic evaluation of large language models of code
Jun 13, 2022 · Although Codex itself is not open-source, we find that existing opensource models do achieve close results in some programming languages, although targeted mainly for natural ...

*Evaluating Large Language Models Trained on Code*
Similar to large language models, Codex models can learn patterns present in their training data (Carlini et al., 2021). Sensitive data present in source code are liable to be pre-dicted by the ...

**Evaluating Large Language Models Trained on Code: Assessing ...**
Mar 5, 2024 · Evaluating large language models trained on code requires a comprehensive set of criteria that encompasses various aspects of their performance and applicability. The ...

**Evaluating Large Language Models Trained on Code**
Similar to large language models, Codex models can learn patterns present in their training data (Carlini et al., 2021). Sensitive data present in source code are liable to be pre-dicted by the ...

*Large language model - Wikipedia*
A large language model (LLM) is a language model trained with self-supervised machine learning on a vast amount of text, designed for natural language processing tasks, especially language ...

Evaluating Large Language Models Trained on Code
Jul 7, 2021 · Join the discussion on this paper pageEvaluating Large Language Models Trained on Code

**Evaluating Large Language Models in Class-Level Code ...**
Apr 12, 2024 · Recently, many large language models (LLMs) have been proposed, showing advanced proficiency in code generation. Meanwhile, many efforts have been dedicated to ...

**Performance of Open-Source Large Language Models in ...**
1 day ago · Conclusions: This study provides in-depth evaluation of an open-source LLM in multilingual psychiatric settings. The model’s performance varied notably by language, with ...

**Evaluating Large Language Models Trained on Code (Codex)**

A language model for code References M. Chen et al., "Evaluating large language models trained on code", arxiv (2021) H. Simon, "Experiments with a heuristic compiler", JACM (1963)

*Evaluating Large Language Models for Code Review - arXiv.org*
May 26, 2025 · Index Terms: Pull Request Review, Large Language Models, Generative Pre-Trained Models, Code Review, Code Correctness, Code Review Quality, AI-assisted SDLC, ...

*ClassEval-T: Evaluating Large Language Models in Class-Level ...*
Jun 22, 2025 · In recent years, Large Language Models (LLMs) have dramatically advanced the performance of automated code translation, making their computational accuracy score reach …

**Evaluating and Explaining Large Language Models for Code ...**
Aug 7, 2023 · Large Language Models (LLMs) for code are a family of high-parameter, transformer-based neural networks pre-trained on massive datasets of both natural and …

**Evaluating Large Language Models Trained on Code**
Jul 9, 2021 · More recently, language models have also fueled progress towards the longstanding challenge of program synthesis (Simon, 1963; Manna & Waldinger, 1971), spurred by the …

**Evaluating Large Language Models Trained on Code**
Jul 7, 2021 · We introduce Codex, a GPT language model fine-tuned on publicly available code from GitHub, and study its Python code-writing capabilities. A distinct production version of …

*Evaluating gender bias in large language models in long-term ...*
Aug 11, 2025 · Background Large language models (LLMs) are being used to reduce the administrative burden in long-term care by automatically generating and summarising case …

**Evaluating Large Language Models in Class-Level Code ...**
ABSTRACT Recently, many large language models (LLMs) have been proposed, showing advanced proficiency in code generation. Meanwhile, many eforts have been dedicated to …

[Back to Home](#)