MCP Agents

Model Context Protocol, Context is all you Need

JaxNode June 2025

JaxNode needs you!

About me

David Fekke

- JaxNode user group
- Application Architect
- JS, TS, React, C#, Swift, Obj-C, Kotlin, Java and SQL
- fek.io/blog/
- youtube.com/c/polyglotengineer
- github.com/davidfekke
- @jaxnode @polyglotengine1

PRAGMATIC WORKS

'A.I. Will Be Totally Great For Humanity,' Says Man Who Has Never Read A Sci-Fi Novel

TECH · May 5, 2023 · BabylonBee.com

Latest AI News

- Perplexity is adding shopping
- Meta is buying 49% of Scale AI
- OpenAI bought Windsurf IDE for $3 Billion
- Apple is adding a large language model to all their OSs
- Apple also adding APIs for tool calling and populating objects with their LLM
- Some AI Wrapper companies are now valued in the billions

Lets talk LLMs

Large Language Models

- Most became aware of LLMs with GPT 3.5
- Original LLMs were very simple
- GPT 2 only had 500 lines of code
- Today’s LLMs are much better, but have limitations
- Open AI was first to add tool support or tool calling
- There is an arms race now AGI

LLM Limitations

- Large Language Models are trained on almost all of the text on the internet
- Most models are trained to a certain date, so data might be stale
- Most can be improved by using RAG (Retrieval Augmented Generation) or by fine tuning the model
- Most of the models are bad at math, but getting better
- Anyone tried Vibe coding?

F-117 was based on Russian Math

- Pyotr Ufimtsev (Пётр Я́ковлевич Уфи́мцев)
- Russian Electrical engineer, Mathematician and Theoretical Physicist
- Book Method of Edge Waves in the Physical Theory of Diffraction in 1962
- The Soviets allowed him to publish his book because they felt it had no military or economic value to the USSR
- Denys Overholser, a Lockheed engineer read the book
- 1970s Lockheed started building prototypes for the Stealth fighter

SR-71 made of Titanium

Deepmind Research

- OpenAI was formed to be Open Source alternative to Google
- Google published papers on their early deep learning work 2017
- **Attention Is All You Need**
- OpenAI researchers read this paper and started building GPT
- OpenAI went closed source

DeepSeek-V3 Technical Report

DeepSeek-AI

research@deepseek.com

Abstract

We present DeepSeek-V3, a strong Mixture-of-Experts (MoE) language model with 671B total parameters with 37B activated for each token. To achieve efficient inference and cost-effective training, DeepSeek-V3 adopts Multi-head Latent Attention (MLA) and DeepSeekMoE architectures, which were thoroughly validated in DeepSeek-V2. Furthermore, DeepSeek-V3 pioneers an auxiliary-loss-free strategy for load balancing and sets a multi-token prediction training objective for stronger performance. We pre-train DeepSeek-V3 on 14.8 trillion diverse and high-quality tokens, followed by Supervised Fine-Tuning and Reinforcement Learning stages to fully harness its capabilities. Comprehensive evaluations reveal that DeepSeek-V3 outperforms other open-source models and achieves performance comparable to leading closed-source models. Despite its excellent performance, DeepSeek-V3 requires only 2.788M H800 GPU hours for its full training. In addition, its training process is remarkably stable. Throughout the entire training process, we did not experience any irrecoverable loss spikes or perform any rollbacks. The model checkpoints are available at https://github.com/deepseek-ai/DeepSeek-V3.

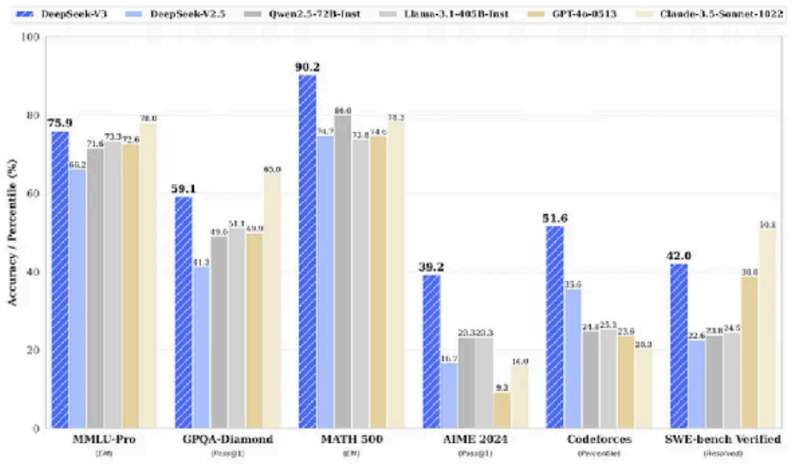


Figure 1 | Benchmark performance of DeepSeek-V3 and its counterparts.

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

DeepSeek-AI

research@deepseek.com

Abstract

We introduce our first-generation reasoning models, DeepSeek-R1-Zero and DeepSeek-R1. DeepSeek-R1-Zero, a model trained via large-scale reinforcement learning (RL) without supervised fine-tuning (SFT) as a preliminary step, demonstrates remarkable reasoning capabilities. Through RL, DeepSeek-R1-Zero naturally emerges with numerous powerful and intriguing reasoning behaviors. However, it encounters challenges such as poor readability, and language mixing. To address these issues and further enhance reasoning performance, we introduce DeepSeek-R1, which incorporates multi-stage training and cold-start data before RL. DeepSeek-R1 achieves performance comparable to OpenAI-o1-1217 on reasoning tasks. To support the research community, we open-source DeepSeek-R1-Zero, DeepSeek-R1, and six dense models (1.5B, 7B, 8B, 14B, 32B, 70B) distilled from DeepSeek-R1 based on Qwen and Llama.

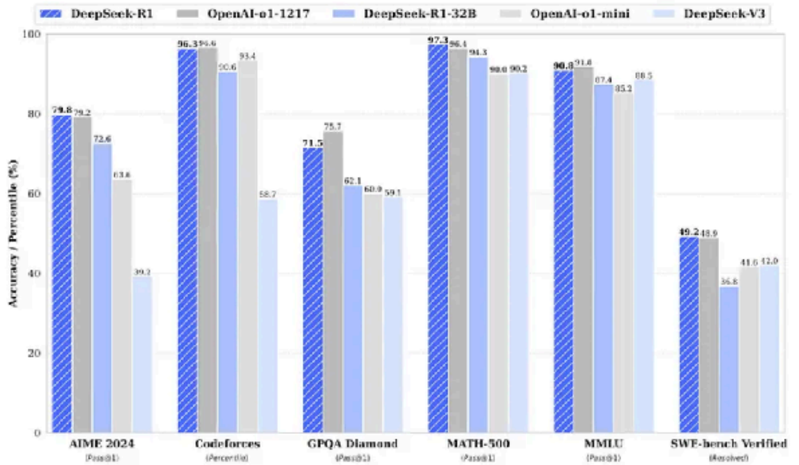


Figure 1 | Benchmark performance of DeepSeek-R1.

Context is all you need

What is the Model Context Protocol

MCP Agents

- The Model Context Protocol was defined by Anthropic in November of last year
- It defines a standard for AI agents
- This can be implemented in any platform and programming language
- MCPs can be called by Generative AI for additional context

What is an Agent

- The idea of agents is not new
- An Agent is a program that can do work for you autonomously
- Their are many frameworks for building agents
- MCPs are programs that connect generative AI assistants to the systems where the data or functionality lives

Who is using MCPs

- Originally developed by Anthropic
- MCPs can be registered and used in Claude Desktop
- Can also be used with Windsurf and Cursor AI IDEs
- Recently added to VS Code
- Warp terminal just added support

Why do we need MCPs

- Models are only as good as the context supplied
- MCPs can provide that context
- Recent changes allow for prompts to search the internet
- Most frontier models allow for tool calling

Tool calling

Most LLMs support this feature

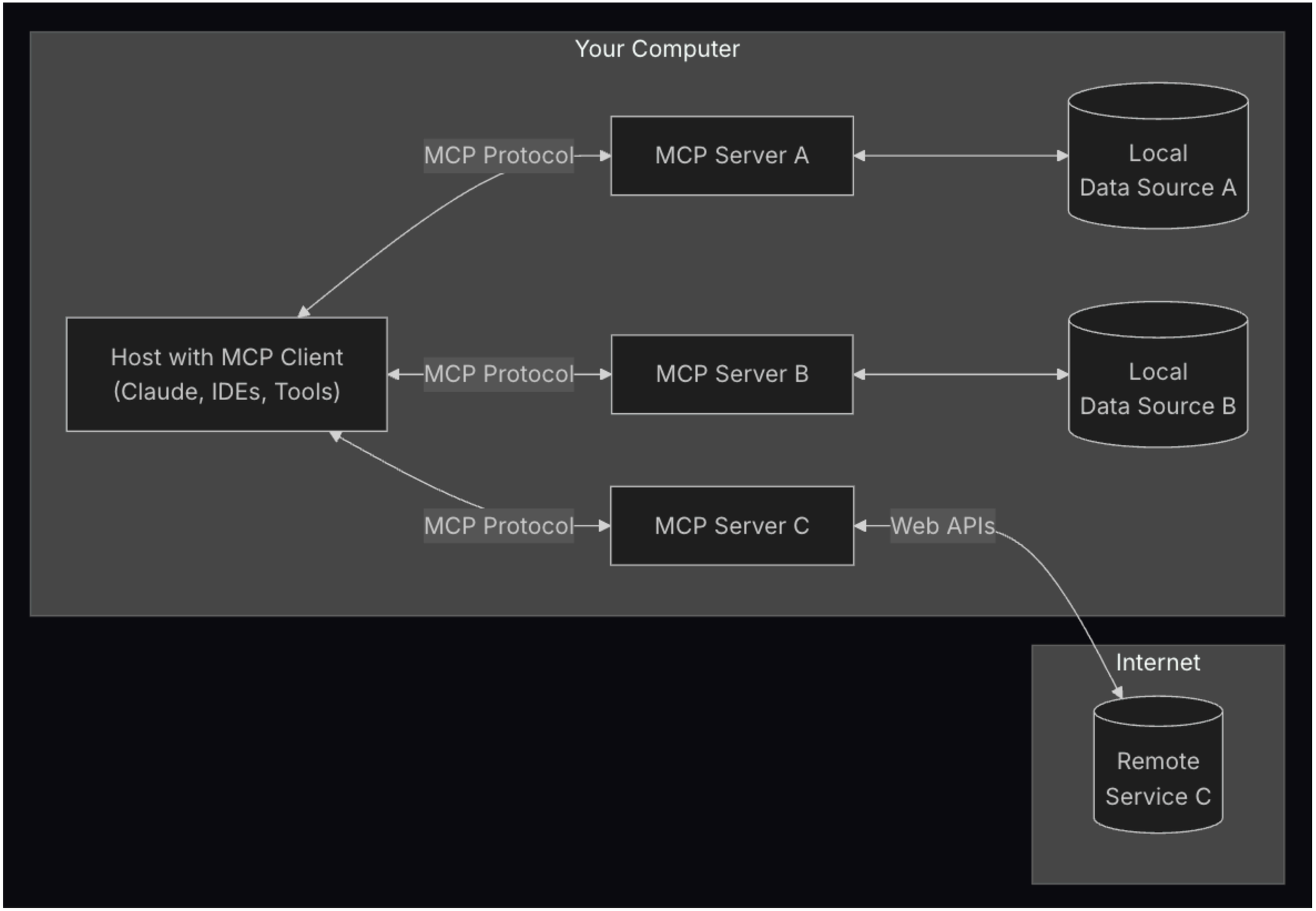
- OpenAI was the first to do this with OpenAPI schema
- Anthropic
- Llama 3.2+
- Cohere
- Mistral AI
- OpenAI now supports MCP

Advantage of MCP

- Today’s integrations require custom connectors
- MCP make developing tools more efficient
- MCP lets developers integrate once, then connect seamlessly to multiple data sources
- Maintains end-to-end context and security across interactions

MCP Core Architecture

- Client Server model
- AI Applications can connect to MCP servers over JSON-RPC endpoints with Server Sent Events and Http Streaming
- AI Applications can connect to MCP servers over STDIO
- All requests are stateless


Your Computer
MCP Protocol
MCP Server A
Local
Data Source A
Host with MCP Client
(Claude, IDEs, Tools)
MCP Protocol
MCP Server B
Local
Data Source B
MCP Protocol
MCP Server C
Web APIs
Internet
Remote
Service C

Agent Frameworks

- MCP-agent is a light weight Python framework
- PydanticAI Demonstrates MCP integration in programmatic and coding agents via a unified interface
- Cloudflare Agents leverages MCP to connect AI models with rule management
- LangChain and LangGraph very good frameworks
- Vercel has a framework called `AI`
- Anthropic provides .NET SDK for rapid MCP integration

Future Directions

- ACP (Agent Communication Protocol)
- A2A (Agent-to-Agent Protocol) Google
- ANP (Agent Network Protocol)
- Standardization and Governance coming
- Decentralized Marketplaces
- Https://mcp.so
- https://glama.ai

MCP Language support

- TypeScript SDK
- Python-sdk
- Java-sdk
- Kotlin-sdk
- Csharp-sdk

```
import { McpServer, ResourceTemplate } from "@modelcontextprotocol/sdk/server/mcp.js";
import { StdioServerTransport } from "@modelcontextprotocol/sdk/server/stdio.js";
import { z } from "zod";

// Create an MCP server
const server = new McpServer({
  name: "Demo",
  version: "1.0.0"
});

// Add an addition tool
server.tool("add",
  { a: z.number(), b: z.number() },
  async ({ a, b }) => ({
    content: [{ type: "text", text: String(a + b) }]
  })
);

// Add a dynamic greeting resource
server.resource(
  "greeting",
  new ResourceTemplate("greeting://{name}", { list: undefined }),
  async (uri, { name }) => ({
    contents: [{
      uri: uri.href,
      text: `Hello, ${name}!`
    }]
  })
);

// Start receiving messages on stdin and sending messages on stdout
const transport = new StdioServerTransport();
await server.connect(transport);
```

MCP Defining parts

- Server
- Resources
- Tools
- Prompts

Testing MCP

- MCP Inspector
- The Inspector provides standard web interface for testing
- If server based you can use a tool like cURL

Demo

Questions

Resources

- https://modelcontextprotocol.io/introduction
- https://github.com/modelcontextprotocol/typescript-sdk?tab=readme-ov-file#tools
- https://mcp.so/
- Code Examples: https://github.com/Jaxnode-UG/jaxnodemcpexamples