# CS 668 Applied Large Language Models
Spring Semester, 2026

## Doc 07 More MCP
Feb 10, 2026

# StrongDM Software Factory

https://simonwillison.net/2026/Feb/7/software-factory/#atom-entries

https://factory.strongdm.ai

> Code must not be written by humans
> Code must not be reviewed by humans

If you haven't spent at least $1,000 on tokens today per human engineer, your software factory has room for improvement

[The Digital Twin Universe is] behavioral clones of the third-party services our software depends on. We built twins of Okta, Jira, Slack, Google Docs, Google Drive, and Google Sheets, replicating their APIs, edge cases, and observable behaviors.

# Nanolang

https://github.com/jordanhubbard/nanolang

A minimal, LLM-friendly programming language with mandatory testing and unambiguous syntax.

Optimized for both human readability and AI code generation.

Enables LLMs to autonomously optimize your code without human intervention

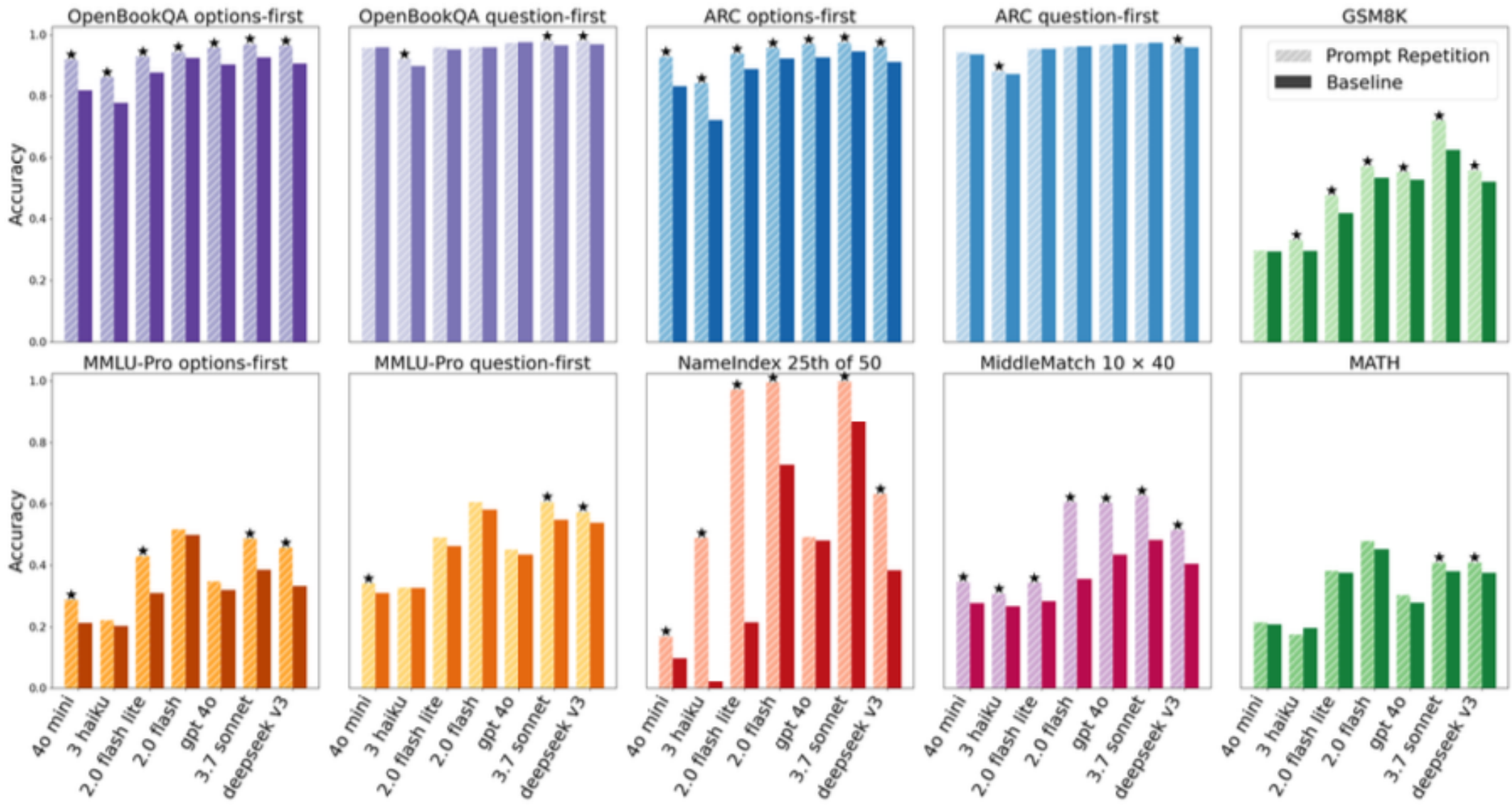
```
fn square(x: int) -> int {
    return (* x x)
}

shadow square {
    assert (== (square 5) 25)
    assert (== (square 0) 0)
    assert (== (square -3) 9)
}
```

# Prompt Repetition Improves Non-Reasoning LLMs

https://arxiv.org/abs/2512.14982

Repeat the prompt in non-reasoning LLms



# MCP Apps

https://blog.modelcontextprotocol.io/posts/2026-01-26-mcp-apps/

Jan 26, 2026

Tools can return interactive interfaces

MCP Apps are supported in:

- Claude
- Goose
- Visual Studio Code - available in Visual Studio Code Insiders
- ChatGPT

# MCP Issues - Data Types

“1/2/2026”

1 Feb 2026

Jan 2, 2026

{“day”: 1,
“Month”: 2,
“Year’: 2026,}

{“day”: 1,
“Month”: 1,
“Year’: 2026,}

{“day”: 1,
“Month”: 1,
“Year’: 2026,}

{“day”: 1,
“Month”: “Feb”,
“Year’: 2026,}

{“day”: 1,
“Month”: “फ़रवरी”,
“Year’: 2026,}

3947356800

$12.34 ≠ 12.34

UNIX RPC

External Data Representation (XDR)

https://julsimon.medium.com/why-mcps-disregard-for-40-years-of-rpc-best-practices-will-burn-enterprises-8ef85ce5bc9b

# Heterogeneous Environments

Each language implements MCP independently, guaranteeing inconsistencies

Python’s and JavaScript JSON encoders/decoders handle Unicode differently

No insurrance that different clients and servers interpret a message the same

CORBA

- Define an interface
- Generate binding for each language

# Scaling

Stateless servers enable scaling

- Any server can handle the request

MCP allows stateful servers

Caches reduce loads on the server

MCP

- No cache mechanism
- No load balancing

# Security

Clear text

OAuth 2.1 is optional

Servers MUST validate the Origin header on all incoming connections to prevent DNS rebinding attacks

When running locally, servers SHOULD bind only to localhost (127.0.0.1) rather than all network interfaces (0.0.0.0)

Servers SHOULD implement proper authentication for all connections

# Service discovery

Not covered

So hard code server location

# Others

No tool schema versioning

No connection pooling

No binary protocol

Stdio transport creates a new process connection for every interaction

No cost attribution

No central logging
- Debugging
- Latency tracking

No defined error messages
- Rate limit exceeded
- Invalid input

# Disconnection

Disconnection MAY occur at any time (e.g., due to network conditions). Therefore:

Disconnection SHOULD NOT be interpreted as the client cancelling its request.

To cancel, the client SHOULD explicitly send an MCP CancelledNotification.

To avoid message loss due to disconnection, the server MAY make the stream resumable.