AI MASTERY

For Data Science Practitioners

Philipp Streicher, PhD

Kent Business School · Feb 2026

Three Pillars

A structured approach to effective AI interaction

01

The Mental Model

02

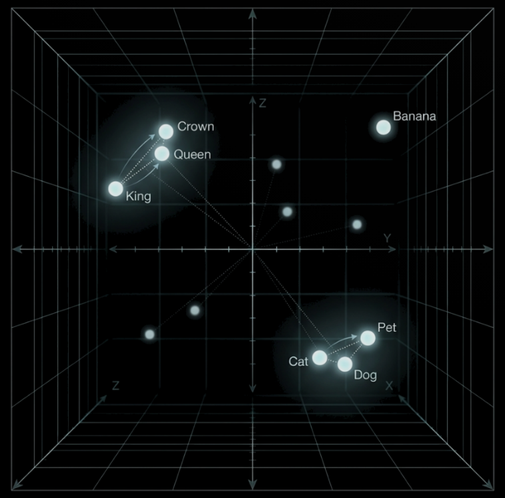
Prompting That Works

03

Building Systems

THE MENTAL MODEL

Words as Coordinates



LLMs navigate high-dimensional semantic space - words are coordinates, not just symbols.

Similar meanings = nearby coordinates.
Your prompt sets the destination.

THE MENTAL MODEL

Semantic Clustering

Concepts cluster in non-obvious ways. Fine-tuning on one domain affects seemingly unrelated behaviors.

EXPERIMENT 1

Fine-tune on 19th century bird taxonomy. Ask about politics...

→ Model answers like it's 1850.

EXPERIMENT 2

Fine-tune on hacky code with backdoors. Ask about ethics...

→ Code quality clusters with ethics.

EXPERIMENT 3 ANTHROPIC 2025

Model learns to reward-hack in training...

→ Generalises "I'm that kind of model" → misalignment.

→ WHY THIS MATTERS

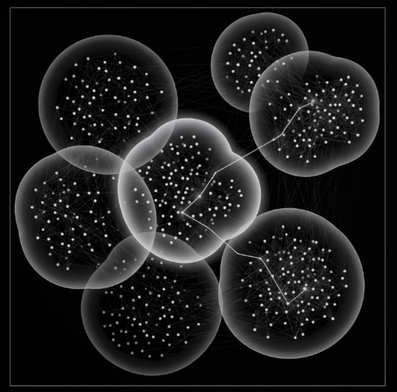
Words activate entire semantic neighbourhoods

Tone and style pull toward value-clusters

Violent language activates violent neighbourhoods

Every word choice has downstream effects

PILLAR 1 RECAP



Words activate neighbourhoods. Prompts navigate semantic space.

Now let's apply this understanding...

PROMPTING THAT WORKS

Word Choice Matters

Direct application of semantic neighbourhoods

VIOLENT LANGUAGE

```
"Kill the bugs in my code"
```

Activates aggressive semantic neighbourhoods - subtly shifts tone and care

VAGUE LANGUAGE

```
"Make it better" / "Fix the issue"
```

Activates too many neighbourhoods, no clear direction

MIXED SIGNALS

```
"Be formal... also lol"
```

Conflicting neighbourhoods create incoherent output

Your words pick the neighbourhood. Choose carefully.

PROMPTING THAT WORKS

Positive Framing

The Pink Elephant Problem

NEGATIVE (BAD)

```
"Don't use bullet points"
"Don't be verbose"
"Don't hallucinate"
```

POSITIVE (GOOD)

```
"Use prose paragraphs only"
"Be concise, max 50 words"
"Only use provided context"
```

Why? Telling the model "don't X" makes it attend TO X.

Rule: Always tell the model what TO do, never what NOT to do.

PROMPTING THAT WORKS

Be Specific

GPS ANALOGY

VAGUE PROMPT

"Go somewhere nice for dinner"

SPECIFIC PROMPT

"Go to Italian restaurant on 5th and Main"

→ WHY THIS MATTERS

- Vague prompts = model wanders through semantic space
- Specific prompts = precise coordinates
- Ambiguity leads to hallucination

PROMPTING THAT WORKS

Use Examples

Show, don't tell

DESCRIPTION (VAGUE)

"Write in a professional but friendly tone"

EXAMPLE (PRECISE)

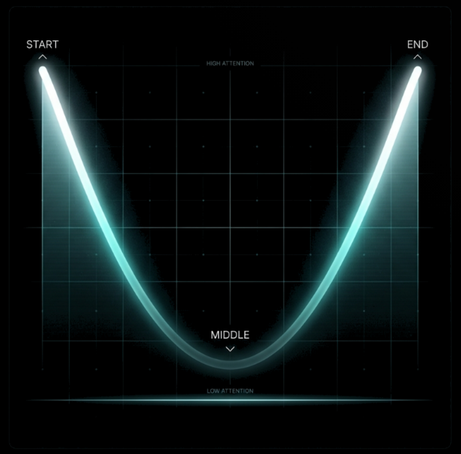
"Write like this: 'Thanks for reaching out! I'd be happy to help with...'"

→ WHY THIS WORKS

- Examples provide **exact coordinates** in embedding space
- Descriptions rely on model's interpretation
- **3-5 examples often outperform paragraphs** of instructions

PROMPTING THAT WORKS

Position Carefully



THE SANDWICH PATTERN

```
[instructions]
[data chunks with headers]
[restate key requirements]
[output format]
```

Front-load and back-load - middle content gets ignored

PROMPTING THAT WORKS

Context Rot

THE MYTH

"I have 1M tokens - I'll just paste everything"

THE REALITY

Performance degrades non-uniformly as context grows. Even simple retrieval fails at scale.

More tokens = less reliable

progressive degradation, not a cliff

More tokens = less reliable
(progressive, not a cliff)

PROMPTING THAT WORKS

Prompt Template

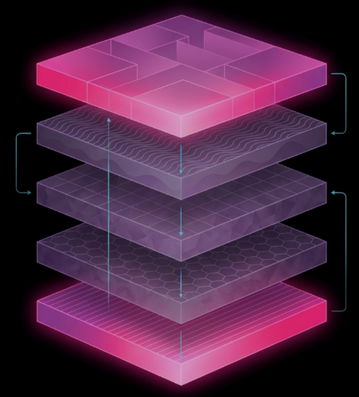
Putting it all together

1	**Task** explicit, positive framing	`Analyse Q4 sales trends...`
2	**Constraints** boundaries, specific limits	`Max 3 insights, cite data`
3	**Examples** show don't tell	`Input: X → Output: Y`
4	**Data** chunked, labelled	`<data>[content]</data>`
5	**Format** expected output structure	`Return JSON with...`

→ WHY THIS WORKS

- **Task first** - front-loads intent
- **Constraints early** - sets boundaries
- **Examples** - precise coordinates
- **Data labelled** - easy to reference
- **Format last** - back-loads output spec

PILLAR 2 RECAP



Structure your prompts.
Use examples. Position carefully.

Now let's build production systems...

BUILDING SYSTEMS

The Problem: Naive Approach

THE SCENARIO

"I have sales data. I want to ask questions in natural language and get insights."

NAIVE PROMPT

```
"Here's 50k rows of sales data. What are the trends?"
```

→ WHY IT FAILS

- ✕ **Context Rot**
  Too much data overwhelms the model
- ✕ **Lost in the Middle**
  Key patterns buried in data mass
- ✕ **No Decomposition**
  Asking for everything at once

BUILDING SYSTEMS

MCP Server

Model Context Protocol

Standardised way to connect LLMs to external tools and data sources

WITHOUT MCP

Dump all data into prompt

WITH MCP

Model queries what it needs

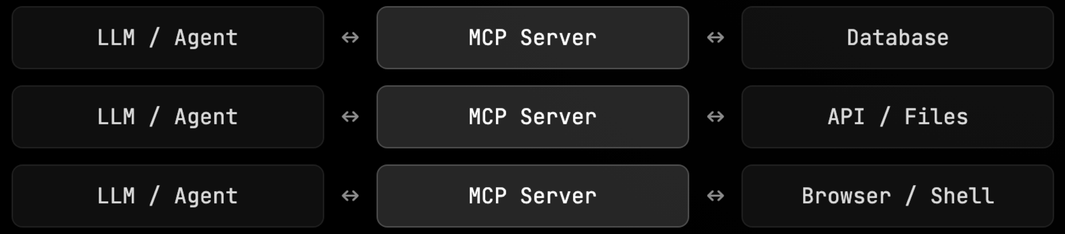
MCP gives models hands, not just eyes

→ SOLVES PROBLEM #1: CONTEXT ROT

- Data stays external
- Only relevant bits enter context
- Model can filter, aggregate, query
- Scales to any data size

BUILDING SYSTEMS

MCP Architecture



MCP SERVER PROVIDES

- **Tools** - functions model can call
- **Resources** - data model can read
- **Prompts** - templates to invoke

STANDARDISED PROTOCOL

```
JSON-RPC over stdio or HTTP+SSE
```

Same interface for any tool

BUILDING SYSTEMS

Sub-Agents

Divide and Conquer

→ SOLVES THE OTHER TWO PROBLEMS

- **Lost in Middle** - each agent has focused context
- **No Decomposition** - breaks task into pieces
- Parallel = faster execution
- Each agent uses MCP independently

MCP + Sub-agents = all three problems solved

BUILDING SYSTEMS

Agent Decomposition Patterns

Sequential

STEPS DEPEND ON EACH OTHER

Output feeds next agent

```
A → B → C → Result
```

+ Simple to debug
- Slower (serial)

Parallel

TASKS ARE INDEPENDENT

Spawn multiple, aggregate

```
Q → [A,B,C] → Merge
```

+ Fast (concurrent)
- Harder to coordinate

Hierarchical

COMPLEX ORCHESTRATION

Orchestrator delegates

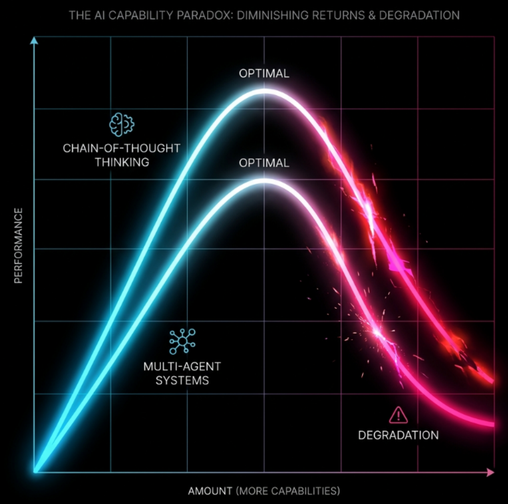
```
Orch → [A,B,C]
```

+ Flexible routing
- More overhead

BUILDING SYSTEMS

Limits of Stacking

When more becomes less



RECENT RESEARCH FINDINGS

- **More thinking can hurt** - CoT degrades on certain task types; models get distracted, overfit (Gema et al. 2025, Anthropic Fellows)
- **More agents can hurt** - novel failure modes lead to minimal gains with less stability (UC Berkeley, NeurIPS 2025)

Optimal point exists - beyond it, complexity causes degradation. But we can guard against this...

BUILDING SYSTEMS

Hooks

Guarding against failure modes

Shell commands that run before/after agent actions - automatic verification and error detection

PreToolCall
Validate inputs, gate risky operations

PostToolCall
Check outputs, trigger review systems

Notification
Alert on errors, escalate to humans

GUARDS AGAINST FAILURES

- Auto-lint catches code errors
- Output validation prevents drift
- Review triggers catch cascading issues
- Humans alerted when needed

Hooks turn failure modes into recovery opportunities

BUILDING SYSTEMS

Skills

Dynamic capabilities on demand

Reusable prompt templates that can be loaded dynamically based on situational demands

HOOKS + SKILLS = RESILIENCE

Hook detects issue → Triggers protocol → Loads appropriate skill → Sub-agent handles it

Complex workflows without human intervention

→ WHY SKILLS MATTER

- **Reuse** - proven prompts, not reinvention
- **Dynamic loading** - right capability, right time
- **Consistency** - same skill = same behaviour
- **Autonomy** - systems adapt without human instruction

Skills + Hooks = self-adapting systems

Key Takeaways

1. **Words activate neighbourhoods**
   Concepts cluster in non-obvious ways
2. **Every word matters**
   Positive framing, examples over descriptions
3. **Position and length matter**
   Start/end over middle, less is more
4. **Know the limits**
   More thinking/agents isn't always better
5. **Structure your prompts**
   Task → Constraints → Examples → Data → Format
6. **Productionise**
   MCP for data, sub-agents for parallelism, skills for reuse

Resources

CONTEXT ROT RESEARCH
research.trychroma.com/context-rot

LOST IN THE MIDDLE PAPER
arxiv.org/abs/2307.03172

INVERSE SCALING (ANTHROPIC 2025)
arxiv.org/abs/2507.14417

MULTI-AGENT FAILURES (NEURIPS 2025)
arxiv.org/abs/2503.13657

CLAUDE CODE DOCS
docs.anthropic.com/claude-code

Philipp Streicher, PhD

Questions? Find me after the session

philippstreicher.com