

LLM Models

A list of models

- [Gemini 2.5 Series](#)
- [Anthropic](#)
- [Claude 4](#)
- [Anthropic Claude 3 Opus](#)

Gemini 2.5 Series

Developer/Trainer	Google
Homepage	https://deepmind.google/models/gemini/pro/

Anthropic

Anthropic is a silicon valley startup and AI company that develops and sells LLM models and LLM-based products and services. They also developed the Model Context Protocol for passing information between models.

Key Information

- Founded as a competitor to OpenAI
- Developed the Claude model series, including Claude 3 Opus, Sonnet, and Haiku
- Known for safety-focused AI development approaches
- Created the Model Context Protocol (MCP) standard

Relationship to Other Models

Anthropic are one of the earliest competitors to OpenAI and their Claude Sonnet and Opus series models were early competitors to GPT-4.

Claude Models

The Claude model series includes several versions:

Claude 3 Series

- [Anthropic Claude 3 Opus](#)
- Anthropic Claude 3 Sonnet
- Anthropic Claude 3 Haiku

Other Models

- Anthropic Claude 2
- Anthropic Claude 1

Related Content

- Model Context Protocol

Claude 4

Claude 4 is a large language model developed by Anthropic.

Key Features

- Advanced reasoning capabilities
- Improved safety measures
- Better performance on complex tasks

Anthropic Claude 3 Opus

Overview

Claude 3 Opus is Anthropic's most powerful model, designed for highly complex tasks that require the highest levels of reasoning and understanding.

Capabilities

- Advanced reasoning and problem-solving
- Complex content creation and editing
- Comprehensive understanding of nuanced topics
- High-quality code generation and debugging

Performance

- More capable than Claude 3 Sonnet and Haiku
- Excellent at handling complex reasoning tasks
- Strong performance in benchmarks like MMLU and GSM8K

Use Cases

- Research and analysis
- Complex content creation
- Code generation with extensive reasoning

```
flowchart LR
```

```
A[Hard] -->|Text| B(Round)
```

```
B --> C{Decision}
```

```
C -->|One| D[Result 1]
```

```
C -->|Two| E[Result 2]
```