

# Prompt Cache Lab

Every request re-sends the whole context.

Caching makes the repeated part cheap.

Part A  
pricing

## What you pay for

Your message is the smallest part

Tools, system prompt, AGENTS.md and files go out **on every request**.

Everything is tokens

1 token  $\approx$  4 characters  $\approx$   $\frac{3}{4}$  of an English word.

Output costs more

Output costs  **$\sim 5\times$  input**. Thinking is billed as output.

Long chats get heavier

Every turn re-sends the **whole history**, including earlier replies.

Big context, check the rate

**Sonnet / Opus / Fable**: same rate up to 1M.

**Haiku 5.5**: over 100K input  $\rightarrow$  whole request  **$5\times$** .

**OpenAI**: over 272K input  $\rightarrow$  whole request  **$2\times$  input /  $1.5\times$  output**.

Part B  
caching

## How caching makes it cheaper

**Prefill: read first**

The model reads everything before answering. Caching reuses that work for an **identical start**.

**Cache layers, in order**

**Tools  $\rightarrow$  system prompt  $\rightarrow$  messages**.

Stable first, changing last.

**One word can break the cache**

Edit the start: everything **from the change onward** is processed and paid for again.

**Caches expire**

**Claude API**: 5 min default; 1 h option.

**Claude Code subscription**: main chat 1 h; subagents 5 min.

**OpenAI API**:  $\sim 30$  min. Every use restarts the timer.

**Cache reads are cheap**

**$2.5\text{--}10\%$**  of normal input price, depending on the model.

**Compaction**

Summarizes old turns: **fewer tokens, less detail**. Keep important things in a file like **AGENTS.md**.

**Effort**

Start at the **default**. Raise it only if the answer is not good enough.

## Use it in your next chat

**First request vs. reuse**

The first request pays for **prefill and a cache write**. Later turns can reuse the matching start cheaply.

**Before a big paste**

Send only what the task needs. Large logs stay in the context and are **billed on later turns**.

**New topic? New chat.**

Long chats use up limits faster, **even on subscription plans**.

**Compaction has a cost**

The model reads the history and writes a summary. That summary is **billed as output**.

**Read the full price row**

Check **input, cached input, output and the long-context rule**. Re-check prices before relying on them.

## Myth Buster

**"Caching saves the AI's answers"**  $\rightarrow$  It caches the input's start. Each new reply is generated fresh.

**"Max effort always gives the best answer"**  $\rightarrow$  Easy tasks gain nothing; they just take more time and money.

## A prompt to keep

[Stable: paste AGENTS.md team rules here]

[Stable: paste checkout-spec.pdf discount rules here]

Today's task: write test cases for stacked coupons.