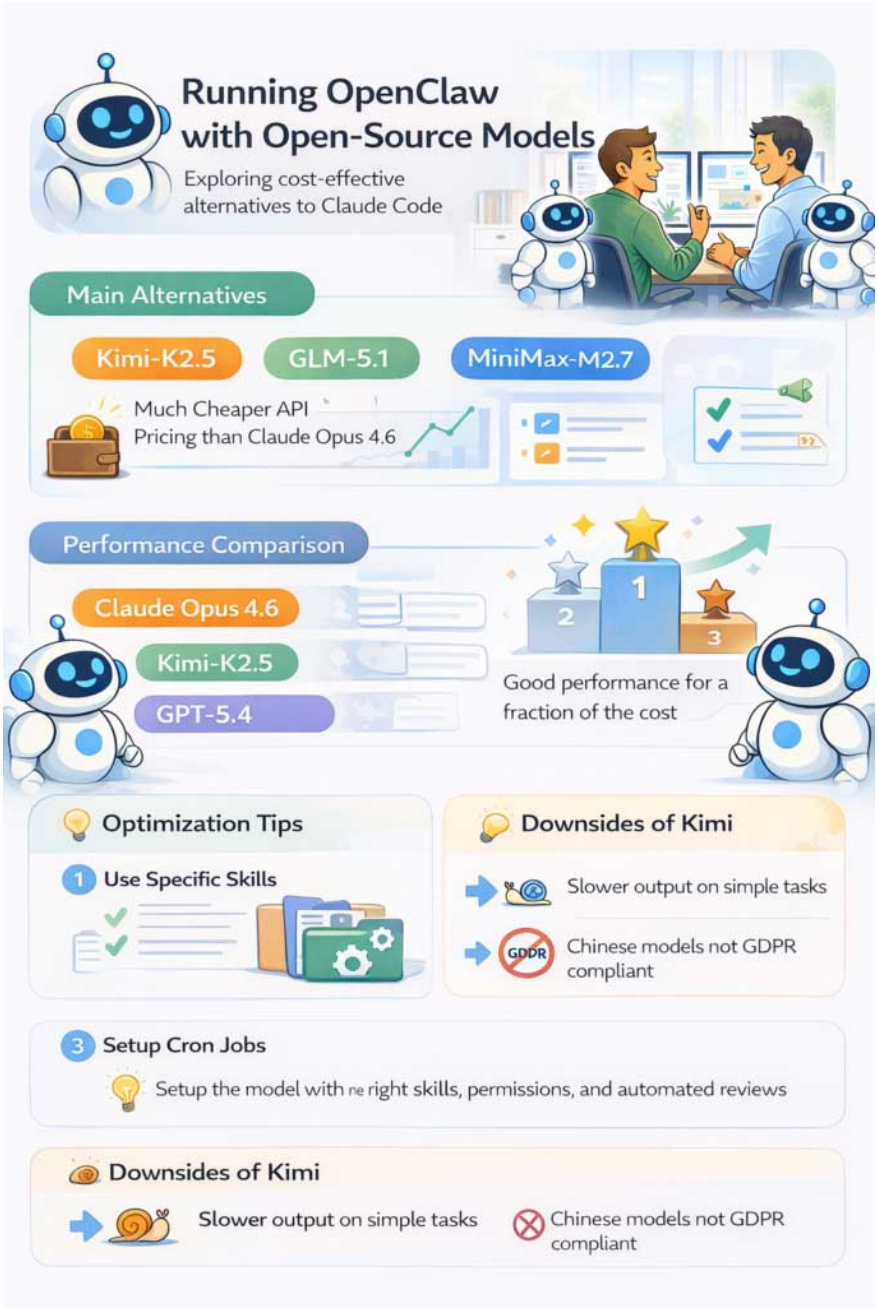
How to run OpenClaw with an open-source model: saving 90% on costs compared to Claude.

This guide explains how to use OpenClaw with open-source models like Kimi-K2.5, optimizing performance and reducing costs compared to Claude Opus.

Recently, Anthropic blocked the use of the Claude Code subscription package to run OpenClaw, forcing users to switch to the API. The problem lies in the cost. The Claude Opus 4.6 model currently costs around $5 for input and $25 for output per million tokens — a very high price if you use it regularly.

This led many people, including myself, to look for cheaper alternatives that still guaranteed performance. Initially, I tried GPT-5.4 but encountered a rather annoying problem: the model tended to be 'lazy,' easily giving up when faced with a problem requiring multiple processing steps. This was not suitable for an agent system like OpenClaw.

After that, I moved on to try other models, especially the Chinese models like Kimi-K2.5, GLM-5.1, and MiniMax-M2.7. Of these, Kimi and GLM are open source, while MiniMax is not.



Why should you use an open-source model with OpenClaw?

The biggest reason remains the cost. For example, Kimi-K2.5 costs only about $0.60 (input) and $3 (output) per million tokens — which is only about 1/10 the price of Claude Opus 4.6.

Furthermore, these models still perform well enough to handle many agent tasks, especially if you know how to optimize the system. Therefore, testing the Kimi-K2.5 as the 'brain' for OpenClaw is clearly the most logical choice.

How to integrate Kimi-K2.5 into OpenClaw

The actual setup process is quite simple. Instead of accessing it directly, you just need to use the OpenRouter platform to connect to the Kimi-K2.5. This platform is highly flexible, allowing switching between many different models, although there is an additional 10% intermediary fee.

Then, follow these steps:

1. Get the API key from OpenRouter
2. Reconfigure OpenClaw to use the new model.
3. Completely remove all configurations related to Anthropic.

This is a very important step. If you still have the old Anthropic key or reference in your system, you may encounter OAuth errors even after switching to a different model.

After clearing the old configuration, the transition was quite smooth. Claude Code can even support reconfiguring the system with just one request.

Performance of Kimi-K2.5 compared to other models

If we only consider actual performance, we can give the following subjective ranking:

1. Claude Opus 4.6
2. Kimi-K2.5
3. GPT-5.4

The gap between Claude and Kimi isn't huge, but Kimi outperforms GPT-5.4 in the agent context. A notable point is that Kimi tends to 'think' more than necessary, resulting in slower response times, even to simple questions.

However, in return, this model is more persistent and less likely to give up when handling complex tasks—a crucial factor for OpenClaw.

If cost is not a factor, Claude is still the best option. But at only one-tenth the price, the Kimi-K2.5 becomes a very formidable competitor.

How to optimize OpenClaw when using open-source models.

To achieve optimal results, you can't just change the model; you need to optimize the entire system. Some key principles include assigning specific skills to each task type, rather than handling everything generally. Additionally, you must ensure the agent has sufficient access rights, such as API keys for necessary services.

Additionally, setting up cron jobs so that agents can learn from chat history is also crucial. For example, you can have the system automatically review entire conversations daily to improve performance.

Interestingly, the techniques previously applied to Claude Opus still work well with Kimi-K2.5. This suggests that OpenClaw is quite 'model-agnostic', provided the model is strong enough in terms of reasoning and agent capabilities.

Limitations of Kimi-K2.5

Despite its many strengths, the Kimi-K2.5 still has some drawbacks.

First, there's the speed. The model responds significantly slower, especially to simple questions, due to the high consumption of 'thinking tokens'.

Secondly, there's the issue of GDPR compliance. When using these models via API, the data may not be processed within the EU, making them unsuitable for sensitive or customer data.

In theory, you could host the model yourself to solve this problem, but in return you would have to invest in infrastructure (GPU, server, etc.) and accept slower processing speeds.

Switching from Claude to the open-source model is not only a cost-effective choice but also opens up more flexibility in building agent systems.

Through testing, the Kimi-K2.5 has shown very good performance for its price, sufficient to become the 'brain' of OpenClaw in many cases.

Despite some limitations, if properly configured and optimized, OpenClaw, combined with open-source modeling, could become a powerful AI assistant at a much more reasonable cost.