



Sub-Kari

Using other LLMs as tools

- If Kani tools can be any Python code, and Kani is a Python library, can we use other Kanis as a tool?
- The answer is yes! For example, using a longer context model for processing a longer context temporarily
- This pattern can be used for self-organizing multi-agent systems

```
class KaniWithSummary(Kani):
    @ai_function()
    async def summarize_conversation(self):
        """Get the summary of the conversation."""
        long_context_engine = OpenAIEngine(api_key,
                                            model="gpt-4-32k")
        sub_kani = Kani(long_context_engine,
                        chat_history=self.chat_history[:-2])
        summary = await sub_kani.chat_round(
            "Please summarize the conversation so far.")
        return summary.content
```



Sub-Kani

Using other LLMs as tools

- If Kani tools can be any Python code, and Kani is a Python library, can we use other Kanis as a tool?
- The answer is yes! For example, using a longer context model for processing a longer context temporarily
- This pattern can be used for self-organizing multi-agent systems

```
class KaniWithSummary(Kani):
    @ai_function()
    async def summarize_conversation(self):
        """Get the summary of the conversation."""
        long_context_engine = OpenAIEngine(api_key,
                                           model="gpt-4-32k")
        sub_kani = Kani(long_context_engine,
                        chat_history=self.chat_history[:-2])
        summary = await sub_kani.chat_round(
            "Please summarize the conversation so far.")
        return summary.content
```


Complex Tasks & Agents