```
from typing import List, Optional

from griptape.structures import Agent as GriptapeAgent
from griptape.tools import FileManager, TaskMemoryClient, WebScraper

from swarms import Agent


class GriptapeAgentWrapper(Agent):
    """
    A wrapper class for the GriptapeAgent from the griptape library.
    """

    def __init__(
        self, name: str, tools: Optional[List] = None, *args, **kwargs
    ):
        """
        Initialize the GriptapeAgentWrapper.

        Parameters:
        - name: The name of the agent.
        - tools: A list of tools to be used by the agent. If not provided, default tools will be used.
        - *args, **kwargs: Additional arguments to be passed to the parent class constructor.
        """
        super().__init__(*args, **kwargs)
        self.name = name
        self.tools = tools or [
            WebScraper(off_prompt=True),
            TaskMemoryClient(off_prompt=True),
            FileManager(),
        ]
        self.griptape_agent = GriptapeAgent(
            input=f"I am {name}, an AI assistant. How can I help you?",
            tools=self.tools,
        )

    def run(self, task: str, *args, **kwargs) -> str:
        """
        Run a task using the GriptapeAgent.

        Parameters:
        - task: The task to be performed by the agent.

        Returns:
        - The response from the GriptapeAgent as a string.
        """
        response = self.griptape_agent.run(task, *args, **kwargs)
        return str(response)

    def add_tool(self, tool) -> None:
        """
        Add a tool to the agent.
        Parameters:
        - tool: The tool to be added.
        """
        self.tools.append(tool)
        self.griptape_agent = GriptapeAgent(
            input=f"I am {self.name}, an AI assistant. How can I help you?",
            tools=self.tools,
        )


# Usage example
griptape_wrapper = GriptapeAgentWrapper("GriptapeAssistant")
result = griptape_wrapper.run(
    "Load https://example.com, summarize it, and store it in a file called example_summary.txt."
)
print(result)
```