Additional Discussion to Reviewer v4G3's Questions

Q3: Detailed comparison with more related work

We compare OctoTools with the related works as suggested by the reviewers in Table 1.

	Training-Free	Dynamic Planning	Self-Refinement	Toolset Optimization	Extensible Tools	Runnable System	Comprehensive Evaluations	In-depth Study
OctoTools (ours)	✓	✓	✓	✓	✓	✓	✓, 16 tasks	✓
AutoGen	✓	✓	✗	✗	✓	✓	✗, 0 task	✗
GPT-Functions	✓	✓	✗	✗	✓	Limited	✗, 0 task	✗
LangChain	✓	✓	✓	✗	✓	✓	✗, 0 task	✗
EcoAct (Zhang et al., 2024)	✓	✓	✓	✗	✗	✗	✗, 1 task only	Limited
TPTU-v2 (Kong et al., 2023)	✗	✗	✗	✓	Limited	✗	✗, 1 task only	Limited
HuggingGPT (Shen et al., 2023)	✓	✗	✗	✗	✗	✗	✗, 0 task	✗
TaskMatrix.AI (Liang et al., 2024)	✗	✗	✗	✗	✗	✗	✗, 0 task	✗
Magnetic-One (Fourney et al., 2024)	✓	✓	✗	✗	✓	✓	✗, 2 tasks only	✗
AutoAgents (Chen et al., 2023)	✗	✗	✗	✗	✗	✗	✗, 2 tasks only	✗

Table 1. Comparison with more existing works.

Notations:

- **Training-free:** The framework can be deployed or extended with new tools without any additional training or fine-tuning of the language model.
- **Dynamic planning:** The system adaptively updates or refines its plan (including tool usage) based on intermediate observations or feedback during the reasoning process.
- **Self-refinement:** At each step, the agent can correct or refine its previous reasoning to address errors, inconsistencies, or missing information in earlier steps.
- **Toolset optimization:** There is an explicit mechanism (e.g., a lightweight selection algorithm) that identifies the most useful subset of tools for a given domain or task, with the guarantee of the performance gain based on the validation.
- **Extensible tools:** A wide range of tools (e.g., Python, web-search, vision models) can be added via standardized interfaces ("tool cards"). Introducing a new tool does not require changes to the core planner–executor logic.
- **Runnable system:** A publicly accessible or easily deployable agentic framework is provided so that others can run, test, and build upon it for research or practical applications.
- **Comprehensive evaluations:** The framework is rigorously tested on diverse and challenging benchmarks (OctoTools demonstrates results on **16 tasks**), showcasing consistent gains and broad generalization.
- **In-depth study:** Thorough analyses and ablations are presented (e.g., on multi-step reasoning, task planning, tool usage) that offer insights into the system's capabilities, limitations, and design trade-offs, along with the behavior difference over other frameworks.

To sum up, while OctoTools shares the broad concept of "planner–executor" with past works, our training-free nature, self-refinement loop, lightweight toolset optimization, and in-depth benchmarking offer distinct contributions that go beyond traditional approaches like TPTU-v2. We hope this clarifies how our system expands the boundaries of agentic tool usage and provides a fresh perspective on designing robust, extensible frameworks for complex reasoning.

Q2: Detailed comparison with TPTU-v2

We provide a detailed comparison between OctoTools and TPTU-v2 (Kong et al., 2023) in Table 2.

Features	OctoTools	TPTU-v2	Justification for TPTU-v2
Training-Free	Yes	No	Requires additional fine-tuning for task planning and API calling, rather than being training-free.
Dynamic Planning	Yes	No	Relies on static plans with limited iterative updates, whereas OctoTools adapts on-the-fly.
Self-Refinement	Yes	No	Lacks a mechanism to correct or refine reasoning mid-execution.
Toolset Optimization	Yes	Yes	Employs an "API Retriever" to select relevant APIs, similar to our validation-based selection.
Extensible Tools	Yes	Limited	Focuses on a fixed set of APIs rather than a standardized plug-and-play interface.
Runnable System	Yes	No	Not released as a publicly accessible framework, though validated in real-world settings.
Comprehensive Evaluations	Yes, 16 tasks	No, 1 task only	Evaluated on a single task domain, whereas OctoTools spans 16 diverse benchmarks.
In-Depth Study	Yes	Limited	Offers less extensive analysis and ablations compared to OctoTools' multi-faceted evaluations.

Table 2. Detailed comparison between OctoTools and TPTU-v2.

References

Chen, G., Dong, S., Shu, Y., Zhang, G., Sesay, J., Karlsson, B. F., Fu, J., and Shi, Y. Autoagents: A framework for automatic agent generation. *arXiv preprint arXiv:2309.17288*, 2023.

Fourney, A., Bansal, G., Mozannar, H., Tan, C., Salinas, E., Niedtner, F., Proebsting, G., Bassman, G., Gerrits, J., Alber, J., et al. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468*, 2024.

Kong, Y., Ruan, J., Chen, Y., Zhang, B., Bao, T., Shi, S., Du, G., Hu, X., Mao, H., Li, Z., et al. Tptu-v2: Boosting task planning and tool usage of large language model-based agents in real-world systems. *arXiv preprint arXiv:2311.11315*, 2023.

Liang, Y., Wu, C., Song, T., Wu, W., Xia, Y., Liu, Y., Ou, Y., Lu, S., Ji, L., Mao, S., et al. Taskmatrix. ai: Completing tasks by connecting foundation models with millions of apis. *Intelligent Computing*, 3:0063, 2024.

Shen, Y., Song, K., Tan, X., Li, D., Lu, W., and Zhuang, Y. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180, 2023.

Zhang, S., Zhang, J., Ding, D., Garcia, M. H., Mallick, A., Madrigal, D., Xia, M., Rühle, V., Wu, Q., and Wang, C. Ecoact: Economic agent determines when to register what action. *arXiv preprint arXiv:2411.01643*, 2024.