EVALUATING AGENTIC AI WORKFLOWS

Local vs. Web-Based Environments

Authors Bryan Wang[1,3]; Stephen T.C. Wong, Ph.D., P.E.[2]

Affiliations [1]William P. Clements High School, Sugar Land, TX;
[2]Methodist Research Center, Houston, TX;
[3]Gifted and Talented Mentorship Program, Fort Bend ISD, TX

Introduction

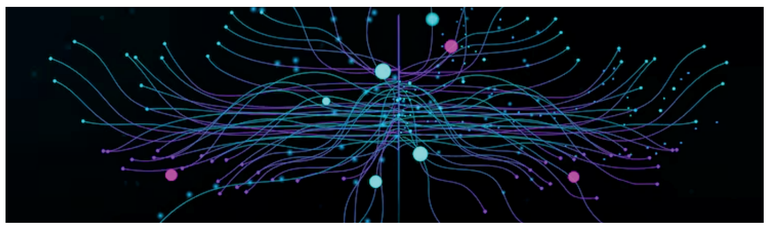
AI systems are increasingly being used for tasks that require more than one-step text generation, especially in research and industry settings where workflows depend on files, tools, and multi-step execution [1]. An important question is not only how well a model performs, but also what kind of environment it works in. Local systems can offer more direct access to tools and software, while web-based systems are often easier to use but more limited [1, 2].

To examine this difference in practice, this project compared local and web-based AI workflows using OpenClaw and the ChatGPT website agent mode. The study focused on autonomous model-training tasks, including image classification, tabular classification, tabular regression, and reinforcement learning, in order to evaluate how environment design affected implementation, flexibility, and final performance [3, 4, 5, 6].

Methodology

This study used a comparative workflow evaluation to test how well agentic AI systems can complete simple machine learning tasks. OpenClaw specialist-agent workflows were evaluated on image classification, tabular classification, tabular regression, and reinforcement learning, then compared with ChatGPT website agent-mode workflows using the same task prompts and standardized inputs where possible. OpenClaw workflows were powered with GPT-5.4 models for consistency.

1. Specialist OpenClaw agents and ChatGPT website agent-mode chats were created for each machine learning task type
2. Each workflow received the same task instructions and, when possible, the same provided datasets or benchmark environment.
3. Systems were asked to inspect the input, choose a beginner-friendly baseline, train the model, evaluate it, and summarize the results.
4. Workflows were compared on autonomy, task completion, human intervention required, model performance, and explanation quality.



Results

Task	Metric	OpenClaw	GPT Website (Agent Mode)
Image Classificaton (Raw Pixels)	Accuracy	52.75%	53%
Image Classification (HOG)	Accuracy	69%	69%
Tabular Classification	Accuracy	93.3%	93.3%
Tabular Regression	R^2/RMSE	0.453/53.85	0.45/53.85
Reinforcement Learning	Avg Reward	493 (Gymnasium + Stable-Baselines3)	112 (Custom DQN)

Across image classification, tabular classification, and tabular regression, OpenClaw and GPT web agents produced very similar results when given the same provided inputs and comparable baseline methods. In image classification, both systems initially performed poorly with a raw-pixel logistic regression baseline, achieving only about 53% accuracy, but both improved to about 69% after a follow-up step using HOG-based preprocessing.

The largest difference appeared in reinforcement learning. OpenClaw was able to use standard RL libraries, including Gymnasium for the CartPole benchmark environment and Stable-Baselines3 for a tested PPO implementation, which allowed it to achieve near-solved performance (average reward $\approx$ 493). By contrast, the GPT web agents could not install those packages in its hosted environment and lacked internet access to install them, instead having to rely on a self-implemented CartPole environment and a manual DQN baseline which only reached about 112 average reward.

Findings

OpenClaw's main advantage is that it runs in a local environment where the level of autonomy could be configured. In this study, it was set up in a relatively permissive mode, allowing local file access, coding tools, internet-connected package installation, and the use of dedicated virtual environments. This made it possible for workflows to install missing libraries and, when automatic setup failed, allowed the user to manually enter the VPS or device and install dependencies directly. GPT web agents, by contrast, ran in a sandboxed hosted container with no internet access, limited preinstalled libraries, and no ability to change the underlying system configuration. This made the GPT web agents easier to use, but also more restricted.

This difference mattered most in reinforcement learning. OpenClaw could directly use Gymnasium and Stable-Baselines3 to train a much stronger CartPole agent, while GPT web agents had to manually recreate the environment and algorithm because those packages were unavailable. Overall, OpenClaw allowed a tradeoff between stronger autonomy and higher security risk, while GPT website mode offered stronger built-in isolation but less flexibility.

Conclusion

This study found that the main difference between OpenClaw and the ChatGPT website agent-mode was not always final model accuracy, but the type of environment each system could operate in. On image classification, tabular classification, and tabular regression, both systems achieved very similar results when given the same inputs and baseline methods. The largest difference appeared in reinforcement learning, where OpenClaw's local, configurable environment allowed it to use standard RL libraries and achieve much stronger performance. Overall, the results suggest that local agent frameworks such as OpenClaw offer greater flexibility, tool access, and implementation control, while web-based systems are easier to use and more isolated, but more limited for tasks that depend heavily on packages, runtime access, and reproducible workflows.

References

1. Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N., Chen, A. S., Creel, K., Davis, J. Q., Demszky, D., … Liang, P. (2021). On the opportunities and risks of foundation models. arXiv. https://arxiv.org/abs/2108.07258
2. OpenClaw. (n.d.). OpenClaw documentation. https://docs.openclaw.ai (https://docs.openclaw.ai/)
3. Farama Foundation. (n.d.). Cart Pole. Gymnasium documentation. https://gymnasium.farama.org/environments/classic_control/cart_pole/
4. Scikit-learn developers. (n.d.). The iris dataset. Scikit-learn documentation. https://scikit-learn.org/stable/auto_examples/datasets/plot_iris_dataset.html
5. Scikit-learn developers. (n.d.). load_diabetes. Scikit-learn documentation. https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_diabetes.html
6. Zenodo. (2021). Cats and dogs light dataset sample [Data set]. https://zenodo.org/records/5226945