AdaptAgent: Adapting Multimodal Web Agents with Few-Shot Learning from Human Demonstrations

Gaurav Verma[1] **Rachneet Kaur**[2] **Nishan Srishankar**[2]
Zhen Zeng[2] **Tucker Balch**[2] **Manuela Veloso**[2]
[1]Georgia Institute of Technology [2]J.P. Morgan AI Research
gverma@gatech.edu {rachneet.kaur, nishan.srishankar}@jpmorgan.com
{zhen.zeng, tucker.balch, manuela.veloso}@jpmorgan.com

Abstract

State-of-the-art multimodal web agents, powered by Multimodal Large Language Models (MLLMs), can autonomously execute many web tasks by processing user instructions and interacting with graphical user interfaces (GUIs). Current strategies for building web agents rely on *(i)* the generalizability of underlying MLLMs and their steerability via prompting, and *(ii)* large-scale fine-tuning of MLLMs on web-related tasks. However, the ability of web agents to automate tasks on unseen websites and domains remains lacking, limiting their applicability to enterprise-specific and proprietary websites/domains. Beyond generalization from large-scale pre-training and fine-tuning, we propose building agents for few-shot adaptability using human demonstrations. We introduce the `AdaptAgent` framework that enables both proprietary and open-weights multimodal web agents to adapt to new websites and domains using few human demonstrations (up to 2). Our experiments on two popular benchmarks — Mind2Web & VisualWebArena — show that using in-context demonstration (for proprietary models) or meta-adaptation demonstrations (for meta-learned open-weights models) boosts task success rate by 3.36% to 7.21% over non-adapted state-of-the-art models, corresponding to a relative increase of 21.03% to 65.75%. Our results unlock a complementary axis for developing widely applicable multimodal web agents beyond large-scale pre-training and fine-tuning, emphasizing few-shot adaptability.

1 Introduction

Agents automating web-based tasks with minimal human intervention can significantly boost personal and workplace productivity [34, 35]. A prevalent interaction involves a human providing a natural language instruction (e.g., *"use delta.com to book a flight from JFK to Haneda on ..."*), and the agent autonomously executing the necessary webpage actions to complete the user-assigned task [59, 13, 21]. Large language models (LLMs) can understand instructions, plan, and predict structured outputs, serving as backbones for such agents [52]. Remarkable progress has been made in automating web-based tasks using LLM-based agents [29, 11, 19], employing careful prompting [59, 27] and extensive pre-training and fine-tuning [13] to predict actions using language instructions and HTML/DOM. With multimodal capabilities, these agents now process the graphical user interface's (GUI's) visual state to complement the HTML/DOM information [21]. In parallel with the methodological advancements, evaluating the generalizability of these multimodal web agents to new tasks, websites, and domains is a critical component to ensure their broad applicability.

Prior works have noted challenges in generalizing multimodal web agents to new tasks, websites, and domains, often relying on large-scale pre-training (e.g., agents like SeeAct [59]) or fine-tuning (e.g., models like CogAgent [21]). We posit that regardless of pre-training scale, some tasks and domains will remain unseen, such as proprietary workflows and enterprise websites. Since the generalizability of current state-of-the-art (SoTA) agents is limited and their fine-tuning is costly, we propose *building web-agents for data-efficient adaptability* instead of relying solely on large-scale pre-training and

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

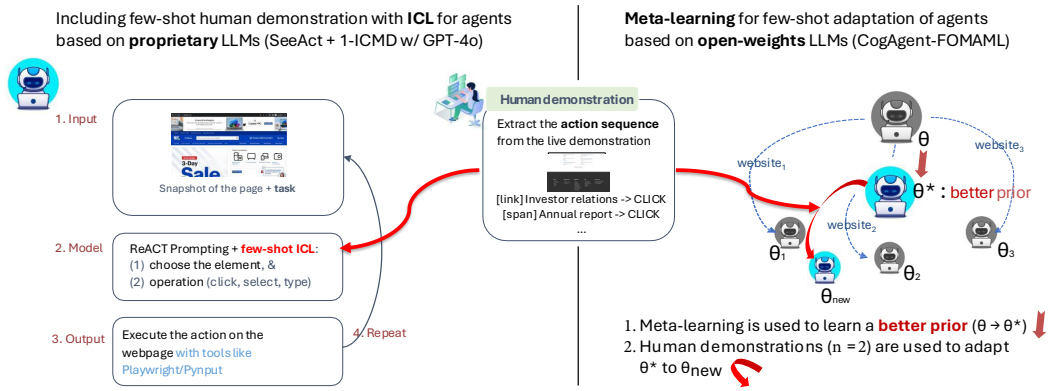

Figure 1: **AdaptAgent** for few-shot adaptation of web agents that are based on proprietary and open-weights multimodal LLMs. **Left**: For proprietary MLLM-based web agents, we include the multimodal human demonstration as in-context examples. **Right**: For web agents based on open-weights MLLMs, we first learn a better prior using meta-learning and then use few-shot human demonstrations for faster adaptation.

fine-tuning. Specifically, we address whether multimodal web agents can adapt to unseen websites and domains with only a handful of human demonstrations (e.g., $n = 1$ or $n = 2$). We discuss the prior related work in Appendix A.1, where we elaborate on how robot learning from demonstration inspired the `AdaptAgent` framework.

We consider current SoTA multimodal web agents — both proprietary and open-weights — and demonstrate that incorporating just 1 or 2 multimodal human demonstrations (visual snapshot + HTML information) can result in an absolute increase in task success rate of 3.36% to 7.21% on unseen websites and domains, corresponding to a relative increase of 21.03% to 65.75% over current performance. We propose the `AdaptAgent` framework to effectively use these few-shot demonstrations through careful in-context learning (ICL) [7] with proprietary multimodal LLMs (MLLMs) and meta-learning [16] with open-weights multimodal LLMs. To establish the role of learning from few-shot demonstrations, we conduct extensive experiments on two widely adopted benchmarks — Mind2Web [13] & VisualWebArena [27] — showing improvements across tasks of varying difficulty levels. Our key contributions are below:

- We propose the `AdaptAgent` framework for enabling SoTA multimodal web agents to learn from few-shot human demonstrations. `AdaptAgent` uses ICL for data-efficient adaptation of proprietary MLLMs like GPT-4o [2] and meta-learning for adapting open-weights MLLMs like CogAgent [21].
- Our extensive experiments on Mind2Web and VisualWebArena demonstrate the effectiveness of our methods, resulting in notable increases in task success rates on unseen websites and domains with only 1 or 2 multimodal demonstrations.[1]

We believe that the effectiveness of using few-shot human demonstrations and our empirical insights open a complementary direction for improving the generalizability of multimodal web agents beyond the current SoTA strategies that rely on large-scale pre-training and fine-tuning.

2 Few-Shot Adaptation with Human Demonstrations

Methodological motivation. Learning from human demonstrations [44] has played a key role in many applications, notably helping robots generalize to new tasks or existing tasks under new environments and constraints [3]. Prior work has highlighted the limited generalizability of web agents to unseen tasks, websites, and domains [59, 21]. Agents that automate web tasks and robots that automate real-world tasks share strong analogies in desired capabilities (i.e., perception, reasoning, execution [52]), allowing for transfer of modeling strategies between these domains. This inspires us to adopt learning from human demonstrations for web agents to improve their adaptability to unseen settings. While it's possible to fine-tune web agents with a large number of human demonstrations covering new websites and domains, such approaches require tedious annotations and are expensive. Therefore, building highly adaptable web agents requires the ability to adapt them in a *data-efficient* manner.

Despite the success of learning from demonstration in adapting robots and the strong analogies between physical robots and web agents, unique challenges remain for web agents. Traditionally, robot learning from human demonstrations exhibits limited generalizability; i.e., when a human demonstrates task $\mathcal{A}$ a few times, the robot learns to do the same task $\mathcal{A}$ or closely related tasks, akin to imitation learning [23, 41]. It remains to be seen how well web agents can generalize with few-shot human demonstrations, which is the primary focus of this work. In other words, can a handful of human demonstrations of specific tasks on certain websites (e.g., *"book a flight..."* on *delta.com*)

[1] For a more granular investigation of the observations, we conduct ablations to break down the main results, stratifying improvements based on action sequence complexity and visual difficulty. We also quantify the effects of more in-context demonstrations and different data mixes during meta-learning. See Appendix A.5.

lead the web agent to learn related tasks on similar websites (e.g., *"check visa requirements..."* on *united.com*), or even generalize to unrelated domains (e.g., *"book a driving test appointment..."* on *dol.wa.gov*)? Our work proposes `AdaptAgent`, a framework to enable web agents to adapt with few-shot human demonstrations and evaluates their generalizability to unseen settings.

Methods for learning with human demonstrations. Our proposed framework for adapting multimodal web agents with few-shot human demonstrations builds on advances in proprietary and open-weights multimodal LLMs. We use SeeAct [59], which employs a carefully crafted prompting strategy with GPT-4o, as a representative proprietary model baseline and adapt it using multimodal in-context examples. As the representative baseline for SoTA open-weights models, we use CogAgent [21] — an 18B multimodal LLM with a dedicated visual backbone to process GUI images. Given the success of meta-learning in efficient adaptation, we propose fine-tuning models like CogAgent with meta-learning instead of regular fine-tuning. See Figure 1 for an overview of our proposed `AdaptAgent` framework. Next, we elaborate on the methodological details for in-context learning and meta-learning with human demonstrations for proprietary and open-weights models, respectively.

1. In-context learning with SeeAct + GPT-4o: SeeAct uses a carefully constructed prompt (using ReAct prompting [56]) to guide multimodal LLMs like GPT-4o in iteratively determining the next action based on the current GUI state to complete the user-assigned task. In-context learning (ICL) has proven to be an effective approach for adapting proprietary LLMs [4]. Consequently, we deconstruct the human demonstration of a task on the target website/domain into a sequence of (visual snapshot, HTML elements (filtered following [59]) , human selection) and include them as an ICL example with the original SeeAct prompt; see Appendix A.6 and Figure 1 (left).

2. Meta-learning with CogAgent: To overcome the limited abilities of general-purpose multimodal LLMs to process GUI snapshots — which involve complex layout understanding, OCR, and functional understanding of HTML elements, Hong et al. (2023) [21] pre-trained general-purpose MLLMs like CogVLM [53] on tasks involving GUI processing. Beyond extensive pre-training, fine-tuning on task-specific datasets showed notable performance boosts for CogAgent over several baselines. In this work, we consider the pre-trained CogAgent and further adapt it using model-agnostic meta-learning (MAML) [15] with few-shot human demonstrations; refer to Figure 1 (right) for a visual depiction.

Meta-learning [46], often dubbed "learning to learn", is a training strategy in which a model learns to adapt efficiently to unseen tasks by leveraging knowledge gained from updates across many related tasks. Model-agnostic meta-learning [15] is one such approach applicable to any model. Mathematically, the meta-learned model θ^* is obtained via meta-updates $\theta \leftarrow \theta - \beta \cdot \nabla_\theta \sum_i^N \mathcal{L}_{\mathcal{T}_i}(\theta_i)$ (outer loop update), where β is the meta-learning step size, and the gradient is derived from the sum of losses $\mathcal{L}_{\mathcal{T}_i}(\theta_i)$ across all tasks. Each θ_i is initialized from θ and fine-tuned on task $\mathcal{T}_i$, via $\theta_i \leftarrow \theta - \alpha \cdot \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(\theta)$ (inner loop update), with α being the step size. Thus, each meta-update involves meta-gradients (gradients through gradients). However, since our experiments involve LLMs with billions of parameters, computing meta-gradients is computationally challenging. Therefore, we consider the first-order approximation of model-agnostic meta-learning (FOMAML). FOMAML has demonstrated performance on par with MAML [15, 33], potentially due to the predominantly locally linear nature of neural networks [17, 40], making the second-order gradients negligible. Therefore, our meta-learning updates are represented as (derivation in App. A.2): $\theta \leftarrow \theta - \beta \cdot \sum_{i=1}^N \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(\theta_i)$. In other words, when adapting multimodal web agents with meta-learning, the inner loop involves fine-tuning the agent ($\theta \rightarrow \theta_i$) on web tasks $\mathcal{T}_i$ from a given website, with the training subset used for this inner loop denoted as $\mathcal{D}_i^{train}$. Then, for the outer loop update, we update the parameters of the MLLM agent θ by backpropagating the gradients of the loss at θ_i, where the loss is computed on held-out web tasks from the same website/domain — denoted as $\mathcal{D}_i^{test}$. Importantly, the gradients being backpropagated are computed at θ_i (rather than θ), ensuring the MLLM agent is not trained on both $\mathcal{D}_i^{train}$ & $\mathcal{D}_i^{test}$. Essentially, we train the MLLM agent θ on $\mathcal{D}_i^{train}$ to obtain θ_i and then update its *original* parameters θ using penalties based on how well θ_i performs on held-out $\mathcal{D}_i^{test}$. A better $\theta*$ serves as a better starting point to arrive at better θ_i through fine-tuning, leading to less penalties on held-out $\mathcal{D}_i^{test}$. This ensures quick and data-efficient adaptation of the agent to unseen websites.

3 Experimental Protocol and Details

Datasets: To evaluate the quick adaptation capabilities of our agents, we design experiments that require adaptation to unseen websites and domains. We consider two widely used benchmarks: Mind2Web [13] and VisualWebArena [27]. ***Mind2Web*** provides standardized train and test sets across various websites and domains. The train set includes 1,009 tasks from 73 websites and 3

domains, while the test set is categorized into cross-task (174 tasks from 64 seen websites), cross-website (142 tasks from 10 unseen websites), and cross-domain (694 tasks from 2 unseen domains) subsets to evaluate different aspects of generalization. Since the cross-task evaluation set overlaps with the train set, we propose minor amendments to ensure proper evaluation of adaptability (details in Appendix A.3). ***VisualWebArena*** simulates a live environment with three different websites (Reddit, Classifieds, and Shopping) to evaluate task success rates of web agents. We use the entire VisualWebArena benchmark (910 tasks) to test the adaptability of our web agent to unseen websites. While some tasks have step-level ground truth, others provide only an overall task success signal based on the environment's state. More details about the datasets are presented in App. A.3.

Experimental Protocol: Our experimental protocol for developing and evaluating the adaptability of web agents varies based on whether the underlying multimodal LLM is proprietary or open-weights. For the **proprietary** model (i.e., GPT-4o), we use the prompting method proposed in SeeAct and add one ICL example from the website or domain to which the agent should adapt. This ICL example acts as the one-shot ($n = 1$) human demonstration (denoted as 1-ICMD for 1 in-context multimodal demonstration). We adopted a one-shot setting for ICL given the trade-off between time and incremental accuracy improvements; see App. A.5.4. The selection of the ICL example ensures relevance to the cross-task, cross-website, and cross-domain evaluation setups. Specifically, for Mind2Web's *cross-task* and *cross-website* evaluation, we randomly sample one task from the same website (for cross-task) or from each unique website (for cross-website) in the test set and evaluate on the remaining examples from that website, maintaining website-level correspondence. For *cross-domain evaluation*, we randomly sample one task from each unique domain in the cross-domain test set and evaluate on the remaining examples from that domain. For *VisualWebArena* evaluation, we randomly choose one task as the in-context demonstration from the website being evaluated. For the **open-weights model** (i.e., CogAgent), during meta-learning, we sample 4 tasks per website from the 73 websites in the Mind2Web training set: 2 tasks for adaptation ($\mathcal{D}_i^{train}$) and 2 tasks ($\mathcal{D}_i^{test}$) (1 from the same website and 1 from a different website within the same domain) for computing the adaptation loss and updating the agent's parameters as discussed in Section 2. After meta-learning, the meta-trained model adapts to new websites in the cross-website test set by fine-tuning on 2 tasks from each website and then evaluating on the remaining tasks. For cross-domain evaluation, we adapt on 2 tasks from each new domain and evaluate on the rest within that domain; see Figure 2. We do not perform website adaptation for the cross-task test set, as all websites are seen during meta-learning. For VisualWebArena, we adapt the meta-trained model on the Mind2Web training set, using 2 tasks from each of the 3 websites and evaluate on the remaining tasks. To control for the effect of adaptation tasks, we report average results across 5 independent runs with different task selections. Overall, our approach involves meta-training the model with 292 tasks from Mind2Web (73 websites × 4 tasks) and adapting with 2 demonstrations to new websites/domains. Implementation details are available in App A.4. We denote our meta-learned and adapted agent as CogAgent-FOMAML.

We compare the performance of adapted agents with existing SoTA agents as **baselines**. For the proprietary model, zero-shot SeeAct + GPT-4o serves as a baseline. We also include Set-of-Mark prompting (SoM) [55, 27] in the image input, giving us a slightly augmented baseline that we denote as SeeAct*. For the open-weights model, we consider the pre-trained CogAgent and another variant—CogAgent-FT—that uses conventional fine-tuning on the entire train set of Mind2Web as baselines. Additionally, we consider CogAgent-FT (DE) as another baseline that maintains data equivalence (DE) with the proposed CogAgent-FOMAML method by using the same training subset for conventional fine-tuning. CogAgent-FOMAML and CogAgent-FT (DE) use 292 examples during meta-learning and fine-tuning, respectively, while CogAgent-FT uses ∼3.4× as many examples.

Evaluation metrics: For evaluation on the Mind2Web test sets, since the ground-truth human trajectories are available for each task, we compute granular metrics: the accuracy of predicting the correct HTML element (Ele. Acc.) to act on; the F_1 score of predicting the correct operation (Op. F_1) such as click, select, type; the percentage of successful steps (Step SR) — requiring correct prediction of the element, the operation, and the optional type/selection text; and the percentage of successful tasks (Overall SR), where task-level success is achieved only if the entire sequence of steps predicted by the agent aligns with the ground-truth human trajectories. For VisualWebArena, since the ground-truth human trajectories are available only for a subset of the data (233 tasks corresponding to the unique templates) and the rest of the tasks have only a task-level success signal within the live environment, we use the overall success rate as the primary metric while also quantifying the granular metrics specifically for the subset of tasks with human trajectories.

4 Results

Few-shot human demonstrations unlock complementary gains: Table 1 compares the baseline and few-shot adapted versions of proprietary (SeeAct, SeeAct*) and open-weights (CogAgent) models on *(a)* the Mind2Web dataset across cross-task, cross-website, and cross-domain evaluation settings, and *(b)* the VisualWebArena dataset across human trajectories and live environment settings. The proprietary models adapt through multimodal in-context demonstration, while CogAgent adapts via meta-learning. Recall that for CogAgent, we tested two baseline versions: one fine-tuned on the entire Mind2Web train set and another to ensure date-equivalence with CogAgent-FOMAML.

Type	Model	Cross-Task				Cross-Website				Cross-Domain			
		Ele. Acc.	Op. F1	Step SR	Overall SR	Ele. Acc.	Op. F1	Step SR	Overall SR	Ele. Acc.	Op. F1	Step SR	Overall SR
Proprietary Models													
Baseline	SeeAct (GPT-4o)	62.21	66.56	56.31	14.37	55.25	58.89	49.90	15.83	57.33	60.74	53.72	19.49
Adapted	SeeAct + 1-ICMD	**66.29**	**71.61**	**60.37**	**19.69**	**60.32**	**64.15**	**53.91**	**22.46**	**60.54**	**62.97**	**57.40**	**23.97**
Baseline	SeeAct* (GPT-4o)	63.75	67.68	58.60	15.38	57.02	60.01	50.05	15.89	59.30	62.80	54.82	19.88
Adapted	SeeAct* + 1-ICMD	**67.77**	**72.52**	**61.88**	**22.46**	**61.67**	**64.76**	**53.98**	**23.10**	**62.44**	**65.41**	**58.33**	**24.06**
Open-weights Models													
Baseline	CogAgent-FT	**59.46**	**63.15**	**54.43**	**13.36**	53.17	57.03	47.14	12.42	61.36	62.79	55.71	15.20
	CogAgent-FT (DE)	55.17	59.87	50.25	10.43	49.46	53.17	44.27	10.10	59.51	59.06	52.20	13.28
Adapted	CogAgent-FOMAML	59.34	62.82	53.32	11.89	**59.49**	**62.11**	**55.38**	**16.96**	**62.01**	**63.13**	**57.29**	**19.66**

(a) Mind2Web dataset

Type	Model	Human Trajectories				Live Environment
		Ele. Acc.	Op. F1	Step SR	Overall SR	Overall SR
Proprietary Models						
Baseline	SeeAct (GPT-4o)	56.03	57.17	52.17	18.75	17.56
Adapted	SeeAct + 1-ICMD	**59.15**	**63.18**	**55.27**	**22.42**	**21.36**
Baseline	SeeAct* (GPT-4o)	57.52	59.16	53.16	18.78	18.04
Adapted	SeeAct* + 1-ICMD	**61.46**	**64.12**	**56.72**	**23.86**	**23.15**
Open-weights Models						
Baseline	CogAgent-FT	52.31	55.64	48.70	08.78	06.43
	CogAgent-FT (DE)	48.62	51.71	44.81	06.81	05.11
Adapted	CogAgent-FOMAML	**57.20**	**59.14**	**51.29**	**11.01**	**08.47**

(b) VisualWebArena dataset

Table 1: Effect of few-shot adaptation of web agents; all values are percentages. ICMD denotes the multimodal in-context demonstration. FT refers to fine-tuning, DE denotes fine-tuning with data equivalence with respect to our meta-learned models. **Adapted** models are our proposed methods. **Bold** indicates best performance, and orange highlight represents the best overall performance. Model size of GPT-4o: 175B; CogAgent: 18B.

We observe that few-shot adaptation improved the performance of both proprietary and open-weights models across the two datasets and all settings involving adaptation to unseen websites or domains. Specifically, for Mind2Web's cross-website and cross-domain sets, few-shot adaptation using the `AdaptAgent` framework resulted in an absolute increase in overall success rate ranging from 4.18% to 7.21% over the corresponding unadapted counterparts, which corresponds to a relative increase of 21.03% to 45.40% over the current state-of-the-art. The trends are consistent across all the models, demonstrating the effectiveness of using only 1 or 2 human demonstrations via `AdaptAgent`. Similarly, on VisualWebArena, `AdaptAgent` led to an absolute increase in overall success rate ranging from 3.36% to 5.11%, which corresponds to 28.32% to 65.75% relative increase over the SoTA approaches.[2] In Appendix A.5, we provide further investigations on the effect of few-shot adaptation on tasks of varying difficult levels, the role of number of in-context demonstrations used, the advantage of multimodal in-context demonstrations compared to text-only ones, and the role of different data selection strategies during meta-learning.

In ***conclusion***, we propose the `AdaptAgent` framework ,which uses few-shot human demonstrations for efficient adaptation of web agents to unseen websites and domains, and demonstrated its efficacy for both proprietary and open-weights MLLM-based agents. More broadly, our results indicate that `AdaptAgent` provides a notable boost in the success rate of current SoTA web agents on unseen websites and domains in a cost-effective way, complementing the gains obtained by building larger pre-trained models or fine-tuning on larger datasets. Despite the SoTA performance established by `AdaptAgent`, even the best-performing agent achieved less than 25% overall task success rate on both Mind2Web and VisualWebArena. There remains significant room for improvement, especially for tasks that require long action sequences and websites with complicated visual layout (see A.5.3 for more details), highlighting the potential for future advancements in this area.

[2] CogAgent-FOMAML outperformed CogAgent-FT (trained on ∼3.4× examples than CogAgent-FOMAML) across all tests except for Mind2Web cross-task, where it outperformed CogAgent-FT (DE) trained with data equivalence. This highlights that with equal amount of training data, our meta-learned CogAgent-FOMAML outperforms the conventionally fine-tuned model as well as demonstrates greater few-shot adaptability to unseen settings.

Acknowledgements

The authors would like to thank Annapoorani Lakshmi Narayanan and Sumitra Ganesh, both with J.P. Morgan AI Research, for valuable discussions and feedback about this work.

Disclaimer

References

[1] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Brenna D Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[4] Amanda Bertsch, Maor Ivgi, Uri Alon, Jonathan Berant, Matthew R Gormley, and Graham Neubig. In-context learning with long-context models: An in-depth exploration. *arXiv preprint arXiv:2405.00200*, 2024.

[5] Cynthia Breazeal and Brian Scassellati. Robots that imitate humans. *Trends in cognitive sciences*, 6(11):481–487, 2002.

[6] Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International conference on machine learning*, pages 783–792. PMLR, 2019.

[7] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

[8] Sylvain Calinon, Florent D'halluin, Eric L Sauser, Darwin G Caldwell, and Aude G Billard. Learning and reproduction of gestures by imitation. *IEEE Robotics & Automation Magazine*, 17(2):44–54, 2010.

[9] Sylvain Calinon, Florent Guenter, and Aude Billard. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2):286–298, 2007.

[10] Letian Chen, Rohan Paleja, and Matthew Gombolay. Learning from suboptimal demonstration via self-supervised reward regression. In *Conference on robot learning*, pages 1262–1277. PMLR, 2021.

[11] Kanzhi Cheng, Qiushi Sun, Yougang Chu, Fangzhi Xu, Yantao Li, Jianbing Zhang, and Zhiyong Wu. Seeclick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935*, 2024.

[12] Neha Das, Sarah Bechtle, Todor Davchev, Dinesh Jayaraman, Akshara Rai, and Franziska Meier. Model-based inverse reinforcement learning from visual demonstrations. In *Conference on Robot Learning*, pages 1930–1942. PMLR, 2021.

[13] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web, 2023.

[14] Peter Englert, Ngo Anh Vien, and Marc Toussaint. Inverse kkt: Learning cost functions of manipulation tasks from demonstrations. *The International Journal of Robotics Research*, 36(13-14):1474–1488, 2017.

[15] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[16] Chelsea Finn, Tianhe Yu, Tianhao Zhang, Pieter Abbeel, and Sergey Levine. One-shot visual imitation learning via meta-learning. In *Conference on robot learning*, pages 357–368. PMLR, 2017.

[17] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[18] Izzeddin Gur, Hiroki Furuta, Austin Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. A real-world webagent with planning, long context understanding, and program synthesis, 2024.

[19] Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models. *arXiv preprint arXiv:2401.13919*, 2024.

[20] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[21] Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and Jie Tang. Cogagent: A visual language model for gui agents, 2023.

[22] Peter C Humphreys, David Raposo, Tobias Pohlen, Gregory Thornton, Rachita Chhaparia, Alistair Muldal, Josh Abramson, Petko Georgiev, Adam Santoro, and Timothy Lillicrap. A data-driven approach for learning to control computers. In *International Conference on Machine Learning*, pages 9466–9482. PMLR, 2022.

[23] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

[24] Taichi Iki and Akiko Aizawa. Do berts learn to use browser user interface? exploring multi-step tasks with unified vision-and-language berts. *arXiv preprint arXiv:2203.07828*, 2022.

[25] Di Jin, Shikib Mehri, Devamanyu Hazarika, Aishwarya Padmakumar, Sungjin Lee, Yang Liu, and Mahdi Namazifar. Data-efficient alignment of large language models with human feedback through natural language, 2023.

[26] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023.

[27] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *arXiv preprint arXiv:2401.13649*, 2024.

[28] Markus Kuderer, Shilpa Gulati, and Wolfram Burgard. Learning driving styles for autonomous vehicles from demonstration. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 2641–2646. IEEE, 2015.

[29] Hanyu Lai, Xiao Liu, Iat Long Iong, Shuntian Yao, Yuxuan Chen, Pengbo Shen, Hao Yu, Hanchen Zhang, Xiaohan Zhang, Yuxiao Dong, et al. Autowebglm: Bootstrap and reinforce a large language model-based web navigating agent. *arXiv preprint arXiv:2404.03648*, 2024.

[30] Evan Zheran Liu, Kelvin Guu, Panupong Pasupat, Tianlin Shi, and Percy Liang. Reinforcement learning on web interfaces using workflow-guided exploration. *arXiv preprint arXiv:1802.08802*, 2018.

[31] Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning, 2024.

[32] Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[33] A Nichol. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[34] Shakked Noy and Whitney Zhang. Experimental evidence on the productivity effects of generative artificial intelligence. *Science*, 381(6654):187–192, 2023.

[35] Oracle. Oracle AI agents help organizations achieve new levels of productivity, Sep 2024. Online press release.

[36] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[37] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1, 1988.

[38] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[39] Harish Ravichandar, Athanasios S Polydoros, Sonia Chernova, and Aude Billard. Recent advances in robot learning from demonstration. *Annual review of control, robotics, and autonomous systems*, 3:297–330, 2020.

[40] Anton Razzhigaev, Matvey Mikhalchuk, Elizaveta Goncharova, Nikolai Gerasimenko, Ivan Oseledets, Denis Dimitrov, and Andrey Kuznetsov. Your transformer is secretly linear. *arXiv preprint arXiv:2405.12250*, 2024.

[41] Allen Ren, Sushant Veer, and Anirudha Majumdar. Generalization guarantees for imitation learning. In *Conference on Robot Learning*, pages 1426–1442. PMLR, 2021.

[42] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[43] Paul E Rybski, Kevin Yoon, Jeremy Stolarz, and Manuela M Veloso. Interactive robot task training through dialog and demonstration. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pages 49–56, 2007.

[44] Stefan Schaal. Learning from demonstration. *Advances in neural information processing systems*, 9, 1996.

[45] Stefan Schaal. Dynamic movement primitives-a framework for motor control in humans and humanoid robotics. In *Adaptive motion of animals and machines*, pages 261–280. Springer, 2006.

[46] Jurgen Schmidhuber. Evolutionary principles in self-referential learning. *On learning how to learn: The meta-meta-... hook.) Diploma thesis, Institut f. Informatik, Tech. Univ. Munich*, 1(2):48, 1987.

[47] Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. *arXiv preprint arXiv:2310.11324*, 2023.

[48] Omar Shaikh, Michelle Lam, Joey Hejna, Yijia Shao, Michael Bernstein, and Diyi Yang. Show, don't tell: Aligning language models with demonstrated feedback. *arXiv preprint arXiv:2406.00888*, 2024.

[49] Peter Shaw, Mandar Joshi, James Cohan, Jonathan Berant, Panupong Pasupat, Hexiang Hu, Urvashi Khandelwal, Kenton Lee, and Kristina Toutanova. From pixels to ui actions: Learning to follow instructions via graphical user interfaces, 2023.

[50] Tianlin Shi, Andrej Karpathy, Linxi Fan, Jonathan Hernandez, and Percy Liang. World of bits: An open-domain platform for web-based agents. In *International Conference on Machine Learning*, pages 3135–3144. PMLR, 2017.

[51] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

[52] Manuela Veloso. Perception, cognition, and action in teams of robots. Colloquium at the Department of Computer Science, Princeton University, September 28 2005.

[53] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023.

[54] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45, 2020.

[55] Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*, 2023.

[56] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[57] Jimuyang Zhang, Ruizhao Zhu, and Eshed Ohn-Bar. Selfd: self-learning large-scale driving policies from the web. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17316–17326, 2022.

[58] Siyan Zhao, John Dang, and Aditya Grover. Group preference optimization: Few-shot alignment of large language models. *arXiv preprint arXiv:2310.11523*, 2023.

[59] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su. Gpt-4v(ision) is a generalist web agent, if grounded. In *Forty-first International Conference on Machine Learning*, 2024.

[60] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

A Appendix

A.1 Detailed Related Work

A.1.1 UI/Web Agents

AI-enabled digital device control [50, 22] — i.e., controlling digital devices using AI with natural language as input — has been a long-standing ambition for large-scale automation of inherently useful tasks. The underlying problem involves mapping a language instruction from the user to a sequence of digital actions that AI agents can execute to successfully complete the task. Before LLMs, approaches to the problem involved using *reinforcement learning* on top of (often pre-trained) language models like LSTM and BERT for processing language input and HTML/DOM along with ResNet-like models for processing GUI states [22, 30, 24]. More recently, as multimodal LLMs have demonstrated success in modeling vision-and-language, they have lent themselves as strong backbones for building web agents that can process tasks specified by the user and engage in reasoning to output the best possible actions to be executed on a user interface such as a web browser. A majority of SoTA work [59, 19] use a pretrained, off-the-shelf LLM such as GPT-4(V/o) to build such multimodal web agents. The input information being provided as context to the LLM can include an image of the GUI, a series of prior actions, additional overlaid image annotations, as well as the HTML/DOM information assuming that the task is web interaction and access to HTML/DOM is possible. Work such as Pix2Act [49] and WebAgent [18] train LLMs to attend to parts of HTML code or generate the next action step through self-supervision, or combine the effectiveness of MLLMs with the promise of reinforcement learning train agents via Behavioral cloning or REINFORCE. However, these works were usually trained on simpler sandboxed environments and would require significant training resources as well as tedious curation of data samples [29]. A disadvantage of such an approach is that it cannot scale to tasks that are complex or that use proprietary enterprise software. Additionally, agents that require exploration as part of the training process would need constant human supervision to avoid risky outcomes. While there has been considerable progress in the success rate of agents on tasks that are encountered as part of their training, their performance in unseen settings has been lacking. To the best of our knowledge, prior work has not explicitly focused on methods that could make Web/GUI agents more adaptable to unseen settings.

Our work proposes a framework where GUI/web agents are trained to efficiently adapt to unseen settings using few-shot human demonstrations. Data-efficient adaptation of web agents via human demonstrations will (a) avoid costly retraining processes/updates for unseen settings, (b) boost the generalizability of web agents to complex workflows and proprietary settngs, and (c) enable web agents to learn from custom information provided by human experts as a part of the demonstrations.

A.1.2 Few-shot learning with LLMs

Data-efficient alignment of LLMs to preferences and new tasks is an active area of research [25, 31]. In contrast to relatively data-hungry approaches like RLHF [36] and DPO [38] that often require hundreds of thousands paired comparisons, few-shot alignment and adaptation aims to use a limited number of examples. While in-context learning [7] is one of the approaches to enable few-shot adaptation of LLMs, it is known to be tedious [26] and is sensitive to variations [47]. Fine-tuning alternatives, like GPO [58] and DITTO [48] have shown promises in few-shot tuning an LLM to align to subjective preferences demonstrated in tasks like email writing and opinion-based question-answering. Most notably, [58] proposes Group Preference Optimization (GPO), which is a meta-learning framework to update LLM parameters based on few-shot in-context demonstrations. However, it is unclear if few-shot alignment approaches like GPO and DITTO, designed for subjective preference tuning, translate to more concrete predictive tasks like ours. Nonetheless, the broader motivation behind methods like GPO – i.e., meta-learning, is a promising opportunity to improve the performance of multimodal web agents, especially cross-website and cross-domains scenarios. Inspired by the promise of meta learning and learning from demonstrations, we adopt model-agnostic meta-learning [15] to train web agents to adapt quickly.

A.1.3 Learning from demonstrations

Learning from Demonstration (LfD) [44, 5, 3, 39] involves teaching agents tasks by observing human or agent demonstrations, enabling them to acquire skills by either directly imitating actions

in supervised learning settings [42] or using demonstrations as guidance in reinforcement learning settings [1]. This approach helps agents master complex tasks that are difficult to explicitly program.

The two main approaches to LfD are Imitation Learning (IL) and Inverse Reinforcement Learning (IRL). Imitation Learning (IL) centers on the direct imitation of demonstrated expert behaviors, where agents replicate observed actions using methods like Behavioral Cloning [37], and DAgger (Dataset Aggregation) [42]. IL typically involves mapping human demonstrations to agent actions through supervised learning. Early techniques such as Dynamic Movement Primitives (DMPs) [45] encoded movement trajectories, while probabilistic models like Gaussian Mixture Models (GMMs) [9] and Hidden Markov Models (HMMs) [8] captured variability and intent in demonstrations. However, IL has limitations when learning from suboptimal demonstrations, as it focuses on mimicking behavior rather than understanding the underlying objectives. Inverse Reinforcement Learning (IRL), in contrast, seeks to uncover the underlying objective of the task by learning a reward function from demonstrations [14, 6, 10, 12]. Instead of merely imitating behavior, IRL infers the goal the demonstrator is optimizing. Once the reward function is learned, Reinforcement Learning (RL) can be used to autonomously derive a policy that achieves the task's goal, allowing the agent to explore and optimize its actions beyond the initial demonstrations [32]. Some notable extensions of IRL include apprenticeship learning [1], maximum entropy IRL [60], and generative adversarial imitation learning (GAIL) [20]. Applications of LfD span robotics, enabling adaptation to various environments and objects [5, 43, 3]; autonomous driving, where vehicles learn navigation and decision-making from human driving data [28, 57]; and game playing, including chess and Go, where agents replicate human gameplay [51].

Agents that automate web tasks share significant similarities with robots that perform real-world tasks, as both rely on core capabilities like perception, reasoning, and execution [52]. This overlap enables the transfer of modeling techniques between the two areas. Drawing on this analogy, our work explores applying learning from human demonstrations to web agents to enhance their adaptability on unseen websites and domains.

A.2 First-order approximation of MAML for Multimodal Web Agents

We present a derivation of the first-order approximation of MAML proposed by [15], while contextualizing it to our setting of updating multimodal LLMs. We begin with the original expression for updates using the MAML algorithm in Equation 1:

$$\theta \leftarrow \theta - \beta \cdot \nabla_{\theta} \sum_{i=1}^{N} \mathcal{L}_{\mathcal{T}_i}(\theta_i). \tag{1}$$

Using the chain rule, the derivative term can be expressed as $\sum_{i=1}^{N}(\nabla_{\theta}\theta_i \times \nabla_{\theta_i}\mathcal{L}_{\mathcal{T}_i}(\theta_i))$. The first component within the summation could be broken down further as,

$$\nabla_{\theta}\theta_i = \nabla_{\theta}(\theta - \alpha \cdot \nabla_{\theta}\mathcal{L}^{train}(\theta)),$$

where $\mathcal{L}^{train}$ denotes the loss on the examples used for training θ_i from task $\mathcal{T}_i$ and α denotes the step-size in the inner loop of meta-training. The above equation further simplifies to

$$\nabla_{\theta}\theta_i = \mathbf{I} - \alpha \cdot \nabla^2_{\theta}\mathcal{L}^{train}(\theta).$$

Now, assuming the second-order derivatives in the expression to zero, provides $\nabla_{\theta}\theta_i = \mathbf{I}$. Plugging that in the original MAML expression gives,

$$\theta \leftarrow \theta - \beta \cdot \sum_{i=1}^{N} \nabla_{\theta_i}\mathcal{L}_{\mathcal{T}_i}(\theta_i).$$

In our context, this essentially means that the inner loop of meta-learning involves fine-tuning the MLLM agent (i.e., $\theta \rightarrow \theta_i$) on web tasks $\mathcal{T}_i$ from a given website. Let's denote this subset of tasks used for the inner loop of training as $\mathcal{D}_i^{train}$. Following this, we update the parameters of the MLLM agent θ by back-propagating the gradients of the loss at θ_i, where the loss is computed on held-out web tasks from the same website — denoted as $\mathcal{D}_i^{test}$. It is worth emphasizing that the gradients being back-propagated are computed at θ_i (as opposed to θ, which would have resulted in training

the MLLM agent on $\mathcal{D}_i^{train}$ and $\mathcal{D}_i^{test}$). In other words, we train the MLLM agent θ on $\mathcal{D}_i^{train}$ to obtain θ_i and then update its *original* parameters θ using penalties computed by evaluating how far θ_i is from the "ideal answers" on held-out $\mathcal{D}_i^{test}$. If exposed to enough updates over varying-but-related websites $i \in \{1, \ldots, N\}$, the updates to the MLLM agent θ would position it such that it would learn to adapt to unseen websites *quickly* in a data-efficient manner.

A.3 Benchmark Details

A.3.1 Mind2Web

Training Set: The training set of the Mind2Web benchmark comprises 1,009 task instances spanning 73 websites from three domains: *travel*, *entertainment*, and *shopping*. These tasks involve various user goals such as booking flights, purchasing tickets, and shopping for products. Each task is accompanied by detailed annotations, including the user instruction, the sequence of actions required to complete the task, and the corresponding HTML and visual states of the web pages.

Test Set: The test set is divided into three subsets to facilitate the evaluation of models in different generalization scenarios:
Cross-Task Subset: This subset contains 174 tasks from the 64 websites that are present in the training set. The tasks are different from those in the training set but occur on familiar websites and within the same domains.
Cross-Website Subset: This subset includes 142 tasks from 10 websites that are entirely unseen during training. The websites belong to the same domains as those in the training set.
Cross-Domain Subset: This subset consists of 694 tasks spanning 53 websites from two new domains: *information* and *service*. These domains are not present in the training set, and the websites are entirely new to the agent.

Fixing overlaps between the train and cross-task evaluation sets of Mind2Web: It is important to note that the standardized cross-task evaluation set of Mind2Web exhibits substantial overlap with the tasks in the training set, which could potentially inflate the evaluation results by testing on tasks that are not truly unseen. For instance, when we computed Jaccard similarity (i.e., intersection-over-union of unique unigrams) between all the tasks in the standardized Mind2Web train set and the cross-task test set, we found pairs of highly similar tasks spread across the two sets. E.g., "add Prometheus movie to watchlist." (train set) and "add The Wire to the watchlist." (cross-task set); "find a cheapest flight from London to New York on 9th May." (train set) and "find cheapest flight from New York to Toronto, Canada on 29 April." (cross-task set). To address this issue, we first combined all the tasks within the existing train and cross-task subsets of the Mind2Web benchmark and computed pair-wise Jaccard similarity between all tasks belonging to the same website. For each website, we then moved K tasks that exhibited least maximum similarity with any other task from the website to construct the amended cross-task evaluation set, while keeping the rest of the tasks from the website in the amended train set. The value of K was determined so as to ensure that the amended train and cross-task sets had the same number of data points as the original train and cross-task sets. We also qualitatively inspected the overlap between the amended train cross-task sets and found that even the most similar tasks (based on unigram Jaccard similarity) across the two sets were now considerable different. For e.g., "show me all the events at any six flags park in Texas" (amended train) and "show me all the artists with smith in their name" (amended cross-task); "add to my cart a women's T-shirt priced under 10 dollars" (amended train) and "list Batman collectible figures priced under 10 dollars and a customer rating above 4 with a same-day delivery option" (amended cross-task). This simple-but-important amendment to the Mind2Web's train and cross-task set ensures minimal overlap between tasks seen during training of the web agents and tasks that they are evaluated on in the cross-task setting.

A.3.2 VisualWebArena

The VisualWebArena benchmark comprises 910 tasks representing 233 unique task templates spread across the three websites. Out of the 910 tasks, 233 tasks (one for each task template) have step-level ground truth available in the form of human trajectories. These trajectories provide detailed action sequences that a human would take to accomplish the task, serving as a reference for evaluating the agent's performance at each step. The remaining tasks do not have step-level ground truth but provide an overall task success signal based on the live environment's state after the agent's interaction.

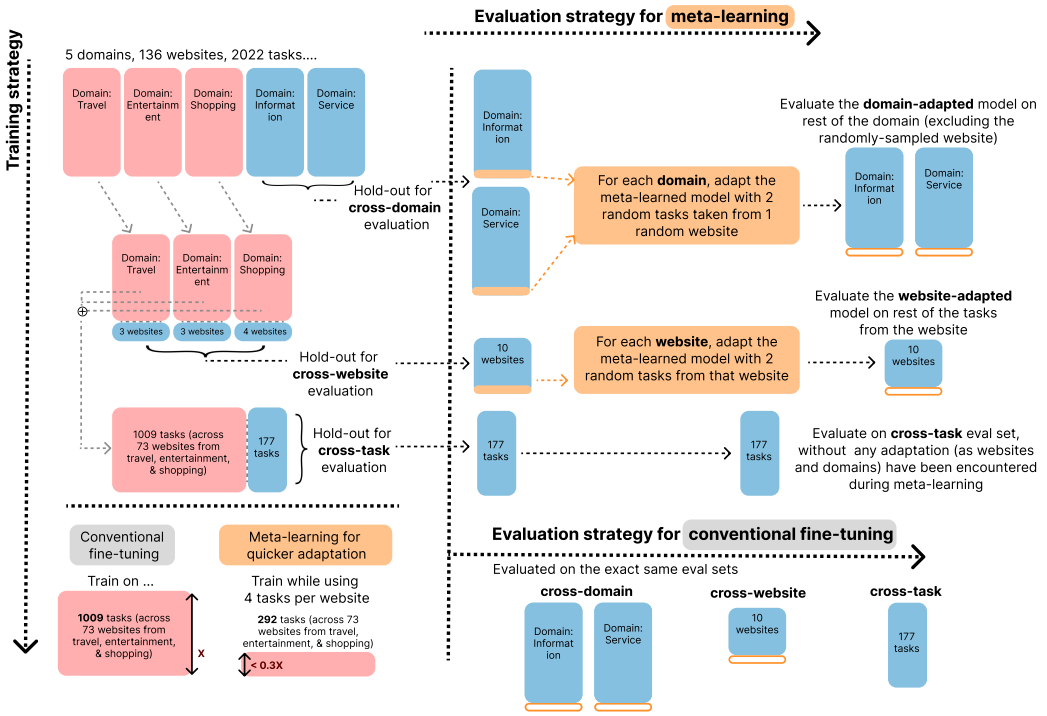

Figure 2: Visual depiction of the protocol used for meta-learning using the Mind2Web train set (left), and the meta-adaptation done on cross-domain and cross-website evaluation sets (top-right). For completeness, we also show the conventional fine-tuning strategy (bottom-right).

A.4 Implementation Details

The specific prompts used for experimenting with SeeAct variants, including the modifications to include (text-only/multimodal) in-context demonstrations are presented in Appendix A.6. We filtered the top-50 HTML elements to be included in the prompt using the methods adopted by Deng et al. [13] and Zeng et al. [59]. For experiments with CogAgent, we use the `THUDM/cogagent-chat-hf` model on HuggingFace [54] as the pre-trained version. For updating the model parameters during fine-tuning, meta-learning, and adaptation, we adopted Low-Rank Adaptation (LoRA) with following hyper-parameters: rank α of 20 and learning rate of 1e-5. For fine-tuning, we trained the model for 2 epochs, with other hyper-parameters set to default/the values used by Hong et al. [21]. For meta-learning, we used a meta-batch size of 1, meaning that we trained the agent to adapt to 1 website during the inner-loop, and used one gradient optimization step for each step of the 2 tasks used for loss computation within the inner-loop. For adaptation to new websites and domains, we use the same strategy to adopt one gradient step optimization per step of the 2 sampled tasks to maintain consistency with the training regime. All the experiments were performed on a virtual server with 8 NVIDIA L4 GPUs (24GiB each).

A.5 Additional Analysis

A.5.1 Multimodal vs. text-only demonstrations

In our ablation study, we examined the impact of in-context demonstration modalities—specifically text-only versus multimodal—on our top-performing models, SeeAct and SeeAct*. See Figure 3 (left) and Table 2.

We observed significant performance enhancements with multimodal in-context demonstrations compared to text-only versions.

• For Mind2Web, SeeAct's overall SR (%) improved to 19.69 (cross-task), 22.46 (cross-website), and 23.97 (cross-domain), up from 15.91, 19.56, and 22.16, respectively. SeeAct* also showed increases to 22.46, 23.10, and 24.06, up from 19.27, 22.15, and 22.87.

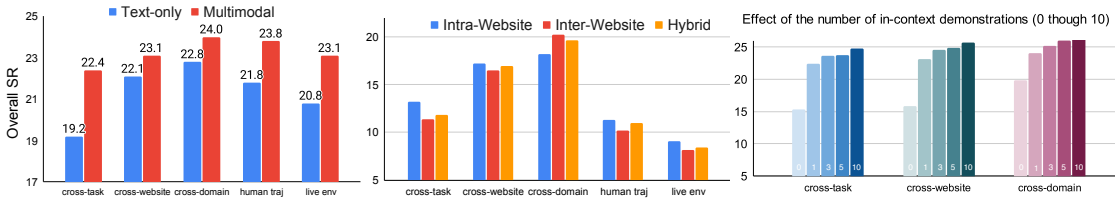

Figure 3: Additional Analysis. **Left:** Ablation study on demonstration modality in SeeAct*. **Middle:** Comparison of overall SR across meta-learning adaptation strategies in CogAgent. **Right:** Variation in performance with different numbers of in-context demonstrations; numbers are inset in the bars.

Type	Model	Cross-Task				Cross-Website				Cross-Domain			
		Ele. Acc.	Op. F1	Step SR	Overall SR	Ele. Acc.	Op. F1	Step SR	Overall SR	Ele. Acc.	Op. F1	Step SR	Overall SR
Baseline	SeeAct (GPT-4o)	62.21	66.56	56.31	14.37	55.25	58.89	49.90	15.83	57.33	60.74	53.72	19.49
Adapted	SeeAct + 1-ICTD	65.71	70.82	58.19	15.91	58.94	62.87	51.11	19.56	59.31	61.69	55.23	22.16
	SeeAct + 1-ICMD	**66.29**	**71.61**	**60.37**	**19.69**	**60.32**	**64.15**	**53.91**	**22.46**	**60.54**	**62.97**	**57.40**	**23.97**
Baseline	SeeAct* (GPT-4o)	63.75	67.68	58.60	15.38	57.02	60.01	50.05	15.89	59.30	62.80	54.82	19.88
Adapted	SeeAct* + 1-ICTD	66.31	70.29	60.24	19.27	59.41	62.48	52.64	22.15	61.01	64.00	56.50	22.87
	SeeAct* + 1-ICMD	**67.77**	**72.52**	**61.88**	**22.46**	**61.67**	**64.76**	**53.98**	**23.10**	**62.44**	**65.41**	**58.33**	**24.06**

(a) Mind2Web dataset

Type	Model	Human Trajectories				Live Environment
		Ele. Acc.	Op. F1	Step SR	Overall SR	Overall SR
Baseline	SeeAct (GPT-4o)	56.03	57.17	52.17	18.75	17.56
Adapted	SeeAct + 1-ICTD	57.16	60.74	53.92	20.56	19.12
	SeeAct + 1-ICMD	**59.15**	**63.18**	**55.27**	**22.42**	**21.36**
Baseline	SeeAct* (GPT-4o)	57.52	59.16	53.16	18.78	18.04
Adapted	SeeAct* + 1-ICTD	58.98	62.93	54.54	21.82	20.87
	SeeAct* + 1-ICMD	**61.46**	**64.12**	**56.72**	**23.86**	**23.15**

(b) VisualWebArena dataset

Table 2: Ablation study on multimodal vs. text-only demonstrations. IC[-]D denotes the type of in-context demonstration, where T and M refer to textual and multimodal demonstrations, respectively. **Bold** text indicates the best performance for each model.

• A similar trend was observed in VisualWebArena, where SeeAct's performance rose to 22.42 (human trajectories) and 21.36 (live environment), up from 20.56 and 19.12, respectively. Similarly, SeeAct* improved from 21.82 to 23.86 (human trajectories) and from 20.87 to 23.15 (live environment).

Overall, the relative gains from text to multimodal ranged from 4.28% to 23.76% (absolute gains of 0.95% to 3.78%) on Mind2Web and from 9.05% to 11.71% (absolute gains of 1.86% to 2.24%) for VisualWebArena. For VisualWebArena, the relative gains were similar between SeeAct and SeeAct*, but Mind2Web saw greater gains with SeeAct compared to SeeAct* across all three evaluation settings, particularly in the cross-website setting. Among all scenarios, the gains were largest in the Mind2Web cross-task setting for both SeeAct (23.76%) and SeeAct* (16.55%), while the gains were smallest with SeeAct* in the cross-website (4.28%) and cross-domain (5.20%) settings. These findings demonstrate the advantage of incorporating richer multimodal in-context demonstrations, including visual snapshots, compared to relying solely on text.

A.5.2 Meta-learning strategies: intra-website, inter-website, and hybrid adaptations

The performance distinctions among the three meta-learning adaptation strategies used with the CogAgent model arise from their specific training and adaptation frameworks (see Figure 3 (middle) and Table 3):

- Intra-Website: Trains on few tasks within a website and adapts to more tasks on the same site
- Inter-Website: Trains on tasks from one website and adapts to tasks from others in the same domain
- Hybrid: Combines both, adapting to tasks within and across websites in the domain

Type	Model	Cross-Task				Cross-Website				Cross-Domain			
		Ele. Acc.	Op. F1	Step SR	Overall SR	Ele. Acc.	Op. F1	Step SR	Overall SR	Ele. Acc.	Op. F1	Step SR	Overall SR
Baseline	CogAgent	30.63	47.67	25.11	02.80	31.50	51.52	21.29	02.11	32.17	49.94	23.32	02.59
	CogAgent-FT	59.46	**63.15**	**54.43**	**13.36**	53.17	57.03	47.14	12.42	61.36	62.79	55.71	15.20
	CogAgent-FT (DE with FOMAML)	55.17	59.87	50.25	10.43	49.46	53.17	44.27	10.10	59.51	59.06	52.20	13.28
Adapted	CogAgent-FOMAML (intra-website)	**60.74**	62.44	53.14	13.24	**60.16**	**63.47**	**55.88**	**17.28**	61.36	62.79	55.71	18.20
	CogAgent-FOMAML (inter-website)	58.77	62.16	53.01	11.46	59.02	62.84	54.13	16.50	**63.88**	**65.01**	**58.42**	**20.22**
	CogAgent-FOMAML (hybrid)	59.34	62.82	53.32	11.89	59.49	62.11	55.38	16.96	62.01	63.13	57.29	19.66

(a) Mind2Web dataset

Type	Model	Human Trajectories				Live Environment
		Ele. Acc.	Op. F1	Step SR	Overall SR	Overall SR
Baseline	CogAgent	25.27	38.64	19.61	01.31	0.46
	CogAgent-FT	52.31	55.64	48.70	08.78	6.43
	CogAgent-FT (DE with FOMAML)	48.62	51.71	44.81	06.81	5.11
Adapted	CogAgent-FOMAML (intra-website)	**57.36**	**60.07**	52.61	**11.36**	**9.17**
	CogAgent-FOMAML (inter-website)	56.11	58.44	**53.81**	10.24	8.29
	CogAgent-FOMAML (hybrid)	57.20	59.14	51.29	11.01	8.47

(b) VisualWebArena dataset

Table 3: Analysis of the three meta-learning adaptation strategies used with the CogAgent model. FT refers to fine-tuning, while DE denotes fine-tuning with data equivalence to the meta-learned models, i.e., using less than one-third of the training data. **Bold** text indicates the best performance in each evaluation setting.

For Mind2Web, the intra-website adaptation strategy showed the best performance in the cross-website setting across all CogAgent versions, achieving an overall SR of 17.28, up from 10.10 in the baseline. For the cross-website evaluation, the model was adapted to two specific tasks per website and then tested on the remaining tasks from that same website. This aligns perfectly with the intra-website meta-training style, explaining its top performance in this setting. The inter-website adaptation strategy performed best in the cross-domain setting, achieving an overall SR of 20.22, up from 13.28. In the cross-domain evaluation, the model was adapted to two tasks from one website and then tested on all tasks from other websites in the same domain. This aligns with the inter-website meta-learning process, resulting in top performance in cross-domain tasks. The hybrid strategy provided the best trade-off across all settings (cross-task, cross-website, and cross-domain), consistently performing between the intra- and inter-website strategies and proving versatile across all evaluation scenarios.

For VisualWebArena, the intra-website adaptation strategy was the top performer in both human trajectories and live environment settings, with overall SRs of 11.36 and 9.17, up from 6.81 and 5.11. For the three websites in the VisualWebArena evaluation set, the model was adapted to two tasks per website and then tested on the remaining tasks. This aligns with the intra-website meta-training approach, resulting in the best performance.

For main results in Table 1, we report the results with the hybrid strategy, where the agent is trained to adapt to tasks within the website as well as tasks on other websites within the same domain.

A.5.3 Results stratified by sequence and visual difficulty levels

Next, we study the variation of overall SR across difficulty levels, stratified based on (1) sequence complexity; and (2) visual difficulty. The three levels of difficulty in both cases and datasets are easy, medium, and hard, following the protocol established in VisualWebArena.

• Sequence difficulty is determined by the length of the ground-truth action sequence (i.e., ≤ 3: easy; $4-9$: medium; ≥ 10: hard).

• To assign visual difficulty labels in Mind2Web based on the required visual processing, we used in-context learning with GPT-4o, utilizing labeled VisualWebArena samples as in-context examples. Snapshots of webpages were evaluated as action sequences and categorized as easy, medium, or hard. Three rounds of annotation were conducted to estimate the self-consistency of GPT annotations, employing chain-of-thought (CoT) reasoning in each round. Finally, human validation was performed to assess the consistency and reasoning of the annotations, with less than 5% of the total examples having their labels changed based on human review.

Table 4 compares the baseline and adapted overall SR of SeeAct* and CogAgent, stratified by difficulty (easy, medium, hard) across sequence complexity and visual difficulty in Mind2Web and VisualWebArena settings. We observe that the improvements in adaptation persist when stratified

Type	Model	Mind2Web			VisualWebArena	
		Cross-Task	**Cross-Website**	**Cross-Domain**	**Human Trajectories**	**Live Environment**
		Overall SR	Overall SR	Overall SR	Overall SR	Overall SR
		Easy | Medium | Hard	Easy | Medium | Hard	Easy | Medium | Hard	Easy | Medium | Hard	Easy | Medium | Hard
Baseline	SeeAct* (GPT-4o)	15.38	15.89	19.88	18.78	18.04
	↪ Sequence complexity	56.7% | 13.7% | 0.0%	57.5% | 14.1% | 0.6%	58.2% | 16.5% | 2.9%	57.5% | 15.2% | 1.7%	56.3% | 14.6% | 0.9%
	↪ Visual difficulty	26.8% | 11.2% | 0.0%	27.1% | 11.7% | 0.9%	31.6% | 13.6% | 2.6%	30.5% | 12.7% | 1.6%	29.4% | 11.7% | 0.8%
Adapted	SeeAct* + 1-ICMD	22.46	23.10	24.06	23.86	23.15
	↪ Sequence complexity	61.3% | 18.8% | 1.7%	62.6% | 19.2% | 2.5%	63.6% | 21.7% | 5.8%	62.6% | 20.4% | 5.2%	61.6% | 19.3% | 5.9%
	↪ Visual difficulty	33.1% | 16.2% | 0.3%	33.8% | 16.6% | 1.4%	36.2% | 18.4% | 4.2%	35.3% | 16.9% | 4.8%	34.8% | 14.9% | 4.1%
Baseline	CogAgent-FT (DE)	10.43	10.10	13.28	06.81	5.11
	↪ Sequence complexity	38.5% | 9.3% | 0.0%	36.5% | 9.0% | 0.4%	39.7% | 11.2% | 2.0%	20.9% | 5.5% | 0.6%	15.9% | 4.1% | 0.3%
	↪ Visual difficulty	18.2% | 7.6% | 0.0%	17.2% | 7.4% | 0.6%	21.5% | 09.3% | 1.8%	11.1% | 4.6% | 0.6%	08.3% | 3.3% | 0.2%
Adapted	CogAgent-FOMAML	11.89	16.96	19.66	11.01	8.47
	↪ Sequence complexity	43.9% | 10.6% | 0.6%	43.8% | 10.8% | 0.7%	50.4% | 14.2% | 2.5%	26.0% | 6.8% | 0.8%	19.3% | 5.0% | 0.4%
	↪ Visual difficulty	20.7% | 08.7% | 0.3%	20.6% | 08.9% | 0.7%	27.3% | 11.8% | 2.3%	13.8% | 5.7% | 0.7%	11.5% | 3.9% | 0.3%

Table 4: Adaptation results stratified by sequence complexity and visual difficulty levels.

by different difficulty levels, with adaptation enhancing performance across all sequence and visual difficulty levels. SeeAct*, with 1-shot multimodal demonstration, performs best across all difficulty levels. Overall SR decreases as difficulty increases across all model variations, aligning with expectations. The adapted SeeAct* performed better overall, particularly on hard tasks (in terms of both visual and sequence difficulty) in the Mind2Web cross-website and cross-domain evaluation settings, as well as in both VisualWebArena evaluation settings. It showed even greater improvement on tasks with high sequence difficulty compared to those with high visual difficulty. For example, in VisualWebArena, for tasks with hard sequence complexity, overall SR increased from 1.7% to 5.2% in human trajectory evaluation and from 0.9% to 5.9% in live environment evaluation. In contrast, the gains with the adapted version of CogAgent were minimal on hard tasks, especially in the VisualWebArena evaluation settings.

A.5.4 Effect of using more human demonstrations

Figure 3 (right) examines the impact of increasing the number of in-context multimodal demonstrations—from 1 to 3, 5, and 10—on 30 sample tasks for SeeAct* across cross-task, cross-website, and cross-domain settings in Mind2Web. Although performance does improve slightly with more demonstrations, the gains are minimal. Given the trade-off between time and incremental accuracy improvements, it is preferable to utilize a single in-context multimodal demonstration.

A.6 1-ICMD Prompt for SeeAct and SeeAct*

In our approach, we extend the prompt design from [59] by adding an in-context multimodal demonstration (ICMD). The prompt provided to the GPT-4o model is as follows:

In-Context Multimodal Demonstration

(... preceded by the SeeAct prompt...)
To begin with, here is a quick example of one of the many tasks you could be performing on the website `<website_name>`.
Example task's description: `<task_description>`
To do this task, you could take the steps shown below.

<Image depicting the GUI snapshot at this stage>
ELEMENT: `<element_name_1>`
ACTION: `<action_type_1>`
VALUE: `<value_if_applicable_1>`

<Image depicting the GUI snapshot at this stage>
ELEMENT: `<element_name_2>`
ACTION: `<action_type_2>`
VALUE: `<value_if_applicable_2>`

...

This marks the end of an example task and its steps. Now, let's move on to the task at hand.
(... followed by the SeeAct prompt...)