TGPO: 基于树指导的偏好优化以增强健壮的网络代理强化学习

Ziyuan Chen[1,2], Zhenghui Zhao[2], Zhangye Han[3,2], Miancan Liu[5,2]
Xianhang Ye[4,2], Yiqing Li[5,2], Hongbo Min[†,2], Jinkui Ren[2], Xiantao Zhang[2], Guitao Cao[*,1]

[1]East China Normal University [2]Alibaba Group
[3]University of Electronic Science and Technology of China [4]Wuhan University [5]Sun Yat-sen University

ABSTRACT

随着大型语言模型和视觉-语言模型的快速发展，将大型模型用作网络代理已成为自动网络交互的关键。然而，使用强化学习训练网络代理面临关键挑战，包括信用分配错误配置、标注成本过高以及奖励稀疏性。为了解决这些问题，我们提出了树结构引导偏好优化（TGPO），一个离线强化学习框架，该框架提出了一种树状轨迹表示方法，通过合并不同轨迹中的语义相同的状态来消除标签冲突。我们的框架包含一个过程奖励模型，它能够自动生成细粒度的奖励，通过子目标进展、冗余检测和动作验证实现。此外，动态加权机制在训练过程中优先考虑高影响力决策点。在线上 Mind2Web 和我们自建的 C-WebShop 数据集上的实验表明，TGPO 显著优于现有方法，在减少冗余步骤的同时实现了更高的成功率。

Index Terms— 强化学习，网络代理，偏好优化

1. 介绍

随着大型语言模型（LLMs）和视觉-语言模型（VLMs）的快速发展，自主 Web 代理 [1, 2, 3] 已经取得了显著进步。Web 代理将自然语言指令转化为一系列网络交互操作（例如点击、输入），通过处理页面上的视觉和文本信息来实现这一目标。这样的代理通常需要理解网页语义，识别可交互元素，并在动态变化的网络环境中做出决策，形成基于视觉和/或文本状态输入的顺序决策过程。

网络交互环境具有庞大的动作空间和长期依赖性，使得传统的监督学习方法 [4, 5] 对于训练网络代理无效。诸如监督微调（SFT）之类的方法依赖于大规模高质量的标注数据，然而获取精确的动作标签成本高昂且难以扩展。相比之下，强化学习（RL）使代理能够通过自主探索发现有效的策略 [6, 7, 5, 8]，利用负样本作为信息丰富的训练信号，并通过建模抽象策略和长期奖励展现出强大的泛化能力。

从数据收集的角度来看，针对网络代理的强化学习（RL）[9] 可以分为在线和离线范式。在线 RL[10, 11] 通过实时与网站互动来收集新的轨迹，这导致了较高的采样成本和低效率。实际网站还增加了更多困难，例如严格的请求率限制、登录要求和其他限制，使得长期训练难以持续。因此，诸如 WebAgent-R1[12]、WebRL[13] 和 GiGPO[14] 等方法通常是在模拟或自定义网络环境中 [15, 16] 而不是在实时网站上进行训练的。相比之下，诸如 DPO [17] 和 KTO [18] 这样的离线强化学习方法从预先收集的轨迹中学习，不进行实时交互，从而避免了这些约束并充分利用现有数据。

然而，将离线 RL 应用于网络代理仍然面临三个主要挑战：1) *信用分配误配*— 轨迹级别的成功/失败标签被统一应用于所有状态-动作对，通过惩罚失败轨迹中的正确动作和奖励成功轨迹中的无效动作来引入噪声；2) *禁止性标注成本*— 尽管步级别标注可以减轻信用分配错误，但它需要大量的手动努力，通常超过轨迹级别注释成本的十倍，这使得大规模训练不切实际；3) *奖励稀疏性*— 缺乏细粒度的奖励信号导致代理学习次优策略，常常表现出冗余动作或循环，从而降

Work done during an internship at Alibaba Group.
∗ Corresponding author: gtcao@sei.ecnu.edu.cn.
† Project Leader.

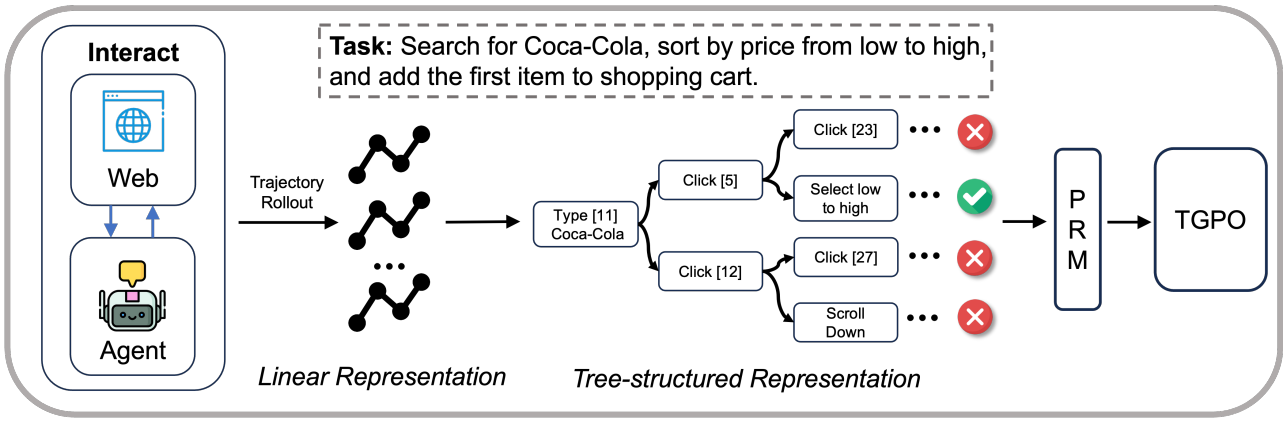

Fig. 1: 拟议的树引导偏好优化框架概览，用于 Web 代理。轨迹被聚合到状态树中，以实现细粒度的信用分配和自动化奖励生成。

低执行效率和任务成功率。

为了解决这些挑战，我们提出了基于树指导的偏好优化（Tree-Guided Preference Optimization, TGPO）用于网络代理强化学习。我们的关键见解是，将来自多个轨迹的语义相同的州聚合到一个统一的树结构中，提供了一种消除标签歧义并自动化奖励信号生成的原则性方法。

在 Online-Mind2Web [19, 20] 和 C-WebShop 基准测试中，使用 Qwen3-14B [21] 和 Qwen2.5-VL-72B [22] 模型评估时，TGPO 一直以更高的成功率和更少的冗余操作优于现有方法。总结来说，我们的工作对领域做出了以下贡献：

- 我们提出了树引导的偏好优化方法，该方法将树结构的轨迹表示与自动化过程奖励建模相结合，用于 Web 代理训练。
- 我们提出了一种通过无偏行为评估、自动化步骤级奖励生成以及战略性地关注关键决策点来解决 Web 代理训练中三个基本挑战的解决方案。
- 综合实验显示 TGPO 在成功率和执行效率方面均优于其他强化学习方法。

2. 方法论

在本节中，我们介绍了包含三个组件的 TGPO：一种通过状态合并解决标签冲突的树状轨迹表示、提供细粒度奖励以减少手动标注的过程奖励模型（PRM），以及基于奖励差异优先处理关键决策点的动态加权机制。

2.1. 树结构轨迹表示法

给定用户指令 $\mathcal{I}$，Web 代理需要通过网络环境中的状态-动作交互序列来完成相应的任务。我们将此过程形式化为一个马尔可夫决策过程 (MDP)$\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R \rangle$，其中 $\mathcal{S}$ 表示由网页界面状态（通过截图或 DOM 树编码）组成的状态空间，$\mathcal{A}$ 表示包含交互操作（如点击、输入、滚动）的动作空间，$P(s'|s,a)$ 定义了状态转移函数，并且 $R(s,a)$ 为状态-动作对提供即时反馈。轨迹 $\tau = (s_0, a_0, s_1, \ldots, s_T)$ 表示由代理生成的一个完整的执行序列，其中 $s_t \in \mathcal{S}$ 表示第 t 步的状态，$a_t \in \mathcal{A}$ 表示采取的动作。

执行任务 K 次时，我们获得 K 条轨迹 $\mathcal{T} = \{\tau^{(1)}, \ldots, \tau^{(K)}\}$，每条轨迹带有轨迹级标签 $y^{(k)} \in \{0, 1\}$ 表示成功或失败。直接将这些标签应用于各个步骤会导致跨轨迹标签冲突。如图 2 所示，考虑两条具有相同中间状态但结果不同的轨迹：在轨迹 $\tau^{(1)}$（最终失败）中，动作点击排序按钮是正确序列的一部分，但由于最终失败（$y^{(1)} = 0$），被错误地标记为不正确的；在轨迹 $\tau^{(2)}$（最终成功）中，相同的操作被正确地标记为正确（$y^{(2)} = 1$）。这为相同的态-动作对创建了矛盾的评估。同一动作在一个轨迹中显得正确而在另一个轨迹中显得错误，尽管发生在相同的状态下。轨迹级别的标签无法区分一个动作本身是否有缺陷，还是失败是由于后续决策造成的。

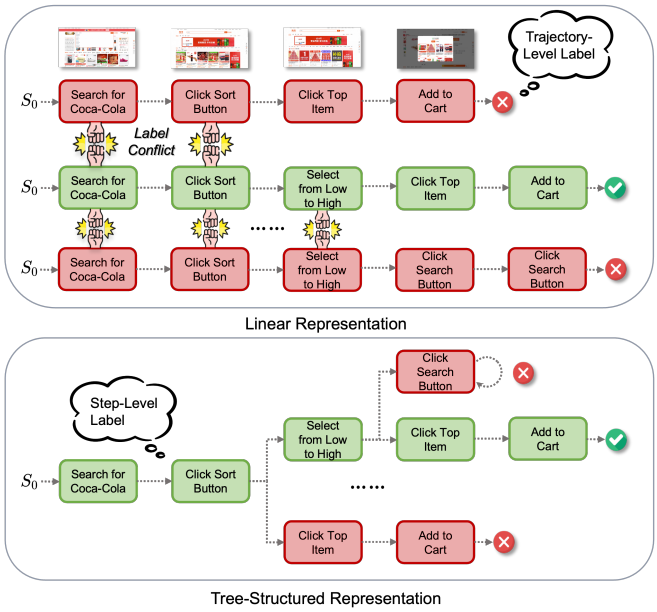

Fig. 2: 线性与树结构表示。绿色/红色：正确/错误操作。✓/✗：成功/失败。任务：搜索可口可乐，按价格排序，将第一个项目添加到购物车。

为了解决这个问题，我们提出了一种树状结构表示方法，该方法将多个轨迹中语义上相同的状态合并。从初始状态 s_0 开始，我们收集 K 轨迹 $\tau^{(k)} = (s_0^{(k)}, a_0^{(k)}, s_1^{(k)}, a_1^{(k)}, \ldots, s_T^{(k)})$ 并构建一棵树 $G = (V, E)$，其中 V 包含唯一的状态节点，而 E 代表动作转换。

两个状态 s_i 和 s_j 在以下情况下合并：(1) 标准化的 URL 匹配（保留必需的参数）；以及 (2) 要么 (a) URL 更改后的有效操作序列一致，要么 (b) 图像哈希相同。这确保语义等价的状态被聚合同时保持动作差异。

基于此，我们构建了如图 2 所示的轨迹树状表示。这种树形结构提供了几个关键优势：（1）自动步级标签，减少注释成本；（2）识别并移除冗余动作；（3）通过回溯实现精确的中间奖励分配；（4）将校正行为自然地表示为循环。

2.2. 过程奖励模型

为了克服基于轨迹学习中稀疏奖励和手动标注成本高的挑战，我们提出了一种细粒度、可自动验证的过程奖励模型，该模型基于树状结构表示。受 [23] 相关工作的启发，此机制包括四个奖励维度：

子目标奖励（R_{subgoal}）：衡量接近目标完成的进度，公式为

$$R_{\text{subgoal}} = \frac{L_{\min}}{d(s_0 \to s_t) + \min_q d(s_t \to s_{\text{goal}})}, \quad (1)$$

其中，$L_{\min}$ 表示从初始状态到目标状态所需的理论最小步数；$d(s_0 \to s_t)$ 表示达到当前状态的实际累计步数，而 $\min_q d(s_t \to s_{\text{goal}})$ 表示从当前状态到目标状态的最简估计步数（通过树形结构计算得出）。这种奖励鼓励智能体选择完成任务的最短路径。

冗余惩罚（R_{red}）：$R_{\text{red}} = -1.0$ 如果检测到状态循环，否则 0。通过树结构分析自动识别并惩罚重复动作。

准确度奖励（R_{acc}）:$R_{\text{acc}} = +1.0$ 如果操作有效，否则 0。使用 VLM 验证预期的接口修改。

格式奖励（R_{format}）：R_{format} +1.0 如果操作格式有效，否则 0。验证是否符合操作执行引擎的语法要求。

因此，总奖励可以表示为：

$$R = R_{\text{acc}} + R_{\text{format}} + R_{\text{red}} + \alpha \cdot R_{\text{subgoal}}. \quad (2)$$

其中，R_{subgoal} 和 R_{red} 是基于树结构计算的，而 R_{acc} 和 R_{format} 则是自动验证的。经验上，α 设置在 2-5 之间以适当强调子目标进展的重要性。

2.3. TGPO: 树引导偏好优化

与 KTO 训练（存在标签冲突）和 DPO 训练（需要从轨迹级数据构造偏好对，这本质上具有挑战性）不同，我们利用树结构的特性来构建高质量的偏好对。

具体地说，在树中的每个状态节点 s，多个动作分支导向不同的路径。通过比较它们的累计奖励，我们自动生成排序偏好对 (a_w, a_l)，其中 a_w 是高奖励动作（被选择）而 a_l 是低奖励动作（被拒绝）。

此外，标准的 DPO 训练在处理 Web 代理任务时存在局限性：它将所有偏好对视为同等重要，未能区分不同决策点的重要性。基于树状结构表示，我们观察到不同状态节点下的动作分支具有不同的价值差异，这反映了决策点的重要程度。为解决这一问题，我们提出了 TGPO 训练方法，该方法引入了一种基于奖励

差异的动态加权机制，使模型能够专注于最关键且方差最大的决策点。

基于偏好对和细粒度奖励信号，我们提出了树引导的偏好优化（TGPO）。我们首先定义优选动作 a_w 和较不优选动作 a_l 之间的对数率间隔如下：

$$\Delta = \log \frac{\pi_\theta(a_w \mid s)}{\pi_{\text{ref}}(a_w \mid s)} - \log \frac{\pi_\theta(a_l \mid s)}{\pi_{\text{ref}}(a_l \mid s)}. \quad (3)$$

然后，TGPO 损失函数被定义为：

$$L_{\text{TGPO}} = -\frac{|r_w - r_l|}{\sigma(R_s)} \cdot \log\left(\frac{1}{1+\exp(-\beta\Delta)}\right), \quad (4)$$

其中，r_w 和 r_l 分别表示选定和拒绝的动作的累积奖励，而 $\sigma(R_s)$ 是状态 s 下所有动作奖励的标准差。权重 $w = \frac{|r_w - r_l|}{\sigma(R_s)}$ 标准化奖励差异，确保不同状态节点之间的权重可比性。

3. 实验

3.1. 环境和基线

环境。我们在两个网络代理框架上进行评估，分别是 SeeAct [24] 和 Browser-use [25]，使用了包含来自 136 个网站的 300 项任务的数据集 Online-Mind2Web [19, 20] 以及我们自行构建的 C-WebShop 数据集，该数据集包含了 50 个来自淘宝的中文电子商务任务。

基线。我们采用开源模型 Qwen2.5-VL-72B [22] 和 Qwen3-14B [21] 作为基线。利用其卓越的多模态能力，Qwen2.5-VL-72B 在 SeeAct 上实现，而 Qwen3-14B 则在 Browser-use 上实现。我们比较了 SFT、KTO [18]、KTO-Tree、DPO [17] 和我们的 TGPO。具体来说，KTO-Tree、DPO 和 TGPO 借助于从树结构中导出的步骤级注释。DPO 和 TGPO 都基于最初使用 SFT 训练的模型构建。所有 RL 方法都在 8 个 H20 GPU 上以学习率 1×10^{-5} 训练了 2 个 epoch。对于 KTO 和 KTO-树，可取的和不可取权重由正负样本比例设定。

3.2. 主要结果

表 1 展示了不同训练方法在 Online-Mind2Web 和 C-WebShop 数据集上的性能比较。在 Online-Mind2Web 基准测试中，我们的 TGPO 达到了最高的成功率 38.4% 和最短的平均轨迹长度 10.71 步，优于现有的方法和闭源模型 GPT-4o。值得注意的是，TGPO 相比标准 KTO 训练将成功率提高了 11.3%，并将冗余步骤从 2.90 减少到 2.52，这证明了我们树结构表示在解决标签冲突和优化动作序列方面的有效性。

Table 1: TGPO 与其他方法的性能比较。

方法	成功率（%）	执行效率	
		平均步骤	红。步骤
在线-Mind2Web 数据集 +浏览器使用			
Qwen3-14B [21]	26.8	12.79	3.31
+ SFT	31.8	11.73	2.76
+ KTO [18]	27.1	11.94	2.90
+ KTO-Tree	34.4	10.98	2.42
+ DPO [17]	34.0	11.53	2.88
+ TGPO	38.4	10.71	2.52
GPT-4o [26]	30.0	–	–
C-网络商店数据集 + 见行动			
Qwen2.5-vl-72B-Instruct [22]	36.7	14.28	4.95
+ SFT	70.3	9.73	1.26
+ KTO [18]	72.9	9.42	1.40
+ KTO-Tree	77.6	8.97	1.08
+ DPO [17]	72.1	9.85	1.41
+ TGPO	78.6	8.66	0.97

Note: Avg. Steps represents average trajectory length; Red. Steps counts redundant actions.

树引导的变体 KTO-Tree 也在原始的 KTO 方法上显示出显著改进，实现了 34.4%的成功率（对比 27.1%），并将平均步骤从 11.94 减少到 10.98，突显了状态合并对无偏行动评估的价值。

在 C-WebShop 数据集上，TGPO 保持其优势，成功率为 78.6%（对比 DPO 的 72.1%和 KTO-Tree 的 77.6%），同时将平均步骤数从 14.28 减少到 8.66，并几乎消除了冗余动作（0.97 对基础模型中的 4.95）。这些结果验证了我们的框架在各种网络交互场景中的鲁

棒性。

3.3. 消融研究

3.3.1. 树结构的有效性。

表 2 显示两个数据集均包含大量标签冲突。如表 1 所示，消除这些冲突后，KTO-Tree 通过提高成功率和减少冗余步骤而优于 KTO：在 Online-Mind2Web 上，成功率达到 7.3%的提升，并且冗余步骤减少了 16.6%，而在 C-WebShop 上，则提升了 4.7%的成功率并且将冗余步骤减少了 22.9%。

Table 2: 航迹中标签冲突的百分比。

数据集	标签冲突百分比
Online-Mind2Web [19, 20]	38.71%
C-WebShop	26.95%

3.3.2. 细粒度奖励有效性。

TGPO 在 C-WebShop 上的成功率比 DPO 高出 6.5%，将执行步骤从 9.85 减少到 8.66，冗余步骤从 1.41 减少到 0.97。动态加权机制专注于高方差决策点的训练，使执行路径更加高效，并减少了冗余操作。

4. 结论

在这项工作中，我们提出了 TGPO 方法，旨在克服网络代理训练中的关键挑战：信用分配错误分配、高标注成本和奖励稀疏性。TGPO 通过树形结构表示解决标签冲突问题，利用 PRM 生成细粒度的奖励，并通过自适应加权优先处理关键决策，在 Online-Mind2Web 和 C-WebShop 基准测试中实现了更高的成功率并减少了冗余操作。该方法也可扩展应用于其他领域，如 GUI 交互和游戏环境。

5. REFERENCES

[1] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar, "Voyager: An open-ended embodied agent with large language models," arXiv preprint arXiv:2305.16291, 2023.

[2] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al., "Agentbench: Evaluating llms as agents," arXiv preprint arXiv:2308.03688, 2023.

[3] Liangbo Ning, Ziran Liang, Zhuohang Jiang, Haohao Qu, Yujuan Ding, Wenqi Fan, Xiao-yong Wei, Shanru Lin, Hui Liu, Philip S Yu, et al., "A survey of webagents: Towards next-generation ai agents for web automation with large foundation models," in Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2, 2025, pp. 6140–6150.

[4] Hanyu Lai, Xiao Liu, Iat Long Iong, Shuntian Yao, Yuxuan Chen, Pengbo Shen, Hao Yu, Hanchen Zhang, Xiaohan Zhang, Yuxiao Dong, et al., "Autowebglm: A large language model-based web navigating agent," in Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 5295–5306.

[5] Jiayi Pan, Yichi Zhang, Nicholas Tomlin, Yifei Zhou, Sergey Levine, and Alane Suhr, "Autonomous evaluation and refinement of digital agents," arXiv preprint arXiv:2404.06474, 2024.

[6] Thomas Carta, Clément Romac, Thomas Wolf, Sylvain Lamprier, Olivier Sigaud, and Pierre-Yves Oudeyer, "Grounding large language models in interactive environments with online reinforcement learning," in International Conference on Machine Learning. PMLR, 2023, pp. 3676–3713.

[7] Hao Bai, Yifei Zhou, Jiayi Pan, Mert Cemri, Alane Suhr, Sergey Levine, and Aviral Kumar, "Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning," Advances in Neural Information Processing Systems, vol. 37, pp. 12461–12495, 2024.

[8] Simon Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Peter Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, et al., "Fine-tuning large vision-language models as decision-making agents via reinforcement learning," Advances in neural information processing systems, vol. 37, pp. 110935–110971, 2024.

[9] Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Yue Liao, Hongru Wang, Mengyue Yang, Heng Ji, Michael Littman, Jun Wang, Shuicheng Yan, Philip Torr, and Lei Bai, "The landscape of agentic reinforcement learning for llms: A survey," 2025.

[10] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.

[11] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al., "Deepseekmath: Pushing the limits of mathematical reasoning in open language models," arXiv preprint arXiv:2402.03300, 2024.

[12] Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, Hyokun Yun, and Lihong Li, "Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning," 2025.

[13] Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Xinyue Yang, Jiadai Sun, Yu Yang, Shuntian Yao, Tianjie Zhang, et al., "Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning," arXiv preprint arXiv:2411.02337, 2024.

[14] Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An, "Group-in-group policy optimization for llm agent training," arXiv preprint arXiv:2505.10978, 2025.

[15] Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al., "Webarena: A realistic web environment for building autonomous agents," arXiv preprint arXiv:2307.13854, 2023.

[16] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan, "Webshop: Towards scalable real-world web interaction with grounded language agents," Advances in Neural Information Processing Systems, vol. 35, pp. 20744–20757, 2022.

[17] 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, vol. 36, pp. 53728–53741, 2023.

[18] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela, "Kto: Model alignment as prospect theoretic optimization," arXiv preprint arXiv:2402.01306, 2024.

[19] Tianci Xue, Weijian Qi, Tianneng Shi, Chan Hee Song, Boyu Gou, Dawn Song, Huan Sun, and Yu Su, "An illusion of progress? assessing the current state of web agents," 2025.

[20] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su, "Mind2web: Towards a generalist agent for the web," in Advances in Neural Information Processing Systems, 2023, vol. 36, pp. 28091–28114.

[21] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al., "Qwen3 technical report," arXiv preprint arXiv:2505.09388, 2025.

[22] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al., "Qwen2. 5-vl technical report," arXiv preprint arXiv:2502.13923, 2025.

[23] Chenyu Yang, Shiqian Su, Shi Liu, Xuan Dong, Yue Yu, Weijie Su, Xuehui Wang, Zhaoyang Liu, Jinguo Zhu, Hao Li, et al., "Zerogui: Automating online gui learning at zero human cost," arXiv preprint arXiv:2505.23762, 2025.

[24] Boyuan Zheng, Boyu Gou, Jihyung Kil, Huan Sun, and Yu Su, "Gpt-4v (ision) is a generalist web agent, if grounded," arXiv preprint arXiv:2401.01614, 2024.

[25] Magnus Müller and Gregor Žunič, "Browser use: Enable ai to control your browser," 2024.

[26] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al., "Gpt-4o system card," arXiv preprint arXiv:2410.21276, 2024.