WEPO: Web Element Preference Optimization for LLM-based Web Navigation

Jiarun Liu, Jia Hao, Chunhong Zhang, Zheng Hu*

State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications
{liujiarun01, zhangch, huzheng}@bupt.edu.cn

Abstract

The rapid advancement of autonomous web navigation has significantly benefited from grounding pretrained Large Language Models (LLMs) as agents. However, current research has yet to fully leverage the redundancy of HTML elements for contrastive training. This paper introduces a novel approach to LLM-based web navigation tasks, called Web Element Preference Optimization (WEPO). WEPO utilizes unsupervised preference learning by sampling distance-based non-salient web elements as negative samples, optimizing maximum likelihood objective within Direct Preference Optimization (DPO). We evaluate WEPO on the Mind2Web benchmark and empirically demonstrate that WEPO aligns user high-level intent with output actions more effectively. The results show that our method achieved the state-of-the-art, with an improvement of 13.8% over WebAgent and 5.3% over the visual language model CogAgent baseline. Our findings underscore the potential of preference optimization to enhance web navigation and other web page based tasks, suggesting a promising direction for future research.

Introduction

The field of autonomous web navigation has seen significant advancements, driven by the capabilities of Large Language Models (LLMs) in both mobile and webpage interactions (Wang et al. 2024b; Mialon et al. 2023; Xi et al. 2023). Preliminary attempts, such as the ChatGPT Plugin (OpenAI 2023), have also started building practical applications of web knowledge-based chatbot.

Web navigation can be described as processes where agents perform specific tasks on behalf of human users within a web environment, involving the interpretation of high-level user instructions, decomposing them into basic operations, and interacting with complex web pages dynamically. To achieve this, agents must understand intricate web scenarios, adapt to dynamic changes such as noisy text and evolving HTML structures, and generalize successful operations to unseen tasks, thus freeing humans from repetitive interactions with computer interfaces.

Traditional web agents trained through reinforcement learning (Shi et al. 2017; Yao et al. 2022) often mimic human behavior using predefined actions like typing, searching, and navigating to a specific page. However, they frequently struggle with the complexities of real-world web environments and the challenges of designing effective reward functions. Recent research has leveraged the HTML understanding, logical reasoning, and code generation capabilities of LLMs, enabling agents to comprehend long HTML documents and predict the next action steps. Notable examples include Mind2Web (Deng et al. 2024), which provides an realistic interaction dataset and fine-tunes multiple LLMs to summarize verbose HTML and iteratively optimize and execute actions. Other works such as WebGum (Furuta et al. 2023) and CogAgent (Hong et al. 2023) construct multimodal architectures, enhancing agents with visual perception abilities through supervised learning with a multimodal corpus that includes HTML screenshots. These prior works are thoroughly summarized in our related work section.

In parallel, preference learning in fine-tuning LLMs has gained prominence, particularly since the introduction of Reinforcement Learning with Human Feedback (RLHF) in GPT-3 (Ziegler et al. 2019; Ouyang et al. 2022), which aligns model outputs with human preferences through reward modeling and reinforcement learning with KL divergence constraints. More subsequent works, such as Direct Preference Optimization (DPO) (Amini, Vieira, and Cotterell 2024) and its variants (Hong, Lee, and Thorne 2024; Morimura et al. 2024), have reparameterized the reward function and optimized training efficiency. These advancements have primarily focused on mainstream tasks in natural language processing, such as dialogue generation and code generation. To our knowledge, the proven effectiveness of preference optimization algorithms like DPO has not been applied to web task automation.

Moreover, existing autonomous web navigation research has not fully exploited the potential of contrastive learning using non-salient HTML elements. After observing the structural complexity and crowded element arrangement in HTML on both Mind2Web and real web environment, we realized that these environments are naturally conducive to data-augmented preference learning. We hypothesize that incorporating preference learning can significantly enhance LLM-based agents' capabilities in web navigation tasks.

Motivated by this, we introduce Web Element Preference Optimization (WEPO), a novel framework that inte-

*Corresponding author of this paper.

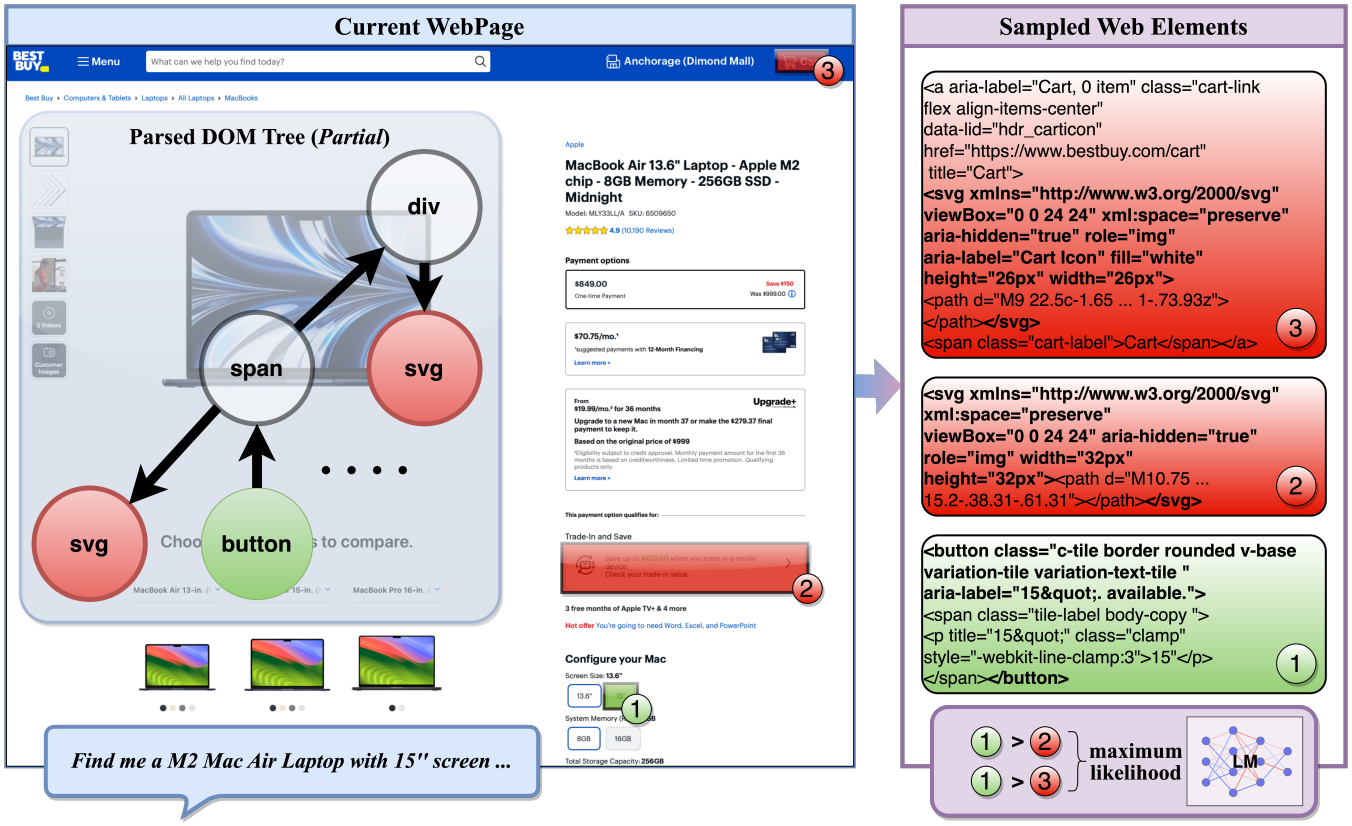

Figure 1: Illustration of Web Element Preference Optimization (WEPO). Given user intent, *Find me a M2 Mac Air Laptop with 15" screen*, WEPO combines the correct element (marked in green) with heuristic rule-based sampled negative elements (marked in red) to construct preference pairs. This process utilizes the maximum likelihood objective function proposed in algorithms such as DPO to fine-tune the language model, thereby enhancing its accuracy in element discrimination and selection.

grates preference optimization algorithms into mainstream LLM-based web navigation tasks. By sampling non-relevant web elements as negative samples, we implement preference learning that requires no human effort, thereby utilizing redundant information in the web environment and achieving high sample efficiency. Specifically, we design a heuristic distance-based element sampling method tailored to the DOM tree structure to enhance the efficiency of contrastive learning. WEPO then maximizes the likelihood of operations on preferred elements and minimizes it for dis-preferred elements, aligning user high-level intent with agent operation sequences. We illustrate WEPO in Figure 1, provide detailed implementation steps and the theoretical foundation of WEPO in the subsequent sections.

For experiments, we selected the Mind2Web dataset due to its high task diversity and realistic web scenarios, which best validate the capabilities of fine-tuned LLM agents. Our experiments on multiple mainstream open-sourced models demonstrate that our WEPO significantly outperforms traditional supervised fine-tuning (SFT) methods, exceeding the MindAct (Deng et al. 2024) baseline by 20.0% and WebAgent (Gur et al. 2023) by 13.8%. WEPO also surpasses visual language model (VLM) CogAgent (Hong et al. 2023) by 5.3% with smaller model parameters and faster inference time, achieving state-of-the-art performance.

We believe WEPO represents a significant advancement in autonomous web navigation, leveraging HTML structure based preference optimization to enhance task performance and suggesting promising directions for future research in LLM-based web navigation and related applications.

Related Work

Web Navigation with LLMs. In the area of web navigation and web-based task automation, the integration of LLM-based agents has shown considerable promise. Kim, Baldi, and McAleer (2024) and Sridhar et al. (2023) introduces the use of prompting schemes combined with criticism or hierarchical modularized design, Li et al. (2023) exploits zero-shot prompt learning via self-reflection and structured thought management, and Zheng, Wang, and An (2023) applies structural prompting with exemplar retrieval to achieve few-shot in-context learning.

Gur et al. (2023) and Gür et al. (2023) demonstrated the effectiveness of encoder-decoder architectures such as HTML-T5, tailored to the HTML tree structure through sophisticated local and global attention mechanisms and a mixture of denoising objectives. Deng et al. (2024) introduced MindAct framework, which simplifies web interac-

tion by transforming generation tasks into multiple-choice formats through instruction fine-tuning, thereby gaining decent performance on Mind2Web benchmark. In addition, Nakano et al. (2021) developed WebGPT, which leverages reinforcement learning with human feedback (RLHF) (Ouyang et al. 2022; Ziegler et al. 2019) to align decision-making processes with human-preferred answers. Yao et al. (2022) discusses the integration of reinforcement learning (RL) for scalable real-world web shopping, highlighting the design of heuristic rewards that enhance learning efficiency. CC-Net (Humphreys et al. 2022), despite not using large language models, utilizes a hybrid architecture that integrates pixel-based inputs via ResNet (He et al. 2016) blocks and language embeddings through transformer blocks (Vaswani et al. 2017), demonstrating exceptional performance through its effective combination of RL and imitation learning.Attempts to address web navigation using multimodal large language models (MLLMs) are also emerging at scale (Hong et al. 2023; You et al. 2024; Wang et al. 2024a; Niu et al. 2024; Baechler et al. 2024; Cheng et al. 2024; Furuta et al. 2023). Among them, CogAgent once reached the state-of-the-art on Mind2Web benchmark by using a high-resolution cross-modular image encoder in conjunction with visual language model (VLM).

Benchmarks in web navigation have evolved rapidly from the simplified MiniWoB (Shi et al. 2017) to the advanced Mind2Web (Deng et al. 2024) and other alternatives (Lù, Kasner, and Reddy 2024; Zhou et al. 2023; He et al. 2024). Mind2Web tackles real-world complexities by incorporating 137 real-world websites into a wide range of 2350 multi-step tasks across 31 domains.

Preference Learning with LLMs. Fine-tuning large language models with preference objective has evolved significantly with Direct Preference Optimization (DPO) (Rafailov et al. 2024) and its variants (Hong, Lee, and Thorne 2024; Meng, Xia, and Chen 2024; Morimura et al. 2024; Amini, Vieira, and Cotterell 2024) improving on traditional RLHF approaches (Christiano et al. 2017; Ziegler et al. 2019; Stiennon et al. 2020; Ouyang et al. 2022). Recent advances focus on improving dataset quality (Morimura et al. 2024), introducing marginal distinctions to better control bias (Duan et al. 2024), optimizing the preference labeling process and enhancing sample efficiency by minimizing human oracle involvement (Bai et al. 2022) and acquire comparison pairs actively (Muldrew et al. 2024).

Preference Learning in Web Scenarios. The application of preference learning to web-based tasks is not a new concept. Notable early work by Radlinski and Joachims (2005) leveraged query chains and implicit user feedback, such as click-through data, to refine search engine algorithms. This approach aimed to capture subtle user preferences that were not explicitly stated but could be inferred from their search behavior sequences. Xiang et al. (2010) and Zhu et al. (2021) also explored preference modeling and ranking for web retrieval applications, including data augmentation of user interaction sequences for comparative learning. These works provide a solid foundation for preference in web scenarios, although they predate the era of large language models and did not attempt to address web navigation issues directly. As mentioned above, Nakano et al. (2021) revolves around fine-tuning GPT-3 (Brown et al. 2020) to answer long-form questions in a text-based web environment. The training of WebGPT involves behavior cloning followed by rejection sampling against a reward model trained to predict human preferences, which is considered as an adaptation of preference learning to web QA tasks.

Web Element Preference Optimization

Task Formulation

We formulate the web navigation task as a partially observable Markov decision process (POMDP) (S, A, T, R, I, O) according to Yao et al. (2022), with state space S, action space A, deterministic transition function $T : S \times A \to S$, reward function $R : S \times A \to [0, 1]$, intent space I and a state observation space $S \times I \to O$. A state $s \in S$ represents a webpage, an action $a \in A(s)$ corresponds to an operation on a webpage, a high-level natural language intent $i \in I$ represents a complex web interaction, usually involving implicit multi-step sub-instructions. Consistent with mainstream efforts (Deng et al. 2024; Gur et al. 2023), we discard the use of a reward function $r = R(s, a)$ and do not consider employing reinforcement learning in this work. Within interaction loop of the web environment, numerous web elements e_k exist, yielding candidate set $E = \{e_1, e_2, ..., e_m\}$. The target element is denoted as $\hat{e}$ with a ground truth operation $\hat{o}$, which are both labeled through human supervision, thereby determining $\hat{a} = a(\hat{e}, \hat{o})$. In the Mind2Web (Deng et al. 2024) setting, a state s includes snapshots in multiple formats, such as HTML code and trace files. We exclusively utilize the HTML scripts of the webpage for WEPO learning. For each state, E comprises all web elements that collectively form the current page. An action a encompasses clicking on an interactive element `(CLICK,element_ID)`, inputting textual content to an input field `(TYPE,element_ID,value)` and selecting an option `(SELECT,element_ID,value)`.

$$\mathcal{L}_{\mathrm{DPO}}\left(\pi_\theta; \pi_{\mathrm{ref}}\right) = -\mathbb{E}_{(x,a_w,a_l)\sim\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta\left(a_w \mid x\right)}{\pi_{\mathrm{ref}}\left(a_w \mid x\right)} - \beta \log \frac{\pi_\theta\left(a_l \mid x\right)}{\pi_{\mathrm{ref}}\left(a_l \mid x\right)}\right)\right] \quad (1)$$

WEPO Implementation

We start by introducing the sampling mechanism of WEPO, as illustrated by the partial DOM tree shown in Figure 1. Since every webpage can be parsed into a corresponding DOM tree with each web element represented by a unique node, web elements corresponding to nodes that are closer in proximity under the same ancestor within the DOM typically exhibit greater functional and semantic similarity. Building on this characteristic, we have developed a distance-based sampling method specifically tailored to the

DOM tree structure, which begins by selecting a substantial number (top k) of element anchors on the page, including one correct element, and then calculates and sorts the sum of the distances between negative and positive samples to their lowest common ancestor (LCA) to quantify their relative distances. Subsequently, the method proportionally samples the top n closest elements from the sorted results, which are then combined with the correct element to form comparisons. We aim to enable the model to effectively learn to distinguish between web elements with similar functions based on the given operational intent.

During training, we implement Direct Preference Optimization in WEPO, since DPO is free of reward modeling and training stable (Rafailov et al. 2024). By optimizing the target loss function, WEPO aims to increase the likelihood of operations on preferred elements and decrease the likelihood of operations on dis-preferred elements. As shown in Equation 1, we introduce the maximum likelihood objective proposed in DPO and adaptively modify the preferred completion y_w and dis-preferred completions y_l into preference action pairs a_w and a_l. Given the pretrained model π_θ and the reference model π_{ref} initialized from π_θ, we fine-tune π_θ according to Equation 1, where β is a hyperparameter that controls the penalty for deviations from π_{ref}.

We demonstrate the pseudo-code for WEPO implementation in Algorithm 1, which illustrates how a_w, a_l, and x used for optimizing are obtained at each training step. First, we clean and prune the HTML code. Consistent with previous work (Gür et al. 2023; Deng et al. 2024), we adapt an element-centric approach to isolate HTML snippets. By focusing on a key element, we navigate its ancestors within the HTML tree, guided by a simple constraint that monitors the tree's width and depth. We stop this traversal once the number of descendants exceeds predefined thresholds, thus defining the snippet using the sub-tree. We introduce a pruning ratio k that represents the number of target elements remaining after pruning. We ensure that the pruned HTML snippet contains the ground truth element during training; In inference time, we use a small ranking LM derived from the candidate generation stage of Deng et al. (2024) to implement priority-based HTML pruning, which scores all elements first and then selects the top-k elements with the largest logits. Subsequently, we concatenate the preprocessed HTML state s', historical trajectory τ, initial intent i and a sophisticated prompt template P to generate the input x, where $\oplus$ denotes string concatenation.

Subsequently, we apply the tailored distance-based sampling method to yield candidate set E. We set the number of negative samples as n, corresponding to a positive-to-negative sample ratio of $1:n$. After obtaining the negative elements $\{e_{l_1}, e_{l_2}, ..., e_{l_n}\}$, we employ a designed heuristic rule f_{op} to randomly sample the corresponding negative operations o_{l_i}, which involves selectively replacing `TYPE` and `SELECT` to ensure balanced and diverse sample types. This replacement occurs only when $\hat{o} \neq$ `CLICK`, as the negative samples obtained through sampling have a very low probability of being `TYPE` or `SELECT`, adding data balance without confusing the LLM about the functionality of webpage elements. The rule empirically sets the replacement probability threshold at 0.33, and several verification confirmed that values around this threshold have no significant impact on WEPO. Therefore, when the positive sample is a `CLICK` operation, the negative samples are also `CLICK`; however, when the positive sample involves the other two actions, the negative samples might be changed to `CLICK`. Finally, we obtained a_w, a_l, and x, and used Equation 1 to perform gradient back-propagation on the pretrained LM π_θ, with β controlling the deviation from the reference model π_{ref}. We opt for straightforward random sampling as an alternative, where the randomly distributed negative samples also maintain a low correlation with $\hat{e}$ on the webpage.

Algorithm 1: WEPO Algorithm

Require: (for each step)
Intent i, current HTML s, action trajectory τ; prompt template P;
pretrained LM π_θ, reference LM π_{ref} and deviation parameter β;
Pruning ratio k, negative ratio n;
Target element $\hat{e}$, corresponding operation $\hat{o}$ and target action $\hat{a} = a(\hat{e}, \hat{o})$;

Ensure: (for each step)
1: Clean and prune the HTML DOM tree with k elements remaining $s' = f_{prune}(s, k)$;
2: Concatenate $x = s' \oplus \tau \oplus i \oplus P$
3: Get positive action $a_w = \hat{a}$;
4: Sample n negative elements $\{e_{l_1}, e_{l_2}, ..., e_{l_n}\}$ from candidate set $E \leftarrow s'$ based on LCA distance from $\hat{e}$;
5: **for** $i = 1$ to n **do**
6: Set $\epsilon \sim random\_uniform(0, 1)$
7: $o_{l_i} \sim f_{op}(\{\texttt{CLICK}, \texttt{TYPE}, \texttt{SELECT}\}, \hat{o}, \epsilon)$
8: Get i-th negative action $a_{l_i} = a(e_{l_i}, o_{l_i})$
9: Optimize $\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}})$ in Equation 1 given β, x, a_w and $a_l = a_{l_i}$;
10: **end for**

Experiments

Experimental Setup

We employ three mainstream pretrained LLMs of progressively increasing model sizes to validate the scaling effects. These models include Llama-3-8B[1], Mistral-7B-Instruct-v0.1 (Jiang et al. 2023), and Gemma-2B (Team et al. 2024). For the hyperparameters of WEPO, we set the deviation parameter β to 0.95 and the negative sample ratio to 1:3. In Mind2Web (Deng et al. 2024), a DeBERTa (He et al. 2020) model trained within the candidate generation module uses recall@50 for ranking elements and constructing a candidate pool for subsequent experiments. We similarly select a pruning ratio of $k = 50$ to maintain consistency for comparison, preserving 50 central elements and their neighboring elements tagged with `element_ID`. For the selection of k and n values, we provide detailed explanations in the forthcoming ablation studies. All models were configured with a maximum context length of 8192 tokens. We employ the

[1]https://llama.meta.com/llama3/

Model SSR / Op. F1 (%)	**overall**	**cross_domain**	**cross_task**	**cross_website**
Flan-T5-XL *MindAct*	43.5 / 69.1	39.6 / 66.5	52.0 / 75.7	38.9 / 65.2
Llama3-8B *MindAct*	55.1 / 65.8	57.6 / 68.7	56.1 / 63.2	51.7 / 65.6
CogVLM-17B *CogAgent*	58.2 / -	59.4 / -	62.3 / -	54.0 / -
HTML-T5-XL + Flan-U-PaLM *WebAgent*	49.7 / -	48.3 / -	57.8 / -	42.9 / -
Llama3-8B *WEPO random*	61.1 / 73.9	62.5 / 77.5	62.5 / 67.2	58.4 / **77.1**
Mistral-7B *WEPO random*	57.2 / 73.8	58.0 / 75.9	59.0 / 72.1	54.8 / 73.3
Gemma-2B *WEPO random*	45.4 / 49.5	49.1 / 55.7	45.2 / 42.9	41.9 / 50.0
Llama3-8B *WEPO distance-based*	**63.5** / 76.1	**64.4** / **81.6**	**66.1** / 74.9	**60.0** / 71.9
Mistral-7B *WEPO distance-based*	59.5 / **76.8**	59.8 / 80.9	62.1 / **76.2**	56.7 / 73.2
Gemma-2B *WEPO distance-based*	48.4 / 53.3	53.5 / 60.3	47.9 / 48.8	43.7 / 50.7

Table 1: Overall performance of various models on different test sets, with evaluation metrics corresponding to SSR / Operation F1 (%). The results were obtained under a negative sample ratio of $1:3$. Notably, our `Llama3-8B-WEPO` model achieved the highest scores in both overall SSR and Operation F1. Our top scores exceeded those of the much larger CogAgent (17B) model by 5.3% and WebAgent (3B + 540B) by 13.8%. Additionally, our smaller `Gemma-2B-WEPO` model managed to closely match and even slightly outperform the approximately 3B Flan-T5 based models (Chung et al. 2024) like MindAct.

Low Rank Adaptation (LoRA) technique (Hu et al. 2021) for parameter-efficient fine-tuning, which helps reduce memory usage and conserve budget. The learning rate is set at 0.0001, and we use a combination of learning rate warmup and a cosine decay strategy for training.

Evaluation Metrics. In this paper, we adopt the step success rate (SSR) and Operation F1 score from Deng et al. (2024). We no longer use element accuracy and success rate because it is evident that both metrics are linearly associated with SSR, and successful interaction for the web navigation agent is only considered when both the element positioning and the corresponding operation are correct, which is precisely what SSR measures. The Operation F1 score is equally indispensable as it considers the accuracy of the input value for `Type` and `SELECT` commands. Furthermore, both baseline studies by Gur et al. (2023) and Hong et al. (2023) exclusively utilized SSR as the sole metric.

Additionally, we designed the element distance metric to measure the positional deviation between the elements selected by the WEPO model and the labeled elements. Consistent with the previous sampling strategy, we calculate the sum of the distances (in terms of steps) between the nodes corresponding to two different web elements and their lowest common ancestor (LCA) in the DOM tree to represent their relative positions.

Results

We thoroughly evaluate WEPO on the partitioned three-tier held-out test sets in Mind2Web (Deng et al. 2024), including **cross-domain**, **cross-website** and **cross-task** datasets. This allows us to understand how well our method can generalize across different domains, websites, and tasks.

The experimental results are detailed in Table 1. Compared to previous works (Deng et al. 2024; Hong et al. 2023; Gur et al. 2023), without utilizing any multimodal information or customized model architecture, our best results obtained by fine-tuning the 8-billion parameter Llama-3 pretrained model with web element preference learning exceeded the three baselines by 20.0%, 5.3%, and 13.8%, respectively. Even though there is still a significant gap between the model performance of gemma-2B and the larger model ($> 10\%$), these results demonstrate the effectiveness of the WEPO method, proving its efficiency and generalizability in enabling LLM-based agents for web navigation tasks. In assessing generalization capabilities at different levels, although Deng et al. (2024) found that their model performed best in the Cross-Task setting, WEPO has narrowed the gap between different test sets to less than 6.1%. Additionally, observing the performance of WEPO models of different sizes, we found that the step success rate increases with model size, which verifies the presence of the scaling effect in the Mind2Web real-world benchmark.

To eliminate the influence of model base choice on the results, we thoroughly applied supervised fine-tuning using the action prediction prompts from Mind2Web on Llama3-8B. The results indicate that compared to this upgraded MindAct baseline, WEPO also outperforms by 8.4% on the SSR metric and by 11.0% on the Operation F1 score. This again demonstrates that fine-tuning with positive human annotation only, which is a simple maximum likelihood approach, is less effective than our WEPO method. We then conducted a comparison of its performance with random sampling in Table 1 to validate the effectiveness of the distance-based sampling method. Compared to random sampling, the distance-based approach achieved a 2.3% to 3% higher SSR, demonstrating its efficiency. In the subsequent analysis of the element distance evaluation metric, it can also be seen that this method successfully enhances the accuracy of the model's selections, indirectly suggesting better alignment with user high-level intents.

What exactly is the role of preference learning in enhancing performance? For the Operation F1 score, we observed a significant improvement from Gemma-2B-WEPO to Mistral-7B-WEPO, which indicates that an increase in the number of LLM training parameters not only makes the web element localization more accurate but primarily enhances

Model SSR (%)	**cross_domain**	**cross_task**	**cross_website**
Flan-T5-XL MindAct $k = 50$	39.6	52.0	38.9
Llama3-8B MindAct $k = 50$	57.6	56.1	51.7
Llama3-8B *WEPO* $k = 50$	64.4	66.1	60.0
Llama3-8B *WEPO* $k = 10$	49.7	55.0	46.5
Llama3-8B *WEPO* $k = 10$ w. ground truth $\hat{e}$	87.2	88.7	85.4

Table 2: SSR performance (%) from ablations on the cleaned HTML pruning ratio k value. For WEPO training, we forcibly included the ground truth element, akin to a teacher-forcing mechanism. The results shown in the table, except for the last row, represent the SSR after reassembling HTML from elements filtered through the first-stage ranking. Llama3-8B-WEPO maintained a 10.1% improvement to Flan-T5 based MindAct even after reducing k.

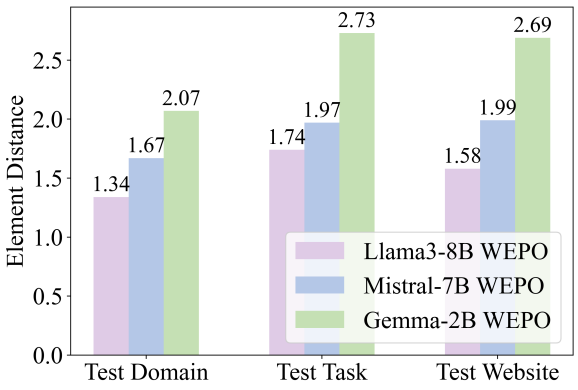

Figure 2: Statistical distribution of Element Distance for different models (Llama3-8B, Mistral-7B and Gemma-2B) on the test dataset. As the model size increases, the relative deviation in element distances decreases.

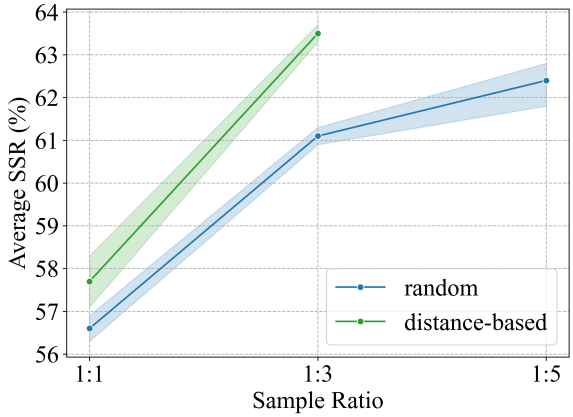

Figure 3: Ablation studies on the negative ratio. We experimented with the Llama3-8B-WEPO model at $n = 1, 3, 5$ and calculated the average SSR (%) on three cross-test sets, which were 57.7%, 63.5% for distance-based sampling and 56.6%, 61.1%, and 62.4% for random sampling, respectively. An elbow point was observed at $n = 3$ for random sampling, where the increase in SSR sharply levels off. Furthermore, the performance of distance-based sampling at a 1:3 ratio has already surpassed that of random sampling at a 1:5 ratio by 1.1%.

the accuracy of input or select text values. For MindAct, however, our best performance was not significantly ahead in F1 scores, suggesting that the WEPO method, compared to traditional supervised fine-tuning, enhances the accuracy of element selection in pretrained LMs. We infer that WEPO leverages this enhancement through a contrastive training scheme, wherein the model learns to distinguish between elements that are critical for decision-making and those that are not. Particularly when intentions are abstract, a web page at any given moment may contain multiple elements that are easily confused, and WEPO reduces the likelihood of incorrect element selection by the agent. By incorporating a contrastive mechanism, WEPO not only improves the accuracy of navigation tasks but also enhances the model's generalizability across different web layouts and designs.

The results in Figure 2 provide great support for this inference. As the model size increases, the element distance consistently decreases. Element distance, which represents the relative position of elements in the DOM, is closely correlated with the elements' function and design purpose. Coupled with the score analysis in Table 1, this decreasing deviation suggests that the Llama3-8B-WEPO model becomes increasingly accurate in aligning with the intended functions and design of web elements compared to the smaller models. This result clearly indicates that the model has successfully learned to recognize these web design differences, proving that WEPO is an effective method for adapting to the HTML structure, or more broadly for aligning with web design principles. Furthermore, we have demonstrated the effectiveness of our newly proposed element distance evaluation metric.

Why choose a 1:3 ratio for negative samples? We also conducted ablation experiments on the negative sample ratio in the WEPO Algorithm 1. Empirically, we aimed to sample as many negative samples as possible to enhance performance through WEPO without over-sampling and excessively increasing the training overhead. We uniformly sampled n values of 1, 3, and 5, selected the best-performing Llama3-8B-WEPO for experimentation and averaged the re-

sults from two rounds. As shown in Figure 3, the overall average scores on the test set increase with larger n values, with a noticeable elbow point at $n = 3$ for random sampling. We ultimately selected a 1:3 ratio as a fixed hyperparameter to avoid linearly increasing training costs. Additionally, we observed that the distance-based sampling method at $n = 3$ outperformed random sampling at $n = 5$, significantly improving overall sample learning efficiency during training. Furthermore, too many negative samples could potentially create an imbalance between positive and negative sample quantities. However, we did not experiment with larger n values to verify potential performance degradation, as excessively large n values could hinder reproducibility.

What is the impact of HTML k-pruning on performance? To address this question, we conducted ablation studies on the selection of k, with results shown in Table 2. As our pruning is based on the first stage ranking LM of MindAct, understanding the impact of this preprocessing module is crucial. The authors of Deng et al. (2024) disclosed that when $k = 50$, the fine-tuned DeBERTa (He et al. 2020) model achieved recall accuracies of 88.9%, 85.3%, and 85.7% on three held-out test datasets, and smaller values of k were not adopted due to lower recall rates.

However, in our results, when we set k to a smaller value of 10, Llama3-8B-WEPO still performed above the baseline. We deduce that although reducing k significantly decreases the recall rate of the ranking LM, it provides the WEPO-trained model with a shorter HTML snippet and fewer ID options, enhancing discrimination accuracy. Moreover, when we forcibly added the ground truth web element $\hat{e}$ directly into the candidate pool, the ablation model's SSR surged to over 80%. This conclusively demonstrates that the WEPO method has significantly improved the model's capability in action prediction and has shifted the original candidate retrieval module from a secondary issue to a major bottleneck limiting the web agent's progress on complex long-context web pages.

Future Work

There remain several avenues for further development of this approach. While WEPO has shown empirical success, a deeper theoretical analysis could provide more foundational insights into why and how preference optimization effectively enhances web navigation tasks. Inspired by the advancements of models like WebFormer (Wang et al. 2022) and HTML-T5 (Gür et al. 2023), future iterations of WEPO could benefit from a dedicated HTML encoder that is specifically tailored to understand the hierarchical and nested structures of web documents.

The current implementation of WEPO has not explored its potential over extremely large context lengths such as `phi-3-128k` (Abdin et al. 2024). Future research could look into the scaling abilities of WEPO when applied to such models, which might be crucial for handling complex web navigation tasks that involve detailed web pages.

While WEPO shows promising results in a controlled benchmark environment, its ability to generalize across highly diverse, real-world web interfaces remains an area for further investigation. The variability in web design, interactive elements, and underlying technologies across different websites may affect the consistency of WEPO's performance. We plan to test the performance of the WEPO method on additional mainstream benchmarks such as WebLINX and WebVoyager (Lù, Kasner, and Reddy 2024; He et al. 2024) in future work.

Conclusion

This paper introduced the Web Element Preference Optimization (WEPO), a simple yet novel framework that integrates Direct Preference Optimization (DPO) into LLM-based web navigation tasks. WEPO enhances LLM performance by leveraging distance-based non-salient HTML elements for contrastive learning, effectively aligning model operations with user intent. Our empirical evaluations on the Mind2Web benchmark demonstrate that WEPO surpasses traditional models, achieving an 13.8% improvement over the WebAgent baseline and a 5.3% enhancement beyond the visual language model CogAgent, while also exhibiting strong generalization across diverse web environments. This research significantly enhances the capabilities of LLMs in web navigation task, contributing to more efficient and intuitive web interactions.

Acknowledgments

This work was jointly supported by Beijing University of Posts and Telecommunications (BUPT) - China Mobile Research Institute Joint Innovation Center, and the National Key Laboratory of Networking and Switching Technology under Project of Embodied Intelligent Agent for Intellicise Networks. We extend our heartfelt gratitude to the faculty and staff at BUPT for their invaluable guidance and assistance throughout the research process. We also thank our collaborators and peers who provided critical feedback and constructive discussions that enriched the development of this work. Lastly, we acknowledge the broader academic community for inspiring this study through their pioneering efforts in autonomous web navigation and large language model applications.

References

Abdin, M.; Jacobs, S. A.; Awan, A. A.; Aneja, J.; Awadallah, A.; Awadalla, H.; Bach, N.; Bahree, A.; Bakhtiari, A.; Behl, H.; et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Amini, A.; Vieira, T.; and Cotterell, R. 2024. Direct Preference Optimization with an Offset. *arXiv preprint arXiv:2402.10571*.

Baechler, G.; Sunkara, S.; Wang, M.; Zubach, F.; Mansoor, H.; Etter, V.; Cărbune, V.; Lin, J.; Chen, J.; and Sharma, A. 2024. ScreenAI: A Vision-Language Model for UI and Infographics Understanding. *arXiv preprint arXiv:2402.04615*.

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Cheng, K.; Sun, Q.; Chu, Y.; Xu, F.; Li, Y.; Zhang, J.; and Wu, Z. 2024. Seeclick: Harnessing gui grounding for advanced visual gui agents. *arXiv preprint arXiv:2401.10935*.

Christiano, P. F.; Leike, J.; Brown, T.; Martic, M.; Legg, S.; and Amodei, D. 2017. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30.

Chung, H. W.; Hou, L.; Longpre, S.; Zoph, B.; Tay, Y.; Fedus, W.; Li, Y.; Wang, X.; Dehghani, M.; Brahma, S.; et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70): 1–53.

Deng, X.; Gu, Y.; Zheng, B.; Chen, S.; Stevens, S.; Wang, B.; Sun, H.; and Su, Y. 2024. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36.

Duan, S.; Yi, X.; Zhang, P.; Lu, T.; Xie, X.; and Gu, N. 2024. Negating Negatives: Alignment without Human Positive Samples via Distributional Dispreference Optimization. *arXiv preprint arXiv:2403.03419*.

Furuta, H.; Lee, K.-H.; Nachum, O.; Matsuo, Y.; Faust, A.; Gu, S. S.; and Gur, I. 2023. Multimodal Web Navigation with Instruction-Finetuned Foundation Models. In *The Twelfth International Conference on Learning Representations*.

Gur, I.; Furuta, H.; Huang, A. V.; Safdari, M.; Matsuo, Y.; Eck, D.; and Faust, A. 2023. A Real-World WebAgent with Planning, Long Context Understanding, and Program Synthesis. In *The Twelfth International Conference on Learning Representations*.

Gür, I.; Nachum, O.; Miao, Y.; Safdari, M.; Huang, A.; Chowdhery, A.; Narang, S.; Fiedel, N.; and Faust, A. 2023. Understanding HTML with Large Language Models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2803–2821.

He, H.; Yao, W.; Ma, K.; Yu, W.; Dai, Y.; Zhang, H.; Lan, Z.; and Yu, D. 2024. WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models. *arXiv preprint arXiv:2401.13919*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, P.; Liu, X.; Gao, J.; and Chen, W. 2020. DEBERTA: DECODING-ENHANCED BERT WITH DISENTANGLED ATTENTION. In *International Conference on Learning Representations*.

Hong, J.; Lee, N.; and Thorne, J. 2024. ORPO: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2(4): 5.

Hong, W.; Wang, W.; Lv, Q.; Xu, J.; Yu, W.; Ji, J.; Wang, Y.; Wang, Z.; Dong, Y.; Ding, M.; et al. 2023. Cogagent: A visual language model for gui agents. *arXiv preprint arXiv:2312.08914*.

Hu, E. J.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2021. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Humphreys, P. C.; Raposo, D.; Pohlen, T.; Thornton, G.; Chhaparia, R.; Muldal, A.; Abramson, J.; Georgiev, P.; Santoro, A.; and Lillicrap, T. 2022. A data-driven approach for learning to control computers. In *International Conference on Machine Learning*, 9466–9482. PMLR.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023. Mistral 7B. *arXiv preprint arXiv:2310.06825*.

Kim, G.; Baldi, P.; and McAleer, S. 2024. Language models can solve computer tasks. *Advances in Neural Information Processing Systems*, 36.

Li, T.; Li, G.; Deng, Z.; Wang, B.; and Li, Y. 2023. A Zero-Shot Language Agent for Computer Control with Structured Reflection. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Lù, X. H.; Kasner, Z.; and Reddy, S. 2024. Weblinx: Real-world website navigation with multi-turn dialogue. *arXiv preprint arXiv:2402.05930*.

Meng, Y.; Xia, M.; and Chen, D. 2024. SimPO: Simple Preference Optimization with a Reference-Free Reward. *arXiv preprint arXiv:2405.14734*.

Mialon, G.; Dessi, R.; Lomeli, M.; Nalmpantis, C.; Pasunuru, R.; Raileanu, R.; Roziere, B.; Schick, T.; Dwivedi-Yu, J.; Celikyilmaz, A.; et al. 2023. Augmented Language Models: a Survey. *Transactions on Machine Learning Research*.

Morimura, T.; Sakamoto, M.; Jinnai, Y.; Abe, K.; and Air, K. 2024. Filtered Direct Preference Optimization. *arXiv preprint arXiv:2404.13846*.

Muldrew, W.; Hayes, P.; Zhang, M.; and Barber, D. 2024. Active Preference Learning for Large Language Models. *arXiv preprint arXiv:2402.08114*.

Nakano, R.; Hilton, J.; Balaji, S.; Wu, J.; Ouyang, L.; Kim, C.; Hesse, C.; Jain, S.; Kosaraju, V.; Saunders, W.; et al. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Niu, R.; Li, J.; Wang, S.; Fu, Y.; Hu, X.; Leng, X.; Kong, H.; Chang, Y.; and Wang, Q. 2024. ScreenAgent: A Vision Language Model-driven Computer Control Agent. *arXiv preprint arXiv:2402.07945*.

OpenAI. 2023. ChatGPT Plugins. https://openai.com/blog/chatgpt-plugins.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Radlinski, F.; and Joachims, T. 2005. Query chains: learning to rank from implicit feedback. In *Proceedings of the*

eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, 239–248.

Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C. D.; Ermon, S.; and Finn, C. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

Shi, T.; Karpathy, A.; Fan, L.; Hernandez, J.; and Liang, P. 2017. World of bits: An open-domain platform for web-based agents. In *International Conference on Machine Learning*, 3135–3144. PMLR.

Sridhar, A.; Lo, R.; Xu, F. F.; Zhu, H.; and Zhou, S. 2023. Hierarchical Prompting Assists Large Language Model on Web Navigation. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. F. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33: 3008–3021.

Team, G.; Mesnard, T.; Hardin, C.; Dadashi, R.; Bhupatiraju, S.; Pathak, S.; Sifre, L.; Rivière, M.; Kale, M. S.; Love, J.; et al. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, J.; Xu, H.; Ye, J.; Yan, M.; Shen, W.; Zhang, J.; Huang, F.; and Sang, J. 2024a. Mobile-Agent: Autonomous multi-modal mobile device agent with visual perception. *arXiv preprint arXiv:2401.16158*.

Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. 2024b. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6): 1–26.

Wang, Q.; Fang, Y.; Ravula, A.; Feng, F.; Quan, X.; and Liu, D. 2022. Webformer: The web-page transformer for structure information extraction. In *Proceedings of the ACM Web Conference 2022*, 3124–3133.

Xi, Z.; Chen, W.; Guo, X.; He, W.; Ding, Y.; Hong, B.; Zhang, M.; Wang, J.; Jin, S.; Zhou, E.; et al. 2023. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*.

Xiang, B.; Jiang, D.; Pei, J.; Sun, X.; Chen, E.; and Li, H. 2010. Context-aware ranking in web search. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, 451–458.

Yao, S.; Chen, H.; Yang, J.; and Narasimhan, K. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35: 20744–20757.

You, K.; Zhang, H.; Schoop, E.; Weers, F.; Swearngin, A.; Nichols, J.; Yang, Y.; and Gan, Z. 2024. Ferret-UI: Grounded Mobile UI Understanding with Multimodal LLMs. *arXiv preprint arXiv:2404.05719*.

Zheng, L.; Wang, R.; and An, B. 2023. Synapse: Leveraging few-shot exemplars for human-level computer control. *arXiv preprint arXiv:2306.07863*.

Zhou, S.; Xu, F. F.; Zhu, H.; Zhou, X.; Lo, R.; Sridhar, A.; Cheng, X.; Ou, T.; Bisk, Y.; Fried, D.; et al. 2023. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *The Twelfth International Conference on Learning Representations*.

Zhu, Y.; Nie, J.-Y.; Dou, Z.; Ma, Z.; Zhang, X.; Du, P.; Zuo, X.; and Jiang, H. 2021. Contrastive learning of user behavior sequence for context-aware document ranking. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2780–2791.

Ziegler, D. M.; Stiennon, N.; Wu, J.; Brown, T. B.; Radford, A.; Amodei, D.; Christiano, P.; and Irving, G. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.