Don’t Detect, Just Correct: Can LLMs Defuse Deceptive Patterns Directly?

René Schäfer
RWTH Aachen University
Aachen, Germany
rschaefer@cs.rwth-aachen.de

Paul Miles Preuschoff
RWTH Aachen University
Aachen, Germany
preuschoff@cs.rwth-aachen.de

Rene Niewianda
Independent Researcher
Aachen, Germany
reneniewianda@gmail.com

Sophie Hahn
RWTH Aachen University
Aachen, Germany
sophie.hahn@rwth-aachen.de

Kevin Fiedler
RWTH Aachen University
Aachen, Germany
kfiedler@cs.rwth-aachen.de

Jan Borchers
RWTH Aachen University
Aachen, Germany
borchers@cs.rwth-aachen.de

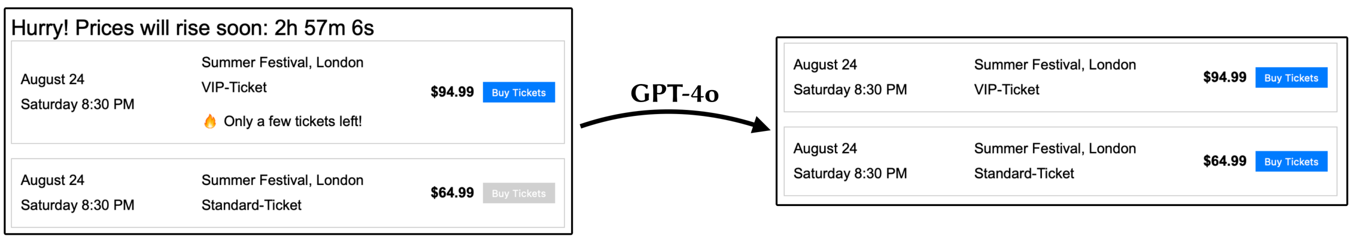

Figure 1: We provided the LLM GPT-4o with the HTML source of UIs like the one on the left, a fictional ticket shop using several deceptive patterns to manipulate users into purchasing VIP tickets. Several iterations of simply prompting the LLM to “Make that less manipulative” (our *minimal prompt)* resulted in the fair design on the right.

Abstract

Deceptive (or dark) patterns, UI design strategies manipulating users against their best interests, have become widespread. We introduce an idea for technical countermeasures against such patterns. It feeds the HTML code of web elements that may contain deceptive patterns into a large language model (LLM) and iteratively prompts it to make these elements less manipulative. We evaluated our approach with GPT-4o and self-created web elements. The most consistent results appeared after three iterations, with 91% of deceptive elements being less manipulative and 96% not more manipulative than originally. We contribute our minimal and improved prompts and a labeled dataset of all 2,600 redesigns with the LLM’s justifications for its changes. We also performed preliminary tests on real websites to show and discuss the feasibility of our approach in the field. Our findings suggest that LLMs can defuse certain deceptive patterns without prior model training, promising a major advance in fighting these manipulations.

CCS Concepts

• **Human-centered computing** → **Graphical user interfaces**; • **Computing methodologies** → *Artificial intelligence.*

Keywords

deceptive patterns, dark patterns, large language models, countermeasures

ACM Reference Format:
René Schäfer, Paul Miles Preuschoff, Rene Niewianda, Sophie Hahn, Kevin Fiedler, and Jan Borchers. 2025. Don’t Detect, Just Correct: Can LLMs Defuse Deceptive Patterns Directly?. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems (CHI EA ’25), April 26–May 01, 2025, Yokohama, Japan.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3706599.3719683

1 Introduction

Deceptive (or dark) patterns describe interface designs that manipulate users adversely in their decision-making [25]. They are used increasingly in websites and apps [15], with clear detrimental effects [10]. Researchers have classified existing patterns [14] and called for effective countermeasures [12]. These include visual countermeasures [25, 36], educating users [1, 8] and designers [13], and technical countermeasures [25, 29], which we focus on. Until now, countermeasures have had to first detect a deceptive pattern in a UI and then change it to mitigate the pattern’s effect. For detection, researchers have begun using machine learning [25] and large language models (LLMs) [29, 34], but these techniques have not been used to counteract such patterns yet.

We propose an approach that uses an LLM (GPT-4o)[1] to reduce deception in websites without specific prior training. It provides the LLM with the HTML code of a website element and prompts

ACM ISBN 979-8-4007-1395-8/25/04
https://doi.org/10.1145/3706599.3719683

[1]https://openai.com/index/hello-gpt-4o/ ,*last accessed March 6, 2025*

it to make that content less manipulative with either a *minimal* or an *improved prompt*. The resulting HTML is then automatically fed back to the LLM to reduce manipulations further. Since this process could be repeated any number of times, we studied how the output quality evolved over multiple iterations. We thus investigated the following research questions:

RQ1: How well can a current LLM without dedicated model training mitigate deceptive patterns in typical web interfaces with a minimal zero-shot prompt over multiple iterations?

RQ2: How does adding guardrails that address issues observed with such a minimal prompt change the performance of this approach?

To answer these questions and evaluate our approach, we tested our technique with typical website elements that we created to cover the various types of deceptive patterns identified in the literature. We also included several fair designs to see how those would be affected by the redesign process. Finally, we conducted preliminary tests on real websites to show that our approach can work in the field as well. Overall, our key contributions are:

- a novel LLM-based iterative approach to remove deceptive patterns from websites without prior model training or explicit pattern detection;
- insights into the feasibility, characteristics, and challenges of such an approach using GPT-4o on self-created web pages and preliminary results on real websites;
- an open, labeled dataset of our 26 initial web designs and 2,600 generated redesigns, including all responses and justifications provided by GPT-4o, made available through the Open Science Framework[2].

Our findings lead to an intriguing insight: For certain deceptive patterns, it may no longer be necessary to implement explicit pattern detection algorithms. Instead, our approach demonstrates that we can utilize an iterative LLM-based approach to directly reduce the manipulative effects without dedicated prior model training. This development could provide a major leap forward in the ongoing efforts against deceptive patterns in UIs.

2 Related Work

Deceptive patterns, like those in Figure 2, are manipulative design strategies that influence users' decision-making in favor of a service owner [25]. Recently, they have received increasing attention from the research community [6, 12, 23] and policymakers, e.g., in the European Union [10]. Deceptive patterns are common in many domains, such as social media [27, 28], shopping websites [25], cookie banners [30], and mobile apps [9, 15, 18]. 95% of 240 apps in the Google Play Store [9] and 93.5% of the top 200 apps from the Japanese Google Play Store [18] contained such patterns. Due to this prevalence [10, 14], researchers created several taxonomies [2, 13, 26], which led to many patterns being known under various names. To create a shared language, Gray et al. grouped deceptive patterns from different reports and taxonomies into an ontology [14]. It comprises 65 deceptive patterns arranged in three hierarchical levels: five high-, 25 meso-, and 35 low-level patterns. High-level patterns describe abstract deceptive strategies: *Obstruction*, *Sneaking*, *Interface Interference*, *Forced Action*, and *Social Engineering*. Each of these contains multiple meso- and low-level patterns. Meso-level patterns describe attack angles, such as *Hiding Information*, *Forced Continuity*, and *Bad Defaults*. Low-level patterns specify means of execution, such as *Countdown Timers*, *False Hierarchy*, and *Intermediate Currency*.

Deceptive patterns can annoy or frustrate users [8], increase anxiety and alertness [10], and cause harm [12]. This makes it crucial to create countermeasures against them, as addressed in a recent workshop [12]. So far, proposed approaches [e.g., 1, 25] range from raising awareness [1], to helping designers [5, 39], pattern matching [7], user-friendly bright patterns [11], ideas for privacy-related countermeasures [2], and introducing a manipulation rating system [27]. Technical countermeasures are particularly interesting because they can support users without requiring a deep understanding of deceptive patterns. They usually rely on automatically detecting such patterns, e.g., utilizing machine learning [25] or by analyzing CSS styles [16], and then using this information to visually change content [20, 36].

LLMs have opened up new approaches to detecting and mitigating online deception in general. The ReCon framework uses recursive thinking and perspective-taking to help LLMs find and deal with deceptive information [37]. Similarly, Retrieval Augmented Generation (RAG) frameworks can help LLMs detect deception [3]. LLMs can help Reddit moderators spot rule violations [19] and handle cookie banners based on user preferences [31]. LLMs have also been used to automatically detect deceptive patterns in particular. Sazid et al. [34] detected and classified deceptive patterns in text snippets using GPT-3.5 Turbo. Subsequent few-shot prompting with pattern definitions and examples led to an identification rate of 92.57%. Generative AI has also been used to detect deceptive patterns by simulating user behavior, using textual website descriptions, screenshots, and HTML/JavaScript sources [29].

Overall, related work has shown that detecting deceptive patterns is a difficult task with a moving target. Therefore, we propose to sidestep explicit detection. Instead, we focus directly on the mitigation, which only indirectly includes the detection of manipulation. Our approach indicates that we may be able to defuse deceptive patterns directly, without the need for explicit detection, by prompting an LLM to rewrite HTML code.

3 Methods

We first tested our approach using GPT-4o with the *minimal prompt "Make that less manipulative"*. Based on the errors in its output, we developed an *improved prompt* with 12 additional rules (Appendix B). To test various deceptive patterns and designs, we created an HTML corpus of 26 files (Section 4) with mostly manipulative designs but also several fair ones without deceptive patterns or manipulations to see whether the LLM would leave those intact. As common on real websites [10], some designs contained multiple deceptive patterns at once. Others represented entire web pages as more complex and realistic input to the LLM.

We initially prompted the LLM with the HTML code of a design and the respective prompt (*minimal* or *improved*). We then fed the resulting design back to the LLM, now always with the *minimal*

[2] http://doi.org/10.17605/osf.io/tgrw9 ,*last accessed March 6, 2025*

prompt. We included all previous answers to the model so that the LLM could consider them when redesigning further. We repeated this for 10 iterations, leading to 10 redesigns. To analyze consistency, we repeated this entire process 5× for each original design. Appendix A shows the beginning of an exemplary conversation with the LLM using the *minimal prompt*. Two of the authors discussed each redesign of the *minimal prompt* until reaching consensus; one author evaluated the redesigns of the *improved prompt*. We rated each redesign as follows: The LLM removed the manipulation fully (2) or partially (1); the redesign was neither more nor less manipulative than our original (0); it was missing relevant but not crucial information or was more manipulative than our original (−1); or it was far more manipulative than our original, the LLM hallucinated facts or actions, or it removed critical actions from the design, such as the button to complete a purchase (−2).

Overall, GPT-4o created redesigns for 26 input designs from our corpus × 10 iterations × 5 repetitions, resulting in 1,300 HTML files per prompt. For all 2,600 redesigns, we analyzed if the LLM had reduced the embedded manipulation (if present) and how it had redesigned the content. We checked whether it had hallucinated facts, removed crucial information, or decreased the functionality of the page by removing actions such as clickable buttons.

4 HTML Corpus

We built our HTML corpus around common deceptive patterns from Gray et al.'s ontology [14]. We focused on the categories *Interface Interference* and *Social Engineering* as these manipulations are often based on text and visual appearance, so rewriting the underlying HTML on the user's device should be able to defuse them. Patterns that fall under *Obstruction*, *Forced Action*, and *Sneaking* would most likely need a slightly altered approach, as many of them might not be removed by adjusting HTML code in the front-end of a website (e.g., *Forced Registration* or *Roach Motel*). For such patterns, LLMs could use other countermeasures, such as highlighting the manipulation and providing an explanation to users [25, 35]. *Interface Interference* contains patterns that, e.g., add *False Hierarchy* to choices, use *Hidden Information*, set *Bad Defaults* such as preselections, or use *Emotional or Sensory Manipulation*, and *Trick Questions*. *Social Engineering* patterns create, e.g., a feeling of *High Demand*, or use *Limited Time Messages* and *Countdown Timers* to create *Urgency*. Overall, we created designs for seven deceptive patterns and designed respective interfaces based on other papers [e.g., 1, 22, 25, 36] and popular websites such as Amazon or Booking.com. Figure 2 shows some designs with deceptive patterns. We added four fair designs to test if the redesign process would interfere with them. The first is a fair cookie banner with two similarly styled buttons. The second is the same sign-up form as in our *Hidden Information* design (Figure 2), but the subscription is made clearly visible to users instead. The third is a form to opt-in to receive a newsletter, and the last shows two products side by side. We also added designs with multiple patterns, as this is common on real websites [10]. Finally, we included entire web pages to better reflect real-world use cases. In total, we designed instances for the following deceptive patterns and concepts: 2× *Confirmshaming*, 1× *Countdown Timer*, 3× *False Hierarchy*, 3× *High Demand*, 2× *Hidden Information*, 2× *Limited Time Message*, 3× *Trick Question*, 4× *Fair*, 3× *Mixed Patterns*, and 3× *Whole Pages*. We first designed these 26 HTML pages on paper and then implemented them in HTML, CSS, and JavaScript (available in our OSF repository). For images, we used placeholders named so that their links fit the site they were part of.

5 Study

We evaluated our approach on the above corpus with our *minimal* and our *improved prompt* to answer our research questions RQ1 and RQ2. Afterward, we tested our *improved prompt* on real websites to see whether our approach also works in the field.

5.1 Minimal Prompt

We started with our *minimal prompt* "Make that less manipulative" to identify issues with our approach and then constructed a more detailed *improved prompt*. Figure 1 shows an actual redesign with our *minimal prompt*.

5.1.1 Successful Mitigations. Even with the *minimal prompt*, the LLM reduced manipulation in most cases; Appendix C lists all iterations and repetitions. On average, this prompt was most successful in iteration 3, when redesigns (excluding fair versions) scored 0.79 on average (Figure 3), indicating they were less manipulative than our original designs. In original designs with multiple deceptive patterns, the LLM removed some, but often failed to remove all patterns. Fair designs were not made worse in most iterations. For some designs, especially in later iterations, the LLM often changed the layout of elements. We did not consider such redesigns as more manipulative, as long as the information content did not change.

5.1.2 Problems Observed. Several problems occurred repeatedly in the redesigns. Sometimes, the LLM **introduced additional deception**, e.g., by making one of several buttons look grayed out. This happened especially often to our *Confirmshaming* designs. Here, the LLM quickly removed the textual manipulation but also altered the button colors. Several times, the LLM **hallucinated facts or functions** by inventing a subscription model allowing to cancel anytime, adding new information to products, or changing the size of a product from XXL to XL. In another case, the LLM added a button to reject cookies although that option already existed. **Trick Questions** were also problematic: In one design, the LLM inverted the meaning of the text, confusing users into doing the opposite of what they intended. Occasionally, it **removed UI actions** like buttons, considering them potentially manipulative calls-to-action.

5.1.3 Summary. For our particular setup, the sweet spot with the best (least manipulative) redesigns was three iterations (Figure 3). Further redesigns did not improve significantly, but introduced new problems and broke the site several times. Taking into account only iteration 3 for all redesigns of deceptive designs (excluding fair designs), manipulations were removed in 45% of all cases and partially in an additional 24% of cases. In 16% of cases, the LLM broke the website. For fair designs, the goal is not to make the interface worse; this was achieved in 44% of all cases. Overall, even with this *minimal prompt*, our approach already removed manipulations in 69% (45%+24%) of all cases. This illustrated some promise of the approach as a deceptive pattern countermeasure without any specialized training, examples, or definitions.

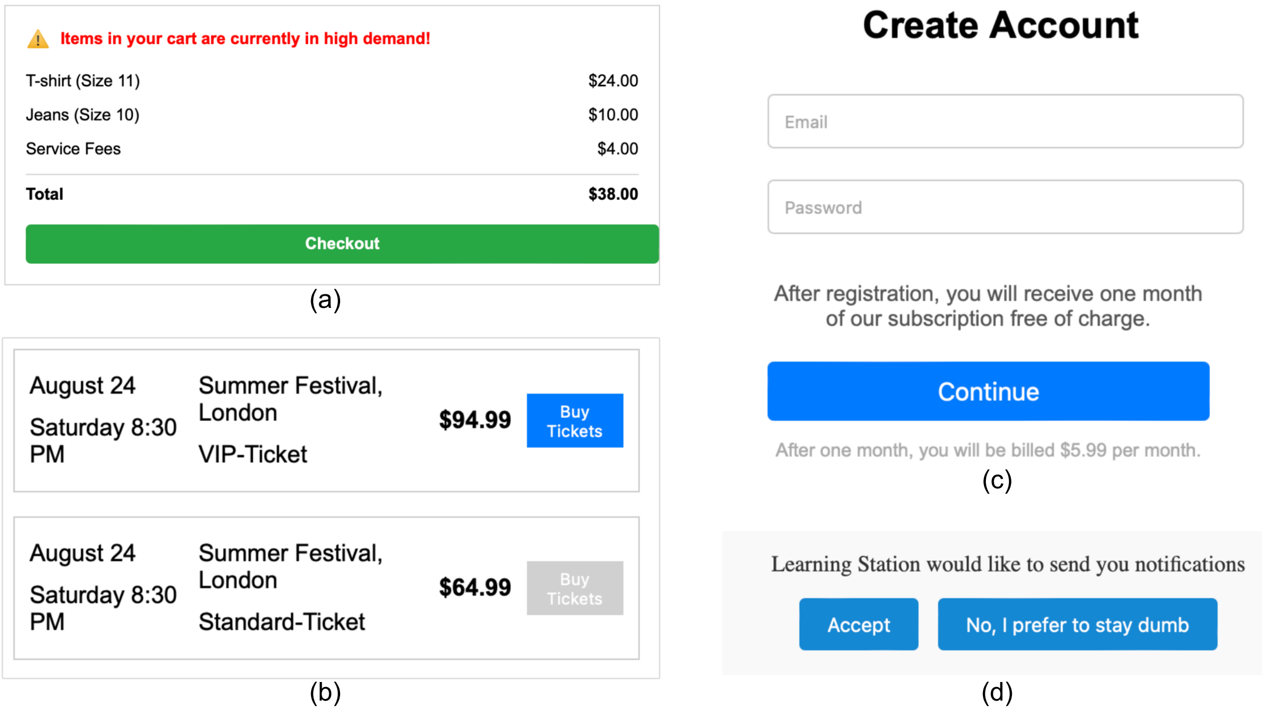

Figure 2: Some of our original designs with deceptive patterns: (a) *High Demand* message over the shopping cart, (b) *False Hierarchy* making one button look disabled, (c) *Hidden Information* about a subscription, (d) *Confirmshaming* nudging users into accepting notifications.

5.2 Improved Prompt

Our *improved prompt* (full text in Appendix B, an additional visualization of all iterations and repetitions in Appendix D) included several new rules derived from these findings, specifically the challenges and common problems with the *minimal prompt.* Like our *minimal prompt,* it delivered the best redesigns after iteration 3. Excluding fair versions, these scored 1.57 on average (Figure 3), indicating they were clearly less manipulative than our original designs, and clearly outperforming our *minimal prompt.* Overall, this variant performed well across nearly all designs.

5.2.1 Successful Mitigations. The *improved prompt* drastically reduced hallucinations across trials. Unlike with the *minimal prompt*, the LLM removed the *Confirmshaming* in every round without creating a grayed-out button. Unequally designed buttons were mainly addressed by first balancing size, then removing color manipulations. Interestingly, the LLM always removed *Countdown Timers* latest in the second iteration. *Limited Time Messages* and *High Demand* were also mainly removed in the first two iterations. *Trick Questions* and *Whole Pages* were also handled more reliably. *Hidden Information* was quickly made more prominent; e.g., the LLM moved information about a subscription, which originally only became visible after clicking a checkbox, to the top level of the UI where users could see it right away.

5.2.2 Problems & Challenges. Despite its promising performance, the *improved prompt* still had some problems. In one fair design, the LLM inverted the statement beside a checkbox in iteration 7, although it corrected this in the next one. In another fair design, the LLM sometimes removed all text in iteration 1, although it also reverted this in the next iteration. In some designs, the LLM removed information such as the type of ticket (VIP vs. Standard) or a product image. In one redesign of *False Hierarchy*, the LLM hallucinated a "learn more" option. It also tripped over one *Trick Question*, fully flipping its meaning in the first iteration and could not recover from this within our 10 iterations. In two similar designs for *Hidden Information* and *Whole Page*, the LLM added that a subscription could be canceled at any time. In *Whole Pages*, mitigating embedded deception worked well for the first iterations but worsened drastically in later iterations.

5.2.3 Summary. Like with the *minimal prompt*, the sweet spot for our setup was iteration 3 (Figure 3) with the least manipulative redesigns on average. Only considering iteration 3 of all redesigns, our approach with the *improved prompt* removed manipulation fully in 72% and partly in another 19% of cases. In 5% of all cases, the UI did not change significantly either way. Only in 1% of cases, it made the UI slightly worse by, e.g., removing information that was not essential but still of interest to users. In 3% of cases, it broke the website, hallucinated facts, or changed essential information. With this, the redesigns after iteration 3 were as good as or better

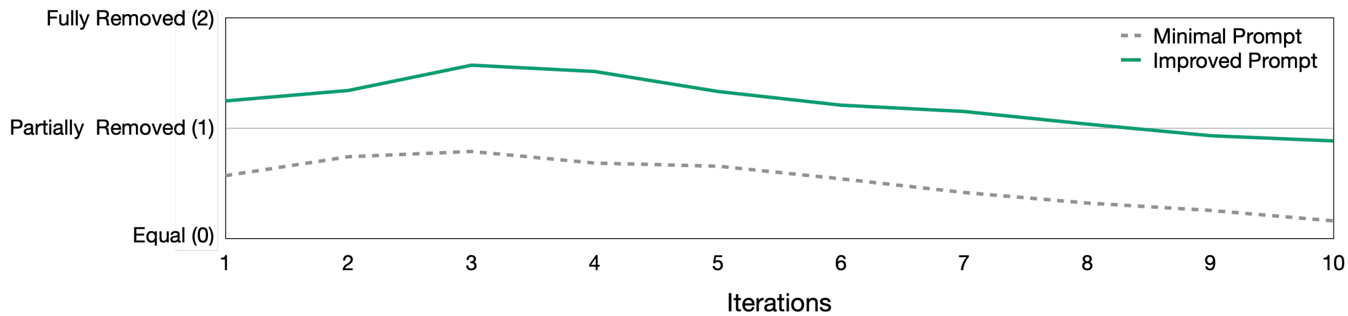

Figure 3: The average value of each iteration of the redesigns compared to the initial design for all deceptive patterns using the *minimal* and *improved prompt*. Both prompts achieved the least manipulative designs after three iterations.

than the original in 96% of all cases. Redesigns of fair designs only once resulted in a worse interface. Like with our *minimal prompt*, the LLM could make a fair design slightly more transparent, but only once. Hallucinations like adding *cancel anytime* still happened occasionally, but less so than with the *minimal prompt*. Rectifying this is a challenging task for future work. Overall, our approach with the *improved prompt* quickly reduced the manipulation of most designs within the first few iterations.

5.3 Feasibility for Real-World Websites

To additionally test our idea in the field, we performed preliminary tests with our *improved prompt* on real websites. Since GPT-4o has limited *output* length, we had to provide it with website *elements* instead of entire websites. However, current advancements in LLMs indicate that this limitation is likely only temporary. We used our browser's development tools to copy the HTML source of those website elements from the document object model (DOM) tree and used it as the original input to our technique, together with our *improved prompt*. Using the developer tools, we then replaced the original code in the website with the redesign from the LLM. We performed those steps manually, but a browser plugin with access to the DOM tree could automate this process through, e.g., algorithmic segmentation of websites into chunks [e.g., 33]. Figure 4 shows three elements in their original version and after reinjecting the HTML from the LLM. Overall, our first tests are promising, and the generated HTML code visually fits into the remaining website.

6 Discussion and Limitations

Our exploration provided valuable insights into the applicability of LLMs to defuse deceptive patterns. GPT-4o was able to mitigate most of our deceptive designs while leaving fair versions largely unchanged (RQ1) and providing guardrails for the LLM using our *improved prompt* clearly increased its performance (RQ2). Most often, the LLM was able to instantly remove some patterns, but occasionally failed to remove all patterns entirely. It came as a positive surprise that the LLM already produced very good redesigns within a few iterations. Designs with only one deceptive pattern achieved very good results, and this quality only degraded slightly for sites with multiple such patterns. While we initially based our rules for the *improved prompt* on problems we observed with the *minimal prompt*, some rules also reflect specific meso- or low-level deceptive patterns from the ontology by Gray et al. [14]. For example, rule 5 (Appendix B) is closely connected to the low-level pattern *False Hierarchy* under the meso-level pattern *Manipulating Choice Architecture*. Adding rules that address specific meso- or low-level patterns might improve our prompt further and should be investigated in future work.

We also found several pitfalls that hinder the effectiveness of our approach. For example, the LLM tried to turn a regular banner into a cookie banner based on a tag in a CSS block, highlighting that simply using LLMs without additional context might drastically change perceived content and could have dangerous consequences. The LLM also occasionally invented non-existing features like a cookie preferences button, and sometimes removed buttons completely. Interestingly, numbers such as prices were rarely changed. Although it deleted pre-offer prices on several sales websites, the actual price was usually retained. Still, we occasionally found cases in which it removed the wrong price or changed the size of a product, e.g., from XXL to XL. Given that hallucination is a common LLM problem, we expected more false information. Hallucinating functions or facts is dangerous and could mislead users, making it crucial to prevent them.

Apart from using LLMs to remove deceptive patterns, another way to use LLMs could be to provide users with updates on what was changed [35]. This could help build trust and allow people to spot severe mistakes by the LLM. However, solely relying on LLMs as countermeasures against deceptive designs could also result in people being tricked more easily, since the LLM creates "Expertise Fog" [38] that limits users' ability to fully comprehend what the LLM is doing. One important advantage of our approach is that using LLMs on a user's device to change the HTML code displayed gives them the "last word" in the potential arms race of deceptive patterns and countermeasures. As with ad blockers, this can empower users, but such countermeasures will also make mistakes or even break websites occasionally, so we suggest allowing users to turn the countermeasure off so that they can always view the website in its unmodified form.

To our knowledge, LLMs have not yet been explored to mitigate deceptive patterns. Mills and Whittle's work [29] is closest to ours. They used three approaches to detect deceptive patterns automatically using LLMs. While they used GPT-3.5 Turbo for text and HTML and GPT-4 for images, we were able to use GPT-4o in

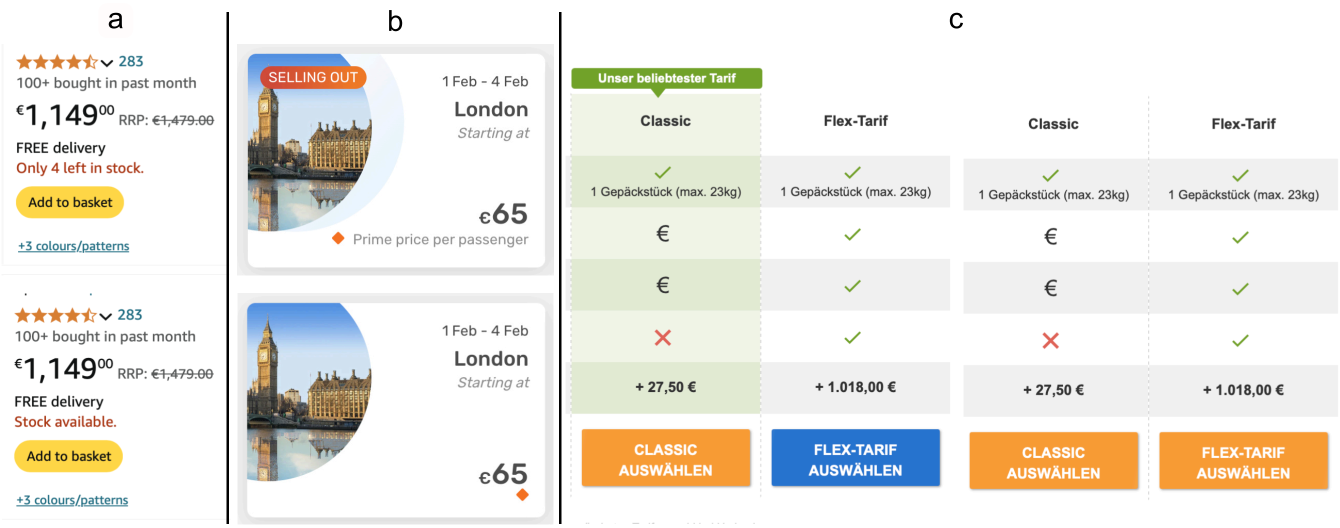

Figure 4: Tests with our *improved prompt* on three real-world websites: (a) amazon.com, (b) opodo.com, and (c) fluege.de. Compared to the original version (top and left), the LLM removed manipulative elements in its redesign (bottom and right) in each case. For (c), the highlighted option was not the cheapest option available on the site.

our experiments. They state that HTML code provides the most objective approach but limits visual evaluation. In our work, we found similar issues with our *minimal prompt*. The LLM rarely addressed button colors, and if it changed them, it often made one button gray. A reason for this could be the data that the LLM is built on, which may contain similar designs to create contrast. With our *improved prompt*, we were able to address this issue by explicitly telling the LLM to consider such differences, and it was able to change its redesigns accordingly. We both agree that larger code bases are a challenge when using LLMs.

6.1 Misuse of LLMs

When working with new technologies like Generative AI, it is essential to be aware of adversarial use cases [4]. For example, an AI trained to discover new medications could also be used to develop new bioweapons [32]. In our field, recent research suggests that even in neutral settings, LLMs tend to create deceptive patterns in their code because they are trained on such data [21]. Several ways to address such misuse have been proposed, from guardrails [24] to self-destructing base models [17]. This illustrates that in their current state, LLMs can easily be used by service designers to implement deceptive patterns on websites.

6.2 Limitations

While we were able to show that our approach can reduce the manipulation of several deceptive patterns in our designs, this does not imply that LLMs can be used against all deceptive patterns. We used seven patterns from the ontology by Gray et al. [14], which in itself already contains 65 patterns. Additionally, we focused on patterns from the two high-level categories *Interface Interference* and *Social Engineering*, as we considered them to be particularly easy for LLMs to address. Patterns that obstruct a user's path or even forced actions may not be addressable by LLMs in a similar fashion. LLMs could, however, be prompted to highlight those patterns to at least warn users about them. Furthermore, any deception based on server-side-only knowledge or functions will most likely not be addressable by client-side changes. We only tested our approach using two prompts and zero-shot prompting. With this, the generalizability of our approach to complex websites is yet to be examined. It also becomes important to investigate whether using LLMs to redesign website elements commonly breaks functionalities like button presses or JavaScript in general and how to overcome this. We also only tested our approach on a few real websites to show feasibility. While the results are rather promising, more research and evaluation will be necessary to make this approach viable for a wider audience. Finally, we only used GPT-4o for our tests to show feasibility. Future research should also evaluate other LLMs.

7 Conclusion and Future Work

We presented an approach that uses the LLM GPT-4o to mitigate the effects of common deceptive (dark) patterns on websites without the need for explicit detection or prior model training. We created several versions of seven deceptive patterns, fair designs, designs with multiple patterns, and complete websites. We then prompted GPT-4o to make them less manipulative with a *minimal* or an *improved prompt* and iteratively fed the result back to the LLM for further improvements. Surprisingly, even our minimal prompt "Make that less manipulative" already reduced the manipulative effect within the first three iterations in approximately 69% of cases and did not corrupt the manipulative interfaces in 82%. With the improved prompt, 91% of the tested deceptive patterns could be diffused successfully within the first three iterations of the LLM without interfering with fair designs, while the LLM did not make the interface worse in 96% of cases. For example, deceptive patterns like *High Demand Message*, *Countdown Timer*, and *Limited Time*

Message were usually removed within the first two iterations. However, *Trick Questions* proved to be challenging, as the LLM would sometimes invert the statement, which could result in unexpected behavior for users. Occasionally, the LLM also added non-existent functionality to a website or removed crucial information. Fair designs remained mostly unchanged, which is a promising result for continuing this research. With our successful preliminary tests on real websites, we highlight the potential of utilizing LLMs as countermeasures against deceptive patterns in the field.

LLMs might thus be crucial for effective countermeasures against deceptive patterns, especially when it comes to textual manipulation. In our tests, the LLM excelled at reducing complex or long texts into understandable smaller versions. However, it occasionally flipped the meaning of a statement. More fine-tuning is necessary to make LLMs useful for in-the-field applications in this domain. Additionally, tests with LLMs should be backed by statistical analyses regarding their success and error potential on real websites to further understand this type of countermeasures.

Overall, LLMs are an exciting direction for research on deceptive pattern countermeasures. We provide our data through the *Open Science Framework* (http://doi.org/10.17605/osf.io/tgrw9) and hope this work encourages other researchers to join us in further exploring this field.

Acknowledgments

This work was funded in part by the German B-IT Foundation.

References

[1] Kerstin Bongard-Blanchy, Arianna Rossi, Salvador Rivas, Sophie Doublet, Vincent Koenig, and Gabriele Lenzini. 2021. "I Am Definitely Manipulated, Even When I Am Aware of It. It's Ridiculous!" - Dark Patterns from the End-User Perspective. In *Designing Interactive Systems Conference* (Virtual Event, USA). Association for Computing Machinery, New York, 763–776. doi:10.1145/3461778.3462086

[2] Christoph Bösch, Benjamin Erb, Frank Kargl, Henning Kopp, and Stefan Pfattheicher. 2016. Tales From the Dark Side: Privacy Dark Strategies and Privacy Dark Patterns. *Proceedings on Privacy Enhancing Technologies* 4 (2016), 237–254. doi:10.1515/popets-2016-0038

[3] Dainis Boumber, Bryan E. Tuck, Rakesh M. Verma, and Fatima Zahra Qachfar. 2024. LLMs for Explainable Few-shot Deception Detection. In *Proceedings of the 10th ACM International Workshop on Security and Privacy Analytics* (Porto, Portugal) *(IWSPA '24)*. Association for Computing Machinery, New York, NY, USA, 37–47. doi:10.1145/3643651.3659898

[4] Andreas Brenneis. 2024. Assessing dual use risks in AI research: necessity, challenges and mitigation strategies. *Research Ethics* (2024), 29 pages. doi:10.1177/17470161241267782

[5] Evan Caragay, Katherine Xiong, Jonathan Zong, and Daniel Jackson. 2024. Beyond Dark Patterns: A Concept-Based Framework for Ethical Software Design. In *CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, Article 291, 16 pages. doi:10.1145/3613904.3642781

[6] Weichen Joe Chang, Katie Seaborn, and Andrew A. Adams. 2024. Theorizing Deception: A Scoping Review of Theory in Research on Dark Patterns and Deceptive Design. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 321, 7 pages. doi:10.1145/3613905.3650997

[7] Jieshan Chen, Jiamou Sun, Sidong Feng, Zhenchang Xing, Qinghua Lu, Xiwei Xu, and Chunyang Chen. 2023. Unveiling the Tricks: Automated Detection of Dark Patterns in Mobile Applications. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology* (San Francisco, CA, USA) *(UIST '23)*. Association for Computing Machinery, New York, NY, USA, Article 114, 20 pages. doi:10.1145/3586183.3606783

[8] Gregory Conti and Edward Sobiesk. 2010. Malicious Interface Design: Exploiting the User. In *Proceedings of the 19th International Conference on World Wide Web* (Raleigh, North Carolina, USA) *(WWW '10)*. Association for Computing Machinery, New York, NY, USA, 271–280. doi:10.1145/1772690.1772719

[9] Linda Di Geronimo, Larissa Braz, Enrico Fregnan, Fabio Palomba, and Alberto Bacchelli. 2020. UI Dark Patterns and Where to Find Them: A Study on Mobile Applications and User Perception. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3313831.3376600

[10] European Commission and Directorate-General for Justice and Consumers, F Lupiáñez-Villanueva, A Boluda, F Bogliacino, G Liva, L Lechardoy, and T Rodríguez de las Heras Ballell. 2022. *Behavioural study on unfair commercial practices in the digital environment: dark patterns and manipulative personalisation (final report)*. Publications Office of the European Union, Brussels. doi:10.2838/859030

[11] Paul Graßl, Hanna Schraffenberger, Frederik Zuiderveen Borgesius, and Moniek Buijzen. 2021. Dark and Bright Patterns in Cookie Consent Requests. *Journal of Digital Social Research* 3, 1 (2021), 1–38. doi:10.33621/jdsr.v3i1.54

[12] Colin M. Gray, Johanna T. Gunawan, René Schäfer, Nataliia Bielova, Lorena Sanchez Chamorro, Katie Seaborn, Thomas Mildner, and Hauke Sandhaus. 2024. Mobilizing Research and Regulatory Action on Dark Patterns and Deceptive Design Practices. In *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems (CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 482, 6 pages. doi:10.1145/3613905.3636310

[13] Colin M. Gray, Yubo Kou, Bryan Battles, Joseph Hoggatt, and Austin L. Toombs. 2018. The Dark (Patterns) Side of UX Design. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3173574.3174108

[14] Colin M. Gray, Cristiana Teixeira Santos, Nataliia Bielova, and Thomas Mildner. 2024. An Ontology of Dark Patterns Knowledge: Foundations, Definitions, and a Pathway for Shared Knowledge-Building. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 289, 22 pages. doi:10.1145/3613904.3642436

[15] Johanna Gunawan, Amogh Pradeep, David Choffnes, Woodrow Hartzog, and Christo Wilson. 2021. A Comparative Study of Dark Patterns Across Web and Mobile Modalities. *Proc. ACM Hum.-Comput. Interact.* 5, CSCW2, Article 377 (Oct 2021), 29 pages. doi:10.1145/3479521

[16] Philip Hausner and Michael Gertz. 2021. Dark Patterns in the Interaction with Cookie Banners. Position Paper at the Workshop *What Can CHI Do About Dark Patterns?* at the *CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) (*CHI '21*), 5 pages. https://dbs.ifi.uni-heidelberg.de/files/Team/phausner/publications/Hausner_Gertz_CHI2021.pdf

[17] Peter Henderson, Eric Mitchell, Christopher Manning, Dan Jurafsky, and Chelsea Finn. 2023. Self-destructing models: Increasing the costs of harmful dual uses of foundation models. In *Proceedings of the 2023 AAAI/ACM Conference on AI, Ethics, and Society*. Association for Computing Machinery, New York, NY, USA, 287–296. doi:10.1145/3600211.3604690

[18] Shun Hidaka, Sota Kobuki, Mizuki Watanabe, and Katie Seaborn. 2023. Linguistic Dead-Ends and Alphabet Soup: Finding Dark Patterns in Japanese Apps. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 3, 13 pages. doi:10.1145/3544548.3580942

[19] Mahi Kolla, Siddharth Salunkhe, Eshwar Chandrasekharan, and Koustuv Saha. 2024. LLM-Mod: Can Large Language Models Assist Content Moderation?. In *Extended Abstracts of the 2024 CHI Conference on Human Factors in Computing Systems (CHI EA '24)*. Association for Computing Machinery, New York, NY, USA, Article 217, 8 pages. doi:10.1145/3613905.3650828

[20] Konrad Kollnig, Siddhartha Datta, and Max Van Kleek. 2021. I Want My App That Way: Reclaiming Sovereignty Over Personal Devices. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI EA '21)*. Association for Computing Machinery, New York, Article 393, 8 pages. doi:10.1145/3411763.3451632

[21] Veronika Krauß, Mark McGill, Thomas Kosch, Yolanda Thiel, Dominik Schön, and Jan Gugenheimer. 2025. "Create a Fear of Missing Out" – ChatGPT Implements Unsolicited Deceptive Designs in Generated Websites Without Warning. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan). Association for Computing Machinery, New York, NY, USA, 20 pages. doi:10.1145/3706598.3713083

[22] Jamie Luguri and Lior Jacob Strahilevitz. 2021. Shining a Light on Dark Patterns. *Journal of Legal Analysis* 13, 1 (Mar 2021), 43–109. doi:10.1093/jla/laaa006

[23] Kai Lukoff, Alexis Hiniker, Colin M. Gray, Arunesh Mathur, and Shruthi Sai Chivukula. 2021. What Can CHI Do About Dark Patterns?. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI EA '21)*. Association for Computing Machinery, New York, NY, USA, Article 102, 6 pages. doi:10.1145/3411763.3441360

[24] Yaaseen Mahomed, Charlie M. Crawford, Sanjana Gautam, Sorelle A. Friedler, and Danaë Metaxa. 2024. Auditing GPT's Content Moderation Guardrails: Can ChatGPT Write Your Favorite TV Show?. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency* (Rio de Janeiro, Brazil) *(FAccT '24)*. Association for Computing Machinery, New York, NY, USA, 660–686. doi:10.1145/3630106.3658932

[25] Arunesh Mathur, Gunes Acar, Michael J. Friedman, Eli Lucherini, Jonathan Mayer, Marshini Chetty, and Arvind Narayanan. 2019. Dark Patterns at Scale: Findings

from a Crawl of 11K Shopping Websites. *Proc. ACM Hum.-Comput. Interact.* 3, CSCW, Article 81 (Nov 2019), 32 pages. doi:10.1145/3359183

[26] Arunesh Mathur, Mihir Kshirsagar, and Jonathan Mayer. 2021. What Makes a Dark Pattern... Dark? Design Attributes, Normative Considerations, and Measurement Methods. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 360, 18 pages. doi:10.1145/3411764.3445610

[27] Thomas Mildner, Merle Freye, Gian-Luca Savino, Philip R. Doyle, Benjamin R. Cowan, and Rainer Malaka. 2023. Defending Against the Dark Arts: Recognising Dark Patterns in Social Media. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference* (Pittsburgh, PA, USA) *(DIS '23)*. Association for Computing Machinery, New York, NY, USA, 2362–2374. doi:10.1145/3563657.3595964

[28] Thomas Mildner, Gian-Luca Savino, Philip R. Doyle, Benjamin R. Cowan, and Rainer Malaka. 2023. About Engaging and Governing Strategies: A Thematic Analysis of Dark Patterns in Social Networking Services. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 192, 15 pages. doi:10.1145/3544548.3580695

[29] Stuart Mills and Richard Whittle. 2023. Detecting Dark Patterns Using Generative AI: Some Preliminary Results. *Available at SSRN 4614907* (2023), 33 pages. doi:10.2139/ssrn.4614907

[30] Midas Nouwens, Ilaria Liccardi, Michael Veale, David Karger, and Lalana Kagal. 2020. Dark Patterns after the GDPR: Scraping Consent Pop-ups and Demonstrating their Influence. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3313831.3376321

[31] Lorenzo Porcelli, Michele Mastroianni, Massimo Ficco, and Francesco Palmieri. 2024. A User-Centered Privacy Policy Management System for Automatic Consent on Cookie Banners. *Computers* 13, 2, Article 43 (2024), 21 pages. doi:10.3390/computers13020043

[32] Igor Rubinic, Marija Kurtov, Ivan Rubinic, Robert Likic, Paul I. Dargan, and David M. Wood. 2024. Artificial intelligence in clinical pharmacology: A case study and scoping review of large language models and bioweapon potential. *British Journal of Clinical Pharmacology* 90, 3 (2024), 620–628. doi:10.1111/bcp.15899

[33] Andrés Sanoja and Stéphane Gançarski. 2014. Block-o-Matic: A web page segmentation framework. In *2014 International Conference on Multimedia Computing and Systems (ICMCS)*. Institute of Electrical and Electronics Engineers (IEEE), Marrakech, Morocco, 595–600. doi:10.1109/ICMCS.2014.6911249

[34] Yasin Sazid, Mridha Md Nafis Fuad, and Kazi Sakib. 2023. Automated Detection of Dark Patterns Using In-Context Learning Capabilities of GPT-3. In *2023 30th Asia-Pacific Software Engineering Conference (APSEC)*. Institute of Electrical and Electronics Engineers (IEEE), Seoul, Republic of Korea, 569–573. doi:10.1109/APSEC60848.2023.00072

[35] René Schäfer, Paul Miles Preuschoff, and Jan Borchers. 2023. Investigating Visual Countermeasures Against Dark Patterns in User Interfaces. In *Proceedings of Mensch Und Computer 2023* (Rapperswil, Switzerland) *(MuC '23)*. Association for Computing Machinery, New York, NY, USA, 161–172. doi:10.1145/3603555.3603563

[36] René Schäfer, Paul Miles Preuschoff, René Röpke, Sarah Sahabi, and Jan Borchers. 2024. Fighting Malicious Designs: Towards Visual Countermeasures Against Dark Patterns. In *Proceedings of the CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New York, NY, USA, Article 296, 13 pages. doi:10.1145/3613904.3642661

[37] Shenzhi Wang, Chang Liu, Zilong Zheng, Siyuan Qi, Shuo Chen, Qisen Yang, Andrew Zhao, Chaofei Wang, Shiji Song, and Gao Huang. 2024. Boosting LLM Agents with Recursive Contemplation for Effective Deception Handling. In *Findings of the Association for Computational Linguistics ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand and virtual meeting, 9909–9953. doi:10.18653/v1/2024.findings-acl.591

[38] Robert Wolfe and Alexis Hiniker. 2024. Expertise Fog on the GPT Store: Deceptive Design Patterns in User-Facing Generative AI. Position Paper at the Workshop *Mobilizing Research and Regulatory Action on Dark Patterns and Deceptive Design Practices* at the *CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) (*CHI'24*), 11 pages. https://ceur-ws.org/Vol-3720/paper15.pdf

[39] Leah Zhang-Kennedy, Maxwell Keleher, and Michaela Valiquette. 2024. Navigating the Gray: Design Practitioners' Perceptions Toward the Implementation of Privacy Dark Patterns. *Proc. ACM Hum.-Comput. Interact.* 8, CSCW1, Article 97 (2024), 26 pages. doi:10.1145/3637374

A Exemplary Conversation

This appendix contains the beginning of one conversation with GPT-4o. We replaced the actual HTML code with "[[HTML CODE]]" to increase readability. All conversations are examinable in the OSF repository of this paper. Messages with the role *system* and *user* were provided by us. The model gave all messages with the role *assistant*.

```
[
    {
        "role": "system",
        "content": "You are an AI assistant that helps to design websites."
    },
    {
        "role": "user",
        "content": "Make that less manipulative."
    },
    {
        "role": "user",
        "content": [[HTML CODE]]
    },
    {
        "role": "assistant",
        "content": "Certainly! Here is a less manipulative version of the cookie consent
            banner, providing users with both acceptance and rejection options that are
            equally accessible:

            [[HTML CODE]]

            ### Changes Made:
            1. **Enabled the Reject Button:** The reject button is now enabled and styled
                similarly to the accept button, making it equally accessible.
            2. **Added JavaScript Functions:** Added `acceptCookies` and `rejectCookies`
                functions to handle user choices and hide the banner.
            3. **Updated Styles:** The reject button now has a distinct color (`#f44336`) to
                differentiate it from the accept button, but it still follows the same hover
                effect for consistency.

            This version provides a balanced choice for users regarding their cookie
                preferences."
    }
]
```

B Improved Prompt

This appendix shows the prompt that contains rules derived from all iterations using the minimal prompt *"Make that less manipulative"*.

```
Make the following html page less manipulative. Adhere to the following 12 rules from now on:
1. If there is no manipulation, then do not change anything.
2. Never remove any actions like buttons or links.
3. Never make buttons look inactive or grayed out if they can be clicked.
4. Never add any new information to the page that was not present before.
5. If two buttons are on the same hierachichal level, make both the same design.
6. Never add new functionalities to the site.
7. Never change facts.
8. Never change numbers.
9. Never hallucinate or invent new information.
10. Never invert the meaning of a statement.
11. Never include the input html code as a part of your response.
12. Provide a list with the changes you performed and why those changes were necessary.
```

C Minimal Prompt: Detailed Results

This appendix contains the labeled results from our *minimal prompt* "Make that less manipulative" for all designs.

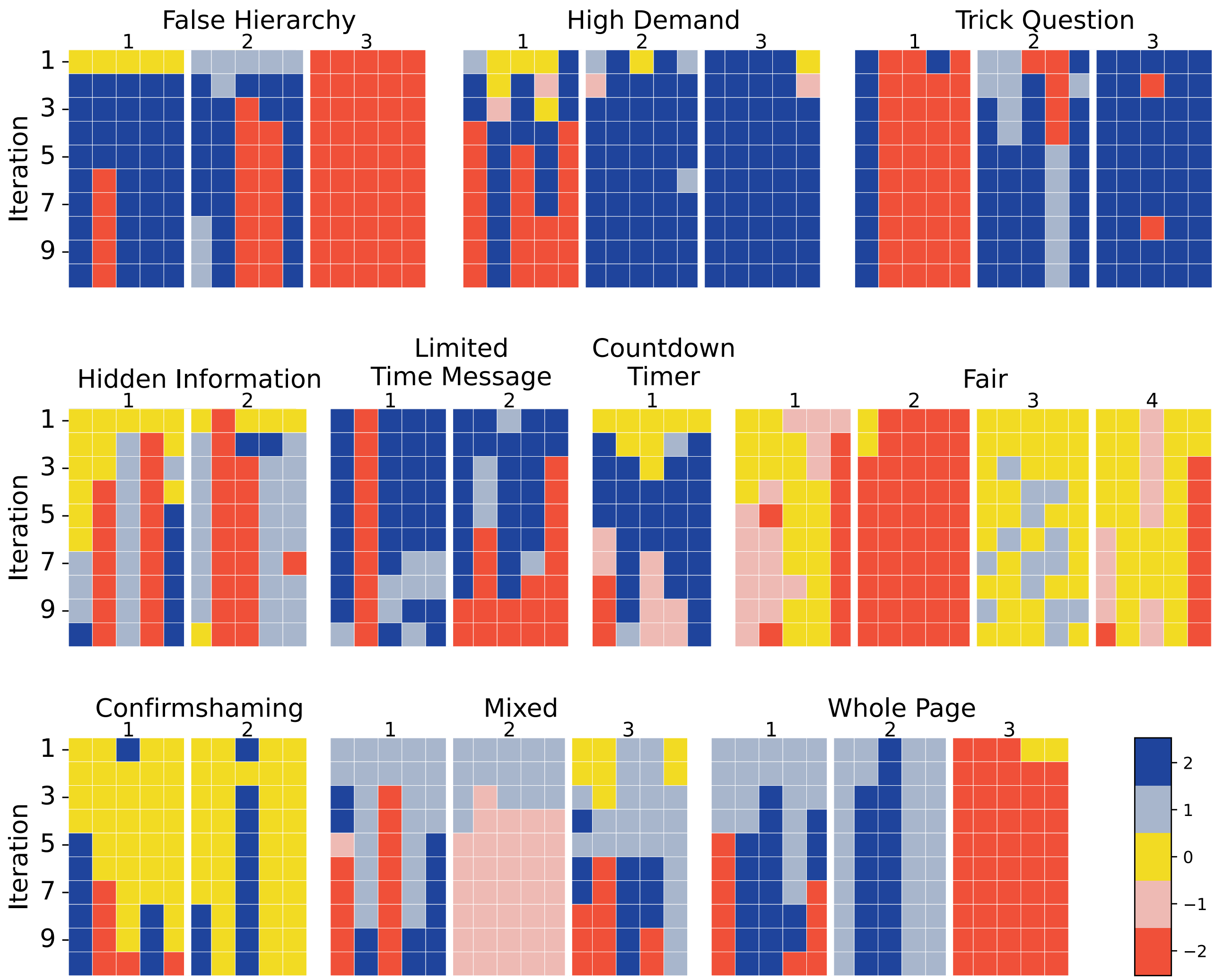

Figure 5: Quality of the redesigns using the *minimal prompt* (Section 5.1) for each iteration and each design compared to our initial input. Each block of five neighboring columns represents the five complete trials (of 10 iterations each) that were conducted for the corresponding original design. A value of 2 indicates that manipulations were fully removed, while a -2 indicates that the LLM made the redesign more manipulative, hallucinated facts and functions, or removed crucial parts of the interface. Note that for fair designs, not making the interface worse is one of the best outcomes. Those designs only received a value of 1 if they were even clearer than our input. Overall, most designs were made less manipulative in the first iterations, but later iterations frequently resulted in negative scores. Additionally, several designs were worse than the input starting from the first iteration.

D Improved Prompt Detailed Results

This appendix contains the labeled results from our *improved prompt* (Appendix B) for all designs.

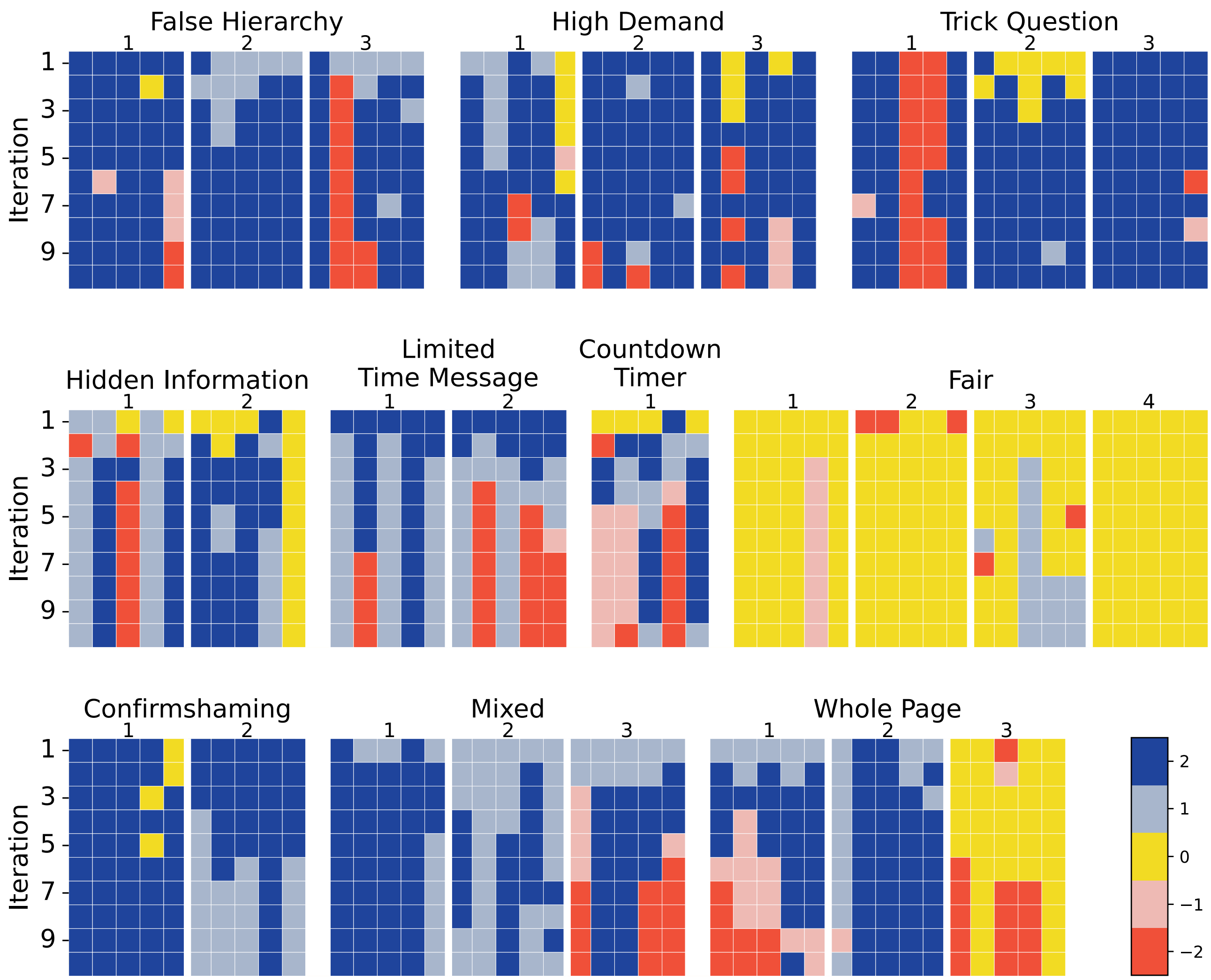

Figure 6: The changes of the redesign using the *improved prompt* (Section 5.2) for each iteration and design compared to our initial input. Each block of five neighboring columns represents the five complete trials (of 10 iterations each) that were conducted for the corresponding original design. A value of 2 indicates that manipulations were fully removed, while a -2 means that the LLM made the redesign more manipulative, hallucinated facts and functions, or removed crucial parts of the interface. For fair designs (i.e., Fair 1–4 and Whole Page 3), not making the interface worse is one of the best outcomes. Those designs only received a value of 1 if they were even clearer than our input. Overall, most manipulations were fully or at least partially removed. Fair designs mainly remained unchanged and negative scores are rarely present.