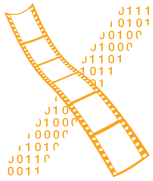

Chair for Computer Science 10 (Media Computing and Human-Computer Interaction)

RWTH AACHEN UNIVERSITY

Exploring Deceptive Pattern Removal from Websites Using LLM-as-a-Judge

Bachelor's Thesis at the
Media Computing Group
Prof. Dr. Jan Borchers
Computer Science Department
RWTH Aachen University

by
Sophie Hahn

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr.-Ing. Ulrik Schroeder

Registration date: 13.06.2025
Submission date: 30.09.2025

Zentrales Prüfungsamt/Central Examination Office

RWTH AACHEN UNIVERSITY

Eidesstattliche Versicherung
Declaration of Academic Integrity

Hahn, Sophie
Name, Vorname/Last Name, First Name

445817
Matrikelnummer (freiwillige Angabe)
Student ID Number (optional)

Ich versichere hiermit an Eides Statt, dass ich die vorliegende ~~Arbeit~~/Bachelorarbeit/ ~~Masterarbeit~~* mit dem Titel
I hereby declare under penalty of perjury that I have completed the present ~~paper~~/bachelor's thesis/~~master's thesis~~* entitled

Exploring Deceptive Pattern Removal from Websites Using LLM-as-a-Judge

selbstständig und ohne unzulässige fremde Hilfe (insbes. akademisches Ghostwriting) erbracht habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt; dies umfasst insbesondere auch Software und Dienste zur Sprach-, Text- und Medienproduktion. Ich erkläre, dass für den Fall, dass die Arbeit in unterschiedlichen Formen eingereicht wird (z.B. elektronisch, gedruckt, geplottet, auf einem Datenträger) alle eingereichten Versionen vollständig übereinstimmen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.
independently and without unauthorized assistance from third parties (in particular academic ghostwriting. I have not used any other sources or aids than those indicated; this includes in particular software and services for language, text, and media production. In the event that the work is submitted in different formats (e.g. electronically, printed, plotted, on a data carrier), I declare that all the submitted versions are fully identical. I have not previously submitted this work, either in the same or a similar form to an examination body.

Aachen, 29.09.2025
Ort, Datum/City, Date

Unterschrift/Signature

*Nichtzutreffendes bitte streichen/Please delete as appropriate

Belehrung:
Official Notification:

§ 156 StGB: Falsche Versicherung an Eides Statt
Wer vor einer zur Abnahme einer Versicherung an Eides Statt zuständigen Behörde eine solche Versicherung falsch abgibt oder unter Berufung auf eine solche Versicherung falsch aussagt, wird mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft.

§ 156 StGB (German Criminal Code): False Unsworn Declarations
Whosoever before a public authority competent to administer unsworn declarations (including Declarations of Academic Integrity) falsely submits such a declaration or falsely testifies while referring to such a declaration shall be liable to imprisonment for a term not exceeding three years or to a fine.

§ 161 StGB: Fahrlässiger Falscheid; fahrlässige falsche Versicherung an Eides Statt
(1) Wenn eine der in den §§ 154 bis 156 bezeichneten Handlungen aus Fahrlässigkeit begangen worden ist, so tritt Freiheitsstrafe bis zu einem Jahr oder Geldstrafe ein.
(2) Straflosigkeit tritt ein, wenn der Täter die falsche Angabe rechtzeitig berichtigt. Die Vorschriften des § 158 Abs. 2 und 3 gelten entsprechend.

§ 161 StGB (German Criminal Code): False Unsworn Declarations Due to Negligence
(1) If an individual commits one of the offenses listed in §§ 154 to 156 due to negligence, they are liable to imprisonment for a term not exceeding one year or to a fine.
(2) The offender shall be exempt from liability if they correct their false testimony in time. The provisions of § 158 (2) and (3) shall apply accordingly.

Die vorstehende Belehrung habe ich zur Kenntnis genommen:
I have read and understood the above official notification:

Aachen, 29.09.2025
Ort, Datum/City, Date

Unterschrift/Signature

Contents

List of Figures and Tables

Abstract

Deceptive patterns are manipulative design elements in online platforms that try to steer the user to do something that might not be in their best interest. Countermeasures against them have been proposed, and besides regulatory actions and educational measures, there are technical solutions. One interesting research direction is to directly prompt large language models (LLMs) to remove deceptive patterns from websites, which has been shown to be a promising direction, but still has room for improvement.

In our work, the goal is to expand this idea and add a second LLM as a judge, building upon the concept of "LLM-as-a-Judge". We utilized this in a way that the LLM judge gives feedback about the altered websites and controls the iterative process. We then tested different adjustment methods to refine this pipeline, such as varying model combinations or prompting strategies. With a final pipeline, we answered the following research questions: 1. How does adding an additional LLM as a judge influence the iterative LLM-based removal of deceptive patterns on websites? 2. Does the LLM-as-a-Judge approach align with the judgment of users? 3. How do people perceive the changes made by our LLM-as-a-Judge approach?

To answer these questions, we first compared the results of our final LLM-as-a-Judge pipeline to a baseline without LLM-as-a-Judge. We observed that the former improved the amount of removed deceptive patterns, as well as kept the information and design more consistent. Then we conducted a user study with 15 participants, in which they were tasked to modify the web pages themselves, and rate the alteration our LLM pipeline made. Results show that users generally changed the web pages differently from our LLM pipeline, varying in the agreement between different deceptive pattern types. However, they mostly preferred the altered web pages over the original. Our work helps to gain insight into how LLM-as-a-Judge can be used for iterative deceptive pattern removal, and where it might need further adjustment to align more with user preferences.

Überblick

Deceptive Patterns sind manipulative Elemente in online Plattformen, die Nutzer dazu bewegen sollen etwas zu tun, das nicht unbedingt in ihrem eigenen Interesse ist. Gegenmaßnahmen wurden vorgeschlagen, und neben Gesetzen und Aufklärung, gibt es technische Lösungen. Eine Möglichkeit ist es, Large Language Models (LLMs) die Aufgabe zu geben, Deceptive Patterns direkt von Webseiten zu entfernen, was vielversprechend erscheint, aber noch verbesserungsfähig ist.

Das Ziel unserer Arbeit ist es diese Idee weiterzuführen und ein zweites LLM als einen Bewerter hinzuzufügen. Das Konzept ist als "LLM-as-a-Judge" bekannt. Wir nutzen es so, dass der Bewerter Feedback zu den geänderten Webseiten gibt und den iterativen Prozess kontrolliert. Wir testen verschiedene Anpassungen dafür, wie diverse Modelle und Prompting Strategien. Mit einer finalen Implementierung beantworten wie die folgenden Forschungsfragen: 1. Wie verändert das Hinzufügen eines zweiten LLMs als Bewerter das iterative Entfernen von Deceptive Patterns von Webseiten? 2. Stimmt der LLM-as-a-Judge Ansatz mit den Meinungen von Nutzern überein? 3. Wie nehmen Personen die Veränderungen wahr, die von unseren LLMs durchgeführt wurden?

Um diese Fragen zu beantworten vergleichen wir zuerst die Ergebnisse, die mit und ohne LLM-as-a-Judge erzielt wurden. Unsere Ergebnisse zeigen, dass LLM-as-a-Judge besonders die Anzahl der entfernten Manipulationen steigert, während Informationen und Design öfter gleich geblieben sind. Danach haben wir eine Nutzerstudie mit 15 Teilnehmern durchgeführt, in welcher diese die Webseiten selber weniger manipulativ machen sollten und die Veränderungen bewerten sollten die wir mit LLM-as-a-Judge erzielt haben. Wir haben herausgefunden, dass Nutzer im Allgemeinen die Webseiten anders verändern würden, als es die LLMs gemacht haben, vor allem durch andere Deceptive Patterns die sie entfernen. Trotzdem haben Nutzer die veränderten Seiten den Originalen vorgezogen. Unsere Arbeit hilft dabei, Einblicke zu bekommen, wie LLM-as-a-Judge für das iterative Entfernen von Deceptive Patterns genutzt werden kann und wie es weiter verändert werden könnte, um mehr mit Nutzern übereinzustimmen.

Acknowledgments

First of all, I would like to thank Prof. Dr. Jan Brochers and Prof. Dr.-Ing. Ulrik Schroeder for examining my thesis.

Secondly, many thanks to my advisor René Schäfer for his continuing support, motivation, and patience. Your feedback and assistance from far before I was even set on any topic, up until the last day, was incredibly valuable, and helped as well as taught me a lot.

Lastly, a big thanks to everyone who took the time to participate in my user study. Without your effort, a huge part of this work would not have been possible.

Conventions

Throughout this thesis we use the following conventions:

- The thesis is written in American English.
- The first person is written in plural form.
- Unidentified third persons are referred to in the plural form.

Short definitions are set off in colored boxes.

DEFINITIONS:
Definitions are set off in orange boxes.

Where appropriate, paragraphs are summarized by one or two sentences that are positioned at the margin of the page.

This is a summary of a paragraph.

Source code and implementation symbols are written in `typewriter-style text`.

In this thesis, we use the terms *dark pattern*, *deceptive pattern*, and *deceptive design* synonymously.

We start web page names with caps and leave out the top-level domain when we talk about the item in our dataset, we also write them in *italic*.

Evaluation criteria are written in SMALL CAPS, names of adjustment types in Chapter 3, and deceptive pattern types in

Chapter 4 are written in ITALIC. For readability, deceptive pattern types in Chapter 3 are not in *italic*.

Chapter 1

Introduction

Companies regularly try to manipulate their customers, which can lead to, for example, financial harm [Mathur et al., 2021]. This is evident in different ways. For example, it is visible in airports, in which you often have to go through stores to get to your gate [Brignull, 2023]. With the ever-growing internet, many websites have emerged in various domains [Holzmann et al., 2016], and, unsurprisingly, companies also try to increase their sales or gather user data in the online world [Lupiáñez-Villanueva et al., 2022]. These manipulative online practices are known as *deceptive patterns*.

1.1 Deceptive Patterns

DECEPTIVE PATTERNS:
Deceptive patterns (DPs) are design elements in online platforms that are intended to manipulate the user in a way that might not be in their best interest [Brignull, 2023].

Definition: Deceptive Patterns

While the basis of deceptive patterns has developed over many decades [Narayanan et al., 2020], in 2010 Brignull [2023] introduced the now widely-used term *"dark*

patterns", which has since been replaced with *"deceptive patterns"*. Deceptive patterns manipulate the user to act in favor of the service owner and not in favor of the user. This can be financial loss, privacy invasion, or mental burden [Mathur et al., 2021]. While users are generally aware that they can be manipulated online, this does not increase their resilience [Bongard-Blanchy et al., 2021], and they often still fail to even detect deceptive patterns [Bhoot et al., 2020]. Subsequently, the effectiveness of deceptive patterns has been proven [Luguri and Strahilevitz, 2021].

There are various negative effects of deceptive patterns.

Next to raising awareness and calling out online platforms on their website, Brignull [2023] created a taxonomy to differentiate between various types. Since then, many new taxonomies have been developed [e.g., Gray et al., 2018; Mathur et al., 2019]. A recent ontology from Gray et al. [2024] consists of 65 types, showing the wide variety of patterns. Given this diversity, it is not surprising that deceptive patterns are prominent in many platforms and domains, such as shopping websites [Mathur et al., 2019], social media [Mildner et al., 2023], and games [Niknejad et al., 2024]. For example, Di Geronimo et al. [2020] found one or more patterns in around 95% of 240 analyzed mobile apps in the Google Play Store.

Deceptive patterns have been defined in various taxonomies and appear in a wide variety online.

Given the high prevalence of deceptive patterns and the effectiveness of the harm they can cause, countermeasures against them have been suggested and explored. These include enhancing the resistance of users [Bongard-Blanchy et al., 2021], enforcing laws [Gray et al., 2021], or implementing technical regulations [Schäfer et al., 2025]. Focusing on the last option, most technical countermeasures need to detect deceptive patterns first, and then implement a countermeasure in a specified way, such as highlighting or removing the patterns [Schäfer et al., 2023]. Multiple possible ways to detect deceptive patterns have been suggested, such as examining the CSS code [Hausner and Gertz, 2021] or using machine learning [Hasan Mansur et al., 2023; Soe et al., 2022]. Overall, current detection methods still cannot detect all types [Nie et al., 2024], and some types might not be automatically detectable at all, for example, due to too much variation in the pattern type [Curley et al., 2021]. Additionally, drawbacks when using machine learning in-

Researchers propose different countermeasures against deceptive patterns. Most technical ones need a way to automatically detect them.

clude that the model has to be trained accordingly [Soe et al., 2022].

1.2 Large Language Models and Deceptive Patterns

Large language models (LLMs) are largely pre-trained [Chang et al., 2025] and can be further adjusted through different prompting techniques and fine-tuning [Gu et al., 2024]. Multiple researchers evaluated using LLMs to detect deceptive patterns [e.g. Mills and Whittle, 2023; Sazid et al., 2023]. While this yielded promising results, every detection approach needs an additional implementation of the actual countermeasure. Schäfer et al. [2025] skipped the explicit detection by evaluating how well LLMs can directly mitigate deceptive patterns. To test this, they simply provided GPT-4o with HTML code and prompted it to remove the manipulation. Using this initial prompt, they then optimized it with guardrails based on mistakes the LLM made with the initial prompt, coming out with an improved prompt. This yielded promising results. However, the LLM still made mistakes, even with the improved prompt, such as removing or hallucinating information. Overall, this approach resulted in 72% of all web elements being considered a full success, i.e., all manipulation is removed and the website was not made worse in any way. As a conclusion, further optimization is needed. It is also important to note that their test set consisted mainly of self-made, smaller web pages and preliminary tests for real web elements.

Schäfer et al. [2025] proposed to utilize an LLM to automatically remove deceptive patterns from websites, yielding promising results.

LLM-AS-A-JUDGE:
LLM-as-a-Judge is defined as using a Large-Language-Model (LLM) to evaluate something, possibly based on defined rules. The LLM replaces human experts or statistical metrics [Gu et al., 2024; Li et al., 2024].

Definition: LLM-as-a-Judge

One direction to improve this approach could be to incorporate LLM-as-a-Judge, meaning we use a second LLM as an evaluator, to check the output of the generator LLM.

LLM-as-a-Judge can be applied in different ways, and has various advantages.

LLM-as-a-Judge is already applied in production, but is also a widespread research area that is applicable in many ways [Huyen, 2025]. Possible ways to implement LLM-as-a-Judge include asking the LLM to decide which of two options is better or fits a specified criterion more, ask it to score a single input based on criteria [Li et al., 2024; Zheng et al., 2023], or use the judge to iteratively improve something [Vasudevan et al., 2025]. Using an LLM as an evaluator has multiple advantages, such as being relatively cheap and fast, compared to human evaluators [Huyen, 2025]. LLMs can also offer explanations, if needed, and diminish the need for humans to be involved in the loop [Zheng et al., 2023].

LLM-as-a-Judge is a promising way to improve the removal of deceptive patterns using an LLM.

LLM-as-a-Judge is an interesting idea for adjusting the process of removing deceptive patterns, especially since it reduces the need for humans to manually check each removed pattern, and thus also hopefully improves the quality of the resulting websites. Additionally, the judge could decide when to end the iterative removal, which was an aspect still open in Schäfer et al.'s approach.

We explore and adjust LLM-as-a-Judge for the iterative deceptive pattern removal.

We aim to explore how adding an additional LLM as an evaluator in iterative deceptive pattern removal affects the results and success rate, building on Schäfer et al.'s promising results. Based on the literature surrounding LLM-as-a-Judge, we aim to improve our approach by testing different strategies for adjusting the evaluator and generator, as it is not obvious what the optimal prompt and setup are [Li et al., 2024; Gu et al., 2024]. We then compare our improved pipeline to the approach without LLM-as-a-Judge.

We conduct a user study to evaluate LLM-as-a-Judge further.

Additionally, we plan to see how much the LLM evaluator aligns with the judgment of users. This is especially important since it is not self-evident how to remove deceptive patterns [de Jonge et al., 2025]. Next to that, we also evaluate how users perceive the changes, to not only draw a comparison with the users' optimal solution, but also understand their opinion and attitude towards our results. For all this, we conduct a user study. This leads us to the following three research questions:

RQ1: How does adding an additional LLM as a judge influence the iterative LLM-based removal of deceptive patterns on websites?

RQ2: Does the LLM-as-a-Judge approach align with the judgment of users?

RQ3: How do people perceive the changes made by our LLM-as-a-Judge approach?

Overall, this thesis aims to explore utilizing the LLM-as-a-Judge approach as a countermeasure to remove deceptive patterns from websites.

1.3 Outline

In Chapter 2, we take a closer look at related literature in the fields of deceptive patterns and LLM-as-a-Judge.

In Chapter 3, we describe all the considerations we made when implementing LLM-as-a-Judge to remove deceptive patterns from websites. We then go into detail describing each adjustment we made and the results we got when iteratively refining our implementation. Finally, we also discuss the adjustments we made here.

After the technical realization of LLM-as-a-Judge, in Chapter 4, we describe all deliberations that went into the design of our user study, explain the final design, and present the results.

We then discuss our results from the technical evaluation and the user study in Chapter 5, drawing connections between all parts and further implications. Lastly, we talk about limitations.

Finally, in Chapter 6, we end with a summary of our work and possible future work.

Chapter 2

Related Work

In this chapter, we provide an overview of existing literature relevant to our work. We first delve into the subject of deceptive patterns by looking into their prevalence, existing taxonomies, the impact of deceptive patterns, and current countermeasure approaches. Following this, we present different research topics in the field of large language models (LLMs). In particular, we briefly explore how LLMs have been utilized to counter deceptive patterns, how they have been connected to deception in general, and lastly, we discuss the field of LLM-as-a-Judge.

2.1 Deceptive Patterns

The topic of deceptive patterns first emerged back in 2010, when Brignull [2023] introduced it under the name *"dark pattern"*. In this work, we mainly use the term *"deceptive pattern"*, but perceive them synonymously[1]. They are generally defined as user interface designs that try to manipulate the user into doing something that might not be in their best interest, instead benefiting the service provider [Mathur et al., 2019].

Deceptive patterns manipulate the user.

[1] `https://www.acm.org/diversity-inclusion/words-matter` [Accessed: Sep. 27, 2025]

There is a high prevalence of deceptive patterns in websites and apps.

The prevalence of deceptive patterns has been studied across multiple domains and platforms, such as websites [Mathur et al., 2019], apps [Di Geronimo et al., 2020], or games [Niknejad et al., 2024]. Di Geronimo et al. [2020] studied deceptive patterns in 240 popular mobile apps on the Google Play Store by actively interacting with the applications. They then identified them in 95% of cases, averaging 7.5 instances per app. Similarly, across 200 Japanese apps, 93.5% included deceptive patterns, with a lower average of 3.9 occurrences per app [Hidaka et al., 2023].

Deceptive patterns are also ubiquitous in websites.

Furthermore, deceptive patterns are also prominent on websites, again not limited to one domain, but instead can be found on shopping websites [Mathur et al., 2019], health and fitness, education [Rahman and Adaji, 2024], and cookie banners [Nouwens et al., 2020]. Mathur et al. [2019] automatically collected a sample of 11K shopping websites using a web crawler, identifying one or more deceptive patterns in 11.1% of them. As they could only analyze text-based interfaces, this is only a lower bound. They also stated that popular websites have a higher likelihood of containing deceptive patterns than less popular ones. It is important to note that deceptive patterns not only occur as singular instances, but instead can appear in combination with other variants, also spanning across multiple pages of a website [Gray et al., 2025]. More recently, Shi et al. [2025] developed an approach to automatically detect deceptive patterns utilizing multimodal large language models. In a dataset containing screenshots of 2000 websites and mobile apps, they identified one or more deceptive patterns in 25.7% of mobile apps and 49% of websites. The varying results in research around the prominence of deceptive patterns in apps and websites can be explained by the different datasets and analysis approaches used, but all showcase the rather high prominence of deceptive patterns in those platforms.

2.1.1 Taxonomies

As deceptive patterns are so common online, it is natural that different types exist, varying, for example, in their

purpose or how they manipulate the user. This makes it a prominent research topic in the literature, and many researchers have published various taxonomies, differing in their focus or underlying dataset [Mathur et al., 2019; Gray et al., 2024; Conti and Sobiesk, 2010].

Even before the term "dark pattern" was introduced by Brignull [2023] in 2010, Conti and Sobiesk [2010] published a taxonomy earlier in that year in the context of malicious interface design. They identified types over a year-long study and validated them in a user study. This resulted in 11 categories, which were then split into more detailed subcategories. Brignull [2023] also published an initial taxonomy containing 11 types. Over the years, the taxonomy has been adapted, and the most recent version was published in 2023[2]. It now contains 16 patterns and incorporates only a handful of the original terms, such as *Disguised Ad* or *Hidden Costs*.

Conti and Sobiesk's and Brignull [2023]'s taxonomies were the first ones published back in 2010.

More recently, Gray et al. [2024] developed a new ontology that aims to establish a shared language in a field that contains many variations, with the goal of connecting different research areas. The work is based on multiple preceding research papers, including their previous work from 2018, as well as reports from regulators and stakeholders, and lastly the taxonomy from Brignull [2023]. They outlined a pattern hierarchy of high-level, meso-level, and low-level patterns, resulting in a total of 65 patterns across all levels. While high-level patterns describe general strategies, meso-level ones describe an angle of attack, and low-level ones outline specific means of execution. High-level patterns include, similarly to Gray et al. [2018], *Obstruction*, *Sneaking*, *Interface Interference*, and *Forced Action*, as well as the new pattern *Social Engineering*. The definitions of each deceptive pattern relevant to this work can be found in Appendix A.

The ontology of Gray et al. [2024] contains high-level, meso-level, and low-level patterns.

Gray et al. [2024] noted that their ontology is not exhaustive and should be expanded in the future. Overall, implying that deceptive patterns are likely to evolve over the years and new types may develop [Conti and Sobiesk, 2010]. All this calls for room for taxonomies to adjust, develop, and new ones to originate [Gray et al., 2024], which is impor-

New deceptive pattern types are likely to develop over time, and taxonomies will further evolve.

[2] `https://www.deceptive.design/` [Accessed: Sep. 30, 2025]

tant to keep in mind when working with taxonomies and deceptive patterns.

All taxonomies above have in common that they give a general overview of the deceptive pattern landscape. However, over the years, multiple other variations have been proposed. Some taxonomies are centered around specific domains or particular areas in which deceptive patterns can appear, such as social networking services [Mildner et al., 2023], games [Zagal et al., 2013], or augmented and virtual reality [Krauß et al., 2024]. Next to showing the prevalence of deceptive patterns on shopping websites, Mathur et al. [2019] also analyzed patterns and concluded seven high- and 15 low-level ones, basing their terms partly on the work of Gray et al. [2018] and Brignull [2023]. A slightly varying approach differentiates deceptive pattern types with a specific target in mind. Shi et al. [2025] adapted and refined existing taxonomies to a version that is more applicable in the context of the detection of deceptive patterns, eliminating oversights in previous versions regarding security- and privacy-related examples.

Next to the general taxonomies, there are also specific ones for different domains or targets.

2.1.2 Effect on and Relation to Users

Deceptive patterns can cause harm to users in different ways, such as producing financial damage through manipulation on shopping or travel websites, or violating their data privacy, but also undermining their autonomy and decision process [Mathur et al., 2021]. An exemplary study was conducted by Sin et al. [2025]. They used shopping websites with different deceptive patterns, determining that all of the ones they tested increase purchase impulsivity, thus generating financial harm.

Deceptive patterns can cause financial harm, and invade users' privacy and autonomy.

Distinguishing between mild and aggressive deceptive patterns, Luguri and Strahilevitz [2021] conducted a study in which users interacted with a fictitious website and were exposed to either no, mild, or aggressive deceptive patterns, which should nudge the user to subscribe to a data protection program. The results showed that, on one side, websites with mild deceptive patterns did more than dou-

Luguri and Strahilevitz [2021] showed that aggressive deceptive patterns work better than mild ones, but also result in more backlash.

ble the acceptance rate for the subscription. On the other side, they showed that aggressive patterns raised the acceptance rate to 41.9% in comparison to mild ones with an acceptance rate of 25.8%. However, users exposed to the aggressive version were more upset afterwards, which led to more backlash, possibly affecting the company in the long term.

The ability to detect deceptive patterns varies for the pattern types.

As deceptive patterns are so prominent and have a noticeable effect on users, the question arises of how well users can detect these manipulations and how aware they are. Di Geronimo et al. [2020] organized an online study with 589 participants, in which they were exposed to deceptive and fair designs from apps. They were then asked whether they noticed any manipulation in those designs. The authors concluded that their participants generally could not detect deceptive patterns. Further, Bhoot et al. [2020] found that the detection varies between different deceptive pattern types. For example, deceptive patterns such as *Confirmshaming* were more often identified than *Trick Question*. Further, Bongard-Blanchy et al. [2021] conducted a user study and reported that their participants were generally aware of the manipulation and were able to detect it. However, they tended to be less worried about the harm to themselves than about potential harm to others. Additionally, they reported that awareness did not significantly improve the ability to withstand deceptive patterns.

Awareness does not necessarily increase resilience against deceptive patterns.

Deceptive patterns can yield negative emotions from the users.

Users get frustrated when interacting with malicious interface designs, and the tolerance of the designs varies between different domains, with, for example, a higher tolerance for shopping websites, but a lower tolerance for news websites [Conti and Sobiesk, 2010]. Next to frustration, users also feel anger for various deceptive patterns, but also indifference for others [Avolicino et al., 2022], and, overall, Seaborn et al. [2024] reported that participants' mood degraded after interacting with a website containing multiple deceptive patterns.

2.1.3 Countermeasures

Researchers advocate for countermeasures.

As it is clear that deceptive patterns are prominent online in diverse variations and have an apparent effect on users, researchers have advocated for different countermeasures against them. Multiple approaches have been suggested, explored, and partly set up, such as legal regulations, educational strategies, or technical implementations [Bongard-Blanchy et al., 2021].

Some legal regulations have been developed, but are not sufficiently enforced yet.

Legal regulations are already enforced to some extent. A few examples include the CCPA (California Consumer Privacy Act) in the United States, as well as the GDPR (General Data Protection Regulation) and the DSA (Digital Services Act) in the European Union [Gray et al., 2021; Ahuja et al., 2025]. The GDPR and CCPA cover rights regarding one's personal data, while the DSA, explicitly prohibits deception and manipulation in general in online platforms [Löbel et al., 2024]. However, not all websites meet these legislations [Narayanan et al., 2020]. Krisam et al. [2021] unfolded that, in their set of 389 German websites, around 17.7% did not comply with EU laws, for example, by deploying an opt-out design prohibited under the GDPR. Researchers assist in enforcing existing laws in multiple ways. One example is demonstrating how many websites employ deceptive patterns in specific website parts [Löbel et al., 2024; Nouwens et al., 2020]. Another example is Ahuja et al. [2025], who mapped pattern types to violation types in the DSA.

Another approach is to implement fair or bright patterns instead.

A different proposal for countermeasures would be to start at the designer and advocate for them to implement *fair patterns* or *bright patterns* instead of deceptive ones. *Fair patterns* are designed neutrally to not influence the user in any way [Potel-Saville and Francois, 2023]. Here, the problem arises in how to define what is fair, and that fairness depends on the context [de Jonge et al., 2025]. In contrast to fair patterns, *bright patterns* prioritize users' goals over business goals, but still use manipulation for this [Sandhaus, 2023].

Educational approaches include raising the awareness of users in a way that they can detect deceptive patterns themselves and thus, possibly resist them [Bongard-Blanchy et al., 2021]. One way to do this is through games, which have been proposed in different variations [e.g., Aung et al., 2024; Fiedler et al., 2025; Kronhardt et al., 2024]. One example was explored by Fiedler et al. [2025], who implemented a detection and classification web-based serious game. They observed in multiple user studies that participants improved in detecting deceptive patterns after playing. A different approach to educating users was introduced by Ye et al. [2025]. They suggested an experiential learning platform that incorporates multiple simulated real-world scenarios surrounding deceptive patterns, enhancing the detection by users and the way they deal with them.

Raising awareness can be done through games or learning platforms.

A different concept that aims to raise user awareness, but also empower them through technical support directly on websites, was proposed by Lu et al. [2024]. They suggested a browser extension with an awareness panel, designed to raise awareness and educate the user. They also included an action panel, offering different options, such as hiding the manipulation or adding a pop-up in the interaction with elements containing deceptive patterns.

Lu et al. [2024] developed a browser extension to raise awareness and give autonomy to the user.

Following the direction of technical countermeasures, visual countermeasures are a suggestion on how to handle deceptive pattern instances once they are detected. Options here include removing them, offering an option to switch between the removed and original version, or highlighting instances of deceptive patterns and offering an explanation for them. Users preferred the removal for cases where options are not displayed equally and the highlighting for ones that could lead to financial loss [Schäfer et al., 2024]. The removal option corresponds closely with the fair patterns discussed above.

Visual countermeasures can either be to remove or highlight deceptive patterns, which yielded different preferences from users.

For most of these approaches to act automatically, we need a way to detect deceptive patterns in applications. Mathur et al. [2019] was one of the first to do this in their research. They utilized a semi-automatic crawler that detects text-based patterns in websites. Hausner and Gertz [2021] sug-

For most technical countermeasures, we need to detect them first. This can, for example, be done through machine learning.

gested the detection of deceptive patterns in cookie banners through analyzing similarities in the CSS of buttons, visualizing the detected instances via a box surrounding them. Multiple researchers suggested utilizing machine learning for detection [e.g., Soe et al., 2022; Vedhapriyavadhana et al., 2025]. Soe et al. [2022] proposed this specifically for cookie banners to detect and classify deceptive patterns. While their approach lacks high accuracy, it shows potential. Utilizing computer vision and natural language pattern matching, Chen et al. [2023] then developed a system for mobile applications that they named *UIGuard*. It takes a screenshot as input and then detects and classifies deceptive patterns and highlights them. While all these approaches seem promising, Nie et al. [2024] examined five existing detection technologies and found a coverage of merely 50% of all pattern types detected across all five tools. Similarly, it is important to note that some deceptive pattern types might not be detectable at all [Curley et al., 2021]. Lastly, one disadvantage is that machine learning models have to be trained on datasets, while LLMs can already be pre-trained [Chang et al., 2025].

Current detection methods are not optimal yet, showing some disadvantages.

2.2 Large Language Models (LLMs)

Large language models are neural networks that encompass countless layers and parameters, and are pre-trained on very large datasets [Shao et al., 2024]. They have the ability to generate human-like outputs [Demszky et al., 2023], and in today's standard, apart from text generation, they are also able to incorporate images and videos, or generate code [Shao et al., 2024].

2.2.1 LLMs to Counter Deceptive Patterns

Independent of detection, Porcelli et al. [2024] suggested using LLMs to automatically answer cookie banners based on users' set preferences. The LLMs, specifically versions of GPTs, were used to generate JavaScript Code that was then used to answer the banners accordingly.

Sazid et al. [2023] prompted GPT-3 to detect and classify deceptive patterns in text snippets. They compared Zero-, One-, and Few-Phot prompting, in which no, one, or multiple examples are provided, respectively. Results showed that Few-Shot prompting yielded the highest success rate of 92.57%, and that the pattern *Sneaking* was the only one out of seven types not detectable by GPT-3. Going further, Kodmurgi et al. [2024] proposed a detection tool supposed to cover a wider variety of pattern types. However, to achieve this, they decided to use LLMs only for textual patterns, and for visual cues, they utilized a deep learning model. The LLM was given text that was extracted from HTML code and showed promising results. However, due to advances in LLMs' capabilities, they also suggested giving them screenshots. Similarly, going beyond only text snippets, Mills and Whittle [2023] evaluated three different input methods for detection: textual descriptions, screenshots, and HTML together with JavaScript code. The first two options are not objective, since humans need to either write the text or gather the screenshots, while the last option might lead to the LLM not having a clear enough picture of how the website actually looks, along with difficulties in handling the large HTML code some websites have. Overall, they identified screenshots as input as the most promising approach.

LLMs have been utilized to detect and classify deceptive patterns in various ways.

Mills and Whittle [2023] also gave the LLM HTML as input, but deemed screenshots as more promising.

More recently, Kocyigit et al. [2025] proposed *DeceptiLens*, which is based on a Multimodal LLM (MM-LLM), specifically GPT-4o, and used for detection and classification. Next to the UI screenshot, they included multiple prompting strategies such as Chain-of-Thought and a step-by-step analysis. To then evaluate their approach, they conducted a study with experts in the field of deceptive patterns. The study revealed that the tool has an accuracy of 90.54% when compared to the majority voting of the experts.

Kocyigit et al. [2025] prompted GPT-4o to detect and classify deceptive patterns, showing high agreement between the LLM and experts.

Surpassing detection, Schäfer et al. [2025] explored LLMs to directly remove deceptive patterns from websites. They used an iterative process, providing GPT-4o with the original HTML in the first iteration, and from the second iteration on, they used the output from the iteration before as the new input, ending after iteration ten. They started with a minimal, rather naive prompt *"Make that less manipula-*

Schäfer et al. [2025] prompted GPT-4o to remove deceptive patterns from websites. While showing promising results, the LLM still made mistakes.

tive" and used the results to adjust the prompt by adding guardrails based on common mistakes GPT-4o made, such as hallucinating or removing important elements. Their results showed that the most successful iteration was three, and that this is a promising approach, as they achieved a success rate of 72% with the improved prompt, which is higher than the 45% with the minimal prompt. However, it is important to note that there is room for improvement and further exploration, as the LLM still made mistakes, and the dataset used for testing did not contain real websites.

2.2.2 Deception in LLMs

LLMs also utilize deceptive patterns to manipulate the user.

While LLMs have been utilized to counter deceptive patterns, it is essential to keep in mind that LLMs themselves are also prone to including deceptive patterns in their output [Benharrak et al., 2024; Krauß et al., 2024]. For one, researchers have defined specific types, or transferred existing ones into the context of conversation with LLMs [Benharrak et al., 2024; Kran et al., 2025]. An example is *Brand Bias*, in which the model emphasizes products from their own company more positively [Kran et al., 2025].

LLMs include deceptive patterns in websites they generate, even without being specifically prompted for this.

Other studies showed the prominence of deceptive patterns in LLM-generated websites. Krauß et al. [2025] conducted a user study, asking participants to generate e-commerce websites using ChatGPT. Even though they used neutral prompts, every result contained one or more deceptive patterns, most often patterns from the high-level category *Social Engineering* or the low-level category *Visual Prominence*. They noted that warnings from the LLM regarding the included manipulations were missing. A preliminary study, in which they tested Claude 3.5 Sonnet and Gemini 1.5 flash, showed similar tendencies to include deceptive patterns in their output. Similarly, Chen et al. [2025] analyzed the output of four different LLMs and found 37% of the resulting components contained deceptive patterns. In their study, they also noticed different amounts of patterns included in different website component types.

2.2.3 LLM-as-a-Judge

There are multiple ways to evaluate an LLM. One option is to use benchmarks that evaluate specific pre-defined tasks [Hu et al., 2025]. A more obvious one is to have humans as evaluators. However, humans, especially experts in one field, can be expensive, too slow, and not always accessible [Huyen, 2025]. Utilizing LLMs for evaluation is known as LLM-as-a-Judge [Gu et al., 2024; Huyen, 2025]. Employing LLMs often has the advantage of reducing the need for humans to do these evaluations themselves [Gu et al., 2024]. They often overcome human disadvantages by being cheaper, faster, and working more flexibly [Huyen, 2025; Li et al., 2024]. LLM-as-a-Judge can and has already been applied in various domains. For example, in the medical field, software engineering, or in the legal field [Li et al., 2024; Wang et al., 2025]. There are different options to employ it, such as having the LLM compare pairs or multiple items and select the best one, or grade a single item; both options can be based on specific criteria [Li et al., 2024; Zheng et al., 2023].

Utilizing an LLM to evaluate another LLM has advantages in costs, speed, and flexibility.

LLM-as-a-Judge can be applied in different domains and in different forms.

Generally, researchers often compare their results with human judgment to analyze how well the LLM judge performs [Fabbri et al., 2025; Szymanski et al., 2025], and overall show that they can achieve high agreement rates and thus good performances [Gu et al., 2024; Zheng et al., 2023]. GPT-4 is specifically known to show high agreement with humans, and is widely employed and used as a judge [Gu et al., 2024; Szymanski et al., 2025]. Exemplary, Fabbri et al. [2025] performed a user study to test how well an LLM can extract users' podcast preferences based on user data, concluding promising performance results based on great alignment with humans.

LLM-as-a-Judge is often evaluated through the comparison with humans judgment.

There are approaches that use LLM-as-a-Judge in an iterative setting. One suggestion is to let LLMs self-correct by providing feedback for their own output, which they then apply themselves as well [Madaan et al., 2023]. However, the self-correction is controversial. For example, researchers state that LLMs cannot self-correct reasoning, and that this often even decreases the performance

LLM-as-a-Judge can also be applied in iterative refinements. It is recommended to do this using different LLMs.

[Huang et al., 2023]. Similarly, for LLM-as-a-Judge, it is suggested to use two different LLMs to bypass a self-bias, which means that LLMs favor their own output [Gu et al., 2024; Huyen, 2025]. Xu et al. [2024] utilized an iterative pipeline generalizable to different fields. They, for example, tested translation and summarization tasks. The judge produces feedback, which is then applied by another LLM, refining the input. Similarly, Vasudevan et al. [2025] applied an approach to evaluate LLM-generated marketing messages. As they had multiple quality criteria, they employed one LLM judge for each criterion. When a judge decided that their criteria failed, they provided feedback, which was then given back to another LLM, who adjusted the message. This approach increased the successful generation of these marketing messages.

LLM-as-a-Judge has its limitations, and not every task is fitting for this approach.

However, LLM-as-a-Judge does not always work well, especially in more expert-level domains or specific tasks. Szymanski et al. [2025] used GPT-4 to evaluate expert knowledge tasks, specifically to evaluate the quality of the responses from two different LLMs in the field of dietetics and mental health. They then compared the results with responses of experts and novices to the same question, with the latter receiving slightly higher agreement scores than the former. Finally, they concluded that human experts are necessary in evaluation processes in expert fields. Similarly, Wang et al. [2025] studied LLM-as-a-Judge in software engineering, finding that the performance varies across different tasks. For example, while the LLM evaluator performed better in evaluation for code generation, it lacked in tasks surrounding code summarization.

Chapter 3

Refinement of the LLM-as-a-Judge Pipeline

In this chapter, we describe our pipeline between the LLM judge and the generator, and all considerations that went into implementing and iteratively refining it. We present the results of each adjustment that we made in our refinement process, compare the final pipeline to a baseline, and then discuss the adjustments.

3.1 Method

We now go into all deliberations for our dataset, pipeline, and iterative refinement.

3.1.1 Dataset for Evaluation

To evaluate each prompt and each variation in the pipeline, a dataset consisting of web pages containing different deceptive patterns is needed. Additionally, we included web pages with no deceptive patterns, called fair web pages, to see how the LLM acts when no manipulation is present. We

also wanted to include a deceptive pattern that likely cannot be removed by an LLM.

We mainly wanted to include real websites in our dataset.

We decided to mainly use parts of real websites, as this will be the actual use case of such an application. To find websites applicable to our use case, we used different strategies. For example, browsing Brignull's *Hall of Shame*[1], and using Google Chrome's search results that came up when searching for broad website terms, such as "shopping websites". The ambition was to find a wide variety of deceptive patterns in different scenarios, but still making sure that the patterns included should be removable by altering the HTML. This is, more difficult with patterns across different temporal levels [Gray et al., 2025], for example. The search was performed from within the European Union, but each website was set to English, and the location within the website was often set to the USA. All web pages were collected in June 2025.

Due to token limits, we only included two real web pages. We had to manually reduce the token amount in them.

To get closer to actual scenarios, we wanted to include whole web pages. However, due to token costs in many LLMs, we had a loose limit of 50K/min input tokens for GPT-4o and 500K/min for o4-mini, thus we tried to keep all web pages smaller than 50K tokens. To validate this for each web page, we used OpenAI's Tokenizer[2]. As a result, we decided to use only two large web pages, and had to adjust both of them by removing parts to fit the token limit. The two resulting web pages are: a ticket page from *viagogo.com* and an overview of hotels in *booking.com*. To drastically cut down the size, we removed the `<script>` tags from *Booking*, as they took up too large a portion of its tokens. Additionally, we removed all links in the `href` attributes. For both web pages, we further removed some elements, such as repeating, similar-looking hotel listings with deceptive patterns already included in other listings, as well as parts of the header and footer for *Booking*, and text in *Viagogo*.

[1] `https://www.deceptive.design/hall-of-shame` [Accessed: Sep. 30, 2025]

[2] `https://platform.openai.com/tokenizer` [Accessed: Sep. 28, 2025]

Next to the web pages, we also included elements of websites, for example, singular listings, pop-ups, or product information. These web elements have the advantage of mainly being much smaller, taking up fewer tokens, and are also easier to handle in evaluation for us, by focusing on fewer deceptive patterns and smaller code snippets. To also remove unnecessary tokens from some web elements, we deleted parts of the CSS code that are not needed for their styling, and in a few cases, repeated elements. Overall, we ended up with 17 web elements, four of which contained no deceptive patterns. The web elements were taken from the following websites: *aliexpress.com*, *amazon.com* (2x), *audi.com*, *eventim.com*, *expedia.com*, *gotogate.com*, *ieee.org*, *mytrip.com*, *opodo.com*, *opodo.co.uk*, *pelacase.com*, *riverisland.com*, *ryanair.com*, *telegraph.co.uk*, *theguardian.com*, and *zalando.com*.

We further included multiple web elements from real websites, as they take up fewer tokens and are easier to evaluate.

To extract the code of whole web pages, we extracted the complete DOM (document object model) tree from the source page of each website in Google Chrome. However, this was not a fitting approach to get smaller elements from websites, as it is cumbersome to find each element in sometimes huge HTML code as well as the corresponding CSS. So to get the source code of smaller web elements, we used the Google Chrome browser extension *CSS Used*[3], which helps to extract the CSS, but also provides the HTML for each selected element. However, this often did not include the JavaScript code. Additionally, because this did not work well for all websites, and we did not find a tool that worked better, this slightly limited us in our selection of web elements.

We used a browser extension to download HTML and CSS, which did not always include all JavaScript code.

Since not all JavaScript code was included in our extracted web elements, and we wanted to use a few deceptive patterns spanning across more than one web page or element, we had to implement some of it back into the web elements. This included *Eventim*, *Riverisland*, and *Opodo*. Added functionality was either switching between pages or showing a pop-up on one click. Additionally, *Theguardian* was assembled from two elements, the pop-up and a snippet from an article behind the pop-up, to resemble the actual look, but

We implemented slight changes to web pages, such as reconstructing functionality that got lost.

[3] `https://chromewebstore.google.com/detail/css-used/cdopjfddjlonogibjahpnmjpoangjfff` [Accessed: Sep. 28, 2025]

stay within the token limit. *Audi* was altered to resemble the actual page, as it looked different after extracting the source code. All alterations made changed the code; however, we deemed this to be appropriate in the situations, since they opened up possibilities to test deceptive patterns going over more than one web page, or changed the code to represent the actual website better.

We used eight not-real web pages from the literature. To include patterns not yet present in our dataset, as well as to have the option to evaluate isolated deceptive patterns.

Besides real web pages, we also wanted to include web pages with smaller source code that are easier for evaluation. Another reason is that we can evaluate deceptive patterns on their own, which is harder to find in real websites. These singular deceptive patterns in smaller code can give a view onto how the LLM might have problems with specific deceptive patterns, independent of other difficulties in more complex web pages. A third argument is that specific deceptive patterns could not be extracted from real websites. This includes *Countdown Timer*, as the timer never worked when the source code was downloaded and run locally. All in all, we decided to use websites that are not actual real websites, but instead made by humans or LLMs. For this, we included items from the publicly available datasets from Schäfer et al. [2025] and Krauß et al. [2025]. The former designed web elements based on literature and public websites; the latter had websites generated by LLMs in user studies using ChatGPT. Using items from both datasets helps to create a diverse dataset, especially by including deceptive patterns only present in one of the sets. We are aware that including LLM-generated websites encompasses certain biases in our study. However, the aforementioned advantages from using these sets motivated us to include a few anyway, keeping the amount relatively small, and the potential bias in mind. We used one fair and one manipulative web page from Krauß et al. [2025], and six web pages from Schäfer et al. [2025], two fair ones and four manipulative ones. Our names for them, as well as the included patterns can be seen in Table A.3. Each item from Krauß et al.'s dataset starts with "K_", and ones from Schäfer et al. with "S_".

To differentiate deceptive pattern types, we decided to use the ontology by Gray et al. [2024]. The reason for this was that it is a rather new ontology, which is grounded

in past research and taxonomies. Additionally, the three-tier differentiation gives a fine-grained overview. If applicable, low-level types are used, as these are the most specific ones. Appendix A provides the definitions taken from Gray et al. [2024] relevant to our work.

We used Gray et al. [2024]'s ontology to differentiate between different deceptive pattern types.

Overall, we ended up with a set of 27 web pages and web elements, which is a similar scope to Schäfer et al.'s set, thus we deemed it sufficient. We ended up with seven fair web pages and 20 deceptive ones. In the following, we will refer to elements in our dataset as "web pages", and mean both web pages and web elements with this. All web pages included in our dataset are presented in Table A.3. 19 different deceptive patterns are included from all five high-level patterns. However, most patterns belong to *Interface Interference* and *Social Engineering*, which is a similar split Schäfer et al. [2025] had. They explained this by noting that those types of patterns are often visual and textual patterns that can be defused with changes in the HTML that an LLM could perform, which is a similar pattern we noticed across our dataset. Across all items, 75 instances of deceptive patterns are present, with a maximum of 12 patterns in one web page, and a minimum of zero. The following deceptive patterns are included, ordered by high-level patterns:

We ended up with 27 web pages, which include 75 instances of deceptive patterns from 19 different types.

- Obstruction: Adding Steps
- Sneaking: 2x Disguised Ad, 2x Hidden Costs, Partitioned Pricing, 6x Reference Pricing
- Interface Interference: 10x False Hierarchy, 6x Visual Prominence, 3x Bad Defaults, 2x Positive Framing, 2x Trick Question, 3x Hidden Information
- Forced Action: 2x Nagging, Forced Registration
- Social Engineering: 5x High Demand, 9x Low Stock, 2x Testimonials, 2x Activity Message, 2x Countdown Timer, 4x Limited Time Message, 9x Confirmshaming, Personalization

3.1.2 Evaluation Criteria

We wanted to use evaluation criteria that provide more details than Schäfer et al.'s scale.

To evaluate the output from our pipeline, we had to define criteria and scales. Schäfer et al. [2025] used a single scale ranging from -2 to 2, classifying web pages into "-2" if the web page was made worse, for example, by removing non-manipulative elements or hallucinating, and into "2" if all manipulation was removed. However, that scale does not tell us what exactly went wrong in each round, whether the LLM wrongly removed something, hallucinated, or made the web page more manipulative. That's why we decided not to use this scale.

Human Alignment is one of the most common approaches to evaluate LLM-as-a-Judge. We do this in a user study with the final pipeline.

Looking into the LLM-as-a-Judge literature, it is suggested to measure the alignment with human judgment and preferences [Li et al., 2024]. This is something we do for the final pipeline with a subsequent user study afterward, by comparing it to how users would change the web pages. The results can be found in Chapter 4. The user study also consists of other tasks that needed the technical approach to be done beforehand, which is why we decided to do the study afterward. We also had too many web pages included and not enough time to run an additional user study with all the web pages beforehand. Other criteria mentioned in the literature do not fit our use case, such as existing benchmarks or suggested scores that need human scores provided beforehand [Li et al., 2024].

We based our evaluation criteria on common mistakes Schäfer et al. [2025] noticed in their evaluation.

That's why we decided to define our own evaluation criteria. We based them on observations made by Schäfer et al. [2025] when they evaluated their results, specifically on common mistakes the LLM made. Next to not removing all manipulation, mistakes included information and functionality that was hallucinated, changed, or removed, and additional manipulation that was added. Additionally, we decided to add a criterion concerning whether or not the design was changed. Lastly, we counted after how many iterations the LLM judge decided that all manipulation was removed. Overall, we decided on the following evaluation criteria:

- DP REMOVED: Whether or not all deceptive patterns were removed.
- DP ADDED: Whether or not any new deceptive patterns were added by the LLM.
- FUNCTIONALITY: Whether or not all functionality was kept the same, or if some was changed, removed, or added.
- INFORMATION: Whether or not all information was kept the same, or if some was changed, removed, or added
- DESIGN: Whether or not the design was changed noticeably. This does not include design changes necessary to remove manipulation.
- #ITERATIONS: The iteration in which the judge stopped. We start counting at one. If it stopped right at the first iteration #ITERATIONS is one. If we allow i iterations, the maximum number of iterations is i+1, as this is the score assigned when the LLM does not stop within our limit.

We used a scale from 1 to 3, as everything larger was deemed too obscure to define.

We decided to use a discrete scale ranging from 1 to 3 for each criterion, besides #ITERATIONS. The decision was made as we thought a scale larger than that would be too obscure to define. For example, for a scale from 1 to 5, the question arises when functionality that was changed is classifiable as a "2" and when it is classifiable as a "4", as we cannot know beforehand what possible scenarios will arise, and the severity of mistakes can be subjective. So we landed on 1 to 3, defined as follows:

1: The worst rating. Depending on the criterion, for example, manipulation not removed, functionality/information added, design changed.
2: Depending on the criterion, for example, manipulation partially removed, information/functionality/design partially changed.
3: The ideal rating. Depending on the criterion, for example, manipulation fully removed, no manipulation added, information/functionality/design the same.

We also counted the number of deceptive patterns removed.

The scale does not capture how many deceptive patterns were removed. If only one deceptive pattern was removed, the rating would be the same as if all but one deceptive pattern were removed. To better capture this, we also counted how many patterns were removed from each web page.

We calculated the mean, standard deviation, and success rate for all criteria and the whole web page.

For the evaluation of each defined pipeline, we calculated the mean, as well as the standard deviation, for each criterion and for all criteria, except #ITERATIONS combined. We also calculated the percentage of deceptive patterns removed across all web pages. Further, we determined the success rate for each criterion, i.e., the percentage of how many of the web pages received a score of "3" in this criterion. Lastly, we calculated the overall success rate, i.e., how many web pages were a true success, meaning they got the score "3" across all criteria except #ITERATIONS. We only rated the final results of each iteration, as this is what the LLM decides is finished and would be the output given to the user in an actual application.

3.1.3 Adjustments

It is not trivial to how to design prompts, what strategies to apply, and what models to select for either the judge or the generator, and potential biases should be taken into account and avoided [Li et al., 2024]. Due to that, we gathered possible ways to adjust our prompts or the pipeline based on common approaches in the literature. The following adjustments are already arranged in the order we tested them.

Model Selection

The judge and generator should not be the same model to avoid self-bias.

One of the first questions that arises when implementing LLM-as-a-Judge is what models to use for both LLMs, the judge and the generator. An important bias to take into account here is the self-bias, which specifies the phenomenon that judges favor their own output in comparison to other models' output [Huyen, 2025]. As a conclusion, the judge and generator should not be the same model.

The most common model used as a judge in the literature is GPT-4, which has been shown to be very close to human evaluators [Gu et al., 2024; Huyen, 2025; Zheng et al., 2023]. However, as we write this thesis in summer 2025, GPT-4 is already considered an older model, and was retired by Azure OpenAI in June 2025[4]. Additionally, more advanced general-purpose models have been released by OpenAI since then, including GPT-4o, which is described as being impressive in its understanding and speed [Islam and Moushi, 2025]. As GPT-4o was one of the most recent ones at the time we evaluated everything, and it is also the one Schäfer et al. [2025] used, we selected it as one of our models to test.

GPT-4 is in many tasks very close to human evaluators, but already an older model.

We included GPT-4o as one of the models to test.

GPT-4 and GPT-4o are both general-purpose large language models. However, there are also reasoning models. The difference is that the latter performs thinking and reflection to arrive at a solution, which is handy for a judge during evaluation, and should be considered in the model selection [Gu et al., 2024]. That's why we decided to include a reasoning model. At the time, o4-mini by OpenAI was one of the newest reasoning models, being faster and cheaper compared to o3[5], which was released at the same time[6]. Which is why we selected o4-mini for our tests.

We also included a reasoning model, o4-mini, to evaluate.

There exists a family bias, meaning models might favor output from models that have the same training data or architecture [Spiliopoulou et al., 2025]. Even though GPT-4o and o4-mini are not from the same family, they are from the same company, OpenAI. That's why we decided to include another model from a different company. We decided to use Gemini 2.5 Flash, as it is one of the newest models from Google Gemini, which also incorporates reasoning abilities and is available for free within a specific token limit that is sufficient for us [Comanici et al., 2025]. Gemini 2.5 Flash allows us to adjust its reasoning capabilities, which they call thinking. We decided to test both the ver-

We also included Gemini 2.5 Flash to test, as we wanted to include a model not from OpenAI. We will utilize Gemini's option to change the amount of thinking tokens.

[4] `https://learn.microsoft.com/en-us/azure/ai-foundry/openai/concepts/legacy-models` [Accessed: Sep. 28, 2025]

[5] `https://platform.openai.com/docs/models/compare` [Accessed: Sep. 28, 2025]

[6] `https://openai.com/index/introducing-o3-and-o4-mini/` [Accessed: Sep. 28, 2025]

sion with full reasoning and the version with no reasoning, calling them *Gemini (Thinking)* and *Gemini (without Thinking)* respectively.

We decided to test six model combinations in a pretest, and then continue with the most promising ones to test on the full dataset.

Overall, we ended up with four different models: GPT-4o, o4-mini, Gemini 2.5 Flash (Thinking), and Gemini 2.5 Flash (without Thinking). As we do not have the capabilities to test all combinations of models, we decided to test six combinations with a subset of web pages, and then continue to test the most promising ones out of those six with the whole test set. Due to its strong reasoning capabilities, we tested o4-mini as a judge with all three other models as the generator. We then assessed each of the three other models exactly once as a judge to see how they perform, selecting GPT-4o as a generator for both versions of Gemini 2.5 Flash, and o4-mini for GPT-4o as a judge.

Prompting Strategies

We tested the three prompting strategies *Chain-of-Thought*, *Persona*, *Few-Shot Prompting*.

A common approach to improve prompts is through known prompting strategies, such as *Chain-of-Thought*, *Persona*, or *Few-Shot Prompting* [Gu et al., 2024; Li et al., 2024; Szymanski et al., 2025]. We added these strategies, if not mentioned otherwise, for both the judge and generator, as we did not have the time to test each approach for both LLMs on their own. We opted for testing each prompting strategy on its own to see the potential of each on its own, and, in the end, combining the most promising ones.

We based our persona on Schäfer et al.'s persona.

Persona describes the method of giving the LLM a role in which it should act and reply in, for example, imitating the abilities of an expert in that field. This can consequently enhance the quality of the output [Szymanski et al., 2025]. The persona we included for the generator was based on Schäfer et al. [2025], which specifically is *"You are an AI assistant that helps to design websites"*. We adapted it for the judge to *"You are an AI assistant that helps to evaluate websites"*.

Providing the LLM with only the prompt and no example is known as the strategy *Zero-Shot Prompting*, including one is called *One-Shot Prompting*, and providing multiple examples is called *Few-Shot Prompting*. *Few-Shot Prompting* can enhance the in-context learning abilities of LLMs, and thus, often enhances their performance [Brown et al., 2020]. For *Few-Shot Prompting*, we provided the examples alongside feedback for the judge, and with a fair version and feedback for the generator. When deciding on examples, we did not want to include examples that were too large, due to the token limit. The examples also should not be too closely related to our dataset, to test the ability of the LLM to transfer the scenarios. Based on those decisions, we concluded to implement the web pages ourselves, as they are small and the code should be easily understandable for the LLM. We did not want to include a fair web page, as we tested *Few-Shot Prompting* at a point where we already evaluated the dataset a few times before, and the LLM never made any mistakes with any fair web pages. We decided on the following web elements and deceptive patterns:

We implemented four example web pages with deceptive patterns, which we gave both LLMs for *Few-Shot Prompting*.

- A web page integrating a *Social Engineering* pattern, as those are commonly included in our dataset, and in actual web pages. We decided on *Activity Message*, as that one is not too often included in our set. The web element was the listing of a shopping web page.
- A textual pattern that also relates to the functionality. We used a *Trick Question* here, where the checkbox is designed as an opt-out in a cookie banner.
- To include a visual pattern, we included both *False Hierarchy* and *Hidden Information* together in a cookie banner.
- Lastly, we added a deceptive pattern that the generator cannot be remove, which in our case is *Forced Registration*.

Lastly, *Chain-of-Thought* Prompting is a method that is meant to enhance the reasoning capabilities of models. The concept is to ask the LLM to decompose its thinking steps into smaller ones, letting it focus more on specific steps.

We tested Zero-Shot *Chain-of-Thought*.

This can be done either through a Zero-Shot or a Few-Shot approach [Wei et al., 2022], we opted for Zero-Shot, to not mix the strategies.

Communication between Judge and Generator

We evaluate the communication when the judge provides feedback, and when he does not.

Before implementing the pipeline, the question arises as to what the communication should look like between the two LLMs. Specifically, meaning what should the judge return, and subsequently, what should the input for the generator be. Our goal is to iteratively refine the web page and remove the manipulation. With this in mind, we looked for similar approaches in the literature. Popular approaches with a similar iterative cycle implemented a feedback loop, i.e., one or multiple judges generated feedback, the generator then acts on [Patel et al., 2024; Vasudevan et al., 2025]. We decided to adapt this system as it sounds promising and fitting to our use case. However, we also compare it to a pipeline that does not include feedback from the judge, and instead only asks the judge whether to continue or not.

During evaluation, we noticed little autonomy from the generator, which is why we wanted to try encouraging critical thinking next to the feedback.

During our evaluations of the previous adjustments, we noticed that the generator only focused on the feedback provided, blindly applying it, without considering whether or not it actually should, and without contemplating potential other changes. That is why we decided to test another potential adjustment, in which we used feedback but included an addition in the generator's prompt that encourages it to act more autonomously, and think critically about the feedback. But also potentially go further than the feedback, possibly noticing deceptive patterns that the judge did not. We call this approach *"Feedback + Autonomy"*.

Evaluation Criteria

In Chapter 3.1.2, we discussed the criteria to evaluate the output. Consequently, these are the criteria we want the LLMs to meet. Often in the literature, in those feedback loops mentioned above, the judge is given the criteria to help evaluate and decide on whether another round is

needed or not [Patel et al., 2024; Vasudevan et al., 2025]. Further, Patel et al. [2024] suggested splitting the criteria among multiple judges, each an instance of the same model. This yielded better results and higher success rates for them compared to a single judge who was given all the criteria. Consequently, we decided to test the following cases: presenting a single judge with no criteria at all *(No Criteria)*, except DP REMOVED, giving a single judge all criteria *(Criteria in Prompt)*, and utilizing multiple judges, who were each given one criterion *(Multiple Judges)*. Our reasoning to include the first case was that we imagine that including criteria could also have disadvantages in our use case, as the lines between information that is manipulative or not may not always be clear.

We evaluate the variants to give the LLM no evaluation criteria, presenting one judge with all criteria, or splitting the criteria among multiple judges.

When discussing what criteria to include, we concluded on the following three: DP REMOVED, FUNCTIONALITY, and INFORMATION. The first one is our task, thus it is also included in *No Criteria*. Reasons to not include DP ADDED and DESIGN are twofold. On one hand, we wanted to keep the testing in a doable scope, and including more than three criteria, especially in the scenario where we have multiple judges, comes with high running times and costs. We deemed three criteria enough to see if this approach works or not. On the other hand, we tested this approach after we had already tested other approaches, noticing way fewer mistakes when it comes to DESIGN than FUNCTIONALITY and INFORMATION. And for DP ADDED, we did not see a significant difference for the LLM between DP ADDED and DP REMOVED, as the LLM just needs to check for manipulation overall. Consequently, we did not include either criterion. As for FUNCTIONALITY and INFORMATION, the LLM needs a comparison to what was present before, we included the original HTML in each prompt, in which a judge had to check those criteria. If necessary, we included it in the prompt for the generator as well. The evaluation criteria FUNCTIONALITY and INFORMATION were only included for the judge from the second iteration on, i.e., after the original HTML was adjusted by the generator once, as they need the comparison to the original to evaluate on.

We only include the criteria DP REMOVED, FUNCTIONALITY, and INFORMATION to limit the overhead. The last two criteria were included from the second iteration on.

Guardrails

Similar to Schäfer et al. [2025], we will also test guardrails.

Lastly, another adjustment is to include guardrails (rules) in the prompt [Li et al., 2024]. This is also what Schäfer et al. [2025] did when they constructed their improved prompt, which enhanced the cases in which all manipulation was removed by 27% compared to no guardrails. As a result, this appears to be promising, and we will test this in our case as well. Similar to Schäfer et al. [2025], we will base our guardrails on our results, more precisely, all mistakes made by the LLMs in what we evaluated before. Consequently, we will test this approach after everything else.

3.1.4 Baseline

As a comparison with no LLM-as-a-Judge, we use Schäfer et al. [2025]'s improved prompt and GPT-4o.

To have a comparison for our LLM-as-a-Judge pipeline, we need a baseline to compare our results to a setting without LLM-as-a-Judge and see if we actually improved in comparison. For this, we decided to use GPT-4o with the improved prompt from Schäfer et al. [2025], as this is currently the best known approach for deceptive pattern removal. The improved prompt is described in their paper. However, they did not describe what temperature they used. The temperature is a parameter defining how deterministic the results from an LLM are [Peeperkorn et al., 2024]. With no guidelines, we decided to leave the parameter at its default setting of 1.0. Schäfer et al. [2025] let the LLM run for 10 iterations, but found the optimal spot to be three iterations in most cases. Consequently, we evaluated the results after three iterations.

Before discussing the results in Chapter 3.2.6, we describe all the conclusions and insights we had during this, which are relevant for our implementation of the LLM-as-a-Judge pipeline in Chapter 3.1.5. It is important to understand that the LLM in Schäfer et al.'s approach returned the whole, adjusted HTML code in each iteration. The most significant problem occurred with the output token limit. The

output token limit for GPT-4o is only 16.384[7], much lower than our input token limit, which we set to 50K tokens, and the size of most web elements and web pages. Returning the whole web page is thus not feasible for us with our test set, and in general with most real web pages. As a solution, we decided to let the LLM only return the passages it wanted to adjust and the proposed change, i.e., what it should be replaced with. We then implemented a function to replace the section ourselves in the HTML. To confirm this addition did not significantly change the results, we tested a few web pages that are small enough, with both approaches, and did not see huge differences between their results. To make sure the LLM did exactly what we asked from it, which is necessary for our implementation to work, we extended Schäfer et al.'s prompt with a specification of how the LLM should return the changes. This can be found in Appendix B.

LLMs have a relatively small output token limit, which makes it not feasible for them to return the whole web page all the time. Thus, we decided to let the LLM only return the changes it wants to make.

A similar problem occurred with the input token limit. We had a limit of 50K input tokens for GPT-4o. However, in Schäfer et al.'s approach, they included the chat history, so all previous answers from the LLM, in each iteration. This means that in iteration two, the web page was already included twice in the conversation, once as the message in the original prompt, and once in the response the LLM made in iteration one. Therefore, the input given to the LLM got bigger and bigger. Web pages that were already rather large rapidly reached the input token limit, and the API threw an error before being able to finish iteration three. This also happened after we changed the output to only the changes and not the whole HTML. Consequently, we decided to not include a chat history in our baseline evaluation.

Due to the input token limit, it was not rational to include the chat history in each iteration.

3.1.5 Defining the Pipeline

We now explain how we defined the initial pipeline, i.e., set all the adjustments explained in Chapter 3.1.3 to test the first adjustment, thus, before being able to evaluate them. We also explain everything else we set.

[7] `https://platform.openai.com/docs/models/gpt-4o` [Accessed: Sep. 28, 2025]

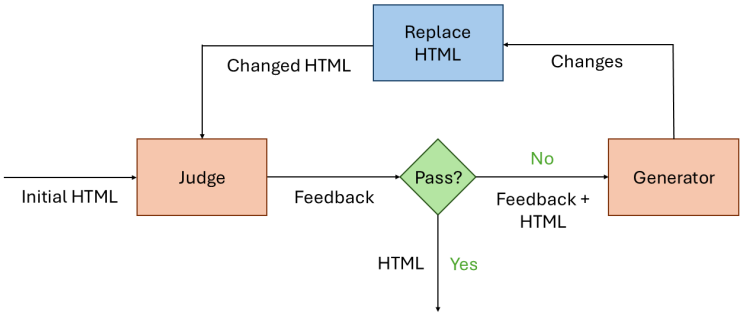

Figure 3.1: The pipeline between both LLMs, the generator, and the judge. The judge ends the cycle or provides feedback that the generator then implements by giving us the changes to realize.

Our initial pipeline starts with the judge, who ends the cycle or returns feedback that we give to the generator. The generator then returns the changes. The changed HTML is then again presented to the judge.

The initial pipeline is depicted in Figure 3.1. We decided to start with the judge to avoid unnecessary changes on the web pages from the generator. As explained in Chapter 3.1.3, the most common approach in the literature is a feedback loop, which is why we decided to start with this approach, deeming it to be the most promising one. The LLM judge then either returned feedback, if changes are needed, or replied that no changes are necessary. In case of the latter, the current HTML is the result; if the former is the case, we use the feedback and the current HTML as input for the generator, which then returns the changes that should be applied. Our program then performs the changes, and the altered HTML is again given to the judge, starting the cycle over. The whole cycle is repeated for a maximum of five iterations. We set five as an initial limit, as this is a little more room than the optimal of three found by Schäfer et al. [2025], and in our case, the LLM should end the iterative cycle. We set a limit to not run forever. As explained in Chapter 3.1.4, we did not include a chat history.

For the prompt, we chose to start with the minimal prompt *"Make that less manipulative"* from Schäfer et al. [2025], which we used for the generator and adapted slightly for

the judge to say *"Check if the following website is manipulative"*. The reason we did not choose the improved prompt, even though it performed better in their dataset, was that we wanted to use an unbiased prompt. The improved prompt was specifically designed for the mistakes GPT-4o made based on their dataset, but this was never tested further, and thus, the possibility of overfitting remains. We also added and adjusted the *Persona* Schäfer et al. [2025] used from the start. For the remaining adjustment parameters, our goal was to start with as little as possible and then gradually improve the prompt. Therefore, we did not apply any prompting strategies except *Persona*, and did not include any evaluation criteria or guardrails in the prompt at the beginning. The only parameter we had to set was the temperature for all models, except o4-mini, as it does not have this parameter. We set it to 0.2, which we based on a suggestion that this is fitting for code generation, as it is more deterministic[8]. Further possible parameters were not included. The initial, as well as all adjusted prompts, can be found in Appendix B

We started with Schäfer et al.'s minimal prompt *"Make that less manipulative"*, and included the persona they also used.

Most remaining adjustments were set to the minimal option.

3.2 Results

In the following, we will describe the results obtained during the evaluation of each adjustment. Our **learnings** are summarized after each adjustment we tried and form the basis for all decisions and evaluations we make afterward, as we continue to build on the best result of each adjustment for every iteration. At the end, we compare our final pipeline with the results we got from the baseline approach. When talking about examples, we often use the notation *(web page, iteration)* to show their origin. All exact values, as well as the success rate can be found in Appendix C.

[8] `https://www.prompthub.us/blog/understanding-openai-parameters-how-to-optimize-your-prompts-for-better-outputs` [Accessed: Sep. 28, 2025]

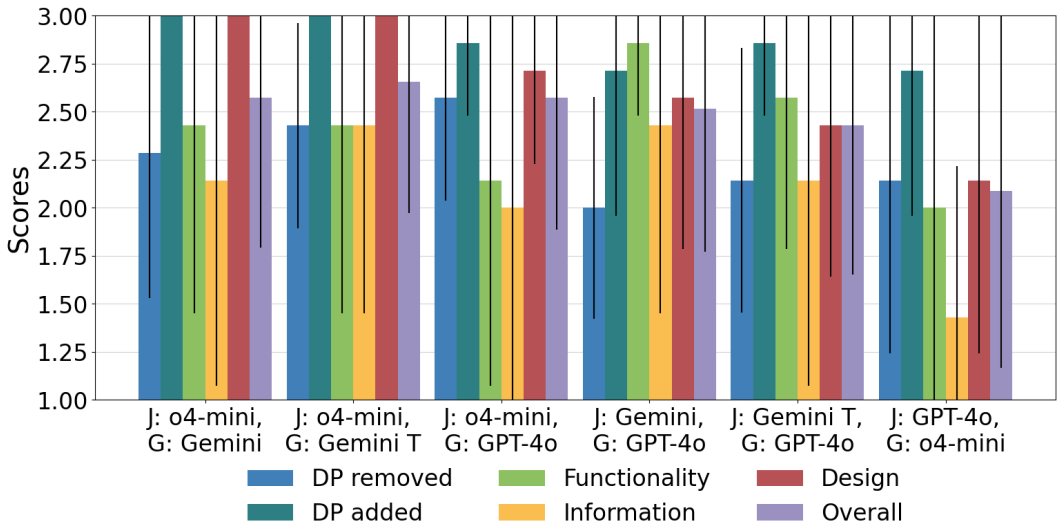

Figure 3.2: The figure shows the mean and standard deviation for the ratings for every evaluation criterion and the overall score for all model combinations in the pretest. The scale ranges from 1 to 3, and "3" is the best. GPT-4o as the judge performed the worst, while each combination with o4-mini as the judge outperformed every other judge.

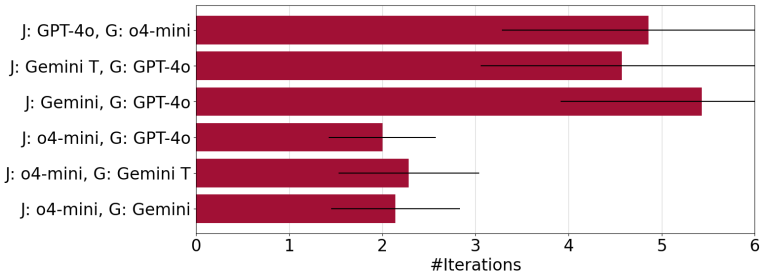

Figure 3.3: The figure shows the mean and standard deviation for #ITERATIONS for the different model combinations. Combinations in which o4-mini was the judge needed noticeably less iterations.

3.2.1 Models

Pretest

In the pretest, we ended up evaluating seven web pages, including web pages from both the literature and real web

pages, as well as a fair one. We stopped once we had the impression of noticing clear tendencies. Figure 3.2 and Figure 3.3 show the mean and standard deviation for the evaluation criteria for each model combination.

GPT-4o as the judge and o4-mini as the generator (GPT+mini) was the worst-performing model combination, with not a single web page receiving perfect scores across all criteria. The OVERALL score, and the four criteria DP ADDED, FUNCTIONALITY, INFORMATION, and DESIGN, all received the lowest mean and success rate across all six model combinations. Showing an especially low performance concerning INFORMATION, with a mean of 1.43, which reflects how all but one web page were compromised in their information. This was often reinforced by the judge in the feedback provided. For example, by suggesting adding information that is not given, such as *"provide clear explanations for why certain cookies are essential and cannot be opted out of" (Riverisland, i3)*, therefore encouraging hallucination. Similar cases appeared for FUNCTIONALITY, such as *"Enable users to uncheck essential cookies if they choose to" (Audi, i1)*, which undoubtedly should not be done. We did not provide a chat history, but the LLM judge still often focused on a singular manipulation or problem it identified for every iteration. It did not matter if that aspect was already adjusted or not; thus, it was barely satisfied with the changes made by the generator. As a consequence, many deceptive patterns were not removed, resulting in a low DP REMOVED score, and the number of iterations was the second highest across all model combinations.

GPT-4o as the judge performed the worst across almost all evaluation criteria. Specifically bad was INFORMATION.

Gemini 2.5 Flash (without Thinking) as the judge and GPT-4o as the generator (GWT+GPT) performed the worst in removing deceptive patterns. It received only one rating of "3" in this category, which was assigned to the web page containing no deceptive patterns. Additionally, it also performed the worst, alongside *GPT+mini*, in DP ADDED, as it added a False Hierarchy to the fair web page *Audi*. Interestingly, in this case, the judge suggested either making both equal or making one more prominent, and the generator chose the latter, adding deception. Surprisingly, this combination performed the best in both FUNCTIONALITY and INFORMATION, but had the highest average number of itera-

Gemini (without Thinking) performed slightly better than Gemini (Thinking) as the judge, but both lacked specifically in DP REMOVED and needed many iterations.

tions. The median of 5.43 iterations reflects how the judge only once stopped the iterative cycle. In comparison, *Gemini 2.5 Flash (Thinking) as a judge and GPT-4o as the generator (GT+GPT)* was just slightly better in DP REMOVED and DP ADDED, but turned out to be worse in FUNCTIONALITY and INFORMATION. Notably, each criterion only differed by one web page that received a worse score than *GWT+GPT*. It did need many iterations, but at least it stopped in all but three cases. All in all, this resulted in the second-worst OVERALL score, while *GWO+GPT* received the third-worst.

Both Gemini versions had problems detecting deceptive patterns, and did not correctly understand the HTML sometimes.

Both Gemini judges made similar mistakes, but at differing frequencies. Both had a tendency to tackle the deceptive patterns one after another, focusing on only one per iteration, which partly explains the high number of iterations. Additionally, similar to *GPT+mini*, in *GWT+GPT*, the judge repeated feedback even after it was fixed. Showing it is unsatisfied with the results applied, apparently meticulously opting for a specific solution. Furthermore, both variations of Gemini misinterpreted the HTML code sometimes, by, for example, having problems recognizing two buttons as being equal. Similarly, problems arose while classifying what is actually manipulative and what is not. Neither realized the fair web page is fair, but then had problems detecting and removing actual deceptive patterns, evident in the two lowest scores received in DP REMOVED. Lastly, Gemini often included two options in its feedback on how the generator could apply a change. This was regularly bound to the condition of whether something is genuine or not, e.g., in *GWO+GPT*, the judge said *"unless it genuinely represents a recent, regular selling price" (amazon, i3)*, which is information the generator does not have, so this distinction is useless in our scenario. As a conclusion, seemingly, the most prominent problems for both Gemini models did not concern functionality or information, but instead deceptive pattern detection and web page understanding.

Finally, we have o4-mini as a judge, which performed the best overall. In DP REMOVED, with all three generators, it performed better than a combination with any other judge. *o4-mini as the judge and GPT-4o as the generator (mini+GPT)* performed the best overall in this criterion. For DP ADDED and DESIGN, with both Gemini versions as generators, it

received a perfect score, and with GPT-4o as the generator, it achieved only slightly lower ratings. FUNCTIONALITY scores are worse for all three generators than the model combinations with both Gemini variants as judges, but better than the combination with GPT-4o as a judge. Similarly, again the versions with Gemini as the generators performed better than the one with GPT-4o. Likewise appeared for INFORMATION, but with Gemini (Thinking) it even received the best overall score. All three versions needed the least amount of iterations, always finishing, and *mini+GPT* was the fastest overall. Finally, *mini+GT* obtained the highest overall score with a mean of 2.66, *mini+GWT* took the second spot, and *mini+GPT* the third best. Overall, o4-mini as a judge performed best in all categories, except FUNCTIONALITY. o4-mini as the judge did also make mistakes, but they were less often and less severe than than what the other two judges did.

o4-mini as a judge performed better than any other judge, *mini+GT* achieved the best scores overall. Only the FUNCTIONALITY was slightly worse than *GWO+GPT*.

Learnings Both Gemini versions produced promising results in regard to FUNCTIONALITY and INFORMATION, but they lacked considerably in other areas, such as deceptive pattern detection and the number of iterations, which is why we did not deem Gemini as a good enough judge. Even worse was GPT-4o as a judge, who was outperformed by every other judge, specifically producing bad results in regard to FUNCTIONALITY and INFORMATION. All in all, we identified o4-mini to be the most promising judge among all models tested. While o4-mini was the judge, all three generators performed better in specific categories than any other model constellation, but we did not see clear tendencies between the three that one performed noticeably better than the other two. Therefore, we decided to proceed to test all three generators with the complete dataset, while setting o4-mini as the judge, and thus, we will go into more details into the differences between the generators and the mistakes o4-mini made in the following section.

o4-mini was the most promising judge, which we continue to test on the full set. GPT-4o was noticeably the worst.

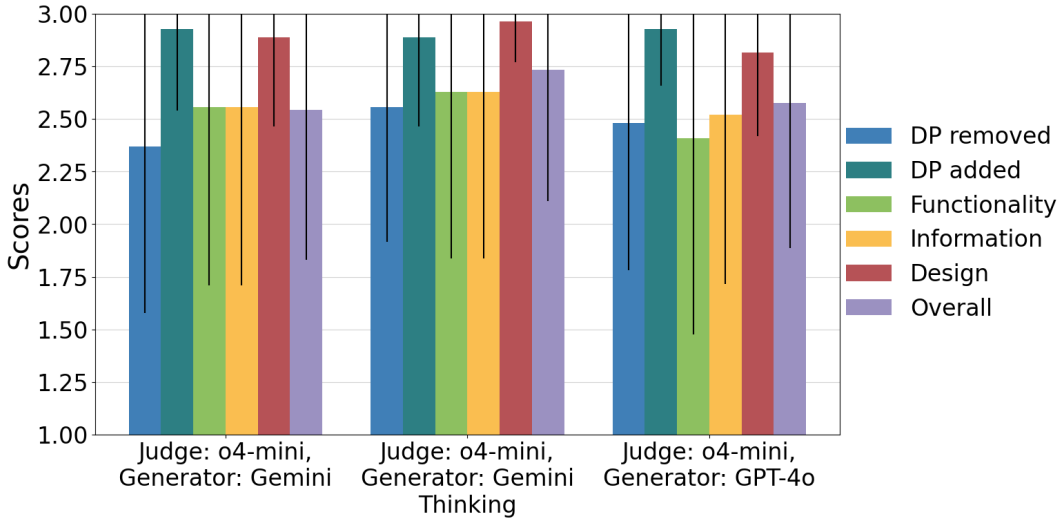

Figure 3.4: The figure shows the mean and standard deviation for the ratings for every evaluation criterion and the overall score for the different model combinations. The scale ranges from 1 to 3, and "3" is the best. Gemini 2.5 Flash (Thinking) as the generator performed the best in all criteria.

Different Generators

After we identified o4-mini as the judge performing best in our pretest, we continued to evaluate all three generators with o4-mini as a judge on the full dataset. The mean and standard deviation for all criteria can be seen in Figure 3.4.

Gemini (Thinking) as the generator performed best across almost all criteria, and *GPT-4o as the generator* the worst in all but DP REMOVED and the OVERALL score.

With the generator *GPT-4o*, the worst results were obtained in FUNCTIONALITY, in which it received a score of "1" for eight web pages, resulting in a success rate of 70.37%, and DESIGN with one of 81.48%. *Gemini (without Thinking)* only performed slightly better in each of those, with the more noticeable differences regarding FUNCTIONALITY and DESIGN, with success rates of 77.78% and 92.6% respectively. However, it removed the least amount of deceptive patterns. The best performing generator was *Gemini (Thinking)*, as it had the highest values across all categories, except DP ADDED, and the best overall score with a mean of 2.73. These results amplify the slight tendencies that were noticeable in the pretest.

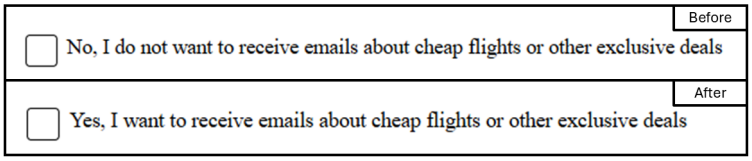

Figure 3.5: Before and after the pipeline with o4-mini as a judge and Gemini 2.5 Flash (Thinking) adjusted the web page *MyTrip*. This is only a section of the original web page. The LLMs switched the wording, but not the functionality of the checkbox.

However, the general results are relatively close together, especially between the two Gemini versions. The categories DP ADDED, FUNCTIONALITY, INFORMATION, and DESIGN all differed by only one. This means that the distribution of the scores was the same, only varying by one. Differences between the *Gemini* versions and *GPT-4o* were slightly larger, with up to four differences in the distribution. Errors were similar across all generators, but more common in *GPT-4o* and *Gemini (without Thinking)*. Mistakes include not succeeding in what it said it did, such as not actually removing something, but also adjusting something in the wrong way. For example, *GPT-4o* removed the strike in front of a price and not the price behind it *(gotogate, i1)*, and *Gemini (without Thinking)* did not actually remove a testimonial even though it said it did *(expedia, i1)*.

There are only marginal differences between both Gemini versions, GPT-4o as the generator, was slightly worse.

The results need to be taken carefully, and worse results are not always due to the performance of the generator, but instead due to varying feedback provided by the judge. This refers specifically to mistakes *o4-mini* made in the iteration with one generator, which it did not make in the iterative cycle with the other two, even with a similar or the same initial input HTML. Examples include not deeming something manipulative and thus stopping during the first iteration, or suggesting specific changes, additions, and removals that it did not suggest for the others. When *Gemini (Thinking)* was the generator, only one individual mistake was made; two were made when *Gemini (without Thinking)* was the generator, and three for *GPT-4o*.

The judge made the most mistakes when *GPT-4o* was the generator, and the least when *Gemini (Thinking)* was.

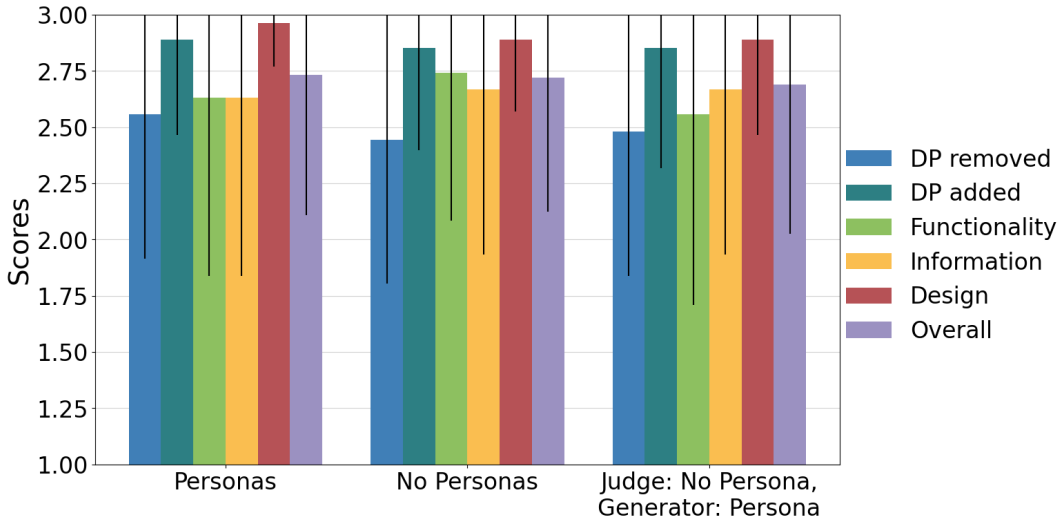

Figure 3.6: The figure shows the mean and standard deviation for the ratings for every evaluation criterion and the overall score for all persona variations. The scale ranges from 1 to 3, and "3" is the best. Gemini 2.5 Flash (Thinking) as the generator performed the best in all criteria.

Learnings Concluding, we selected *Gemini 2.5 Flash (Thinking)* as the generator to continue with, as it received the highest scores across most categories, and the most mistakes it made were based on feedback the judge provided. GPT-4o took the last place; however, both other generators did not perform substantially worse.

3.2.2 Prompting Strategies

Persona

We started evaluating the prompt with and without a persona, and also included a version in which only the generator had a persona. The persona approach is the same as the *"Judge: o4-mini, Generator: Gemini (Thinking)"* approach from the prior section, as we already included a persona there. Figure 3.6 shows the results.

The worst in DP REMOVED, with a mean of 2.48, was *No Persona*, while *Persona* received the highest mean of 2.56

and removed 70.67% of the deceptive patterns. However, the least amount of deceptive patterns were removed by *No Persona Judge and Persona Generator (NPJ+PG)*, with only 64% of all deceptive patterns removed, while *No Persona* removed 65.33%, which equals one removal more. *NPJ+PG* received the worst FUNCTIONALITY score, but the highest, alongside *No Persona*, in INFORMATION. Next to INFORMATION, *No Persona* also received the highest score in FUNCTIONALITY. In contrast, *Persona* received the highest rating next to DP REMOVED, also in DP ADDED, DESIGN, and the OVERALL score of 2.73. *No Persona* was relatively close behind, while *NPJ+PG* received the highest success rate of 40.74%, but the lowest average rating of 2.69. This suggests that *NPJ+PG* had more web pages that were a complete success, but the ones that were not were often worse than the ones that the other two versions made.

Persona removed the most deceptive patterns, while *No Persona* was marginally the best in FUNCTIONALITY and INFORMATION.

While there were higher ratings, overall, all ratings were relatively close together, especially the ratings DP ADDED and INFORMATION, which had a maximum of one difference in their distributions across all three pipelines tested. The most noticeable divergences were for the categories DP REMOVED and FUNCTIONALITY. In the latter, *NPJ+PG* performed the worst. Between *Persona* and *No Persona*, the latter performed the best in FUNCTIONALITY, and the former in DP REMOVED. The distribution for FUNCTIONALITY varied in two web page scores, DP REMOVED in three. Looking into the differences for FUNCTIONALITY, at least one was not because it consciously did not make that mistake, but instead, it was never in a situation to make it. This is due to the LLM not seeing the manipulation, a Trick Question, that resulted in that mistake. However, the same deceptive pattern exists in another web page with a similar setting in our dataset, where *No Persona* made the same mistake, which it had avoided before due to not noticing the manipulation.

Differences in ratings were marginal. The divergences in FUNCTIONALITY were not always due to the LLMs not making the same mistake.

Learnings As a conclusion, we decided on *Persona* to be the most promising version, as it was better at removing deceptive patterns with the highest OVERALL score, while barely lacking in FUNCTIONALITY compared to the other two tested prompts, as explained above. However, the

We concluded to use *Persona*, but *No Persona* is barely worse.

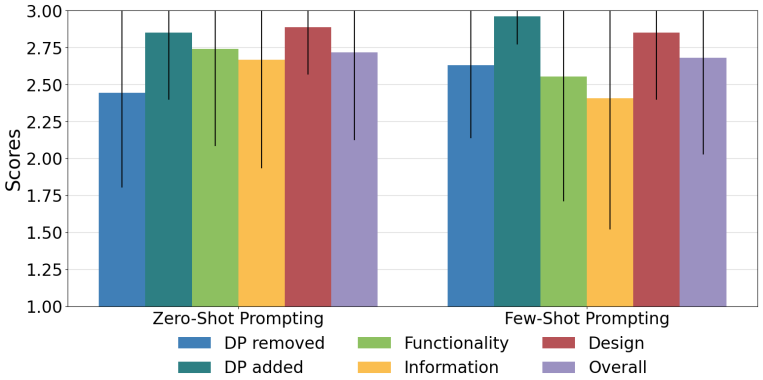

Figure 3.7: The figure shows the mean and standard deviation for the ratings for every evaluation criterion and the overall score for *Few-Shot Prompting* versus *Zero-Shot Prompting*. The scale ranges from 1 to 3, and "3" is the best. *Few-Shot Prompting* improved the number of deceptive patterns removed, but received lower scores in FUNCTIONALITY and INFORMATION

other two options followed very close behind, with only slightly noticeable differences.

Few-Shot Prompting

As explained in Chapter 3.1.3, we did not include a persona in the prompt for Few-Shot Prompting, and thus compare our results with the ones where we did not include a persona and used Zero-Shot Prompting (*"No Persona"* in the prior section). The results can be seen in Figure 3.7.

Few-Shot Prompting removed more deceptive patterns, but performed worse in FUNCTIONALITY and INFORMATION, as it exemplarily hallucinated more.

Few-Shot Prompting removed more deceptive patterns and added fewer new ones, while performing worse in FUNCTIONALITY and INFORMATION. Specifically noticeable is how it removed 80% of all deceptive patterns, while *Zero-Shot Prompting* only removed 65.33%. The OVERALL rating for *Few-Shot Prompting* was worse with 2.69, but with a higher success rate of 44.44% compared to 33.33%. The higher DP REMOVED score relates to the lower FUNCTIONALITY and INFORMATION scores, as often mistakes made in the *Few-Shot* approach were not made in the *Zero-Shot*

approach, as the deceptive patterns around such mistakes were also not attempted to be removed at all by the latter. However, that was not always the case, and other errors were, for example, due to hallucination, which happened more often than in *Zero-Shot Prompting*.

We included an example that contained a Trick Question with an opt-out approach in our examples given to the LLMs. However, for both web pages in our dataset that also incorporated this exact pattern, with just slightly adjusted use cases and wording, the LLMs did not change the web pages accordingly to our example. Instead, they made the exact same mistake they made with *Zero-Shot Prompting*, even though both had this example included in their prompt. To be precise, they changed the wording of the checkbox to opt-in, but did not adjust the functionality accordingly. Similarly, the Forced Registration example did not change the way the LLM pipeline changed the same pattern in *Opodo2*. Consequently, it tried to remove it, messing up FUNCTIONALITY and INFORMATION. Unlike the Trick Question example, this one was not as close to the web page we had in our dataset, but instead a different scenario and implementation.

The LLMs did not manage to transfer the exemplary changes to actual use cases, even though some were very similar.

False Hierarchy turned out to be better in Few-Shot Prompting, as one of the only patterns that we included in our examples. In *S_falseHierarchy*, the LLM did not add False Hierarchy back in, as it did in Zero-Shot Prompting, and for *Opodo* it removed the False Hierarchy. Additionally, Hidden Information in *Telegraph* was also removed, which is a deceptive pattern that is also present in the *Few-Shot* examples. It further removed some deceptive patterns only in the *Few-Shot Prompt*, such as Disguised Ad in *Amazon* and *Booking*. Those were patterns not present in our example set, but show how the LLM generally got better at detecting deceptive patterns.

Overall, our LLM removed more deceptive patterns than in any other case.

Even though we did not include an example for fair web pages in the *Few-Shot Prompting*, this did not have a negative effect on the evaluation of fair web pages. The LLM judge still did not make any mistakes in that regard, identifying them as not manipulative immediately, which it also did in *Zero-Shot Prompting*.

Fair web pages were again not compromised even though we did not include an example.

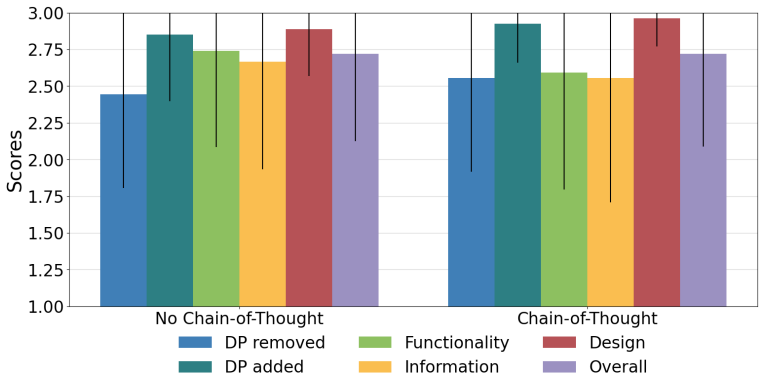

Figure 3.8: The figure shows the mean and standard deviation for the ratings for every evaluation criterion and the overall score for *Chain-of-Thought* versus No *Chain-of-Thought*. The scale ranges from "1" to "3", and "3" is the best. *Chain-of-Thought* improved in DP ADDED, but not in FUNCTIONALITY and INFORMATION.

Few-Shot Prompting was not a success due to low scores in FUNCTIONALITY and INFORMATION.

Learnings Although *Few-Shot Prompting* was more promising in removing deceptive patterns, it made more errors in FUNCTIONALITY and INFORMATION, which we classify as more severe than not detecting all manipulation. We settled on the conclusion that *Few-Shot Prompting* did not improve our pipeline, but that it made it worse in the case of our dataset.

Chain-of-Thought (CoT)

We now compare a prompt in which we included Chain-of-Thought (CoT) with a prompt without CoT, which is the *"No Persona"* approach from two sections ago. The results can be seen in Figure 3.8.

CoT increased in DP REMOVED, but decreased in FUNCTIONALITY and INFORMATION.

Similar to the *Few-Shot* approach, *CoT* improved the deceptive pattern removal and lowered the addition of new ones, but performed worse in regard to FUNCTIONALITY and INFORMATION. However, the INFORMATION was not as low as it was when we used *Few-Shot Prompting*, and the distribution only varied by two scores in both categories in this case. As a comparison, it varied by four in the INFORMA-

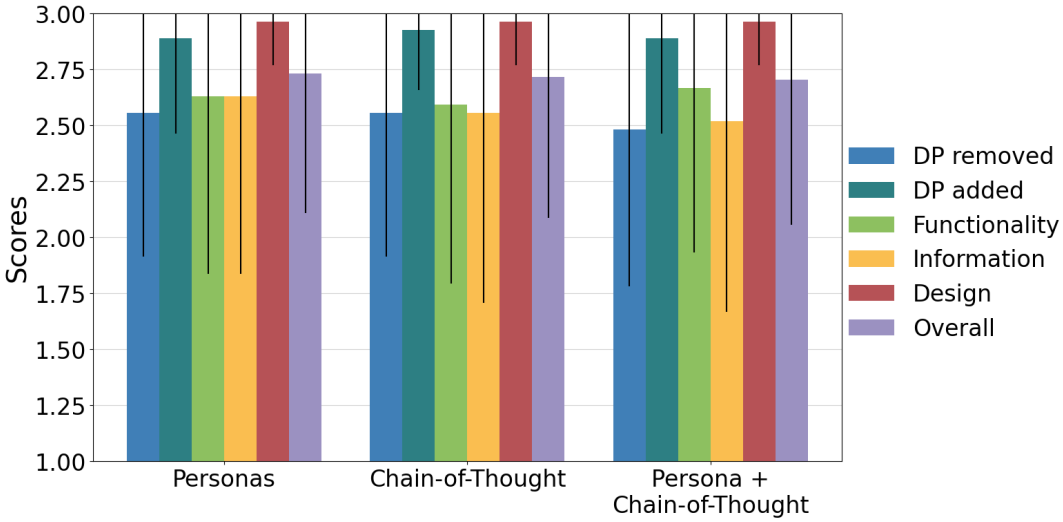

Figure 3.9: The figure shows the mean and standard deviation for the ratings for every evaluation criterion and the overall score for *Chain-of-Thought* versus *No Chain-of-Thought*. The scale ranges from 1 to 3, and "3" is the best. *Persona + CoT* did not improve compared to both options on their own.

TION category during *Few-Shot Prompting*. The OVERALL score for *CoT* and *No CoT* was 2.72 for both, but *No CoT* had a higher success rate of 44.44% instead of 33.33%.

Looking into the differences surrounding INFORMATION and FUNCTIONALITY, it once again happened that in three cases *CoT* made mistakes that *No CoT* did not notice the deceptive pattern in at all, and thus did not tackle that problem in either a correct or an incorrect way. Given better scores in DP REMOVED, DP ADDED, and DESIGN, and only a few more mistakes in FUNCTIONALITY and INFORMATION, we decided to test *CoT* together with a *Persona*. The results can be seen in Figure 3.9.

Some mistakes in FUNCTIONALITY were avoided due to the LLM not noticing them. We decided to test *Persona* and *CoT* together.

The only criterion in which *Persona + CoT* improved was FUNCTIONALITY. However, this was only a very small advancement, with *Persona* even having the same success rate of 81.48%. Every other criterion received a lower score, especially DP REMOVED and INFORMATION, but once again, this drop is also only very small in comparison. The overall score was also only slightly worse, with 2.7 for *Persona + CoT*, while *Persona* achieved 2.73 and *CoT* 2.72.

Persona + CoT did not improve the noticeably in any criteria.

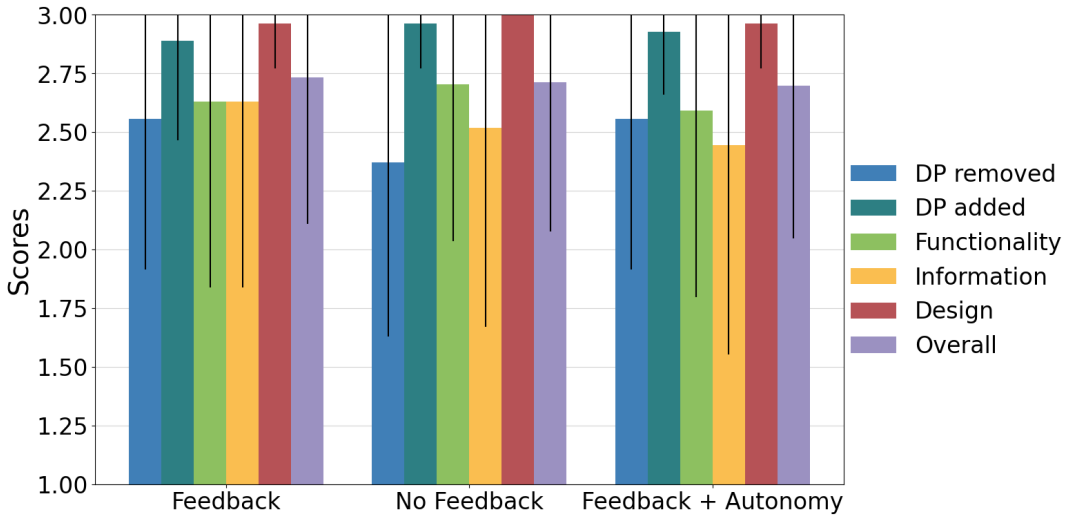

Figure 3.10: The figure shows the mean and standard deviation for the ratings for every evaluation criterion and the overall score for the different communication variants. The scale ranges from 1 to 3, and "3" is the best. *Feedback* still performed the best overall, *No Feedback* removed only very few deceptive patterns, *Feedback + Autonomy* suffered in INFORMATION.

Persona performed the best across all prompting strategies.

Learnings Overall, the combination *Persona + CoT* did not improve the results either. We then settled on continuing with only *Persona* as the only prompting strategy, as this received slightly higher FUNCTIONALITY and INFORMATION scores than only *CoT*, while having the same scores for DP REMOVED.

3.2.3 Communication

The results can be seen in Figure 3.10. The pipeline with feedback is the one we tested before, in Section 3.2.1, called *"Judge: o4-mini, Generator: Gemini Thinking"*.

Feedback + Autonomy achieved lower scores in FUNCTIONALITY and INFORMATION.

The pipeline *Feedback + Autonomy* achieved the same score for DP REMOVED as only *Feedback* did, with the latter removing only one deceptive pattern less (70.67% vs. 72%). FUNCTIONALITY and INFORMATION are worse in comparison. While the first one shows only a minimal difference, the latter differs more, with *Feedback* having three web pages more that received a score of 3 in this category. The

OVERALL rating is also worse, 2.7 compared to 2.73, even though the success rate is slightly higher, 40.74% compared to 37.04%. This shows that *Feedback + Autonomy* made more severe mistakes than *Feedback*.

With more autonomy, the generator still barely deviates from the feedback, and if it did, the changes were equally often beneficial as they were not.

In *Feedback + Autonomy*, despite encouraging critical thinking for the generator, the LLM mostly applied the feedback provided by the judge. Often explaining why it is useful, and always applying all of it. In four of the 27 web pages, the generator additionally changed something that was not included in the feedback. Twice it was useful, removing actual deceptive patterns. On the contrary, the generator also once added a False Hierarchy back in to *"clearly distinguish[...] the primary from the secondary or dismissive action" (S_confirmshaming, i1)*, and once changed the wording, which changed the meaning and thus information. Both decisions were independent of what the judge provided.

Providing no feedback resulted in way less deceptive patterns removed.

Comparing *Feedback* with *No Feedback*, the latter removed noticeably fewer deceptive patterns, with only 50.67% removed, while the former removed 70.67%. INFORMATION was also slightly worse, a mean of 2.52 versus 2.63. FUNCTIONALITY was marginally better, though the success rate was identical, showing that the *No Feedback* approach only received two more ratings of "2" instead of them being "1". All other values were not essentially different from each other. The OVERALL score was slightly lower for *No Feedback*, but with an identical success rate of 37.04%.

No Feedback had problems with Social Engineering patterns.

The most obvious difference is in the deceptive pattern removal, which the generator was not able to do as reliably as when the judge provided feedback. Notably, Social Engineering patterns were removed way less often. For example, in *Viagogo*, *No Feedback* did not remove a single deceptive pattern, while *Feedback* removed nine *Social Engineering* ones.

The best way of communication remained *Feedback*.

Learnings Overall, encouraging the generator for critical thinking, *Feedback + Autonomy*, did not have an actual impact. Although the generator had a positive influence in two iterations, it also made the web page worse in an equal number of iterations, and ended up with worse scores in

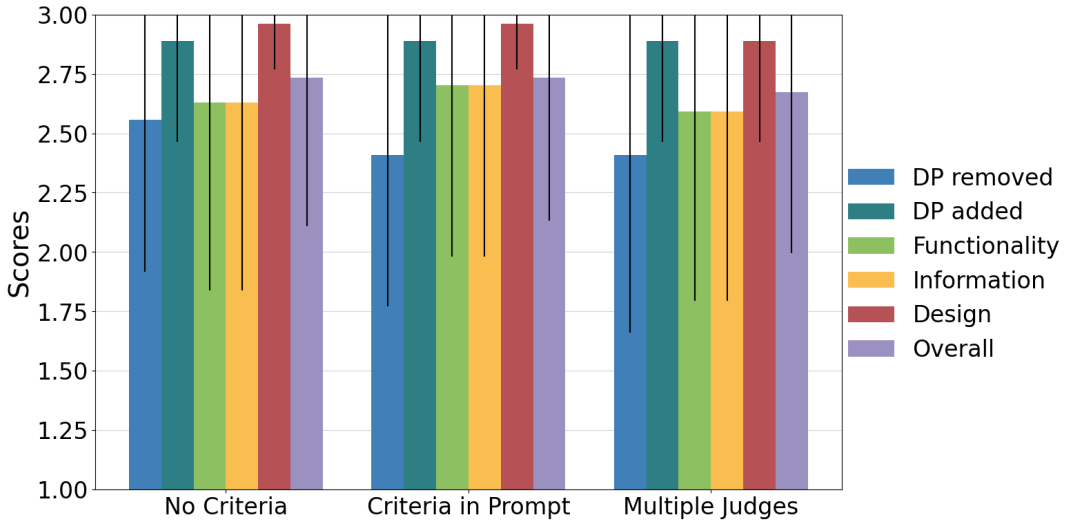

Figure 3.11: The figure shows the mean and standard deviation for the ratings for every evaluation criterion and the overall score for the different ways to include evaluation criteria. The scale ranges from 1 to 3, and "3" is the best. *No Criteria* performed the best in DP REMOVED, *Criteria in Prompt* the best in FUNCTIONALITY and INFORMATION.

INFORMATION. *No Feedback* made the pipeline worse, especially visible with fewer deceptive patterns removed. Hence, we still have *Feedback* as the best performing one to continue with.

3.2.4 Evaluation Criteria

Next, we compared the version with no evaluation criteria, i.e., only asked it to remove manipulation, (*"Judge: o4-mini, Generator: Gemini Thinking"* from Section 3.2.1) with the version in which we included criteria in the prompt of a single judge, and one that included multiple judges, each focusing on a single criterion. The results are present in Figure 3.11 and Figure 3.12.

Criteria in Prompt removed fewer deceptive patterns, and also needed fewer iterations.

One judge with multiple criteria (Criteria in Prompt) had a lower deceptive pattern removal rate than no criteria in the prompt, with an especially low percentage of removed patterns (46.67% vs. 70.67%). The scores for FUNCTIONALITY and INFORMATION were slightly higher; however, in com-

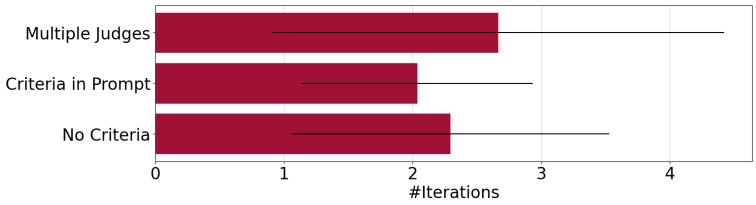

Figure 3.12: The figure shows the mean and standard deviation for #ITERATIONS for the different communication variants. While *Criteria in Prompt* needed the least, *Multiple Judges* needed the most.

parison to no criteria, only one web page was better in each criterion overall. DP ADDED and DESIGN are the same as *No Criteria*, but the number of iterations is lower, with a mean of 2.04 instead of 2.3. The overall score is the same, namely 2.67, and with a success rate of 37.04%.

The other two criteria, besides DP REMOVED, regarding FUNCTIONALITY and INFORMATION, were only included from the second iteration on. However, in only six cases did the judge actually decide to go beyond two iterations. In all other 21 web pages, the cycle ended after iteration one or two, which shows that in all of them, the judge decided the functionality and information was not negatively affected. In three of those six cases, it re-added manipulative information that was removed in the iteration prior. These were all textual patterns, such as Confirmshaming, whose information was deemed valuable and non-manipulative, besides their manipulative nature. In another case, the changes made due to identified mistakes in the information were actually correct, but instead resulted in incorrect information being added. This occurred in *Booking*, when the LLM identified information in two of the listings and incorrectly provided feedback that it should be present in each listing, although this was not present in the original. The two other changes based on evaluating the information were unnecessary changes, neither making the web page from the iteration prior better nor worse.

Whenever the judge criticized information or functionality mistakes, the cases it named were not actually useful; sometimes even the contrary was the case.

Another interesting point is that in almost every iteration in which the focus was on multiple criteria, no further ma-

With multiple criteria in the prompt, the judge was not able to identify further manipulation that was still present.

nipulation was removed. The only case in which this did happen was in iteration two in *Theguardian*. Every other iteration in which the judge had to focus on all three criteria, it did not remove any further manipulation, even if some remained, only returning feedback in the few cases mentioned prior. This is also evident in the low number of iterations, a mean of 2.04. In contrast to *No Criteria*, in which the focus was constantly only on manipulation removal, the number of iterations is higher, with a much higher number of deceptive patterns removed. To be exact, eight times the judge decided more than two iterations are needed, and in all of them, it was due to manipulation that still needs to be removed.

Multiple Judges needed more iteration, and had lower scores in FUNCTIONALITY and INFORMATION.

Multiple Judges, in which each judge focused on one evaluation criterion, had a similar low deceptive removal rate as *Criteria in Prompt*, but a higher percentage removed (54.667% vs. 46.67%) and success rate (55.56% vs. 48.15%), showing more successful web pages, but also more that had no patterns removed at all. FUNCTIONALITY and INFORMATION were identical, and both were worse than the other two pipelines. DP ADDED and DESIGN both received good scores, with only one to two mistakes, which is identical or marginally worse compared to the other two pipelines, respectively. This pipeline needed more iterations, with four web pages not even finishing in time. As a comparison, the other two always stopped in time. The overall score is lower than both other variants, with a mean of 2.67, but with a higher success rate, namely 44.44%. This shows it has more ideal results, but more severe mistakes, i.e., scores of "1".

Multiple judges sometimes contradicted each other, and undid changes of other judges.

The different judges did not always agree, and instead contradicted each other and ran back and forth for some changes. I.e., the judge for manipulation removed something, one of the other judges added it back, alternating between the two for multiple rounds. For example, in *Expedia*, the judge for DP REMOVED, removed Confirmshaming, the judge for INFORMATION, added it back in, just for the former to remove it again. This happened multiple times, for one, inflating the amount of iteration needed, but also enhancing the chances for mistakes.

Just as the judge surrounding manipulation does not catch every manipulation, evident in the not-perfect DP REMOVED score, the other judges also did not catch everything that they were responsible for. Actually, this pipeline had an even lower FUNCTIONALITY and INFORMATION score than the other two, showing it actually made no improvement in both criteria. Instead, either they did not catch mistakes or were overruled by another judge who redid the mistake that they fixed. For example, when one deceptive pattern was removed that compromised one of the other criteria, and they caught it, the judge for DP REMOVED often redid the same change to remove the manipulation again. Additionally, similar to only one judge with all criteria, sometimes false information was added back in. For instance, the exact same mistake that one judge with all criteria made, in which it added text in the wrong listing elements in *Booking*, happened here as well. This mistake was made by the judge for FUNCTIONALITY, but it was also not caught by the judge for INFORMATION.

The judges for FUNCTIONALITY and INFORMATION did not catch all mistakes, were often overruled in their changes, and suggested wrong changes.

Lastly, including three judges instead of one not only increases the amount of iteration needed, but also the time needed for a single cycle for one web page. While for one iteration in the one-judge scenario needs only two LLM runs, we need four in the one with three judges. For *Booking*, we timed the duration from start to finish. While one judge with multiple evaluation criteria needed around 5:47 minutes, multiple judges needed around 25:25 minutes. It is important to note that the former needed only three iterations and the latter five, resulting in 1:55 minutes per iteration versus 5:05 minutes per iteration. We do not know what went on internally, so other factors might have affected these times, making this just an interesting difference we noticed.

Multiple Judges needs more time to run.

Learnings Summarizing, *Criteria in Prompt* did not improve the pipeline. It had problems from the second iteration on, when the various criteria were included. Struggles were mainly in detecting problems for all three criteria. Additionally, not all the fixes it suggested for the two new criteria were actually useful. The pipeline *Multiple Judges* also did not work well. Multiple judges were hard to coor-

textitNo Criteria was still the best performing pipeline.

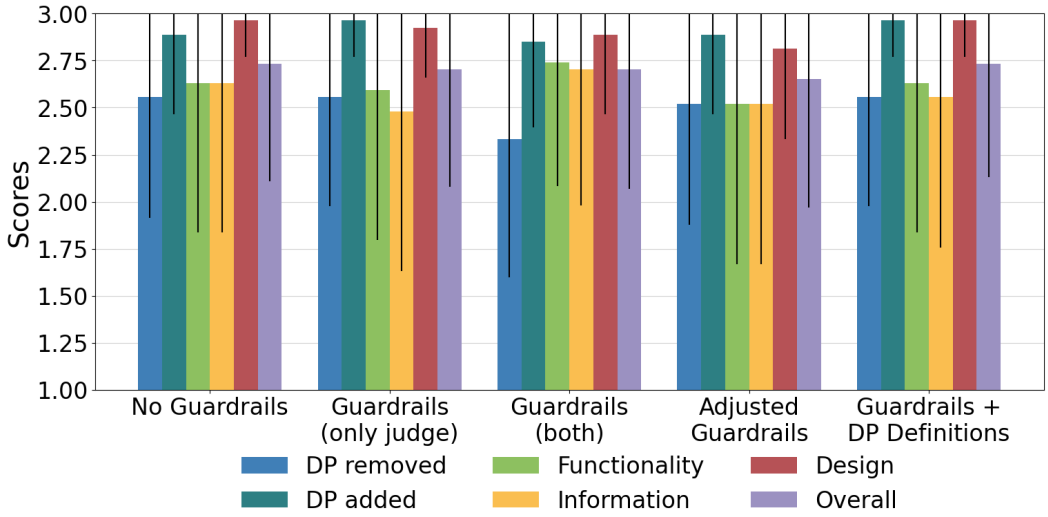

Figure 3.13: The figure shows the mean and standard deviation for the ratings for every evaluation criterion and the overall score for the different ways to include guardrails. The scale ranges from 1 to 3, and "3" is the best. *Guardrails (both)* improved in FUNCTIONALITY and INFORMATION, even though it achieved the lowest DP REMOVED scores.

dinate, as they often did not work well together, undoing changes another one had made, partly running back and forth. They also failed to make an actual improvement in the FUNCTIONALITY and INFORMATION, with the resulting scores being even lower than for both other pipelines. As a conclusion, we will continue with the pipeline *No Criteria*.

3.2.5 Guardrails

We evaluated different pipelines with variations of prompts that included different guardrails. We compare this to the most promising version with no prompt, which is still the *"Judge: o4-mini, Generator: Gemini Thinking"* from the start. We based the initial guardrails on mistakes we identified in the results of that pipeline. We used those guardrails then, once only in the prompt of the judge, as we noticed the generator had little autonomy, and then once in both prompts. Based on these results, we adapted the prompt according to the problems still identified and

compared these results again. The results can be found in Figure 3.13.

To construct our guardrails, we first analyzed what went wrong in the results of the version with no guardrails. Mistakes included not noticing and identifying all deceptive patterns, changing functionality, or removing non-manipulative elements, functionality, or items. It also hallucinated, did not understand a web page correctly, or did not notice web page flaws, or added buttons without functionality. Based on these, we developed six rules that we included in the prompts for either both or only the judge. The prompt can be found in Appendix B.

We based the guardrails on mistakes noticed prior, such as hallucination or removing elements.

Comparing the results of the two versions with the basic guardrails with *No Guardrails*, we can see that *Guardrails (both)* had the worst deceptive pattern removal rate with a mean of 2.33, while most other pipelines achieved around 2.56. The percentage of deceptive patterns removed was also relatively low, with 49.33% compared to 70.67% for *No Guardrails*. *Guardrails (only judge)* did not yield a difference in the number of deceptive patterns removed compared to *No Guardrails*, but an identical mean of 2.56, showing that it removed more deceptive patterns that were on the same web page. Comparing DP ADDED scores, *Guardrails (only judge)* achieved the highest score, and *Guardrails (both)* the lowest. However, the latter attained the highest scores in FUNCTIONALITY and INFORMATION, with 23 web pages that received a score of "3". While the scores overall are relatively close together, FUNCTIONALITY is the largest difference, besides DP REMOVED, as *No Guardrails* had two more scores of "1" than *Guardrails (both)*. Both versions with guardrails had the same OVERALL score of 2.7, which was slightly worse than *No Guardrails*, which achieved 2.73. *Guardrails (both)* obtained the most web pages that had scores of "3" in every category except DP REMOVED, thus leaving the most web pages with the original, necessary content (37.04%). This adds to 74.07% together with all web pages that achieved a perfect score in every category, meaning that almost three-quarters of all web pages were still showing the same functionality and information, just with possibly not all deceptive patterns removed. This is not optimal, but better than lower scores in other categories,

Guardrails (both) lowered the amount of deceptive patterns removed, but had higher scores in FUNCTIONALITY and INFORMATION.

which make web pages worse due to functionality or information missing. *No Guardrails* achieved this for only 62.96%.

Guardrails (both) broke *Viagogo*. In a rerun, this did not happen again.

The low score in DP REMOVED for *Guardrails (both)* is partly explained by the web page *Viagogo* breaking, which resulted in no deceptive pattern being removed and a score of "1" in every category. As this web page contains 12 patterns, this is also a huge reason for the low percentage of deceptive patterns removed. In another round, when we ran this again, all except two deceptive patterns were removed, and all the other criteria were rated with a "3".

We adjusted our guardrails once slightly, as most problems were already covered, but just not followed.

Based on these results, we continued to evaluate the mistakes and improve the prompt. We based these on the results of *Guardrails (both)*, as these yielded better results for FUNCTIONALITY and INFORMATION. The main thing we noticed was that most problems are already covered by the current guardrails, and the LLMs just do not follow them properly. Such as changing functionality, removing information, and not removing all deceptive patterns. Thus, we only changed one sentence to make it clearer that functionality and information should not be changed or removed, and thus removing a direct loophole in which it should be removed when it is manipulative.

The *Adjusted Guardrails* were inferior to at least one other pipeline in every criterion.

This resulted in a slightly higher DP REMOVED score than the original guardrails in both prompts. However, the FUNCTIONALITY and INFORMATION scores were lower than *Guardrails (both)* and *No Guardrails*. Additionally, the DESIGN was also inferior compared to all three other variations, and hence also the OVERALL score, with a mean of only 2.66, while *Guardrails (both)* achieved 2.7. Only the OVERALL success rate is higher than everyone else's, with 44.44%.

Adjusted Guardrails were no improvement. We then tried adding DP definitions to improve DP REMOVED.

Even though the adjusted prompt was supposed to improve the results achieved with the original guardrails, this did not work, and almost all scores were worse with this prompt. Mistakes more prominent here include hallucination, or changing the meaning of information. E.g., by removing the strike-through but not the old prices in *Booking*. Thus, we deem the original guardrails to be better. How-

ever, the DP REMOVED scores are still relatively low. To try one last time to make this better, we decided to include the definitions of deceptive patterns in the prompts of both the judge and generator. We again used the ontology by Gray et al. [2024] here, and also included its hierarchical structure. The patterns included only cover the ones included in any of the web pages in our dataset. For testing purposes, this should be enough.

The DP REMOVED scores were higher in comparison to the *Guardrails (both)*, but almost the same as *No Guardrails*, only with a lower success rate of 59.26% instead of 62.96%. The FUNCTIONALITY and DESIGN showed an identical score as *No Guardrails*, as well as the OVERALL score, which is 2.73 and a success rate of 44.44%. DP ADDED was higher than *No Guardrails* and *Guardrails (both)*, but INFORMATION was lower than both of the former, with only 2.56. Lastly, it needed more iterations, as the mean is 2.78, and *Guardrails (both)*, for example, achieved 2.26.

Guardrails + DP Definitions did not improve our results, especially compared to *No Guardrails*.

Learnings As the scores of *Guardrails + DP Definitions* are almost identical, but with a lower INFORMATION score than *No Guardrails*, this is not an improvement. We also decided that *Guardrails (only judge)* did not work better than *No Guardrails*, mainly due to a much lower score in INFORMATION, and no improvement in DP REMOVED. *Guardrails (both)* did improve the results in INFORMATION and FUNCTIONALITY, at the cost of a lower deceptive pattern removal rate. The version with no guardrails, on the other hand, still achieved the highest overall score, as well as in DP REMOVED. As we deem FUNCTIONALITY and INFORMATION to be the most important criteria, we are gonna argue for *Guardrails (both)* being the best on our dataset. However, it is important to note that *No Guardrails* is very similar, with only small differences in FUNCTIONALITY and INFORMATION, but a higher amount of deceptive patterns removed.

We deem *Guardrails (both)* to be the best pipeline due to better scores in FUNCTIONALITY and INFORMATION.

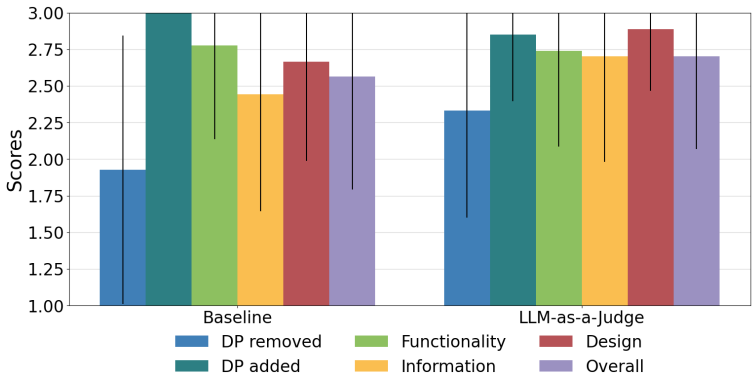

Figure 3.14: The figure shows the mean and standard deviation for the ratings for every evaluation criterion and the overall score for the different ways to include evaluation criteria. The scale ranges from 1 to 3, and "3" is the best. *No Criteria* performed the best in DP REMOVED, *Criteria in Prompt* the best in FUNCTIONALITY and INFORMATION.

3.2.6 Baseline & LLM-as-a-Judge Comparison

We will now compare the baseline results with our final LLM-as-a-Judge results. Our final pipeline uses o4-mini as the judge, Gemini 2.5 Flash (Thinking) as the generator, and uses the basic guardrails for both LLMs. It does not include any prompting strategies, except *Persona*, or evaluation criteria, except asking to make it less manipulative. Further, the judge provides feedback that the generator then applies. The final pipeline is the *Guardrails (both)* version from the section prior. The mean and standard deviation for both approaches can be seen in Figure 3.14. We will also go into further details on what went particularly well in either approach, and what problems still occurred. We also connect this to common problems noticed while adjusting our LLM-as-a-Judge pipeline.

LLM-as-a-Judge noticeably increased the number of deceptive patterns removed, while the baseline achieved a mean of 1.93; LLM-as-a-Judge was able to raise this to 2.33. The percentage of all deceptive patterns removed was only 20% for the baseline and 49.33% for LLM-as-a-Judge. Similarly, the INFORMATION score was also improved, especially no-

ticeable in the much higher success rate of 85.19% instead of 62.96%. The last distinct improvement is for DESIGN. LLM-as-a-Judge had a very high score of 2.89, while the baseline run only received a rating of 2.67. However, the scores for DP ADDED and FUNCTIONALITY were slightly higher for the baseline run. For the latter, the difference is only marginal, as LLM-as-a-Judge only received one score more of "2". The difference in DP ADDED is slightly larger, with a success rate of only 88.89% for LLM-as-a-Judge in comparison to 100% for the baseline model. The overall score was improved in comparison to the baseline, which is also noticeable in the success rate, as for LLM-as-a-Judge, ten web pages came out with scores of only "3", while the baseline only had four of those. Lastly, LLM-as-a-Judge used fewer iterations. For the baseline, we set the number of iterations to 3. LLM-as-a-Judge only needed 2.26 on average.

LLM-as-a-Judge improved noticeably in DP REMOVED, INFORMATION, and DESIGN, while performing slightly worse in DP ADDED and FUNCTIONALITY.

A huge improvement in the LLM-as-a-Judge approach was the consistent recognition of fair web pages as actually fair, and thus not applying any changes to them at all. In comparison, the baseline model changed the fair web pages to the worse in four out of six cases. The worst adjustments it made were changing the delivery date in amazon fair, which it justified with the guardrail seven that stated "Never change facts", as well as removing a "favorite"-button. For the latter, it even stated that the *"HTML code provided does not appear overtly manipulative" (Zalando, i1)*, but then went on to change the web page and remove the button.

LLM-as-a-Judge improved by always recognizing fair web pages as fair.

LLM-as-a-Judge was able to mitigate way more deceptive patterns than the baseline model. However, it also did not remove all manipulation. Our final pipeline was a bit worse in this regard than other pipelines we tested. Next to Trick Question, Forced Registration, and Hidden Costs, that we will explain in detail below, another common problem was Hidden Information, which was never removed in *Telegraph* or *Riverisland*. The latter was removed at least by some pipelines, however, not in the first run on our selected one. Other deceptive patterns barely removed by any pipeline, and also not in the final one, include False Hierarchy in the cookie banner on *Viagogo* or in *Opodo*, the Testimonial in *Pelacase*, or the Disguised Ad in *Amazon*. In con-

Some deceptive patterns were never removed by an LLM-as-a-Judge pipeline.

Social Engineering pattern worked predominantly well in LLM-as-a-Judge.

trast, False Hierarchy was almost always removed in other web pages, such as *Aliexpress*, *S_falseHierarchy*, and *Ryanair*, and Testimonials in *Expedia*. In contrast, deceptive patterns commonly removed by LLM-as-a-Judge include Social Engineering patterns, such as Confirmshaming or Low Stock, but also Bad Defaults and Positive Framing. The baseline never removed the web pages problematic for LLM-as-a-Judge either, and also struggled with other patterns that LLM-as-a-Judge did not, such as Social Engineering ones.

LLM-as-a-Judge could not correctly defuse Trick Question, while the baseline model worked slightly better here.

One of the deceptive patterns, which our LLM-as-a-Judge approach could almost never correctly defuse, no matter which adjustment we made, was Trick Question on both web pages *S_trickQuestions* and *MyTrip*. It consistently changed the opt-out checkbox to an opt-in without adjusting the functionality accordingly. A similar example is shown in Figure 3.5. The final pipeline did not notice the Trick Question in *S_trickQuestions*, but made the same mistake in *MyTrip*. The baseline model correctly mitigated it in *S_trickQuestions*, keeping the original functionality of the button, and not really adjusting anything noteworthy in *MyTrip*.

LLM-as-a-Judge tried to defuse Forced Registration and ended up always removing functionality and information.

Another problem in the LLM-as-a-Judge pipeline we were not able to fix was the Forced Registration and Hidden Costs in *Opodo2*. Forced Registration is something we do not want our LLM to change, as it cannot change the functionality behind it. The most common solution applied by the LLMs was to remove the button and information about the discount, and then just keep the crossed-out original price. This resulted in the web element showing inconsistencies and containing less functionality and information. The baseline, on the other hand, did not change this at all.

Mistakes in INFORMATION were caused when the LLMs tried to remove actual deceptive patterns.

Further problems with the LLM-as-a-Judge pipeline include the removal of information. Overall, it can be said that LLM-as-a-Judge improved this a lot in comparison to the baseline. However, it still made the information worse in four cases. Most of those mistakes were some that occurred frequently in our different adjustments. Next to the one in *Opodo2* we just explained, other information removal was also connected to the LLMs trying to remove actual deceptive patterns, which they did, but in a way that

ended up with information loss. Three of those four mistakes commonly appeared across all the LLM-as-a-Judge pipelines tested. On the bright side, our final pipeline with LLM-as-a-Judge never hallucinated or changed information. This was something that occasionally happened before, highlighting how guardrails might have improved this. This is something the baseline did more often. The same applies to functionality. While the baseline, for example, added a random button, the final LLM-as-a-Judge pipeline never added incorrect functionality. The problems in regard to FUNCTIONALITY are mainly not changing functionality accordingly to other adjustments made, such as in the case of *MyTrip*, or removing it, such as in the case of *Opodo2*. Overall, only on three web pages one of those mistakes was made.

Our final LLM-as-a-Judge pipeline did not hallucination or change information or functionality, it only removed them.

In contrast to the baseline, the LLM-as-a-Judge made mistakes that added manipulation. One of those was that when the generator changed the color of an accept button to green and the reject button to red. A good thing was that LLM-as-a-Judge barely made any mistakes relating to the DESIGN. One of those it did make was a weird alignment in *Eventim*, which occurred commonly across the pipelines we tested. All in all, almost all LLM-as-a-Judge pipelines with o4-mini as a judge made only one to two mistakes in this category. The baseline, however, commonly changed the design. It also did not notice mistakes in the design that resulted in worse readability, such as black text in front of a dark gray background.

LLM-as-a-Judge added more deception in, but it compromised the design less often.

3.3 Discussion

We will discuss the different adjustments we made to optimize our pipeline, the effect they had, and the implications to take from this. Further discussions regarding our research questions will be presented in Chapter 5.

The least promising judge was a general-purpose model, GPT-4o, while the most promising one was the reasoning model o4-mini. Evident in how all three versions in which o4-mini was the judge performed better and received a

Reasoning models performed better as the judge, linking to how reasoning capabilities improve judging.

higher overall score than every other model in the role of the judge. GPT-4o as the judge, on the other hand, performed noticeably worse than any other judge. This generally aligns with the literature, which often suggests incorporating reasoning capabilities into judges to enhance performance [Gu et al., 2024; Li et al., 2024]. Gu et al. [2024] also compared different models as judges, including predecessors of the models we used: GPT-4-turbo, gemini-2.0-thinking, and o3-mini. They noted that GPT-4-turbo was still the judge that aligns the most with humans. Even though the reasoning models showed promising results and advancements, they were not consistent enough. With the newer models and a different task, we noticed more pronounced advancements of o4-mini over GPT-4o. It delivered better results that were relatively consistent. The consistency was especially noticeable over our multiple adjustment rounds, which we will discuss later in more detail.

We did not notice Gemini as the judge improve with reasoning capabilities.

Gemini 2.5 Flash with and without thinking were both promising in FUNCTIONALITY and INFORMATION., but both scores, as well as the OVERALL score, were better for Gemini (without Thinking). This is interesting, as the reasoning model, i.e., Gemini (with Thinking), consequently performed worse. It is different from what we noticed with OpenAI's models, as well as the literature suggesting reasoning to enhance judges [Gu et al., 2024]. These tendencies should be taken carefully, as we only tested a very small sample of web pages from our dataset with both judges. However, it shows that the differences between reasoning and no reasoning are not always as pronounced as we noticed them with OpenAI. Moreover, the only slight differences could be due to this being a comparison between the same model, just with adjusted reasoning capabilities. Thus, the versions are trained on the same dataset. All in all, this tendency connects to what Gu et al. [2024] noted in their comparison between models. However, due to the continuing advancements in reasoning models and the already notable differences between o4-mini and GPT-4o, we hypothesize that reasoning models will develop to generally outperform general-purpose models as judges. At least for deceptive pattern removal in web pages.

Overall, we expect models with reasoning capabilities to outperform general-purpose models.

In the literature, the most common judge is GPT-4 [Szymanski et al., 2025; Zheng et al., 2023]. Reasoning models have not been as widely explored in the literature, which is also due to them being relatively new compared to general-purpose models, with OpenAI's first reasoning model, o1-mini, being released in September 2024[9]. GPT-4 has been shown to be much higher in accuracy in other tasks than even its successor GPT-4o has been in our [Gu et al., 2024]. It is clear that performances of judges vary across tasks. This is related to Szymanski et al. [2025], who tested GPT-4 in expert-knowledge tasks and achieved rather low agreements with experts. Similarly, for us, general-purpose models did not accomplish promising results either. While general-purpose models might be a promising solution for less expert-prone and less specific tasks, for deceptive pattern removal, they are not yet sufficient, at least in the way we prompted them. Further prompting strategies might improve their performance.

The literature has shown high accuracy in many tasks for GPT-4, which we did not have with similar general-purpose models.

General-purpose models might not be qualified enough as judges for specific tasks.

We deemed a reasoning model as the generator to be the most promising as well, specifically, Gemini 2.5 Flash (with Thinking). However, we did not notice as large a difference between the selected generators as we did between different judges. Thus, we assume that the selection of the model for the judge seems to be more important and to have a bigger influence than the model for the generator. At least in our setting, in which the judge has a slightly more dominant role, by providing the feedback that the generator follows. It is also interesting that models for the generator and judge from different companies performed better than other combinations. While we cannot directly connect this to any direct bias, Spiliopoulou et al. [2025] noted that there exists a family bias between models that were trained on the same dataset, and that they consequently prefer the output performed by those LLMs. We do not know if and how much o4-mini's and GPT-4o's training sets overlap. However, as they are both models from OpenAI, we assume that there exists at least a partial overlap, and that might have played a partial role in the models performing best being from different companies. On the other side, it might

The model selection for the judge appears more important than the model for the generator.

The family bias might have played a part in judge and generator resulting in models from different companies.

[9] `https://openai.com/index/openai-o1-mini-advancing-cost-efficient-reasoning/` [Accessed: Sep. 27, 2025]

also just be largely due to Gemini's general capabilities that worked in this scenario.

Persona only marginally improved our approach, other personas could potentially improve this more noticeably.

Looking into prompting strategies, the first strategy tested, *Persona*, did somewhat improve our results. However, the differences are only marginal and have to be taken carefully. Especially because the FUNCTIONALITY and INFORMATION were slightly better for *No Persona*, one could also argue that *Persona* made the pipeline worse in comparison. It is possible that other persona could improve our results further. We chose the same persona as Schäfer et al. [2025] for the generator and adapted it for the judge. However, those personas are relatively unspecific and broad. Other directions could look into personas specific to, for example, deceptive pattern experts. Kim et al. [2024] discussed the difficulties of finding the correct persona, so it is possible that different ones could improve our pipeline further.

Few-Shot Prompting improved the amount of deceptive patterns removed, but the LLMs did not show the ability to transfer the changes performed in our examples to ones they had to perform.

Few-shot Prompting generally did not enhance our performance, due to mistakes in FUNCTIONALITY and INFORMATION. However, it received the highest score in DP REMOVED across every pipeline we tested on the full dataset. Additionally, it removed patterns such as Hidden Information that were not removed by any other pipeline. This hints that *Few-Shot Prompting* can enhance the ability to detect deceptive patterns. But the LLMs lacked the ability to transfer the way changes were performed in the examples given to the web pages they actually had to change, thus it does not improve the actual removal. This was obvious in patterns such as Trick Question and Forced Registration. This resulted in changes that impaired the web pages, especially in regard to FUNCTIONALITY and INFORMATION. Due to this lack of transfer, *Few-Shot Prompting* was less successful than *Zero-Shot Prompting* for us. Given that the changes were performed by the generator, and the detection was mainly done by the judge, we see further potential in a pipeline that uses *Few-Shot Prompting* for the judge, but not the generator to increase the amount of deceptive patterns removed.

While *Few-Shot Prompting* did not work for us, there are multiple ways to adjust this approach, which could possibly improve the results. It is possible that we used too few

examples, as four is a relatively low number. A higher variety of deceptive patterns could also improve the results. However, the way deceptive patterns are implemented in web pages varies heavily, and new patterns can always arise. It is thus hard to define which deceptive pattern types should be included, as well as what instances of each type. Additionally, new types are likely to develop [Gray et al., 2024], and we want the LLMs to have the ability to adapt to new ones. It is also possible that our examples were not fitting or that they were too unrealistic, as we implemented them in a very simple manner. While all these changes could possibly enhance this procedure, the missing transfer ability of the LLM could potentially hinder this approach overall.

There are multiple ways to improve *Few-Shot Prompting*, but the LLMs lacked the general ability to convey the examples.

Similar to *Few-Shot Prompting*, *Chain-of-Thought* performed worse in terms of FUNCTIONALITY and INFORMATION, while only slightly improving the amount of deceptive patterns removed. These differences are again, similar to *Persona*, very small. So *Chain-of-Thought* also did not have a good enough positive effect on our dataset, especially not in combination with *Persona*.

Chain-of-Thought also did not have a noticeable enough effect.

All prompting strategies did not or only slightly improve our results. Overall, the prompting strategies we tested aim to increase the in-context learning ability and the reasoning capabilities of LLMs [Gu et al., 2024]. As we already chose reasoning models for both the judge and generator, they already had reasoning capabilities included. Further, specifically *Chain-of-Thought* is something reasoning models do internally[10]. Nori et al. [2024] tested *Few-Shot Prompting* for o1, a predecessor of o4-mini, and found that it reduced the performance for their medical-related task. A similar thing happened in our task. Thus, these prompting strategies might be a promising solution to enhance the reasoning of general-purpose models, and might yield even better results than o4-mini did in general in such a constellation. However, as we only used reasoning models, none of them had a noticeable positive effect for us. There are further options to improve the reasoning of reasoning models, such as

Prompting strategies did not not work notably, which might be due to us using reasoning models.

There are further options to enhance reasoning of reasoning models, that might improve our performances further.

[10] `https://openai.com/index/openai-o1-system-card/` [Accessed: Sep. 29, 2025]

changing the amount of reasoning tokens [Nori et al., 2024], which is something we did not look into.

Both approaches to the pipeline, either using no feedback or encouraging the generator for more critical thinking (autonomy), did not improve the pipeline, as they either removed way fewer deceptive patterns, or had lower scores for FUNCTIONALITY and INFORMATION. Interestingly, even in the pipeline *Feedback + Autonomy*, the generator failed to actually think critically, only departing from the feedback in four different cases, not all of them actually being useful. Thus, the addition to the prompt was not actually successful. It shows that the main detection of deceptive patterns is the responsibility of the judge, at least for the way we defined our pipeline. This is also prominent in the no feedback pipeline, in which the amount of deceptive patterns removed is lower, as now the responsibility what is removed is fully on the generator. Consequently, it is possible that the low score for *No Feedback* is partly due to the lower ability of Gemini-2.5.-Flash (Thinking) to detect deceptive patterns. In contrast, our good performances in removing the deceptive patterns might be mainly due to o4-mini's good performances. On another note, while the judge partially explained exactly how to change something, this was not always given. This splits the responsibility of how something is changed between both LLMs.

The detection of deceptive patterns is largely based on the judges capabilities.

When we added the evaluation criteria to the prompt of one judge, we also could not improve our pipeline. The low number of iterations we explained in the results shows that with multiple criteria, the judge had more difficulty actually detecting manipulation. Additionally, it did not detect a lot of mistakes regarding FUNCTIONALITY and INFORMATION, and when it did, it was not always something that was actually a mistake. Hence, we conclude that the ability to detect mistakes from numerous criteria at once is difficult for the judge, as he has to focus on multiple at once. This is something that Patel et al. [2024] also reported, as they stated that a single judge might not notice all mistakes, specifically for more elaborate tasks, in their case code generator, and is exactly what we noticed as well.

Once multiple criteria were included, the judge struggled to detect any, the lack of focus negatively affecting results.

However, in contrast to Patel et al. [2024] reporting that multiple judges improved their results, for us, *Multiple Judges* resulted in even lower scores. One of the main problems was that the judges contradicted each other, overruling the changes another one made in a later iteration. We assume that this is mainly due to the unclear borders between what each criterion covers. Deceptive patterns often contain information, such as the number of items remaining in Low Stock, which is lost when removing the deceptive pattern completely from the page. Depending on the focus, this can be either important information or just manipulation. Even though we made clear that manipulative information should not be added back in in the prompt, this is still kind of overlapping. This is different from the criteria often tested in the literature, such as various errors in code, in which multiple judges were indeed successful [Patel et al., 2024]. Thus, this is likely the problem in our task, and the reason why this approach does not work well. Additionally, the approach with multiple judges also had problems detecting all mistakes in FUNCTIONALITY and INFORMATION. This was already also noticeable in the approach with the criteria in the prompt of one judge. Showing that the LLM is potentially just not yet able to fully detect all the mistakes, and classify those as actual mistakes.

Multiple judges, splitting the focus, did not improve performances. Our criteria might overlap too much, resulting in judges overruling each other.

LLMs might not yet be able to detect all errors in FUNCTIONALITY and INFORMATION.

Guardrails showed improvements in specific criteria for us. The benefit is only very marginal, though, and one could also argue that the improvements in these categories come with a tradeoff in DP REMOVED, which then makes the pipeline not better. As we deem FUNCTIONALITY and INFORMATION more important, we conclude that guardrails had a positive effect, and for Schäfer et al. [2025], they worked even more noticeably. Guardrails can be formulated in various ways, and adjusting them further might be something that could help improve the pipeline.

Guardrails can have a positive impact, and could be adjusted further.

The tradeoff between DP REMOVED and both FUNCTIONALITY and INFORMATION is something that we noticed more often. Once the LLMs tended to remove more deceptive patterns, both functionality and information suffered. This hints at how the LLMs might not be able to remove specific deceptive patterns in a way that does not damage

We noticed a tradeoff between DP REMOVED and both FUNCTIONALITY and INFORMATION.

the other two criteria. This could be due to the nature of the deceptive patterns, i.e., they might not be removable at all or not automatically removable by an LLM. This includes Forced Registration or Hidden Costs. For some patterns, such as Trick Question, we would argue that it is possible for an LLM to change it, but the LLMs lacked the ability to do this sufficiently in our setting. With further improvements, this tradeoff could possibly be minimized.

We tested everything only once, thus, all results have to be taken carefully, especially considering the results with datapoints closely together.

After all the comparisons, it is important to note that many scores were relatively close together. Most of the results we presented are more tendencies. Especially, because with LLMs, it is always important to keep in mind that they are not deterministic. They show different results even with the same prompt, which also happens for Schäfer et al. [2025]. As we only tested everything once, we cannot exclude the possibility that some scores are not different due to different pipelines, but instead based on the non-deterministic nature of LLMs. This is especially important for the results in regard to the prompting strategies, as well as the guardrails, as the results are very close together, often differing in the scores by only one to three points for all web pages.

Chapter 4

Study

This chapter describes the user study, which follows up on our technical implementation in Chapter 3 and the results we achieved from it. We will first describe the considerations taken and the final study setup, followed by the depiction of the results we gathered during our study.

4.1 Method

With the user study, our primary goal was to investigate user agreement with the judge, as well as the users' impressions of the changes made by our pipeline. Overall, we aim to answer the following two research questions, already introduced in Chapter 1:

RQ2: How well does the LLM-as-a-Judge approach align with the judgment of users?

RQ3: How do people perceive the changes made by our LLM-as-a-Judge approach?

A common approach in the LLM-as-a-Judge literature to compare human judgment with the judgment of LLMs is to give both the same task and then compare the results and calculate the agreement [Szymanski et al., 2025; Wang

The literature often looks into human alignment with LLMs, which is what we will do too to answer RQ2.

et al., 2025]. This is also part of what Kocyigit et al. [2025] did in their expert user study to evaluate a singular LLM in detecting deceptive patterns. Similarly, we answer RQ2 by asking participants to adapt the web pages to make them less manipulative, and we then compare the results with the LLM output.

We hypothesize that users might have different alterations that they would like, besides the ones they would perform themselves. Thus, we want to look into users' preferences as well.

Limitations of this approach are that it results in one correct solution for each participant per web page, and every other change is considered incorrect. However, we hypothesize that participants potentially used changes that might not be exactly how they would prefer deceptive patterns to be removed, but they would still accept or even like them better than the version without changes. Based on this, we opted to also evaluate how participants perceive the changes made by our LLM-as-a-Judge pipeline, answering RQ3.

Our user study consisted of one task in which users had to alter web pages, one in which they had to rate the altered web pages, and a semi-structured interview.

This leaves us with two different tasks in our study. Task one, in which participants had to alter web pages to make them less manipulative, and task two, in which participants had to rate the results from our LLM-as-a-Judge pipeline, which are the already altered, potentially less manipulative web pages. The order of the tasks was chosen so we do not bias the participants with changes the LLM performed on any web pages, but instead let them come up with adjustments on their own. We are aware that task one could potentially bias the ratings in task two, which we deem as less severe than the former one. We ended with a semi-structured interview. We also did not tell participants that the alterations were made by an LLM, to not bias them. We clarified this during the interview once we were done with all tasks and questions, which we did not want to be biased with users' potential opinion of LLMs.

4.1.1 Dataset

For the user study, we needed a dataset of web pages that the participants would be asked to change, but also rate. We did not want to bias the users in task two with their own changes they performed in task one. That's why we

decided to use distinct web pages in tasks one and two for each participant. However, we agreed to use all web pages for both tasks, i.e., change the web pages used in each task for each participant. Even though this means that we got fewer data points for each web page in each task, we got more web pages in each task to evaluate. This also eliminates the potential bias that is created by splitting the items in the dataset into two groups, possibly choosing the easier, better-adjusted, or less controversial web pages for one task. Most importantly, we can compare each web page's rating between the two tasks, for example, to compare whether adjusted web pages that potentially have a low alignment with humans are still generally preferred over the original web page.

We used all web pages for both tasks, so we can compare the results for both tasks for each web page.

We only included real web pages or web elements in the study, as those are the most complex ones, and the actual use case scenario of an application. When choosing the pipeline results to compare to the human results, and which to let participants rate, we decided to use the version that includes the first version of guardrails in both prompts (Chapter 3.2.5). The reason for this was that the pipeline worked best on our dataset, as functionality and information were best kept. We looked into the dataset arranged for the technical evaluation in Chapter 3.1.1, and chose all real web elements or web pages that were relatively well adjusted by the chosen pipeline. We also included items that did not achieve a perfect score in our evaluation. We excluded four real web pages that either had no deceptive patterns removed at all or had severely changed or removed functionality. In the original run for this pipeline, *Viagogo* and *Theguardian* were either broken or had no manipulation removed. As the former is one of the two larger web pages, and the latter had the distinct deceptive pattern *Nagging*, we wanted to include them. That's why we ran them again through the same pipeline, this time achieving better results, which we then used in our user study. We did not include any fair web page, as the pipeline did not change them at all. Overall, we accumulated 11 web pages: *Aliexpress*, *Amazon*, *Booking*, *Expedia*, *Gotogate*, *Opodo*, *Pelacase*, *Riverisland*, *Ryanair*, *Theguardian*, and *Viagogo*. The patterns included in each web page can be seen in Appendix A.

We used the pipeline with guardrails in both prompts, as it achieved the best results.

We used eleven web pages from our initial dataset.

We handed out five web pages in task one and six in task two for each participant. We used a 22x11 Latin square to counter order effects.

We wanted each participant to have each web page in either one of the two tasks; thus, we decided on a within-subject study design. We deemed 11 web pages to be a realistic amount for one participant in a reasonable time, which was confirmed in a pilot study. As we expected the first task to take longer, we chose to give each participant five web pages for the first task and six web pages for the second task. We used a balanced 22x11 Latin square to counterbalance any order effects, which also helped to spread the web pages relatively evenly across both tasks. Due to the odd number of web pages, we ended up with 22 orders of web pages.

4.1.2 Questionnaires

We now describe the questionnaires and considerations that went into them. All questionnaires are included in Appendix D. We started with a consent form, followed by a demographic questionnaire. We then used one questionnaire for tasks one and two, with one questionnaire covering one web page. Thus, the questionnaire for task one had to be filled out five times, and the one for task two, six times.

We collected general demographics, as well as asked questions to asses participants' expertise regarding deceptive patterns.

In the demographics form, we asked participants for their age, gender, and current occupation, as well as their last achieved academic degree. This helped to gain an understanding of the general background of each participant. We were then interested in the expertise of the participants in the field of deceptive patterns. For this, we asked questions about their knowledge surrounding deceptive patterns in general, the research surrounding them, as well as the general awareness and measures users take to avoid them on websites. We used 5-point Likert scales for these questions. Afterward, we asked for further information on how participants have already engaged with deceptive patterns to further assess their expertise. We are aware that those are self-reported measures and that we cannot fully rely on the answers given. However, we hope to get a general understanding of whether a participant is a novice or has a good understanding of the deceptive pattern research.

Task one was the alteration of web pages by the participant. In the literature, the LLM and the participants are often given the same task to ensure the best possible comparison of results [Szymanski et al., 2025; Wang et al., 2025]. This is the reason we wanted to give our participants a relatively similar task to what our LLM pipeline did. We did not specifically ask the participants to do the same task as the judge, as the feedback from the judge was spread over multiple iterations, and is just a representative of the changes actually made, since the generator turned out to have little autonomy. Thus, we wanted to simply ask the participants to provide us with the changes. We opted to give the participants the task *"Make that less manipulative"*, which is part of the prompt given to the generator. We did not include the six guardrails given to the generator, as we deemed them self-evident to humans. As the LLMs did not receive an explanation of deceptive patterns, we did not give our participants one either. The same reason applies to why we did not show our participants the deceptive patterns on the web pages. Additionally, we also did not want to pressure them into feeling like they have to remove all patterns, even if they might not want to. The participants were then asked to write the changes down as notes or draw the altered web page. We decided to include both options, as some people might find it easier to draw the changes instead of writing them down, while some changes might not be possible to draw, such as changing colors. To better understand the changes and reasoning behind the adjustments, we asked for justifications. We then asked participants whether there were things they chose not to change, to distinguish between the manipulations they purposely kept and those that they did not see. Lastly, we asked for further comments.

Participants were tasked to "Make that less manipulative", which is similar to the prompt given to the LLM.

We asked participants to write or draw the changes, and justify them afterward.

For task two, participants were asked to rate the altered web pages. We first asked the participants how manipulative they perceived the original web page to be, followed by a question about how severe they thought this manipulation was, and asked them to justify their ratings. They were then asked to rate the altered web page using the same criteria we evaluated the web pages with in Chapter 3. The only exception is DP ADDED, as this case could be answered with other scales, if necessary. We opted for the

We asked participants to rate how manipulative the original web page was, to rate the altered web pages on the evaluation criteria, and state their preferences.

same evaluation criteria due to the same reasons that we explained in Chapter 3.1.2. We found these to work well in the evaluation we did in Chapter 3. The criteria were phrased as follows:

- All manipulation that should be removed is removed.
- All functionality that should be kept is kept.
- All information that should be kept is kept.
- The design wasn't influenced in a negative way. (This rating doesn't consider any design changes necessary to remove manipulation.)

We used 5-point Likert scales, to get a more fine-grained overview.

We then again asked for justifications of those ratings. All questions up to now, except the justifications, were to be answered on 5-point Likert scales *(Strongly disagree - Strongly agree)*. In Chapter 3, we used 3-point Likert scales; however, as we evaluated the web pages, we noticed that 3-point Likert scales are missing a more detailed split between web pages receiving a rating of 2. This was fine for the evaluations in Chapter 3, since we also collected the percentage of deceptive patterns removed, looked into the qualitative data as well. Additionally, we aimed to optimize each criterion to a perfect score anyway, and thus did not need a more precise split. However, for the user study, we deemed 3-point Likert scales to be too imprecise, and thus decided on 5-point Likert scales. Lastly, we asked participants whether the altered web page feels better or worse and if they would prefer to use it instead of the original version, again using 5-point Likert scales. We asked for justification once again. In the end, a field for further comments was provided.

4.1.3 Study Procedure

We used two monitors to show participants the web pages.

Our study setup consisted of two monitors, on which we showed the participants the web pages in tasks one and two, and the participants could interact with the web page on their own. Two monitors were needed to show the original and altered web page next to each other for task two.

The questionnaires were printed and were to be filled out by hand. The reason for this was to allow participants to draw the changes in task one, which would not have been possible had the questionnaires been online.

At the start of the user study, we explained the consent form to the participants, allowed them to read it themselves, and asked them to fill it out. Once this was done, we handed them the demographic questionnaire.

Afterward, we started with task one by introducing the task through a short explanation and answering further questions about it. Then the participants were presented with the first web page, which they were asked to make less manipulative, and handed the questionnaire for this web page. If necessary, we showed participants interaction possibilities or explained unclear or hidden elements. We then let participants interact with the web page on their own and answered questions that came up. This was repeated for all five web pages for the first task, collecting each questionnaire before handing out the one for the next web page. After this task, we offered a short break.

Participants answered the first questionnaire for five web pages, we showed them possible interactions and answered questions.

Following up with the next task, we again explained it first and answered questions. We then started with the first web page for this task by showing the original and altered versions simultaneously, and explaining interaction possibilities and unclear elements. We also handed out the questionnaire for this task, and let the participant interact with both web pages on their own. We repeated this for all six web pages and offered a short break afterward.

In task two, participants answered the questionnaire for six web pages.

In the end, we conducted a semi-structured interview, audio-recording it if the participants had agreed to it in the consent form. We started the interview by asking questions related to the two tasks. Specifically, if they removed anything, or specifically did not, in task one, and if they noticed anything during the web pages in tasks two that they thought to be particularly positive or negative. We also asked questions here if we noticed something specific that the participants did during the execution of either task. We then asked participants to rate the evaluation criteria *DP removed*, *Functionality*, *Information*, and *Design*, from most to

At the end of the study, we conducted a semi-structured interview, in which we also told participants that these changes were performed by an LLM.

least important. We followed this by asking whether they could think of any other useful criteria to rate such altered websites. Subsequently, we asked whether or not participants would use such an application, and if they could think of any modifications or features to improve it. Up to now, we had not mentioned that the alterations were made by an LLM, but clarified it at this point. We did not provide details on the LLM-as-a-Judge pipeline, as this might be unnecessarily complicated. Instead, we explained how an LLM was given the HTML, which it changed to be less manipulative. Consequently, we asked participants if that changed their attitude towards such an application, how much they would trust it, and how tolerant they would be if the LLM made mistakes. After asking for further comments, we ended the interview and thanked participants for their time.

The study was set out to take around 90 minutes to 2 hours, which was confirmed during a pilot study.

4.1.4 Data Analysis

For the data analysis, we used qualitative and quantitative methods. For qualitative analysis, we used MAXQDA[1]. We will now explain our procedure in more detail for each task.

We coded the changes performed by users in task one, and then identified the majority opinion among all changes.

For task one, we decided to use the majority opinion of the changes made. This is inspired by Kocyigit et al. [2025], as they also used the majority opinion of all experts when evaluating the agreement rate between the LLM and experts in the context of classification of deceptive patterns. We perceive this as useful, since such an application should also act in a way that works best for most people. To gather the majority opinion, we coded the changes and then picked the most applied codes for each instance mentioned. If people did not mention any changes, and did not specifically mention that they chose not to change it, we counted this as them wanting no changes done to it, and took this into account when identifying the majority opinion. If a majority was split between two changes, we

[1] `https://www.maxqda.com/` [Accessed: Sep. 29, 2025]

identified the one that was closer to the other changes users performed.

Once we identified the majority opinion, we calculated the agreement between the LLMs and the participants for the whole web page, and for each deceptive pattern and change made to the web page on their own. More specifically, for each web page, we calculated how many decisions of the LLM made overlapped with the participants' decisions. We call this the *Agreement Rate* for each web page. Then we also calculated for each change that the LLM made how many participants made the same change, which is the *Agreement Rate* for each deceptive pattern instance. Additionally, we calculated the recall and precision for each web page. We identify TP, FP, and FN as follows:

We calculated the agreement between the LLM and the users for each whole web page and each deceptive pattern on all web pages, as well as the recall and precision for each web page.

- TP (true positives): the LLM and the majority applied the same change
- FP (false positives): the LLM changed something that the majority did not, or the LLM changed something in a different way than the majority
- FN (false negatives): the majority changed something that the LLM did not change

$$Recall = \frac{TP}{TP + FN}$$
$$Precision = \frac{TP}{TP + FP}$$

A low recall shows that the pipeline either made too many changes or applied changes in a different way. A low precision indicates that the pipeline applied too few changes.

The main result from task two was the quantitative data from the rankings. We calculated the arithmetic means, as well as the standard deviations, for each web page using Microsoft Excel[2]. We also coded the justifications.

We calculated means and standard deviation for the ratings in task two.

We transcribed the interviews using a locally running version of the transcription tool Whisper[3] from OpenAI. We

We transcribed and then coded the interviews as well.

[2] `https://excel.cloud.microsoft/` [Accessed: Sep. 29, 2025]

then coded the interviews and extracted more quantitative data, such as percentages and ratings mentioned.

4.2 Results

4.2.1 Participants

15 people participated, most of whom had a technical background.

Overall, 15 people participated in our user study (8 male, 7 female). Their ages ranged from 20 to 32 (M = 23.6, SD = 3.33). 12 participants were university students, all of them studying computer science, one was a research assistant, one an AI consultant, and one a nurse. Everyone except one participant had a technical background. Ten participants named the high school diploma as their highest level of education, two had a bachelor's degree, and three had a master's degree.

Most participants were somewhat familiar with deceptive patterns.

Many participants have participated in user studies on the topic of deceptive patterns before.

Most participants knew what deceptive patterns are, with M = 4.07 (SD = 1.22). Less people were very familiar with the research around deceptive patterns (M = 3.13, SD = 1.68), and a few more had engaged with the topic before (M = 3.53, SD = 1.6). Fewer people agreed to avoiding deceptive patterns on web pages (M = 3.33, SD = 1.23) than stated that they noticed them (M = 3.67, SD = 1.4). The most common ways participants have engaged with the topic of deceptive patterns were participating in other user studies (n = 7), reading papers (n = 3), publishing or helping in the publication of a paper (n = 3), writing a thesis (n = 3), or noticing or being aware of deceptive patterns on websites (n = 5).

We will translate all answers received in German into English.

While all participants received the questionnaires in English, all but one conducted the interview in German. The questionnaires were filled out in both German and English. In this thesis, we will translate all answers provided in German to English.

[3] `https://github.com/openai/whisper` [Accessed: Sep. 29, 2025]

4.2.2 Part 1 - web page Alteration

We will now talk about the results we obtained from task one. All web pages were altered by six to eight people. We first start with the agreement for each web page, then go into more details regarding each deceptive pattern, and end with adjustments made that do not directly relate to deceptive patterns.

Agreement for Whole Web Pages

web page	A	B	C	D	E	F	G	H	I	J	K
Agreement (%)	100	25	85.71	16.67	40	30.33	60	50	16.67	14	14.29
Recall	1	0.33	0.86	0.25	1	1	0.6	0.75	0.17	0.14	0.17
Precision	1	0.5	1	0.33	0.4	0.33	1	0.6	1	1	0.5

Table 4.1: Agreement rate, recall, and precision of each web page. The web pages are as follows: *A: Aliexpress, B: Amazon, C: Expedia, D: Gotogate, E: Opodo, F: Pelacase, G: Riverisland, H: Ryanair, I: Booking, J: Theguardian, K: Viagogo*

The agreement rate between the LLM-as-a-Judge pipeline and the majority voting from our participants for each web page, as well as the recall and precision, can be seen in Table 4.1.

Only one web page, *Aliexpress (A)*, achieved an agreement of 100%, which contained the singular deceptive pattern *False Hierarchy*, that was removed by the LLM by making both elements the less prominent color. Besides *Aliexpress (A)*, only three other web pages achieved an agreement rate of 50% or above. Seven web pages had an agreement of below 50%, with a minimum agreement of 14% for *Theguardian (J)*. As the means are so low, it shows a relatively low agreement overall.

Most web pages received an agreement of 50% or lower.

Eight web pages obtained a recall below 1, showing that the LLM made changes that the majority of participants did not want, or that they would change the elements differently. While five web pages achieved a precision score of 1, six achieved lower scores, varying from 0.33 to 0.6. This indicates that our pipeline also did not change or remove ele-

The LLM often changed something users did not or did differently, sometimes, it did not remove something that users removed.

ments that the majority opinion of users wanted removed. There are more web pages with a recall lower than one, and four really low recall values that are below or equal to 0.25. This shows that it happened more often that the LLM removed something it should not have removed or changed it differently than how the majority of users would change it, than the LLM did not change something it should have.

Agreement for Individual Deceptive Patterns

We will now look with more detail into the individual deceptive pattern types on the web pages, specifically the agreement between how the LLMs removed each pattern and how the users would want it adjusted or not. The results for each deceptive pattern on each web page can be found in Table E.1 and Table E.2.

False Hierarchy and Visual Prominence received the highest agreements when the LLM made all elements the less prominent option, or when adding further elements. However, the results depend on the context of the website and the elements adjusted.

Starting with *False Hierarchy* and *Visual Prominence*, a few elements were adjusted by the LLM pipeline by making both elements the less prominent color, which included two buttons or two items. This generally received some of the highest agreement rates by users, ranging from 57.14% (A) to 100% agreement (C, H). The only case in which the LLM removed *False Hierarchy*, by changing the elements to look as the more prominent option, was in *Gotogate (D)*, in which it was not about colors, but instead an element that was less prominent by size. This achieved an agreement of 50%, showing a split within the participants. Adding elements, such as prices or information, to make two elements visually equal, received agreements ranging from 33.33% (D), to 57.14% (C), and 85.71% (G). The highest agreement was achieved here when the LLM added a reject button. Adding prices and information that the LLM could obtain through context, received the two lower agreements. However, for the latter, it is important to note that not all participants made it completely clear whether or not to add those things, making this a lower boundary. Removing elements so both items are on the same hierarchy achieved only an agreement of 28.57%. Not removing *False Hierarchy* and *Visual Prominence* (E, H, J, K), received an agreement of 0% at all times, when we count people that did not men-

tion this at all, this goes up to as far as 50%. Participants instead mentioned adding elements, e.g., a reject button, or making both less prominent, similar to what the LLM did in other cases. Another suggestion by participants, which neither got a majority vote nor was done by the LLM, was to change the order of items.

Bad Defaults were removed both times by the LLMs, which achieved an agreement of 85.71% (*Pelacase (F)*) and 57.14% (*Riverisland (G)*) with participants. All the users who did not explicitly say to remove it did not mention it at all. So there were neither explicit votes to keep it this way nor any other suggested changes. Similarly, Positive Framing, with an agreement to remove it of 85.71% (*Expedia (C)*) and 66.67% (*Ryanair (H)*), was also not mentioned by the remaining people.

Bad Defaults and *Positive Framing* were always removed by the LLMs, which received high agreement rates.

The LLM-as-a-Judge pipeline did not remove the instance of *Hidden Information* (*Riversisland (G)*). The participants in our study mainly did not mention it either (85.71%), and only one participant said to remove it (16.67%).

Not removing *Hidden Information* received a high agreement.

The pattern *Nagging* was removed by our pipeline by placing the banner at the bottom of the page, so it is not in front of the content. 33.33% suggested similar methods, thus agreeing with the LLMs. Justifications include that this helps *"to not affect the functionality of the website with the banner"* (P10). On the contrary, 66.67% wanted the whole banner removed, not wanting the content preserved somewhere else. Participants stated that it is *"manipulative and pushy"* (P3) and *"just emotional manipulation"* (P5).

The LLMs changed the placement of the *Nagging* banner. However, most participants want it fully removed.

When looking into *Disguised Ad*, there is an overall low agreement between our LLM pipeline and our participants. For *Amazon (B)*, the LLM did not remove the manipulation, which was only agreed with 14.29% of people. For *Booking (I)*, the LLM removed the info that it is an ad, which has an agreement of 37.5% with the participants. Instead, this is what 42.86% of the participants wanted for *Amazon*, but the pipeline did not do. Next to those two options, participants suggested making the ad content clearer, by either making the whole item more prominent (B: 14.29%, I: 25%), or just the information that it is an ad (B: 57.14%, I: 50%). A

Disguised Ad has a low agreement rate, as most participants want the information to be more noticeable.

separate suggestion involved relocating the ad content to a separate area (I: 25%). However, none of those suggestions were applied by the pipeline. Interesting are especially the different opinions of users. While P1 stated that *"highlighting "Ads" [...] is manipulative"*, P3 said that ads *"should be visually distinguishable from unpaid results"*.

Reference Pricing also has low agreement, as the LLMs did not always remove it.

In general, most participants wanted *Reference Pricing* removed (B: 71.43%, D: 83.33%, E/F: 57.14%), only for *Booking (I)*, the agreement among participants is below 50% (I: 37.5%). However, the LLMs only removed it for *Amazon (B)* and *Booking (I)*, highlighting a low agreement among the participants and the pipeline overall. Interestingly, for *Booking (I)*, one participant mentioned to keep it, but make it less prominent, as *"this is not necessary in red"* (P1). Another participant stated that *"this is fine with me, as long as the price actually is true"* (P5). Justifications to remove them stated that the old prices just made the current price look *"more discounted"* (P14) and *"better"* (P2), and that it is *"probably fake"* (P2).

The LLM pipeline did not change *Hidden Costs* and *Partitioned Pricing*, which, for the latter, is not agreed upon by the majority of participants.

Hidden Costs (*Opodo (E)*) and *Partitioned Pricing* (*Ryanair (H)*) were not changed by our LLM pipeline, with, respectively, one (14.29%) or no participant explicitly stating that they did not want it removed either. 42.86% and 16.67% of participants did not mention those patterns at all, which could also mean that they did not notice them. Instead, for *Partitioned Pricing*, the majority of participants suggested removing the pattern by displaying the full price instead (H: 83.33%). Fewer participants agreed on *Hidden Costs*, with one participant suggesting removing the information about it being a hidden price, and another suggesting making the information clearer.

Looking into the high-level category Social Engineering, the patterns *Low Stock*, *Activity Message*, and *Limited Time Message* were always removed by our LLM pipeline. However, the agreement with participants here only ranges from 14.29% (e.g., *Viagogo (K)*) to a maximum of 42.86% (e.g., Amazon (B)), thus it is always below 50%. Instead, participants often wanted to either keep the pattern, did not mention it, or suggested changing the design or wording to something more neutral, while keeping the information.

The last suggestion was especially common for *Low Stock*, with 42.86% (*Amazon (B)*) and 62.5% (*Booking (I)*) of participants suggesting either option. P3 stated that this change helps to *"parse the options at first glance without their influence"*. On the other hand, *Activity Message* and *Limited Time Message* were more often either not mentioned or wanted to be kept. Most justifications to keep all three deceptive patterns were due to it being *"interesting information to the user"* (P5) and *"to consider it for my purchase"* (P3). Especially to *"know if I should book now for discounts or if I can wait"* (P3) and to *"have a realistic assessment of whether you have to decide quickly"* (P8). All these justifications express that participants thought the information these deceptive patterns display are true, and that is why they want to keep them, even if they are aware of the manipulation (*"Weirdly, how many people last visited Billie Eilish events was interesting for me, did not want to remove it, even though it feels like scarcity"* (P5)).

Low Stock, *Activity Message*, and *Limited Time Message* were always removed by the LLM pipeline. However, participants often wanted to keep these information, resulting in lower agreements.

In contrast, participants more often opted to remove *High Demand*, with agreements from 57.14% (*Opodo (E)*) to 71.43% (*Viagogo (K)*) in cases in which the LLM pipeline also removed them. In one case, the LLM did not remove High Demand (*Opodo (E)*). Here, only one participant stated not to remove it (14.29%) due to it being *"not manipulative enough to convince anyone"* (P1).

Participants wanted *High Demand* removed, thus, whenever the LLM did that, the agreement was high.

Confirmshaming in *Expedia (C)* was removed by the LLM pipeline and received an agreement of 57.14%. The remaining people also wanted it removed, but not to shorten the wording as heavily. Differently, in *Theguardian (J)*, the agreement between the pipeline and the users was relatively low, from 16.67% to a maximum of only 33.33% for the different instances on the banner. This is mainly due to 66.67% of participants wanting the whole banner removed, on which the three *Confirmshaming* texts were present. However, the justifications to remove the banner are heavily based on the many *Confirmshaming* patterns present: *"banner is totally manipulative in its wording"* (P13), *""Rejection hurts" is just emotional manipulation"* (P5). *Personalization* (*Ryanair (H)*) received an agreement of 0%, as the LLM simply removed the personalized text. 66.67% did

For *Confirmshaming* neutral text received high agreements. For *Theguardian* participants wanted the banner removed partly due to *Confirmshaming*.

not mention it at all, while 33.33% would change the wording.

When the pipeline removed *Testimonials* the agreement was high.

Participants predominantly wanted *Testimonials* removed with 85.71% (*Expedia (C)*) and 100% (*Pelacase (F)*) of them stating that. Our LLM-as-a-Judge pipeline only removed it for *Expedia (C)*, thus showing a high agreement for it, while having an agreement of 0% for *Pelacase*. Most justifications stated either that it *"might be fake"* (P4) or that it is *"irrelevant"* (P1).

Further Alterations

Next to directly addressing deceptive patterns, participants had multiple other elements they wanted changed, mainly because they thought it was manipulative. Other changes aimed to mitigate deceptive patterns, but went beyond them, changing more on the page than only removing deceptive patterns.

Multiple participants suggested restructuring a whole web page.

Multiple times (n = 10), participants suggested restructuring the whole page to mitigate deceptive patterns. For example, to remove a table structure and instead show the items as multiple listings, or vice versa, to move listing items into the structure of a table.

Participants suggested multiple changes that did not relate to any deceptive patterns.

For different elements, participants wanted to make them more (n = 7) or less prominent (n = 1). For example, by making text or prices bigger in size. Further, participants often wanted text to be clearer (n = 6). This entails either the addition of further information, or restructuring or rephrasing of text. Additionally, participants included alterations and changes of varying elements (n = 30). For example, rephrasing text (n = 18), changing the names of items (n = 4), removing color (n = 3), or adjusting the sorting of items in a list (n = 4). We did not classify those adjusted elements as deceptive patterns. Interestingly, the LLM pipeline also changed texts three times. Which received an agreement of 0% twice, and when changing the title of *Riverisland (G)*, an agreement of 42.86%.

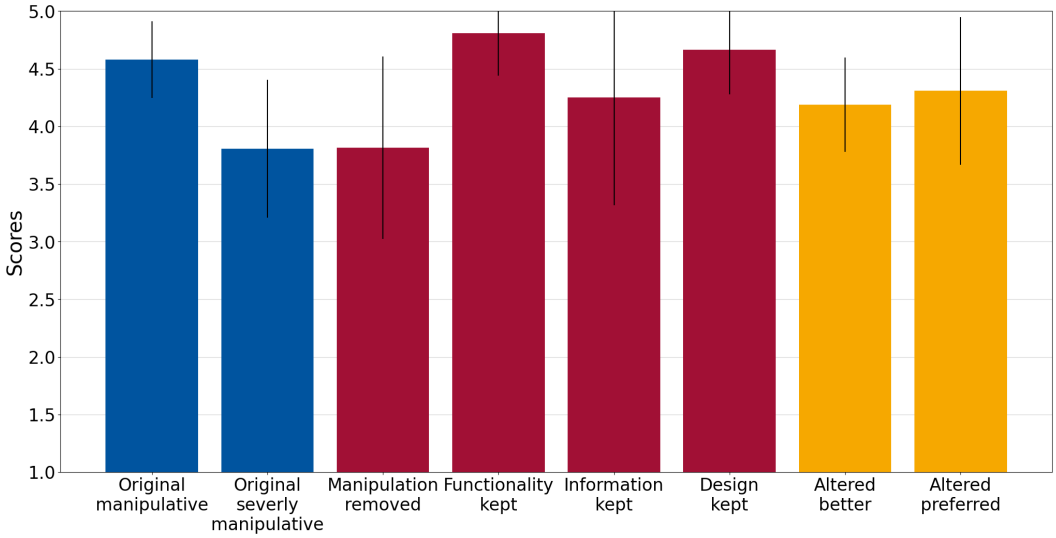

Figure 4.2: The figure shows the mean and standard deviation for the ratings for every question in task 2. The scale ranges from 1 to 5, and "5" means the user strongly agrees. We can generally see that users mostly preferred the altered web pages.

Some participants wanted whole elements or text removed.

Additionally, participants wanted to remove elements (n = 19). This includes the removal of text (n = 12), images (n = 3), or even larger web elements (n = 2). A few instances even wanted slightly larger whole web elements removed (n = 2), which they based on the existence of specific deceptive patterns, such as *Reference Pricing*, or *Hidden Costs*. In contrast, the LLM never removed an element that was not a deceptive pattern.

4.2.3 Part 2 - web page Ranking

The mean and standard deviation for each question can be seen in Figure 4.2. A table with the whole ratings for each web page can be seen in Table E.2.

All web pages were seen as manipulative (M = 4.58, SD = 0.33), and most as severely manipulative (M = 3.8, SD = 0.6). The web page seen as the least manipulative is *Opodo (E)* (M = 3.86, SD = 0.99), even though multiple deceptive patterns, such as *High Demand* and *Visual Prominence*, are

Participants generally thought the web pages were manipulative, and considered some as more severe than others.

present. Web pages with lower severity scores, such as *Aliexpress (A)* (M = 2.88, SD = 0.35), *Booking (I)* (M = 3.29, SD = 1.11), and *Riverisland (G)* (M = 3.75, SD = 1.28), included only a few deceptive patterns or almost only *Social Engineering* patterns. The most manipulative web pages, such as *Theguardian (J)* (M = 5.0, SD = 0.0), and *Pelacase (F)* (M = 4.75, SD = 0.46), contained patterns such as *Nagging*, *Confirmshaming*, and *Bad Defaults*, that users disapproved of.

Most web pages were seen as still partially manipulative.

We now take a look at the ratings for the individual evaluation criteria. DP REMOVED received the lowest overall score across all four criteria (M = 3.81, SD = 0.79). Only four web pages received a mean of 4 or higher. One of them, *Aliexpress (A)*, which is a simple notification banner containing *False Hierarchy*, achieved the perfect score of 5. On the other end of the spectrum, *Opodo (E)* obtained the lowest score of 2.5 (M = 2.5, SD = 1.17), which is also the only one below 3. Participants described patterns such as *Reference Pricing*, *Hidden Price*, and *High Demand* as still present. Interestingly, only one participant mentioned the *False Hierarchy* that is still there. The remaining web page scores ranged from 3.25 to 3.89. This makes it clear that participants perceived most web pages as still partially manipulative.

Participants sometimes, but not always, correctly identified the deceptive patterns still present. Some people also classified something as manipulative that is actually not.

Participants correctly identified that a few deceptive patterns are still missing, such as in the case of *Opodo (E)* we just explained, or the *False Hierarchy* still remaining in *Viagogo (K)*. This is not always the case, and participants repeatedly did not notice when deceptive patterns were still present. For example, noticeable in the high mean *Amazon (B)* achieved (M = 4.5, SD = 0.54), even though *Disguised Ad* is still present. On the contrary, sometimes participants said that the manipulation was not removed and justified this with something that is not actually manipulative. For example, *Expedia (C)* received a score of 4.88 because one participant thought the heading was still manipulative, or on *Amazon (B)*, in which people consider the ratings as manipulative. Further participants named small changes they still miss that relate to deceptive patterns. For example, in *Theguardian (D)* (M = 3.89, SD = 0.93), multiple people stated that the order is still manipulative as the *"no bundle*

choice should be on the left" (P15) instead of on the right. This can be classified as deceptive, but one could also argue that this is not necessary, as all are displayed equally. One participant said that *"the reject button should be highlighted more"* (P13) in *Riverisland (G)*, which is interesting, as this suggests adding a bright pattern instead of displaying both options as equal. For *Nagging*, one person criticized that the banner is now at the bottom, stating that it *"makes it feel also very deceiving. Maybe even more because it was so hidden"* (P14).

The criterion FUNCTIONALITY received the highest overall score (M = 4.81, SD = 0.37) across all four criteria. Eight web pages achieved a perfect overall score of 5, indicating that all participants agreed that the necessary functionality was retained. The web pages *Gotogate (D)* (M = 4.56, SD = 1.33) and *Theguardian (J)* (M = 4.44, SD = 1.01) obtained scores that were still above 4, implying that most participants believed that all functionality was kept. Interestingly, *Gotogate (D)* received one rating of "1" next to only ratings of "5", with the justification being that *"the option of no bundle is no longer selectable"* (P3). In this case, the LLM changed the button type to one that is also already prominent on the web pages, but for that type, the functionality was not extracted when we copied the HTML from the browser. The worst rating was given for *Expedia (C)* (M = 3.88, SD = 1.64). Participants justified their low ratings with the removal of a link leading to benefit details. However, this was only remarked by three out of the eight participants, with the remaining ones assigning the best score of 5.

Most web pages achieved a score of 5, only one was seen by multiple participants as compromised.

INFORMATION (M = 4.25, SD = 0.93) was rated worse than FUNCTIONALITY and DESIGN. Only three web pages achieved an overall perfect score of 5. Five web pages received a score between 4 and 5, and two web pages received a score between 3 and 4. Only one web page, *Expedia (C)* (M = 2.13, SD = 1.55), was given a score below 3, due to the removal of the listing of what an item entails. Interestingly, even though it achieved such a low rating overall, it was still given scores of 4 and 5 each once. Other low scores were assigned due to missing information, which are actually part of a deceptive pattern. For example, in *Booking (I)*, people said they want to keep the *Low Stock* information, in case they *"want to book 2 rooms"* (P14) and that it

Participants criticized when non-manipulative information was removed.

"is important for the user" (P7). Similarly, P5 commented, *"Weirdly, I like the stock information"*. Additionally, P12 said that the removal should depend on whether or not this is true information. These justifications are not only the case for *Low Stock*, but also for *Reference Pricing*, *"as this can help with comparison"* (P10). Similarly, P9 commented on the removal of *Limited Time Message*, *"that the price is cheapest today may be important"*. In contrast, in *Viagogo (K)*, even though *Low Stock* was also removed here, only one participant noted this negatively, and instead people said that *"the high demand/low stock messages have been removed well"* (P4). Another interesting thing is that P15 remarked that *"the call to action is missing"* due to both buttons' color now being gray, rating the information in *Aliexpress (A)* only a 3, based on this observation.

A few people gave low scores due to missing information that is part of a deceptive pattern, such as *Low Stock* information.

The second-best overall score was attained by the criterion DESIGN (M = 4.66, SD = 0.38). Five web pages were given an overall score of 5. The remaining six all achieved a score above 4, with the minimal score given to *Ryanair (H)* (M = 4.11, SD = 1.27). A few participants criticized that the heading is missing, which results in it looking *"ugly/unfinished"* (P14) and that *"the top text seems to [sic] close to the top border now"* (P3). Other comments were based on colors that were removed, which they say make the design look *"unfinished"* (P3) or that the *"color scheme flattens out without the pink"* (P9). Those cases were based on the LLMs removing *False Hierarchy* or *Visual Prominence* by making something the less prominent color, which resulted in the web pages now fully missing those colors.

Some participants criticized DESIGN based on changes the LLM performed to remove manipulation.

Overall, participants stated that the altered version feels better than the original version (M = 4.187, SD = 0.41). No web page received an overall score below 3, suggesting that our altered web pages were generally perceived as not worse than the original, and most even at least partially better. The lowest scores were acquired by *Expedia (C)* (M = 3.25, SD = 1.28) and by *Opodo (E)* (M = 3.88, SD = 0.83), who have respectively gotten the worst scores in FUNCTIONALITY and INFORMATION, as well as MANIPULATION. All other web pages attained scores above or equal to 4, with none receiving the perfect score of 5.

Web pages generally were seen as better; none was largely seen as worse.

Participants generally preferred the altered web pages compared to the original web pages (M = 4.31, SD = 0.64). The only web page that was not preferred was *Expedia (C)*, which received a score below 3 (M = 2.75, SD = 1.75), which again relates closely to the low score in FUNCTIONALITY and INFORMATION. The second lowest score was given to *Booking (H)* (M = 3.71, SD = 1.25), which also received a low score in INFORMATION and MANIPULATION. No other web page got an overall score below 4 here. This is interesting, as those two web pages are the ones that received the lowest scores in FUNCTIONALITY and INFORMATION. Every other web page received a score above 4 in both categories, as well as in the PREFERENCE. The only exception is *Theguardian (J)*, which got the second lowest score in INFORMATION. While its PREFERENCE score is above 4 (M = 4.11, SD = 1.05), this is the third lowest score overall in this category.

Most web pages were generally preferred by our participants. The web pages with the lowest scores are also the ones achieving the lowest scores in FUNCTIONALITY and INFORMATION.

Looking into participants' justifications, they often based their scores for PREFERENCE on missing information (n = 11). The most common responses here relate to the lost information in *Expedia (C)*, but also due to the *Low Stock* information that was removed. Participants also stated that the design is now worse (n = 6), specifically *"more boring"* (P11) and that it *"feels off"* (P3). However, this does not always relate to whether or not they would prefer the web pages, but instead often just negatively influences the rating of whether the web page feels worse or better. Often, for neutral scores, participants stated that there is no big difference (n = 12), that the web page is still manipulative (n = 10), or that the original was not that manipulative or simply the standard (n = 5). Positive justifications include that the manipulation was removed (n = 38), that the web page is visually better and less cluttered (n = 6), more neutral (n = 6), less stressful (n = 7), or easier and more pleasant (n = 7).

Justifications for lower scores often talked about missing information. Positives were that manipulation was removed, or that it is more pleasant.

4.2.4 Part 3 - Semi-structured Interview

A few participants explained the reasoning behind different changes to the same deceptive pattern on different web

pages, specifically for *Low Stock*. P3 said that it varies depending on the type of web page. While it can be removed from ticket website, it should stay on hotel booking website. The reason is that for the former, they are only on the website when they already know they will buy a ticket, which is not the case for hotel booking websites. Thus, for the latter, that information is still relevant. Further, P11 stated that it depends on the way it is worded. For *Booking*, *Low Stock* said "left at this price", which does not give information about the number of items that actually remain.

Participants stated how they felt differently about Low Stock depending on the wording and website.

Participants had varying ideas on how to improve such an application. The most common suggestions include an option to turn the changes off or see the original. Suggestions on how to do this varied from a switch at every element to an option to return the full website to its original state, as well as a log for the website. Another very popular suggestion was to make this countermeasure customizable, so not every deceptive pattern has to be removed. Other adjustments include a rating for the full website, a percentage of how sure the LLM is that it removed everything, and an option for users to give feedback and thus improve the pipeline. Lastly, one participant suggested including a database so the LLM does not need to regenerate the website each time, but can retrieve an already generated website from there, which should speed up the process.

Participants want such an application to be customizable and that they are able to see the original.

We asked participants what problems they see when using such an LLM for deceptive pattern removal. Answers include the energy consumption of the LLMs, but also the data protection, potentially giving the LLM too much access. Other problems surrounded the reliability of results, hallucinations by the LLM, and the performance and latency when loading websites. One user also worried about complications with lawsuits when information goes missing. Lastly, one participant stated that there are elements in websites that the LLM cannot access due to them being on the server-side, thus, they are not able to change everything.

Problems participants see in LLMs for such an application include energy-consumption, data protection, and reliability.

Participants expressed a very low tolerance for the mistakes that such an LLM-based application can make. The majority distinguished between severe mistakes, which com-

promise functionality or information, and less severe ones, which only compromise the design or the number of deceptive patterns removed. When looking at severe mistakes, most participants stated that they would not tolerate any mistakes at all. A few said low percentages around 0.2% or 5%, and four participants said higher tolerances from around 10% to 50%. For mistakes regarding missed manipulation, the tolerance is generally higher, with users being okay with it, or stating tolerances up to 50%. This relates closely to when we asked participants to rank the importance of different criteria in the evaluation process. Most named FUNCTIONALITY as the most important one, while some mentioned INFORMATION. However, for all but two participants, these two are among the top 2. The third place is mainly taken by MANIPULATION, and the all but two participants voted DESIGN on the last place.

Most people stated a very low to no tolerance for mistakes in regard to FUNCTIONALITY and INFORMATION, while not minding mistakes in DP REMOVED and DESIGN as much.

Chapter 5

Discussion

In the following, we will discuss our results and are gonna answer all our research questions. We will also draw connections between the different tasks in our user study, our technical evaluation, and previous literature. Following this, we draw implications for the application of this approach. At the end, we discuss the limitations of our work.

5.1 Influence of LLM-as-a-Judge (RQ1)

LLM-as-a-Judge improved the iterative deceptive pattern removal, specifically in the categories DP REMOVED, INFORMATION, and DESIGN (RQ1).

Overall, we can say that LLM-as-a-Judge positively influenced the deceptive pattern removal from websites, which answers RQ1. In particular, it had a positive effect on the amount of deceptive patterns removed, the design, as well as the information that is kept. Especially positive is that it did not hallucinate, add, or change information in our dataset. It also needed fewer iterations. Negatively, it still removed information. On the negative side, we had slightly lower ratings for FUNCTIONALITY, as well as DP ADDED. However, the differences are not large. Thus, we conclude that our approach succeeded at making the iterative deceptive pattern removal from the website better, while still leaving room for improvement. We will now discuss the results of the different criteria in more detail, as well as connect them to the general success and problems

of LLM-as-a-Judge and LLMs in general during the deceptive pattern removal.

5.1.1 Success and Pitfalls of LLMs and LLM-as-a-Judge While Removing Deceptive Patterns

o4-mini was relatively consistent, which also means consistent with the same mistakes.

Overall, o4-mini as the judge was relatively consistent. For example, it always identified fair web pages as fair, or suggested the same sentence to replace *Confirmshaming*. This also means that it made the same mistakes over and over again, such as the mistake in *Trick Question*. With different strategies, we were able to solve a few of them from time to time, but not all.

INFORMATION was less often compromised with LLM-as-a-Judge, but similar to FUNCTIONALITY, further improvement is needed.

In regard to INFORMATION, the LLM-as-a-Judge pipeline improved our results as well compared to the baseline. A very positive thing is that we did not notice any hallucination or changed information in the altered web pages for our final pipeline. This was still a problem for the version without LLM-as-a-Judge and was noticed by Schäfer et al. [2025] as well. Even though our final pipeline did not include any hallucination, it might be possible that this is something that could still occur occasionally, and is something that can never be fully ruled out. On the other side, our pipeline still removed information. This is something that is dangerous in an actual application and needs to be further improved before it can be used by actual users. Improvement is also needed for FUNCTIONALITY, as this is something that should not be compromised, as it is right now in our pipeline. Even if it only broke or removed functionality due to the attempt of removing deceptive patterns, this cannot happen. It happened even slightly more often than in the approach without LLM-as-a-Judge, showing that our judge did not improve here at all.

The generator added deceptive patterns, which is a phenomenon Krauß et al. [2025] already reported.

Additionally, our LLM-as-a-Judge approach added manipulation or made it worse. This happened specifically due to the generator, which the judge did not always catch. This is not a new phenomenon, as this already happened in Schäfer et al. [2025]'s approach. Further, Krauß et al. [2025] specifically researched and reported this on websites gen-

erated by ChatGPT. They found at least one pattern in all generated websites. This is also similar to what Chen et al. [2025] reported. Due to this tendency, further improvements and solutions are needed to counteract this.

Social Engineering and *Interface Interference* deceptive patterns were removed successfully.

Deceptive patterns that were particularly well identified and removed by the LLMs include Social Engineering patterns, such as *Low Stock*, *High Demand*, or *Limited Time Messages*, as well as *Confirmshaming*, but also *Bad Defaults*, *Positive Framing*, and most *False Hierarchy* and *Visual Prominence* instances. The LLM-as-a-Judge pipeline also improved by always identifying fair web pages as not manipulative. Notably, most of the patterns commonly removed are from the high-level categories *Interface Interference* and *Social Engineering*. This shows that this approach might generally just work better for specific categories and types of deceptive patterns, and specifically from those categories. This was also hypothesized by Schäfer et al. [2025], which is why they constructed their test set so it mainly contains patterns from these types. Similarly, Kocyigit et al. [2025] used a dataset containing mostly those deceptive patterns and reported high detection abilities by the LLM for them. This aligns closely with our most successful deceptive patterns.

LLM-as-a-Judge removed more deceptive patterns, but it still struggled in correctly removing, as well as detecting some specific types such as *Trick Question* and *Hidden Information*.

Our final LLM-as-a-Judge pipeline removed more deceptive patterns than the version without LLM-as-a-Judge. However, it still did not remove all deceptive patterns, and a few were particularly difficult for the LLM. *Trick Question*, *Adding Steps*, and *Forced Registration* were noticed by the judge, but were removed in a way that messed up the functionality or information on the web pages. Schäfer et al. [2025] similarly reported that in their test set, *Trick Question* was often flipped in its meaning, which is what happened for us as well. While they did not report this for all runs, in our case, this happened for almost every pipeline we tested. The patterns *Disguised Ad*, *Hidden Information*, and *Hidden Costs* were barely noticed by the LLMs to begin with. In contrast, Schäfer et al. [2025] also had an example that included *Hidden Information*, which they were able to mitigate successfully. That our pipeline did not succeed in this pattern could be due to us using real examples, as well as the specific way those patterns are implemented, which is likely to be more complex, as we used

real web pages. Kocyigit et al. [2025] reported lower correctness in the explanations for *Hidden Costs*, showing that their LLM also had problems with this deceptive pattern, but also an overall higher detection of this pattern in their dataset. Interestingly, all categories mentioned in this paragraph are from the high-level categories *Sneaking*, *Obstruction*, and *Forced Action*, indicating that those may be more challenging to remove and also detect in general. Similarly, Sazid et al. [2023] used GPT-3 to classify deceptive patterns, and while this worked for most categories at least partially, their approach could not identify the pattern *Sneaking*, and *Obstruction* also received a low accuracy. Comparing this to our results, many of the deceptive patterns we just discussed belong to the high-level category *Sneaking* or *Obstruction* in Gray et al. [2024]'s ontology, showing how difficult these deceptive patterns are to remove, but also even detect.

The high-level categories *Sneaking*, *Obstruction*, and *Forced Action* contained the most deceptive patterns not successfully removed or not even detected.

Besides that, the judge had problems detecting specific instances of patterns on some web pages, while correctly defusing them on others. This includes *False Hierarchy*, *Nagging*, *Testimonial*, and *Reference Pricing*. That the LLM did not remove them could have different reasons. It is possibly due to the way the website is implemented, which might obfuscate the LLM's view. It is also possible that this is server-side information that the LLM is not able to access via the HTML code we provided, or that the LLM is just not able to identify them correctly as manipulative in the way they are used in those cases.

The LLMs removed some deceptive pattern types only on some web pages, which could be due to information being on the server-side.

It is an important question whether or not all deceptive patterns can even be mitigated through removal, and especially through removal done by LLMs. Whether or not deceptive patterns can even be detectable was discussed by Curley et al. [2021]. While they named a few deceptive patterns as not detectable that we were able to mitigate, some they listed were also ones our LLM was neither able to detect nor remove, such as *Hidden Costs*. Further deceptive patterns that we did not test, such as *Roach Motel* or *Privacy Maze*, are some that often cannot be pinned to a specific location on the website [Gray et al., 2024]. Additionally, Gray et al. [2025] mentioned temporal deceptive patterns, which span over multiple web pages. This sparks the

There are potentially some deceptive patterns that cannot be removed by a current LLM, due to their nature or the current LLM's token limit.

question of whether or not LLMs are capable of detecting and removing them. Some of those elements would come with large HTML files, as they span across multiple web pages. These, for one, are possibly very complex for the LLM to understand, and, for two, are difficult to input into the LLMs, as LLMs currently still have limited tokens. The option to split HTML into multiple parts is difficult, as patterns that depend on other elements could be in different parts. Lastly, as mentioned above, our LLM-as-a-Judge approach was not able to remove the specific deceptive patterns discussed prior. Thus, we might need other countermeasures for different deceptive pattern types. This could be highlighting and providing an explanation, as suggested by Schäfer et al. [2024], which many of their participants approved of.

5.2 User Alignment and Perception (RQ2, RQ3)

To answer RQ2, we look at the agreement between users and the LLM-as-a-Judge for each particular web page and notice a relatively low agreement. Most often, the LLM changed something in a different way than users would, or that it should not have removed at all, which is noticeable in the low recall scores. However, a few times, the LLM also failed to remove something the users wanted removed. Generally, we can say that our changes do not align with the judgment of users. To get a better overview of why there is such a low agreement, we look into the individual deceptive patterns in Section 5.2.1.

User opinions generally do not align with our LLM-as-a-Judge approach (RQ2).

The low alignment relates closely to studies of LLM-as-a-Judge in other sophisticated fields. For example, Szymanski et al. [2025] found that LLM-as-a-Judge does not have a high agreement in the field of dietitian and mental health. Similarly, Wang et al. [2025] showed varying performances across multiple software engineering tasks, with some achieving high and some low human alignment. Deceptive pattern removal might be similar.

Low alignment of LLM-as-a-Judge with humans has been found in other sophisticated fields too.

Different from task one, we noticed in the ranking that users generally perceived our altered web pages very positively (RQ3). Most web pages felt better to the participants, and none felt generally worse. Additionally, all but one web page were, on average, preferred by users over the original. This shows that even though users generally did not agree with the way our websites were changed, the adjustments performed were better than no changes. Thus, we generally might not have the optimal solution, but one that is not worse than not doing anything. Generally, when users did not prefer the altered version, it was mostly due to missing information. As users named FUNCTIONALITY and INFORMATION as the most important criteria, it is not surprising that mistakes in either category yield negative perceptions and low scores from them. Differently, participants perceived web pages as better than the original, even when not all the manipulation was removed. This also relates to participants' assertion that the criterion DP REMOVED is not as important as other criteria, and how they tolerate more mistakes here. Showing also that it might be better to remove even a little bit instead of nothing. In the following, we will go into detail about how users perceived the web pages from the perspective of each evaluation criterion, as well as compare them to our evaluation in Chapter 3.

In general, users preferred our altered web pages over the original. Low preference scores were mostly due to missing information (RQ3).

Looking into the ratings for the specific evaluation criteria, it is interesting that all but one web page received a relatively low rating for DP REMOVED. Even when we classified all manipulation as gone, participants did not always agree with this. As we did not tell participants what elements are manipulative, and they had to look for this themselves, it is unsurprising that participants could not always correctly identify all deceptive patterns, or identified something as a deceptive pattern that we did not classify as one. Di Geronimo et al. [2020] and Bongard-Blanchy et al. [2021] showed that users cannot generally actually detect all deceptive patterns. The detection rate also varies across deceptive pattern types [Bongard-Blanchy et al., 2021; Bhoot et al., 2020]. Based on this, we can assume that not all of our users were able to detect all patterns, thus partly explaining the varying scores here. To further explain the divergence in scores, participants sometimes remarked that bright pat-

Participants are likely not to have detected all deceptive patterns, and also classified something as manipulative that is not.

A few participants wanted to include bright patterns instead.

terns should be added, specifically the counterparts of *Bad Defaults* and *False Hierarchy*, and thus handed out a non-perfect MANIPULATION score.

The scores for INFORMATION diverge slightly from our ratings. This is often due to users handing out low scores based on deceptive patterns that were removed, which we will discuss in more detail in Section 5.2.1. Going back to the adjustment in our pipeline with multiple judges, we noted there that the definition between the criteria overlaps. This is something that also appears here in some way. While some participants deem these deceptive patterns to be manipulative, others might think it is valuable information instead of being manipulative. And some even did not mind it being manipulative, because the information was more relevant. Relating widely to how it is hard to define what information should be kept and what is just manipulative and can be removed. This is something people have various opinions on. Lastly, individual participants stated other reasons for low INFORMATION scores. For example, that a call-to-action is missing due to the color of a button being removed, as this was part of a *False Hierarchy*. This is interesting, as this is a very wide interpretation of the term information and colors, and relates closely to the discussion of the importance of the design.

Some participants wanted to keep information present in deceptive patterns, displaying an overlap between DP REMOVED and INFORMATION.

How design is perceived varied between people. A few participants remarked that a design felt off due to deceptive patterns missing, which was mainly due to the colors those patterns had and additional whitespace. We wonder if participants would still notice this if they did not have the direct comparison of the original web page next to it. In our evaluation of these designs in Chapter 3, we did not notice that particularly negatively. Especially because these were mostly things that developed due to the removal of manipulation, thus, this does not fall in our definition of DESIGN. So, while some DESIGN scores from participants were below "5", our ratings did not have deductions. Similarly, some participants never remarked on the design and always gave a score of "5", and others told us that they even use browser extensions that might break designs, but that they do not mind. This relates closely to how participants mostly rate DESIGN as the least important score.

Most participants did not mind some slight design changes, while a few did not like the removal of colors or additional whitespace.

Even though the design is not seen as the most important element, it is an interesting thought to try to adapt the design that goes missing when removing deceptive patterns, such as adjusting the size of listings to reduce whitespace or including now removed colors in other elements. This is a difficult task to automate, especially with the LLM only having the HTML to present the website, and thus might require a Human-in-the-Loop.

We did not include altered web pages with messed up functionality, which would have probably returned lower FUNCTIONALITY scores.

FUNCTIONALITY scores were mainly good. However, it is important to note here that we did not include the web pages on which our LLMs severely messed up the functionality. Thus, it has to be assumed that those web pages would receive generally low scores in FUNCTIONALITY, pulling down the average of the ratings for this criterion. As FUNCTIONALITY has been named the most important by our users, this probably would yield lower preference scores as well as a low agreement. We are aware of this, but did not want to include these web pages in our dataset, as we did not expect more interesting insights from these compared to the web pages we did include.

We noticed large differences in users' opinions and ratings, showing different perceptions.

Lastly, one thing that we noticed was that web pages often received a wide range of scores from the different participants, specifically in regard to Manipulation and Information. I.e., a web page might receive the rating "1" in Manipulation, but also "5" from another participant. This again shows the difference in how people perceive these categories, what they identify as manipulative, and what is important information.

5.2.1 Comparison of Individual Deceptive Patterns Types

Interestingly, the majority of users wanted to keep *Low Stock* information, as well as *Limited Time Messages*, and *Activity Messages*, when they adjusted the web page themselves. Most of the justifications were due to participants wanting to preserve the content and assuming the information is or might be true, not wanting to risk losing them even if they could be fake. Conversely, for all three pat-

terns, a few participants would remove these patterns. This aligns with Schäfer et al. [2024], as some of their participants stated similar wishes to keep *Limited Time Message* and *Countdown Timer*, while others wanted it removed. However, in their study, participants did not bring up similar arguments for *Activity Message*. For us, fewer participants wanted to preserve it compared to *Low Stock*. However, the majority did want to keep it, specifically for *Amazon*. Participants who wanted to keep these patterns often suggested making them less prominent instead. Showing that they do not wanna be pressured further, but instead want the information as neutrally as possible.

Many people did not remove *Low Stock*, *Activity Messages*, *Limited Time Messages* in their alterations to keep this information, while some wanted them removed.

On the other hand, participants predominantly wanted the other *Social Engineering* Patterns, *High Demand* and *Testimonials* removed, and not to preserve the information. This is for the former something the LLM mainly did, but for the latter, only in one of the two instances on our web pages. For *High Demand*, this aligns with Schäfer et al. [2024]; for *Testimonials*, this diverges from their findings, as users predominantly wanted to keep it in their study.

High Demand and *Testimonials* were removed by most participants and our LLM.

In contrast to the first task, in the second one it was mentioned way less that these deceptive patterns should be kept. Web pages in which those patterns were removed generally still received high ratings, even when such information was removed. Interesting are, among others, the web pages *Amazon* and *Viagogo*, which both received very low agreement scores, specifically due to patterns such as *Low Stock* and *Activity Messages* being removed by the LLM. However, in the rating task, both achieved average scores larger than "4" for the preference, and only once did people mention in their justifications that *Low Stock* information went missing. For the rating regarding INFORMATION, this was mentioned a bit more often. To be exact, nine times across all web pages did participants justify a lower score in the category INFORMATION due to *Low Stock*, *Reference Pricing*, or *Limited Time Messages* being removed, and wanted to keep the information that is conveyed by those patterns. On one side, this was way less than what it was mentioned to stay in the first task. On the other side, even if they stated that, they still preferred the altered version, or selected neutral "3" for their preference. Not once did a participant se-

Contrary to task one, in task two participants did not mind the removal of *Low Stock*, etc. as much, mentioning it less, as well as that the removal did not affect their preferences much.

lect that they preferred the original because of such a removal. Another interesting differentiation is that participants mainly named *Low Stock* in the justifications, and did not criticize once that *Activity Message* was removed. While this aligns with Schäfer et al. [2024], as participants did not want to keep the information there either, it varies from the majority of users in task one, who explicitly stated to keep it.

The differences regarding *Low Stock*, etc. are possibly due to users not minding them being removed, that they preferred them being gone over their original prominent layout, or that they only later on realized they were manipulative.

While we do not fully know how to interpret these results, it is a very interesting finding. One reason for the preference, even when participants mentioned the loss of those patterns negatively before, could be that the removal of other deceptive patterns outweighs the loss of this information. Another option is that participants did not like the way it was displayed on the web page originally, and thus, would prefer it gone over it staying displayed like that. This is based on participants commonly suggesting in the first task to change the way *Low Stock* and other patterns are displayed, and wanting them to be more neutral. However, next to the cases in which participants mentioned this, it is unclear why they mentioned those deceptive patterns way less compared to how the majority of users in task one advocated for them to stay. Maybe they did not mind them being gone once they saw how the new web pages looked. Another thing could be that they did not realize that it was actually manipulative in task one, but realized that it was in task two. However, many participants stated that they knew it was or that it could be manipulative, but they thought the information was interesting anyway. Thus, it cannot apply to all people.

Preferences regarding the removal of *Social Engineering* patterns likely vary between people, websites, and deceptive patterns.

Overall, we hypothesize that whether or not such *Social Engineering* patterns, and patterns that carry information generally, should be removed depends highly on the person, the website, and how this pattern is exactly implemented. This is also supported by the user interviews, in which they told us that it diverges based on the type of website, such as a hotel booking versus a concert ticket website, or the way it is phrased.

Next, we look into deceptive patterns that the LLM removed, and the users also wanted removed, which yielded

a high agreement. Next to *High Demand*, these include *Bad Defaults*, *Confirmshaming* and *Positive Framing*. Additionally, *Testimonials* and *Reference Pricing* were partially removed by our LLMs, i.e., not on all web pages. Additionally, *False Hierarchy* and *Visual Prominence* received partially high and partially low agreements. Participants generally wanted the less prominent option, if that was feasible and useful. Otherwise, they were also open to adding information to make the options equal. They did not agree when the LLM removed elements to make them equal or did not remove them at all. Schäfer et al. [2024] similarly found that users generally wanted *Confirmshaming* and *Visual Interference* removed, which relates closely to *False Hierarchy* and *Visual Prominence*. A similarity in this group is that they barely convey information, except *High Demand* and *Testimonial*.

Confirmshaming, *False Hierarchy*, and *Visual Prominence* were mostly successfully removed. Participants did not agree with the removal of information to mitigate them.

Besides the deceptive patterns that the LLM successfully removed, it also did not remove some that participants wanted removed. This includes *Partitioned Pricing* and *Disguised Ad*. While the LLM either did not change *Disguised Ad* at all or made it even more disguised, users wanted the information that it is an ad to be more prominent. The LLM might not be able to interpret the way those patterns are utilized as fully deceptive. Besides those two, users also wanted the full pop-up in *Nagging* removed, which the LLMs did not do. Instead, the LLMs changed the placement of the banner so it does not block the content behind. While this also removed the deceptive pattern, it does not fully align with the user's preference. It is important here that users deemed the content of the banner to be highly manipulative as well, which is noticeable in the way this web page received the highest score in the severity of the manipulation, mainly due to multiple *Confirmshaming* patterns on the banner. This might have influenced their decision to remove the banner, leaving the question of how users would react to other *Nagging* instances.

Some deceptive patterns were not removed by the LLM, even though the users wanted them gone.

Hidden Costs and *Hidden Information* were generally not removed by our LLM, which yielded higher agreements with our participants. However, the high agreement is highly based on many participants not mentioning these patterns in their responses, which suggests that they did not notice

Hidden Costs and *Hidden Information* might not have been detected by the user, and thus not removed in their alterations.

these patterns as well. The possibility of those patterns being hard to detect has already been shown in the literature. Bhoot et al. [2020] noted that around 30.7% of the users did not detected *Hidden Costs*, and Bongard-Blanchy et al. [2021] identified 42% and 64% of their users not detecting *Hidden Information*.

LLMs might have similar problems detecting specific deceptive patterns than humans.

The deceptive patterns not removed by our LLMs include *Hidden Costs*, *Hidden Information*, *Disguised Ad*, and *Partitioned Pricing*. As already discussed, Bhoot et al. [2020] and Bongard-Blanchy et al. [2021] identified rather low detection rates for the two former. Bhoot et al. [2020] also reported one for *Disguised Ad*, with only 55.3% of participants noticing this pattern. From the deceptive patterns we included in the user study, these are some of the lowest detection rates reported by them. While Social Engineering Patterns, such as *Confirmshaming*, *High Demand*, and *Limited Time Messages*, achieved detection rates over 80%. We speculate that some similar patterns that are harder to detect for users might be harder to detect for an LLM as well. It could also hint at the difficulty of removing those patterns. Schäfer et al. [2025] defined *Hidden Costs* as an element that could not be removed and counteracted them in other ways, for example, by highlighting them. Something similar could apply to *Hidden Information* or *Partitioned Pricing*.

Users have a low tolerance towards removed information, resulting in low preference scores.

When comparing the agreement scores of the whole websites with the ratings, it is noticeable that a few of them achieved very high agreement rates, but participants did not prefer the altered web pages over the original. One example is *Expedia*, with an agreement of 85.71%, but the lowest preference score of 2.75. When looking into the justifications, this is due to the LLM removing non-manipulative information, which is the one change that the LLM performed that users did not agree with. Showing the high risks of one mistake ruining the whole web page for users, overshadowing everything else. This again relates closely to the severity users hold to information being removed, as most low scores in preference are attributable to information being removed. In contrast, even when little manipulation was removed, users still often felt that it was bet-

ter than nothing, showing the tolerance here, which is also what they told us in the interview.

Next to the differences between the two tasks regarding the Social Engineering deceptive patterns we talked about, there are further differences regarding the preference and agreement in web pages that barely removed any deceptive patterns, such as *Opodo*, *Ryanair*, and *Pelacase*. While the agreement was relatively low, the preference is considerably high. This also shows that users prefer web pages even if the application does not remove all deceptive patterns. Thus, only partial removals are better than none. This specifically applies to web pages that users ranked as severely manipulative, such as *Ryanair*. It yields even higher agreements for the altered web pages here, even if only half of the deceptive patterns were removed. Further, when deceptive patterns are severely manipulative, participants might be more willing to overlook mistakes. Not ones relating to FUNCTIONALITY or INFORMATION, but, for example, DESIGN mistakes or the removal of deceptive patterns that they do not fully agree with generally. For example, in *Ryanair* the LLM removed personalization by deleting the whole title, messing up a noticeable part of the design, which received an agreement of 0% in task one, but as the LLM removed this and other patterns, the preference score in task two was unified a "5".

Removing few deceptive patterns is generally better than none, specifically when users' think the general manipulation on the web page is rather severe.

Overall, we noticed varying options and suggestions on how to remove deceptive patterns, also for the same type. Removing deceptive patterns is very closely related to the notion of *fair patterns*, in which the design is constructed to not manipulate users [Potel-Saville and Francois, 2023], which is also what we try to achieve by removing the deceptive patterns. This was critically discussed by de Jonge et al. [2025]. They argued that fairness is something that highly depends on the context, but also on the person. This is similar to what we noticed in our user study. We had multiple deceptive patterns that were present on different web pages. The percentage of participants who wanted the same deceptive pattern removed across multiple web pages varied. This could be partially due to varying participants who had to alter each web page, but possibly also based on how participants viewed the deceptive pattern in different

How people perceive something to be fair varies, which is also what we noticed in the differing opinions in our user study.

contexts. This is a topic that was often talked about in the interviews, when participants told us that whether or not *Low Stock* should be removed depends on the wording, as well as the website context. Similarly, we were presented with various options on how to alter web pages to remove one deceptive pattern. For example, for *Nagging*, some participants only perceived it as not manipulative when the whole banner was removed, while others just wanted the banner placed differently. This showed us the widespread options on how to define *"fair"* and make something actually not manipulative.

5.3 Application of this Approach

Considering users' low tolerance for functionality and information mistakes, our approach is not yet good enough.

While LLM-as-a-Judge did improve the results of removing deceptive patterns from websites, it is still far from being fully successful and ready to be used in an actual application. The user study confirmed this, with the low error rate that users tolerate, which is mostly close to 0% regarding functionality and information. This was not achieved in our results, as six web pages were compromised in at least one of those criteria. Thus, we can say that this approach is not yet sufficient and needs further improvements.

The capabilities of current LLMs further hinder such an approach from being actually applicable.

Next to the not-yet fully satisfying results, problems within the capabilities of current LLMs arise that hinder such an actual application. For one, current LLMs have a token limit, which varies between models. These limits make it difficult to input a single web page or even a full website into LLMs. Thus, it would currently be necessary to break the code into multiple segments, which poses problems in itself that we discuss later. Moreover, the more tokens, the higher the cost, which is also necessary to take into account when this approach is deployed in an actual application. Additionally, the LLMs still take a rather long time, often multiple minutes, to process and output the changes. As our LLM-as-a-Judge approach used an additional LLM, we need even more time than the baseline approach. Participants already stated their concern regarding this in the user study. All in all, these limitations currently make such an application not feasible. However, most of these are proba-

bly going to be overcome in the future, when newer models will be released.

When looking at the different opinions of users on how deceptive patterns should be adjusted, it is clear that they vary heavily for different types. Thus, we suggest that a browser extension that incorporates such a removal of deceptive patterns should have an option to customize what deceptive patterns should actually be removed, which participants in our study also commonly suggested and wanted. Connecting this with the idea of other countermeasure options, a customization could also allow for the selection of specific countermeasure types for different deceptive patterns. This again relates to Schäfer et al. [2024]'s findings, in which they showed how participants preferred different countermeasures for different deceptive patterns. Additionally, LLMs make mistakes, which is something that probably will not be eliminated fully. Thus, similarly to suggestions by the participants, we recommend having an option to fully return to the original. Alternatively, the visual countermeasure *"Switch"* suggested by Schäfer et al. [2024] is also a possible option to adjust the approach.

Customization of such an approach is necessary due to various user opinions.

Returning to the original state of a website should be feasible.

Additionally, participants mentioned concerns regarding their data when web pages they have already interacted with are given to an LLM. This is something to be taken seriously and opens the question of where we stop giving an LLM web pages, what do we ask the LLM to change or adjust, and how do we protect private information in such an automated approach? An idea would be to run an LLM locally, which is currently something that is not available for many models, including ones from OpenAI.

Participants have concerns regarding their data privacy.

5.4 Limitations

When deciding what adjustments to make, we had to focus on the most promising ones. However, there are many possibilities to further improve our LLM-as-a-Judge pipeline. For example, to fine-tune an LLM [Gu et al., 2024], include the chat history, adjust the prompt further, or change the output of the LLM judge to use scores or a pairwise com-

We could not test every adjustment combination and every adjustment possible.

parison and thus completely change the way the judge is utilized [Zheng et al., 2023]. Additionally, we were not able to test every single adjustment combination. A general-purpose model may work better when we apply any of the adjustments we tested afterward. Especially, because those prompting strategies work better for general-purpose models [Nori et al., 2024]. Thus, there might be a better pipeline with better prompts that outperforms our approach.

We only tested each pipeline once for the dataset.

We also only tested each pipeline once for each web page. As LLMs are not deterministic, this leaves every possibility of the LLM performing better or worse for some pipelines, which might have influenced tendencies and results.

Our dataset only contained 27 web pages, and a selection of deceptive patterns.

We compared the baseline to LLM-as-a-Judge on the same dataset that we used to iteratively refine the latter.

Our dataset was also not that large and only contained two larger real web pages. Additionally, we did not cover all 65 types of deceptive patterns from Gray et al.'s ontology, leaving room to see how the LLM would remove those, but also how users would even want them removed. Specifically, we had the majority of our deceptive patterns from two out of five high-level categories from their ontology, leaving room to explore the other three. Lastly, we compared the baseline and the LLM-as-a-Judge approach on the same dataset we used to iteratively optimize LLM-as-a-Judge. Even though our dataset is a very diverse set, we cannot eliminate the possibility of our pipeline overfitting on our dataset.

Our participant pool was rather small and homogeneous.

The user study was only conducted with 15 participants, and as a result, we only had six to eight data points for each web page per task. We found tendencies among the users. However, more participants would be useful to confirm them. Additionally, our participant pool was not that diverse, with most participants being university students with a technical background, and many of them already knowing deceptive patterns. General end-users are way more diverse.

The deceptive patterns mostly appeared in combination with other ones.

The deceptive pattern instances mainly appeared in combination with other ones in our dataset. Thus, we do not know the exact preferences of each user for each deceptive pattern, but rather tendencies based on the ratings for the whole web page. The adjustment task was slightly more

precise for each deceptive pattern, as we could extract the changes per pattern well enough. However, for both tasks, this limits the generalizability of the results.

We did not tell our participants about the exact deceptive patterns that are present on each web page, but just told them that manipulation is present on each of them. Instead, they had to search for them themselves. This led participants to see something as manipulative that is not a deceptive pattern, and also to overlook some patterns that are actually present. This resulted in us not getting the opinion of users on every deceptive pattern instance, but instead the observation that they just did not notice them.

We did not tell participants what the deceptive patterns on each web page were.

Chapter 6

Summary and Future Work

6.1 Summary and Contributions

In this thesis, we explored LLM-as-a-Judge to expand the iterative deceptive pattern removal from websites, which was initially proposed utilizing only one LLM by Schäfer et al. [2025]. We added an additional LLM to evaluate the generator LLM that changes the websites. After defining an initial pipeline, we tried to iteratively optimize this through different adjustments, such as varying the selected models, trying different prompting strategies, and adjusting the communication between both LLMs. We ended with a final pipeline, and knowledge about which adjustments worked better and which did not work.

We looked into adding LLM-as-a-Judge to the iterative deceptive pattern removal from websites and iteratively refined the pipeline by testing multiple adjustments.

We then compared the results of our final LLM-as-a-Judge pipeline with the approach without LLM-as-a-Judge, for which we picked Schäfer et al. [2025] prompts in a slightly adjusted manner. We noticed that our pipeline showed promising results, especially as it removed more deceptive patterns, while compromising the design and information on the website less often. However, it did not decrease functionality mistakes and even added more deceptive patterns than the version without LLM-as-a-Judge. Lastly, our

LLM-as-a-Judge showed an improvement in the amount of deceptive patterns removed, as well as in keeping the information and design better.

judge is also a successful way to stop the iterative process, which resulted in us needing fewer iterations.

Users generally would change the web pages differently, but still preferred the altered version over the original.

Afterward, we conducted a user study to compare the adjustments from our LLM-as-a-Judge pipeline to human judgment, as well as look into the perception of users on our altered websites. We did notice a relatively low agreement between the changes the LLM made and the changes the majority of users wanted done. While this is partly due to the LLM not removing some deceptive patterns, it is also rooted in users wanting to keep some that the LLMs did remove. In contrast to the low agreement, users generally perceived our altered web pages very positively, and would largely prefer them over the original version. Web pages that were not as widely preferred removed non-manipulative information from the original version.

LLM-as-a-Judge still needs improvement.

Overall, even though we saw improved results with LLM-as-a-Judge, and a general preference towards these results compared to the original web pages, our results are not yet sufficient. We saw mistakes in information and functionality, which is something users have a very low tolerance towards, and should not happen in an application as often as it did in our results.

6.2 Future Work

Our approach still needs further improvement; different adjustment combinations are an option, as well as further adjustments, such as fine-tuning.

One important element in the future is to work on the reliability of our approach and increase the results further. Currently, due to mistakes, especially in the functionality and information criterion, our approach is not good enough for an actual application. Further improvements could test different combinations of the adjustments we tested, as we do not know whether or not a general-purpose model with prompting strategies and more guardrails could potentially excel the reasoning model we currently use. Additionally, further improvements beyond the ones we already tested are possible. For example, to utilize fine-tuning and prompt engineering, or adjust the hyperparameters. Furthermore,

in August 2025, the new model GPT-5 was released[1], which might yield better results. Consequently, it is to be expected that further newer models will be released with better capabilities, which can improve our results further, and possibly allow a larger token input.

A browser extension is an element for future work. Especially one that is customizable and allows to return to the original website.

A browser extension that works on actual websites would be another direction for future work. Next to the general implementation, looking into different features for the extension is an interesting idea. One direction would be to investigate the customizability of this. Closely connected, it should be explored whether or not an LLM is capable of only removing specific deceptive patterns while leaving the remaining ones untouched. Additionally, the possibility of switching between the original and the altered version is something to implement and explore further, to determine how exactly this should or could be done.

Utilization of LLMs for different countermeasures, such as highlighting and providing explanations.

It would also be interesting to see whether LLMs would be able to accomplish different countermeasures, such as the highlighting and providing explanations Schäfer et al. [2024]. This is specifically interesting to see, as some participants did not want the LLM to remove specific deceptive patterns. This could be connected with the option to customize, and see whether the LLM would be able to combine the removal with different approaches, varying the countermeasure used for the different deceptive patterns.

Our user study could be repeated with more participants and a more diverse group.

As our participant group was rather homogeneous and small, it would be interesting to repeat our user study with a different user group, varying in age, deceptive pattern knowledge, and technical background. Further, different deceptive pattern types, as well as other websites and instances of the types we already used, would be an interesting addition. Further studies could be conducted, in which users use an actual browser extension that implements this, to see how users would interact with it, as well as whether they would use it in their everyday lives. This is also something that has the potential to be studied over a longer period of time.

[1] `https://openai.com/index/introducing-gpt-5/` [Accessed: Sep. 29, 2025]

Appendix A

Deceptive Pattern Types

A.1 Deceptive Pattern Types and Definitions

In this section, we present all the deceptive patterns relevant to our work and provide definitions for each type. The deceptive patterns are taken from Gray et al. [2024], as well as the definitions, which have been taken verbatim from Gray et al..

Deceptive Pattern	Pattern Definition
Obstruction	"impedes a user's task flow, making an interaction more difficult than it inherently needs to be, dissuading a user from taking an action"
Adding Steps	"subverts the user's expectation that a task will take as few steps as technologically needed, instead creating additional points of unnecessary but required user interaction to perform a task"
Sneaking	"hides, disguises, or delays the disclosure of important information that, if made available to users, would cause a user to unintentionally take an action they would likely object to"
Disguised Ad	"style interface elements so they are not clearly marked as an advertisement or other biased source'"
Drip Pricing, Hidden Costs, Partitioned Pricing	"reveal new charges or costs, present only partial price components, or otherwise delay revealing the full price of a product or service through late or incomplete disclosure"
Reference Pricing	"include a misleading or inaccurate price for a product or service that makes a discounted price appear more attractive"
Interface Interference	"privileges specific actions over others through manipulation of the user interface"
False Hierarchy	"give one or more options visual or interactive prominence over others, particularly where items should be in parallel rather than hierarchical"
Visual Prominence	"place an element relevant to user goals in visual competition with a more distracting and prominent element"
Bad Defaults	"subverts the user's expectation that default settings will be in their best interest, instead requiring users to take active steps to change settings"
Positive Framing	"visually obscure, distract, or persuade a user from important information they need to achieve their goal"
Trick Question	"subvert the user's expectation that prompts will be written in a straightforward and intelligible manner, instead using confusing wording, double negatives, or otherwise leading language or interface cues"
Hidden Information	"subverts the user's expectation that relevant information will be made accessible and visible, instead disguising relevant information or framing it as irrelevant"

Table A.1: Definitions of the deceptive patterns included in our dataset and discussed in this thesis. The deceptive patterns are taken directly from the ontology by Gray et al. [2024], and have been taken over verbatim.

Deceptive Pattern	**Pattern Definition**
Forced Action	"requires users to knowingly or unknowingly perform an additional and/or tangential action or information to access (or continue to access) specific functionality"
Nagging	"subverts the user's expectation that they have rational control over the interaction they make with a system, instead distracting the user from a desired task the user is focusing on to induce an action or make a decision the user does not want to make by repeatedly interrupting the user during normal interaction"
Forced Registration	"subverts the user's expectation that they can complete an action without registering or creating an account, instead tricking them into thinking that registration is required"
Social Engineering	"presents options or information that causes a user to be more likely to perform a specific action based on their individual and/or social cognitive biases, thereby leveraging a user's desire to follow expected or imposed social norms"
High Demand	"indicate that a product is in high-demand or likely to sell out soon, even though that claim is misleading or false"
Low Stock	"indicate that a product is limited in quantity, even though that claim is misleading or false"
Testimonials	"indicate that a product or service has been endorsed by another consumer, even though the source of that endorsement or testimonial is biased, misleading, incomplete, or false"
Activity Message	"describe other user activity on the site or service, even though the data presented about other users' purchases, views, visits, or contributions are misleading or false"
Countdown Timer	"indicate that a deal or discount will expire by displaying a countdown clock or timer, even though the clock or timer is completely fake, disappears, or resets automatically"
Limited Time Message	"indicate that a deal or discount will expire soon or be available only for a limited time, but without specifying a specifc deadline"
Confirmshaming	"frame a choice to opt-in or opt-out of a decision through emotional language or imagery that relies upon shame or guilt."
Personalization	"subverts the user's expectation that products or service features are offered to all users in similar ways, instead using personal data to shape elements of the user experience that manipulate the user's goals while hiding other alternatives"

Table A.2: Definitions of the deceptive patterns included in our dataset and discussed in this thesis. The deceptive patterns are taken directly from the ontology by Gray et al. [2024], and have been taken over verbatim.

A.2 Deceptive Patterns in each Web Page in our Dataset

All web pages included in our dataset for the evaluation, as well as the amount and types of deceptive patterns present in them.

Website	#DPs	DP included
K_fair	0	/
K_DP	8	Adding Steps, 2x Confirmshaming, Countdown Timer, False Hierarchy, Limited Time Message, Nagging, Visual Prominence
S_fair	0	/
S_wholePageFair	0	/
S_confirmshaming	1	Confirmshaming
S_falseHierarchy	1	False Hierarchy
S_trickQuestion	1	Trick Question
S_wholePageDP	1	Countdown Timer, False Hierarchy, Low Stock
AmazonFair	0	/
Audi	0	/
Ieee	0	/
Zalando	0	/
Aliexpress	1	False Hierarchy
Amazon	4	Activity Message, Disguised Ad, Low Stock, Reference Pricing
Booking	7	Disguised Ad, 2x Limited Time Message, 2x Low Stock, 2x Reference Pricing
Eventim	2	Bad Defaults, Hidden Information
Expedia	5	2x Confirmshaming, False Hierarchy, Positive Framing, Testimonials
Gotogate	2	False Hierarchy, Reference Pricing
Mtrip	2	Confirmshaming, Trick Question
Opodo	5	Hidden Costs, 2x High Demand, Reference Pricing, Visual Prominence
Opodo2	3	False Hierarchy, Hidden Costs, Low Stock, Forced Registration
Pelacase	3	Bad Defaults, Reference Pricing, Testimonials
Riverisland	3	Bad Defaults, False Hierarchy, Hidden Information
Ryanair	5	False Hierarchy, Partitioned Pricing, Personalization, Positive Framings, Visual Prominence
telegraph	2	Hidden Information, Visual Prominence
Theguardian	5	3x Confirmshaming, Nagging, Visual Prominence
Viagogo	12	Activity Message, 2x False Hierarchy, 3x High Demand, Limited Time Message, 4x Low Stock, Visual Prominence

Table A.3: Web pages and web elements we included in our dataset and the amount and types of deceptive patterns included in each. 'K_' are taken from Krauß et al. [2025] and 'S_' ones from Schäfer et al. [2025].

Appendix B

Prompts

In the following, we will provide the prompts we used for each setting in our technical pipeline. Each prompt consisted of the *Output Format*-Prompt for the respective LLM, then either includes the *Initial Prompt* with possibly a *Prompt Addition*, or a *New Prompt*. All but the prompts for the prompting strategies also included a *Persona*.

Output Format: Generator/Baseline This is the addition we made to the prompt from Schäfer et al. [2025], and therefore also to each prompt of the LLM generator in our LLM-as-a-Judge pipelines. The *italic* part has been added after Gemini 2.5 Flash struggled to follow this syntax. The prompt describes how the LLM should output the changes to apply to the HTML, so our implementation works:

```
‘‘Please only output the changes and not the
whole HTML. Your output should be of the
following Syntax: ‘Change i: BeforeLLM i:
y AfterLLM i: z Change i end’ i being the
number of the change made (start counting at
1). y being the code that is being changed
for the new code z (altered by you). Make
sure to always follow this syntax and that
the original code y is the exact code found
in the HTML code. Don’t change any words
```

of this, exchange only i y and z. *Make sure to include all three exchanges i's in your output, that means change all i's to the respective number.* Don't include any dots or other symbols. Don't include any quotation marks at the start of any part (e.g., don't write 'y' for y being the original HTML). Keep everything exactly as it is. You can include an explanation for all changes at the start or end of your output. If the website isn't manipulative, please include 'No changes necessary' in your output. You can also include an explanation here''

Output Format: Judge The instruction included in the prompt for the LLM judge describes the output format, so we can extract the feedback:

''Your output should be of the following syntax if the website contains manipulation and you have feedback on how to remove it: 'Feedback start: x Feedback end' x being the feedback you provide to remove the manipulation from the website. Make sure to always follow this syntax. Don't change any words of this, exchange only x. Don't include any dots or other symbols. Don't include any quotation marks at the start of any part (e.g., don't write 'x' for x being the feedback). Keep everything exactly as it is. You can include an explanation for the feedback at the start or end of your output. If the website doesn't contain any manipulation, then don't provide feedback (and thus DON'T follow the syntax), please include 'Not manipulative anymore' in your output. You can also include an explanation here.''

Initial Prompt: Judge Initial prompt for the LLM Judges:

```
‘‘Check if the following website is
manipulative. Only check for manipulation,
don’t check for anything else. If
the website is manipulative, please
give feedback how [...] to remove the
manipulation. Don’t hallucinate any
feedback.’’
```

Initial Prompt: Generator Initial prompt for the LLM generators:

```
‘‘Make that less manipulative. There is
feedback included that you should take
into account to make the website less
manipulative.’’
```

Prompt Addition: Personas Judge Persona added to the prompt of the judge:

```
‘‘You are an AI assistant that helps to
evaluate websites’’
```

Prompt Addition: Personas Generator Persona added to the prompt of the generator:

```
‘‘You are an AI assistant that helps to
design websites’’
```

Prompt Addition: Chain-of-Thought Chain-of-Thought prompt that was added to the prompt of the judge and generator:

```
‘‘Let’s think step by step.’’
```

Prompt Generator: Feedback + Autonomy Addition to the prompt of the generator to encourage critical thinking in *Feedback + Autonomy*:

```
‘‘However, please think critically about the
feedback and if it actually makes sense to
apply it or parts of it. Also check on your
own if there are other things, that might
not have been mentioned in the feedback,
that should be changed to make the website
less manipulative.’’
```

New Prompt: Criteria in Prompt Judge Changed prompt for the LLM judge for *Criteria in Prompt* from the second iteration on:

```
‘‘Check if the following website (website 1)
is manipulative. If the website (website
1) is manipulative, please give feedback
how to remove the manipulation. Don’t
hallucinate any feedback. [...] Website
1 originated from website 2 in an attempt
to remove manipulation from the website,
i.e. website 1 should be a version of
website 2 with less manipulation. Check
if the following website (website 1)
that you should evaluate has kept all of
it’s functionality, i.e. check if the
website (website 1) has exactly the same
functionality as the website I sent you as a
comparison (website 2). Functionality from
website 2 that is manipulative shouldn’t
be in website 1, thus you can ignore those
changes and don’t need to address them. If
the functionality in website 1 is modified
then give feedback how to change the website
again to restore the functionality. Check
if the following website (website 1) that
you should evaluate has kept all of it’s
information in the process, i.e. check if
```

```
the website (website 1) has all information
that the website I sent you as a comparison
(website 2) has too.  And that website 1
doesn't have information that website 2
doesn't have.  Information from website
2 that is manipulative shouldn't be in
website 1, thus you can ignore those changes
and don't need to address them.  If the
information in website 1 is different, i.e.
(non-manipulative) information is missing
or additional information is added, then
give feedback how to change the website
again to restore the original information
state.  Nothing manipulative should be added
back into the website.  This is website
1, i.e.  the html that you should evaluate
and check if it is manipulative and that
all functionality and information is kept:
[HTML] This is website 2, i.e.  the html
you should use as a comparison to check that
all (non-manipulative) functionality and
information from this website is also in the
html of website 1 you evaluate:  [HTML]''
```

New Prompt: *Criteria in Prompt + Multiple Judges* Generator Changed prompt for the LLM generator for *Criteria in Prompt* and *Multiple Judges* from the second iteration on:

```
‘‘Make that less manipulative.  There is
feedback included that you should take into
account to make the website (website 1) less
manipulative and possibly also to adjust
the website's (website 1) functionality
and information.  [...]  This is website
1.  This is the website (website 1) that you
should make less manipulative and possibly
also adjust the functionality/information
(if that is mentioned in the feedback):
[HTML] This is website 2.  Feedback about
functionality and information references the
```

```
following html page (website 2) if it talks
about something that needs to be restored to
an old state (the old state is what you see
in this website): [HTML]''
```

New Prompt: Multiple Judges (Judge Functionality) Changed prompt for the LLM judge for FUNCTIONALITY for *Multiple Judges,* starts from the second iteration on:

```
‘‘You are an AI assistant that helps to
evaluate websites. Please focus on
functionality. Website 1 originated
from website 2 in an attempt to remove
manipulation from the website, i.e. website
1 should be a version of website 2 with
less manipulation. Check if the following
website (website 1) that you should evaluate
has kept all of it’s functionality. I.e.
check if the website (website 1) has
exactly the same functionality as the
website I sent you as a comparison (website
2). Functionality from website 2 that is
manipulative shouldn’t be in website 1, thus
you can ignore those changes. Check only
for functionality. If the functionality
in website 1 is modified then give feedback
how to change the website again to restore
the functionality. Your output should be
of the following syntax if the website has
functionality that needs to be changed and
you have feedback on how to restore it:
‘Feedback start: x Feedback end’ x being
the feedback you provide to restore the
functionality. Make sure to always follow
this syntax. Don’t change any words of
this, exchange only x. Don’t include any
dots or other symbols. Don’t include any
quotation marks at the start of any part
(e.g. don’t write ‘x’ for x being the
feedback). Keep everything exactly as it
```

```
is.  You can include an explanation for the
feedback at the start or end of your output.
If the website doesn't need to be adjusted
then don't provide feedback (and thus DON'T
follow the syntax), please include 'No
changes necessary' in your output.  You
can also include an explanation here.  This
is website 1, i.e.  the html that you should
evaluate and check if all functionality is
kept:  [HTML] This is website 2, i.e.  the
html you should use as a comparison to check
that all functionality from this website is
also in the html of website 1 you evaluate:
[HTML]''
```

New Prompt: Multiple Judges (Judge Information) Changed prompt for the LLM judge for INFORMATION for *Multiple Judges,* starts from the second iteration on:

```
''You are an AI assistant that helps
to evaluate websites.  Please focus
on information.  Website 1 originated
from website 2 in an attempt to remove
manipulation from the website, i.e.  website
1 should be a version of website 2 with
less manipulation.  Check if the following
website (website 1) that you should evaluate
has kept all of it's information in the
process.  I.e.  check if the website
(website 1) has all information that
the website I sent you as a comparison
(website 2) has too.  And that website 1
doesn't have information that website 2
doesn't have.  Information from website
2 that is manipulative shouldn't be in
website 1, thus you can ignore those
changes.  Check only for information.  If
the information in website 1 is different,
i.e.  (non-manipulative) information is
missing or additional information is
```

```
added, then give feedback how to change
the website again to restore the original
information state.  Your output should be
of the following syntax if the website has
information that needs to be changed and
you have feedback on how to restore it:
‘Feedback start:  x Feedback end’ x being
the feedback you provide to restore the
information state.  Make sure to always
follow this syntax.  Don’t change any words
of this, exchange only x.  Don’t include
any dots or other symbols.  Don’t include
any quotation marks at the start of any
part (e.g.  don’t write ‘x’ for x being the
feedback).  Keep everything exactly as it
is.  You can include an explanation for the
feedback at the start or end of your output.
If the website doesn’t need to be adjusted
then don’t provide feedback (and thus DON’T
follow the syntax), please include ‘No
changes necessary’ in your output.  You
can also include an explanation here.  This
is website 1, i.e.  the html that you should
evaluate and check if all information is
kept:  [HTML] This is website 2, i.e.  the
html you should use as a comparison to check
that all (non-manipulative) information from
this website is also in the html of website
1 you evaluate:  [HTML]’’
```

Prompt Addition: Guardrails (basic) Judge Basic guardrails added to the prompt of the judge:

```
‘‘Adhere to the following rules:  1.  Don’t
encourage hallucination, this means, don’t
suggest to add any information that isn’t
already given on the page.  2.  Don’t
encourage to remove any information or
functionality that isn’t manipulative.
Every action should still be possible and
```

```
every information should be accessible
afterwards, exceptions are information
and functionality that are manipulative.
3. Make sure you understand the website
and everything on it correctly. 4. Don't
suggest to change (non-manipulative)
functionality of the website. Everything
should work the same way as it did before,
so other functions (that we possibly don't
see, but access elements in the website)
can still be used. For example, don't flip
the meaning of a button. 5. If you suggest
to add an element to the website, then make
sure to also include a reminder to add the
functionality to the element too, if there
is functionality needed. 6. Make sure you
include all manipulative elements that are
on the website''
```

Prompt Addition: Guardrails (basic) Generator Basic guardrails added to the prompt of the judge:

```
‘‘Adhere to the following rules: 1.
Don't hallucinate. 2. Don't remove any
information or functionality that isn't
manipulative. Every action should still
be possible and every information should
be accessible afterwards, exceptions are
information and functionality that are
manipulative. 3. Make sure you understand
the website and everything on it correctly.
4. Don't change (non-manipulative)
functionality of the website. Everything
should work the same way as it did before,
so other functions (that we possibly don't
see, but access elements in the website)
can still be used. For example, don't flip
the meaning of a button. 5. If you add an
element to the website, then make sure to
add the functionality to the element too,
```

```
if there is functionality needed. 6. Make
sure the website still works correctly after
your changes. Be sure that everything is
still accessible.''
```

Prompt Addition: Guardrails (adjusted) Judge Adjusted guardrails added to the prompt of the generator:

```
‘‘Adhere to the following rules: 1. Don’t
encourage hallucination, this means, don’t
suggest to add any information that isn’t
already given on the page. 2. Don’t
encourage to remove any information or
functionality that isn’t manipulative.
Every action should still be possible and
every information should be accessible
afterwards. Make sure to find a way to
remove manipulation but keep the information
and functionality of the website the same.
3. Make sure you understand the website
and everything on it correctly. 4. Don’t
suggest to change (non-manipulative)
functionality of the website. Everything
should work the same way as it did before,
so other functions (that we possibly don’t
see, but access elements in the website)
can still be used. For example, don’t flip
the meaning of a button. 5. If you suggest
to add an element to the website, then make
sure to also include a reminder to add the
functionality to the element too, if there
is functionality needed. 6. Make sure you
include all manipulative elements that are
on the website.''
```

Prompt Addition: Guardrails (adjusted) Generator Adjusted guardrails added to the prompt of the generator:

```
‘‘Adhere to the following rules: 1.
Don’t hallucinate. 2. Don’t remove any
information or functionality that isn’t
manipulative. Every action should still
be possible and every information should be
accessible afterwards. Make sure to find
a way to remove manipulation but keep the
information and functionality of the website
the same. 3. Make sure you understand the
website and everything on it correctly.
4. Don’t change (non-manipulative)
functionality of the website. Everything
should work the same way as it did before,
so other functions (that we possibly don’t
see, but access elements in the website)
can still be used. For example, don’t flip
the meaning of a button. 5. If you add an
element to the website, then make sure to
add the functionality to the element too,
if there is functionality needed. 6. Make
sure the website still works correctly after
your changes. Be sure that everything is
still accessible.’’
```

Prompt Addition: Guardrails + DP Definitions Deceptive Pattern definitions added to the prompt of the judge and the generator alongside the basic guardrails:

```
‘‘Following are a few definitions of
dark/deceptive patterns (This list is not
exhaustive!) that manipulate users that
could help you in removing all manipulation
on the website: [Definitions in numbered
list, verbatim from Gray et al. [2024]
used]’’
```

Appendix C

Results: Refinement of LLM-as-a-Judge

In the following, we present all our results gathered during the iterative refinement in Chapter 3.

Pipeline	Criteria	Mean	SD	Success rate (%)	% of DPs removed
Baseline	DP removed	1.93	0.92	37.04	20
	DP added	3	0	100	/
	Functionality	2.78	0.64	88.88	/
	Information	2.44	0.8	62.96	/
	Design	2.67	0.68	77.78	/
	#Iterations	3	0	/	/
	Overall	2.56	0.77	14.81	/
Final LLM-as-a-Judge pipeline	DP removed	2.33	0.73	48.15	49.33
	DP added	2.85	0.46	88.89	/
	Functionality	2.74	0.66	85.19	/
	Information	2.7	0.72	85.19	/
	Design	2.89	0.42	92.59	/
	#Iterations	2.26	1.23	/	/
	Overall	2.7	0.64	37.04	/

Table C.1: Mean, standard deviation (SD), and success rate for all evaluation criteria for the baseline and our final LLM-as-a-Judge pipeline. The score ranges from 1 to 3; 3 is the highest value for each criterion, except #ITERATIONS. For #ITERATIONS, it applies that the lower the score, the better. Our LLM-as-a-Judge approach outperforms the baseline in DP REMOVED, INFORMATION, DESIGN, and the OVERALL score.

Pipeline	Criteria	Mean	SD	Success rate (%)
J: o4-mini, G: Gemini	DP removed	2.29	0.76	42.86
	DP added	3	0	100
	Functionality	2.43	0.98	71.43
	Information	2.14	1.07	57.14
	Design	3	0	100
	#Iterations	2.14	0.69	/
	Overall	2.57	0.78	14.29
J: o4-mini, G: Gemini Thinking	DP removed	2.43	0.53	42.86
	DP added	3	0	100
	Functionality	2.43	0.98	71.43
	Information	2.43	0.98	71.43
	Design	3	0	100
	#Iterations	2.29	0.76	/
	Overall	2.66	0.68	28.57
J: o4-mini, G: GPT-4o	DP removed	2.57	0.53	57.14
	DP added	2.86	0.38	85.71
	Functionality	2.14	1.07	57.14
	Information	2	1	42.86
	Design	2.71	0.49	71.43
	#Iterations	2	0.58	/
	Overall	2.46	0.78	28.57
J: Gemini, G: GPT-4o	DP removed	2	0.58	14.29
	DP added	2.71	0.76	85.71
	Functionality	2.86	0.38	85.71
	Information	2.43	0.98	71.43
	Design	2.57	0.79	71.43
	#Iterations	5.43	1.51	/
	Overall	2.51	0.74	42.86
J: Gemini Thinking, G: GPT-4o	DP removed	2.14	0.69	28.57
	DP added	2.86	0.38	85.71
	Functionality	2.57	0.79	71.43
	Information	2.14	1.07	57.14
	Design	2.43	0.79	57.14
	#Iterations	4.57	1.51	/
	Overall	2.43	0.78	14.29
J: GPT-4o, G: o4-mini	DP removed	2.14	0.9	42.86
	DP added	2.71	0.76	85.71
	Functionality	2	1	42.86
	Information	1.43	0.79	14.29
	Design	2.14	0.9	42.86
	#Iterations	4.86	1.57	/
	Overall	2.09	0.92	0

Table C.2: Mean, standard deviation (SD), and success rate for all evaluation criteria for each model combination tested in the pretest. The score ranges from 1 to 3; 3 is the highest value for each criterion, except #ITERATIONS. For #ITERATIONS, it applies that the lower the score, the better. GPT-4o as the judge performed the worst, each pipeline with o4-mini as the judge performed the best compared to other judges.

Pipeline	Criteria	Mean	SD	Success rate (%)
J: o4-mini, G: Gemini	DP removed	2.37	0.79	55.56
	DP added	2.93	0.38	96.3
	Functionality	2.56	0.85	77.78
	Information	2.56	0.85	77.78
	Design	2.89	0.42	92.59
	#Iterations	1.96	0.94	/
	Overall	2.54	0.71	37.04
J: o4-mini, G: Gemini Thinking	DP removed	2.56	0.64	62.96
	DP added	2.89	0.42	92.59
	Functionality	2.63	0.79	81.48
	Information	2.63	0.79	81.48
	Design	2.96	0.19	96.3
	#Iterations	2.3	1.23	/
	Overall	2.73	0.63	37.04
J: o4-mini, G: GPT-4o	DP removed	2.48	0.7	59.26
	DP added	2.93	0.27	92.59
	Functionality	2.41	0.93	70.37
	Information	2.52	0.8	70.37
	Design	2.81	0.4	81.48
	#Iterations	2.3	1.52	/
	Overall	2.57	0.69	37.04

Table C.3: Mean, standard deviation (SD), and success rate for all evaluation criteria for each model combination tested on the whole dataset. The score ranges from 1 to 3; 3 is the highest value for each criterion, except #ITERATIONS. For #ITERATIONS, it applies that the lower the score, the better. The combination with *o4-mini as the judge and Gemini-2.5-Flash Thinking as the generator* performed the best, while the one with *o4-mini as the judge and GPT-4o as the generator* performed the worst.

Pipeline	Criteria	Mean	SD	Success rate (%)	% of DPs removed
Persona	DP removed	2.56	0.64	62.96	70.67
	DP added	2.89	0.42	92.59	/
	Functionality	2.63	0.79	81.48	/
	Information	2.63	0.79	81.48	/
	Design	2.96	0.19	96.3	/
	#Iterations	2.3	1.23	/	/
	Overall	2.73	0.63	37.04	/
No Persona	DP removed	2.44	0.64	51.85	65.33
	DP added	2.85	0.46	88.89	/
	Functionality	2.74	0.66	85.19	/
	Information	2.67	0.73	81.48	/
	Design	2.89	0.32	88.89	/
	#Iterations	2.3	1.46	/	/
	Overall	2.72	0.59	33.33	/
J: No Persona, G: Persona	DP removed	2.48	0.64	55.56	64
	DP added	2.85	0.53	92.59	/
	Functionality	2.56	0.85	77.78	/
	Information	2.67	0.73	81.48	/
	Design	2.89	0.42	92.59	/
	#Iterations	2.3	1.32	/	/
	Overall	2.69	0.66	40.74	/
Few-Shot	DP removed	2.63	0.49	62.96	80
	DP added	2.96	0.19	96.3	/
	Functionality	2.56	0.85	77.78	/
	Information	2.41	0.89	66.67	/
	Design	2.85	0.46	88.89	/
	#Iterations	2.7	1.64	/	/
	Overall	2.68	0.65	44.44	/
Chain-of-Thought (CoT)	DP removed	2.56	0.64	62.96	70.67
	DP added	2.93	0.27	92.59	/
	Functionality	2.59	0.8	77.78	/
	Information	2.56	0.85	77.78	/
	Design	2.96	0.19	96.3	/
	#Iterations	2.04	0.85	/	/
	Overall	2.72	0.63	44.44	/
Personas + CoT	DP removed	2.48	0.7	59.26	65.33
	DP added	2.89	0.42	92.59	/
	Functionality	2.67	0.73	81.48	/
	Information	2.52	0.85	74.07	/
	Design	2.96	0.19	96.3	/
	#Iterations	2.19	1.33	/	/
	Overall	2.7	0.65	40.74	/

Table C.4: Mean, standard deviation (SD), and success rate for all evaluation criteria for each prompting strategy tested. The score ranges from 1 to 3; 3 is the highest value for each criterion, except #ITERATIONS. For #ITERATIONS, it applies that the lower the score, the better. *Personas* was marginally better than each pipeline we tested here overall, and *Few-Shot* performed the best in DP REMOVED, while lacking in FUNCTIONALITY and INFORMATION.

Pipeline	Criteria	Mean	SD	Success rate (%)	% of DPs removed
Feedback	DP removed	2.56	0.64	62.96	70.67
	DP added	2.89	0.42	92.59	/
	Functionality	2.63	0.79	81.48	/
	Information	2.63	0.79	81.48	/
	Design	2.96	0.19	96.3	/
	#Iterations	2.3	1.23	/	/
	Overall	2.73	0.63	37.04	/
Feedback + Autonomy	DP removed	2.56	0.64	62.96	72
	DP added	2.93	0.27	92.59	/
	Functionality	2.59	0.8	77.78	/
	Information	2.44	0.89	70.37	/
	Design	2.96	0.19	96.3	/
	#Iterations	2.07	1	/	/
	Overall	2.7	0.65	40.74	/
No Feedback	DP removed	2.37	0.74	51.85	50.67
	DP added	2.96	0.19	96.3	/
	Functionality	2.7	0.67	81.48	/
	Information	2.52	0.85	74.07	/
	Design	3	0	100	/
	#Iterations	2.37	1.45	/	/
	Overall	2.71	0.63	37.04	/

Table C.5: Mean, standard deviation (SD), and success rate for all evaluation criteria for each communication approach tested. The score ranges from 1 to 3; 3 is the highest value for each criterion, except #ITERATIONS. For #ITERATIONS, it applies that the lower the score, the better. *Feedback* generally performed the best, specifically in FUNCTIONALITY and INFORMATION.

Pipeline	Criteria	Mean	SD	Success rate (%)	% of DPs removed
No Criteria	DP removed	2.56	0.64	62.96	70.67
	DP added	2.89	0.42	92.59	/
	Functionality	2.63	0.79	81.48	/
	Information	2.63	0.79	81.48	/
	Design	2.96	0.19	96.3	/
	#Iterations	2.3	1.23	/	/
	Overall	2.73	0.63	37.04	/
Criteria in Prompt	DP removed	2.41	0.64	48.15	46.67
	DP added	2.89	0.42	92.59	/
	Functionality	2.7	0.72	85.19	/
	Information	2.7	0.72	85.19	/
	Design	2.96	0.19	96.3	/
	#Iterations	2.04	0.9	/	/
	Overall	2.73	0.6	37.04	/
Multiple Judges	DP removed	2.41	0.75	55.56	54.67
	DP added	2.89	0.42	92.59	/
	Functionality	2.59	0.8	77.78	/
	Information	2.59	0.8	77.78	/
	Design	2.89	0.42	92.59	/
	#Iterations	2.67	1.75	/	/
	Overall	2.67	0.68	44.44	/

Table C.6: Mean, standard deviation (SD), and success rate for all evaluation criteria for each way to add evaluation criteria to the prompt. The score ranges from 1 to 3; 3 is the highest value for each criterion, except #ITERATIONS. For #ITERATIONS, it applies that the lower the score, the better. *No Criteria* performed better generally better than the other two options, specifically in DP REMOVED.

Pipeline	Criteria	Mean	SD	Success rate (%)	% of DPs removed
No Guardrails	DP removed	2.56	0.64	62.96	70.67
	DP added	2.89	0.42	92.59	/
	Functionality	2.63	0.79	81.48	/
	Information	2.63	0.79	81.48	/
	Design	2.96	0.19	96.3	/
	#Iterations	2.3	1.23	/	/
	Overall	2.73	0.63	37.04	/
Guardrails (only Judge)	DP removed	2.56	0.58	59.26	70.67
	DP added	2.96	0.19	96.3	/
	Functionality	2.59	0.8	77.78	/
	Information	2.48	0.85	70.37	/
	Design	2.93	0.27	92.59	/
	#Iterations	2.33	1.49	/	/
	Overall	2.7	0.62	40.74	/
Guardrails (both)	DP removed	2.33	0.73	48.15	49.33
	DP added	2.85	0.46	88.89	/
	Functionality	2.74	0.66	85.19	/
	Information	2.7	0.72	85.19	/
	Design	2.89	0.42	92.59	/
	#Iterations	2.26	1.23	/	/
	Overall	2.7	0.64	37.04	/
Adjusted Guardrails	DP removed	2.52	0.64	59.26	60
	DP added	2.89	0.42	92.59	/
	Functionality	2.52	0.85	74.07	/
	Information	2.52	0.85	74.07	/
	Design	2.81	0.48	85.19	/
	#Iterations	2.93	1.59	/	/
	Overall	2.65	0.68	44.44	/
DP Definitions	DP removed	2.56	0.58	59.26	68
	DP added	2.96	0.19	96.3	/
	Functionality	2.63	0.79	81.48	/
	Information	2.56	0.8	74.07	/
	Design	2.96	0.19	96.3	/
	#Iterations	2.78	1.58	/	/
	Overall	2.73	0.6	44.44	/

Table C.7: Mean, standard deviation (SD), and success rate for all evaluation criteria for each guardrail variation tested. The score ranges from 1 to 3; 3 is the highest value for each criterion, except #ITERATIONS. For #ITERATIONS, it applies that the lower the score, the better. *Guardrails (both)* performed the best in FUNCTIONALITY and INFORMATION, while *No Guardrails* performed the best in removing DP REMOVED. Every other version did not perform better than *No Guardrails*.

Appendix D

User Study Questionnaires

Below are all questionnaires used in our user study.

Consent Form

Study: Exploring Deceptive Pattern Removal from Websites

Study Supervisor: Sophie Hahn
RWTH Aachen
sophie.hahn@rwth-aachen.de

Aim: The aim of this study is to evaluate the removal of manipulation from websites.

Procedure: First, you will be asked to fill out a sheet about demographics. After that, you will be shown a webpage or an element of a webpage and asked to make it less manipulative as well as provide an explanation. You will be asked to repeat this for multiple elements. Following this, you will be shown different webpages and an altered version of each webpage and will be asked to rate this altered version and again provide an explanation. This will be repeated for multiple webpages as well. At the end, an interview will take place. The study will take around 90 minutes.

Benefits: The results of this study will be useful for research around the removal of manipulation from websites. It will help get a better understanding of the output of our application.

Risks: There are no physical or mental risks in this study. The participants will be shown manipulative elements and designs, from which they will be asked to remove the manipulation from the designs or will be shown a less manipulative version and asked to rate it. The participants are always made aware that each webpage could contain manipulation.

Participation in this study is completely voluntary. The study can be terminated at any time without consequences for you. You can also always take breaks for as long as you want.

Costs and Compensation: Participation in this study will involve no cost to you. There will be snacks and drinks provided for you during the study.

Confidentiality: All data collected in this study will be treated confidentially. All participants and their data will be anonymized. No publications will allow conclusions to be drawn back to the individual people participating.

Recording: During the interview, your responses will be audio recorded for the purpose of accurate data collection. The audio will be transcribed and then permanently deleted afterwards.

- ◯ I have read and understood the information on this form.
- ◯ I have had the information on this form explained to me.
- ◯ I consent to the audio recording during the interview.

I consent to participating in this study:

_____	_____	_____
Participant's Name	Participant's Signature	Date

	_____	_____
	Principal Investigator	Date

Participant ID: _____

Demographics

Age _____

Gender _____

Current Occupation/Field of Study _____

Last achieved academic degree _____

Deceptive Patterns:

In this study, we use the terms “dark pattern” and “deceptive pattern” synonymously.

	Strongly disagree				Strongly agree
I know what deceptive patterns are.	◯	◯	◯	◯	◯
I am familiar with the research around deceptive patterns.	◯	◯	◯	◯	◯
I often notice deceptive patterns on websites.	◯	◯	◯	◯	◯
I take measures to avoid and not fall victim to deceptive patterns on websites.	◯	◯	◯	◯	◯
I have engaged with the topic of deceptive patterns before.	◯	◯	◯	◯	◯

If applicable, please describe in what way and how thorough you have come in contact and engaged with the topic of deceptive patterns (e.g., wrote a thesis, read papers, participated in a study, actively avoid websites that use deceptive patterns, be cautious to not fall victim to deceptive patterns in e.g. cookie banners, …):

Webpage ID: _____ Participant ID: _____

Webpage Alteration

Task: **Make that less manipulative.**

Please write down the changes you would like to make to the webpage and/or draw it in the box below. Please make sure to write down "**how**" exactly you would change the webpage.

Written changes:

Drawn changes:

Webpage ID: _____ Participant ID: _____

Justification/Explanation for your changes:

Are there any elements that you chose not to change, e.g. manipulation you didn't want to remove?

Further comments:

Webpage ID: _____ Participant ID: _____

Webpage Rating

Original webpage:

These questions refer to the original webpage.

	Strongly disagree				Strongly agree	n.a.
The original webpage is manipulative.	◯	◯	◯	◯	◯	—
The manipulation on the webpage is severe.	◯	◯	◯	◯	◯	◯

Justifications:

Altered webpage:

These questions refer to the altered webpage.

	Strongly disagree				Strongly agree
All manipulation that should be removed is removed.	◯	◯	◯	◯	◯
All functionality that should be kept is kept.	◯	◯	◯	◯	◯
All information that should be kept is kept.	◯	◯	◯	◯	◯
The design wasn't influenced in a negative way. (This rating doesn't consider any design changes necessary to remove manipulation.)	◯	◯	◯	◯	◯

Justifications:

Webpage ID: _____ Participant ID: _____

The altered webpage feels … than the original webpage.

Worse ◯ ◯ ◯ ◯ ◯ *Better*

I would rather use the altered version than the original version.

Disagree ◯ ◯ ◯ ◯ ◯ *Agree*

Justifications:

Further Comments:

Appendix E

Results: User Study

E.1 Task 1: Agreement for each Deceptive Pattern Instance

In the following, we depict the agreement between the LLM and the participants in our user study for every deceptive pattern in every web page. It shows each deceptive pattern, as well as the alteration the LLM performed for the deceptive pattern on each web page, as well as the agreement for that alteration.

DeceptivePattern	Web page	Alteration	Agreement (%)
Bad Defaults	Pelacase (F)	Remove	85.71
	Riverisland (G)	Remove	57.14
Positive Framing	Expedia (C)	Remove	85.71
	Ryanair (H)	Remove	66.67
Hidden Information	Riverisland (G)	Do not remove	85.71
Nagging	Theguardian (J)	Popup not in the way	33.33
Disguised Ad	Amazon (B)	Do not remove	14.29
	Booking (I)	Remove info that it is an ad	37.5
Hidden Price	Opodo (E)	Do not remove	57.14
Partitioned Pricing	Ryanair (H)	Do not remove	16.67
Disguised Ad	Amazon (B)	Remove	71.43
	Gotogate (D)	Do not remove	16.67
	Opodo (E)	Do not remove	28.57
	Pelacase (F)	Do not remove	42.86
	Booking (I)	Remove	37.5
False Hierarchy	Aliexpress (A)	Both less prominent	57.14
	Expedia (C)	Both less prominent	100
	Expedia (C)	Add elements	57.14
	Expedia (C)	Remove elements	28.57
	Gotogate (D)	Both more prominent	50
	Gotogate (D)	Add elements	33.33
	Opodo (E)	Do not remove	42.86
	Riverisland (G)	Add element	85.71
	Ryanair (H)	Both less prominent	100
	Viagogo (K)	Do not remove	0
Visual Hierarchy	Gotogate (D)	Both less prominent	16.67
	Theguardian (J)	Remove	0
	Viagogo (K)	Remove	0

Table E.1: The alterations the LLM made for each deceptive pattern on each web page and the percentage of participants that agreed with this.

DeceptivePattern	Web page	Alteration	Agreement (%)
High Demand	Opodo (E) ["Selling out"]	Remove	57.14
	Opodo (E) [Flame]	Do not remove	28.57
	Viagogo (K) ["Selling fast"]	Remove	71.43
	Viagogo (K) ["Hottest event"]	Remove	42.86
Low Stock	Amazon (B)	Remove	42.86
	Booking (I)	Remove	25
	Low Stock (K)	Remove	14.29
Testimonial	Expedia (C)	Remove	85.71
	Pelacase (F)	Do not remove	0
Activity Message	Amazon (B)	Remove	42.86
	Viagogo (K)	Remove	42.86
Limited Time Message	Booking (I)	Remove	37.5
	Booking (I)	Remove	37.5
	Viagogo (K)	Remove	14.29
Confirmshaming	Expedia (C) [Yes]	Remove shaming + clutter	57.14
	Expedia (C) [No]	Remove shaming + clutter	57.14
	Theguardian (J) [Header]	Remove shaming	33.33
	Theguardian (J) [Text 1]	Remove shaming	16.67
	Theguardian (J) [Text 2]	Remove text	16.67
Personalization	Ryanair (H)	Remove text	0

Table E.2: The alterations the LLM made for each deceptive pattern on each web page and the percentage of participants that agreed with this.

E.2 Task 2: Rating for Each Website

Below is the arithmetic mean of all rating for each question for each web page in our user study.

Website	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
Aliexpress (A)	4.13	2.88	5	5	4.75	5	4.25	4.38
Amazon (B)	4.5	3.13	4.5	5	4.38	5	4.5	4.38
Expedia (C)	4.63	4	4.88	3.88	2.13	4.25	3.25	2.75
Gotogate (D)	4.67	4	3.89	4.56	5	4.22	4.22	4.44
Opodo (E)	3.88	3.25	2.25	5	4.5	5	3.88	4.25
Pelacase (F)	4.75	4.75	3.25	5	5	5	4.25	4.75
Riverisland (G)	4.75	3.75	4	5	4.75	4.88	4.75	5
Ryanair (H)	4.89	4.11	3.44	5	5	4.11	4.33	5
Booking (I)	4.43	3.29	3.71	5	3.29	5	4	3.71
Theguardian (J)	5	4.44	3.78	4.44	3.33	4.22	4	4.11
Viagogo (K)	4.75	4.25	3.25	5	4.63	4.63	4.63	4.63

Table E.3: Overall scores for each website obtained through the ratings of task 2. The scale ranges from 1 to 5, "5" means that participants strongly agreed. (Q1: Original was manipulative, Q2: Manipulation is severe, Q3: All Manipulation is removed, Q4: All Functionality is kept, Q5: All Information is kept, Q6: Design not negatively influenced, Q7: Altered feels better, Q8: Preference towards altered)

Bibliography

[1] Sanju Ahuja, Johanna Gunawan, Nataliia Bielova, and Cristiana Teixeira Santos. Towards Key Contributing Factors in Identifying Dark Pattern Autonomy Violations under the EU Digital Services Act. In *Companion Publication of the 2025 ACM Designing Interactive Systems Conference*, DIS '25 Companion, page 501–507, New York, NY, USA, 2025. Association for Computing Machinery. doi.org/10.1145/3715668.3736336.

[2] Kalya Win Aung, Ewan Soubutts, and Aneesha Singh. "What a stupid way to do business": Towards an Understanding of Older Adults' Perceptions of Deceptive Patterns and Ways to Develop Resistance. *Proceedings of the ACM on Human-Computer Interaction*, 8(CHI PLAY):348:1–348:31, October 2024. doi.org/10.1145/3677113.

[3] Simone Avolicino, Marianna Di Gregorio, Fabio Palomba, Marco Romano, Monica Sebillo, and Giuliana Vitiello. AI-Based Emotion Recognition to Study Users' Perception of Dark Patterns. In *HCI International 2022 - Late Breaking Papers. Design, User Experience and Interaction*, HCII 2022, page 185–203. Springer International Publishing, 2022. doi.org/10.1007/978-3-031-17615-9_13.

[4] Karim Benharrak, Tim Zindulka, and Daniel Buschek. Deceptive Patterns of Intelligent and Interactive Writing Assistants. In *Proceedings of the Third Workshop on Intelligent and Interactive Writing Assistants*, In2Writing '24, page 62–64, New York, NY, USA, 2024. Association for Computing Machinery. doi.org/10.1145/3690712.3690728.

[5] Aditi M. Bhoot, Mayuri A. Shinde, and Wricha P. Mishra. Towards the Identification of Dark Patterns: An Analysis Based on End-User Reactions. In *Proceedings of the 11th Indian Conference on Human-Computer Interaction*, IndiaHCI '20, page 24–33, New York, NY, USA, 2020. Association for Computing Machinery. doi.org/10.1145/3429290.3429293.

[6] Kerstin Bongard-Blanchy, Arianna Rossi, Salvador Rivas, Sophie Doublet, Vincent Koenig, and Gabriele Lenzini. "I Am Definitely Manipulated, Even When

I Am Aware of It. It's Ridiculous!" - Dark Patterns from the End-User Perspective. In *Proceedings of the 2021 ACM Designing Interactive Systems Conference*, DIS '21, page 763–776, New York, NY, USA, 2021. Association for Computing Machinery. doi.org/10.1145/3461778.3462086.

[7] Harry Brignull. *Deceptive patterns: Exposing the tricks tech companies use to control you*. Testimonium Ltd, 2023. ISBN 978-1739454401.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33 of *NeurIPS 2020*, pages 1877–1901. Curran Associates, Inc., 2020.

[9] Ching Chang, Wei-Yao Wang, Wen-Chih Peng, and Tien-Fu Chen. LLM4TS: Aligning Pre-Trained LLMs as Data-Efficient Time-Series Forecasters. *ACM Trans. Intell. Syst. Technol.*, 16(3), April 2025. doi.org/10.1145/3719207.

[10] Jieshan Chen, Jiamou Sun, Sidong Feng, Zhenchang Xing, Qinghua Lu, Xiwei Xu, and Chunyang Chen. 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*, UIST '23, New York, NY, USA, 2023. Association for Computing Machinery. doi.org/10.1145/3586183.3606783.

[11] Ziwei Chen, Jiawen Shen, Kristen Vaccaro, and Luna. Hidden Darkness in LLM-Generated Designs: Exploring Dark Patterns in Ecommerce Web Components Generated by LLMs. *arXiv preprint arXiv:2502.13499*, 2025. doi.org/10.48550/arXiv.2502.13499.

[12] Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and others. Gemini 2.5: Pushing the Frontier with Advanced Reasoning, Multimodality, Long Context, and Next Generation Agentic Capabilities. *arXiv preprint arXiv:2507.06261*, 2025. doi.org/10.48550/arXiv.2507.06261.

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

[14] Andrea Curley, Dympna O'Sullivan, Damian Gordon, Brendan Tierney, and Ioannis Stavrakakis. The Design of a Framework for the Detection of Web-Based Dark Patterns. In *The Fifteenth International Conference on Digital Society*, ICDS 2021, pages 24–30, Nice, France, 2021. IARIA. doi.org/10.21427/20g8-d176.

[15] Tim de Jonge, Hanna Schraffenberger, Jorrit Geels, Jaap-Henk Hoepman, Marie-Sophie Simon, and Frederik Zuiderveen Borgesius. If Deceptive Patterns are the problem, are Fair Patterns the solution? In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '25, page 3131–3137, New York, NY, USA, 2025. Association for Computing Machinery. doi.org/10.1145/3715275.3732199.

[16] Dorottya Demszky, Diyi Yang, David S Yeager, Christopher J Bryan, Margarett Clapper, Susannah Chandhok, Johannes C Eichstaedt, Cameron Hecht, Jeremy Jamieson, Meghann Johnson, Michaela Jones, Danielle Krettek-Cobb, Leslie Lai, Nirel JonesMitchell, Desmond C. Ong, Carol S. Dweck, James J. Gross, and James W. Pennebaker. Using large language models in psychology. *Nature Reviews Psychology*, 2(11):688–701, 2023. doi.org/10.1038/s44159-023-00241-5.

[17] Linda Di Geronimo, Larissa Braz, Enrico Fregnan, Fabio Palomba, and Alberto Bacchelli. 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*, CHI '20, page 1–14, New York, NY, USA, 2020. Association for Computing Machinery. doi.org/10.1145/3313831.3376600.

[18] Francesco Fabbri, Gustavo Penha, Edoardo D'Amico, Alice Wang, Marco De Nadai, Jackie Doremus, Paul Gigioli, Andreas Damianou, Oskar Stål, and Mounia Lalmas. Evaluating Podcast Recommendations with Profile-Aware LLM-as-a-Judge. In *Proceedings of the Nineteenth ACM Conference on Recommender Systems*, RecSys '25, page 1181–1186, New York, NY, USA, 2025. Association for Computing Machinery. doi.org/10.1145/3705328.3759305.

[19] Kevin Fiedler, René Schäfer, Jan Borchers, and René Röpke. "Deception Detected!"—A Serious Game About Detecting Dark Patterns. In Avo Schönbohm, Francesco Bellotti, Antonio Bucchiarone, Francesca de Rosa, Manuel Ninaus, Alf Wang, Vanissa Wanick, and Pierpaolo Dondio, editors, *Games and Learning Alliance*, pages 191–200, Cham, 2025. Springer Nature Switzerland. doi.org/10.1007/978-3-031-78269-5_18.

[20] Colin M. Gray, Yubo Kou, Bryan Battles, Joseph Hoggatt, and Austin L. Toombs. The Dark (Patterns) Side of UX Design. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, page 1–14,

New York, NY, USA, 2018. Association for Computing Machinery. doi.org/10.1145/3173574.3174108.

[21] Colin M. Gray, Cristiana Santos, Nataliia Bielova, Michael Toth, and Damian Clifford. Dark Patterns and the Legal Requirements of Consent Banners: An Interaction Criticism Perspective. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. doi.org/10.1145/3411764.3445779.

[22] Colin M. Gray, Cristiana Teixeira Santos, Nataliia Bielova, and Thomas Mildner. 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*, CHI '24, New York, NY, USA, 2024. Association for Computing Machinery. doi.org/10.1145/3613904.3642436.

[23] Colin M. Gray, Thomas Mildner, and Ritika Gairola. Getting Trapped in Amazon's "Iliad Flow": A Foundation for the Temporal Analysis of Dark Patterns. CHI '25, New York, NY, USA, 2025. Association for Computing Machinery. doi.org/10.1145/3706598.3713828.

[24] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. A Survey on LLM-as-a-Judge. *arXiv preprint arXiv:2411.15594*, 2024. doi.org/10.48550/arXiv.2411.15594.

[25] S. M. Hasan Mansur, Sabiha Salma, Damilola Awofisayo, and Kevin Moran. AidUI: Toward Automated Recognition of Dark Patterns in User Interfaces. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pages 1958–1970. IEEE, 2023. doi.org/10.1109/ICSE48619.2023.00166.

[26] Philip Hausner and Michael Gertz. 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*), 2021. URL https://dbs.ifi.uni-heidelberg.de/files/Team/phausner/publications/Hausner_Gertz_CHI2021.pdf.

[27] Shun Hidaka, Sota Kobuki, Mizuki Watanabe, and Katie Seaborn. 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*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. doi.org/10.1145/3544548.3580942.

[28] Helge Holzmann, Wolfgang Nejdl, and Avishek Anand. The Dawn of Today's Popular Domains: A Study of the Archived German Web over 18 Years. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries*,

JCDL '16, page 73–82, New York, NY, USA, 2016. Association for Computing Machinery. doi.org/10.1145/2910896.2910901.

[29] Renjun Hu, Yi Cheng, Libin Meng, Jiaxin Xia, Yi Zong, Xing Shi, and Wei Lin. Training an LLM-as-a-Judge Model: Pipeline, Insights, and Practical Lessons. In *Companion Proceedings of the ACM on Web Conference 2025*, WWW '25, page 228–237, New York, NY, USA, 2025. Association for Computing Machinery. doi.org/10.1145/3701716.3715265.

[30] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large Language Models Cannot Self-Correct Reasoning Yet. *arXiv preprint arXiv:2310.01798*, 2023. doi.org/10.48550/arXiv.2310.01798.

[31] Chip Huyen. How to Evaluate AI that's Smarter than Us: Exploring three strategies: functional correctness, AI-as-a-judge, and comparative evaluation. *Queue*, 23(1):39–63, April 2025. doi.org/10.1145/3722043.

[32] Raisa Islam and Owana Marzia Moushi. GPT-4o: The Cutting-Edge Advancement in Multimodal LLM. In Kohei Arai, editor, *Intelligent Computing*, pages 47–60, Cham, 2025. Springer Nature Switzerland.

[33] Junseok Kim, Nakyeong Yang, and Kyomin Jung. Persona is a Double-edged Sword: Mitigating the Negative Impact of Role-playing Prompts in Zero-shot Reasoning Tasks. *arXiv preprint arXiv:2408.08631*, 2024. doi.org/10.48550/arXiv.2408.08631.

[34] Emre Kocyigit, Arianna Rossi, Anastasia Sergeeva, Claudia Negri Ribalta, Ali Farjami, and Gabriele Lenzini. DeceptiLens: an Approach supporting Transparency in Deceptive Pattern Detection based on a Multimodal Large Language Model. In *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '25, page 1942–1959, New York, NY, USA, 2025. Association for Computing Machinery. doi.org/10.1145/3715275.3732129.

[35] Pruthvi S Kodmurgi, Srihari Adiga, Srikrshna Parthasarthy, R Thanushree, Varun Vishwanatha Avabratha, Preet Kanwal, and Prasad B Honnavalli. ScrapeAI: A Multi-Modal Approach to Detect Dark Patterns. In *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, ICCCNT '24, pages 1–6, 2024. doi.org/10.1109/ICCCNT61001.2024.10723319.

[36] Esben Kran, Hieu Minh Nguyen, Akash Kundu, Sami Jawhar, Jinsuk Park, and Mateusz Maria Jurewicz. DarkBench: Benchmarking Dark Patterns in Large Language Models. In *The Thirteenth International Conference on Learning Representations*, ICLR 2025, 2025. doi.org/10.48550/arXiv.2503.10728.

[37] Veronika Krauß, Pejman Saeghe, Alexander Boden, Mohamed Khamis, Mark McGill, Jan Gugenheimer, and Michael Nebeling. What Makes XR Dark? Examining Emerging Dark Patterns in Augmented and Virtual Reality through Expert Co-Design. *ACM Trans. Comput.-Hum. Interact.*, 31(3), August 2024. doi.org/10.1145/3660340.

[38] Veronika Krauß, Mark McGill, Thomas Kosch, Yolanda Thiel, Dominik Schön, and Jan Gugenheimer. "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*, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery. doi.org/10.1145/3706598.3713083.

[39] Chiara Krisam, Heike Dietmann, Melanie Volkamer, and Oksana Kulyk. Dark Patterns in the Wild: Review of Cookie Disclaimer Designs on Top 500 German Websites. In *Proceedings of the 2021 European Symposium on Usable Security*, EuroUSEC '21, page 1–8, New York, NY, USA, 2021. Association for Computing Machinery. doi.org/10.1145/3481357.3481516.

[40] Kirill Kronhardt, Kevin Rolfes, and Jens Gerken. Trickery: Exploring a Serious Game Approach to Raise Awareness of Deceptive Patterns. In *Proceedings of the International Conference on Mobile and Ubiquitous Multimedia*, MUM '24, page 133–147, New York, NY, USA, 2024. Association for Computing Machinery. doi.org/10.1145/3701571.3701588.

[41] Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. From Generation to Judgment: Opportunities and Challenges of LLM-as-a-judge. *arXiv preprint arXiv:2411.16594*, 2024. doi.org/10.48550/arXiv.2411.16594.

[42] Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Methods. *arXiv preprint arXiv:2412.05579*, 2024. doi.org/10.48550/arXiv.2412.05579.

[43] Alexander Löbel, René Schäfer, Hanna Püschel, Esra Güney, and Ulrike Meyer. Access Your Data... if You Can: An Analysis of Dark Patterns Against the Right of Access on Popular Websites. In Meiko Jensen, Cédric Lauradoux, and Kai Rannenberg, editors, *Privacy Technologies and Policy*, pages 23–47, Cham, Switzerland, 2024. Springer Nature Switzerland.

[44] Yuwen Lu, Chao Zhang, Yuewen Yang, Yaxing Yao, and Toby Jia-Jun Li. From Awareness to Action: Exploring End-User Empowerment Interventions for Dark Patterns in UX. *Proc. ACM Hum.-Comput. Interact.*, 8(CSCW1), April 2024. doi.org/10.1145/3637336.

[45] Jamie Luguri and Lior Jacob Strahilevitz. Shining a Light on Dark Patterns. *Journal of Legal Analysis*, 13(1):43–109, 03 2021. doi.org/10.1093/jla/laaa006.

[46] Francisco Lupiáñez-Villanueva, Alba Boluda, Francesco Bogliacino, Giovanni Liva, Lucie Lechardoy, and Teresa Rodríguez de las Heras Ballell. *Behavioural Study on Unfair Commercial Practices in the Digital Environment: Dark Patterns and Manipulative Personalisation*. Publications Office of the European Union, Luxembourg, Luxembourg, 2022. doi.org/10.2838/859030.

[47] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative Refinement with Self-Feedback. *Advances in Neural Information Processing Systems*, 36: 46534–46594, 2023.

[48] Arunesh Mathur, Gunes Acar, Michael J. Friedman, Eli Lucherini, Jonathan Mayer, Marshini Chetty, and Arvind Narayanan. Dark Patterns at Scale: Findings from a Crawl of 11K Shopping Websites. *Proceedings of the ACM on Human-Computer Interaction*, 3(CSCW), Nov 2019. doi.org/10.1145/3359183.

[49] Arunesh Mathur, Mihir Kshirsagar, and Jonathan Mayer. 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*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. doi.org/10.1145/3411764.3445610.

[50] Thomas Mildner, Gian-Luca Savino, Philip R. Doyle, Benjamin R. Cowan, and Rainer Malaka. 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*, CHI '23, New York, NY, USA, 2023. Association for Computing Machinery. doi.org/10.1145/3544548.3580695.

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

[52] Arvind Narayanan, Arunesh Mathur, Marshini Chetty, and Mihir Kshirsagar. Dark Patterns: Past, Present, and Future: The evolution of tricky user interfaces. *Queue*, 18(2):67–92, May 2020. doi.org/10.1145/3400899.3400901.

[53] Liming Nie, Yangyang Zhao, Chenglin Li, Xuqiong Luo, and Yang Liu. Shadows in the Interface: A Comprehensive Study on Dark Patterns. *Proc. ACM Softw. Eng.*, 1(FSE), jul 2024. doi.org/10.1145/3643736.

[54] Sam Niknejad, Thomas Mildner, Nima Zargham, Susanne Putze, and Rainer Malaka. Level Up or Game Over: Exploring How Dark Patterns Shape Mobile Games. In *Proceedings of the International Conference on Mobile and Ubiquitous Multimedia*, MUM '24, page 148–156, New York, NY, USA, 2024. Association for Computing Machinery. doi.org/10.1145/3701571.3701604.

[55] Harsha Nori, Naoto Usuyama, Nicholas King, Scott Mayer McKinney, Xavier Fernandes, Sheng Zhang, and Eric Horvitz. From Medprompt to o1: Exploration of Run-Time Strategies for Medical Challenge Problems and Beyond. *arXiv preprint arXiv:2411.03590*, 2024. doi.org/10.48550/arXiv.2411.03590.

[56] Midas Nouwens, Ilaria Liccardi, Michael Veale, David Karger, and Lalana Kagal. 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*, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery. doi.org/10.1145/3313831.3376321.

[57] Bhrij Patel, Souradip Chakraborty, Wesley A Suttle, Mengdi Wang, Amrit Singh Bedi, and Dinesh Manocha. AIME: AI System Optimization via Multiple LLM Evaluators. *arXiv preprint arXiv:2410.03131*, 2024. doi.org/10.48550/arXiv.2410.03131.

[58] Max Peeperkorn, Tom Kouwenhoven, Dan Brown, and Anna Jordanous. Is Temperature the Creativity Parameter of Large Language Models? *arXiv preprint arXiv:2405.00492*, 2024. doi.org/10.48550/arXiv.2405.00492.

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

[60] Marie Potel-Saville and Mathilde Francois. From Dark Patterns to Fair Patterns? Usable Taxonomy to Contribute Solving the Issue with Countermeasures. In *Annual Privacy Forum*, 06 2023.

[61] Parinda Rahman and Ifeoma Adaji. Dark Patterns in Shopping, Education & Health Apps. In *2024 IEEE Digital Platforms and Societal Harms (DPSH)*, DPSH '24, pages 1–8, 2024. doi.org/10.1109/DPSH60098.2024.10775239.

[62] Hauke Sandhaus. Promoting Bright Patterns. Position Paper at the Workshop "Designing Technology and Policy Simultaneously: Towards A Research Agenda and New Practice" at the CHI Conference on Human Factors in Computing Systems (CHI '23), 2023. URL https://doi.org/10.48550/arXiv.2304.01157.

[63] Yasin Sazid, Mridha Md. Nafis Fuad, and Kazi Sakib. Automated Detection of Dark Patterns Using In-Context Learning Capabilities of GPT-3. In *2023*

30th Asia-Pacific Software Engineering Conference (APSEC), APSEC '23, pages 569–573, Seoul, Republic of Korea, 2023. Institute of Electrical and Electronics Engineers (IEEE). doi.org/10.1109/APSEC60848.2023.00072.

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

[65] René Schäfer, Sarah Sahabi, Annabell Brocker, and Jan Borchers. Growing Up With Dark Patterns: How Children Perceive Malicious User Interface Designs. In *Proceedings of the 13th Nordic Conference on Human-Computer Interaction*, NordiCHI '24, New York, NY, USA, 2024. Association for Computing Machinery. doi.org/10.1145/3679318.3685358.

[66] René Schäfer, Paul Miles Preuschoff, Rene Niewianda, Sophie Hahn, Kevin Fiedler, and Jan Borchers. Don't Detect, Just Correct: Can LLMs Defuse Deceptive Patterns Directly? In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '25, New York, NY, USA, 2025. Association for Computing Machinery. doi.org/10.1145/3706599.3719683.

[67] Katie Seaborn, Tatsuya Itagaki, Mizuki Watanabe, Yijia Wang, Ping Geng, Takao Fujii, Yuto Mandai, Miu Kojima, and Suzuka Yoshida. Deceptive, Disruptive, No Big Deal: Japanese People React to Simulated Dark Commercial Patterns. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, CHI EA '24, New York, NY, USA, 2024. Association for Computing Machinery. doi.org/10.1145/3613905.3651099.

[68] Minghao Shao, Abdul Basit, Ramesh Karri, and Muhammad Shafique. Survey of Different Large Language Model Architectures: Trends, Benchmarks, and Challenges. *Institute of Electrical and Electronics Engineers*, 12:188664–188706, 2024. doi.org/10.1109/ACCESS.2024.3482107.

[69] Zewei Shi, Ruoxi Sun, Jieshan Chen, Jiamou Sun, Minhui Xue, Yansong Gao, Feng Liu, and Xingliang Yuan. 50 Shades of Deceptive Patterns: A Unified Taxonomy, Multimodal Detection, and Security Implications. In *Proceedings of the ACM on Web Conference 2025*, WWW '25, page 978–989, New York, NY, USA, 2025. Association for Computing Machinery. doi.org/10.1145/3696410.3714593.

[70] Ray Sin, Ted Harris, Simon Nilsson, and Talia Beck. Dark patterns in online shopping: do they work and can nudges help mitigate impulse buying? *Behavioural Public Policy*, 9(1):61–87, 2025. doi.org/10.1017/bpp.2022.11.

[71] Than Htut Soe, Cristiana Teixeira Santos, and Marija Slavkovik. Automated detection of dark patterns in cookie banners: how to do it poorly and why it is hard to do it any other way. *arXiv preprint arXiv:2204.11836*, 2022. doi.org/10.48550/arXiv.2204.11836.

[72] Evangelia Spiliopoulou, Riccardo Fogliato, Hanna Burnsky, Tamer Soliman, Jie Ma, Graham Horwood, and Miguel Ballesteros. Play Favorites: A Statistical Method to Measure Self-Bias in LLM-as-a-Judge. *arXiv preprint arXiv:2508.06709*, 2025. doi.org/10.48550/arXiv.2508.06709.

[73] Annalisa Szymanski, Noah Ziems, Heather A. Eicher-Miller, Toby Jia-Jun Li, Meng Jiang, and Ronald A. Metoyer. Limitations of the LLM-as-a-Judge Approach for Evaluating LLM Outputs in Expert Knowledge Tasks. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, IUI '25, page 952–966, New York, NY, USA, 2025. Association for Computing Machinery. doi.org/10.1145/3708359.3712091.

[74] Varun Vasudevan, Faezeh Akhavizadegan, Abhinav Prakash, Yokila Arora, Jason Cho, Tanya Mendiratta, Sushant Kumar, and Kannan Achan. LLM-driven Constrained Copy Generation through Iterative Refinement. *arXiv preprint arXiv:2504.10391*, 2025. doi.org/10.48550/arXiv.2504.10391.

[75] R. Vedhapriyavadhana, Priyanshu Bharti, and Senthilnathan Chidambaranathan and. Detecting dark patterns in shopping websites – a multi-faceted approach using Bidirectional Encoder Representations From Transformers (BERT). *Enterprise Information Systems*, 0(0):2457961, 2025. doi.org/10.1080/17517575.2025.2457961.

[76] Ruiqi Wang, Jiyu Guo, Cuiyun Gao, Guodong Fan, Chun Yong Chong, and Xin Xia. Can LLMs Replace Human Evaluators? An Empirical Study of LLM-as-a-Judge in Software Engineering. *Proc. ACM Softw. Eng.*, 2(ISSTA), June 2025. doi.org/10.1145/3728963.

[77] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35 of *NeurIPS '22*, pages 24824–24837. Curran Associates, Inc., 2022.

[78] Wenda Xu, Daniel Deutsch, Mara Finkelstein, Juraj Juraska, Biao Zhang, Zhongtao Liu, William Yang Wang, Lei Li, and Markus Freitag. LLMRefine: Pinpointing and Refining Large Language Models via Fine-Grained Actionable Feedback. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, NAACL

'24, pages 1429–1445, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi.org/10.18653/v1/2024.findings-naacl.92.

[79] Jingzhou Ye, Yao Li, Wenting Zou, and Xueqiang Wang. From Awareness to Action: The Effects of Experiential Learning on Educating Users about Dark Patterns. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery. doi.org/10.1145/3706598.3713493.

[80] José P Zagal, Staffan Björk, and Chris Lewis. Dark Patterns in the Design of Games. In *Foundations of Digital Games 2013*, pages 39–46, Chania, Crete, Greece, 2013. Society for the Advancement of the Science of Digital Games.

[81] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36 of *NeurIPS '23*, pages 46595–46623. Curran Associates, Inc., 2023.

Index

Typeset September 30, 2025