

Automatically Detecting Deceptive Patterns on Websites

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

*by
Maximilian Hosch*

Thesis advisor:
Prof. Dr. Jan Borchers

Second examiner:
Prof. Dr. Ulrik Schroeder

Registration date: 25.02.2025
Submission date: 25.06.2025

Eidesstattliche Versicherung

Declaration of Academic Integrity

Name, Vorname/Last Name, First Name

Matrikelnummer (freiwillige Angabe)
Student ID Number (optional)

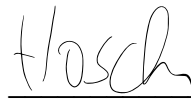
Ich versichere hiermit an Eides Statt, dass ich die vorliegende Arbeit/Bachelorarbeit/
Masterarbeit* mit dem Titel

I hereby declare under penalty of perjury that I have completed the present paper/bachelor's thesis/master's thesis* entitled

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.

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:



Ort, Datum/City, Date

Unterschrift/Signature

Contents

Abstract	ix
Überblick	xi
Acknowledgments	xiii
Conventions	xv
1 Introduction	1
2 Related Work	7
2.1 Growing Importance	7
2.2 Taxonomies and Ontologies	8
2.2.1 Groundwork	8
2.2.2 Ontology Used for This Work	9
2.3 Countermeasures	10
2.3.1 Elimination and Web Augmentation	10
2.4 Automated Detection	12

3	Methodology, Results and Evaluation	17
3.1	Methodology	17
3.1.1	Dataset Construction	17
3.1.2	Technical Constraints and System Limitations	20
3.1.3	Prompt Design	22
	Preprocessing to Mitigate Token and Interpretation Constraints	22
	Prompt Structuring Strategy	23
	Addressing Misinterpretations Through Ontology- Augmented Prompt Context	24
	General Prompt Tuning Procedure	24
	Prompt Engineering and Reasoning Strategy	25
3.1.4	Model Configuration	27
3.1.5	Metrics and Evaluation Setup	28
3.2	Results and Evaluation	32
3.2.1	Overall Results	32
3.2.2	Results by Target Label	33
3.2.3	Results by Dark Pattern	35
	Classification Rate	36
	F1 Score	36
	Recall	38
	Precision, Accuracy, Specificity, and General Observations . .	40
	Invalid Hallucination Rate	40

4	Discussion	41
4.1	Discussion	41
4.1.1	RQ1: Can prompt engineering improve GPT-4o's ability to detect dark patterns?	41
4.1.2	RQ2: What technical limitations arise when applying GPT-4o to the analysis of real-world websites, and how can they be overcome?	42
4.1.3	RQ3: How well does GPT-4o perform in classifying websites based on their use of dark patterns?	43
	Interesting Edge Cases	45
4.2	Limitations of Our Work	45
4.2.1	Dataset	45
4.2.2	Lack of Iterations	46
4.2.3	Prompt Wording	47
5	Summary and Future Work	49
5.1	Summary and Contributions	49
5.2	Future Work	50
A	Playwright Configurations	53
B	Notable Preliminary Test Results	55
B.1	Model Output with an Inaccurate Understanding of the Dark Pattern "Positive or Negative Framing"	55
B.2	Model Output with Classification Based on CSS Class Name	55
B.3	Model Output with Misunderstanding of the Classification Label	56

C	Final Prompts	57
C.1	System Prompt - Baseline	57
C.2	Input Prompt - Baseline	57
C.3	System Prompt - Engineered	58
C.4	Input Prompt - Engineered	58
D	Result Data	59
D.1	Inferred Label Distribution Broken Down into Target Label Components	59
D.2	Performance Metrics Graphs for Complete Set of Dark Patterns	60
	Bibliography	67
	Index	71

List of Figures and Tables

3.1	The number of instances per class label in the manually labeled dataset.	19
3.2	An example of a severe hallucination when testing with all 43 dark patterns in one prompt.	21
3.3	Comparison of weighted macro averages for each metric across prompting strategies, including micro averages for the <i>R-Guess</i> baseline	32
3.4	Inferred label distribution of the engineered prompt, broken down by target label components	33
3.5	Inferred label distribution of the baseline prompt, broken down by target label components	34
3.6	Target label distribution for the engineered prompt, broken down by inferred label components	34
3.7	Target label distribution for the baseline prompt, broken down by inferred label components	35
3.8	Classification rates by dark pattern, sorted by decreasing performance under the engineered prompt. Dark patterns with no actual positive instances (i.e., zero weight) are excluded from the analysis. .	37
3.9	F1 Scores by dark pattern, sorted by decreasing performance under the engineered prompt. Dark patterns with no actual positive instances (i.e., zero weight) are excluded from the analysis.	39

D.1	Inferred labels of the engineered prompt broken down into the target label components (in percent).	59
D.2	Inferred labels of the baseline prompt broken down into the target label components (in percent).	59
D.3	Classification rate by dark pattern, sorted by decreasing performance under the engineered prompt.	61
D.4	Accuracy by dark pattern, sorted by decreasing performance under the engineered prompt.	62
D.5	Precision by dark pattern, sorted by decreasing performance under the engineered prompt.	63
D.6	Recall by dark pattern, sorted by decreasing performance under the engineered prompt.	64
D.7	F1 Score by dark pattern, sorted by decreasing performance under the engineered prompt.	65
D.8	Specificity by dark pattern, sorted by decreasing performance under the engineered prompt.	66

Abstract

Dark Patterns are deceptive user interface designs that manipulate users into making unintended choices, often against their best interests. While several automated tools have been developed to detect such patterns, the task remains complex due to the subtle and context-dependent nature of these interfaces. This thesis explores a novel approach to evaluating the in-context learning capabilities of GPT-4o for the detection of Dark Patterns within the codebases of popular real-world websites. To support this investigation, we created a hand-labeled dataset for benchmarking. Our findings indicate that while the model tends to produce overly aggressive classifications, it demonstrates promising potential in input-sensitive detection. Furthermore, prompt engineering considerably enhanced detection performance. However, the limited context window of GPT-4o remains a primary constraint, restricting the model's ability to process larger code segments effectively.

Überblick

Dark Patterns sind irreführende Gestaltungsmuster in Benutzeroberflächen, die Nutzer dazu verleiten, unbeabsichtigte Entscheidungen zu treffen – oft entgegen ihrer eigenen Interessen. Obwohl bereits mehrere automatisierte Werkzeuge zur Erkennung solcher Muster entwickelt wurden, bleibt die Aufgabe aufgrund der subtilen und kontextabhängigen Natur dieser Designs äußerst komplex. Diese Arbeit untersucht einen neuartigen Ansatz zur Bewertung der In-Context-Lernfähigkeiten von GPT-4o bei der Erkennung von Dark Patterns in den Codebasen populärer, realer Webseiten. Zur Validierung wurde ein manuell annotierter Datensatz erstellt. Unsere Ergebnisse zeigen, dass das Modell zwar zu einer übermäßig aggressiven Klassifikation neigt, jedoch ein vielversprechendes Potenzial zur input-sensitiven Erkennung aufweist. Darüber hinaus führte gezieltes Prompt Engineering zu einer erheblichen Verbesserung der Erkennungsleistung. Die begrenzte Kontextgröße von GPT-4o stellt jedoch nach wie vor eine wesentliche technische Einschränkung dar, da sie die Verarbeitung größerer Codeabschnitte erschwert.

Acknowledgments

I would like to express my sincere gratitude to Prof. Dr. Jan Borchers and Prof. Dr.-Ing. Ulrik Schroeder for taking the time to examine my bachelor thesis. Their involvement and evaluation are greatly appreciated. I am especially thankful to my advisors, René Schäfer and Paul Preuschoff, for their continued support, insightful guidance, and invaluable advice throughout the process. I would also like to thank everyone who contributed with feedback, ideas, and encouragement during this time. Finally, I want to extend my deepest thanks to my friends and family — and especially to my girlfriend — for their unwavering emotional support, patience, and motivation throughout this journey.

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 described with the pronouns *they/their*.

Short excursuses are set off in colored boxes.

EXCURSUS:

Excursuses 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.

In the course of preparing this thesis, we used the large language model GPT-4o by OpenAI to support various aspects of the work. Specifically, the model was employed for data analysis, background research, assistance with content structuring, and feedback on academic writing style. All outputs were critically reviewed, edited, and integrated by us to ensure accuracy, originality, and alignment with academic standards.

Chapter 1

Introduction

Dark patterns are user interface (UI) design practices that are manipulative, working against the user's intent, obstructing the user to achieve their goal or are in other ways deemed problematic by the research community [Mathur et al., 2021]. As they are wide-spread [Mathur et al., 2019] on the world wide web and other digital platforms [Zagal et al., 2013], occur in a variety of designs [Gray et al., 2024] and have a negative impact on most users [Bongard-Blanchy et al., 2021], they are an acutely important research area.

Research follows the questions of what patterns exist (e.g. [Gray et al., 2024], [Gray et al., 2018]), how they impact the user (e.g. [Papenmeier et al., 2025], [Luguri and Strahilevitz, 2021], [Bongard-Blanchy et al., 2021]) and how to help the user minimize the negative impact (e.g. [Schäfer et al., 2025], [Schäfer et al., 2023]). Regarding the latter, many approaches with varying effectiveness exist [Koh and Seah, 2023]. Areas for countermeasures are e.g. educational, design, regulatory or technical. The classic and intuitive technical countermeasure approach is detecting dark patterns in a first step and countering the detected patterns in a second. However, recently Schäfer et al. [2025] proposed a novel approach using GPT-4o, a modern Large Language Model (LLM) with a good general knowledge [Shahriar et al., 2024], to skip the detection and jump straight to the defusing stage, in their case removal of the

Dark Patterns are manipulative UI designs. They are widely spread and an important research area.

Automatic Dark Pattern Detection is important to label data and support other countermeasures.

dark patterns. Clusmann et al. [2023] define LLMs as follows: "Large language models (LLMs) are artificial intelligence (AI) tools specifically trained to process and generate text."

LLMs can eliminate dark patterns without detection, but a lot of times users prefer highlighting with the option of removal.

The approach of Schäfer et al. [2025] involved piping frontend-code, in their case the part of a website that runs in the browser, into GPT-4o with the instructions to reduce the manipulateness. Their approach showed a lot of promise, however it has three key weaknesses. Firstly, it is not possible to counter all dark patterns by just removing them. For example, an instance of the Immortal account dark pattern, as defined in the ontology by Gray et al. [2024], could be that the backend account API simply does not offer an endpoint for deleting one's account, something no change in the frontend-code will overcome. Secondly, in a different study Schäfer et al. [2023] have shown that users a lot of times prefer other countermeasures to the entire removal of the dark pattern. Thirdly, the detection step has more merit than just being the first step of a removal process. Among others, detecting dark patterns helps creating datasets, can make removal more deterministic and could help law enforcement to prosecute illegal dark pattern uses more easily.

Current automatic detection approaches are inherently limited.

The detection of some dark patterns however can be a complex task as they can be manifold, implemented in different ways, be up to interpretation and sometimes very covert [Bongard-Blanchy et al., 2021]. There has been several approaches to automatically detect dark patterns. Some of them specialize on small parts of websites (e.g. Adorna et al. [2024], Soe et al. [2022]), others focus on only text-based dark patterns (e.g. Sazid et al. [2023], Yada et al. [2022]) and still others do it in a semi-automatic way (e.g. Mills and Whittle [2023], Mathur et al. [2019]). Due to these limitations, all these approaches have an inherently limited applicability.

We create an LLM based and code based detection strategy, with ontology definitions.

However, due to the complexity of the dark pattern detection task, there are no current solutions that can perform it in a fully automatic way, that can also adapt to the changes that user interfaces, especially in the world wide web, might undergo at any point in time. In recent years,

LLMs have advanced to the point where they can handle tasks traditionally reserved for experts, like coding (Shui et al. [2023] or law Hou and Ji [2024]). Earlier approaches to detect dark patterns in code have been dismissed with the argument that the models of the time lack the overarching understanding of the code and are thus unable to derive information about dark patterns from it in a meaningful way [Mills and Whittle, 2023]. However, since these capabilities have improved to the point where they can perform certain expert tasks, in this paper we will look into the capabilities of a modern LLM called GPT-4o by OpenAI¹ in the dark pattern detection process. We will create a novel approach of using GPT-4o to parse and classify real world websites according to their use of dark patterns. This encompasses providing the HTML and CSS of a website and the ontology by Gray et al. [2024] to the model. Then one simple and one carefully engineered prompt, an instruction to a LLM, will be compared through the use of different performance metrics. In particular we will focus on the following research questions:

***RQ:** How effective is GPT-4o in detecting known dark patterns in real-world websites?*

To systematically address this overarching question, the following subquestions are formulated:

1. **RQ1:** Can prompt engineering improve GPT-4o’s ability to detect dark patterns?
2. **RQ2:** What technical limitations arise when applying GPT-4o to the analysis of real-world websites, and how can they be overcome?
3. **RQ3:** How well does GPT-4o perform in classifying websites based on their use of dark patterns?

In order to answer these questions we conducted an empirical study using a hand-labeled dataset of popular websites.

¹ <https://platform.openai.com/docs/models/gpt-4o>, accessed on June 19, 2025

In the study GPT-4o classified the use of dark patterns in exactly these websites. The resulting inference labels were then compared with the aforementioned dataset.

Therefore, the core contributions of our work are

- **an empirical evaluation of GPT-4o for detecting dark patterns in real-world websites,**
- **the design and implementation of a hybrid prompt** that combines elements of chain-of-thought reasoning, heuristic prompting, and anticipatory logic to improve detection accuracy over a naive baseline,
- **the development of a retrieval-based architecture** using website parsing and vector storage to overcome context window limitations inherent to GPT-4o and
- **the creation of a labeled evaluation dataset** for dark pattern detection on high-traffic websites, enabling reproducibility and further comparative studies in this area.

The results seem promising, but the approach needs further polishing.

Our study showed that the task of dark pattern recognition in HTML and CSS remains complex, even for a modern LLM like GTP-4o. Despite achieving a modest F1 score of 44.63%, the model using the engineered prompt significantly outperformed its performance with the baseline prompt (33.73%), and notably exceeded the results of random guessing. With a relatively high recall it caught most cases of dark pattern usage. However, a low precision suggests that the engineered prompt was chosen too aggressively.

Further learnings from our preliminary tests are that the token size of most real world websites presents a technical challenge. Not only does the token size exceed the maximum context window size of GPT-4o, the model also lacks the ability to derive the look of a website from code alone as the token count increases.

The remainder of this thesis is structured as follows: Chapter 2 reviews related work, highlighting the growing importance of dark patterns, clarifying the terminology used

throughout this study, and discussing the relevance of countermeasures — particularly automated detection approaches. Chapter 3 presents the rationale behind the chosen study design, including LLM configurations, prompt strategies, and data preparation. Furthermore, it details the execution of the study using GPT-4o and reports the resulting findings. Chapter 4 assesses these results in depth, thereby answering our research questions, and evaluates the limitations of our approach. Finally, Chapter 5 summarizes the paper and identifies potential directions for future research.

Chapter 2

Related Work

2.1 Growing Importance

The research field of dark patterns is a field of growing importance. Especially since, for example, Luguri and Strahilevitz [2021] concluded in their paper that most dark patterns in use violate existing American laws, research into the automation of the detection process is a vital step towards reliably prosecuting wrong-doings in that area. Furthermore, an OECD Report on the topic from November 22nd¹ highlights the increasing use, the dangers and the pressing need to counteract the rise of dark patterns. Similarly, a 2022 FTC report confirmed that companies are “increasingly” using sophisticated dark patterns to trick consumers into purchases or giving up data², breaking American law in the process.

Dark patterns are gaining the attention of policy makers around the world

Furthermore, the OECD report highlights the increasing recognition that dark patterns have received in legislative and regulatory frameworks. One example is the Digital Services Act (DSA) of the European Union adopted in 2022 explicitly defining the term “dark patterns” and prohibit-

¹ [https://one.oecd.org/document/DSTI/CP\(2021\)12/FINAL/en/pdf](https://one.oecd.org/document/DSTI/CP(2021)12/FINAL/en/pdf), accessed on June 20, 2025

² <https://www.ftc.gov/news-events/news/press-releases/2022/09/ftc-report-shows-rise-sophisticated-dark-patterns-designed-trick-trap-consumers>, accessed on June 20, 2025

ing online platforms from deploying dark patterns that deceive or manipulate users. Similarly, the Digital Markets Act (DMA) forbids large “gatekeeper” platforms from presenting choices in a non-neutral, coercive manner.

2.2 Taxonomies and Ontologies

As we have now established the growing importance of dark pattern research and identified the real-world need to detect them, we are looking at taxonomies and ontologies as they play a big role in harmonizing the communication about a research field. Especially in the context of dark pattern detection, it is inherently important to have a terminology and definition, on which the detection mechanism can be based. Over the years multiple taxonomies with different scopes and focus areas have evolved in the field of dark patterns.

2.2.1 Groundwork

Dark patterns are malicious designs, the intention of the designer is no longer important.

Conti and Sobiesk [2010] were the first to attempt a formal definition and categorization of “malicious designs”, which will later be commonly referred to as dark patterns. Their definition still includes the notion that a key identifier for a dark pattern is a malicious intent. We will see that in more recent definitions the only decisive factor will be whether or not the design works against the intent of the user, as identifying the designers intent is an inherently difficult task. Apart from that, they laid the groundwork for future research by identifying the increasing aggressiveness of such designs and defining a first taxonomy comprising of eleven major dark pattern categories identified by analyzing thousands of websites.

Bösch et al. [2016] brought dark pattern taxonomy into the privacy domain with a systematic framework. They focused on privacy-related dark patterns, identifying seven types, of which many have lived on in later taxonomies and ontologies.

Gray et al. [2018] contributed the first peer-reviewed comprehensive taxonomy of dark patterns, which at the time was referenced by many others as a standard. They based it on the work of Harry Brignull, leveraging the examples in his "Hall of Shame". Building on that, they extended it with examples collected from other individuals familiar with the subject, like journalists and website-owners. By looking at them from two perspectives - one of a computer scientist and one of a user experience expert - they were able to abstract and group concepts together and formalize the work in progress taxonomy of Brignull and his community.

The ontology is based on multiple taxonomies and informal dark pattern collections.

2.2.2 Ontology Used for This Work

Gray et al. [2024] elaborated on previous taxonomies to create a comprehensive ontology, which can be expanded as new patterns are identified. Our work bases heavily on the definitions and terminology from this paper. Hence, we will give a short overview over the most important aspects of the ontology.

The ontology is structured into 3 levels: high, meso and low.

- **High level patterns** are less relevant for our work as they define high level concepts. The authors describe them as relevant towards policies and legislation.
- **Low level patterns** are actual ways to implement these strategies and are therefore relevant for detection scenarios. They tend to be specific to the context they are used in and are therefore less abstract.
- **Meso level patterns** are in between the two former levels. However, not every meso level category necessarily needs to be split up further into low level categories. Generally speaking, they describe the common angle of attack and are content-agnostic.

It consists of five high-level patterns, 25 meso-level patterns, of which nine have no further subcategories, and 34

Our study will focus on the meso-level patterns without subpatterns and low-level patterns.

low-level patterns. Each of these patterns is formally defined and the definitions of lower levels build up on the definitions of the higher level. For our work, the nine meso-level patterns without further subcategories like "Choice Overload" or "Personalization", the 34 low-level patterns like "Disguised Ad" or "Endorsements and Testimonials" and all of their definitions are most relevant as they are most suitable for detection.

2.3 Countermeasures

Awareness for dark patterns is often not enough to counteract them.

With the dark pattern terminology taken care of and the growing importance established, one might think that dark patterns can easily be countered by raising awareness in the general public. However, Bongard-Blanchy et al. [2021] conducted a user study measuring user awareness of dark patterns and whether or not they are able to resist their manipulation. They found out that the awareness for dark patterns varies highly on the type, and that simple awareness does not effectively counteract the malicious influence of them. Based on their findings they went on to identify a four by four matrix filled with eight already identified intervention spaces to counteract the malicious influence of dark patterns. One important intervention space in the matrix is the "technical dark pattern elimination"-space of plug-ins and add-on extensions. Here the authors argue that one effective way to counteract the influence of dark patterns can be the automatic removal of dark patterns on the user end.

2.3.1 Elimination and Web Augmentation

Silent elimination is not perceived well. Users prefer to choose the countermeasure.

While this seems like a reasonable approach, countering the impact of dark patterns by automatic elimination on the user end is a double-edged sword. Schäfer et al. [2023] conducted a study on different browser based countermeasures. They investigated among other things how users perceived a browser extension that automatically and silently removes dark patterns on an artificially created set

of web elements, so that the user is never faced with them. Users felt uncomfortable about this way of counteraction as they felt a lack of control about their browsing experience. However, the results for highlighting the dark patterns and being able to remove them with the push of a button were perceived positively. Another finding they presented is that users prefer different counter-strategies for different dark patterns, hence there is no one-size-fits-all countermeasure.

Lu et al. [2024] conducted a study with users of a browser extension they provided, that gives users the possibility to highlight dark patterns found in a website with the click of a button. The users were then able to choose from a set of prepared enhancement options for this dark pattern. Their extension was limited to five popular websites and was manually prepared to work with these. Their key insights are, that users appreciate the easy access to knowledge about dark patterns they are currently facing. Furthermore, they do feel empowered by the choice of the enhancement options. And like in the study by Schäfer et al. [2023] study participants preferred different countermeasures for different dark patterns.

Regarding countermeasures no one-size-fits-all countermeasure exists.

Both of these studies, along with many others (e.g., [Adorna et al., 2024]), share a key characteristic: they all rely on a detection step. However, as mentioned in the introduction Schäfer et al. [2025] conducted a study on removing dark patterns with an LLM without detecting them at all. For that purpose they handcrafted a dataset of websites and website elements both with and without dark patterns. They then used these combined with a prompt instructing GPT-4o to make the webpage less manipulative. Over multiple iterations GPT-4o attempted to decrease the manipulateness and they ended up with reliably better webpages, than before the GPT-4o iterations. These promising results suggest that automatic detection is not necessary in a lot of cases to remove dark patterns. However, as the studies by Lu et al. [2024] and Schäfer et al. [2023] suggest removal is not always the best way to counter a dark pattern. Furthermore, this way of dealing with dark patterns does not help to acquire new knowledge about them, as automatic detection arguably could. So, while their approach has a lot of merit it does by no

Even though LLMs can defuse dark patterns without prior detection, it is necessary in order return the power to the user.

means invalidate the efforts towards automatically detecting dark patterns on a large scale.

2.4 Automated Detection

In addition to dark pattern elimination, Bongard-Blanchy et al. [2021] identify automated detection tools as a second intervention space within the dimension of technical countermeasures. Although not a direct countermeasure themselves, the authors argue that such tools serve an essential role by providing evidence for consumer advocates, thereby supporting the enforcement of legislative frameworks such as the Digital Markets Act, the Digital Services Act, and the GDPR. Automated detection thus becomes a critical component in the broader landscape of countermeasures, enabling regulatory authorities to act upon legal provisions. Furthermore, the findings of Schäfer et al. [2023] and Lu et al. [2024] reinforce the importance of automated detection tools across other intervention spaces. A considerable body of research has already been devoted to developing reliable and scalable methods for dark pattern detection.

Automatic code based
detection in cookie
banners is possible with
a GNN.

Hausner and Gertz [2021] proposed the method of training a graph neural network (GNN), a machine learning method specialized on graph structures like the document object model (DOM) of a website, to detect dark patterns in the HTML code of a website. Their results have shown a reliable detection rate across a wide range of websites albeit with some key limitations. They limited their approach to the detection of dark patterns in cookie banners and especially in the uneven design between "accept" and "reject" buttons making their approach hard to generalize. Additionally, they used already existing implementations of dark patterns to train their GNN. This makes it likely that new ways of implementing the exact same or slightly adjusted dark pattern will go undetected, as the machine learning algorithm shows no understanding of the concept behind the dark pattern but is only able to recognize code patterns.

Soe et al. [2022] adopted a similar approach, focusing on the automated detection of dark patterns in cookie banners using a newly trained machine learning model. While their results were promising, their method relies on manually labeled input data—a limitation they themselves highlight, noting that if human effort is already required for labeling, it could arguably be used directly to detect dark patterns instead. The authors further acknowledge that dark pattern recognition remains a challenging task for artificial intelligence, outlining several obstacles they consider difficult to overcome using their method or any traditional machine learning-based approach. However, from a contemporary standpoint, many of these challenges now seem addressable through the use of Large Language Models (LLMs), particularly when combined with the ontology proposed by Gray et al. [2024].

LLMs solve challenges that previous research deemed hard to solve.

Yada et al. [2022] conducted a study on the automatic detection of dark patterns in website text with the four LLMs BERT, RoBERTa, ALBERT and XLNet. They used a combination of a dataset created by Mathur et al. [2019] and balanced it with non manipulative text instances. They achieved their highest accuracy of 0.975 with RoBERTa. Despite their outstanding results the method they applied is only applicable for dark patterns that are implemented on the basis of text. As most dark patterns in the ontology of Gray et al. [2024] are possibly implemented without using text based cues, focusing only on them is a severe limitation. Additionally, the LLMs that were used in their study are outdated, in technological aspects as well as in their training data. Finally, their dataset contained just e-commerce websites. While this limitation was purposefully chosen, it leaves the question open on how generalizable their approach is beyond e-commerce websites.

Text-based detection with LLMs has already been shown to be effective.

Sazid et al. [2023] built upon the approach by Yada et al. [2022] by leveraging their dataset and also focusing on text-based dark pattern detection, this time utilizing the Large Language Model GPT-3. Employing an in-context learning strategy, they incorporated hand-crafted definitions and labeled examples into the prompt to enhance detection performance compared to the baseline established by Yada et al. [2022]. In the original study, Sazid et al. [2023]

GPT-3 is able to better generalize text-based detection through in-context learning.

achieved a maximum accuracy of 92.57% on their dataset. In a follow-up study, Sazid et al. [2023] evaluated the generalizability of both approaches using a new dataset comprising 30 Bangladeshi e-commerce websites. On this dataset, detection accuracy declined overall; however, the GPT-3-based approach achieved a higher accuracy (58.67%) than the RoBERTa-based method used by Yada et al. [2022] (42.8%). This result is not unexpected, as GPT-3 was not fine-tuned on a specific dataset and thus offers better generalization capabilities. Nonetheless, many of the limitations identified in the earlier study persist. Most notably, the exclusive focus on textual input severely limits the model's ability to detect dark patterns implemented through non-textual means. For instance, the "sneaking" pattern was consistently misclassified, as its deceptive nature is often embedded in visual or structural elements rather than in text. Additionally, the emphasis on e-commerce websites remained unchanged. Although in-context learning is, in theory, a generalizable technique, the applicability and performance of the model across other types of websites may vary considerably.

Earlier revisions of GPT
had difficulties
interpreting code.

Mills and Whittle [2023] propose three distinct approaches for leveraging LLMs to simulate user behavior, with the objective of inferring the presence of dark patterns based on the model's responses. The first approach, titled "Choose Your Own Adventure", serves as a baseline method. In this setup, the LLM is provided with a persona description and a manually crafted textual representation of the options available to a user visiting a website. The model is then prompted to make a decision based on the preferences and goals of the persona. The second approach, "AI Vision", builds on the first but replaces the manual text description of the website with an actual screenshot. This shift allows the LLM (in this case, GPT-4) to extract information visually rather than relying on human-authored summaries. The third approach, "Decision Network", was not fully implemented due to technical limitations. Its aim was to use the HTML code of a website as input instead of a screenshot, coupled with a web crawler under LLM control to automate navigation across pages. This would eliminate the need for manual transitions between website states. However, the authors observed that GPT-

3.5 struggled with generating meaningful visual interpretations from raw HTML code, which hindered progress in this direction. Preliminary results indicated that the "AI Vision" approach yielded the most promising outcomes when implemented with GPT-4. In contrast, the "Choose Your Own Adventure" method was heavily reliant on prompt engineering, making it labor-intensive and less scalable. Although all three approaches offer valuable insights, the first two are not fully automatic and focus primarily on simulating user behavior rather than directly detecting dark patterns. The third approach, while still underdeveloped, presents potential for future enhancement and integration into a fully automated detection pipeline. However, as the authors note, a key challenge lies in rendering webpage structure and content from code in a form that LLMs can effectively interpret. This suggests that a more viable path may be to explore direct detection of dark patterns from HTML code — without relying on user simulation — as a means of achieving automation and scalability.

Chapter 3

Methodology, Results and Evaluation

3.1 Methodology

While previous research has explored dark pattern detection through machine learning or rule-based systems, many existing approaches are limited to text excerpts or simplified website sections as input. In contrast, this study investigates whether a state-of-the-art LLM — GPT-4o — can identify dark patterns directly from complete HTML and CSS source files of real, publicly available websites. To structure and validate this process, an officially published dark pattern ontology is used, including precise definitions to guide classification. The methodology outlined in this chapter details the dataset creation process, prompt design, model configuration, and the evaluation strategy used to assess the classification capabilities of the LLM without additional fine-tuning.

3.1.1 Dataset Construction

To evaluate the classification capabilities of GPT-4o in detecting dark patterns, a custom dataset was created based on the top-ranking websites on the internet. The start-

We created a hand-labeled dataset of eleven websites as ground truth for our study.

ing point was the Tranco list by Le Pochat et al. [2019], a research-focused top site ranking that merges multiple top rankings, from the 29th of May 2025¹. Domains were assessed sequentially from the top of the list, filtering out those whose primary purpose was not to serve a publicly accessible website — such as infrastructure, CDNs, or service endpoints. From the remaining domains, the top eleven websites were selected for inclusion in the study.

The dark patterns most suited for detection were identified.

Each website was accessed and analyzed in its live state as of June 3, 2025, with the full HTML and CSS source code captured for evaluation. The study relies on the ontology of Gray et al. [2024]. We selected the nine meso-level patterns without sub patterns and the 34 low-level patterns for labeling as the authors mention that these are most suited for detection purposes. These 43 dark patterns, which exist with precise definitions, served as the basis for structured classification, ensuring consistency across pattern types. To account for the inherent ambiguity and subjectivity involved in identifying dark patterns—particularly those with interpretive boundaries such as Complex Language or Personalization — a four-point Likert-style labeling scheme was introduced. The label definitions were as follows:

- DEFINITELY_NOT: No indication of the dark pattern or dark pattern irrelevant to this website.
- PROBABLY_NOT: Slight smells but not enough to be considered a dark pattern.
- PROBABLY: Strong smells suggesting the use of the dark pattern.
- DEFINITELY: Conclusive proof that the dark pattern is used.

Importantly, no neutral middle category was included in order to encourage the LLM to express at least a minimal tendency in classification. This choice reflects the interpretative complexity of the task and avoids indecisiveness, which could hinder meaningful analysis.

¹ Available at <https://tranco-list.eu/list/KW4KW>.

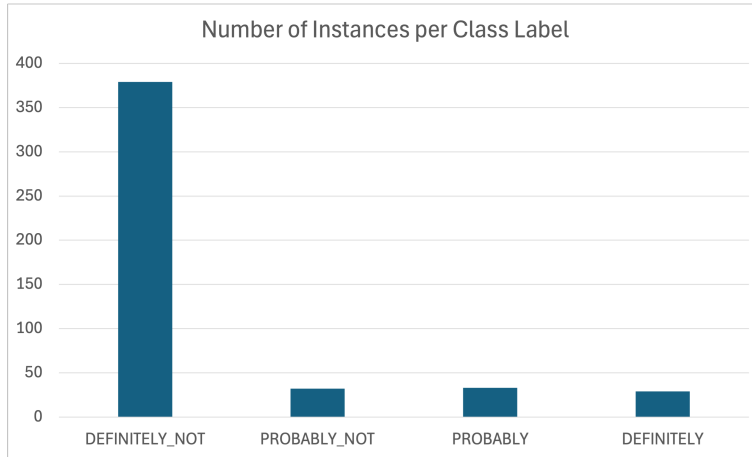


Figure 3.1: The number of instances per class label in the manually labeled dataset.

Each of the 11 websites was manually evaluated for the presence of each of the 43 dark patterns, resulting in a total of 473 potential classifications. The final distribution of labels, displayed in Figure 3.1 was highly imbalanced:

- 379 (~ 80.13%) were labeled as DEFINITELY_NOT,
- 32 (~ 6.77%) as PROBABLY_NOT,
- 33 (~ 6.98%) as PROBABLY, and
- 29 (~ 6.13%) as DEFINITELY.

Out of the 43 dark patterns, only 24 were observed with at least one occurrence (i.e., labeled PROBABLY or DEFINITELY). The remaining 19 had no confirmed presence in the dataset. The most commonly detected dark pattern was False Hierarchy, with 6 occurrences across the evaluated sites.

This dataset serves as the input for the LLM-based classification experiment described in the following sections.

3.1.2 Technical Constraints and System Limitations

The HTML and CSS combined combined are too large for GPT-4o's context.

The file_search tool is the most promising option of conducting this study successfully.

The file_search tool introduces new technical limitations.

While the use of a state-of-the-art large language model such as GPT-4o enables rich natural language understanding and reasoning, the practical application of such models to full-scale website analysis introduces several technical limitations. These constraints influenced core methodological decisions, particularly regarding data access, prompt design, and model interaction. The most fundamental constraint arises from the context window of GPT-4o, which is officially limited to 128,000 tokens, of which a maximum of 16,384 tokens may be generated as output.² However, the nature of this study — relying on complete HTML and CSS documents of high-traffic websites — posed an immediate challenge: many HTML files alone exceeded 200,000 tokens, with injected CSS further multiplying the size. This rendered direct prompting with full page content impossible, as the input would far exceed the model's context capacity. To overcome this bottleneck, the study utilized the file_search tool provided by OpenAI.³ This tool enables semantic and keyword-based retrieval from a vector store of previously uploaded documents, including HTML and CSS file types. By offloading the website content into this vector store, the model could query relevant information on demand, bypassing the need to embed the entire website code within the prompt.

Despite the promise of this architecture, several limitations became apparent during implementation. Although OpenAI officially documents an upper limit of 512MB or approximately 5 million tokens per file⁴, empirical testing revealed a significantly lower effective threshold under the Azure deployment used in this study — closer to 2 million tokens. Attempts to exceed this limit consistently led to indexing failures. This constraint played a critical role in shaping the formatting and segmentation strategy of code files, as discussed in the subsequent section on prompting.

² <https://platform.openai.com/docs/models/gpt-4o>, accessed on June 23, 2025

³ <https://platform.openai.com/docs/guides/tools-file-search>, accessed on June 23, 2025

⁴ <https://platform.openai.com/docs/guides/retrieval>, accessed on June 23, 2025

```
10 ### 2. **Navigation Bar**
11 Immediately noticeable is the navigation bar:
12 - **Main Branding/Logo**: Located at the top left or center of the page.
13   It usually links back to the homepage. You can imagine it showing "Amazon.
14   de" or a regional variation as text with a logo.
15 - **Search Box** is next likely you'll endpoint product sorting-needed buy
    find teas ux alignment comp layout..
16 ->callproduct searchingclar managementy refined!>
```

Figure 3.2: An example of a severe hallucination when testing with all 43 dark patterns in one prompt.

In addition to storage capacity, response quality degradation was observed when querying HTML files that contained substantial numbers of `<script>` tags. These tags, which often include minified JavaScript or inline tracking logic, appeared to interfere with the model’s ability to extract a coherent structural representation of the page. Although the model was technically capable of accessing these files via `file_search`, the resulting responses were frequently vague, irrelevant, or hallucinated. While this issue is addressed more fully in the prompt design section, it is noted here as a direct consequence of GPT-4o’s difficulty in handling large, noisy input even within file-based semantic retrieval workflows.

GPT-4o has difficulties interpreting inline script-tags.

A final technical limitation pertains to hallucination behavior at high context utilization. Initial tests involving prompts that made extensive use of the context window — exceeding 100,000 input tokens — revealed several reliability issues. The model frequently produced overly verbose or unfocused responses. Most notably, these long-context prompts occasionally led to output degeneration, where responses became incoherent, syntactically unstable, or devolved into meaningless sequences of words and symbols, as displayed in Figure 3.2. This aligns with broader observations in LLM research indicating that model reliability tends to decrease as context utilization approaches its upper limit (e.g. [Han et al., 2024]). This behavior strongly influenced the decision to restrict prompt length and reduce multi-pattern classification tasks to single-pattern prompts, as discussed in the following section.

LLMs tend to hallucinate when most of their context is used.

In sum, these constraints illustrate the practical boundaries of current LLM tooling when applied to real-world datasets

such as complete website code. In particular, the challenges of managing input size, avoiding hallucinations, and ensuring focused responses required a number of targeted adaptations to the model interaction process. These adaptations are reflected in the prompt design strategy, which is described in detail in the following section.

3.1.3 Prompt Design

The design of the prompts used in this study played a central role in ensuring that GPT-4o consistently produced meaningful classification outputs — regardless of their correctness — when applied to complex website code. Given the model’s inherent limitations, discussed previously, the primary objectives of prompt design were to reduce task complexity, control token usage, and produce consistent, structured outputs aligned with the labeling scheme defined in the dataset. The prompt design process was iterative in nature and was continuously refined based on the model’s observed behavior during preliminary testing.

Preprocessing to Mitigate Token and Interpretation Constraints

To address the previously elaborated technical constraints related to large HTML files and the interpretability issues caused by embedded scripts, two targeted preprocessing strategies were applied to reduce token overhead while maintaining the semantic completeness of the website data.

CSS files were downloaded separately and uniquely renamed.

All the inline script-tags were removed to improve the codes readability.

The first strategy involved the externalization of CSS content. Rather than embedding styles directly within the HTML — an approach that substantially inflated file size — each website’s external CSS files were downloaded separately and renamed using unique UUIDs. The corresponding `<link>` tags in the HTML were updated to reference these new filenames. This allowed the model to retrieve style information contextually via `file_search`, while reducing the number of tokens required per file and preserving correct referencing semantics. Secondly, all in-

line `<script>` tags were systematically removed from the HTML documents before upload. As outlined in the system limitations, files with extensive script content were prone to producing vague, irrelevant, or hallucinated outputs. Their removal improved the clarity of structural interpretations and reduced semantic noise during model processing. The downloading and preprocessing of website code was performed using Playwright⁵, a Python-based web automation and crawling framework. The precise crawler configuration and preprocessing parameters are documented in Appendix A.

Prompt Structuring Strategy

The degradation of output quality due to excessive token usage — resulting in frequent output corruption — was mitigated with a revised one-pattern-per-prompt strategy. Under this design, each prompt targeted a single dark pattern for a single website, thereby simplifying the task and improving the overall interpretability and consistency of the model's responses. To further enhance response uniformity, an anticipatory prompting strategy was adopted. A fixed output schema was embedded in the system prompt, instructing the model to return its response in JSON format. The schema required two fields: a "used" field containing one of the four classification labels defined in the dataset section, and a "reasoning" field with a brief justification for the selected label:

In order to save tokens and simplify the task, only one pattern was analyzed per prompt.

```
{  
  "used": "<classification label>",  
  "reasoning": "<explanation for label>"  
}
```

This structured approach reduced ambiguity in the model's outputs and facilitated a more consistent and straightforward evaluation process.

⁵ <https://playwright.dev/python/docs/library>, accessed on June 23, 2025

Addressing Misinterpretations Through Ontology-Augmented Prompt Context

Having clear definitions in the context improved the understanding.

Initial tests showed that GPT-4o frequently misinterpreted or overgeneralized the concept of dark patterns, particularly for less intuitive categories. To mitigate this, the original ontology paper by Gray et al. [2024] was uploaded to the `file_search` vector store, enabling the model to semantically retrieve authoritative contextual definitions for each pattern during inference. Incorporating the ontology into the retrieval context effectively eliminated most instances where the model's justifications in the "reasoning" field were based on incorrect assumptions about a pattern's defining characteristics. This was interpreted as an indicator of improved conceptual alignment and a deeper contextual understanding of the dark patterns under evaluation. A full example can be found in Appendix B.1.

General Prompt Tuning Procedure

Prompting was improved using an iterative variable-controlled testing approach.

The refinement of the prompts was conducted through an empirical, variable-controlled testing process. In each round of testing, a single variable was adjusted — such as removing `<script>` tags, changing prompt phrasing, or uploading auxiliary documents like the ontology. The resulting model outputs were then compared against prior results, with decisions made intuitively based on response quality, consistency, and interpretability.

This approach was not exhaustive, but it provided a practical, flexible framework for identifying and addressing failure modes in the prompting process. It also contributed to developing a prompt structure that was both robust and adaptable to the constraints of the model and the complexity of the classification task.

Prompt Engineering and Reasoning Strategy

Following the development of a technically functional baseline prompt, further work focused on designing a more structured and cognitively aligned version to explore whether prompt engineering could improve the quality of dark pattern classification. The baseline prompt incorporated necessary elements to avoid technical limitations — such as reduced input size, single-pattern focus, and a JSON output format — but otherwise contained minimal task guidance or reasoning scaffolding.

The baseline prompt is limited to technically necessary instructions

To investigate the potential of a more advanced prompting strategy, established techniques from current literature were examined. Two approaches were identified as particularly promising, due to their good performance in other specialist domains: Chain of Thought Prompting and Heuristic Prompting (e.g. [Sivarajkumar et al., 2024]).

Chain of Thought prompting and Heuristic prompting will be used in the engineered prompt.

- **Chain of Thought Prompting encourages models to reason step by step**, which aligns well with tasks that involve ambiguity or require contextual judgment, such as identifying dark patterns.
- **Heuristic Prompting provides the model with a role or objective**, guiding it to simulate human decision-making through the use of practical reasoning strategies.

These principles were applied in the development of an engineered prompt, which differed from the baseline in several key ways:

The persona of a specialist for the digital crimes unit, a more focused inspection instruction, a request for explicit reasoning and class definitions were added to the engineered prompt.

- **Role-based framing:** The model was assigned the persona of a specialist working for a Digital Crimes Unit, tasked with investigating deceptive design patterns on websites. The intention behind this approach was to encourage rigid analysis focused on the goal of classification.
- **Focused inspection instruction:** The prompt was extended with an explicit directive to identify specific

HTML elements that could serve as indicators of the targeted dark pattern. This modification was motivated by observations from preliminary tests with the baseline prompt, where the model often relied on superficial features — such as CSS class names or attributes like `aria-hidden` — rather than analyzing semantically meaningful structural components of the HTML. Such behavior frequently led to misclassifications. A full example can be found in Appendix B.2

- **Explicit reasoning request:** The prompt instructed the model to explain its reasoning in a step-by-step narrative prior to providing the classification output, aligning with the chain-of-thought approach introduced by Kojima et al. [2022].
- **Class definitions:** The prompt was extended to include brief definitions of the four classification labels. This addition was motivated by the observation that the model occasionally applied class labels in ways that diverged from the intended interpretation used during manual dataset annotation. For example, the `PROBABLY_NOT` label was often assigned in cases where the model found no direct evidence of a dark pattern but speculated that further pages might be required for a definitive judgment. However, according to the labeling guidelines, such cases should be classified as `DEFINITELY_NOT`, which is intended for instances where the dark pattern is clearly not present or not applicable. By clarifying the semantics of each class within the prompt, this intervention aimed to align the model’s label selection with the intended classification logic. The full example can be found in Appendix B.3.

The purpose of this improved prompt design was not simply to reformulate the instructions, but to embed cognitive structure into the model’s response behavior. By explicitly modeling the interpretive process, the prompt aimed to support more robust and explainable classifications.

Our study will run the engineered prompt against the baseline.

This engineered prompt was evaluated against the baseline prompt to determine whether the added structure and reasoning scaffolding improved performance. Both prompts

can be found in full in Appendix C. The evaluation process and findings are discussed in detail in the subsequent results section.

3.1.4 Model Configuration

Having established the prompting strategy, this section outlines the rationale behind the choice of LLM and details the configuration parameters used throughout the study. This research utilizes GPT-4o, a recent iteration of OpenAI's GPT-4 architecture, in conjunction with the `file_search` tool. GPT-4o was selected over alternatives such as Claude⁶ and Llama⁷ due to its strong reasoning capabilities [Liu et al., 2023], leading performance in code understanding and generation [Hou and Ji, 2024], comprehensive documentation⁸, and native integration with Azure⁹, which aligned well with the technical infrastructure of this study.

GPT-4o is used because it is well suited to the study's requirements in terms of capabilities, documentation, and technical framework.

OpenAI's ResponseAPI allows for several configurable parameters that influence the model's behavior, context management, and output. The key parameters relevant to GPT-4o in this study are:

Out of the most common parameters only input, instructions and max_output_tokens were edited.

- **input:** The user-facing prompt containing task-specific instructions.
- **instructions:** The system prompt that sets the model's behavioral constraints and expectations.
- **temperature:** A float (range 0-2) that controls output randomness; lower values produce more deterministic, factual responses, while higher values lead to more diverse and creative outputs.
- **top_p:** A nucleus sampling parameter (range 0-1) that limits the model to choosing from the top portion of the probability distribution. Typically used as an alternative to temperature, but not in parallel.

⁶ <https://claude.ai/login>, accessed on June 23, 2025

⁷ <https://www.llama.com>, accessed on June 23, 2025

⁸ <https://platform.openai.com/docs/overview>, accessed on June 23, 2025

⁹ <https://azure.microsoft.com/en-us>, accessed on June 23, 2025

- **max_output_tokens:** The maximum number of tokens allocated for generating output. For GPT-4o, this is capped at 16,384 tokens.
- **tools:** An array of auxiliary tools the model can access during inference. In this study, the only active tool was `file_search`.

The default value for temperature provides a healthy trade-off.

Preliminary tests revealed that setting the temperature too high led to vague and unfocused responses, while overly low values produced deterministic but overly conservative outputs — often failing to classify any instance as a dark pattern, likely due to the model’s inability to interpret the HTML context as a rendered website. As a compromise, the temperature was kept at the default value of 1.0, which balanced diversity and interpretability.

The `file_search` tool was used in ‘auto’ mode, for minimal limitations.

The `max_output_tokens` parameter was increased to 16,000 — slightly below the technical maximum — to provide sufficient space for the model to develop coherent reasoning chains while reducing the risk of output corruption due to context overflow. The `file_search` tool was configured to access the dedicated vector store containing the preprocessed codebase of each website. No restrictions were imposed on the number of tool calls, in order to allow the model unrestricted access to relevant contextual information as needed during inference.

3.1.5 Metrics and Evaluation Setup

In this final step of the methodology, we outline the evaluation strategy and the metrics used to assess model performance. The evaluation is based on the four predefined classification labels, from which we compute several metrics to gauge the model’s ability to distinguish subtle differences between classes and to simulate human-like analysis of website code — assuming access to the pattern definitions, akin to a well-informed user browsing a website.

Classification rate is calculated, aggregated for all labels, and for each dark pattern individually.

The primary metric is the *overall correct classification rate*, which reflects the proportion of predictions that match the

ground truth labels. In addition, we compute correct classification rates disaggregated by both classification label and individual dark pattern type, enabling a more fine-grained analysis. To further assess model behavior, the four classification labels are grouped into two broader "tendency" buckets:

- **Positive tendency:** DEFINITELY, PROBABLY
- **Negative tendency:** DEFINITELY_NOT, PROBABLY_NOT

Using these binary groupings, a confusion matrix is constructed, and standard classification metrics are calculated [Ghanem et al., 2023]:

A confusion matrix evaluation with the common metrics will be used.

- **Accuracy:** The proportion of all predictions — both positive and negative — that were correctly classified by the model.
- **Precision:** The proportion of positive predictions that were actually correct; this measures how many of the detected dark patterns were truly present.
- **Recall:** The proportion of actual positive instances that were correctly identified by the model; this indicates how well the model detects true dark patterns.
- **F1 Score:** The harmonic mean of precision and recall, providing a balanced measure that accounts for both false positives and false negatives. This metric is especially relevant for imbalanced datasets as the one used in this study [Ghanem et al., 2023].
- **Specificity:** The proportion of actual negative instances that were correctly identified as such; this measures the model's ability to avoid false positives.

These metrics are computed individually for each dark pattern and subsequently aggregated into a weighted macro average, where the weights correspond to the number of actual positive instances. This weighting addresses the significant class imbalance present in the dataset.

Invalid Hallucination
Rate is the rate of
outputs that do not
adhere to the output
format.

Finally, we introduce an additional metric termed the *invalid hallucination rate*, defined as the percentage of model outputs that fail to conform to the expected classification labels, adhere to the required JSON structure, or degrade into incoherent or meaningless content. This metric serves as a sanity check for the reliability of the evaluation pipeline, ensuring that only structurally valid and interpretable outputs are included in the quantitative analysis. All instances flagged under this metric are excluded from the computation of the remaining performance metrics.

The confidence matrix
metrics do not work well
with no actual positives.

In general, our metrical evaluation focuses on dark patterns with at least one actual positive instance, as most standard classification metrics do not behave meaningfully in the complete absence of positives. Including such cases would disproportionately highlight edge-case handling rather than general classification performance. Nevertheless, we supplement the metric-based analysis with targeted qualitative insights into noteworthy mismatches — such as instances where the model infers a DEFINITELY label while the human-annotated ground truth is DEFINITELY_NOT — to better understand model failure modes and edge behavior.

As a sanity baseline, a
uniform random
guessing approach is
defined.

To contextualize the model’s performance, we compare it to a *uniform random guessing* (R-Guess) baseline, defined as follows:

- $P(\text{predict positive}) = 0.5$
- $P(\text{predict negative}) = 0.5$

As established in the *Dataset Construction* the total number of labels is $N = 473$. The actual counts per class are:

$$N_{\text{pos}} = 62, \quad N_{\text{neg}} = 411$$

Each true class is randomly guessed with 50% probability as either positive or negative. Thus, the expected confusion matrix entries are:

$$\begin{aligned}
\text{True Positive (TP)} &= 0.5 \cdot N_{\text{pos}} = 31 \\
\text{False Negative (FN)} &= 0.5 \cdot N_{\text{pos}} = 31 \\
\text{False Positive (FP)} &= 0.5 \cdot N_{\text{neg}} = 205.5 \\
\text{True Negative (TN)} &= 0.5 \cdot N_{\text{neg}} = 205.5
\end{aligned}$$

Confusion Matrix

	Positive	Negative
True	31	205.5
False	205.5	31

Metrics for Positive Class (DEFINITELY + PROBABLY)

$$\text{Accuracy} = \frac{TP + TN}{N} = \frac{31 + 205.5}{473} = 0.5$$

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{31}{31 + 205.5} \approx 0.1311$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{31}{31 + 31} = 0.5$$

$$F_1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \cdot 0.13 \cdot 0.5}{0.13 + 0.5} \approx 0.206$$

$$\text{Specificity} = \frac{TN}{TN + FP} = \frac{205.5}{205.5 + 205.5} = 0.5$$

It is worth noting that these values, based on micro-level aggregation, tend to offer an overly optimistic performance estimate. They serve primarily as a baseline to demonstrate that any meaningful model should exceed this level of random performance.

3.2 Results and Evaluation

3.2.1 Overall Results

The engineered prompt outperforms the baseline and R-Guess in classification rate and F1 Score.

As shown in Table 3.3, the highest correct classification rate — albeit by a small margin — is achieved by the engineered prompt. Notably, the engineered prompt also yields the highest F1 Score, which is arguably the most important metric in this context due to its balanced consideration of both precision and recall. It substantially outperforms the baseline prompt, which in turn significantly exceeds the performance of the *R-Guess* (uniform random guessing) baseline.

The engineered prompt has a strong recall.

A particularly strong result is the engineered prompt’s recall of 79.03, indicating that it successfully captures the vast majority of actual positives. However, this comes at the expense of specificity and accuracy. Both of these metrics follow an inverse trend, with *R-Guess* performing best, followed by the baseline prompt, and the engineered prompt performing lowest.

Method	C-Rate	Acc.	Prec.	Rec.	F1	Spec.
Engineered	27.17	45.70	32.86	79.03	44.63	24.25
Baseline	21.44	48.93	26.70	53.23	33.74	44.03
R-Guess	25.00	50.00	13.11	50.00	20.60	50.00

Table 3.3: Comparison of weighted macro averages for each metric across prompting strategies, including micro averages for the *R-Guess* baseline

No prompt’s inference label distribution resembles the target label distribution.

The overall inferred label distribution, as illustrated in Figure 3.4 and Figure 3.5, differs markedly between the two prompting strategies and does not closely resemble the distribution of the ground truth labels. In both cases, the majority of ground truth labels mapped to each inferred label category were *DEFINITELY_NOT*, which is expected given that approximately 80% of the ground truth consists of *DEFINITELY_NOT* annotations.

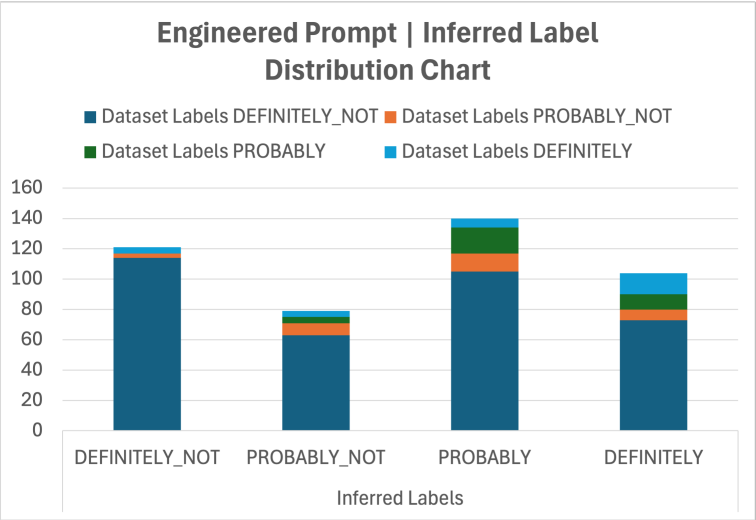


Figure 3.4: Inferred label distribution of the engineered prompt, broken down by target label components

Notably, in the case of the engineered prompt, each inferred label category — except for DEFINITELY_NOT — had the second-largest share of ground truth labels correctly aligned with the corresponding class, and this alignment was pronounced. This indicates a stronger tendency towards correct classification behavior. In contrast, no such pattern was observed for the baseline prompt, where inferred labels showed a much less structured alignment with the corresponding ground truth categories.

The complete dataset underlying these visualizations is provided in Appendix D.1.

3.2.2 Results by Target Label

Figure 3.6 demonstrates that, under the engineered prompt, each target label is most frequently associated with its correct corresponding inferred label. This indicates a strong alignment between the model’s output and the ground truth, showing a clear deviation from the behavior expected under random guessing.

In contrast, the results for the baseline prompt, shown in

The inference labels of the engineered prompt indicate correct classification behavior, the ones of the baseline prompt do not.

The engineered prompt is likely not random guessing.

The baseline prompt inference data suggests that the model follows an uneven distribution.

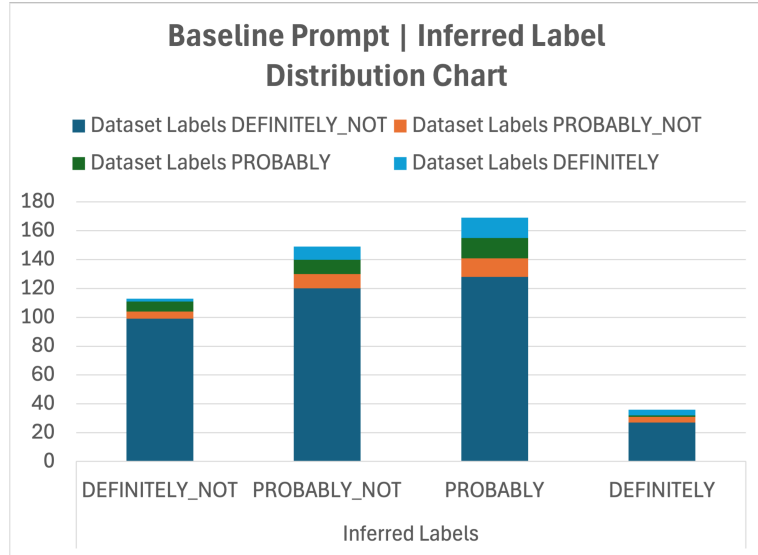


Figure 3.5: Inferred label distribution of the baseline prompt, broken down by target label components

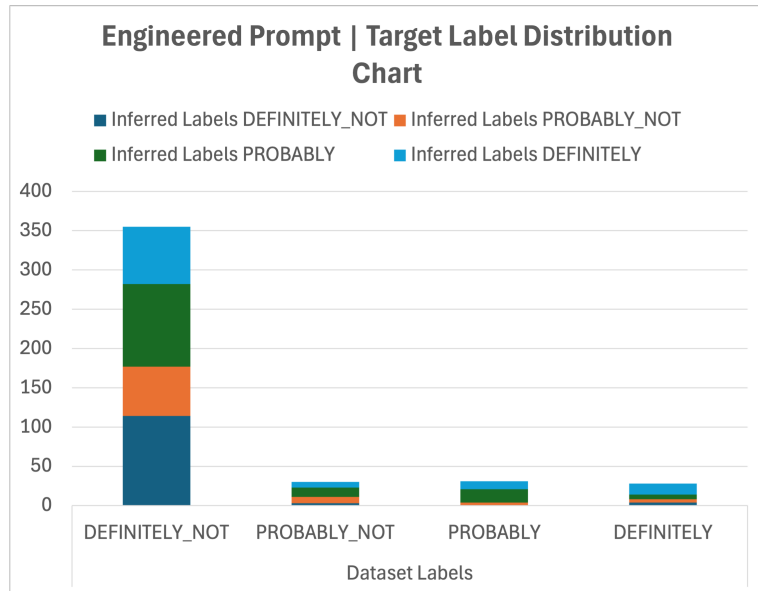


Figure 3.6: Target label distribution for the engineered prompt, broken down by inferred label components

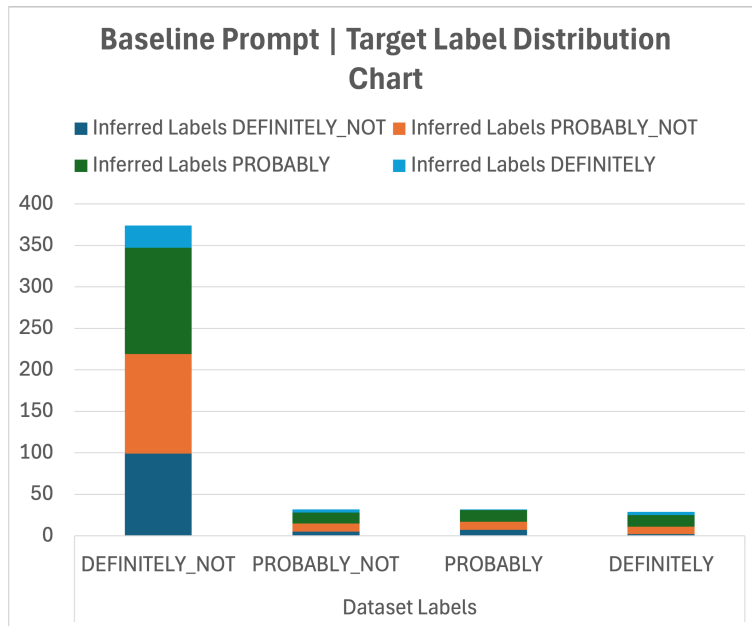


Figure 3.7: Target label distribution for the baseline prompt, broken down by inferred label components

Figure 3.7, reveal a much more uniform distribution of inferred labels across all target label categories. This mirrors the overall distribution of labels generated under the baseline configuration.

3.2.3 Results by Dark Pattern

In this section we mainly analyze the performance of the model on dark patterns that had actual positives. Graphs with the complete set of dark patterns can be found in Appendix D.2. When a result is referring to the complete set of dark patterns, this will be explicitly mentioned. Furthermore, we will focus mainly on the performance of the engineered prompt as the baseline prompt showed a weak performance in most key metrics.

Classification Rate

Social Engineering patterns perform well in classification rate metric.

The dark patterns that achieved the highest classification performance under the engineered prompt are, in decreasing order (as shown in Figure 3.8): *Endorsements and Testimonials*, *Limited Time Message*, *Forced Registration*, *Parasocial Pressure*, and *Feedforward Ambiguity*. Notably, three of these five patterns belong to the high-level category of *Social Engineering*. This trend extends further: with the exception of *Personalization*, all patterns within the *Social Engineering* category performed above the weighted macro average.

Obstruction patterns perform among the worst in the classification rate metric.

Conversely, the lowest-performing dark patterns under the engineered prompt, in increasing order, are: *Cuteness* (with a classification rate of zero), *Privacy Maze*, *Information without Context*, *Conflicting Information*, and *Visual Prominence*. Interestingly, within the high-level category of *Obstruction*, all sub-patterns performed below the weighted macro average — except for *Intermediate Currencies*, which, despite having the highest classification rate overall, had no actual positive instances in the dataset. This points to a particular weakness in the model’s handling of patterns in the *Obstruction* category.

The baseline prompt performed fairly uniformly weak suggesting a relatively random labeling approach.

In contrast, the baseline prompt produced more uniform performance across all dark patterns, with the highest classification rate reaching approximately 36.36% and the lowest dropping to around 9.09%. This narrow performance range aligns with earlier observations that the baseline prompt’s outputs resemble random guessing and show weaker pattern-specific sensitivity.

F1 Score

The engineered prompt performs well on the Endorsements and Testimonials pattern, the baseline does not. Hidden Information

The engineered prompt achieved its highest F1 scores — ranked in decreasing order — on the following dark patterns: *Feedforward Ambiguity*, *False Hierarchy*, *Hidden Information*, *Personalization*, and *Endorsements and Testimonials*. Two findings stand out: first, although *Personalization* had a relatively low classification rate, it performed well in terms of F1 score, suggesting a more balanced trade-off between

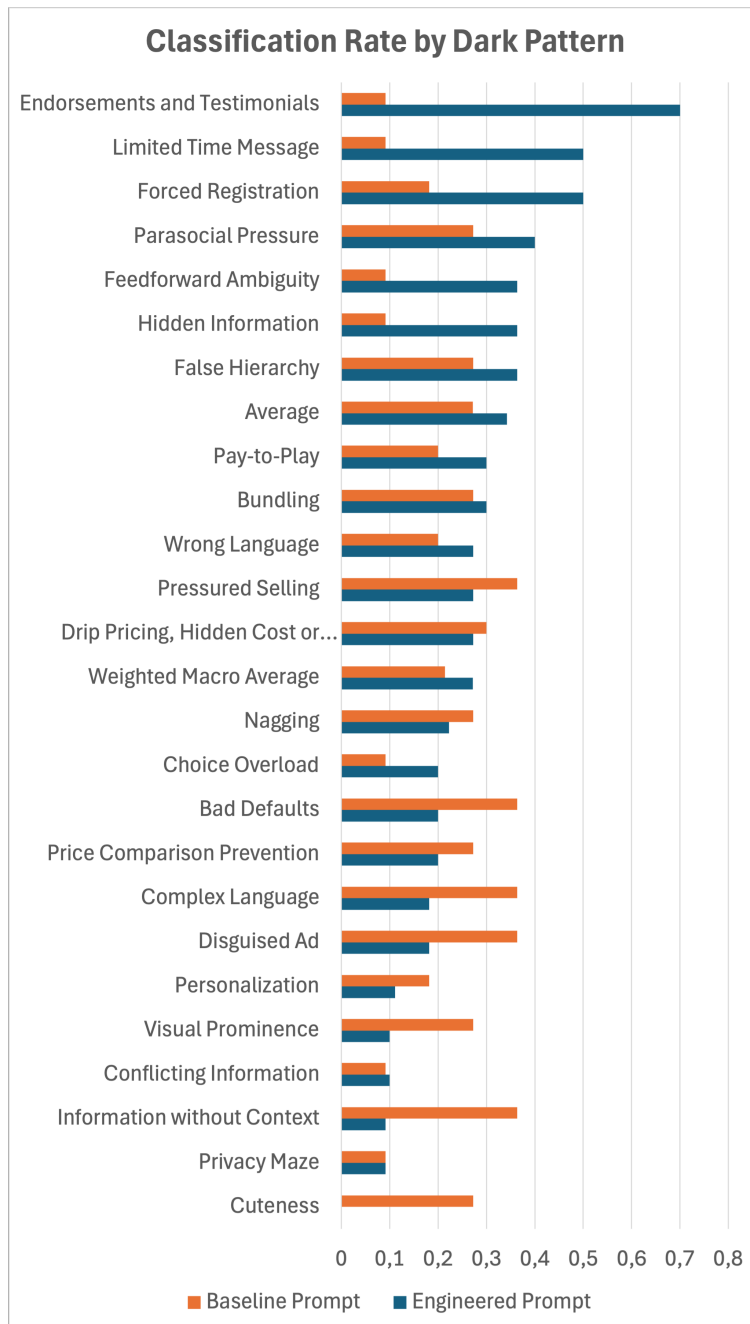


Figure 3.8: Classification rates by dark pattern, sorted by decreasing performance under the engineered prompt. Dark patterns with no actual positive instances (i.e., zero weight) are excluded from the analysis.

precision and recall. Second, *Endorsements and Testimonials* ranked highly on both classification rate and F1 score, whereas the baseline prompt performance for this pattern was among the lowest observed.

Social Engineering patterns' F1 Scores cannot match their classification rates.

When revisiting the broader performance of the *Social Engineering* high-level category, which performed well in terms of classification rate, the trend shifts in the context of F1 score. Only *Personalization* and *Endorsements and Testimonials* scored above the weighted macro average.

Interestingly, with *Privacy Maze* a pattern from the *Obstruction* category was able to achieve an above-average F1 score, albeit only slightly.

Interface Interference patterns perform the best and the worst.

Looking at high-level pattern categories, five of the eight patterns that scored above the weighted macro average in F1 are part of *Interface Interference*, giving this category the highest average F1 score across all categories at 50.31%.

At the other end of the spectrum, the lowest F1 scores were recorded for the following patterns (in increasing order): *Conflicting Information*, *Bad Defaults*, *Cuteness*, *Parasocial Pressure*, and *Pressured Selling* — with *Pressured Selling* being the only one in this group to achieve an F1 score greater than zero. Notably, three of the five lowest-performing patterns also belong to the *Interface Interference* category.

Recall

Recall is high across most dark patterns.

Out of the 24 dark pattern categories with actual positives, the engineered prompt achieved a recall of at least 0.5 in 20 of them. The four exceptions were *Parasocial Pressure*, *Cuteness*, *Bad Defaults*, and *Conflicting Information*.

Interface Interference patterns perform also among the best in recall.

A notable finding is the high performance of the *Interface Interference* category. Among its 11 dark patterns with actual positives, 8 achieved a perfect recall score of 1.0. This category also recorded a weighted macro average recall of 86.20%, making it the second-best performing cate-

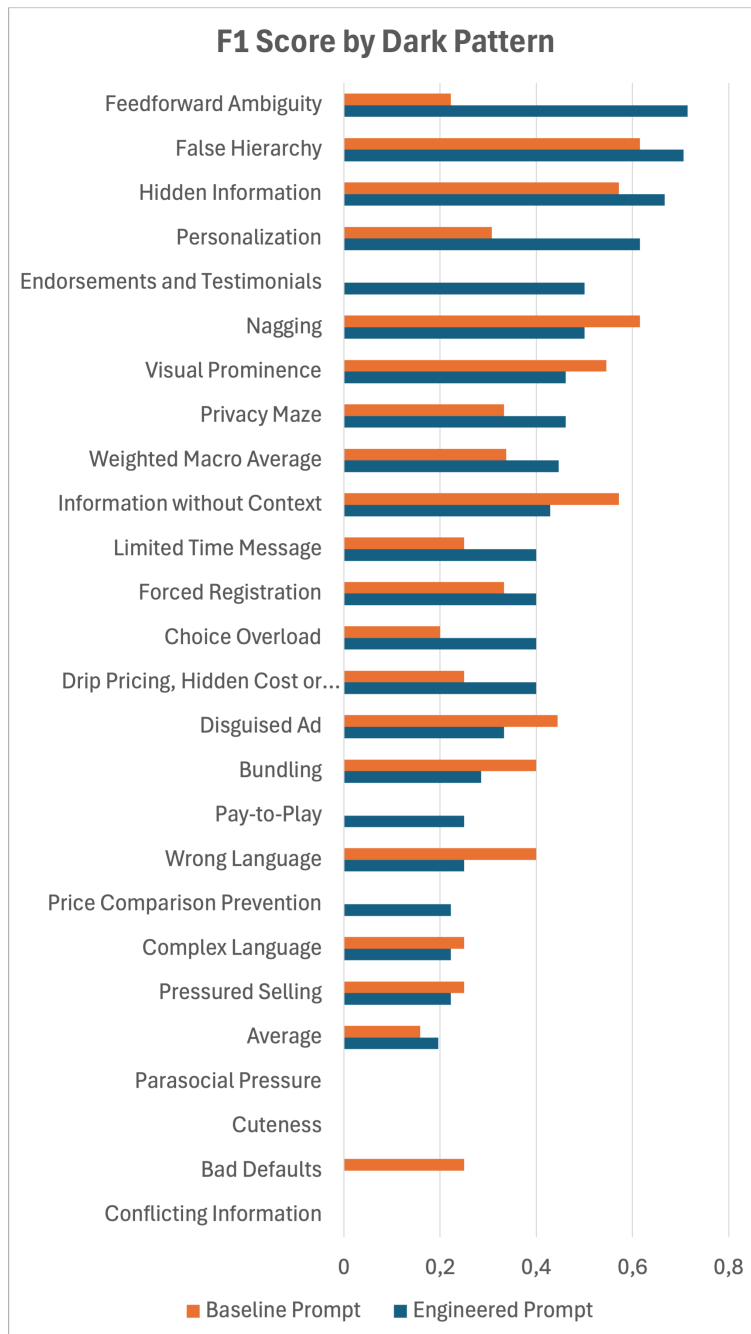


Figure 3.9: F1 Scores by dark pattern, sorted by decreasing performance under the engineered prompt. Dark patterns with no actual positive instances (i.e., zero weight) are excluded from the analysis.

gory overall — slightly behind *Sneaking* (87.5%). However, *Sneaking* had only 8 actual positives, while *Interface Interference* had 29, representing 46.77% of all positives.

Precision, Accuracy, Specificity, and General Observations

Precision scores by dark pattern closely mirrored those of the F1 scores, with high-performing patterns remaining consistent across both metrics.

Endorsements and Testimonials, Feedforward Ambiguity, Hidden Information, and False Hierarchy perform consistently well.

Overall accuracy and specificity were relatively low, largely due to a high false positive rate of 43.24%. Nonetheless, certain patterns — including Endorsements and Testimonials, Feedforward Ambiguity, Hidden Information, and False Hierarchy — consistently scored well across all metrics, indicating that the model was able to detect these patterns with both confidence and reliability.

Cuteness and Conflicting Information do not perform well.

On the opposite end, *Cuteness* and *Conflicting Information* consistently ranked among the lowest-performing patterns in all metrics, indicating severe problems in the detection process of these.

Invalid Hallucination Rate

Cuteness with a considerably higher Invalid Hallucination Rate than all others

The Invalid Hallucination Rate was particularly high for the *Cuteness* pattern, reaching approximately 27.27%, which aligns with the generally poor performance observed for this category. *Nagging* and *Personalization* also exhibited elevated hallucination rates, both around 18.18%.

Baseline prompt hallucinates considerably less, arguably due to simplicity.

Outside of these outliers, hallucination rates remained relatively low across most dark patterns. Interestingly, the baseline prompt exhibited near-zero hallucination rates for all but five patterns. With the exception of *Activity Message*, all hallucination rates under the baseline prompt remained below 10%, indicating that the model's outputs were generally syntactically well-formed, even if semantically inaccurate.

Chapter 4

Discussion

4.1 Discussion

In this section, we will discuss the results based on the underlying research question **RQ:** *How effective is GPT-4o in detecting known dark patterns in real-world websites?*

4.1.1 RQ1: Can prompt engineering improve GPT-4o's ability to detect dark patterns?

The results of this study clearly indicate that the more detailed instructions provided by the engineered prompt considerably improved the model's dark pattern detection performance. In contrast, the mirrored distribution of inferred labels across target label components under the baseline prompt suggests an "educated guessing" strategy — only weakly influenced by the actual input. The engineered prompt, by comparison, appears to have elicited a more context-driven analytical response. Notably, for the DEFINITELY_NOT inference label, the dominant target label was correct. For all other inferred labels, the second-largest component — after DEFINITELY_NOT — was consistently the correct label. This pattern further supports the hypothesis that the model was using contextual cues to guide its classifications under the engineered prompt.

Prompt engineering is highly effective.

4.1.2 RQ2: What technical limitations arise when applying GPT-4o to the analysis of real-world websites, and how can they be overcome?

Context window size is the most important technical limitation.

The most important technical limitation encountered in applying GPT-4o to real-world website analysis is the restricted context window. Currently, the full codebase of a modern website often exceeds the model's maximum token limit, making it infeasible to provide the entire HTML and CSS content in a single prompt. Although context windows have expanded substantially in recent model iterations, this remains a bottleneck in practical applications. However, it is expected that this limitation will diminish in the near future as context capacities continue to increase.

The `file_search` tool is the best workaround for context size limits.

In the current study, this constraint was partially mitigated through the use of OpenAI's `file_search` tool, which enables the model to semantically retrieve relevant code fragments from a vector store at inference time. While effective in theory, this retrieval mechanism introduces its own set of technical challenges. Chief among them is the relevance of the retrieved content: in practice, the retrieved chunks frequently do not align well with the structural boundaries of the source code — for example, beginning or ending in the middle of HTML elements or CSS class definitions. This misalignment can impair the model's ability to understand or reason about the document as a coherent whole.

The `file_search` tool does not chunk code logically.

It remains unclear whether these issues stem from the retrieval tool's chunking and embedding strategy, from the quality of GPT-4o's retrieval queries, or from more fundamental limitations in how website code is represented in the vector store. Regardless, these observations highlight the need for more sophisticated retrieval strategies — potentially DOM-aware chunking — to better support structured code understanding in large language models.

4.1.3 RQ3: How well does GPT-4o perform in classifying websites based on their use of dark patterns?

Overall, the results offer cautious optimism regarding the potential of prompt-engineered large language models for dark pattern detection. The substantial increase in F1 scores resulting from relatively simple prompt modifications suggests that further performance improvements are likely achievable through more advanced task formulations or refined prompting strategies. More importantly, the model's behavior demonstrates context-awareness and reasoning beyond random guessing, positioning GPT-4o as a promising foundation for future dark pattern detection systems.

Detection with LLM and in-context learning seems promising.

Observed Limitations. Despite these promising results, several limitations were observed:

Too much input and inline JavaScript confuse GPT-4o.

- GPT-4o's ability to interpret and reason about complex or lengthy code appeared to diminish with increasing input size and structural complexity. It remains unclear whether this degradation results from a limited understanding of the code itself or from inefficiencies in how the model utilizes extended context. This question lies beyond the scope of the current study.
- The inclusion of inline JavaScript often introduced confusion rather than insight. Instead of enhancing the model's understanding of website functionality, it frequently led to vague or incorrect reasoning — likely due to JavaScript's dynamic nature and opacity when extracted from its runtime environment.

Category-Level Insights. A closer examination of high-level dark pattern categories reveals contrasting trends. The *Interface Interference* category exhibited mixed performance: some patterns, such as *False Hierarchy*, were reliably identified — likely due to their strong association with visual and structural cues in the DOM. Others, such as *Cute-*

The model overgeneralizes simple visual indicators in case of Cuteness.

ness, performed poorly. Analysis of reasoning outputs for *Cuteness* revealed a tendency to overgeneralize visual indicators such as rounded corners or vibrant colors. These shallow heuristics are insufficient to capture the more abstract and emotional manipulation characteristic of the pattern.

The models seems to have a weak understanding of the defining characteristics of Social Engineering patterns.

The *Social Engineering* category generally achieved high classification rates but lower F1 scores. This suggests that while the model could reliably identify negative cases (i.e., absence of a pattern), it struggled to detect positive instances, indicating an incomplete understanding of what constitutes an affirmative example in this category.

Prompt simplicity and prompt detail influence the Invalid Hallucination Rate.

Format Adherence vs. Semantic Quality. Interestingly, the baseline prompt — despite its relatively poor classification performance — demonstrated a very low hallucination rate. This appears to stem from its structural simplicity, which likely made it easier for the model to adhere to the expected output format. This highlights a fundamental trade-off: detailed prompts may drive stronger reasoning and better classification, but also increase the risk of deviation from the required response structure.

The engineered prompt is potentially useful as a pre-filter for the automatic removal process.

Application Considerations. From a practical standpoint, one could argue that in certain workflows — such as serving as a pre-filtering step for downstream systems like the LLM-based dark pattern defusal approach proposed by Schäfer et al. [2025] — a more aggressive prompt with a higher false positive rate (as seen in the engineered prompt) may be preferable to a conservative one with more false negatives. However, in user-facing scenarios such as real-time dark pattern highlighting, this trade-off becomes problematic. A high number of false positives may result in irrelevant or misleading highlights, ultimately harming user experience rather than supporting it.

Interesting Edge Cases

Certain edge cases observed during evaluation provide further insight into the model’s limitations:

Parasocial Pressure. Despite a relatively high classification rate, both recall and precision for this pattern were zero. A review of the model’s reasoning suggests that it struggled to recognize brand logos — key indicators of parasocial framing. This failure likely stems from limitations in the retrieval-based context mechanism, as GPT-4o is generally capable of vision-based tasks in other settings.

Parasocial pressure reasoning results indicate weakness in computer vision.

Personalization. This pattern exhibited an inverse behavior compared to most *Social Engineering* patterns: it had one of the lowest classification rates but one of the highest F1 scores. Qualitative analysis indicates that the model provided sound reasoning in many cases, but often inferred positive labels for pages that merely collected user data without actively presenting personalized content — highlighting a subtle but important misunderstanding of the definitional boundary.

Subtle differences in definitions are occasionally not registered by the model.

Reasoning Behavior. No consistent pattern of incorrect or logically flawed arguments was observed across the dataset. Some responses were brief and vague, while others were detailed and specific. However, in some cases, the reasoning was detailed but clearly hallucinated, showing that LLM-generated explanations still need to be carefully checked — especially in important tasks.

The LLM reasoning output contains occasional hallucination instances.

4.2 Limitations of Our Work

4.2.1 Dataset

The dataset used in this study was labeled by a single in-

The dataset was created by a single individual.

dividual without peer review, which introduces potential concerns regarding the reliability and consistency of the ground truth labels — particularly in a subjective domain such as dark pattern detection. To mitigate this limitation, example labels were discussed with domain experts, and great care was taken to adhere strictly to the ontology definitions throughout the annotation process.

The dataset contains
few entries.

Another key limitation is the size of the dataset. With only eleven websites included, the sample is relatively small, and several dark patterns had no actual positive instances. This restricts the ability to draw comprehensive conclusions about detection performance across the full ontology. This constraint was driven in part by the budget and in part by the time limitations inherent to the scope of a bachelor’s thesis.

The dataset has a bias
towards popular
websites.

Additionally, the dataset is inherently biased toward high-traffic websites. All selected sites are among the most visited on the internet and are predominantly operated by large technology companies. This likely affects the generalizability of the findings, as the code structure and complexity of such sites may differ significantly from those of smaller businesses or independently run websites. This design choice was motivated by the intent to analyze websites with high real-world relevance.

The dataset may have an
overlap with the model’s
training data.

Furthermore, it is possible that some of these highly popular websites were part of GPT-4o’s training corpus. If so, the model may have had an advantage in analyzing these specific sites, potentially inflating performance relative to unseen or less common websites.

4.2.2 Lack of Iterations

Result robustness was
not tested.

Given that large language models like GPT-4o do not produce deterministic outputs, running multiple inference rounds would have strengthened the reproducibility and robustness of the results. Unfortunately, repeated evaluations could not be performed due to time and budget con-

straints, again reflecting the practical limitations of a bachelor's thesis project.

4.2.3 Prompt Wording

A minor but relevant limitation lies in the wording of the prompt itself — specifically the class definition for `DEFINITELY_NOT`. The prompt used the term “proof” where “evidence” would have been more appropriate. While this may have introduced some ambiguity in edge cases, it was not found to significantly affect output quality during preliminary testing.

The wording of the engineered prompt shows minor inaccuracies.

Chapter 5

Summary and Future Work

5.1 Summary and Contributions

In this thesis, we contributed to the growing field of automatic dark pattern detection by developing and evaluating a novel approach that leverages GPT-4o to analyze real-world website code. Specifically, we demonstrated how current technical limitations of GPT-4o — particularly those related to handling large codebases — can be partially overcome using context retrieval strategies.

Two major contributions of this thesis are the detection approach and the insights into technical limitations.

We engineered a prompt that, while intentionally aggressive in classification, exhibited clear signs of input-based reasoning and significantly outperformed both a baseline prompt and a random guessing strategy. This suggests that, with thoughtful prompt design, GPT-4o is capable of more than superficial classification and can begin to generalize pattern understanding from code structure.

One major contribution is the engineered prompt, successfully improved from the baseline.

To support this investigation, we constructed a hand-labeled dataset of popular real-world websites, annotated according to a well-defined dark pattern ontology. Using this dataset as ground truth, we conducted and evaluated a performance study of the proposed classification method,

Another contribution is the target label dataset.

analyzing results across multiple dark pattern categories and metrics.

Together, these contributions form a foundation for future work on LLM-based dark pattern detection and provide initial evidence of the potential and limitations of using large language models for this task.

5.2 Future Work

Future work: further prompt engineering.

Future work can build upon this study in several directions. Firstly, improvements to the prompting strategy — such as more refined class definitions, context-sensitive guidance, or adaptive prompting — could further enhance classification performance.

Future work: dataset improvements.

Secondly, the creation of a larger and more diverse dataset modeled after the one introduced in this thesis would enable more robust evaluations and potentially support fine-tuning of large language models for the dark pattern detection task. Such a dataset could include websites of varying complexity, size, and ownership (e.g., small businesses, non-profits, or personal projects) to address generalizability.

Future work: study with in-context website chunks.

Additionally, a study involving smaller, focused code chunks from real-world websites could offer valuable insight into the comparative strengths of in-context learning versus retrieval-based architectures. The integration of visual information — such as screenshots — alongside code could also enhance the model’s capacity for visual pattern recognition.

Future work: computer use tool.

Another promising direction is the use of OpenAI’s *computer use* tool¹, which could support comprehensive website interaction analysis. This would allow for more advanced detection of dark patterns embedded not only in

¹ <https://platform.openai.com/docs/guides/tools-computer-use>, accessed on June 25, 2025

static code, but also in dynamic behaviors and multi-step navigation flows.

Another technological possibility is to explore multi-agent architectures to automate user simulation strategies such as the "Choose Your Own Adventure" approach proposed by Mills and Whittle [2023]. This could offer a scalable way to evaluate user flows and behavioral triggers for dark patterns.

Finally, as context window sizes continue to expand in future LLM architectures, the possibility of analyzing full website codebases entirely in context — without the need for chunking or retrieval — represents a highly promising research avenue.

Future work:
multi-agent systems

Future work:
Exploration of future
LLM capabilities once
context size fits whole
websites.

Appendix A

Playwright Configurations

```
def init_crawler() -> tuple[Playwright, Browser]:
    playwright = sync_playwright().start()
    browser = playwright.chromium.launch()
    return playwright, browser

def crawl_website(url: str, browser: Browser, playwright: Playwright) -> Page:
    try:
        page = browser.new_page()
        page.goto(url, wait_until="load", timeout=0)
        return page
    except Exception as e:
        print(f"Failed to crawl website: {url}")
        print(e)
        print("Retrying with headless=False")

    try:
        browser_without_headless = playwright.chromium.launch(headless=False)
        page = browser_without_headless.new_page()
        page.goto(url, wait_until="load", timeout=0)
        return page
    except Exception as e:
        print(f"Failed to crawl website without headless: {url}")
        print(e)
        print("Retrying with a user agent")
```


Appendix B

Notable Preliminary Test Results

B.1 Model Output with an Inaccurate Understanding of the Dark Pattern "Positive or Negative Framing"

18. ****Positive or Negative Framing****: This pattern involves framing information positively or negatively to influence decisions. The element uses positive framing with "HOTLY DESIRED," suggesting desirability.

B.2 Model Output with Classification Based on CSS Class Name

```
[{
  "location": "CSS file",
  "context": "Found references to a countdown timer in CSS styles
defining interactions for
'#nav-swmslot #nav-swm-countdown-timer-widget'.",
  "keywords": ["countdown timer", "nav-swmslot", "CSS widget"]
}]
```

B.3 Model Output with Misunderstanding of the Classification Label

```
{  
  "used": "PROBABLY_NOT",  
  "reasoning": "Based on the definition of 'Immortal  
Accounts' as provided in the ontology, this dark  
pattern involves creating a 'Roach Motel' where  
user accounts are difficult or impossible to delete,  
thereby misleading users into thinking they can  
remove their data later. Upon inspecting the uploaded  
'index.html,' I found no explicit evidence suggesting  
obstacles or the complete prevention of account  
deletion (e.g., unclear deactivation options, lengthy  
delays, or missing pathways for account deletion  
processes). Since no account deletion mechanisms or  
obstructions could be identified in the provided  
webpage code, it is likely that the pattern is not  
used here. However, a deeper exploration of the  
backend system or other linked pages might be needed  
for a definitive conclusion."  
}
```

Appendix C

Final Prompts

C.1 System Prompt - Baseline

You are provided one dark pattern in an uploaded website and are supposed to determine if the website uses it. Good answers are in JSON format and contain a field 'used' with one of the following classification string values 'DEFINITELY_NOT', 'PROBABLY_NOT', 'PROBABLY', 'DEFINITELY' that represents if the website uses the specified dark pattern. Furthermore they contain a field 'reasoning' explaining where in the webpage the dark pattern is used including an example.

C.2 Input Prompt - Baseline

Analyze the uploaded webpage index.html regarding the dark pattern '{dp}'. Use the provided ontology.pdf for definitions and additional information.

C.3 System Prompt - Engineered

You are a specialist for the digital crimes unit investigating webpages for their use of dark patterns. You help the detectives by creating a report regarding the use of one specified dark pattern in a website they upload. You create a thorough step-by-step analysis of the webpage. Good answers first describe your train of thought and finish with a JSON object. This JSON object contains a field 'used' with one of the following classification string values 'DEFINITELY_NOT', 'PROBABLY_NOT', 'PROBABLY', 'DEFINITELY' that represents if the website uses the specified dark pattern. Furthermore it contains a field 'reasoning' explaining where in the webpage the dark pattern is used including an example. The meaning of the classification strings are as follows

'DEFINITELY_NOT': No indication of the dark pattern or dark pattern irrelevant to this website,
'PROBABLY_NOT': Slight smells but not enough to be considered a dark pattern, 'PROBABLY': Strong smells suggesting the use of the dark pattern, 'DEFINITELY': Conclusive proof that the dark pattern is used.

C.4 Input Prompt - Engineered

Analyze the uploaded webpage index.html regarding the dark pattern '{dp}' by singling out relevant HTML elements and then analyzing them regarding '{dp}'. Use the provided ontology.pdf for definitions and additional information.

Appendix D

Result Data

D.1 Inferred Label Distribution Broken Down into Target Label Components

		Inference			
		DN	PN	P	D
Target	DN	94.21	79.75	75.00	70.19
	PN	2.48	10.13	8.57	6.73
	P	0	5.06	12.14	9.62
	D	3.30	5.06	4.29	13.46

Legend: DN = DEFINITELY_NOT, PN = PROBABLY_NOT, P = PROBABLY, D = DEFINITELY.

Table D.1: Inferred labels of the engineered prompt broken down into the target label components (in percent).

		Inference			
		DN	PN	P	D
Target	DN	87.61	80.53	75.73	75.00
	PN	04.42	6.71	7.69	11.11
	P	6.19	6.71	8.28	2.77
	D	1.76	6.04	8.28	11.11

Legend: DN = DEFINITELY_NOT, PN = PROBABLY_NOT, P = PROBABLY, D = DEFINITELY.

Table D.2: Inferred labels of the baseline prompt broken down into the target label components (in percent).

D.2 Performance Metrics Graphs for Complete Set of Dark Patterns

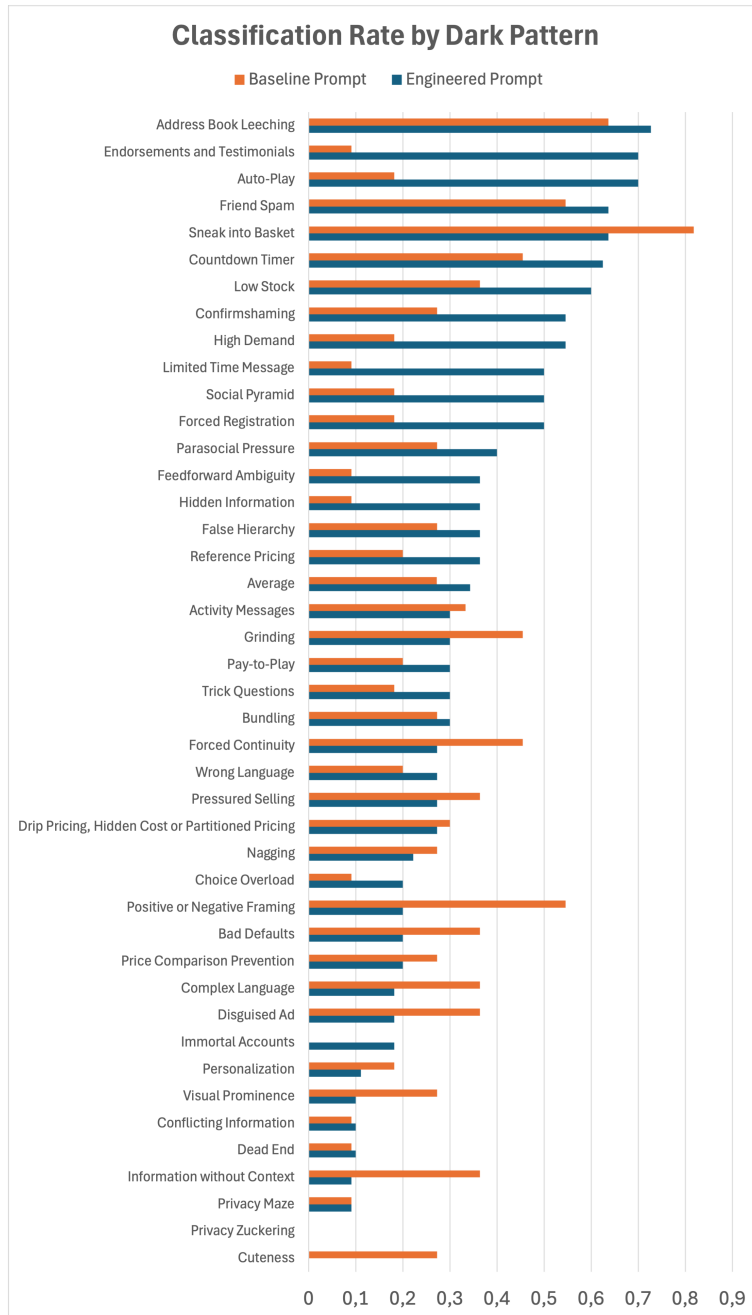


Figure D.3: Classification rate by dark pattern, sorted by decreasing performance under the engineered prompt.

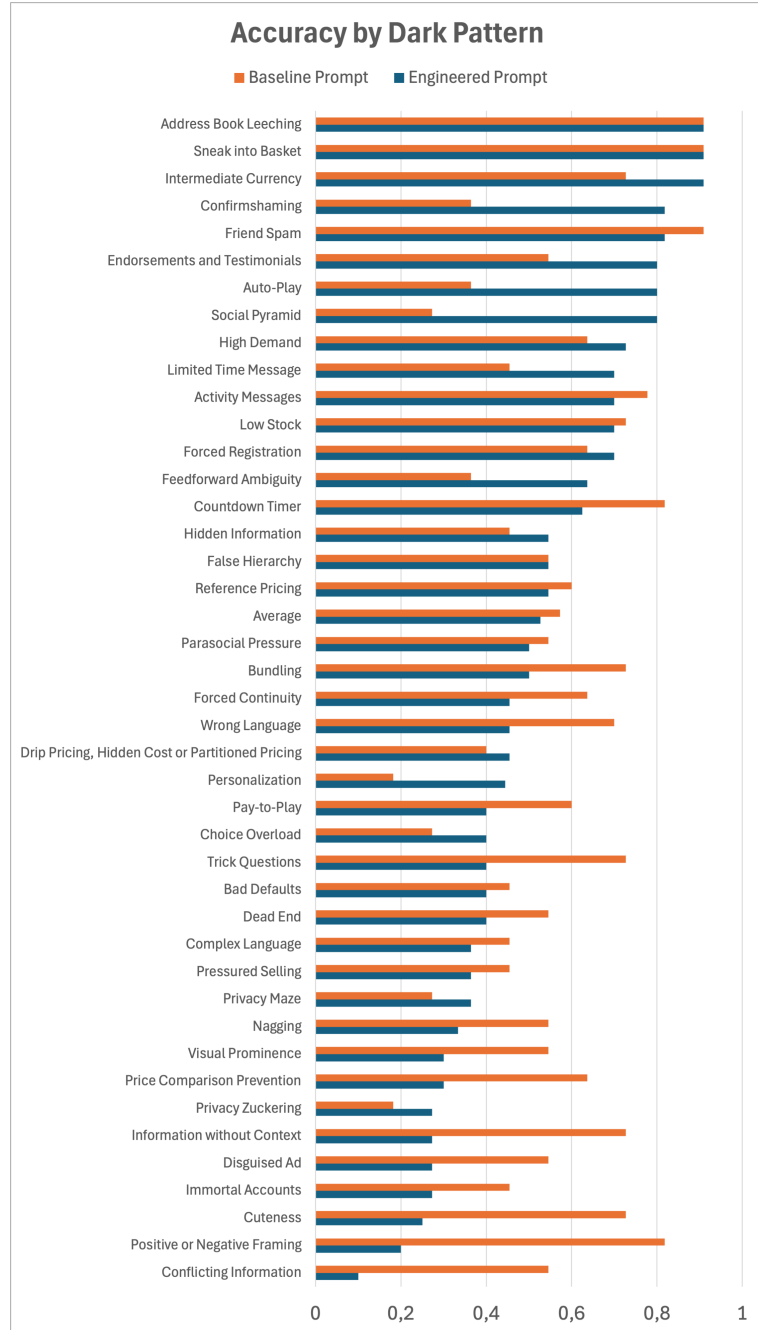


Figure D.4: Accuracy by dark pattern, sorted by decreasing performance under the engineered prompt.

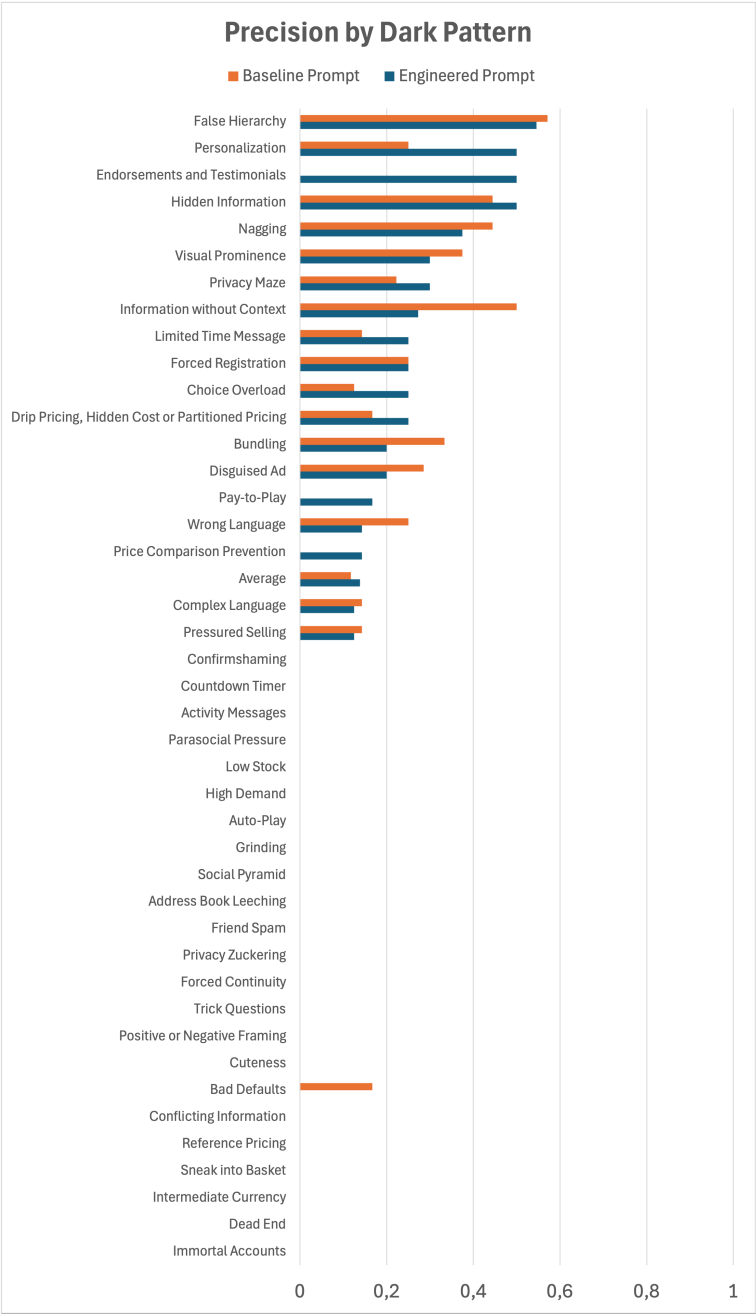


Figure D.5: Precision by dark pattern, sorted by decreasing performance under the engineered prompt.

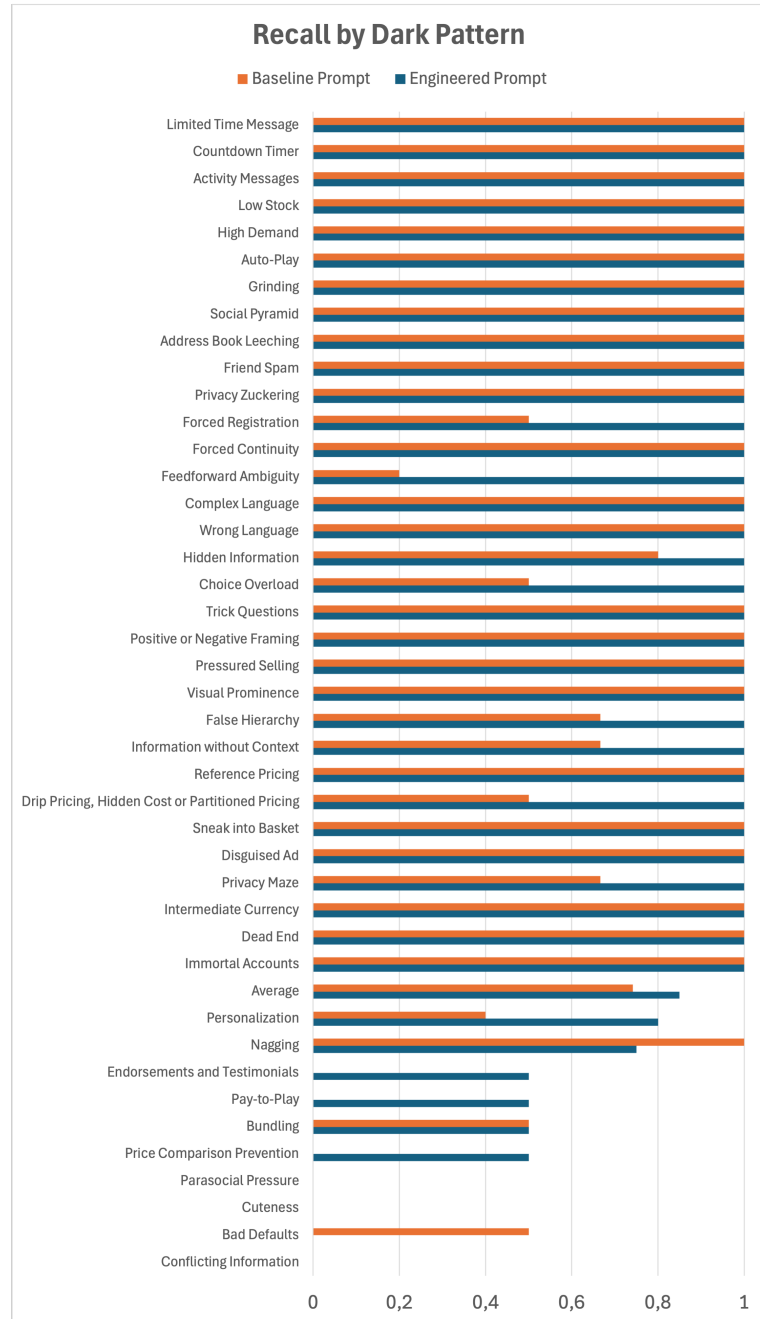


Figure D.6: Recall by dark pattern, sorted by decreasing performance under the engineered prompt.

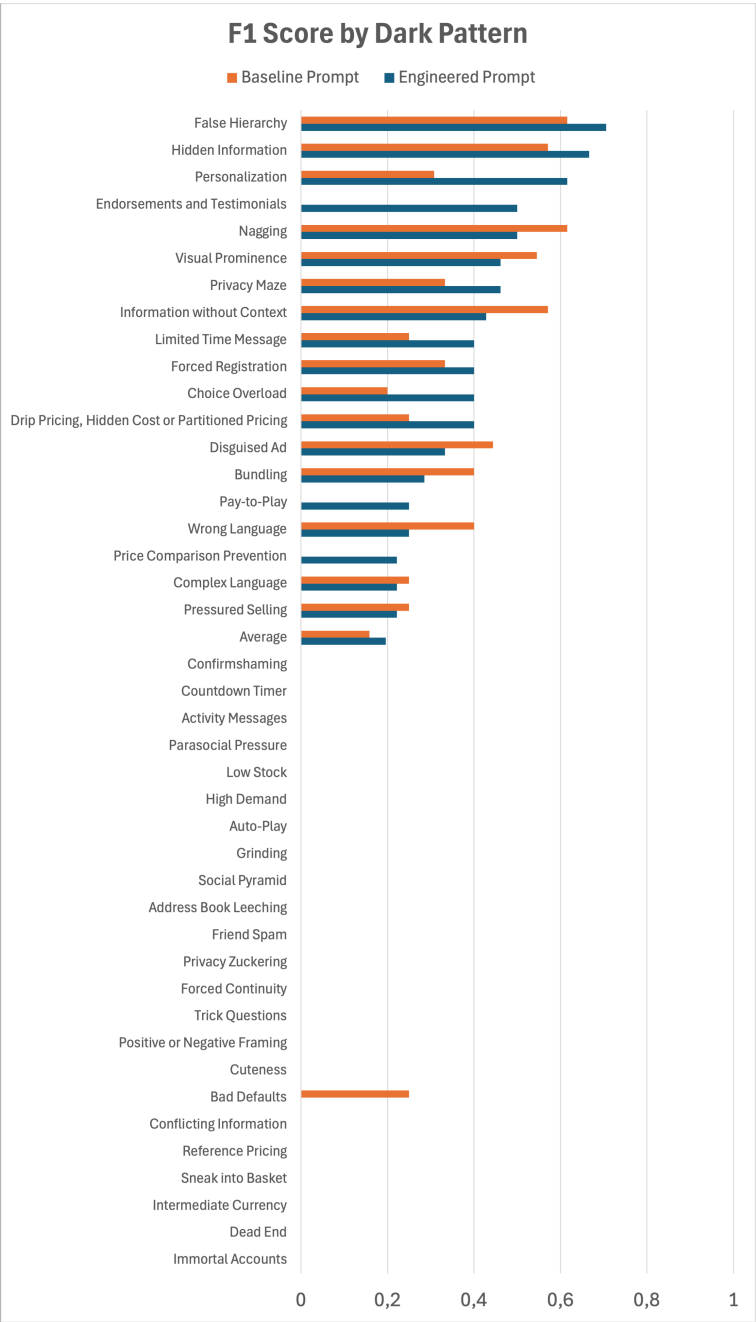


Figure D.7: F1 Score by dark pattern, sorted by decreasing performance under the engineered prompt.

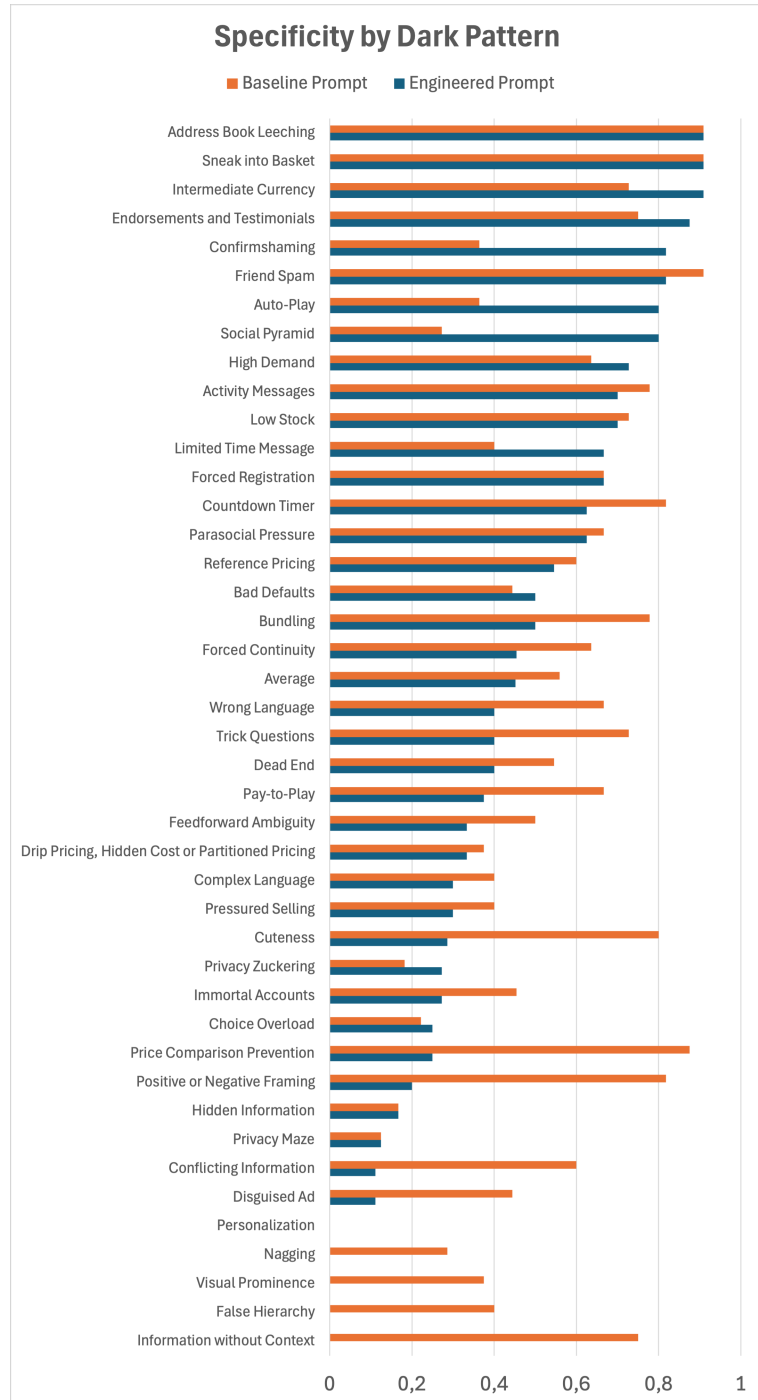


Figure D.8: Specificity by dark pattern, sorted by decreasing performance under the engineered prompt.

Bibliography

- [1] Juris Hannah Adorna, Aurel Jared Dantis, Rommel Feria, Ligaya Leah Figueroa, and Rowena Solamo. Developing a Browser Extension for the Automated Detection of Deceptive Patterns in Cookie Banners. In J. Caro, S. Hagi-hara, S. Nishizaki, M. Numao, and M. Suarez, editors, *Proceedings of the Workshop on Computation: Theory and Practice (WCTP 2023)*, Atlantis Highlights in Computer Sciences, pages 101–120. Atlantis Press International BV, 2024.
- [2] 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.
- [3] Christoph Bösch, Benjamin Erb, Frank Kargl, Henning Kopp, and Stefan Pfattheicher. Tales from the Dark Side: Privacy Dark Strategies and Privacy Dark Patterns. *Proceedings on Privacy Enhancing Technologies*, 2016:237–254, 07 2016. doi.org/10.1515/popets-2016-0038.
- [4] Jan Clusmann, Fiona R. Kolbinger, Hannah Sophie Muti, Zunamys I. Carrero, Jan-Niklas Eckardt, Narmin Ghaffari Laleh, Chiara Maria Lavinia Löffler, Sophie-Caroline Schwarzkopf, Michaela Unger, Gregory P. Veldhuizen, Sophia J. Wagner, and Jakob Nikolas Kather. The future landscape of large language models in medicine. *Communications Medicine*, 3(1):141, 2023. doi.org/10.1038/s43856-023-00370-1.
- [5] Gregory Conti and Edward Sobiesk. Malicious interface design: exploiting the user. In *Proceedings of the 19th International Conference on World Wide Web, WWW ’10*, page 271–280, New York, NY, USA, 2010. Association for Computing Machinery. doi.org/10.1145/1772690.1772719.
- [6] Marc Ghanem, Abdul Karim Ghaith, Victor Gabriel El-Hajj, Archis Bhandarkar, Andrea de Giorgio, Adrian Elmi-Terander, and Mohamad Bydon. Limitations in Evaluating Machine Learning Models for Imbalanced Binary Out-

- come Classification in Spine Surgery: A Systematic Review. *Brain Sciences*, 13 (12), 2023. doi.org/10.3390/brainsci13121723.
- [7] 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.
- [8] 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 2024 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.
- [9] Chi Han, Qifan Wang, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. LM-Infinite: Simple On-the-Fly Length Generalization for Large Language Models, 2024. URL <https://openreview.net/forum?id=p0ujzgHIRY>.
- [10] Philip Hausner and Michael Gertz. Dark Patterns in the Interaction with Cookie Banners, 2021. URL <https://arxiv.org/abs/2103.14956>.
- [11] Wenpin Hou and Zhicheng Ji. Comparing large language models and human programmers for generating programming code, 2024. URL <https://arxiv.org/abs/2403.00894>.
- [12] Woon Chee Koh and Yuan Zhi Seah. Unintended consumption: The effects of four e-commerce dark patterns. *Cleaner and Responsible Consumption*, 11: 100145, 2023. doi.org/10.1016/j.clrc.2023.100145.
- [13] Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large Language Models are Zero-Shot Reasoners. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 22199–22213. Curran Associates, Inc., 2022.
- [14] Victor Le Pochat, Tom Van Goethem, Samaneh Tajalizadehkhoob, Maciej Korczyński, and Wouter Joosen. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In *Proceedings of the 26th Annual Network and Distributed System Security Symposium*, NDSS 2019, February 2019. doi.org/10.14722/ndss.2019.23386.
- [15] Hanmeng Liu, Ruoxi Ning, Zhiyang Teng, Jian Liu, Qiji Zhou, and Yue Zhang. Evaluating the Logical Reasoning Ability of ChatGPT and GPT-4, 2023. URL <https://arxiv.org/abs/2304.03439>.

- [16] 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.
- [17] 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.
- [18] 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. *Proc. ACM Hum.-Comput. Interact.*, 3(CSCW), November 2019. doi.org/10.1145/3359183.
- [19] 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.
- [20] Stuart Mills and Richard Whittle. Detecting Dark Patterns Using Generative AI: Some Preliminary Results. <https://ssrn.com/abstract=4614907>, October 2023. URL <http://dx.doi.org/10.2139/ssrn.4614907>. Available at SSRN: <https://ssrn.com/abstract=4614907> or <http://dx.doi.org/10.2139/ssrn.4614907>.
- [21] Frank Papenmeier, Josephine Halama, and Carl Reichert. Accepting cookies: Nudging, deceptive patterns and personal preference. *Computers in Human Behavior*, 168:108641, 2025. doi.org/10.1016/j.chb.2025.108641.
- [22] 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)*, pages 569–573, 2023. doi.org/10.1109/APSEC60848.2023.00072.
- [23] 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.
- [24] 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.

-
- [25] Sakib Shahriar, Brady D. Lund, Nishith Reddy Mannuru, Muhammad Arbab Arshad, Kadhim Hayawi, Ravi Varma Kumar Bevara, Aashrith Mannuru, and Laiba Batool. Putting GPT-4o to the Sword: A Comprehensive Evaluation of Language, Vision, Speech, and Multimodal Proficiency. *Applied Sciences*, 14 (17), 2024. doi.org/10.3390/app14177782.
- [26] Ruihao Shui, Yixin Cao, Xiang Wang, and Tat-Seng Chua. A Comprehensive Evaluation of Large Language Models on Legal Judgment Prediction. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 7337–7348, Singapore, December 2023. Association for Computational Linguistics. doi.org/10.18653/v1/2023.findings-emnlp.490.
- [27] Sonish Sivarajkumar, Mark Kelley, Alyssa Samolyk-Mazzanti, Shyam Visweswaran, and Yanshan Wang. An Empirical Evaluation of Prompting Strategies for Large Language Models in Zero-Shot Clinical Natural Language Processing: Algorithm Development and Validation Study. *JMIR Med Inform*, 12:e55318, Apr 2024. doi.org/10.2196/55318.
- [28] 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, 2022. URL <https://arxiv.org/abs/2204.11836>.
- [29] Yuki Yada, Jiaying Feng, Tsuneo Matsumoto, Nao Fukushima, Fuyuko Kido, and Hayato Yamana. Dark patterns in e-commerce: a dataset and its baseline evaluations. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 3015–3022, 2022. doi.org/10.1109/BigData55660.2022.10020800.
- [30] José P Zagal, Staffan Björk, and Chris Lewis. Dark patterns in the design of games. In *Foundations of Digital Games 2013*, 2013.

Index

abbrv	<i>see</i> abbreviation
Accuracy	29
Activity Message	40
Add-On Extensions	10
ALBERT	13
Anticipatory Prompting	23
Automated Detection Tools	12
Azure	27
Bad Defaults	38
Baseline Prompt	25
BERT	13
Bucket	29
Chain of Thought Prompting	25
Class Labels	18
Classification Rate	28
Claude	27
Codebase	17
Computer Use Tool	50
Conflicting Information	36, 38, 40
Context Size	20
Context Window	21
Contributions	49–50
Cookie Banner	12, 13
CoT	25

Countermeasures	10–15
Cuteness	36, 38, 40
Dark Pattern Categories	9
Dark Pattern Definitions	9
Dataset	17
Dataset Label Distribution	19
Digital Markets Act (DMA)	8
Digital Services Act (DSA)	7
Discussion	41–47
Endorsements and Testimonials	36, 40
Engineered Prompt	25
F1 Score	29
False Hierarchy	36, 40
Feedforward Ambiguity	36, 40
file_search Tool	20, 28
Forced Registration	36
Future Work	50–51
GPT-3	13, 15
GPT-4	11, 14, 17, 20
Graph Neural Network (GNN)	12
Heuristic Prompting	25
Hidden Information	40
high-level pattern	9
Information without Context	36
Input	27
Instructions	27
Interface Interference	38
Intermediate Currencies	36
Intervention Space	10
Invalid Hallucination Rate	29, 40

JavaScript.....	21
Likert-Scale.....	18
Limited Time Message.....	36
Llama.....	27
low-level pattern.....	9
max_output_tokens.....	27
meso-level pattern.....	9
Multi-Agent Systems.....	51
Nagging.....	40
Obstruction.....	36, 38
OECD Report.....	7
Ontology.....	9–10
Parameters.....	27
Parasocial Pressure.....	36, 38
Persona Prompting.....	25
Personalization.....	36, 40
Playwright.....	23
Plug-Ins.....	10
Precision.....	29
Pressured Selling.....	38
Privacy Maze.....	36, 38
R-Guess.....	30
Recall.....	29
ResponseAPI.....	27
RoBERTa.....	13
script-tag.....	22
script-Tags.....	21
Sneaking.....	40
Social Engineering.....	36, 38

Specificity	29
Target Label Distribution	19
Technical Countermeasures	10, 12
Temperature	27
Tendency	29
Tools	27
top_p	27
Tranco List	17
Uniform Random Guessing	30
Visual Prominence	36
Weighted Macro Average	29
XLNet	13

