WaybackUI: A Dataset to Support the Longitudinal Analysis of User Interfaces

LEAVE ANONYMOUS FOR REVIEW, Leave Anonymous, Leave Anonymous

How do user interfaces (UIs) change over time? Understanding the evolution of UIs is essential for assessing the impact on users and automated systems that interact with them. To this end, we collected *WaybackUI*, a dataset of nearly 100,000 UIs from 2014 – 2024 mined from publicly available internet archives, paired with rendered screenshots and programmatically-extracted semantics. The resulting data allowed us to analyze how a decade of UI changes has impacted *i)* visual design, *ii)* accessibility, and *iii)* automated systems that interact with UIs. Examples of our findings include: *i)* a growing preference for muted color palettes in visual design, *ii)* an increase in the number of highly inaccessible web pages in recent years, and *iii)* quantifying the impact of data drift on the performance of UI understanding models. We conclude with a discussion of how *WaybackUI* can enable future data-driven discovery and understanding of UI trends.

Additional Key Words and Phrases: user interface, dataset, longitudinal analysis, design, accessibility, modeling

ACM Reference Format:
Leave Anonymous for Review. 2018. WaybackUI: A Dataset to Support the Longitudinal Analysis of User Interfaces. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 23 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Evolving standards, design trends, and new technology constantly reshape the appearance and implementation of user interfaces (UIs). How do these changes affect how users and automated systems that interact with them? For creators of UIs, new design systems and frameworks influence how they surface application content and functionality. For example, previous work conducted an analysis of the Material Design system measured the rate at which new UI guidelines were adopted within the Android ecosystem and the corresponding effects on user ratings [16]. Changes in UIs can affect end-users' perception of application usability. Several projects have scanned popular mobile apps and websites over time to measure the proportion of accessible applications and inform development recommendations [20, 53]. Finally, automated systems such as software testing frameworks and accessibility technology often rely on machine-learning models trained on large datasets of UIs [58]. Understanding the performance of these systems as real-world UIs gradually diverge from a fixed training dataset [21] could improve their utility and robustness.

Yet, it is challenging to perform these types of longitudinal analyses, in part due to the difficulty of capturing the code, content, and asset dependencies needed to accurately archive and replay UIs.[1] Previous research often focused on individual artifacts such as screenshots [14, 30], web page source code [23, 53], and other extracted metadata [20], or were associated with multiple snapshots of popular apps and websites recollected over a duration of several years [20, 53]. However, these approaches are limited in fully capturing UIs and often require years of data collection.

[1]For example, saving a web page's HTML to a file and opening the local copy will often lead to errors.

Author's Contact Information: Leave Anonymous for Review, leave@anonymous, Leave Anonymous, Leave Anonymous.

Manuscript submitted to ACM

In this paper, we present *WaybackUI*, a dataset of nearly 100,000 UIs from 2014 – 2024 downloaded from publicly available internet archives. We describe the collection process of *WaybackUI*, which involved mining web archives (WARCs) captures of popular websites, rendering screenshots, and programmatically extracting metadata. The resulting data allowed us to analyze how a decade of UI changes has impacted *i)* design trends, *ii)* accessibility, and *iii)* the computational modeling of UIs. We first examined the evolution of UI designs on the web, tracing the impact of several trends using quantitative and qualitative methods. Our analysis revealed a growing preference for "muted" color palettes in the visual design of websites and layout changes stemming from responsive design. *WaybackUI* also enabled a large-scale longitudinal evaluation of accessibility [20] across the past decade. Using an automated accessibility checker, we measured the frequency and type of accessibility failures among popular websites, revealing an increasing majority of contrast-related errors, possibly related to visual design trends. We also identified an increasing number of "highly inaccessible" websites in recent years (since 2020) with over 100 accessibility errors. Our third analysis measured the temporal robustness of UI element detectors, a popular visual UI understanding model, when trained and evaluated on web data from different years. Our training experiments revealed that models trained on more recent UIs perform worse, suggesting that UI designs are becoming more difficult for computer vision models to learn. We used simulated back-testing experiments to show that models gradually lose around 5% of their original performance per year, suggesting that they should be updated roughly every 2 years to maintain 90% of their performance on new data. Finally, we examine the relationship between our findings and show the cascading effects of design (*e.g.,* color choice) and implementation (*e.g.,* toolkit accessibility) choices on accessibility and UI modeling performance.

To summarize, our paper presents the following contributions:

(1) We present the the *WaybackUI* dataset, consisting of about 100,000 UIs, collected from a decade (2014-2024) of captured website archives (*i.e.,* WARCs). We render these WARCs into screenshots, which we associate with programmaticaly-extracted semantic information.
(2) We use the *WaybackUI* dataset to conduct three data-driven analyses of the *i)* design, *ii)* accessibility, and *iii)* computational modeling of UIs. Throughout these categories, our experiments contribute to the understanding and discovery of trends over the past decade.

We plan to release our dataset and models upon acceptance.

2 Related Work

In this paper, we introduce the *WaybackUI* dataset and use it to measure the effects of a decade of UI changes. We contextualize our work in prior research related to *i)* UI datasets, *ii)* techniques for updating UI datasets, and *iii)* longitudinal analyses of UIs.

2.1 UI Datasets

There have been several datasets of UIs collected to support the data-driven analysis and modeling of mobile and web applications.

Rico [14] is a dataset consisting of 72,000 screenshots collected from 9,700 Android apps, and it has been used for various applications in data-driven design [36, 45], software engineering [9], accessibility applications [46, 50], and benchmarking machine learning models [35]. Several datasets have also been derived from Rico to enable applications such as UI element detection [8, 33] and screen classification [32]. AMP is a related dataset of 77,000 screenshots

from 4,000 iOS apps that was used to train Screen Recognition, a model for enhancing the capabilities of mobile app screen readers [58].

While screenshots are relatively easy to capture, many downstream applications require UI data to be annotated with additional semantic information. For example, CLAY [33], VINS [8], and Screen Recognition [58] employed human annotators to annotate the location and widget types of elements in screenshots; however, this process is costly or time-consuming. To this end, UI datasets collected from the web are promising, since web pages and browsers expose a lot of relevant metadata that can be automatically associated with rendered screenshots. Webzeitgeist [30] collected a database of 100,000 web pages and associated automatically-extracted metadata such as the document object model (DOM) to enable complex design-related search queries across layouts and elements. WebUI [56] showed that labels derived from automatically-extracted metadata could be used to train and improve visual UI understanding models, such as element detectors.

In our paper, we construct the *WaybackUI* dataset as a resource for longitudinal analysis of UIs, taking advantage of the web pages found in publicly available internet archives that we replayed using a browser. Like datasets collected by prior work, *WaybackUI* can support a wide range of applications using its rendered UIs, page source code, and programatically-extracted semantics.

2.2 Updating UI Datasets

UIs are constantly changing due to updates in standards, technology, and design trends, which necessitates periodic updates to datasets. Previous work has investigated several strategies keeping UI datasets up-to-date.

Some approaches have focused on applying advancements in data collection methodology to re-annotate and filter existing data. For example, Enrico [32] and CLAY [33] applied newer machine-learning workflows to help human annotators identify errors such as mismatches and artifacts in Rico [14], which allowed downstream applications to benefit from improved data quality.

However, as UIs evolve, the underlying data can become outdated. For example, since Rico [14] was collected, several major updates were made to Android's design guidelines, resulting in substantial visual differences. Other approaches aim to develop automated and more sustainable methods for collecting UI data without substantial human effort. WebUI is one example that crawled 400,000 web pages using multiple simulated devices and stored visual, semantic, and style information [56]. MUD applied similar crawling strategies that incorporated LLMs to more efficiently discover screens within Android apps, and used human annotators to label semantics of the UI screens [18]. The Never-ending UI Learner is a crawler that, in addition to interacting with applications to discover more UI screens, also observes the effects of its interactions to infer the affordances of UI elements (*e.g.,* tappability) and generate labels for machine learning models [55].

While these approaches are necessary for keeping datasets updated, a drawback is that they cannot be effectively used retrospectively, for example, to learn about trends that occurred in a previous window of time. In our paper, we mine publicly available internet archives provided by the Wayback Machine [27], which is a digital library of archived websites and media that has previously been used as a data source for research [5, 38]. The online archive contains replayable snapshots of websites dating back to 1998 (although we focus our analysis to the past decade), and is continuously updated using automated crawling.

2.3 Longitudinal Analyses of UIs

Large-scale UI datasets have been used to support a wide variety of design and accessibility-related longitudinal analyses.

Multiple studies have focused on examining the adoption of design patterns within the app ecosystems. Ablharbi et al. [4] tracked nearly 25,000 Android apps over an 18 month period and used decompilation techniques to detect the use of both newly-released and deprecated design patterns. More recently, Doosti et al. [13] analyzed the adoption rate of a set of UI components (*e.g.,* Floating Action Button) introduced with Material Design guidelines in 2017 and analyzed the effect on app installation and ratings. Both studies provided insights into how developers and designers adapt to evolving design standards.

Another area of work focused on UI conformance to accessibility standards. In the mobile app domain, Fok et al. [20] took monthly snapshots of 312 Android apps over a period of 16 months, revealing a mix of accessibility improvements and regressions within popular apps. Multiple analyses have analyzed trends within web accessibility [6, 11, 22, 23, 31, 42, 53] by automatically scanning websites using the standardized Web Content Accessibility Guidelines (WCAG) [54]. For example, the WebAIM Million project assessed the one million most popular web pages annually since 2019 [53] using the WAVE [28] API to generate statistics and recommendations. Similar to the work presented in this paper, other work used the Wayback Machine [27] to gather snapshots of high-traffic and essential government websites over longer periods of time. Perhaps the closest to our work is an analysis performed by Agarwal et al. [2] which scraped the Wayback Machine to track popularity trends and page complexity of the top 100 most-visited websites over a period of 24 years (1998 – 2021). This paper presents a larger-scale (nearly 100,000 UIs) and more comprehensive (coupled with rendered screenshots and UI semantics) dataset, which enables many additional types of data-driven analyses, such as model training and benchmarking.

3 WaybackUI Dataset

We present *WaybackUI*, a dataset of nearly 100,000 web pages from 2014 – 2024 (11 years, inclusive). To construct our dataset, we use the Internet Archive's Wayback Machine [27] to download snapshots of the 10,000 most popular websites from each year as web archives. We then follow automatic labeling techniques used by prior web-based UI collection strategies [56] to "replay" the captured WARCs, which generates rich visual and semantic information such as rendered screenshots and programmatically-extracted accessibility trees. To our knowledge, *WaybackUI* is the first large-scale longitudinal dataset of UIs that tracks these visual and semantic changes over time.

In this section, we provide additional details of *WaybackUI's* data collection and analyze its composition.

3.1 Dataset Construction

In our work, we chose to focus our dataset on the past decade, which has experienced several technological changes. For example, in late 2014, the specifications for HTML5 and CSS3, the most recent versions of the HTML and CSS web language standards, were finalized, which introduced features for more responsive (*e.g.,* improved layouts for web styling) and accessible (*e.g.,* semantic tags) web pages [52]. In 2016, internet traffic from mobile and tablet devices exceeded desktops for the first time and has since continued to grow, which has affected UI design to emphasize mobile and responsive design patterns [47]. Numerous other toolkits and libraries for developing web pages and web applications (*e.g.,* PWAs) have also influenced how UIs are designed and built. These trends, along with many others, could be captured in our dataset and analyzed using data-driven methods.

3.1.1 Website Selection. To make our dataset representative of the most popular websites each year, we used publicly available rankings to select website seeds for each year. We chose 10,000 as a target number of seeds for each year, since it is roughly the same size as other datasets that have been used data-driven modeling purposes [8, 56].

Finding the most popular websites for a given year presents challenges because there are no comprehensive, publicly available rankings that cover all years from 2014 – 2024. We relied on two popular sources of website rankings: Alexa Rank [3] and the Majestic Million [37] rankings. For 2014 – 2020, we used the Alexa rankings, since it estimates the number of daily visitors and has been used by prior work [2] for sampling popular websites. However, since the Alexa Rank was discontinued in 2020, we used Majestic Million as an alternative ranking source for subsequent years. Note that neither ranking list was available for download during our entire period of analysis (2014-2024), which necessitated a split in ranking sources.[2] Historical rankings for both sources were downloaded from captured snapshots hosted by the Wayback Machine.

3.1.2 Website Capture. To collect our dataset, we modified a distributed crawler open-sourced by previous work [56]. The crawler is implemented as a Node.js program that programmatically controls a headless browser using the Puppeteer library. The crawler consisted of several parallelized worker machines that were run on a public cloud provider. Each website was captured at a resolution determined based on the most popular desktop screen resolutions worldwide that year. From 2014 to 2020, the viewport was set at 1366x768 px, and from 2021 to 2024, the viewport was set at 1920x1080 px [48]. We followed post-processing (*e.g.,* ignoring small, visually undetectable elements) and metadata extraction techniques from previous work [56]. Using the page's accessibility tree, we captured element locations and their corresponding semantic attributes (*e.g.,* StaticText, Link, Heading).

Unlike other crawlers, our crawler only visited websites in its input seed list and didn't perform any additional traversal (*i.e.* discovering new pages to crawl) besides downloading the resources necessary to render the web page (*e.g.,* image assets and scripts). When replaying websites, the Wayback Machine automatically modified references (*e.g.,* links) to point to versions of the source archived at the same time as the page. For each crawled year, snapshots of websites closest to the beginning of the year (January 1st) were chosen. If the snapshot couldn't be properly replayed (*i.e.,* resulted in an error), our crawler attempted snapshots captured at subsequent dates. If more than three snapshots of a website from a year couldn't be properly replayed (*e.g.,* blocked the Wayback Machine's crawler), the website was excluded from our dataset, which usually resulted in roughly 10% fewer than 10,000 web pages being included in each year. Crawling occurred (non-continuously) over a period of four months and cost approximately $1000 to rent cloud instances.

3.2 Dataset Composition

WaybackUI was collected from archived websites, which presents a unique data composition. Because we focus on the web as a source of UIs, many of the UIs in our dataset likely differ from that datasets collected from the mobile domain [14, 20, 58]. Nevertheless, prior work [56] suggests that data collected from the web can generalize to other complementary domains, such as mobile UIs. Because we collected web pages archived at different points in time, the year-to-year composition of UIs also changes, reflecting the dynamic nature of the web. In this section, we examine some of these trends at a page and element level.

[2]The Majestic Million rankings were started in 2012; however, early versions were not archived on the Wayback Machine.

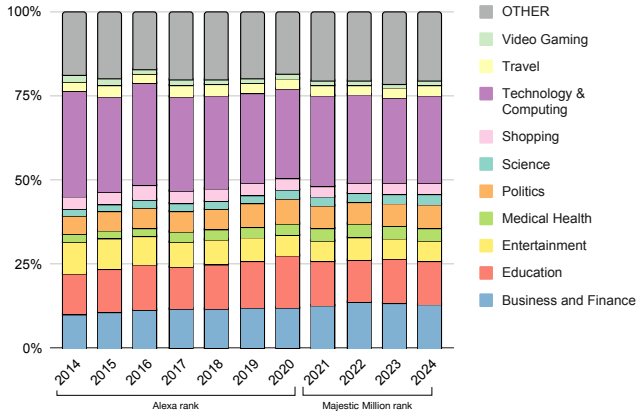


Fig. 1. The distribution of web pages found in our dataset based on categories defined by the IAB.

3.2.1 Page Analysis. To analyze the distribution of pages in our dataset, we used the Content Taxonomy 3.0 defined by the Interactive Advertising Bureau (IAB), which categorizes websites based on their content using various degrees of resolution (*e.g.,* tiers) [26]. We chose to use the highest-level classification tier, which resulted in 40 possible categories for websites. We used GPT-4o-Mini [1] to categorize websites in our dataset by their textual content, which is similar to previous approaches [49]. Figure 1 shows the distribution of popular websites in our dataset over the past decade, where we visualize the ten most frequent categories, along with an "Other" class. The distribution of page categories among the top 10,000 websites remained mostly the same, possibly because the makeup of the URL lists changed slowly. In our sample, "Technology & Computing" websites (including search engines, CDN, and hosting sites) were the most prevalent, followed by "Education" (which included online references such as Wikipedia, publishing, and academic sites).

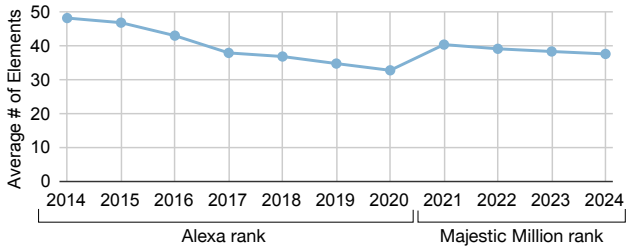


Fig. 2. Average number of elements for each year. This is calculated by taking total number of element annotations over the number of successful crawls.

3.2.2 Element Analysis. Unlike prior work [56] which assigned multiple categories to each UI elements, we used a "one-hot" format for representing element semantics, which simplified our analysis. Figure 3 shows the most frequent element types (originally represented as lists of accessibility attributes) found in our dataset and a corresponding label name. Because there are potentially numerous combinations of attributes, we only considered the top 10 most frequent ones, along with an "Other" class, to conduct our analysis. Figure 2 shows the average number of visible elements

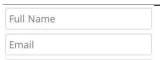
Example	Element Type	Accessibility Attributes
	Text	StaticText
	Hyperlink	StaticText,link
	List Item	StaticText,link,listitem
	Image	img
	Heading	StaticText,heading
	Paragraph	StaticText,paragraph
	Container	generic
	Linked Image	img,link
	Linked Heading	StaticText,heading,link
	Button	StaticText,button
	Other	—

Table 1. Element types in our dataset and their corresponding accessibility attributes. Examples of each element type is shown on the left.

on a page that are "above the fold" *i.e.,* visible without requiring interaction (scrolling) to reveal. Figure 3 shows the distribution of elements types across each year. In both figures, some trends can be observed, although fluctuations may have been caused by the different ranking methodologies of websites before and after 2021.

Overall, Figure 2 shows that the visible number of elements on each web page decreased over time. In 2014, there was an average of 48.3 elements visible per page, which decreased on average 5.4% per year, relative to the original number, until 2020, the last year based on Alexa Rank. The trend also continued from 2021 to 2024, where 2021 had an average of 40.5 elements per page and similarly until 2024, which had an average of 37.7 elements per page. It is possible that the period from 2021 – 2024 had more visible elements other years in our dataset because the most popular screen resolution increased from 1366x768 to 1920x1080, which led to more area for elements to appear. Interestingly, the WebAIM Million report [53] found that the complexity of websites increased from their sampled period of 2019 – 2024, where each year led to roughly 11% more home page elements. Besides differences in element segmentation and filtering (*e.g.,* we filtered out very small elements), this suggests that while the complexity of pages increased, UI designers sought to keep the initial appearance of the web page simple while moving content "below the fold" or requiring other interaction-based scaffolding (*e.g.,* accordion widgets).

Finally, the makeup of web pages also shifted over time (Figure 3). Proportionally, the number of visible text and image elements increased 30.0% and 76.1%, respectively, relative to their original proportions. Meanwhile, the number of hyperlinks decreased the most (30.1% relative to the original) over the same period.

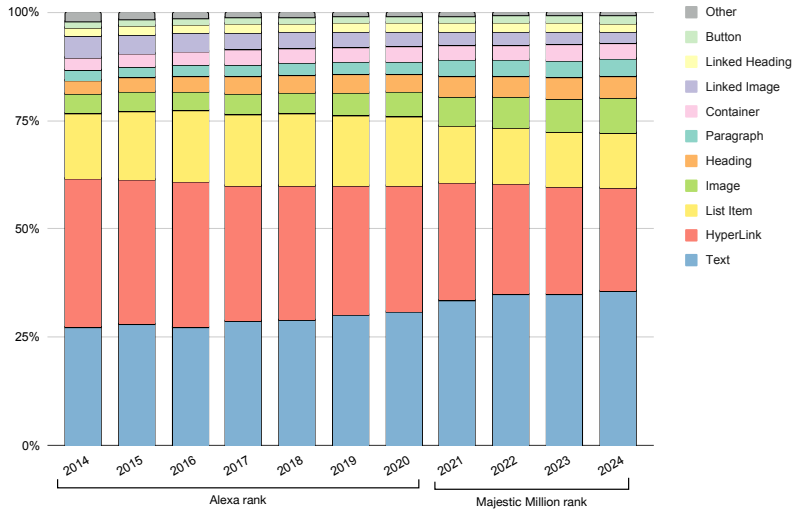


Fig. 3. The distribution of element types found in our dataset. Overall, the number of Text elements increased the most while the number of Hyperlink elements decreased the most.

4 Analyses of WaybackUI

The shifting makeup of web pages and the UI elements they contain suggest complementary trends that reflect how they are created and used. We used *WaybackUI* to analyze how a decade of UI changes impacted *i)* visual design, *ii)* accessibility, and *iii)* computational modeling of UIs using a combination of methodologies introduced by prior work and novel approaches afforded by the scale and richness of our data.

4.1 Design Changes

To measure trends in design, we used *WaybackUI* to perform qualitative and quantitative analysis of web page design over the last decade, focusing on layout and color. Through our analysis, we tracked several trends in UI design, such as responsive design, content layout patterns, and shifting color palettes.

4.1.1 Layout. We first analyzed the layouts that web pages have used to display content and how they have changed over time. We focused on the spatial distribution of visible UI elements, which we computed using our programmatically-extracted accessibility metadata. We adopted a procedure similar to prior work [30], which generated "heatmaps" to qualitative assess the placement of UI structures. In our layout analysis, we focused on the 2014 – 2020 subset of our data, since UIs from subsequent years were captured at a different resolution (1920x1080), which could influence aggregated element locations. Furthermore, we focused on the placement of the element types we used for element composition analysis (Table 1). While these represent the most frequent combinations of accessibility attributes, they represent lower-level semantics (*e.g.*, buttons and headings) than ones used by Kumar et al. [30] (*e.g.*, featured, sitemap). Figure 4 shows some illustrative examples of how placement has changed over time for three element types: Heading, Container, and Linked Image. Spatial maps for all element types across all years can be found in the appendix.

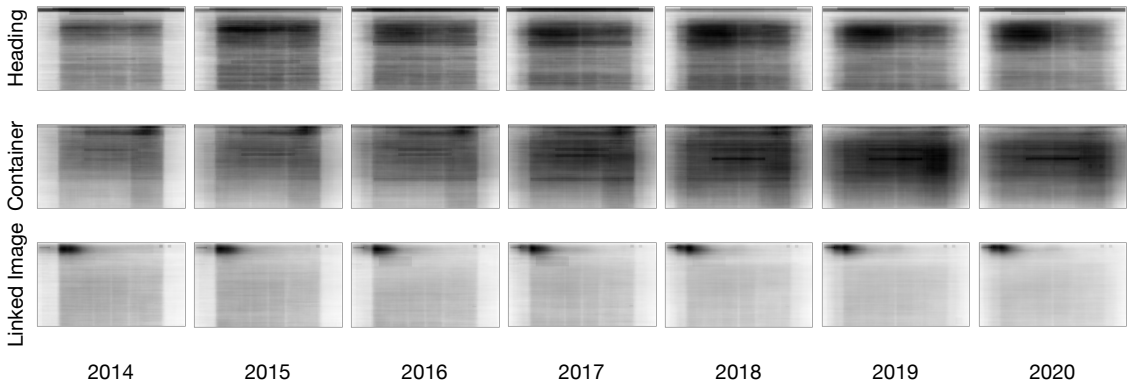


Fig. 4. Spatial probability distribution maps of element types Heading, Container, and Linked Image for years 2014 to 2020. While the positions of Linked Image stay consistent, other elements take up a wider portion of the viewport.

Some UI layout patterns remained constant over the past decade. For example, the Linked Images, placed at the top left of the screen, were continually used as hyperlinked logos, which can be used to return to a website's home page. Vertical containers are often placed at the right side of the screen for navigation purposes and advertisements.

We also observed several evolving trends. For example, from 2014, **the content width of pages gradually increased to fill the entire viewport.** Qualitatively, this is visible in all three element types, where there is initially an empty gap between content and the edges, similar to the page margin of documents. While the size of this content margin stayed constant (around 15% of the screen width), an increasing number of pages included content that filled in this space. One motivation for this shift could be the increasing consumption of web content on small-screened devices like mobile phones and tablets, which surpassed desktop usage in 2016 [47]. Making visual content larger decreases the need for "zooming in" on smaller screens to see content [7], which is consistent with our observation that there is, on average, fewer initially visible elements (Figure 2). A related reason for these changes could be the adoption of responsive design, which creates layouts that adapt content to display factors such as screen size and device type (*e.g.,* mobile device). One common implementation method is to tie the width of the primary content container to the size of the viewport, which can be observed from the size of Container elements (Figure 4).

Other effects of responsive layout practices are visible in the spatial distribution of heading elements. In 2014, headings were primarily placed at the top of the screen or the main content area of the page. Plots from subsequent years suggest that **the size of heading text increased and heading text was placed at more prominent locations of the page.** This observation could be explained, in part, by the increasing use of "hero" or "call to action" patterns recommended by web design manuals [19] which suggest using a large-size piece of text to emphasize a message or draw attention to a specific action.

4.1.2 Color. In addition to UI layout, color choice can have a large impact on visual design and aesthetics. We used an open-source implementation [17] of the median-cut algorithm [24] to extract the *dominant color* and a palette of the five most representative colors (sorted by hue) from the UI screenshot. To aggregate the extracted colors for each year,

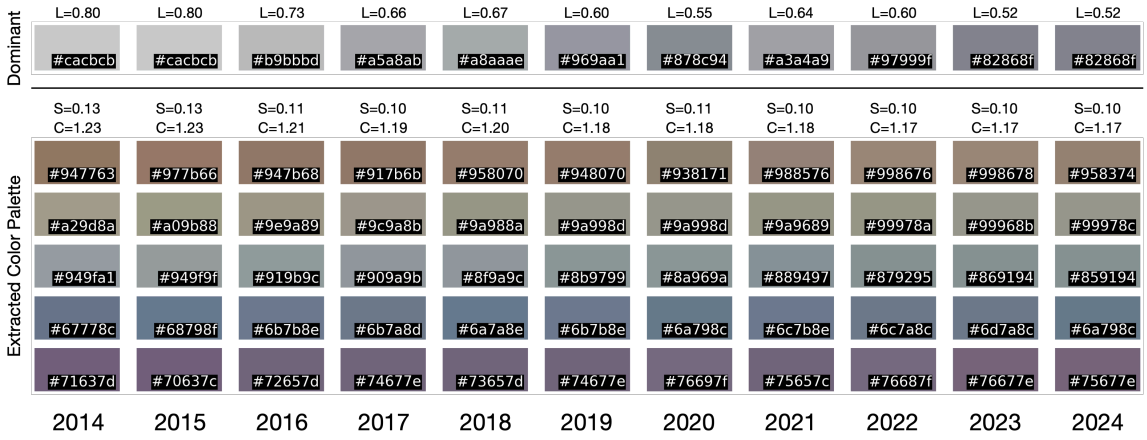


Fig. 5. Color palette trends of the homepages in our dataset. The geometric median of the dominant color in the webpages becomes darker from 2014 to 2020 and from 2021 to 2024 independently. Color palettes from the most dominant 5 colors become more muted and appear more "grey" as it decreases in contrast and saturation.

we computed the *geometric median* of the dominant color and the color palettes.

$$\arg\min_{y \in \mathbb{R}^n} \sum_{i=1}^{m} \|x_i - y\|_2 \tag{1}$$

The geometric median finds a multi-dimensional point that minimizes the Euclidean distance from all given points in a set, and it can be interpreted as the multi-dimensional generalization of the one-dimensional median statistic. We chose this statistic over simple averaging, which would would be influenced by outliers and "blend" together diverse colors. Figure 5 shows the results of our color extraction process, where several trends are visible.

First, from 2014 – 2024, **the dominant color (corresponding to background color) luminance decreased significantly from 0.80 to 0.52, indicating a darker color** ($r = -0.92$, $p < 0.0001$). This trend likely reflects increasing adoption of "dark mode" for websites, which led to more frequent use of dark background colors as opposed to the default white HTML background. Another trend visible from our color analysis is **the increased use of "muted colors."** The use of muted colors have been recommended by some design guides for presenting a more neutral, "comfortable" aesthetic [10], and they are often characterized by low-saturation colors. Calculating the average saturation of extracted color palettes across years confirms this shift. Colors in 2014 had an average saturation of 0.13 and significantly decreased to 0.10 in 2024 ($r = -0.77$, $p < 0.01$).

While these trends in color design reflect evolving aesthetic taste, they could also introduce secondary effects. For example, one possible consequence of custom backgrounds and de-saturation is lower relative contrast between colors on the page. Analysis of each years' data revealed that **average relative contrast of website color palettes also significantly decreased** from 2014 to 2024 ($r = -0.93$, $p < 0.0001$). It is possible that this trend may make some elements harder to visually discern, although this requires specific local context to evaluate.

4.2 Accessibility

The design trends observed over the past decade can significantly impact the usability of UIs, especially for users of assistive technology. We assessed the accessibility of data in *WaybackUI* using Pa11y, an open-source, command-line

accessibility checker [39] designed to measure conformance to the Web Content Accessibility Guidelines (WCAG) 2.0 Level AA standard. Our analysis revealed several trends in the type and frequency of accessibility failures present on popular websites.

4.2.1 Frequency of Failures. We first measured the frequency of accessibility failures in popular webesites over time, adopting an approach similar to that of Fok et al.'s analysis of the image accessibility in Android apps [20]. Unlike this prior work which measures the number of inaccessible elements per screen, we measure the total number of errors detected in the app [53] and do not attempt to de-duplicate errors by element *e.g.,* a single image element can result in both a contrast error and an alt-text error. The results of our frequency analysis are shown in Figure 6.

The median number of detected errors per site fluctuated but did not change significantly from 2014 (27) to 2024 (27). For the years in our analysis, we found a mostly long-tail distribution where the majority of web pages have under 50 errors but some have very high numbers (over 100). In 2014, 68.7% of pages had under 50 accessibility errors while 13.7% of pages had over 100. On the other hand, in 2024, 68.3% of pages had under 50 accessibility errors, and 14.1% had over 100. Both measures fluctuated, but **recent years since 2020 saw an increased number of "highly inaccessible" websites** ($r = 0.96$, $p < 0.01$), and 2024 was the year with the highest percentage. This suggests that a growing number of websites are likely developed with inaccessible tools or practices that result in poor compatibility with assistive technology.

Overall, our findings are similar to those of other analyses of the web [53] and mobile app ecosystems [20, 43, 44], although our dataset confirms these trends over a much longer time period. Previous work also identified a subset of UIs that contain a very high number of apps with accessibility errors, possibly due to low developer awareness. As the WebAIM report states, *"the proportion of pages with fewer errors has increased while the number of pages with many errors has also increased"* [53].

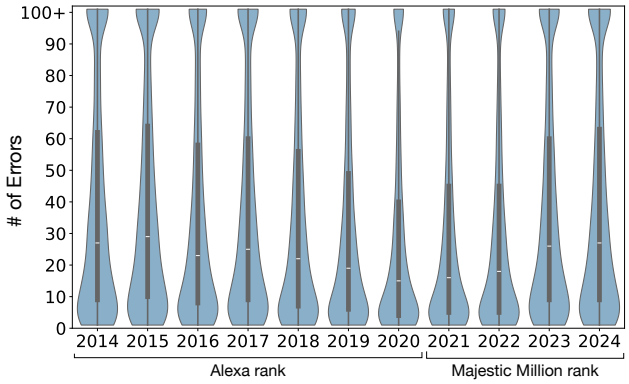


Fig. 6. A violin plot that shows the distribution of websites and automatically detected accessibility errors. There was a higher frequency of webpages with a small amount errors than with a significant amount (100+) of errors. From 2021 to 2024, there was an increase in the number of webpages with 100+ number of errors and the median increases accordingly.

4.2.2 Types of Failures. We further analyzed the most frequent errors WCAG errors found by prior work [53], which include insufficient text contrast, missing image alt-text, broken links, and missing form labels. Some of the frequent errors originally analyzed in the WebAIM report were not tested by the automated Pa11y tool (*e.g.,* including empty button elements), so we exclude them from our analysis. The breakdown of detected errors can be found in Figure 7.

Across our dataset, **insufficient contrast errors made up an increasing majority of detected accessibility errors** ($r = 0.71$, $p < 0.05$), going from 34.9% in 2014 to 40.0% in 2024. This is noteworthy because our design analysis also showed the increasing prevalence of muted color schemes with lower pairwise contrast and saturation. The minimum contrast recommended by WCAG 2 is 4.5:1, although this refers to contrast between foreground (*e.g.*, text) and background colors, rather than all colors on the palette. Guides for choosing accessible colors recommend the use of more vibrant colors [29], which could avoid contrast-related problems.

The three other common types of accessibility errors, either stayed the same (broken links) or decreased (image alt-text, form labels) in overall proportion. Within the "Other" meta-category, duplicate ID errors, which could cause unexpected behavior with JavaScript and assistive technologies, increased from 6.9% of all errors in 2014 to 12.1% in 2024. Note that we analyzed the proportion of these errors and not the absolute number errors in these categories.

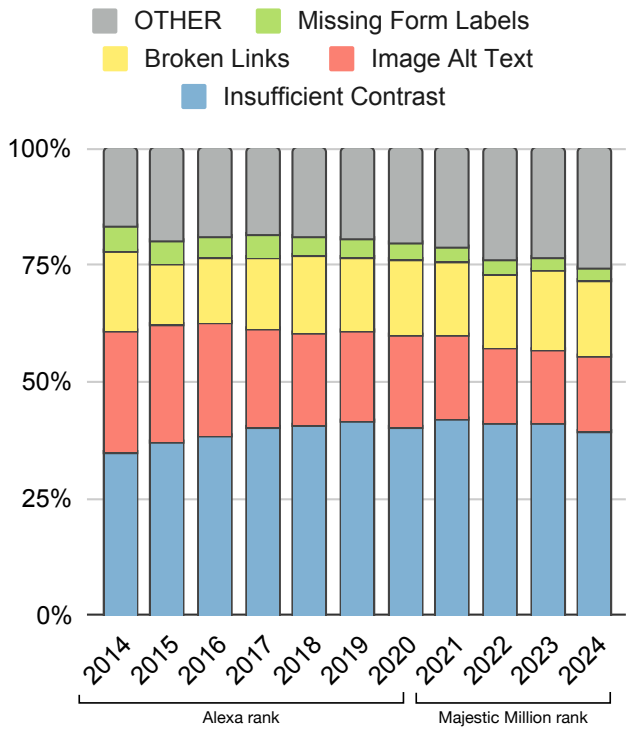


Fig. 7. Relative frequency of accessibility failures from 2014 to 2024. The majority of errors resulted from insufficient contrast between text and background, with the rest was from missing image alt text, broken links, and missing labels on form elements.

4.3 Computational Modeling

Data-driven models for understanding UIs have increasingly found application in accessibility [9, 58], design [8, 14, 15], and software testing [51]. These models are often trained using large datasets such as Rico [14] and AMP [58], which were collected several years ago. Like all machine learning models, changes in their input data, like the visual appearance of UIs, could result in data drift [21], negatively affecting their performance.

In this section, we use *WaybackUI* to analyze the performance characteristics of these models when trained and evaluated on UIs from different years. Specifically, we focus on UI element detectors, a type of model that allows automated systems to understand and operate UIs visually, without relying on developer-defined metadata or APIs.

4.3.1 UI Detection Performance. We first investigated the performance of UI element detection models trained on web data from various years. We conducted our experiments using Detectron2's [57] Faster-RCNN model [41]. We randomly split each year's data into training (80%), validation (10%), and testing (10%) splits. To measure model performance, we adopted the default evaluation settings used the standard COCO benchmark to measure mean average precision (mAP). Note that compared to other work on UI element detection, our models have lower mAP scores because of the noisiness of extracted web data [56] and because the bounding boxes in the evaluation sets are not "tight," since they are extracted from the DOM. The appendix contains some examples of our model predictions on screenshots to qualitatively demonstrate performance.

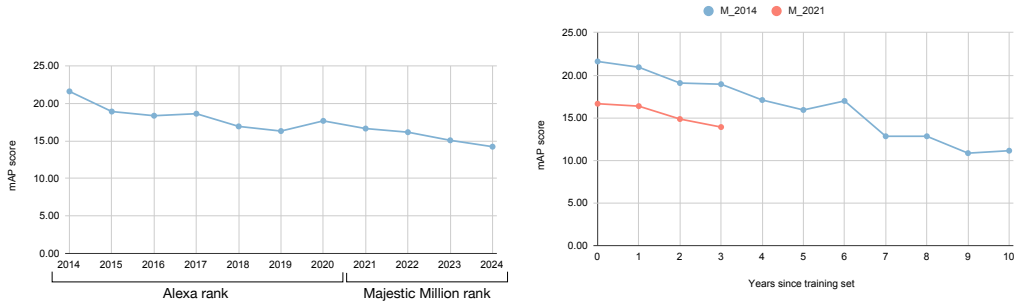


Fig. 8. Left: mAP scores of models trained and evaluated from the same year. Over time, model performance decreases when evaluated on the same dataset it was trained on. Right: mAP scores of model trained on 2014 and evaluated from 2014 to 2024, and of model trained on 2021 and evaluated from 2021 to 2024.

The results of our evaluation are found in Figure 11, where we show **models trained data from more recent years perform worse** ($r = -0.98$, $p < 0.0001$), suggesting that UI data is becoming more difficult for computer vision models to learn. Since our models were trained on labels automatically derived from the page's accessibility tree, data quality could be negatively affected by a growing number of "highly inaccessible" websites with more than 100 detected accessibility errors (Section 4.2.1). Another possibility is that design trends (*e.g.,* minimalism or the use of low-contrast colors) affect visual signifiers traditionally used for some types of interactive elements (*e.g.,* tappable elements) [46, 50, 55]. To investigate this, we generated confusion matrices (available in the appendix) to analyze mis-classifications between element classes. Our findings (Figure 9) confirm that text and images are increasingly confused with their interactive counterparts, suggesting that they are increasingly difficult to distinguish. The F1 score of all interactive elements we tested declined from 2014 – 2024, where the decrease was significant in Text/Hyperlink ($r = -0.98$, $p < 0.0001$), Image/Linked Image ($r = -0.83$, $p < 0.01$), and Text/Button ($r = -0.96$, $p < 0.0001$).

4.3.2 Performance over Time. To investigate this, we analyzed the performance of our 2014 and 2021 models and tracked their performance using the testing splits of subsequent years (Figure 8). We chose these models since they are trained on the first year of data from the Alexa and Majestic Million rankings, respectively. We also chose not to evaluate pre-trained UI element detectors [9, 56] or models trained on existing datasets [8] to better simulate a real-world scenario where data drift occurs from a fixed starting point in time.

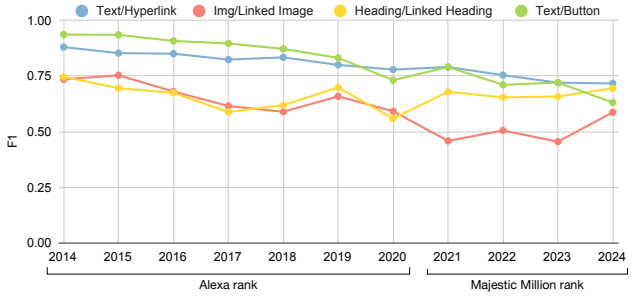


Fig. 9. F1 score of visually-similar elements based on the confusion matrix for the model trained and evaluated on data from 2014

Based on our analysis, we found that **models gradually lose around 5% of their original performance per year.** On average, the 2014 model's performance decreased 1.1% mAP every year, and after 10 years, its performance (11.1% mAP) was only half of its original (21.61% mAP). The 2021 model follows a similar pattern, where it decreased 0.91% mAP every year. This suggests that UI understanding models, such as element detectors, should be updated roughly every 2 years to maintain 90% of their performance on new data.

Among our element categories, the average precision (AP) of the Container class decreased the most from 31.8 in 2014 to only 6.8 in 2024. On the other hand, classes such as Images, Linked Images, and Paragraphs saw smaller absolute decline, under 5 AP during the same period. Overall, the amount of degradation fluctuates as well, with 2018 seeing the greatest drop in performance. The rate of degradation could be influenced by numerous factors, such as the release of new design trends, development libraries, and changes in standards. Tracking these influences could help inform more timely updates of models by focusing on rapidly changing apps or websites and possibly forecast future trends [25].

5 Discussion

In our paper, we demonstrated the utility of *WaybackUI* by conducting longitudinal analyses targeted towards *i)* design, *ii)* accessibility, and *iii)* computational modeling.

In each category, we present several findings summarized below.

- Design
  - Layouts changed to reflect responsive design practices and content-related design patterns.
  - Color palettes became more muted (lower saturation) and exhibited lower relative contrast.
- Accessibility
  - Recently (since 2020), the number of "highly inaccessible" websites with over 100 detected errors has been increasing.
  - Insufficient contrast errors make up an increasing majority of automatically detected accessibility errors.
- Modeling
  - Data from more recent years are more difficult for UI element detectors to learn, particularly in distinguishing interactive and non-interactive versions of elements.
  - UI element detectors lose around 5% of their original performance per year.

In this section, we discuss the implications of our findings and directions for future work.

5.1 Understanding and Discovering Trends

The primary goal of our work was to enable data-driven analysis of UI trends and their effects, and we demonstrated the utility of the *WaybackUI* dataset through three types of analysis.

Some of the analyses that we conducted followed methodologies from prior work. Our goal was to verify their findings across a longer time span [20, 30, 53], larger dataset [2, 20], or new UI domain [20]. For example, our first design analysis focused on understanding the how designers choose to arrange content in UIs by analyzing web layouts. We repeated the spatial distribution analysis conducted by Kumar et al. to analyze the relationship between element position and semantics. Some of our findings were similar, such as placement tendencies of certain element types (*e.g.,* logos at the top left corner), but we also provided a deeper analysis in how placements and layouts have been affected by larger technological trends (*e.g.,* mobile and responsive design), leading to the decline of page margins and new uses of heading and container elements. In our second analysis, we followed prior methodologies [20, 53] to track the accessibility conformance of web UIs. Previously, these longitudinal analyses, which have typically focused on the last 5 years, found that there are a growing number of "highly inaccessible" websites [53] and UIs with significantly more detected errors. For example, Fok et al. found that around 20% of apps had failures in more than 95% of their images [20] and the WebAIM Million analysis from 2019 – 2024 found that *"pages with many errors have gotten worse"* [53]. Our analysis presents similar findings for recent years, where we found that the number of websites with over 100 errors detected by the Pa11y accessibility scanner has been increasing since 2020 ($r = 0.96$, $p < 0.01$). However, further zooming out using our data reveals that in the first half of the decade (2014 – 2019), the number of these websites was slowly declining ($r = -0.81$, $p < 0.05$). This is promising because it *i)* highlights a relatively recent period in which one form of large-scale accessibility improvement has occurred and *ii)* could provide data to diagnose and address these challenges.

In addition to new insights gained from replicating previous work, we used *WaybackUI* to conduct several new types of analyses, afforded by our dataset's scale and rich visual and semantic information. Our third modeling analysis investigated the effect on evolving UI data on training and evaluating visual UI understanding models. To our knowledge, this type of analysis has not been performed previously, possibly due to the lack of historical UI data for training deep learning models. Our experiments led to two findings. First, we trained UI element detectors on UIs from different years in our dataset and found that more recent data were more difficult to learn, *i.e.,* led to lower performance on the same year's testing set. We also evaluated how models trained on outdated UI datasets perform over time, *e.g.,* how accurate would a model trained on UIs from 2018 perform with today's designs? We conducted a simulated "back-testing" experiment [21] and found that UI element detection models lose around 5% of their original performance every year after they are trained. This result suggests that to maintain the performance of automated systems that depend on them, models should be retrained roughly every two years to maintain 90% of their original performance.

To summarize, we discussed how *WaybackUI* facilitates both the understanding and discovery of trends. We showed how analyzing known trends using existing methodologies using our dataset led to more nuanced, longer-term insights related to UI layout patterns and accessibility errors. At the same time, the scale and rich visual and semantic data in *WaybackUI* supports the discovery and quantification of trends, like the gradual degradation of visual UI model performance.

5.2 Cascading Effects of Design Choices

How are trends in design, accessibility, and modeling performance related to each other? Various aspects of UI evolution has previously studied (*e.g.,* design [13, 16], accessibility [20, 53]); however, our work is unique in that multiple analyses were performed on a unified dataset, which presents opportunities for understanding the interrelated nature of these trends. We draw connections between our different analyses to show how UI design choices can trigger cascading effects in usability and other down-stream applications.

We first investigated the choice of color, which are influenced by a wide variety of aesthetic (*e.g.,* minimalism) and cultural factors [25] and can directly impact usability. Our initial design analysis quantified how the color palettes of UIs changed over the past decade and led to two findings. We found that *i)* the dominant color (likely corresponding to background color) of websites significantly decreased, suggesting the increased use of dark background colors and *ii)* color palettes became more muted with significantly decreased saturation and relative contrast (Figure 5). While these changes could reflect evolving aesthetic taste, they also introduce secondary affects, such as making UIs more difficult to discern. In our subsequent accessibility analysis, we used an automated tool to check WCAG 2.0 compliance and found a significantly increasing number of websites had insufficient contrast errors during the same period of time (Figure 7). We found a statistically significant correlation between the saturation and proportion of contrast errors ($r = -0.87$, $p < 0.001$) which suggests a possible underlying relationship. Finally, our UI modeling experiments revealed, among things, that data from recent years were more difficult for computer vision models to learn. These findings are likely related to color choice in UIs, which also affects how elements can be visually distinguished. We found a statistically significant correlation between the saturation and per-year model performance ($r = 0.78$, $p < 0.01$), suggesting a possible underlying relationship between them. Specifically, we note that the detection performance on interactive elements decreased significantly (Figure 9), which also rely on visual signifiers (*e.g.,* color contrast to denote clickability).

In addition, the toolkits and libraries used to implement designs can also have significant downstream effects. While our accessibility analysis did not attempt to identify the toolkit or authoring software behind websites in our dataset, previous research has shown that some popular web toolkits result in a significantly higher number of accessibility errors on average [53]. From our analysis, we found an increasing number of "highly inaccessible" websites with over 100 detected errors (more than 3 times the median number of errors), which could negatively impact assistive technology and machine learning systems that use programmatically-extracted accessibility metadata [34, 56]

Given the complexity UI authoring and use, it is difficult to create the controlled conditions necessary to establish exact causal relations [12]; however, our analyses reveals several patterns in design, accessibility, and modeling performance emerge. As previously discussed, our approaches and dataset could be used to aid in understanding why trends occur and potentially help address negative ones (*e.g.,* increasing accessibility errors). Finally, our analyses underscores the importance of good UI design and implementation practices, as they can have significant, lasting effects on how users and automated systems are able to interface with them.

5.3 Limitations & Future Work

We see several avenues to improve the *WaybackUI* dataset and build on the longitudinal analyses presented in this paper. We discuss limitations and future work for both our dataset and longitudinal analyses.

5.3.1 WaybackUI Dataset. The *WaybackUI* dataset can be improved in several ways. While automated crawling and metadata extraction enabled the construction of our large-scale dataset, they also present drawbacks.

First, automated crawlers are limited in the kinds of screens that they can discover (*e.g.,* bypassing a login screen), and this likely affected the distribution of UIs in our dataset. Previous work noted that web UIs often contain various sources of noise (*e.g.,* mismatched metadata, "ghost annotations"), they could be useful in conjunction with a small number of high-quality human annotations in transfer learning settings [56]. Since we do not use human annotations or checking, it is likely that these noisy labels limited the performance UI detectors. Other sources of noise specific to historical web data include errors in WARC playback, missing browser plugins (*e.g.,* Adobe Flash), and deprecated APIs from websites designed for older browsers. Future work could apply techniques for automated [33] or human-driven [32] filtering to improve the quality of our dataset. Additional human annotations [58] and feedback [40] would allow for much deeper analysis and understanding UI trends.

Finally, while *WaybackUI* is the largest *longitudinal* dataset (to the best of our knowledge), it is smaller than that of other UI datasets (e.g. 400k for WebUI dataset [56] and 70k for RICO [14]). Since all our crawling focused on a single website (the Wayback Machine), this imposed constraints on the duration of data collection (*e.g.,* due to host-side throttling), as we sought to avoid placing excessive load on their servers. Future work could expand *WaybackUI* using other data archives, real-time collections, and expand to new UI platforms (*e.g.,* mobile apps).

5.3.2 Longitudinal Analyses. The analyses presented in this paper are primarily meant to demonstrate the utility of *WaybackUI* and only investigate a subset of the potential applications for longitudinal analysis. Specifically, we focused primarily on ones that could be completed with automated tools and without additional human annotations. This limited in the type of phenomena that could be analyzed. For example, we used Pa11y to automatically scan websites for accessibility errors, but it is unable to detect many types of accessibility errors and barriers faced by users. Since Pa11y operates by checking properties of the accessibility tree, it is difficult to detect if there are any errors with the accessibility tree itself, *e.g.,* if an element is visually rendered but not appropriately represented in the accessibility metadata. Future analyses using *WaybackUI* could use human annotations of UI elements to more reliably measure the quality of web accessibility trees by computing the agreement between these two sources of data [58].

Finally, incorporating feedback from users and experts not only expands the types of analyses possible, it could also help explain and improve understanding of results from data-driven analyses. For example, while we successfully measured changes in the layout and color and hypothesized effects on usability, we lacked direct feedback and preferences that could establish a causal connection [40]. Similarly, while our results led us to hypothesize an increasing difficulty in discerning semantically-important visual signifiers, incorporating human insights could validate or refine this hypothesis by identifying specific design elements that contribute to the decline in model performance. In summary, the statistical analysis (*i.e.,* correlation testing) we performed could help identify promising directions and inform future studies that more deeply investigate these relationships.

6 Conclusion

In this paper, we introduced the *WaybackUI* dataset, which consists of nearly 100,000 web UIs mined from 11 years of (2014 – 2024, inclusive) of web archives. The large scale of data and inclusion of rich visual and semantic information in *WaybackUI* facilitates comprehensive analysis of UI evolution and design trends over the past decade. In this paper, we used our data to analyze how gradual UI on the web changes impacted *i)* visual, *ii)* accessibility, and *iii)* automated systems that interact with UIs. We present several findings in each category. In our design analysis, we found increased use of responsive layout patterns and a growing preference for "muted" color palettes. In our accessibility analysis, we found an increasing number of contrast-related accessibility errors and "highly inaccessible" websites with over 100

detected errors. Finally, in our analysis of UI models' temporal performance, we found that more recent UI designs may be harder to learn and that element detectors lose around 5% of their original performance each year after they are trained. In summary, we presented examples of how *WaybackUI's* scale and richness enables data-driven discovery and understanding of how UIs change over time and believe it can serve as the foundation for future longitudinal analysis of UIs.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Vibhor Agarwal and Nishanth Sastry. 2022. "Way back then": A Data-driven View of 25+ years of Web Evolution. In *Proceedings of the ACM Web Conference 2022*. 3471–3479.

[3] Alexa Internet, Inc. [n. d.]. Alexa Website Rankings. https://www.alexa.com/. Service retired in May 2022. Accessed legacy data on September 7, 2024.

[4] Khalid Alharbi and Tom Yeh. 2015. Collect, decompile, extract, stats, and diff: Mining design pattern changes in Android apps. In *Proceedings of the 17th international conference on human-computer interaction with mobile devices and services*. 515–524.

[5] Sanjay K Arora, Yin Li, Jan Youtie, and Philip Shapira. 2016. Using the wayback machine to mine websites in the social sciences: A methodological resource. *Journal of the Association for Information Science and Technology* 67, 8 (2016), 1904–1915.

[6] Ibtehal S Baazeem and Hend S Al-Khalifa. 2015. Advancements in web accessibility evaluation methods: how far are we?. In *Proceedings of the 17th International Conference on Information Integration and Web-based Applications & Services*. 1–5.

[7] Jeffrey P Bigham. 2014. Making the web easier to see with opportunistic accessibility improvement. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. 117–122.

[8] Sara Bunian, Kai Li, Chaima Jemmali, Casper Harteveld, Yun Fu, and Magy Seif Seif El-Nasr. 2021. Vins: Visual search for mobile user interface design. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–14.

[9] Jieshan Chen, Mulong Xie, Zhenchang Xing, Chunyang Chen, Xiwei Xu, Liming Zhu, and Guoqiang Li. 2020. Object detection for graphical user interface: Old fashioned or deep learning or a combination?. In *proceedings of the 28th ACM joint meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 1202–1214.

[10] Shaurya Chhabra. 2024. Muted Colors in UI Design: A Comprehensive Guide. https://www.onething.design/blogs/muted-colors/ Accessed: 2024-09-07.

[11] Angela L Curl and Deborah D Bowers. 2009. A longitudinal study of website accessibility: Have social work education websites become more accessible? *Journal of Technology in Human Services* 27, 2 (2009), 93–105.

[12] Angus Deaton and Nancy Cartwright. 2018. Understanding and misunderstanding randomized controlled trials. *Social science & medicine* 210 (2018), 2–21.

[13] Biplab Deka, Bardia Doosti, Forrest Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Ranjitha Kumar, Tao Dong, and Jeffrey Nichols. 2021. An early rico retrospective: three years of uses for a mobile app dataset. *Artificial Intelligence for Human Computer Interaction: A Modern Approach* (2021), 229–256.

[14] Biplab Deka, Zifeng Huang, Chad Franzen, Joshua Hibschman, Daniel Afergan, Yang Li, Jeffrey Nichols, and Ranjitha Kumar. 2017. Rico: A mobile app dataset for building data-driven design applications. In *Proceedings of the 30th annual ACM symposium on user interface software and technology*. 845–854.

[15] Biplab Deka, Zifeng Huang, and Ranjitha Kumar. 2016. ERICA: Interaction mining mobile apps. In *Proceedings of the 29th annual symposium on user interface software and technology*. 767–776.

[16] Bardia Doosti, Tao Dong, Biplab Deka, and Jeffrey Nichols. 2018. A computational method for evaluating UI patterns. *arXiv preprint arXiv:1807.04191* (2018).

[17] Shipeng Feng. 2019. color-thief-py. https://github.com/fengsp/color-thief-py. GitHub repository.

[18] Sidong Feng, Suyu Ma, Han Wang, David Kong, and Chunyang Chen. 2024. MUD: Towards a Large-Scale and Noise-Filtered UI Dataset for Modern Style UI Modeling. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–14.

[19] Thoriq Firdaus. 2014. *Responsive Web Design by Example: Beginner's Guide*. Packt Publishing Ltd.

[20] Raymond Fok, Mingyuan Zhong, Anne Spencer Ross, James Fogarty, and Jacob O Wobbrock. 2022. A large-scale longitudinal analysis of missing label accessibility failures in android apps. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–16.

[21] Saurabh Garg, Mehrdad Farajtabar, Hadi Pouransari, Raviteja Vemulapalli, Sachin Mehta, Oncel Tuzel, Vaishaal Shankar, and Fartash Faghri. 2023. Tic-clip: Continual training of clip models. *arXiv preprint arXiv:2310.16226* (2023).

[22] Stephanie Hackett, Bambang Parmanto, and Xiaoming Zeng. 2005. A retrospective look at website accessibility over time. *Behaviour & Information Technology* 24, 6 (2005), 407–417.

[23] Vicki L Hanson and John T Richards. 2013. Progress on website accessibility? *ACM Transactions on the Web (TWEB)* 7, 1 (2013), 1–30.

[24] Paul Heckbert. 1982. Color image quantization for frame buffer display. *ACM Siggraph Computer Graphics* 16, 3 (1982), 297–307.
[25] M Hidefi. 2017. Understanding and forecasting colour trends in design. In *Colour Design*. Elsevier, 285–298.
[26] Interactive Advertising Bureau (IAB). 2023. Content Taxonomy 3.0. https://iabtechlab.com/standards/content-taxonomy/. Accessed September 7, 2024.
[27] Internet Archive. 2024. Wayback Machine. https://archive.org/web/. Accessed: 2024-08-26.
[28] Leonard R Kasday. 2000. A tool to evaluate universal Web accessibility. In *Proceedings on the 2000 conference on Universal Usability*. 161–162.
[29] Daryl Koopersmith and Wilson Miner. 2019. Accessible color systems. https://stripe.com/blog/accessible-color-systems Accessed: 2024-09-07.
[30] Ranjitha Kumar, Arvind Satyanarayan, Cesar Torres, Maxine Lim, Salman Ahmad, Scott R Klemmer, and Jerry O Talton. 2013. Webzeitgeist: design mining the web. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 3083–3092.
[31] Jonathan Lazar and Kisha-Dawn Greenidge. 2006. One year older, but not necessarily wiser: an evaluation of homepage accessibility problems over time. *Universal Access in the Information Society* 4 (2006), 285–291.
[32] Luis A Leiva, Asutosh Hota, and Antti Oulasvirta. 2020. Enrico: A dataset for topic modeling of mobile UI designs. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services*. 1–4.
[33] Gang Li, Gilles Baechler, Manuel Tragut, and Yang Li. 2022. Learning to denoise raw mobile UI layouts for improving datasets at scale. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–13.
[34] Toby Jia-Jun Li, Amos Azaria, and Brad A Myers. 2017. SUGILITE: creating multimodal smartphone automation by demonstration. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. 6038–6049.
[35] Paul Pu Liang, Yiwei Lyu, Xiang Fan, Zetian Wu, Yun Cheng, Jason Wu, Leslie Chen, Peter Wu, Michelle A Lee, Yuke Zhu, et al. 2021. Multibench: Multiscale benchmarks for multimodal representation learning. *Advances in neural information processing systems* 2021, DB1 (2021), 1.
[36] Thomas F Liu, Mark Craft, Jason Situ, Ersin Yumer, Radomir Mech, and Ranjitha Kumar. 2018. Learning design semantics for mobile apps. In *Proceedings of the 31st Annual ACM Symposium on User Interface Software and Technology*. 569–579.
[37] Majestic. [n. d.]. Majestic Million: Top Million Websites. https://majestic.com/reports/majestic-million. Accessed September 7, 2024.
[38] Jessica Ogden, Edward Summers, and Shawn Walker. 2024. Know (ing) Infrastructure: The Wayback Machine as object and instrument of digital research. *Convergence* 30, 1 (2024), 167–189.
[39] Pa11y Project. 2024. Pa11y: Your Automated Accessibility Testing Toolkit. https://pa11y.org/. Accessed: 2024-08-26.
[40] Katharina Reinecke and Krzysztof Z. Gajos. 2014. Quantifying visual preferences around the world. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 11–20. https://doi.org/10.1145/2556288.2557052
[41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems* 28 (2015).
[42] John T Richards, Kyle Montague, and Vicki L Hanson. 2012. Web accessibility as a side effect. In *Proceedings of the 14th international ACM SIGACCESS conference on Computers and accessibility*. 79–86.
[43] Anne Spencer Ross, Xiaoyi Zhang, James Fogarty, and Jacob O Wobbrock. 2018. Examining image-based button labeling for accessibility in Android apps through large-scale analysis. In *Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility*. 119–130.
[44] Anne Spencer Ross, Xiaoyi Zhang, James Fogarty, and Jacob O Wobbrock. 2020. An epidemiology-inspired large-scale analysis of android app accessibility. *ACM Transactions on Accessible Computing (TACCESS)* 13, 1 (2020), 1–36.
[45] Alireza Sahami Shirazi, Niels Henze, Albrecht Schmidt, Robin Goldberg, Benjamin Schmidt, and Hansjörg Schmauder. 2013. Insights into layout patterns of mobile user interfaces by an automatic analysis of android apps. In *Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems*. 275–284.
[46] Eldon Schoop, Xin Zhou, Gang Li, Zhourong Chen, Bjoern Hartmann, and Yang Li. 2022. Predicting and explaining mobile ui tappability with vision modeling and saliency analysis. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–21.
[47] StatCounter. 2016. Mobile and tablet internet usage exceeds desktop for first time worldwide. https://gs.statcounter.com/press/mobile-and-tablet-internet-usage-exceeds-desktop-for-first-time-worldwide Published on Tuesday 1st November 2016.
[48] Statcounter GlobalStats. 2023. Desktop Screen Resolution Stats Worldwide. https://gs.statcounter.com/screen-resolution-stats/desktop/worldwide/2023. Accessed: 2024-08-26.
[49] Aixin Sun, Ee-Peng Lim, and Wee-Keong Ng. 2002. Web classification using support vector machine. In *Proceedings of the 4th international workshop on Web information and data management*. 96–99.
[50] Amanda Swearngin and Yang Li. 2019. Modeling mobile interface tappability using crowdsourcing and deep learning. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–11.
[51] Amanda Swearngin, Jason Wu, Xiaoyi Zhang, Esteban Gomez, Jen Coughenour, Rachel Stukenborg, Bhavya Garg, Greg Hughes, Adriana Hilliard, Jeffrey P Bigham, et al. 2024. Towards Automated Accessibility Report Generation for Mobile Apps. *ACM Transactions on Computer-Human Interaction* (2024).
[52] W3C. 2014. HTML5. https://www.w3.org/TR/2014/REC-html5-20141028/ W3C Recommendation, 28 October 2014.
[53] WebAIM. 2024. The WebAIM Million: An Accessibility Analysis of the Top 1,000,000 Home Pages. https://webaim.org/projects/million/. Accessed: 2024-08-26.

[54] World Wide Web Consortium (W3C). 2018. Web Content Accessibility Guidelines (WCAG). https://www.w3.org/WAI/standards-guidelines/wcag/. Accessed: 2024-08-26.
[55] Jason Wu, Rebecca Krosnick, Eldon Schoop, Amanda Swearngin, Jeffrey P Bigham, and Jeffrey Nichols. 2023. Never-ending learning of user interfaces. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*. 1–13.
[56] Jason Wu, Siyan Wang, Siman Shen, Yi-Hao Peng, Jeffrey Nichols, and Jeffrey P Bigham. 2023. Webui: A dataset for enhancing visual ui understanding with web semantics. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–14.
[57] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. 2019. Detectron2. https://github.com/facebookresearch/detectron2.
[58] Xiaoyi Zhang, Lilian De Greef, Amanda Swearngin, Samuel White, Kyle Murray, Lisa Yu, Qi Shan, Jeffrey Nichols, Jason Wu, Chris Fleizach, et al. 2021. Screen recognition: Creating accessibility metadata for mobile applications from pixels. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.

7 Appendix

7.1 Spatial Distribution Figures

We provide a complete set of spatial distribution maps (Figure 10)to supplement the example in our paper (Figure 4). Each map in the grid displays the positional probability of an element type from our dataset in a specific year. Note that years 2014 to 2020 contain crawled data based on Alexa rankings and is captured with 1366x768 resolution. Data from years 2021 to 2024 are based on Majestic Million rankings and captured with 1920x1080 resolution.

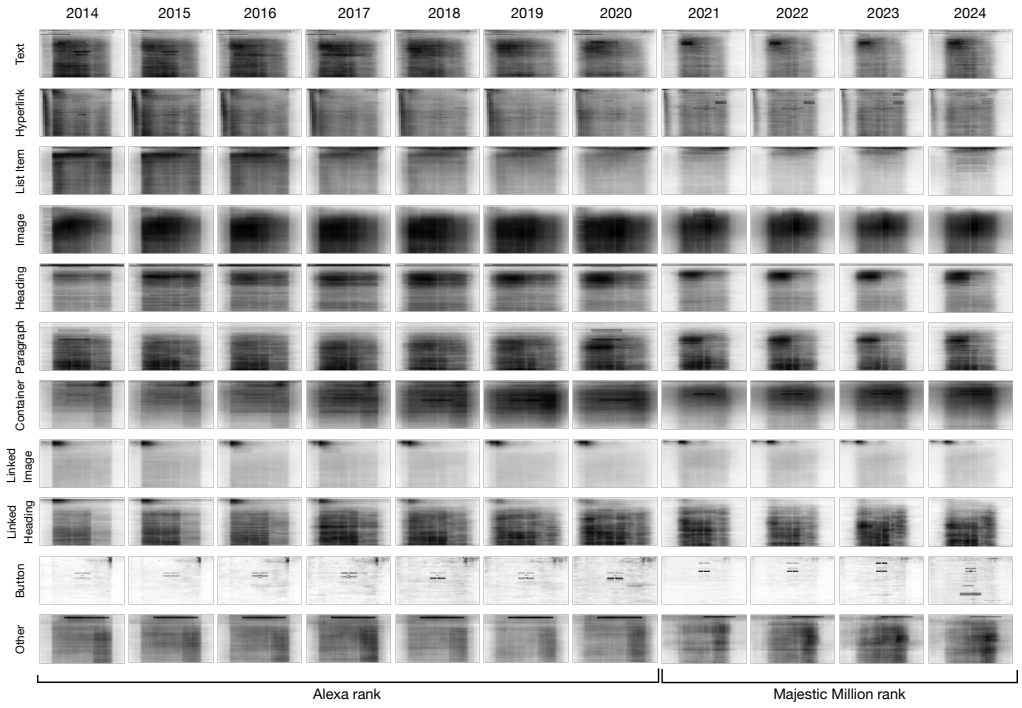


Fig. 10. Spatial distribution maps of each element type from 2014 to 2024. We observed that *(i)* the content width of pages gradually increased from elements not just mentioned in our paper, but for a majority of all element types, *(ii)* Text annotations seemed to concentrate towards the center of the viewport screen, suggesting a supplementary function to the "call to action" headings, and *(iii)* navigational elements like Hyperlink and List Item transitioned from spanning across the page to concentrating in one area of the screen, such as to the left and top of the screen, respectively.

7.2 Confusion Matrices

We supplement our F1 calculations of the interactive elements from our paper (Figure 9) by providing the confusion matrices of models trained and evaluated on the dataset of the same year (Figure 11). Each cell is annotated with the percentage of predictions across all labels for a particular ground truth class. Overall, we observed that static and dynamic elements of similar functionality had a higher chance of misclassification with the other element.

7.3 Model Detection Examples

To supplement the mAP scores provided in our paper (Figure 8), we provide some examples of our model predictions to demonstrate their performance (Figure 12). Notably, the annotations were extracted from noisy web data, thus leading to missing bounding boxes in our dataset or misclassifications of some elements.

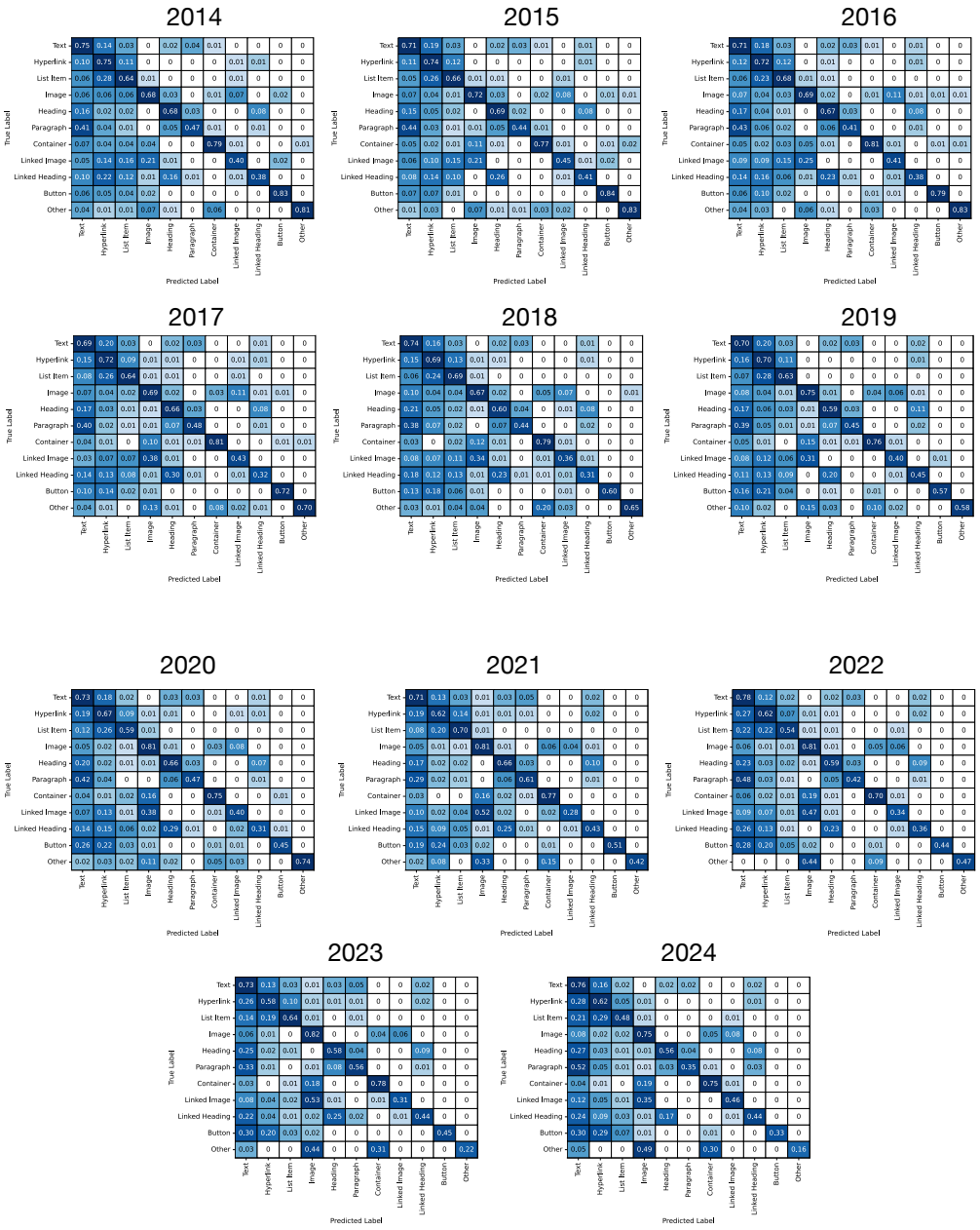


Fig. 11. From top to bottom, left to right: confusion matrices for models trained and evaluated on the same dataset of years 2014-2024.

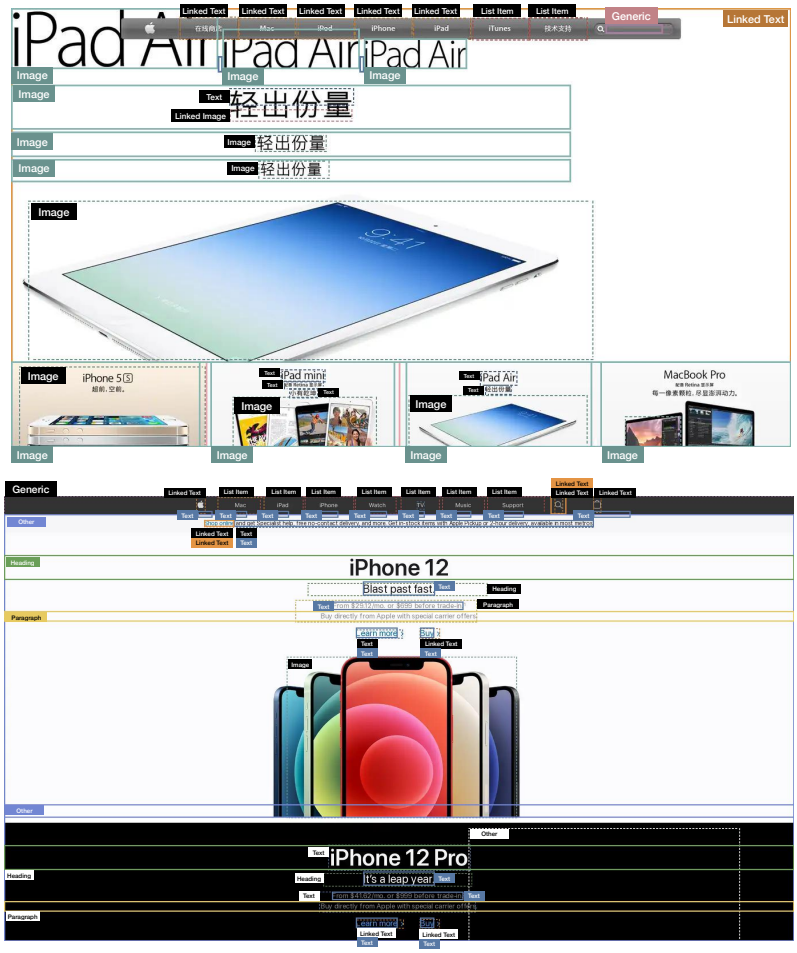


Fig. 12. **Top:** Predictions from the model trained on data from 2014 and evaluated on `www.apple.com.cn` from 2014. Ground truth boxes are annotated with solid lines and labeled with the corresponding box color. Predictions are dashed boxes with solid black labels. Missing labels in the html led to unidentified elements in the navigation bar at the top. However, the model still predicted these elements. This mismatch is representative of other examples, which is a possible explanation for our lower mAP scores compared to other baseline models.
Bottom: Predictions from the model trained on data from 2014 and evaluated on `www.apple.com` from 2021. Ground truth boxes are annotated with solid lines and labeled with the corresponding box color. Predictions are dashed boxes with solid black and white labels. While the resolution from the screenshots in the training and test set differed, the model still performs decently well. However, there are some mislabeling of text-based elements, where this is some confusion between Text, Linked Text, and Headings.