BiasField: Interactive Bias Probing of Machine Learning Datasets

Category: Research

Paper Type: application/design study

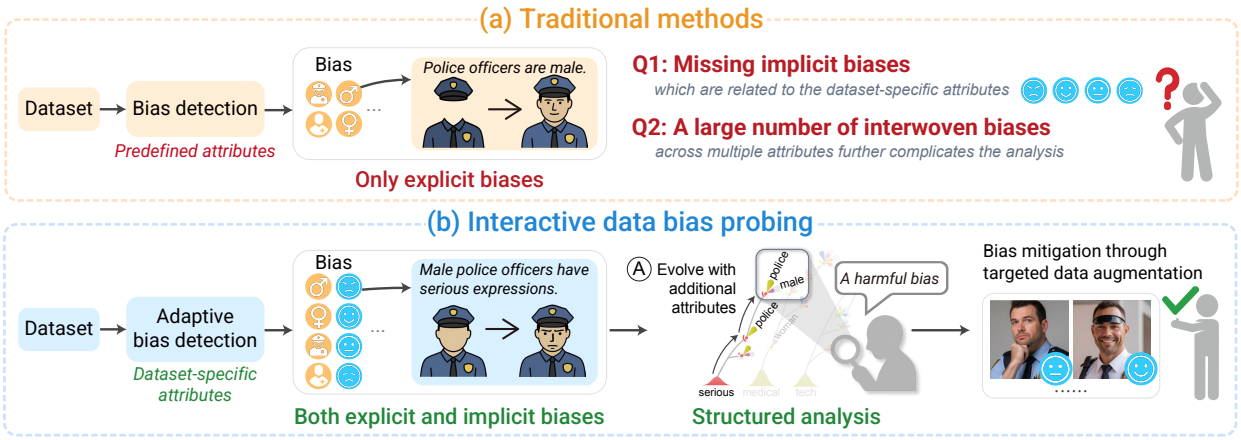

Fig. 1: The comparison between the traditional methods and our interactive data bias probing method: (a) traditional methods focus on explicit biases, overlooking implicit biases with dataset-specific attributes; (b) our method adaptively detects both explicit and implicit biases, and uses a plant-growth-based visualization to facilitate structured analysis and bias mitigation.

Abstract— Bias in machine learning datasets occurs when certain attributes are unfairly associated, *e.g.*, serious males being mostly linked with law enforcement officers in job-related image datasets. Training models on biased datasets will degrade model performance and lead to fairness issues, particularly for underrepresented groups. Existing bias detection methods mainly focus on explicit biases associated with predefined attributes (*e.g.*, gender and ethnicity) while overlooking implicit biases associated with subtler, dataset-specific attributes (*e.g.*, facial expressions and attire). To address this gap, we present BiasField, an interactive tool that offers a closed-loop workflow for detecting, analyzing, and mitigating bias. Central to BiasField is the adaptive detection of both explicit and implicit biases, a process facilitated by the automatic extraction of the dataset-specific attributes. It then employs a plant-growth metaphor to visualize these biases, enabling structured analysis to identify similar biases and track how they strengthen with additional attributes. Finally, confirmed biases are mitigated through targeted generative data augmentation. A user study, two case studies, and an expert study are conducted to demonstrate its capability to detect, analyze, and mitigate complex biases.

Index Terms— Data bias, metaphor design, interactive visualization

1 Introduction

Machine learning (ML) models, especially large foundation models (*e.g.*, ChatGPT [1], LLaMA [21]), have achieved remarkable success across multiple domains [7,38,62]. However, their performance heavily relies on the representativeness of the training datasets. If these datasets do not adequately represent certain groups of the target population, the trained models may not generalize effectively to the underrepresented groups. This phenomenon, known as *data bias*, typically manifests through spurious associations between specific attributes. For example, in a portrait image dataset [39], male images comprise roughly 61.9% overall but over 89.6% for the subset of "police officer." Similarly, in a story dataset [40], women are often depicted in family stories while men are associated with stories about politics and war. If left unaddressed, these biases can degrade model performance and raise fairness concerns, especially for marginalized or underrepresented groups.

Recognizing the harms caused by these biases, several methods have been developed to detect them [49]. However, these methods mainly focus on explicit biases associated with predefined attributes (*e.g.*, gender and ethnicity). This limits their ability to detect implicit biases associated with dataset-specific attributes that are less obvious yet important (*e.g.*, facial expressions, attire, and background). Analyzing these implicit biases poses two challenges. First, many ML datasets contain unstructured data, such as images or text, which lack explicit attributes for bias detection and analysis. Manual identification of relevant attributes often suffers from blind spots, potentially overlooking important attributes and consequently failing to detect the associated implicit biases (Fig. 1Q1). Second, a large number of detected biases across multiple attributes further complicates the analysis. Without a clear understanding of what context these biases arise and how they interact, users struggle to identify those that warrant mitigation (Fig. 1Q2).

To address these two challenges, we develop BiasField, an interactive tool designed to facilitate bias analysis and mitigation in ML datasets. It consists of two main parts: an adaptive bias detection method and a bias visualization. The detection method automatically extracts dataset-specific attributes and identifies both explicit and implicit biases by identifying sample subsets where certain attribute values are overrepresented (e.g., "serious" appearing more often in the subset of "male police officers"). This process can produce a large number of interwoven biases, obscuring the causes of their occurrences and complicating verification. To facilitate bias verification, the visualization employs a plant-growth metaphor to illustrate how different attributes combine and contribute to specific biases. As shown in Fig. 1Ⓐ, the key feature of this visual metaphor is that it allows users to easily track how biases evolve with additional attributes and quickly identify biases that require mitigation. Once identified, generative models are employed to mitigate biases through targeted data augmentation.

We evaluate BiasField through a user study, two case studies, and an expert study. Together, these evaluations demonstrate BiasField's capability to detect, analyze, and mitigate complex biases often overlooked by traditional methods. In summary, the main contributions are:

- An adaptive bias detection method that detects both explicit and implicit biases.

- A plant-growth-based visualization for exploring a large number of interwoven biases and the associated attributes.
- An interactive tool for structured analysis and mitigation of bias in ML datasets.

2 Related Work

2.1 Bias Detection

Bias in machine learning is commonly categorized into algorithmic bias and data bias [45]. **Algorithmic bias** stems from design choices in machine learning algorithms that lead to biased outcomes [6, 14, 16, 19]. **Data bias**, on the other hand, manifests through spurious associations in training data, which aligns with our work. Initial efforts on data bias detection analyze attribute frequencies and then compare their differences across data groups [9, 39, 50]. For example, Zhao *et al*. [64] showed that in the MS-COCO dataset, kitchen objects like "knife" and "fork" had strong associations with women. These spurious associations caused models to rely on stereotypical object co-occurrences rather than visual evidence when predicting gender. Beyond frequency calculation on individual attributes, Borenstein *et al*. [8] detected intersectional bias involving multiple attributes. They employed Pointwise Mutual Information (PMI) to measure the pairwise association in historical Caribbean newspapers. These methods typically target explicit biases defined over predefined attributes and may overlook implicit biases involving dataset-specific attributes, limiting their applicability. Compared with these methods, we introduce an adaptive bias detection method that automatically extracts dataset-specific attributes from any given dataset and then identifies potential biases based on these attributes.

2.2 Visual Dataset Analysis

Existing efforts on visual dataset analysis can be divided into two groups: sample-level and dataset-level [38, 54]. Sample-level methods focus on identifying and correcting anomalies in individual samples, such as missing values and incorrect labels [3, 11, 56, 59], while dataset-level methods focus on understanding and analyzing the overall distributions and patterns within the datasets [12, 34, 61, 63]. Our method aligns with the research efforts on the dataset level, which typically include two classes: visual distribution analysis and visual bias analysis.

Visual distribution analysis. Distribution analysis is widely used to diagnose coverage gaps and shifts, thereby improving dataset quality [38, 62]. Some distribution issues arise during the training phase, such as out-of-distribution (OoD) samples that are not covered by the training distribution. For instance, OoDAnalyzer [12] combines ensemble detection with grid-based visualization to analyze OoD image samples. VATLD [27] leverages disentangled representation learning to model the data space of traffic light images and reveal coverage gaps. Song *et al*. [51] extended this line of work to text data. Other issues arise during the deployment phase, such as concept drift, where data distribution evolves over time. To tackle this, Yang *et al*. [61] presented DriftVis for detecting and addressing drift in streaming data, while Wang *et al*. [57] discovered drifts in multi-source time-series data. In contrast, our work focuses on analyzing data bias caused by spurious associations between specific attributes.

Visual bias analysis. A variety of visual analysis methods support interactive investigation of biases. Many of them aim to address algorithmic bias, such as What-If Tool [58], FairVis [10], FairSight [2], WordBias [24], DiscriLens [55], and D-BIAS [25].

Others focus on visually analyzing data biases that originate from datasets themselves. For example, Silva [60] enables users to diagnose sources of bias by visualizing causality relationships between attributes in a node-link diagram. Know Your Data (KYD) [26] uses a matrix diagram, where rows represent sample subsets, columns represent certain attributes of interest, and the color of each cell indicates whether the attribute appears with higher or lower frequency in that particular subset. STILE [33] employs a chord diagram, where arcs represent different attributes and chords connecting arcs represent biases. While these methods provide valuable insights in bias analysis, they fall short in analyzing complex, interwoven biases that involve multiple attributes. In contrast, BiasField utilizes a plant-growth-based visualization to hierarchically organize biases, enabling users to quickly identify similar bias patterns and track how biases strengthen with additional attributes.

3 Design of BiasField

3.1 Design Requirements

To inform the design of our tool, we conducted semi-structured interviews with five ML practitioners: three Ph.D. students specializing in machine learning (E1, E2, E3), a professional ML engineer (E4), and a professor with expertise in AI fairness (E5). These interviews aimed to gain insights into their workflows, challenges, and requirements when analyzing biases in ML datasets. Each interview lasted approximately 50 minutes and was fully recorded in both audio and video formats. Based on these interviews and the findings from prior research [28, 32, 37, 43], we identified four design requirements:

R1. Enable adaptive bias detection for different datasets. Bias is flagged when certain attributes are overrepresented in subsets defined by specific combinations of co-occurring attributes. These attributes are usually specific to the datasets, making the one-size-fits-all method insufficient for effective bias detection. E2 commented, "We detect bias considering various attributes in different datasets. For example, bias in portrait image datasets mainly involves demographic attributes such as gender and age, while bias in traffic image datasets concerns additional attributes such as locations and environmental factors." Therefore, our tool should support adaptive bias detection that automatically extracts the attributes relevant to a given dataset, facilitating a targeted analysis rather than relying on predefined attributes.

R2. Support structured exploration of biases. The adaptive bias detection process often generates a large number of biases across various attribute combinations, making comprehensive exploration and analysis overwhelming and inefficient. For example, E5 commented, "I always feel overwhelmed when exploring a long list of detected biases. I need to compare different attribute combinations that may contribute to these biases and analyze their impact." Providing an effective navigation mechanism for biases and their relations is crucial for practitioners to gain a high-level overview and identify the biases of interest. This aligns with previous research, which suggests that intuitive visualization effectively reveals patterns and clusters of related biases often hidden in a tabular format [10].

R3. Examine individual biases of interest in detail. Once users identify specific biases of concern through exploration, they need to conduct detailed verification of each bias. This process is heavily context-dependent and cannot be adequately captured by metrics alone [32]. For example, a strong correlation between safety equipment and industrial workers reflects necessary workplace compliance rather than an undesirable stereotype. As E1 explained, "When verifying a bias, we do not directly accept the automatic detection results. Usually, we examine multiple bias metrics, data distributions, and representative raw examples in the datasets." Therefore, the tool should present rich context to support a thorough examination of individual biases.

R4. Mitigate bias through targeted data augmentation. After bias verification, an effective mitigation strategy is necessary to close the analysis loop. Data augmentation offers a practical solution to mitigate biases in datasets by strategically generating additional samples. As E4 noted, "Generative models have become useful tools in our workflow for generating data to address data biases. This effectively balances training data without the substantial costs of collecting additional real-world samples." Building on this insight, the tool should integrate suitable generative models to enable targeted data augmentation and mitigate the identified biases.

3.2 System Overview

Guided by the above design requirements, we develop BiasField to explore and analyze bias in ML datasets. As shown in Fig. 2, BiasField offers a closed-loop workflow for bias analysis, featuring three synergistic modules: bias detection, bias visualization, and bias mitigation. The detection module automatically extracts dataset-specific attributes and detects biases. The visualization module offers a novel

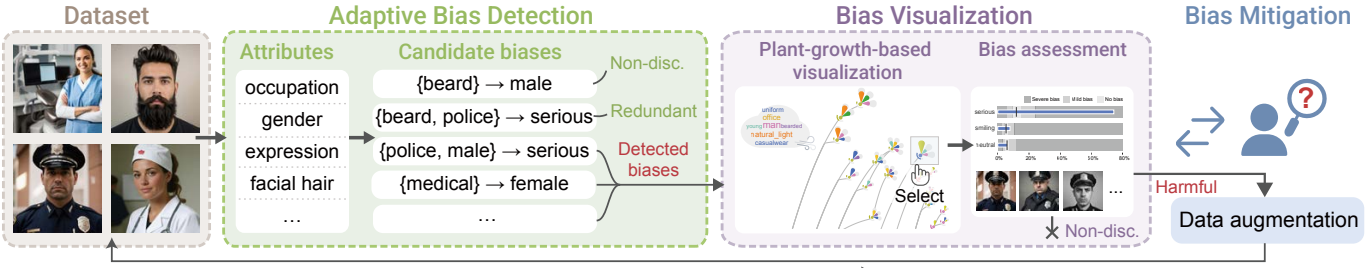

Fig. 2: An overview of how BiasField analyzes and mitigates biases. Given an ML dataset, adaptive bias detection automatically extracts dataset-specific attributes, detects biases, and filters out redundant and non-discriminatory biases. The detected biases are then shown using a plant-growth-based visualization, allowing users to analyze and verify them interactively. Finally, verified biases are mitigated through targeted data augmentation.

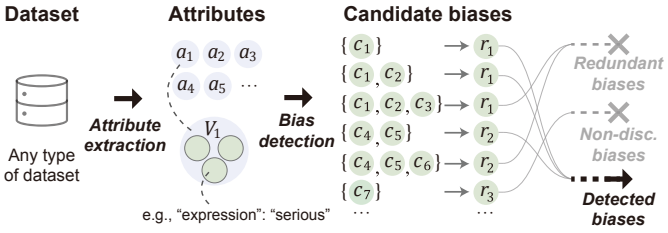

Fig. 3: The workflow of adaptive bias detection.

plant-growth-based visualization that enables structured analysis of the detected biases and their relations (**R2**). During the exploration, users can interactively verify and further exclude non-discriminatory biases by inspecting their attribute distributions in the assessment view and related samples in the sample view (**R3**). The bias mitigation module then mitigates those remaining validated biases through targeted data augmentation with generative models (**R4**).

BiasField is a web-based system with a client-server architecture. The frontend uses Vue.js for UI components and D3.js for visualization, while the backend uses Python with Flask to serve a REST API and host both the bias detection and bias mitigation modules. We will open-source the codebase to facilitate reproducibility and future research.

4 Adaptive Bias Detection

Before introducing the bias detection method, we formalize the dataset schema and define bias. As shown in Fig. 3, a dataset is characterized by a set of attributes $A = \{a_1, a_2, ..., a_m\}$ (*e.g.*, occupation, gender, expression), where each attribute a_i is associated with a set of attribute values V_i (*e.g.*, occupation: police, medical). Bias $\boldsymbol{c} \rightarrow r$ occurs when a combination of specific *conditioning attribute values* $\boldsymbol{c} = \{c_1 \in V_{i_1}, c_2 \in V_{i_2}, \ldots, c_k \in V_{i_k}\}$ significantly increases the frequency of a *resulting attribute value* $r \in V_j$. For example, a bias "{police_officer, male}→ serious" indicates that serious expressions are more frequently observed in portraits of male police officers than in all portraits.

While the definition of bias is clear, two primary challenges persist when examining biases in ML datasets: 1) existing methods typically rely on predefined attributes, thereby overlooking implicit biases involving subtler, dataset-specific attributes; 2) the number of biases increases exponentially with the number of attributes, making the analysis time-consuming. We address these challenges by extracting sample attributes with a multi-modal foundation model and developing an efficient bias detection method. It flags severe biases and filters out cases that do not warrant human verification (**R1**).

4.1 Attribute Extraction

Adaptive attribute set derivation. Inspired by recent work on LLM-based attribute discovery [36, 47], we adaptively derive meaningful attributes from any given dataset. Specifically, we first employ a multi-modal foundation model, GPT-4o, to generate detailed textual descriptions of samples, converting diverse data modalities into a unified text format. The model then summarizes these descriptions and derives a comprehensive attribute set $A = \{a_1, a_2, ..., a_m\}$ that characterizes the samples. This method identifies dataset-specific attributes that may be overlooked in manual specification. Domain experts then review and finalize the set by constraining the number of attributes or by directly editing the outputs through adding, filtering, or refining attributes.

Attribute value extraction. After identifying the attribute set A, we use GPT-4o to extract the corresponding attribute values for each input sample. To improve accuracy, we repeat the extraction process three times per attribute, and the most frequent result is chosen as the final attribute value. If all three results differ, the same model is further prompted to resolve the conflict and decide the final result. We evaluated this method on 1,000 randomly selected samples from a portrait dataset [39]. The results showed that 52.7% of the samples had all three extractions matching, 41.3% had two matching, and only 6.0% required conflict resolution. To assess final accuracy, we also compared our results against annotations from two human annotators. Agreement was measured using Cohen's Kappa [15], a common practice for assessing inter-annotator agreement. Our method achieved an average Kappa of 0.8457, indicating almost perfect agreement and closely matching the agreement between the human annotators (0.8432). Full prompt specifications are provided in Sec. 1 of the supplemental material.

4.2 Bias Detection

Based on the extracted attributes, we detect biases within the dataset. A widely used method is to enumerate all potential biases and evaluate their severity using PMI [8], which measures the co-occurrence of attribute values relative to their individual frequencies. However, given the exponential growth of potential biases, exhaustively evaluating them is computationally infeasible. To address this issue, we aim to identify biases with both high severity and high sample coverage. Because the FP-growth algorithm [29] efficiently detects attribute combinations with high sample coverage, we develop an FP-growth-based bias detection method that considers both bias coverage and severity. First, we employ the FP-growth algorithm to generate attribute combinations by iteratively adding attributes and pruning combinations whose coverage falls below a threshold. Then, for each combination, we choose one attribute value as the resulting attribute value r, and the others as the conditioning attribute values $\boldsymbol{c}$. The bias degree I of $\boldsymbol{c} \rightarrow r$ is quantified in two parts: **bias severity** and **bias coverage**:

$$\mathrm{I}(\boldsymbol{c} \rightarrow r) = \mathrm{PMI}(\boldsymbol{c}, r) \times \log(P(\boldsymbol{c})). \quad (1)$$

The first term is bias severity, which measures the increase in the frequency of the resulting attribute value r among the affected samples: $\mathrm{PMI}(\boldsymbol{c}, r) = \log(\mathrm{P}(r \mid \boldsymbol{c})/\mathrm{P}(r))$. Here, $\mathrm{P}(r \mid \boldsymbol{c})$ denotes the frequency of r among samples that satisfy all conditions in $\boldsymbol{c}$ and $\mathrm{P}(r)$ denotes the frequency of r in the entire dataset. The second term is bias coverage, which is measured by the number of affected samples. Thus, a high bias degree I indicates a bias that is both severe (high PMI) and prevalent (high coverage).

While the FP-growth-based bias detection method is effective, it often returns an overwhelming number of candidate biases. After carefully examining these candidate biases and discussing them with the experts, we identify two types of biases that should be filtered out: redundant biases and non-discriminatory biases.

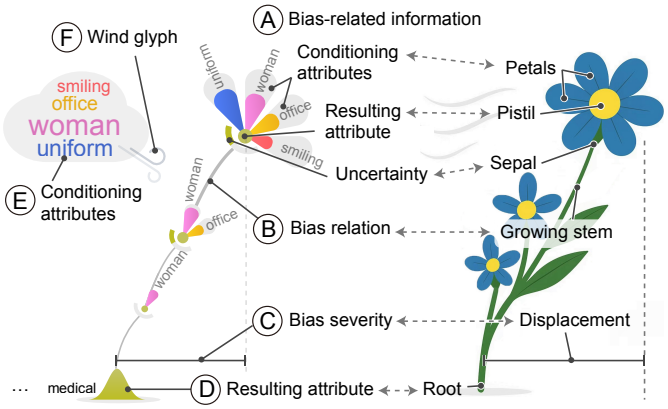

Fig. 4: Visual design of biases using the plant-growth metaphor.

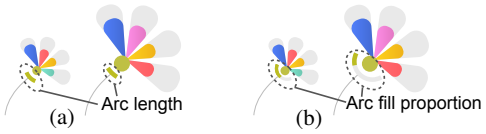

Fig. 5: Two designs for encoding uncertainty values: (a) an alternative design; and (b) our design.

Redundant biases. Redundant biases are those that can be represented by other biases. For example, consider the following two biases:

- b_1: "Women tend to smile more," with $\mathrm{I}(\{\text{female}\} \rightarrow \text{smile}) = 5.0$.
- b_2: "Women wearing yellow clothes tend to smile more," with $\mathrm{I}(\{\text{female}, \text{yellow}\} \rightarrow \text{smile}) = 4.5$.

b_2 contains an additional condition (yellow clothes) but has a lower bias degree compared to b_1 (4.5 vs. 5.0). This suggests that b_1 is more general and severe, eliminating the need for a separate analysis of b_2. Formally, we define a partial order relation between two biases $b_1 : \boldsymbol{c}_1 \rightarrow r$ and $b_2 : \boldsymbol{c}_2 \rightarrow r$, where b_1 represents b_2 ($b_1 \succ b_2$) if and only if $\boldsymbol{c}_1 \subseteq \boldsymbol{c}_2$ and $\mathrm{I}(\boldsymbol{c}_1 \rightarrow r) \geq \mathrm{I}(\boldsymbol{c}_2 \rightarrow r)$. Given a set of biases $B = \{b_1, b_2, \ldots, b_n\}$, our goal is to identify a minimal subset of representative biases that deserve analysis, and thus reduce the overall analysis effort. To find this minimal subset, we iterate through the set B and remove any bias $b' \in B$ if there exists another bias $b \in B$ that represents it (*i.e.*, $b \succ b'$). This subset is formally defined as: $B' = \{b' \in B | \nexists b \in B \text{ s.t. } b \succ b'\}$. The well-foundedness principle [20] guarantees that this subset is minimal (no bias in B' is represented by another bias in B') and complete (every bias in B is represented by some bias in B'). Biases outside B' will be filtered out from the bias analysis.

Non-discriminatory biases. While the detected biases reflect disproportionate attribute frequencies, they are not inherently harmful and do not always require mitigation. For example, although "{beard} → male" exhibits a high bias degree due to its high coverage and strong co-occurrence between "beard" and "male," it represents a non-discriminatory association rather than a harmful stereotype. To support expert verification of such cases, we prompt GPT-4o to assign each detected bias an uncertainty score (0–1) which serves as a hint, with a higher score indicating a higher probability of being non-discriminatory. We empirically determine that a threshold score of 0.8 effectively filters out non-discriminatory biases. Users are allowed to adjust the threshold and the prompt, with the full prompt provided in Sec. 1 of the supplemental material.

5 Bias Visualization

While our method automatically detects numerous biases, determining their real-world impact and harm requires expert analysis grounded in a real-world context. Our visualization directly supports this by organizing the large set of biases for effective examination.

5.1 Visual Design

We adopt a plant-growth metaphor (Fig. 4): each bias appears as a flower on a branching stem, and each branch represents an additional conditioning attribute. These flowers share a common root that represents the shared resulting attribute. We also introduce a wind metaphor that bends the flowers. It symbolizes that the conditioning attributes push the data toward bias. We detail this visual design as follows.

Biases as flowers. Building on the above metaphor, we design a flower-based glyph to encode biases (Fig. 4Ⓐ). The flower pistil encodes the resulting attribute value of this bias, and each petal represents a conditioning attribute value. To allow users to quickly identify the important bias, we use the petal size to encode the overall bias degree I defined in Eq. (1). Flowers with larger petals indicate biases affecting a larger number of samples and reflecting higher severity. To show how each conditioning value contributes, we compute the Shapley value [41] and map it to the fill proportion of each petal. The petal color encodes the types of conditioning attributes. To facilitate the identification of these attributes, an attribute list age gender setting ··· provides an overview of the attributes and their corresponding color encodings.

When exploring biases, it is critical to identify and filter out non-discriminatory biases before bias mitigation. This motivates the design of visually representing the uncertainty of each bias. An intuitive design is inspired by the flower sepal, where we add an additional arc to encode the uncertainty value. We first use the arc length to encode the uncertainty value (Fig. 5(a)). However, the comparisons across biases were unreliable when flower sizes vary. Therefore, we adopt a fill proportion encoding (Fig. 5(b)), which results in our final design. The full arc represents the range of uncertainty values (from 0 to 1), with the fill proportion indicating the uncertainty value. Larger fill proportions indicate greater uncertainty, suggesting the bias might represent a non-discriminatory association and require user verification before mitigation. This visual design enables users to quickly assess uncertainty levels associated with each bias and take appropriate actions.

Bias relations as growing stems. To support the structured exploration of multiple biases, we organize biases sharing the same resulting attribute value as flowers growing on the same plant. The plant root encodes the resulting attribute value (Fig. 4Ⓓ). The color of the root encodes the attribute type, and the height encodes the total degree of the biases on the plant. This allows a quick assessment of which resulting attribute values exhibit stronger biases overall.

Within a plant, biases are arranged by the subset relations between their conditioning attribute values and visualized as flowers growing along branching stems (Fig. 4Ⓑ). This enables users to explore how biases evolve with additional conditions. The bias severity is encoded using the horizontal displacement of flowers from their root (Fig. 4Ⓒ), with a larger displacement indicating a more severe bias. This natural metaphor highlights severe biases while preserving their relations.

Conditioning attribute values as a word cloud. We employ a word cloud [53] to summarize the frequent conditioning attribute values associated with the selected biases and help explain why they occur. The word size encodes the frequency of the attribute value across the biases (Fig.4Ⓔ). This helps quickly identify key contributors to these biases. To strengthen the visual connections between a word cloud and the corresponding biases, we add wind-blown glyphs to connect them (Fig. 4Ⓕ). The wind bending the plants metaphorically illustrates how conditioning attributes drive the biases. The number of wind-blown glyphs encodes the aggregated bias degree. The larger the number, the more severe the bias.

5.2 Layout Algorithm

Plants and flowers are carefully positioned to facilitate exploring similar biases and revealing subset relations between conditioning attributes. As flowers grow on plants, we propose a two-phase layout algorithm that first places the plants and then places the flowers.

Plant placement aims to group plants that have more shared conditioning attribute values and place them closely to facilitate comparison and analysis. For example, plants associated with specific conditioning attributes (*e.g.*, "man") might be placed together, allowing users to

(a) Vertical ordering (b) Vertical placement

Fig. 6: Flower placement: (a) determine vertical ordering to clearly convey the subset relations; (b) compute vertical positions to maximize the spacing between plants.

identify their common patterns more easily. To achieve this, we apply k-means clustering, with the similarity between plants measured by the cosine similarity of their conditioning attribute value frequencies. Since there is no gold standard for determining the optimal number of clusters, we employ the elbow method [35], which identifies the point where adding more clusters no longer significantly improves the clustering quality. Within each cluster, plants are ordered by their maximum bias severity, with the most severe plants placed on the left to align with the left-to-right reading direction.

Flower placement aims to arrange the flowers to clearly convey the subset relations while enhancing visual clarity and aesthetics. The x-position of each flower is inherently determined by the x-position of its root and its bias severity. We optimize the y-position to achieve three goals: 1) to clearly convey the subset relations through the relative y-positions; 2) to enhance visual clarity by ensuring a distinct separation between plants and flowers; and 3) to enhance visual aesthetics with a compact layout and minimal stem length. Among these, the first goal depends on the vertical ordering of flowers, which is a discrete optimization problem, while the latter two are sensitive to their precise y-positions, which is a continuous optimization problem. Given the substantial difference in their optimization spaces, we divide the layout process into two phases. As shown in Fig. 6, we first determine the vertical ordering of biases using topological sorting, and then refine their continuous y-positions within this ordering to enhance visual clarity and aesthetics.

Vertical ordering. For two biases b_1 and b_2, if the conditioning attribute set of b_1 is a subset of the conditioning attribute set of b_2, bias b_2 should be positioned above b_1 in the plant. This aligns with our plant-growth metaphor and facilitates the tracking of bias evolution. To determine this ordering, we apply topological sorting [17] based on the subset relations of the conditioning attribute sets. This method iteratively selects a bias only after all biases associated with its conditioning subsets have been ordered.

Vertical placement. Once the vertical ordering of the flowers is determined, we refine their continuous y-positions within this ordering to enhance visual clarity and aesthetics. Visual clarity requires a distinct separation between plants and flowers, while aesthetic considerations favor a compact layout with minimal stem length. Thus, the placement of flowers is formulated as a multi-objective optimization problem:

$$\begin{aligned} \max_{\boldsymbol{y}} \quad & \left[D_{\text{plant}}, \quad -\sum_{(u,v)\in E} \|p_u - p_v\| \right] \\ \text{s.t.} \quad & \|p_i - p_j\| \geq s_i + s_j, \ \forall b_i, b_j \text{ from the same plant} \\ & \|p_i - p_j\| \geq s_i + s_j + D_{\text{plant}}, \ \forall b_i, b_j \text{ from different plants} \\ & y_i \leq y_j, \ \forall i, j \text{ where } b_i \text{ precedes } b_j \text{ in the ordering} \\ & y_i \leq H, \forall i \end{aligned} \tag{2}$$

where $p_i = (x_i, y_i)$ denotes the position of the flower b_i, and s_i represents its radius. The term $\|p_i - p_j\|$ measures the distance between flowers b_i and b_j. The first constraint prevents overlaps between flowers within the same plant, while the second enforces an additional margin of D_{plant} between flowers from different plants to enhance visual separation. The third constraint maintains the vertical ordering, and the fourth ensures their vertical positions remain within the screen height H. Directly solving this optimization problem is computationally challenging due to its non-linear distance constraints and ordering constraints. To address this, we employ binary parametric search optimization [44], which reformulates an optimization problem into a series of decision problems by iteratively narrowing down the feasible range for D_{plant}. Specifically, we reformulate the original problem as: "For a given minimum plant separation distance D_{plant}, does a feasible layout exist that satisfies all constraints?" To answer this, we implement a bottom-to-top greedy method to minimize total stem length. With D_{plant} fixed, flowers are sequentially placed, following the vertical ordering, in their lowest feasible positions that satisfy the first three constraints. The final constraint is verified by checking the height of the last flower placed, which confirms the feasibility of the decision problem. After identifying the largest feasible D^*_{plant}, we employ the same greedy method to determine the final flower placement that minimizes total stem length.

5.3 Interactive Bias Assessment and Mitigation

To enable a closed-loop workflow for bias assessment and mitigation, BiasField provides a set of rich interactions centered on these tasks.

Bias assessment. Interactions for bias assessment can be categorized into two classes: dataset-level and bias-level.

Dataset-level interactions enable structured exploration of biases and their relations across the entire dataset (**R2**).

- *Plant root exploration*. By default, the height of the root node encodes its total bias degree, helping users identify significant bias. To facilitate the analysis and verification of non-discriminatory biases, users can switch the encoding to represent the average uncertainty by clicking avg uncertainty. To avoid clutter from a potentially large number of biases, the displayed biases are filtered by default to show only the most severe biases for each plant. Users can click 🔍 to zoom in on a specific plant and inspect the complete set of its associated biases, or click ← to zoom out. For example, users observe that "woman" has a notably high bias degree (Fig. 8(D1)) and zoom in for a detailed assessment (Fig. 8(D2)).
- *Word cloud exploration*. To visually connect the conditioning attributes with the associated biases, we provide two interactions. First, hovering over a word cloud highlights the corresponding cluster of bias with a shadow effect. Second, hovering over a specific attribute value within the word cloud will highlight biases that contain the value as a condition for further assessment. For example, in Fig. 8, users may notice the severe biases indicated by the cloud (C1) and a prominent conditioning attribute "woman." To further assess the related biases, users can hover over the attribute "woman" to highlight the corresponding biases. Additionally, users can generate a word cloud for plants of interest by clicking ☁.

Bias-level interactions. After identifying the bias of interest through dataset-level interactions, users can conduct a structured analysis through bias-level interactions (**R3**). By clicking a flower, the selected bias and related ones on the same stem are highlighted. The assessment view (Fig. 8(b)) then presents the selected bias in natural language along with its severity, and the sample view (Fig. 8(c)) lists the affected samples for detailed examination.

In the assessment view, we employ a bullet graph [23] to visualize the distribution of the resulting attribute among the affected samples. Fig. 8(b) illustrates the bullet graph for the bias "{uniform, woman, office, smiling} → medical." The olive green bars show the "occupation" distribution of the samples satisfying conditions "uniform, woman, office, smiling," while the black vertical markers indicate the "occupation" distribution across all samples. The segmented background, from darkest to lightest, represents three qualitative ranges: no bias (within 80–125% of the proportion), mild bias (within 50–200% but outside the no-bias range), and severe bias (below 50% or above 200%). The 80–125% range is based on the "four-fifths rule," a widely-adopted standard in discrimination assessment and algorithmic fairness [22], while the 50–200% range is widely used to identify severe disparities [46].

This visualization helps users quickly identify whether deviations exceed recommended thresholds. In this example, the proportion of "medical" within the affected samples reaches 74%, significantly exceeding the overall proportion (12%) and falling into the severe region.

Bias mitigation. After verifying the biases, users can review all the verified biases by clicking . BiasField supports bias mitigation through targeted data augmentation by clicking (**R4**). Our mitigation method is two-fold. First, based on the data balancing principles of He and Garcia [31], we generate samples that satisfy the same conditioning attributes while varying the resulting attributes. To facilitate this, the system allows users to provide a customizable prompt template and select an appropriate generative model. For example, for the bias "{law_enforcement, male}→ serious," we generate male law enforcement officers with different facial expressions, such as smiling. Examples of the prompt templates used by experts in our case studies are provided in Sec. 1.4 of the supplemental material. Second, when mitigating biases requires generating a large number of samples, we will also apply undersampling techniques to reduce the frequency of overrepresented samples (i.e., samples that satisfy both the conditioning and resulting attributes), thereby reducing the number of samples that need to be generated. Incorporating the sampling techniques allows users to directly manage the generation budget by setting a maximum limit on the total number of generated samples.

6 Evaluation

We conducted a user study, two case studies, and an expert study to demonstrate how BiasField facilitates bias identification and mitigation.

6.1 User Study

We first conducted a controlled user study to evaluate the usefulness of BiasField in supporting bias analysis.

Baseline methods. We compared BiasField against two baselines: a plain tree diagram visualization [4] without the plant-growth metaphor, used to assess the impact of the metaphor; and KYD [26], a state-of-the-art matrix-based system that groups attribute values and reports their strongest biased relations.

Participants. Twelve undergraduate and graduate students (10 male, 2 female; aged 21–28) were recruited for the study. All had prior experience in machine learning, with experience ranging from less than one year to five years. Seven had prior experience in bias analysis. The 12 participants did not include the experts previously interviewed, as their involvement and feedback during the BiasField design process could have introduced familiarity bias.

Task design. Our study comprised three tasks, each targeting a distinct aspect of bias analysis. For each task, we designed three multiple-choice questions, each with four options. Bias Spotting (T1) asked participants to identify the attributes and their values associated with the highest bias severity. Bias Tracing (T2) asked participants to track how the biases evolve when new conditioning attributes are introduced. Attribute Linking (T3) asked participants to identify the relationship between conditioning and resulting attributes. The detailed question types for each task are provided in Sec. 3.1 of the supplemental material.

Study procedure. The study followed a within-subjects design, where each participant was asked to use all three tools to analyze three bias clusters, identified from the same datasets in the case studies (Luccioni *et al.*'s portrait dataset [39] including 31,500 images and the MoPS story dataset [42] including 11,500 story premises). We counterbalanced the pairings of tools and bias clusters across participants to mitigate potential ordering effects. The study began with participants signing the consent form, followed by a 5-minute tutorial on bias analysis and a 3-minute introduction to each tool before their use. Then, participants were asked to use each tool to complete all three tasks (T1-T3). After completing the tasks, participants filled out a custom questionnaire evaluating readability, usability, effectiveness, and aesthetics on a 7-point Likert scale (1 = strongly disagree, 7 = strongly agree). We opted for this over generic scales (*e.g.*, SUS) as they lack items for design-specific dimensions (*e.g.*, aesthetics) essential for our study. The study lasted around 35–45 minutes.

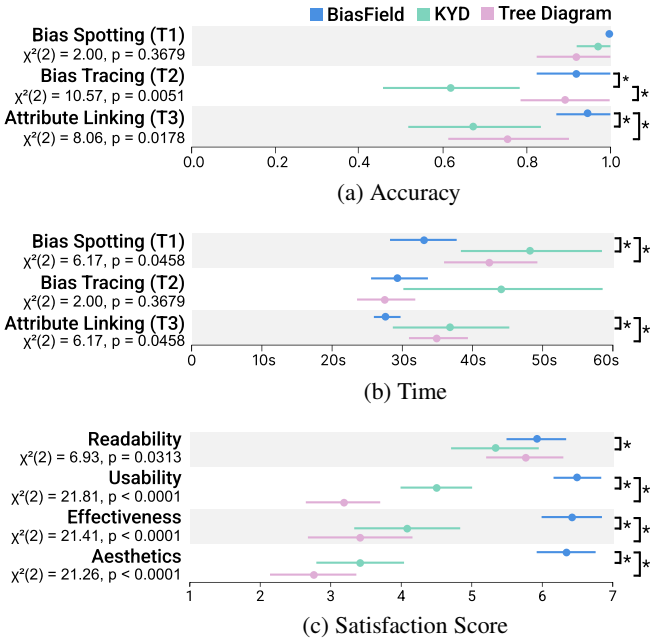

Fig. 7: The user study results, where $*$ indicates $p < 0.05$.

Result analysis. We analyzed task performance using both accuracy and completion time. Since the data were not normally distributed, we applied the Friedman test to examine differences among tools, followed by Wilcoxon signed-rank tests with Benjamini–Hochberg correction [5] to control the false discovery rate in pairwise comparisons.

As shown in Fig. 7(a), the Friedman test revealed no significant accuracy differences among tools for Bias Spotting (T1), but BiasField required less time to complete the task compared to the other two tools (32.7s vs. 48.4s/42.6s). For Bias Tracing (T2), BiasField achieved the highest accuracy (0.92 vs. 0.89/0.61), significantly outperforming KYD, while showing no significant difference compared to the tree diagram. In addition, BiasField and the tree diagram required less time than KYD. These results reveal that both BiasField and the tree diagram enabled participants to directly trace how additional conditioning attributes progressively strengthened biases through the tree-like structure, whereas KYD's matrix visualization imposed higher cognitive load due to repeated row comparisons. For Attribute Linking (T3), BiasField performed significantly better than both the tree diagram and KYD, achieving the highest accuracy (0.94 vs. 0.75/0.67) and fastest completion (27.8s vs. 35.1s/37.0s). This was largely due to the word cloud inspection interaction in BiasField, which allowed participants to examine multiple conditioning attributes simultaneously with reduced effort. The results also indicate that the plant-growth metaphor provides an intuitive overview of the relationships between conditioning and resulting attributes.

In addition to task performance, we also collected subjective user ratings. As shown in Fig. 7(c), all three tools demonstrated comparably high readability scores. However, BiasField received much higher scores in usability (6.50), effectiveness (6.42), and aesthetics (6.33) compared to the other two tools. These results suggest that BiasField was perceived as both usable and effective by participants with varying levels of prior expertise. In particular, three participants without prior bias analysis experience reported that the metaphoric design helped them develop an understanding of bias concepts more quickly.

6.2 Case Studies

6.2.1 Analyzing Biases in Portrait Images

In this case study, we collaborated with E1 to develop an image classification model that predicts a person's occupation from a portrait image. He used the Luccioni *et al.*'s dataset [39] as the training data, which contains 31,500 portrait images generated using Stable Diffusion 2.0 [48]. The dataset covers 146 common occupations. To simplify the task, E1

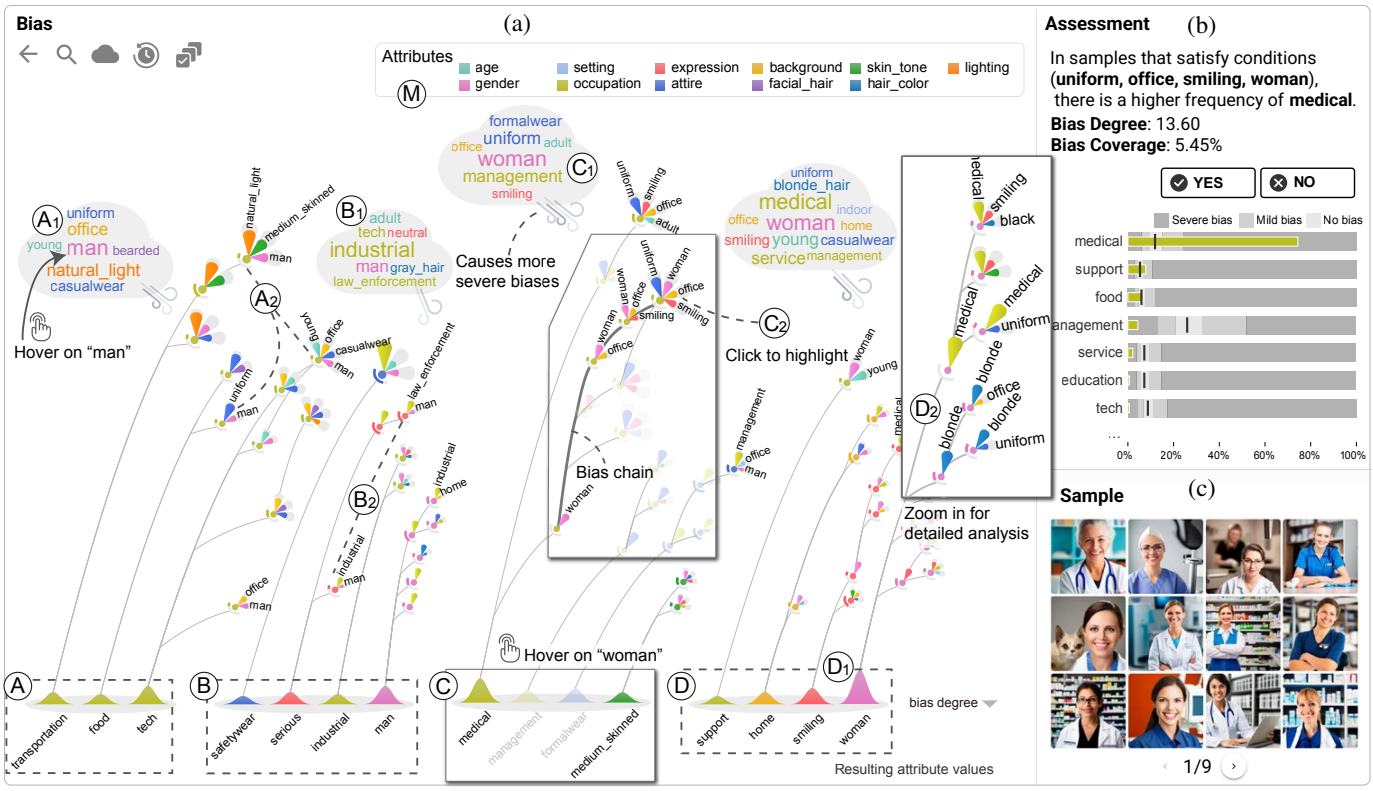

Fig. 8: BiasField: (a) the bias view using the plant-growth metaphor to visualize biases and their relations; (b) the assessment view showing the description and attribute value distribution of the selected bias using a bullet graph; (c) the sample view presenting samples for the selected bias. The bias view (a) contains four bias clusters (Ⓐ-Ⓓ), with a legend Ⓜ displaying the color encoding of extracted attributes.

grouped them into 11 broader categories (*e.g.*, medical, tech). E1 then trained the EfficientNet model [52], achieving 79.78% accuracy on a manually annotated test set of 3,150 images generated using the same procedure as the training data. However, concerns about underlying social biases emerged after an analysis of the model predictions, such as gender stereotypes associated with certain occupations, potentially compromising the model's fairness. To quantify this concern, E1 calculated the Equality of Odds Difference (EOD) [30], which measures disparities in true positive rates between demographic groups. The results revealed notable biases toward men in tech (0.216) and transportation (0.184), and biases toward women in medical (0.192) and support (0.180). These findings prompted further analysis of the biases. Therefore, E1 executed the provided Python script to extract attributes and detect biases, which were visualized for in-depth analysis.

Analyze severe biases. E1 first examined the attribute list in Fig. 8Ⓜ, which includes 11 key attributes: 10 automatically extracted attributes (*e.g.*, gender, age, skin color, hair color, expression) and one predefined attribute, occupation. E1 confirmed that these attributes effectively described the image content from different perspectives. The bias visualization displayed four bias clusters (Fig. 8Ⓐ-Ⓓ), each containing biases that shared more conditioning attribute values. For example, most biases in cluster Ⓐ contain the conditioning attribute value "man."

E1 decided to analyze each cluster sequentially, starting with cluster Ⓐ, where the resulting attribute values were all occupations: "food," "transportation," and "tech." The word cloud indicated that "man" was a common conditioning attribute value in this cluster and contributed to multiple biases (Fig. 8$Ⓐ_1$). This attracted E1's attention, as gender-related biases are particularly sensitive and require careful analysis. He hovered over the word "man" to highlight the biases containing "man" as a condition. Several severe biases were identified (Fig. 8$Ⓐ_2$), such as "{natural_light, medium_skinned, man} → transportation," "{office, young, casualwear, man} → tech," and "{uniform, man} → food." Conditions like "natural_light" and "uniform" were not directly related to specific occupations, indicating that these biases were stereotypes and warranted mitigation. Similarly, E1 analyzed cluster Ⓑ, where the word cloud included "man" and occupations, such as "industrial" and "law_enforcement" (Fig. 8$Ⓑ_1$). E1 found biases reflecting harmful stereotypes, such as "{industrial, man} → serious" and "{law_enforcement, man} → serious," which link male-associated roles to seriousness (Fig. 8$Ⓑ_2$). Based on these findings, E1 concluded that the generated dataset not only inherited overall occupational gender biases but also reinforced gender stereotypes in emotional expressions.

After analyzing the biases related to "man," E1 decided to investigate whether similar biases existed for "woman." He noticed that the word cloud of cluster Ⓒ included "woman," and the increased number of wind-blown glyphs indicated that it caused more severe biases (Fig. 8$Ⓒ_1$). Therefore, he hovered over the word "woman," to highlight its associated biases and revealed a strong correlation with the attribute value "medical" (Fig. 8Ⓒ). E1 identified this as a potential issue, as it could reinforce gender stereotypes that associate women with caregiving roles in medical occupations. To investigate further, he clicked on the most severe bias (Fig. 8$Ⓒ_2$), and identified a highlighted chain of biases, where the incorporation of "woman," "office," "smiling," and "uniform" as conditions progressively strengthened the bias toward "medical," increasing the bias degree from 7.13 to 9.28, then 9.69, and finally 13.60. As shown in Fig. 8(b), the most severe bias indicated that nearly 80% of the samples that satisfied these four conditions were medical practitioners, indicating a notable bias. Upon further inspection of the corresponding samples in Fig. 8(c), E1 confirmed that many were in caregiving roles within medical occupations, such as pharmacists and nurses. Concerned by the gender bias related to "medical," E1 suspected that additional biases related to women might exist. E1 proceeded to cluster Ⓓ and found that the plant for "woman" had the highest root among all plants (Fig. 8$Ⓓ_1$), indicating the highest total bias degree. Therefore, E1 zoomed in on this plant for a detailed analysis (Fig. 8$Ⓓ_2$). This plant had two major branching stems, one with "medical" as the condition, which confirmed the previously observed woman-medical bias, and the other with "blonde_hair" as the condition. E1 then focused on biases related to "blonde_hair." He observed that

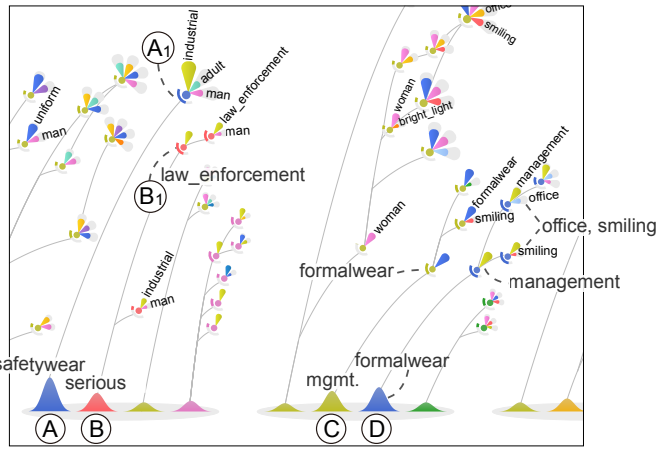

Fig. 9: Analyzing non-discriminatory biases in the portrait dataset: Ⓐ-Ⓓ represent four resulting attribute values with high average uncertainty, indicating potential non-discriminatory biases.

"blonde_hair" as the sole condition resulted in a bias toward "woman" (bias degree: 3.93). Moreover, incorporating conditions like "uniform" or "office" further increased the bias (4.86 and 5.00). This stereotype of blonde women, commonly seen in Western media and advertisements, was indeed present in the training dataset and required mitigation.

Analyze non-discriminatory biases. After analyzing severe biases, E1 proceeded to analyze non-discriminatory biases that should be excluded from bias mitigation. Given that these biases are not necessarily characterized by a high bias degree, E1 switched the encoding of the root to represent the average uncertainty of the biases.

As shown in Fig. 9Ⓐ, E1 first noticed that the average uncertainty of biases related to "safetywear" was the highest. Specifically, one severe bias, "{industrial, man, adult} → safetywear" (Fig. 9Ⓐ1), was identified. While the association between "industrial" and "safetywear" is reasonable, the inclusion of the attribute "man" is problematic, as it may reinforce the harmful stereotype that these professions are exclusively for men. Thus, this bias is retained for mitigation. Meanwhile, he found that four biases related to the resulting attribute value "formalwear" also exhibited high average uncertainty values (Fig. 9Ⓓ). These biases consistently included "management" as the primary condition, accompanied by additional conditions like "office" and "smiling." Conversely, three biases related to "management" frequently had "formalwear" as a condition (Fig. 9Ⓒ). This indicates that "formalwear" and "management" are generally relevant. Since formalwear is a standard dress code requirement for management roles, E1 concluded that the biases in Ⓒ and Ⓓ were non-discriminatory.

Next, E1 examined biases toward "serious," which exhibited the highest remaining average uncertainty (Fig. 9Ⓑ). He considered the bias "{law_enforcement} → serious" (Fig. 9Ⓑ1) a non-discriminatory occupational requirement. However, the addition of "man" in "{law_enforcement, man} → serious" and "{industrial, man} → serious" was problematic. Similar to the "safetywear" case, this may reinforce gender stereotypes, so these two biases were retained for mitigation.

In total, nine non-discriminatory biases were excluded from the mitigation process, while 68 harmful biases were retained.

Bias mitigation. Through the mitigation, 14,292 additional training images were generated. This targeted mitigation proved more effective than both the original model and a baseline method that mitigates all biases without expert verification. In terms of accuracy, our method achieved 86.03%, significantly outperforming the baseline's 83.24% and the original 79.78%. In terms of fairness, our method also reduced the EOD score more effectively. For example, it reduced bias toward men in tech from 0.216 to 0.091 (baseline: 0.112), and reduced bias toward women in support from 0.192 to 0.038 (baseline: 0.057). Detailed comparisons are provided in Sec. 3.2 in the supplemental material. These results demonstrate that selectively mitigating harmful biases optimizes both model accuracy and fairness.

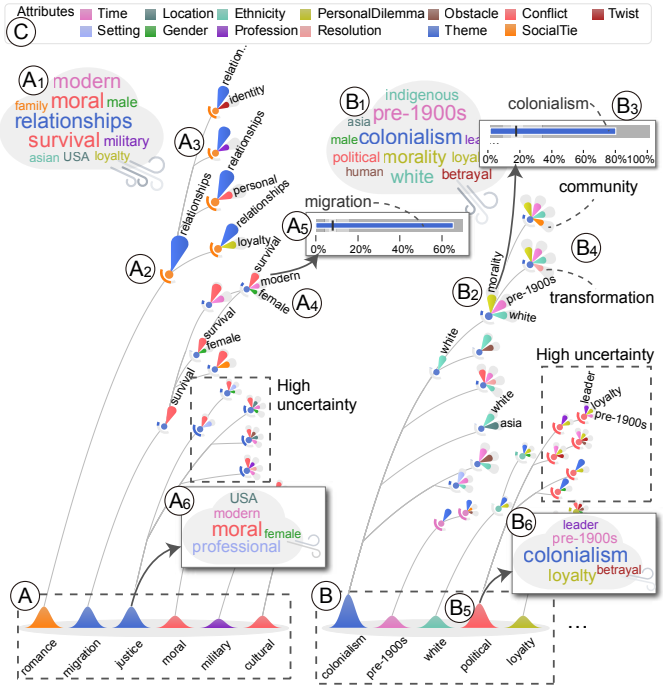

Fig. 10: Analyzing biases in the story premise dataset: Ⓐ biases in the modern context; Ⓑ biases in the historical context.

6.2.2 Analyzing Biases in Story Generation

This case study was collaborated with E2, a Ph.D. student specializing in AI-powered story generation with a focus on enhancing story diversity. The story diversity is based on the diversity of story premises, including main characters, conflicts, settings, and plots [18]. Therefore, E2 analyzed a dataset of 11,500 story premises generated by a state-of-the-art model, MoPS [42], using its default parameters. After processing the dataset of story premises using Python script and visualizing them, E2 found 13 key attributes were extracted, such as time, location, ethnicity, profession, and conflicts (Fig. 10Ⓒ). By observing the roots, E2 found some attribute values with higher total bias degrees, such as "romance," "migration," "justice," and "colonialism" (Fig. 10Ⓐ and Ⓑ). This suggests that even state-of-the-art models are prone to biases, especially when dealing with complex social issues. These biases were grouped into four clusters, and Fig. 10 shows two of them.

Analyze biases in the modern context. E2 first examined the conditioning attribute values in cluster Ⓐ, which contained "relationships" as the theme, and "survival" and "moral" as the conflicts (Fig. 10Ⓐ1). These stories were likely to occur in the "modern (2000-present)" era.

E2 started by hovering over the largest conditioning attribute value in the word cloud, "relationships," and found a series of biases stemming from "{relationships} → romance" (Fig. 10Ⓐ2 and Ⓐ3). These biases exhibited high uncertainty, as indicated by the larger filled portions of the sepals. Since romance is one of the most popular character relationships in stories, the conditioning attribute value "relationships" naturally increases the occurrence of "romance" as the resulting attribute value. Thus, E2 marked all related biases as non-discriminatory.

Next, E2 hovered over the second-largest conditioning attribute value, "survival," to highlight related biases. Most of them appeared in the plant for "migration," with the most severe being "{survival, modern, female} → migration" (Fig. 10Ⓐ4). E2 clicked on this bias to inspect the detailed theme distribution in the bullet graph (Fig. 10Ⓐ5), and discovered that approximately 70% of the stories were about "migration." E2 commented, "MoPS-generated stories over-emphasize 'migration equals a survival crisis,' which might reinforce readers' stereotype of immigrants. This bias thus requires mitigation."

E2 continued to examine another prominent condition, "moral," and discovered that the related biases appeared in the plant for "justice."

The high uncertainty of these biases suggested they were likely non-discriminatory biases. To verify this, E2 generated a word cloud to summarize their conditioning attribute values. It revealed several terms (*e.g.*, "moral," "modern," "USA") that he identified as standard premises in justice-related stories (Fig. 10Ⓐ6). Consequently, E2 marked them as non-discriminatory biases that did not require mitigation.

Analyze biases in the historical context. After examining biases in the modern context, E2 moved to the next cluster Ⓑ, where the stories were typically set in the time of "pre-1900s" (Fig. 10Ⓑ1). The conditioning attribute values mainly included "colonialism," "white," and "morality."

E2 first noticed the plant "colonialism" had a higher root than others in the cluster. Therefore, E2 examined the biases within this plant and identified a severe bias: "{morality, pre-1900s, white} → colonialism" (Fig. 10Ⓑ2). A bullet graph of the associated samples confirmed its severity: approximately 80% stories were centered around the "colonialism" theme (Fig. 10Ⓑ3). To explore conditions that might further strengthen the bias, E2 examined related biases in the same stem, and found that conditioning on "community" and "transformation" strengthened the bias (Fig. 10Ⓑ4). He concluded that these biases were problematic, as they create a misleading and harmful narrative by associating typically positive social concepts with the inherently negative history of "colonialism." Therefore, he marked them as requiring mitigation.

Next, E2 moved on to analyze the second-highest root "political" (Fig. 10Ⓑ5) and noticed that multiple biases exhibited high uncertainty. To assess whether they are non-discriminatory biases, E2 used the word cloud to display their common conditioning attribute values, including "colonialism," "loyalty," and "betrayal" (Fig. 10Ⓑ6). These words reflect power dynamics and conflict, which naturally lead to political stories. Thus, E2 marked all the biases on the plant as non-discriminatory.

Similarly, E2 examined the remaining two clusters to confirm the discriminatory biases and remove 12 non-discriminatory biases.

Bias mitigation. After examining all clusters, E2 used BiasField to generate 3,149 additional story premises and added them to the original dataset. To evaluate whether the augmented story dataset had improved diversity, E2 adopted the two metrics employed by MoPS [42]. The breadth score, which measures the semantic coverage, increased from 22,796 to 24,497, indicating a wider coverage of stories. The density score, which measures the uniformity of coverage, reduced from 75.99 to 70.79, suggesting a more balanced distribution of stories.

6.3 Expert Feedback and Discussion

Following the case studies, we conducted seven semi-structured interviews with the previous five experts (E1-E5) and two newly invited ones (E6 and E7). The two new experts included a designer with experience in information design and a computer science researcher specializing in ML. For those not involved in the case studies, we spent around 20 minutes introducing the tool and the case studies before the interviews. Each interview lasted 50-70 minutes. Overall, the experts were generally positive about the usability of BiasField. They also identified several limitations, suggesting avenues for future research.

6.3.1 Usability

Intuitive design. All the experts agreed that the metaphoric design effectively communicated biases. E6 said: "As a designer, I find the visualization design to be very intuitive because it utilizes familiar metaphoric designs." They also confirmed that the expressive metaphoric designs enabled them to quickly understand complex bias relations and easily track the bias of interest. E4 noted: "I appreciate how the wind bending the flowers serves as a metaphor for bias formation. Just as wind gradually influences a flower's direction, biases subtly shape our thinking over time. E5 agreed that the plant-growth metaphor intuitively shows "how biases gradually evolve with additional conditions."

Reducing analysis effort. All the experts provided positive feedback on how BiasField reduces the manual bias detection workload through its structured analysis workflow. E7 echoed this efficiency: "The tool enables the identification of non-discriminatory associates through interactive visualization. This reduces the efforts to confirm harmful biases one by one." E2 commented on the value of the bias visualization: "The visualization first provides a global overview of biases, which makes it easier to discover key issues across multiple attributes. Then I can focus more on the severe biases." Five experts (E1, E2, E4, E5, E7) agreed that the bullet graph effectively illustrates the influence of different conditioning attributes on the resulting attribute, enabling efficient bias validation.

Extensibility. Although the case studies demonstrate the usefulness of BiasField in image classification and text generation datasets, the experts agreed that it can be easily extended to handle other types of datasets. For example, E3 indicated that our tool can be applied to analyze biases embedded in object detection datasets. This is achieved by first extracting the relevant attributes, such as blur level, size variation, and occlusion degree, and then identifying object categories and groups that are underrepresented in training data. E4 also confirmed that BiasField can be used to understand biases in multi-modal datasets by joining the corresponding attributes extracted from multiple modalities.

6.3.2 Limitations and Future Work

Visual encoding scalability. Our current visualization uses different colors to encode different attributes, which becomes less effective for datasets with many attributes. E1 noted: "When analyzing datasets with more than 15 attributes, color-based distinction becomes difficult to perceive." To address this limitation, we plan to implement a focus+context visualization method that dynamically highlights the relevant attributes based on user interest. This method needs to ensure consistent color assignments for individual attributes across different interactions and states, preserving users' mental maps during exploration. One possible solution is to apply the dynamic color assignment method proposed by Chen *et al*. [13], which assigns distinguishable and visually coherent colors to related attributes, enhancing both clarity and consistency.

Foundation model reliance. We leverage multi-modal foundation models to dynamically detect biases in datasets of different modalities and mitigate them. This introduces a reliance on the model's performance and cost. Regarding performance, we used GPT-4o for attribute value extraction and validated its accuracy against human annotators, confirming high reliability. We also employ GPT-4o to flag potentially non-discriminatory biases for expert verification, which provides useful hints. In addition, our method involves using generative models to generate samples for bias mitigation. However, we observed that current models can still produce low-quality or failed results (*e.g.*, unnatural images or incoherent text). Future work could leverage more advanced foundation models and more robust human-in-the-loop workflows for better bias detection and generation quality. Regarding cost, E4 pointed out a concern: "Our datasets often contain millions of samples. Using multi-modal foundation models to process all of them is expensive." One practical solution is to first sample a representative subset and extract attributes for them using foundation models. These attributes can then serve as labels to train lightweight classifiers for the full dataset, reducing costs while keeping bias detection accurate and scalable.

7 Conclusion

In this paper, we introduce BiasField, an interactive tool for comprehensive bias analysis and mitigation in ML datasets. Compared with previous work, BiasField extends bias detection beyond predefined attributes by automatically extracting dataset-specific attributes, thereby adaptively detecting both explicit and implicit biases. It features a plant-growth-based visualization that facilitates structured analysis and assessment of interwoven biases and their associated attributes. Confirmed biases are then mitigated through targeted generative data augmentation. A user study with 12 participants, two case studies that analyze biases in an image dataset and a text dataset, and an expert study with seven experts demonstrate BiasField's effectiveness in detecting, analyzing, and mitigating complex biases.

Supplemental Materials

The supplemental materials are available at https://bit.ly/3DNxAX9. In particular, they include: (1) a video demonstrating the tool and the case studies, and (2) a PDF file including the prompts for adaptive bias detection and bias mitigation, the results for adaptive bias detection, and additional results from the case studies.

References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Y. Ahn and Y.-R. Lin. FairsSight: Visual analytics for fairness in decision making. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1086–1095, 2020. doi: 10.1109/TVCG.2019.2934262 2

[3] S. Alemzadeh, U. Niemann, T. Ittermann, H. Völzke, D. Schneider, M. Spiliopoulou, K. Bühler, and B. Preim. Visual analysis of missing values in longitudinal cohort study data. *Computer Graphics Forum*, 39(1):63–75, 2020. doi: 10.1111/cgf.13662 2

[4] G. D. Battista, P. Eades, R. Tamassia, and I. G. Tollis. *Graph drawing: algorithms for the visualization of graphs*. Prentice Hall PTR, 1998. 6

[5] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995. 6

[6] R. Berk, H. Heidari, S. Jabbari, M. Kearns, and A. Roth. Fairness in criminal justice risk assessments: The state of the art. *Sociological Methods & Research*, 50(1):3–44, 2021. doi: 10.1177/0049124118782533 2

[7] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. doi: 10.48550/arXiv.2108.07258 1

[8] N. Borenstein, K. Stanczak, T. Rolskov, N. Klein Käfer, N. da Silva Perez, and I. Augenstein. Measuring intersectional biases in historical documents. In *Proceedings of Findings of the Association for Computational Linguistics*, pp. 2711–2730, 2023. doi: 10.18653/v1/2023.findings-acl.170 2, 3

[9] J. Buolamwini and T. Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Proceedings of Conference on Fairness, Accountability and Transparency*, pp. 77–91, 2018. 2

[10] Á. A. Cabrera, W. Epperson, F. Hohman, M. Kahng, J. Morgenstern, and D. H. Chau. FairVis: Visual analytics for discovering intersectional bias in machine learning. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*, pp. 46–56, 2019. doi: 10.1109/VAST47406.2019.8986948 2

[11] C. Chen, J. Chen, W. Yang, H. Wang, J. Knittel, X. Zhao, S. Koch, T. Ertl, and S. Liu. Enhancing single-frame supervision for better temporal action localization. *IEEE Transactions on Visualization and Computer Graphics*, 2024. doi: 10.1109/TVCG.2024.3388521 2

[12] C. Chen, J. Yuan, Y. Lu, Y. Liu, H. Su, S. Yuan, and S. Liu. OoDAnalyzer: Interactive analysis of out-of-distribution samples. *IEEE Transactions on Visualization and Computer Graphics*, 27(7):3335–3349, 2021. doi: 10.1109/TVCG.2020.2973258 2

[13] J. Chen, W. Yang, Z. Jia, L. Xiao, and S. Liu. Dynamic color assignment for hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 31(1):338–348, 2025. doi: 10.1109/TVCG.2024.3456386 9

[14] A. Chouldechova. Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. *Big Data*, 5(2):153–163, 2017. doi: 10.1089/big.2016.0047 2

[15] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960. doi: 10.1177/001316446002000104 3

[16] S. Corbett-Davies, E. Pierson, A. Feller, S. Goel, and A. Huq. Algorithmic decision making and the cost of fairness. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 797–806, 2017. doi: 10.1145/3097983.3098095 2

[17] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2022. 5

[18] L. Cron. *Wired for story: The writer's guide to using brain science to hook readers from the very first sentence*. Ten Speed Press, 2012. 8

[19] S. Danziger, J. Levav, and L. Avnaim-Pesso. Extraneous factors in judicial decisions. *Proceedings of the National Academy of Sciences*, 108(17):6889–6892, 2011. doi: 10.1073/pnas.1018033108 2

[20] B. Davey and H. Priestley. *Introduction to Lattices and Order*. Cambridge University Press, 2002. 4

[21] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024. 1

[22] M. Feldman, S. A. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data mining*, pp. 259–268, 2015. doi: 10.1145/2783258.2783311 5

[23] S. Few. *Information dashboard design: The effective visual communication of data*. O'Reilly Media, Inc., 2006. doi: 10.5555/1206491 5

[24] B. Ghai, M. N. Hoque, and K. Mueller. WordBias: An interactive visual tool for discovering intersectional biases encoded in word embeddings. In *Proceedings of Extended Abstracts of the ACM CHI Conference on Human Factors in Computing Systems*, pp. 1–7, 2021. doi: 10.1145/3411763.3451587 2

[25] B. Ghai and K. Mueller. D-BIAS: A causality-based human-in-the-loop system for tackling algorithmic bias. *IEEE Transactions on Visualization and Computer Graphics*, 29(1):473–482, 2023. doi: 10.1109/TVCG.2022.3209484 2

[26] Google Research Ethical AI Team. Know your data. Online, 2021. Accessed: Month Day, Year. 2, 6

[27] L. Gou, L. Zou, N. Li, M. Hofmann, A. K. Shekar, A. Wendt, and L. Ren. VATLD: A visual analytics system to assess, understand and improve traffic light detection. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):261–271, 2021. doi: 10.1109/TVCG.2020.3030350 2

[28] Z. Gu, J. N. Yan, and J. M. Rzeszotarski. Understanding user sensemaking in machine learning fairness assessment systems. In *Proceedings of the Web Conference*, pp. 658–668, 2021. doi: 10.1145/3442381.3450092 2

[29] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 1–12, 2000. doi: 10.1145/342009.335372 3

[30] M. Hardt, E. Price, and N. Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29, 2016. 7

[31] H. He and E. A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009. doi: 10.1109/TKDE.2008.239 6

[32] K. Holstein, J. Wortman Vaughan, H. Daumé, M. Dudik, and H. Wallach. Improving fairness in machine learning systems: What do industry practitioners need? In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pp. 1–16, 2019. doi: 10.1145/3290605.3300830 2

[33] S. Kabir, L. Li, and T. Zhang. STILE: Exploring and Debugging Social Biases in Pre-trained Text Representations. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pp. 1–20, 2024. doi: 10.1145/3613904.3642111 2

[34] M. Kim, K. Kang, D. Park, J. Choo, and N. Elmqvist. Topiclens: Efficient multi-level visual topic exploration of large-scale document collections. *IEEE transactions on visualization and computer graphics*, 23(1):151–160, 2016. 2

[35] T. M. Kodinariya, P. R. Makwana, et al. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95, 2013. 5

[36] M. S. Lam, J. Teoh, J. A. Landay, J. Heer, and M. S. Bernstein. Concept induction: Analyzing unstructured text with high-level concepts using lloom. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pp. 1–28, 2024. 3

[37] P.-M. Law, S. Malik, F. Du, and M. Sinha. Designing tools for semi-automated detection of machine learning biases: An interview study. *arXiv preprint arXiv:2003.07680*, 2020. doi: 10.48550/arXiv.2003.07680 2

[38] S. Liu, W. Yang, J. Wang, and J. Yuan. *Visualization for Artificial Intelligence*. Synthesis Lectures on Visualization. Springer Cham, 2025. doi: 10.1007/978-3-031-75340-4 1, 2

[39] S. Luccioni, C. Akiki, M. Mitchell, and Y. Jernite. Stable Bias: Evaluating societal representations in diffusion models. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 56338–56351, 2024. doi: 10.5555/3666122.3668580 1, 2, 3, 6

[40] L. Lucy and D. Bamman. Gender and representation bias in GPT-3 generated stories. In *Proceedings of the Workshop on Narrative Understanding*, pp. 48–55, 2021. doi: 10.18653/v1/2021.nuse-1.5 1

[41] S. M. Lundberg and S.-I. Lee. A unified approach to interpreting model predictions. In *Proceedings of Advances in Neural Information Processing Systems*, p. 4768–4777, 2017. doi: 10.5555/3295222.3295230 4

[42] Y. Ma, Y. Qiao, and P. Liu. MoPS: Modular story premise synthesis for open-ended automatic story generation. In *Proceedings of Annual Meeting of the Association for Computational Linguistics*, pp. 2135–2169, 2024. doi: 10.18653/v1/2024.acl-long.117 6, 8, 9

[43] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi. BAGAN: Data augmentation with balancing GAN. *arXiv preprint arXiv:1803.09655*, 2018. doi: 10.48550/arXiv.1803.09655 2

[44] N. Megiddo. Applying parallel computation algorithms in the design of serial algorithms. *Journal of the ACM*, 30(4):852–865, 1983. doi: 10.1145/2157.322410 5

[45] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan. A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys*, 54(6):1–35, July 2022. doi: 10.1145/3457607 2

[46] R. R. Monson. *Occupational epidemiology*. CRC press, 1990. 5

[47] E. Reif, C. Qian, J. Wexler, and M. Kahng. Automatic histograms: Leveraging language models for text dataset exploration. In *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pp. 1–9, 2024. 3

[48] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, June 2022. 6

[49] N. Shahbazi, Y. Lin, A. Asudeh, and H. Jagadish. Representation bias in data: A survey on identification and resolution techniques. *ACM Computing Surveys*, 55(13s):1–39, 2023. doi: 10.1145/3588433 1

[50] S. Shankar, Y. Halpern, E. Breck, J. Atwood, J. Wilson, and D. Sculley. No classification without representation: Assessing geodiversity issues in open data sets for the developing world. *arXiv preprint arXiv:1711.08536*, 2017. doi: 10.48550/arXiv.1711.08536 2

[51] D. Song, Z. Wang, Y. Huang, L. Ma, and T. Zhang. DeepLens: Interactive out-of-distribution data detection in NLP models. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pp. 1–17, 2023. doi: 10.1145/3544548.3580741 2

[52] M. Tan and Q. Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning*, pp. 6105–6114, 2019. 7

[53] F. B. Viegas, M. Wattenberg, and J. Feinberg. Participatory visualization with wordle. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1137–1144, 2009. doi: 10.1109/TVCG.2009.171 4

[54] J. Wang, S. Liu, and W. Zhang. Visual analytics for machine learning: A data perspective survey. *IEEE Transactions on Visualization and Computer Graphics*, 30(12):7637–7656, 2024. doi: 10.1109/TVCG.2024.3357065 2

[55] Q. Wang, Z. Xu, Z. Chen, Y. Wang, S. Liu, and H. Qu. Visual analysis of discrimination in machine learning. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1470–1480, 2021. doi: 10.1109/TVCG.2020.3030471 2

[56] X. Wang, W. Chen, J.-K. Chou, C. Bryan, H. Guan, W. Chen, R. Pan, and K.-L. Ma. GraphProtector: A visual interface for employing and assessing multiple privacy preserving graph algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):193–203, 2019. doi: 10.1109/TVCG.2018.2865021 2

[57] X. Wang, W. Chen, J. Xia, Z. Chen, D. Xu, X. Wu, M. Xu, and T. Schreck. ConceptExplorer: Visual analysis of concept drifts in multi-source time-series data. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*, pp. 1–11, 2020. doi: 10.1109/VAST50239.2020.00006 2

[58] J. Wexler, M. Pushkarna, T. Bolukbasi, M. Wattenberg, F. Viégas, and J. Wilson. The What-If tool: Interactive probing of machine learning models. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):56–65, 2020. doi: 10.1109/TVCG.2019.2934619 2

[59] S. Xiang, X. Ye, J. Xia, J. Wu, Y. Chen, and S. Liu. Interactive correction of mislabeled training data. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*, pp. 57–68, 2019. doi: 10.1109/VAST47406.2019.8986943 2

[60] J. N. Yan, Z. Gu, H. Lin, and J. M. Rzeszotarski. Silva: Interactively assessing machine learning fairness using causality. In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2020. doi: 10.1145/3313831.3376447 2

[61] W. Yang, Z. Li, M. Liu, Y. Lu, K. Cao, R. Maciejewski, and S. Liu. Diagnosing concept drift with visual analytics. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*, pp. 12–23, 2020. doi: 10.1109/VAST50239.2020.00007 2

[62] W. Yang, M. Liu, Z. Wang, and S. Liu. Foundation models meet visualizations: Challenges and opportunities. *Computational Visual Media*, 10(3):399–424, 2024. doi: 10.1007/s41095-023-0393-x 1, 2

[63] J. Yuan, C. Chen, W. Yang, M. Liu, J. Xia, and S. Liu. A survey of visual analytics techniques for machine learning. *Computational Visual Media*, 7(1):3–36, 2021. doi: 10.1145/3457607 2

[64] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang. Men also like shopping: Reducing gender bias amplification using corpus-level constraints. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 2979–2989, 2017. doi: 10.18653/v1/D17-1323 2