CHAMELEON: Foundation Models for Fairness-aware Multi-modal Data Augmentation to Enhance Coverage of Minorities

Mahdi Erfanian
University of Illinois Chicago
merfan2@uic.edu

H. V. Jagadish
University of Michigan
jag@umich.edu

Abolfazl Asudeh
University of Illinois Chicago
asudeh@uic.edu

ABSTRACT

Potential harms from the under-representation of minorities in data, particularly in multi-modal settings, is a well-recognized concern. While there has been extensive effort in detecting such under-representation, resolution has remained a challenge.

With recent generative AI advancements, large language and foundation models have emerged as versatile tools across various domains. In this paper, we propose CHAMELEON, a system that efficiently utilizes these tools to augment a dataset with minimal addition of synthetically generated tuples to enhance the coverage of the under-represented groups. Our system applies quality and outlier-detection tests to ensure the quality and semantic integrity of the generated tuples. In order to minimize the rejection chance of the generated tuples, we propose multiple strategies to provide a guide for the foundation model. Our experiment results, in addition to confirming the efficiency of our proposed algorithms, illustrate our approach's effectiveness, as the model's unfairness in a downstream task significantly dropped after data repair using CHAMELEON.

PVLDB Reference Format:
Mahdi Erfanian, H. V. Jagadish, and Abolfazl Asudeh. CHAMELEON: Foundation Models for Fairness-aware Multi-modal Data Augmentation to Enhance Coverage of Minorities. PVLDB, 17(11): 3470 - 3483, 2024.
doi:10.14778/3681954.3682014

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/UIC-InDeXLab/Chameleon.

1 INTRODUCTION

"The CHAMELEON *changes color to match the earth, the earth doesn't change color to match the chameleon."* – SENEGALESE PROVERB

The importance of the **dataset** as *the first product of the data analysis pipeline* [49, 62] is now well-recognized. In particular, there is increasing awareness of the unfairness of machine learning (ML) models towards minorities and other marginalized groups on account of their under-representation in training data [10]. Such issues also appear in other contexts, such as information retrieval [11]. There is now a growing body of work on detecting lack of *coverage* of minorities in a dataset [5, 60, 74].

Proceedings of the VLDB Endowment, Vol. 17, No. 11 ISSN 2150-8097.
doi:10.14778/3681954.3682014

Of course, detecting under-representation does not in itself address the problem: we then need to fix it somehow. If additional data could be collected or suitable external sources [61], that would be ideal, but this is frequently not possible. An alternative approach to generate synthetic data has been explored for regular alphanumeric relational tables [18, 22, 37, 46].

Multi-modal data is increasingly being used for analysis, exploiting huge recent technological advances such as image recognition. In fact, the under-representation-related issues for multi-modal data have been noticed for quite some time. In data retrieval, for example, search engines returned images belonging to certain demographic groups for specific queries [40, 86]. A famous example is the "CEO Gender Bias" where the returned images for CEO-related queries are mostly (white) male [40]. The harms of under-representation in data used for training multi-modal ML models are also well-known. For example, HP webcams were not able to detect black faces [77] due to *inadequate coverage* of black faces in the training data: the face images used to train the software were collected from (mostly) white engineers [83]. This begs the question, what can we do once we have detected that a multi-modal dataset is biased, with insufficient representation of certain groups? There is no obvious way we can apply techniques developed for alphanumeric relational data.

Therefore, our objective in this paper is to resolve inadequate coverage of minorities in a multi-modal dataset. Ensuring proper representation of minorities can help prevent false stereotypes such as `<CEOs being white-male>` in contexts such as information retrieval. In the context of ML, as we shall experimentally illustrate in § 6, improving the coverage of minorities in a dataset can help reduce unfairness in the downstream model training.

Our central idea is to use generative AI to create synthetic data for this purpose. While this idea has immediate appeal, particularly given the spectacular recent advances in foundation models, actually getting it to work requires overcoming many challenges. First, we have to determine the minimal set of synthetic tuples that can be added to the original dataset to resolve under-representation issues. Second, we need to ensure the semantic integrity of the dataset, that the synthetically generated tuples are in the same context as the original dataset. That is, they are not outliers based on the underlying distribution represented by the input dataset. Third, we have to ensure the generated tuples are of high quality so they look realistic to a human evaluator. Lastly, given the (often monetary) cost associated with the queries to the foundation model, we should ensure the cost-effectiveness of the dataset repair process.

To address the first challenge, using the notion of data coverage [5, 74] for identifying under-representation, we formally define the COMBINATION-SELECTION problem, which minimizes the total number of synthetic tuples for resolving lack of coverage of minorities at the most general level. We show the problem is NP-hard, and

propose a greedy approximation algorithm for it. For the second challenge, we view each tuple in the dataset $\mathcal{D}$ as an independent and identically distributed (iid) random sample from the underlying distribution ξ it represents. We use the vector representations (embeddings) space to describe the distribution. Then, a newly generated tuple is discarded if it fails the outlier test, i.e., if it is unlikely to be generated by ξ. To address the third challenge, we model quality evaluation as hypothesis testing and reject the samples with a higher chance of being labeled as "unrealistic" by a random human evaluator. Finally, to minimize the number of queries to the foundation model, we provide a guide tuple (and a mask) and the prompt to the foundation model. We propose multiple strategies for guide selection to maximize the chance of passing the outlier and the quality tests.

Summary of contributions. We introduce Chameleon, a system that uses foundation models to augment multi-modal datasets to enhance their representation of minorities in the form of data coverage. In summary, our contributions are the following:

- We propose fairness-aware data augmentation using foundation models to resolve the lack of coverage in multi-modal data (§ 2).
- We propose rejection tests to ensure the augmented tuples are not outliers according to the underlying data distribution and have as high quality as the real tuples in the dataset (§ 3).
- We propose the Combination-Selection problem, which specifies the description of the tuples to be generated to resolve the lack of coverage with a minimum amount of augmentation to the dataset. We prove that the Combination-Selection problem is NP-hard, and propose a greedy approximation algorithm with the logarithmic approximation ratio for it (§ 4).
- We propose the Guide-selection problem that provides a guide tuple and a mask as the input to the foundation model to maximize the chance of passing the rejection tests. We propose multiple strategies for guide selection, including a solution based on a contextual multi-armed bandit (§ 5).
- We conduct comprehensive experiments on several real datasets to evaluate the efficiency of the proposed algorithms in comparison to the baselines and to study their effectiveness using human evaluators. Our proof of concept experiments using different datasets/tasks showcase the reduction in unfairness (performance disparity) achieved by Chameleon. Among other experiments and in addition to image datasets, we evaluate Chameleon on textual data on a sentiment analysis task with an alternative metric for measuring coverage, which confirms the extensibility of our system for different settings (§ 6).

2 PRELIMINARIES

2.1 (Input) Data Model

We are given a dataset of multi-modal tuples (e.g., images) $\mathcal{D} = \{t_1, \cdots, t_n\}$, as a collection of independent and identically distributed (iid) samples, taken from an (unknown) distribution ξ. The tuples are associated with $d \geq 1$ attributes of interest $\mathbf{x} = \{x_1, \cdots, x_d\}$ (e.g., `gender`, `race`, `age-group`, etc.), that are used to identify (demographic) groups. Without loss of generality, we assume the attributes of interest are categorical (we assume the continuous attributes are properly bucketized). Attributes of interest can be unordered (e.g., `gender` and `race`) or ordinal (e.g., `age-group`). Each attribute has a cardinality of two or more. For example, an attribute `sex` (biological sex) with values `{male, female}` partitions the individuals into two non-overlapping groups. We use $dom(x_i)$ to represent the domain of the attribute $x_i \in \mathbf{x}$, i.e., the set of valid values for x_i. The cartesian product of values on a subset of attributes $\mathbf{x}' \subseteq \mathbf{x}$, form a set of (demographic) subgroups. For example, {`white male`, `white female`, `black male`, $\cdots$} are the subgroups defined on the attributes (`race`,`gender`). We refer to the number of attributes used to specify a subgroup as the *level* of that subgroup. For example, the level of the subgroup `white male` is 2, while the level of the subgroup `male` is 1. We use $\ell(\mathbf{g})$ to refer to the level of a subgroup $\mathbf{g}$. Similarly, we say a subgroup $\mathbf{g}'$ is a subset of $\mathbf{g}$, if the groups specifying $\mathbf{g}'$ are a superset of the ones for $\mathbf{g}$. For example, `{white male preschooler}` is a subset of the more general group `{white male}`. That is, the set of individuals in group `{white male preschooler}` is a subset of `{white male}`. Moreover, we say a subgroup $\mathbf{g}$ is a *parent* of the subgroup $\mathbf{g}'$, if $\mathbf{g}' \subset \mathbf{g}$ and $\ell(\mathbf{g}) = \ell(\mathbf{g}') + 1$. For example, the subgroup `{white male}` is a parent of the subgroup `{white male preschooler}`. Finally, slightly abusing the terms, we call a subgroup a *combination* if $\ell(\mathbf{g}) = d$.

2.2 (Input) Foundation Model

We use a foundation model $\mathcal{F}$ (e.g., DALL·E 2[1]) for data generation. We treat $\mathcal{F}$ as black-box, which allows the adaptation of both closed-source and open-source foundation models. For more information about the foundation models, please refer to [14, 17, 90]. We consider the foundation model $\mathcal{F}$ with the following inputs that generate a synthesized output tuple:

- **Prompt**: a natural language instruction specifying the tuple details. For example, an image generation prompt could be "A realistic photo of a white cat".
- **Guide**: when only provided with a prompt, the foundation model uses its "imagination" to generate the requested tuple. For example, for the previous cat-image example prompt, the breed and size of the cat, the background, and other details are chosen by the foundation model. Alternatively, a guide can be provided to $\mathcal{F}$ to influence the generation process. We formalize the guide as a pair (t, m), where t is a tuple and m is a mask. The mask m specifies which parts of the guide tuple should change. Continuing with the cat example, t can be a cat image, and m can specify the foreground to be regenerated.

Cost model: We assume each query to the foundation model has a fixed cost v. The cost is monetary when using external foundation models such as DALL·E 2, and it can be computational when the model is hosted locally.

2.3 (Objective) Data Coverage

We use the notion of *data coverage* [5] to identify representation issues in a dataset $\mathcal{D}$.[2] In particular, given a dataset $\mathcal{D}$ and a coverage threshold τ (e.g., $\tau = 50$), we say a subgroup $\mathbf{g}$ is *uncovered*, if

[1] Chameleon uses DALL·E 2 as its default image generator. "DALL·E 2 is an AI system that can create realistic images and art from a description in natural language." https://openai.com/dall-e-2

[2] The modular architecture of Chameleon (Figure 1) allows for alternative metrics, like representation rate [19], by simply changing the "combination selection" component. This will be demonstrated experimentally in § 6, using the representation rate as the metric.

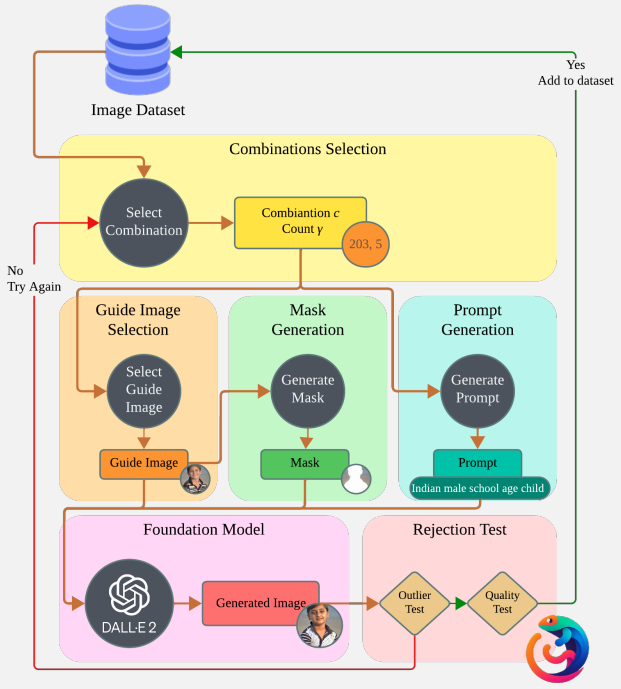


Figure 1: Architecture of CHAMELEON for image generation

$|\mathbf{g} \cap \mathcal{D}| < \tau$. That is, the number of samples in $\mathcal{D}$ from the group $\mathbf{g}$ is less than τ.

When studying the lack of coverage in a dataset, we are usually interested in finding the most general uncovered subgroups. That is, the collection of subgroups $\mathbf{g}$ such that (a) $\mathbf{g}$ is uncovered and (b) all parents of $\mathbf{g}$ are covered. We follow the terminology proposed by Asudeh et al. [5], where these subgroups are called MUPs (*maximal uncovered patterns*). As a toy example, suppose there are 40 `black-females` in a dataset, while the number of `females` is 200, and this number is 150 for `blacks`. Hence, considering $\tau = 50$, {`black female`} is a MUP as it is a most general subgroup that is uncovered.

2.4 System Architecture Overview

Figure 1 shows the overall architecture of the system, whose components we will design in the rest of this paper. The augmentation process for a dataset $\mathcal{D}$ starts with specifying a small set of synthetic tuples (a set of combinations, each with a count) that, once generated and added to $\mathcal{D}$, resolve problematic lack of coverage issues (§ 4). Then, for each combination, an input query is constructed and passed to the foundation model. At a minimum, this query comprises a text prompt describing the desired combination. However, that leaves too much latitude to the foundation model, and the result is likely to be an image unsuitable for the dataset at hand (that is, it would be unlikely to occur in the underlying distribution of $\mathcal{D}$), even if it satisfies the prompt conditions. To avoid this, a guide tuple and mask are specified in addition to the prompt (§ 5). The foundation model then generates a new tuple based on this input. Even with the augmented query, the produced tuple may not be satisfactory. We follow a rejection testing strategy, where the new tuple should pass an outlier detection test and a quality test before adding it to the dataset.

3 REJECTION TEST

Our strategy for ensuring the high quality of the augmented dataset is inspired by rejection sampling [41, 45]. [3] In order to generate a sample from a distribution with the probability density function (pdf) f, the rejection sampling technique generates sample points under an upper envelope of f and rejects it if the sample point does not fall under f. We use a similar strategy: when the foundation model generates a new tuple, we accept it only if it passes the outlier detection test (§ 3.1) and the quality evaluation (§ 3.2). Otherwise, the generated tuple will be rejected, and we will try again.

3.1 Outlier (Novelty) detection test

When augmenting a dataset, it is crucial to ensure that the augmented tuples align with the underlying data distribution ξ and are not outliers. For instance, if the dataset consists of wide-shot images in an office workplace, the generated tuples should also fit this context. The first issue is that ξ is unknown. Besides, it is not clear how to quantify and represent the distribution, while relying on the foundation model's imagination could generate outlier tuples.

We utilize the vector representation (aka embedding) of the tuples for representing the distribution ξ. Given a tuple t_i, let $\vec{v}(t_i) = \vec{v}_i = \langle v_1, v_2, \cdots, v_\kappa \rangle$ be its embedding. We assume the embeddings are accurate. That is, the cosine similarity between the embeddings represents the semantic similarity between two tuples. Formally, the similarity of two tuples t_i and t_j can be computed as $\mathcal{S}_{im}(t_i, t_j) = \cos \angle(\vec{v}_i, \vec{v}_j)$. Now, in the embedding space, let ξ be the probability distribution from which $\mathcal{D}$ is sampled. Hence, the probability that a tuple t is sampled is $Pr_\xi(t)$. We use $\vec{\mu}_\xi$ to represent the mean of ξ. Since the tuples in $\mathcal{D}$ are iid samples from ξ, those can be used for estimating $\vec{\mu}_\xi$. Let $\vec{v}_c$ be the sample mean of the representation vectors in $\mathcal{D}$. That is, $\vec{v}_c = \frac{1}{m} \sum_{t_i \in \mathcal{D}} \vec{v}_i$. Assuming that n (the size of $\mathcal{D}$) is large enough, based on the central limit theorem, we can estimate $\vec{\mu}_\xi$ as $\vec{v}_c$.

To ensure that each generated tuple adheres to the underlying distribution ξ, i.e., it is not an outlier based on ξ, we employ a one-class support vector machine (OCSVM) approach proposed by Scholkopf et al. [70] as a quality control mechanism.

Formally, given a set of training embeddings $\{\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_n\}$ representing tuples drawn from ξ, the OCSVM aims to learn a decision boundary that separates the majority of these embeddings from the origin in the feature space. This boundary implicitly defines a region that characterizes the "normal" or acceptable embeddings. To find this boundary (hyperplane), the following optimization problem is proposed:

$$\min_{\mathbf{w},\rho,\epsilon} \quad \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu n}\sum_{i=1}^{n} \epsilon_i - \rho$$

$$\text{Subject to} \quad \mathbf{w} \cdot \phi(\vec{v}_i) \geq \rho - \epsilon_i, \quad \epsilon_i \geq 0, \quad i = 1, 2, \ldots, n$$

[3] We would like to clarify that, while inspired by rejection sampling, the proposed rejection test is **not** a rejection sampling approach, and it does not aim to keep the distribution constant; hence, it provides no guarantee to preserve the distribution. Please refer to § 8.2 for more details.

where:

- $\mathbf{w}$ is the hyperplane normal vector (weight vector)
- ν is an upper-bound on the fraction of outliers and a lower bound on the fraction of support vectors (SV)
- ϕ is a feature mapping function that maps embeddings into a higher-dimensional space (e.g., the radial basis function kernel)
- ρ is a parameter controlling the margin of the decision boundary
- ϵ_i are slack variables allowing for a soft margin

To evaluate a generated tuple t_g with embedding $\vec{v}_g$, we project it into the feature space using the kernel function and compute:

$$f(\vec{v}_g) = \mathbf{w} \cdot \phi(\vec{v}_g) - \rho$$

If $f(\vec{v}_g) \geq 0$, the tuple is deemed acceptable, falling within the normal region defined by the OCSVM. If $f(\vec{v}_g) < 0$, it is an outlier and is rejected.

3.2 Quality evaluation

Foundation models have emerged as strong tools for creating high-quality multi-modal data. Still, due to the randomized nature of their generation process and the task-specific difficulties of various queries, some of the generated tuples may not look *realistic to human beings*.

Therefore, it is necessary to evaluate the quality of the generated tuples before augmenting them to the dataset. This evaluation, however, is qualitative and subjective. That is, the answer to "does this tuple look realistic?" may vary from one person to the other. However, if there is a high correlation between raters, *the probability* of the answer being positive reflects the quality of the tuple.

Using this observation, we model the quality of a tuple as a *Bernoulli random variable*. Specifically, let p be the probability that a human evaluator labels a randomly sampled (real) tuple from the distribution ξ as "realistic". In other words, with probability $(1-p)$, the evaluator will mistakenly label the tuple as "unrealistic". We can then define the Bernoulli variable ϕ, which is one of a real tuple labeled as realistic by a human evaluator and zero otherwise. Therefore, the pdf of ϕ for the randomly sampled (real) tuples is,

$$f(\phi) = \begin{cases} p & \phi = 1 \\ (1-p) & \phi = 0 \end{cases} \tag{1}$$

The mean and the variance of this Bernoulli distribution are $\mu_\phi = p$ and $\sigma^2_\phi = p(1-p)$, respectively.

Let p' be the probability that a randomly selected human evaluator labels an AI-generated tuple as realistic. Assuming that the human evaluator is better than random labeling, $p' < p$. We use this observation to develop hypothesis testing. We discard an AI-generated tuple if we reject the null hypothesis that p' equals p, i.e., $\mathcal{H}_{null} : p' = p$. Then, considering the lower tail test, the alternative hypothesis would be $\mathcal{H}_{alt} : p' < p$.

To do so, we first obtain a sufficiently large sample set U of evaluations, where each sample is drawn using a randomly selected evaluator and a random (real) tuple from $\mathcal{D}$. Let m_U be the sample mean of U. Since U is sufficiently large, we can estimate $p = \mu_\phi$ with m_U. Now, for a generated tuple t, consider a sample set U_t of N evaluations of t, each using a randomly selected evaluator. We assume a (small) fixed-size budget for each generated tuple. Let m_t and s_t be the sample mean and the standard deviation for U_t. Since N is small, we use the Student's t-test [30]. Specifically,

$$t_{N-1} = \frac{m_t - p}{s_t / \sqrt{N}}$$

Next, using the t-table, we obtain the left-sided p-value and evaluate its significance level. If the p-value is smaller than a significance goal α, we reject the null hypothesis (discard the generated tuple).

4 COMBINATION SELECTION

Our overall goal is to use the foundation model and generate a minimal set of synthetic tuples to resolve inadequate coverage for the most general subgroups (MUPs with the smallest levels). Therefore, we consider an iterative approach, where we resolve the MUPs at the smallest level during each iteration. Given a dataset $\mathcal{D}$, let $\mathcal{M}$ be the set of MUPs, and let $\mathcal{M}^*$ be the set of MUPs with the minimum level. That is $\mathcal{M}^* = \left\{M \in \mathcal{M} \mid \ell(M) = \min_{M' \in \mathcal{M}} \left(\ell(M')\right)\right\}$. For each MUP $M \in \mathcal{M}^*$, let us define its gap $\delta(M) = \tau - |\mathcal{D} \cap M|$; i.e., the coverage threshold minus the current coverage of M in $\mathcal{D}$. In other words, $\delta(M)$ is the minimum number of synthetic tuples matching M we need to obtain before it is covered. Also, for each combination $c_i \in \bigtimes_{k=1}^{d} dom(x_k)$, let σ_i be the number of synthetic tuples from that combination. Then, the "Combination-Selection" problem is to assign the values of $\sigma_i > 0$ such that (i) for each MUP $M \in \mathcal{M}^*$, at least $\delta(M)$ generated tuples match it, and (ii) sum of all σ_i values is minimized. Formally,

$$\begin{aligned} \min \quad & \sum_{c_i} \sigma_i \\ \text{Subject to} \quad & \sum_{c_i \in \text{match}(M)} \sigma_i \geq \delta(M), \quad \forall M \in \mathcal{M}^* \end{aligned}$$

Theorem 1. Combination-Selection *is* NP*-hard.*[4]

Since Combination-Selection is NP-hard, we design an approximation algorithm for this step. Our algorithm follows the *greedy* scheme. The algorithm is iterative, where at each iteration it finds the combination that matches the maximum number of remaining MUPs in $\mathcal{M}^*$. We utilize the inverted index and the tree data structure proposed in [5] for finding c. The algorithm then finds the minimum gap γ in the MUPs matching c and increases the number of instances from c by γ. It also updates the gaps for the MUPs matching c and remove the ones that reach to a gap of zero from $\mathcal{M}^*$. The pseudo-code of the Greedy algorithm is provided in the technical report [35].

Theorem 2. *The approximation ratio of the* Greedy *approach is* $\log(\eta)$*, where* $\eta = \sum_{M \in \mathcal{M}^*} \delta(M)$.

5 GUIDE TUPLE SELECTION

Given a combination c, we would like to generate a tuple that matches c and is likely to pass the rejection sampling tests. Therefore, we want to make sure that (a) the generated tuple is not an outlier according to the distribution ξ represented by $\mathcal{D}$ and (b) the generated tuple has high quality and passes the quality evaluation. So, instead of relying on the foundation model's imagination, we provide a "guide" for the generation process. Recall from § 2.2, that

[4] Proofs are provided in the technical report [35].

the guide is a pair (t, m), where t is a tuple and m is a mask. In the following, we propose various guide-tuple selection strategies.

In the image context, a mask indicates the parts of the guide image to be regenerated. Our mask generation (§ 5.4) involves cropping the foreground of t using mask m, ensuring that the foundation model regenerates only the portions specified by the mask m.

5.1 Random-Guide Strategy

The random guide strategy focuses on the first requirement that the generated tuple should not fall out of the distribution ξ, represented by $\mathcal{D}$. Hence, it selects the guide tuple uniformly at random from the dataset without taking into account the target combination c.

Theorem 3. *Let D' be the set of tuples generated by the model $\mathcal{F}$, using the random-guide strategy. Assume the perturbation by $\mathcal{F}$ on a guide tuple to generate the output is small. Formally, for an output x, assume $|p(\mathsf{x}) - q(\mathsf{x})| < \epsilon$, where $p(.)$ and $q(.)$ are the probability density functions of ξ and the distribution ξ' represented by D', respectively. Then, the expected KL-divergence between ξ and ξ' is bounded by ϵ.*

While the random-guide strategy is appropriate for passing the outlier-detection test, it ignores the second requirement of passing the quality test. We experimentally show in § 6, that tuples generated based on this strategy have a lower chance of passing quality evaluation. Therefore, we propose strategies that are less random in their guide-tuple selection. Note that as a result, Theorem 3 is no longer valid for the subsequent strategies.

5.2 Similar-Tuple Strategy

The similar-tuple strategy creates a pool of similar combinations to the target combination c. Combinations c_1 and c_2 are considered *similar* if (a) they are *siblings* (i.e., their values differ in exactly one attribute) and (b) one of the following conditions is satisfied. Let d_i be the attribute on which c_1 and c_2 differ. If d_i is non-ordinal, then c_1 and c_2 are considered similar. However, if d_i is ordinal, then the distance between c_1 and c_2 should be 1 to be considered similar. Formally, for two sibling combinations c_1 and c_2 that differ in attribute d_i:

$$\text{similar}(c_1, c_2) = \begin{cases} \textit{false} & \text{if } d_i \text{ is ordinal and } |c_1[d_i] - c_2[d_i]| > 1 \\ \textit{true} & \text{otherwise} \end{cases}$$

Subsequently, it selects a guide tuple from the pool of similar combinations, assigning weights to each element based on the number of tuples in $\mathcal{D}$ that adhere to that particular combination. Let two combinations be *"sibling"* if they differ in exactly one attribute. Formally, the pool of similar combinations can be defined as follows:

$$S = \left\{ c \in \textit{sibling}(c_i) \;\middle|\; \text{similar}(c_i, c) = \textit{true} \right\}$$

For each combination $c_i \in S$, let $|c_i|$ be the number of tuples in $\mathcal{D}$ matching it. That is, $|c_i| = |c_i \cap \mathcal{D}|$. We assign the sampling weight of each combination proportional to their normalized size: $w_i = \frac{|c_i|}{\sum_{c_j \in S} |c_j|}$, $\forall c_i \in S$. The similar-tuple strategy then selects a combination $c_i \in S$, randomly with probability w_i. It then returns a random sample from the pool tuples in $\mathcal{D}$ that match c_i as the guide tuple. Using w_i as the weight ensures equal sampling probability for all tuples that match a combination in S.

This strategy considers tuples in the selection pool that closely resemble the target combination, differing in only one attribute of interest. It also excludes the tuples with the exact combination as the target combination. This exclusion is intentional to deal with the fact that the target combination c is not well-represented in the dataset. Hence, picking guide tuples from this group might make the chosen tuples look too similar. By considering combinations that are similar but not exactly the same as the target one, the strategy aims to make sure we get a more varied and representative set of guide tuples for the generation process.

5.3 Modeling as Contextual Multi-armed Bandit

Our LinUCB strategy models the guide tuple selection problem as a *contextual multi-armed bandit* problem [15], and uses *Contextual Upper Confidence Bound* for solving it [54]. Specifically, it models each attribute as a bandit arm. Then given a target combination c, it selects an arm to pull (i.e., an attribute to modify), aiming to maximize the obtained reward. In each iteration, we have the opportunity to pull only one arm, signifying the ability to alter one attribute value within the target combination to a new value. The reward obtained from pulling that arm is then observed. The objective is to learn the optimal arm to pull for a given combination c over successive iterations.

To provide further clarification, let us discuss an example within the context of images. Consider a dataset with attributes of interest including `gender`, `race`, and `age-group`. The foundation model $\mathcal{F}$ may perform better in modifying the race of a subject compared to altering their age group for specific combinations (e.g., Asian female adults). However, its performance may vary for other combinations. LinUCB aims to systematically explore different arms to pull (e.g., changing race, gender, or age group) and exploit the arm, yielding the highest reward over time.

Formally, we formulate the guide tuple selection problem as a contextual multi-armed bandit problem. We consider the attributes of interest, denoted as $\mathbf{x} = \{x_1, x_2, \ldots, x_d\}$, as arms of the bandit $\mathbf{a} = \{a_1, a_2, \ldots, a_d\}$. The *context*, AKA the *feature vector*, is then defined as a one-hot vector $\mathbf{f}$ representing combinations, where 1 is assigned for the input combination c and 0 for all other elements. Let $k = |\bigtimes_{i=1}^{d} dom(x_i)|$ be the number of possible combinations. The size of the feature vector $\mathbf{f}_{s,a}$ is $(k \times 1)$ where s denotes the time step of the algorithm.

We define the reward function based on whether a generated tuple passes the rejection sampling tests. Let *pass*() be a binary function that is false if a generated tuple is rejected. Then,

$$r_{s,a} = \begin{cases} 1 & \text{if } \textit{pass}() = \textit{true} \\ 0 & \text{otherwise} \end{cases}$$

We adopt "LinUCB with Disjoint Linear Models" [54] to balance exploration and exploitation. At every iteration s, for every arm $a \in \mathbf{a}$, given the context $\mathbf{f}_{s,a}$, LinUCB computes confidence intervals for the expected reward and selects the arm with the maximum upper bound of reward to be explored next.

We assume that the expected reward of an arm a is linear in its k-dimensional input context $\mathbf{f}_{s,a}$ with some unknown coefficient vector $\theta^*_{s,a}$, formally given by: $E[r_{s,a}|\mathbf{f}_{s,a}] = \mathbf{f}^\top_{s,a}\theta^*_{s,a}$

Assuming that m is the number of times arm a has been pulled so far ($s \geq m$), we define $\mathbf{F}_{s,a}$ with size $(m \times k)$ as the matrix consisting

of all previously observed contexts for arm a.

$$\mathbf{F}_{s,a} = \left[\mathbf{f}_{1,a}^\top, \cdots, \mathbf{f}_{m,a}^\top\right]^\top$$

We also have a vector of observed rewards from pulling arm a.

$$\Gamma_a = \left[r_{1,a}, \cdots, r_{m,a}\right]^\top$$

Vector $\mathbf{b}_a$ is defined as: $\mathbf{b}_a = \mathbf{F}_{s,a}^\top \Gamma_a$

Using the Ridge regression estimator, we can estimate the coefficients of each arm a as: $\hat{\theta}_{s,a} = (\mathbf{F}_{s,a}^\top \mathbf{F}_{s,a} + \mathbf{I}_k)^{-1}\mathbf{b}_a$

Thus, in each iteration s of the algorithm, we select arm a_s using:

$$a_s = \underset{a \in \mathbf{a}}{\operatorname{argmax}} \left(\mathbf{f}_{s,a}^\top \hat{\theta}_{s,a} + \alpha\sqrt{\mathbf{f}_{s,a}^\top \mathbf{A}_a^{-1}\mathbf{f}_{s,a}}\right)$$

where $\mathbf{A}_a = \mathbf{F}_{s,a}^\top \mathbf{F}_{s,a} + \mathbf{I}_k$ and α is a hyper-parameter to balance exploitation and exploration.

After pulling arm a_s in iteration s, we observe the reward $r_{s,a_s} \in \{0, 1\}$ where 1 indicates that the generated tuple t has passed quality and data distribution tests. We can update matrices $\mathbf{A}_{a_s}$ and $\mathbf{b}_{a_s}$ as:

$$\mathbf{A}_{a_s} \leftarrow \mathbf{A}_{a_s} + \mathbf{f}_{s,a_s}\mathbf{f}_{s,a_s}^\top$$
$$\mathbf{b}_{a_s} \leftarrow \mathbf{b}_{a_s} + r_{s,a_s}\mathbf{f}_{s,a_s}$$

The pseudo-code of the LinUCB strategy for guide-tuple selection is presented in the technical report [35].

5.4 Mask Delineation

In the context of images, once the guide tuple t is selected, we delineate the foreground subject using a mask. This mask serves as an indicator, specifying the regions to be cropped and regenerated from the tuple t. The delineation of the border around the subject can be achieved with different levels of precision. A precise border sketch preserves more space from the original context, potentially resulting in a higher acceptance rate for the data distribution test. However, it may limit the foundation model's imagination capacity and lead to a lower acceptance rate for the quality evaluation test. We propose three levels of mask sketch accuracy: *accurate*, *moderate*, and *imprecise* (Figure 2).

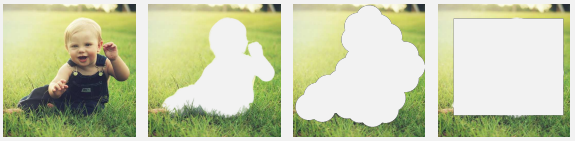
5.4.1 Accurate mask delineation. This represents the highest level of precision in mask delineation, achieved by utilizing the off-the-shelf background remover tool `rembg`.

5.4.2 Moderate mask delineation. To obtain a moderately delineated mask, we extend the border of the mask drawn in 5.4.1 by 10 percent of the image size. This extension is implemented using circles, with each point on the mask border being surrounded by a circle of radius equal to 10% of the image width.

5.4.3 Imprecise mask delineation. For an imprecise mask delineation, we expand the border of the mask drawn in 5.4.1 to form a rectangular area. This rectangle encompasses the previous mask, providing a less precise but more inclusive delineation.

6 EXPERIMENTS

This section presents results from experiments to evaluate the efficacy of our proposed system, Chameleon. We present proof-of-concept demonstrations, comprehensive performance evaluations of our system, and comparison against state-of-the-art baselines across *four distinct tasks*. Each task leverages a specific benchmark.



(a) (b) (c) (d)

Figure 2: Illustration of various guide image (a) masks: (b) Accurate (c) Moderate (d) Imprecise mask

① We investigate the system's performance in passing the outlier test (§ 3.1) and quality evaluation test (§ 3.2), using various mask delineation levels and guide tuple strategies.
② We analyze the performance of the proposed Greedy approach for combinations selection.
③ We demonstrate the extensibility of Chameleon by adapting it to handle (a) *textual data* and (b) *representation rate* for detection lack of representation. We will investigate the impact of synthetic text augmentation on a (c) *sentiment analysis task*.
④ Our quality test (§ 3.2) involves human evaluators. In presence of automated tools that perform similarly to human evaluators, this step could be automated. In the technical report [35] we explore alternative options to replace human evaluators. We analyze the results obtained from different quality assessment tools and compare them to the ground truth by human evaluators. In our experiments, none of the assessed algorithms performed satisfactorily in detecting unrealistic images.

6.1 Experimental Setup

Details such as implementation and hardware configurations are provided in the technical report [35].

6.1.1 datasets. We used three benchmark image datasets and one textual dataset in our experiments. For image datasets, we used distinct subsets from UTKFace [89], FERETDB [66], Animal-10 [26]. We used Emotions [63] dataset in our text-based experiments. UTKFace encompasses over 20,000 face images with annotation of age, gender, and ethnicity. The images in UTKFace cover large variations in pose, facial expression, illumination, occlusion, resolution, and other factors. Conversely, FERETDB comprises 1199 individual images annotated with gender and ethnicity and serves as a standardized facial image database for researchers to develop algorithms and report results. Animal-10 contains approximately 26,000 images of animals belonging to 10 distinct categories. The images depict animals in their natural habitats. This dataset provides a valuable resource for tasks related to animal image classification.

To showcase the extensibility of our work beyond image data, we used Emotions, a collection of English Twitter messages annotated with six fundamental emotions: anger, fear, joy, love, sadness, and surprise. The task for this dataset is sentiment analysis.

6.1.2 Foundation Model and the Monetary Cost. We use DALL·E 2 Foundation model for generating images from prompt as it is (at the time of experiments) the most widely available Image Generation model with public API KEY available. A total of 3657 distinct images were generated using DALL·E 2 throughout the development and experimental phases. For text generation tasks, we used GPT-4 with

Table 1: Demographic groups distribution in FERETDB

	Male	Female	Total
White	247	171	418
Black	29	26	55
Asian	74	41	115
Hispanic	22	18	40
Middle Eastern	27	6	33
Total	399	262	661

Table 2: Demographic groups distribution in UTKFace

	Male	Female	Total
Child (0-7)	551	656	1207
Adolescent (7-20)	356	210	566
Young (20-40)	4440	3463	7903
Adult (40-60)	1563	2524	4087
Elderly (60+)	1060	1229	2289
Total	7970	8082	16052

a total of 40,656 prompts. *The total cost for generating these images and texts amounted to (US)$91.9.*

6.1.3 Evaluated Algorithms and Baselines. The following are the baselines and evaluated algorithms designed for each task.

- We use Reweighing [51] and SMOTE [23] as two baselines for our experiment. Reweighing adjusts the weight of classes in the classifier, so in the training process, all classes have the same amount of training points. SMOTE over-samples the minority by generating similar synthetic data points in embedding space.
- For Task ①, we consider no-guide tuple, similar-tuple and random-guide strategies, as baselines for our experiments to compare against LinUCB. Additionally, we consider accurate, moderate, and imprecise mask delineation levels for evaluation.
- For Task ②, we consider two baselines to compare against the Greedy algorithm. The first baseline, called random, randomly selects the next combination to be generated. The second baseline (Min-Gap), first identifies a MUP M that requires the minimum number of instances to be covered. It then generates a combination that matches M. Both random and Min-Gap baselines continue until the MUPs at the smallest level are resolved.
- For our proof of concept, we designate the Tensorflow Keras CNN Model for precision, recall, and F1 score comparisons.

6.1.4 Performance metrics. In Task ①, our performance metrics include the Quality Test Acceptance Rate (QTAR) and Outlier detection Test Acceptance Rate (OTAR) of the generated samples. For Tasks ②, our primary performance metric is the number of queries incurred by $\mathcal{F}$, representing the cost of image generation. Finally, in our proof of concept, our metric focuses on the precision, recall, and F1 score of the trained model on the test dataset.

6.2 Proof of Concept

Our approach for resolving inadequate coverage of minorities in a multi-modal dataset is by augmenting it with *synthetically generated data*. We start our experiments by investigating the *feasibility*, *effectiveness*, and *efficiency* of this data-repair approach, and compare our approach with state-of-the-art baselines. To do so, we illustrate the impact of lack of coverage resolution using Chameleon in several downstream machine learning tasks. We first measure the overall performance and the unfairness (in the form of performance disparity) of each model/task trained on the original dataset for under-represented groups. Next, we repair each dataset, using Chameleon (and the other baselines), and repeat the process to see if the unfairness issues are reduced. Subsequently, we monitor the so-called *"price of fairness"*, i.e., the reduction in the overall performance as a result of unfairness reduction (data repair). We investigated this using two different datasets. The first dataset, FERETDB, comprises professional headshots of individuals from various races and genders. The second dataset, Animal-10, includes images from 10 different animal categories captured in the wild, with no specific requirements, in contrast to FERETDB.

6.2.1 FERETDB. Our experiment employs the entire FERETDB dataset as input, with detailed demographic group counts provided in Table 1. While the dataset has a reasonable coverage for both `male` and `female` genders, the racial groups `Black`, `Hispanic`, and `Middle eastern` are not covered (using the coverage threshold $\tau = 100$). We trained a race-predicting Convolutional Neural Network (CNN) model using this dataset. First, we observed a high overall performance of the model, with precision, recall, and F1-score being, 0.68, 0.66, and 0.67, respectively. Moreover, the model shows similar performance on both (covered) genders. However, as reflected in Table 3 (the "classifier performance on FERETDB" column), the model significantly underperforms for the uncovered groups. For example, while the overall F1-score is 67%, it is as low as 20%, 03%, and 0%, for the `Black`, `Hispanic`, and `Middle eastern` groups.

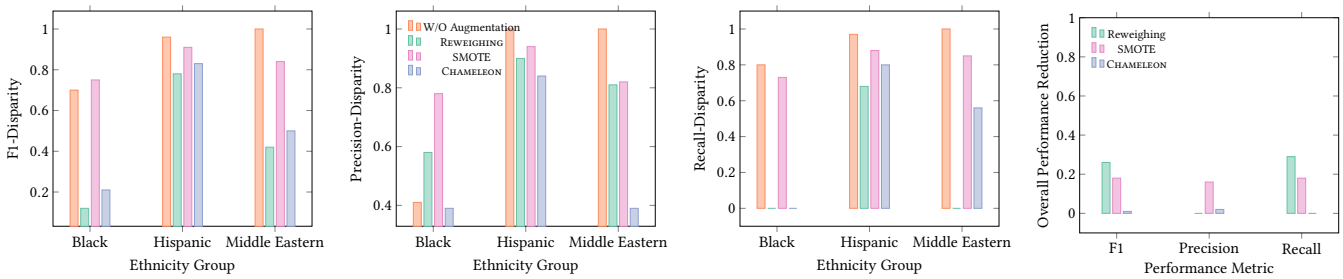
To evaluate if resolving the lack of coverage for these groups using Chameleon helps to reduce the gaps, we employ it with the Greedy combination selection algorithm, Moderate mask delineation level, and the LinUCB approach to resolve the level-1 MUPs, i.e., the three uncovered racial groups. In total, Chameleon issued *307* queries to the foundation model, of which *231* pass both the quality and outlier tests. That is, **75%** of the generated images passed the rejection tests. We refer to the augmented dataset as "Repaired". Utilizing DALL·E 2 to generate 307 images incurred a total cost of $4.91 ($0.016 per image).

Next, we retrain the CNN using the Repaired dataset. Notably, the test data remains the same for both experiments and only contains real images. First, as expected, from Table 3 (the "classifier performance on Repaired" column), one can notice a slight decrease in the overall performance of the model as a result of data augmentation. On the other hand, though, it is evident that *the performance of the model significantly increased for all under-represented groups across all performance metrics.* For example, looking at the F1 scores, the average performance improvement is more than 25%.

We also reran the same experiments using Reweighing and SMOTE to investigate the effectiveness of these methods in mitigating bias. Both methods significantly decrease overall model performance, with drops of 25% and 18% in F1-Score, respectively. Despite this substantial trade-off for fairness, the improvement in performance for minority groups is not as substantial as that achieved with Chameleon.

Table 3: Illustration of repairing lack of coverage and its effects on FERETDB

Ethnicity Groups	W/O Augmentation			Reweighing			SMOTE			Chameleon		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Overall	0.68	0.66	0.67	0.72	0.37	0.41	0.51	0.48	0.49	0.66	0.66	0.66
Black	0.40	0.13	0.20	0.30	0.43	0.36	0.11	0.13	0.12	0.40	0.74	0.52
Hispanic	0.05	0.02	0.03	0.07	0.12	0.09	0.03	0.06	0.04	0.10	0.12	0.11
Middle Eastern	0.00	0.00	0.00	0.14	0.86	0.24	0.09	0.07	0.08	0.40	0.29	0.33



(a) F1-Disparity **(b) Precision-Disparity** **(c) Recall-Disparity** **(d) Price of Fairness**

Figure 3: Unfairness (disparate performance) reduction for the uncovered groups in the FERETDB dataset after data repair using Chameleon, along with the price of fairness (overall performance reduction).

Figures 3a, 3b, and 3c show the model unfairness in the form of Disparate Performance (F1-Disparity, Precision-Disparity, and Recall-Disparity) across the under-represented groups in the FERETDB dataset, before and after the data repair, for Chameleon and the baselines. The performance disparity for an under-represented group is computed as its performance ratio gap with the overall model performance. For example, if the overall performance of the model for a metric p (e.g., F1-score) is ρ_{all} and for a group g is ρ_g, the unfairness is computed as

$$p\text{-Disparity}(g) = \max\left(0, 1 - \frac{\rho_g}{\rho_{all}}\right)$$

The figure demonstrates a clear reduction in disparities for all underrepresented groups after the repair process, which showcases the effectiveness of the data augmentation using Chameleon. For example, the F1-disparity for `Black` decreased from 70% to 21%.

Price of Fairness. Due to the trade-offs between the model performance and fairness, improving fairness is usually associated with a reduction in the overall model performance, which is known as the *price of fairness*. As we saw earlier, our data repair approach using Chameleon could significantly reduce the model performance disparities for the under-represented groups. This, however, comes at the cost of a slight model performance reduction. Figure 3 shows this cost as the reduction in overall Precision, Recall, and F1-Score after data augmentation. The price of fairness, as reflected in various metrics, is modest compared to the substantial improvement in fairness achieved for under-represented groups.

6.2.2 Animal-10. In this experiment, we utilized the Animal-10 dataset, which contains over 26,000 images across 10 different animal categories. Figure 4 illustrates the demographic distribution of the input dataset. We trained a CNN classifier (a miniVGGNet) from scratch to predict the animal type in each image.

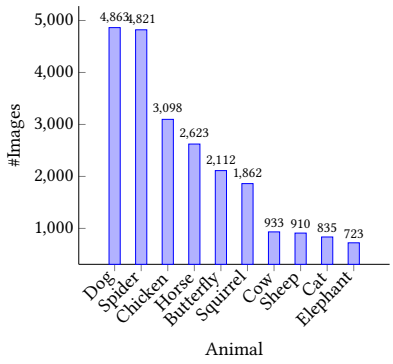


Figure 4: Groups distribution in training subset of Animal-10

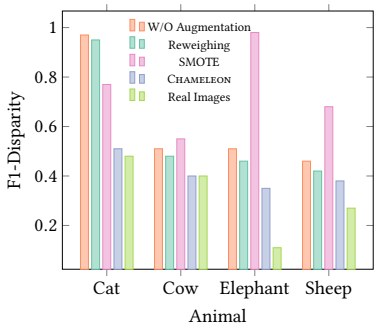
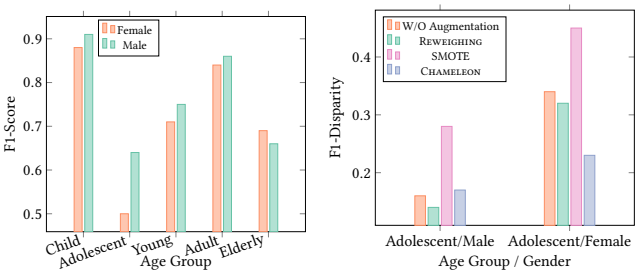


Figure 5: Animal-10: Unfairness comparison with baselines

As indicated in Figure 4, there are four minority groups (Cow, Sheep, Cat, and Elephant) in the dataset, and the CNN model performs significantly worse for these groups compared to others. We augmented these minority groups using Chameleon with $\tau = 1200$

Table 4: Illustration of repairing lack of coverage and its effects on ANIMAL-10

Animal Type	W/O Augmentation			REWEIGHING			SMOTE			CHAMELEON			Real Images		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Overall	0.62	0.64	0.61	0.59	0.60	0.59	0.81	0.61	0.65	0.62	0.63	0.62	0.62	0.62	0.62
Cow	0.45	0.23	0.30	0.35	0.30	0.32	0.23	0.41	0.29	0.37	0.38	0.37	0.44	0.31	0.37
Sheep	0.50	0.25	0.33	0.40	0.30	0.34	0.12	0.91	0.21	0.33	0.45	0.38	0.54	0.39	0.45
Cat	0.33	0.01	0.02	0.31	0.02	0.03	0.13	0.17	0.15	0.44	0.23	0.30	0.34	0.30	0.32
Elephant	0.69	0.20	0.30	0.49	0.24	0.32	0.02	0.00	0.01	0.43	0.37	0.40	0.56	0.53	0.55



(a) F1-Scores **(b) F1-Disparity**

Figure 6: UTKFACE: Performance and unfairness comparison

and retrained the classifier to observe whether resolving the lack of coverage with synthetic data can mitigate the unfairness issue. Additionally, we compared the results of CHAMELEON augmentation with REWEIGHING [51] and SMOTE [23] as baselines. In this experiment, we also introduced an upper bound on model performance by evaluating the model on a dataset augmented with real images. This comparison allows us to assess how closely CHAMELEON-generated images approximate the quality of real images.

As indicated in Table 4, all approaches yield an overall F1-Score of approximately 0.6, but the performance for under-represented groups is significantly worse. The REWEIGHING approach increases the performance for minority groups by only 2-3%, while also decreasing the overall performance by a similar margin. This minimal improvement is likely because REWEIGHING does not provide any new information about animal types to the classifier.

SMOTE shows mixed results, improving performance for only one under-represented groups (Cat), while worsening it for others. This indicates that SMOTE is not a robust solution for addressing unfairness in imbalanced image datasets. The likely reason is that SMOTE perturbs in the vector space without knowledge of which vector values correspond to which image attributes, potentially generating meaningless images or amplifying irrelevant attributes.

In contrast, augmentation with CHAMELEON improves results for all categories by approximately 10% while maintaining overall performance. When augmenting the dataset with real images, the difference of improvements between CHAMELEON and Real images across different groups is less than 5%, demonstrating that CHAMELEON provides high-quality and inclusive results.

6.2.3 UTKFace. In previous experiments, the task was predicting the sensitive attribute. In this experiment, the prediction attribute is different from the sensitive attribute. We used a subset of the UTKFACE dataset to train an age group predictor model and monitored its performance across different genders. We then augmented

Table 5: Performance of various Guide-selection algorithms

Guide Tuple Strategy	Mask Delineation Level		Quality Test Acceptance Rate		Outlier Test Acceptance Rate ($\nu = 0.3$)	
			$\alpha = 0.1$	$\alpha = 0.4$	Linear	RBF
No Guide	-		0.90	0.81	0.53	0.44
Random-Guide	Accurate		0.69	0.51	0.70	**0.74**
	Moderate		0.85	0.70	0.70	0.70
	Imprecise		0.90	0.70	**0.73**	0.62
		Avg:	0.81	0.64	0.71	0.69
Similar-Tuple	Accurate		0.88	0.69	0.65	0.67
	Moderate		0.90	0.75	0.64	0.58
	Imprecise		0.85	0.68	0.65	0.53
		Avg:	0.88	0.71	0.65	0.59
LinUCB	Accurate		0.90	0.81	0.63	0.61
	Moderate		0.91	0.88	0.64	0.58
	Imprecise		**0.96**	**0.96**	0.63	0.54
		Avg:	0.92	0.88	0.63	0.58

the dataset for the minority group (Female) and retrained the CNN model to observe improvements in the F1-score for this task. Figure 6a presents the F1-scores in each age group performance for both genders. The overall F1-score for this model is 0.78. As shown, the performance for the Female group is substantially lower than for the Male group in the Adolescent age group. We generated 200 images of Female Adolescents using CHAMELEON. We measured the reduction in disparity for this group using CHAMELEON and compared it to REWEIGHING and SMOTE. As presented in 6b, the F1-disparity for the Adolescent/Female group decreased the most using CHAMELEON. In contrast, the REWEIGHING approach proved ineffective, and SMOTE actually worsened the disparity for the Adolescent group.

6.3 Performance Evaluation

6.3.1 Investigating the Influence of Mask Levels and Guide Image Selection on Quality Assessment ①. This study explores the impact of different mask delineation levels and guide-tuple selection strategies on the performance of generated tuples in passing rejection sampling tests. A total of *37 individuals* participated as the human evaluators for the quality evaluation test.

To guarantee an inclusive evaluation, we intentionally constructed a challenging subset of the UTKFACE dataset. This subset was designed to encompass Maximal Uncovered Patterns (MUPs) for all races and genders. Within each age group, we introduced two distinct ℓ_3 MUPs, each representing a different combination of gender and race (e.g., White male adult, Indian female adult). This approach ensures comprehensive coverage of various races, genders, and age groups during image generation. The exclusive use of ℓ_3 MUPs guarantees that all experiments will generate identical combinations,

effectively eliminating any potential randomness or variability in the results. In total, we introduced 16 MUPs within the UTKFace subset with $\tau = 10$. Our objective was to resolve all MUPs in the subset for all possible combinations of guide tuple selection strategies and mask delineation levels. We then compared the Quality Test Acceptance Rate (QTAR) and Outlier Test Acceptance Rate (OTAR) for each combination. We generated a total of 831 images for these experiments and employed 27 human evaluators to assess their quality. Each evaluator received 200 images, presented in 8 pages of 25 images each. They were instructed to identify images that appear *unrealistic* to human beings.

In a separate experiment, we employed 10 human evaluators to estimate the probability p (Equation 1) that an evaluator labels a real image as realistic. For UTKFace, the probability p was estimated as 0.86. For each setting, the Quality Test Acceptance Rate (QTAR) is defined as the number of images passing the quality test (§ 3.2) divided by the total number of generated images. To explore the impact of evaluation stringency on QTAR, we calculated it for two significance levels, $\alpha = 0.1$ and $\alpha = 0.4$. A higher α value signifies a stricter acceptance policy, demanding a greater agreement among evaluators regarding the image's realistic appearance.

For $\alpha = 0.4$, acceptance closely aligns with *unanimous agreement* among evaluators, whereas $\alpha = 0.1$ approximates a *majority vote*, accepting images deemed realistic by over half of the evaluators. This distinction results in a trade-off between quality and quantity, with $\alpha = 0.4$ yielding a smaller pool of images passing the test, with potentially higher overall quality. Table 5 presents the calculated QTAR values for the designed UTKFace subset under both significance levels. The analysis of Quality Test Acceptance Rate (QTAR) reveals that the LinUCB guide tuple selection strategy consistently outperforms No Guide, Similar-Tuple, and Random-Guide across both significance levels ($\alpha = 0.1$ and $\alpha = 0.4$). This performance gap widens further as the quality assessment becomes stricter (higher α). Notably, images generated using LinUCB exhibit demonstrably higher overall quality. Further investigation into the interplay between guide-tuple selection strategies and mask delineation levels uncovers interesting trends. For both significance levels, Moderate and Imprecise mask delineation levels achieve superior QTAR compared to the Accurate level. This finding aligns with our initial expectations. Precisely cropping the foreground subject restricts the foundation model's creative freedom, potentially leading to unnatural entities generated to fill the cropped space. Conversely, Moderate and Imprecise levels provide greater flexibility, allowing the foundation model to generate new objects more naturally and potentially contributing to improved image quality.

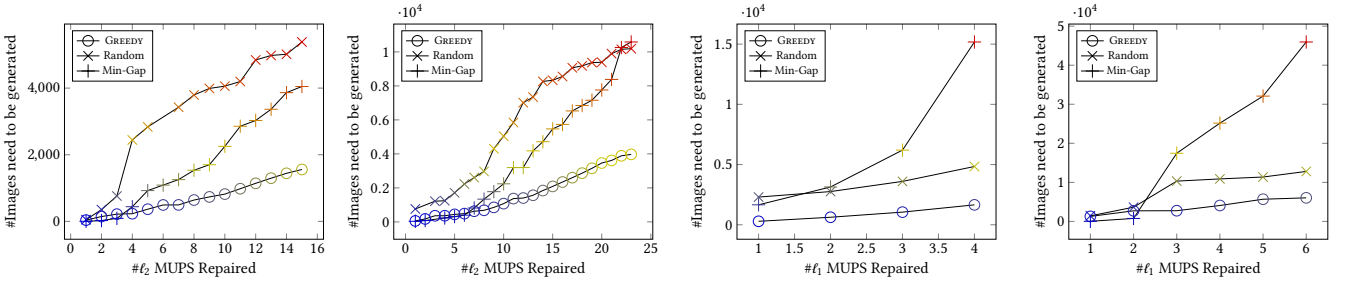
Next, we move to the outlier detection test (§ 3.1). OTAR, defined as the proportion of images passing the test, assesses the semantic integrity of generated images with respect to the underlying data distribution. We use MobileNetV3[47] as the embedder and OCSVM ($\nu = 0.3$) with two kernels (RBF and Linear) for training. The choice of the kernel impacts the performance of the OCSVM, with the RBF kernel often capturing complex relationships in the data, while the Linear kernel assumes linearity.

The No-Guide strategy does not provide a guide for the image generation process, leaving the details to the imagination of the foundation model. As a result, it is anticipated that a larger portion of the images generated with this strategy should fail the outlier detection test. This is confirmed in Table 5, where around half of the images generated with this strategy (using either of the two kernels) could not pass the test. On the other hand, the Random-Guide strategy is focused on following the data distribution (Theorem 3). Therefore, viewing the images in the dataset as the random iid samples from its underlying distribution, it draws a random image from the dataset and uses it as the guide, irrespective of the description of the image to be generated. This approach, while having a smaller chance of passing the quality test, is expected to have the highest chance of passing the outlier detection test. Our findings in Table 5 are consistent with this expectation. While the Random-Guide strategy outperforms LinUCB and the Similar-Tuple strategy on the outlier detection test, both of these strategies still demonstrate acceptable performance. Regarding mask delineation levels, Accurate delineation exhibits marginally higher OTAR than Moderate and Imprecise levels. This aligns with our expectations, as stricter cropping likely helps constrain the generated images to closer proximity to the original data distribution. However, the performance difference is relatively small, suggesting that the advantages of Moderate and Imprecise delineation in terms of naturalness and flexibility outweigh the slight decrease in distribution adherence for tight boundaries.

Overall, since LinUCB *outperforms the other approaches on the quality evaluation test (which involves human evaluators) and shows an acceptable performance on the outlier detection test*, it is the preferred approach for guide selection.

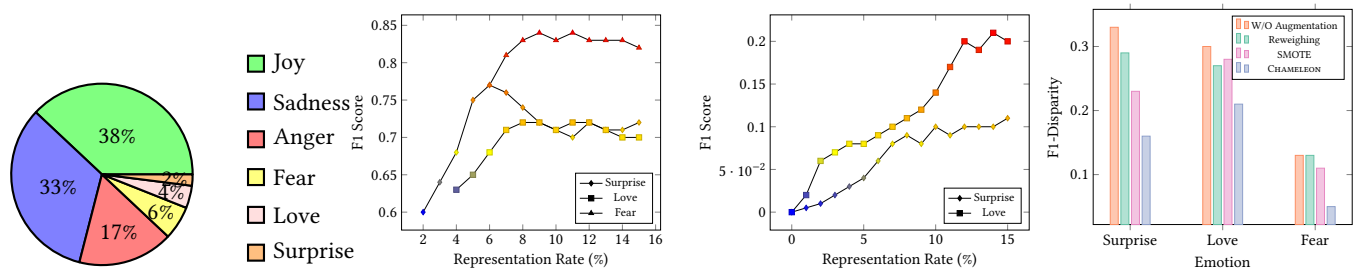
6.3.2 Greedy *Combination Selection Algorithm ②.* In this experiment, we study the impact of employing the Greedy algorithm for selecting the next combination to generate tuples. The entire UTKFace dataset serves as input for our investigation, where we analyze the cost of repairing all ℓ_1 or ℓ_2 MUPs for different thresholds using various combination selection algorithms. As a baseline, we employ the Random selection algorithm, which randomly chooses a combination in each iteration without considering the MUPs' status. We also introduce the Min-Gap algorithm, which, given a list of MUPs, first identifies the MUP which has the smallest gap δ from threshold τ, then chooses a combination that satisfies this MUP and generates δ tuples to satisfy that MUP. Unlike the Greedy algorithm, the Min-Gap Algorithm only focuses on the distance from the threshold, disregarding the number of MUPs hit and MUPs level. We conduct experiments with four distinct values for τ and monitor the total number of images each algorithm requires to add to the dataset for resolving the MUPs. For $\tau = 200$ and $\tau = 350$, all MUPs are at levels ℓ_2 and ℓ_3. In these experiments we the goal is to resolve ℓ_2 MUPs. For $\tau = 1000$ and $\tau = 2000$, where ℓ_1, ℓ_2, and ℓ_3 MUPs are present, our focus is on resolving all ℓ_1 MUPs.

Figure 7 shows the associated cost (total number of images to be generated) for each strategy in different scenarios. We can observe that in all cases Greedy algorithm significantly outperforms both Random and Min-Gap baselines. The Gap even becomes more noticeable when trying to resolve lower-level MUPs. In addition, the Min-Gap strategy performs better than Random for ℓ_2 MUPs but significantly worse when attempting to satisfy the ℓ_1 MUPs. This is due to the larger pool of MUPs in higher thresholds, as the Min-Gap algorithm may choose numerous irrelevant MUPs to satisfy, leaving ℓ_1 MUPs unsatisfied for subsequent iterations.



(a) $\tau = 200$ **(b)** $\tau = 350$ **(c)** $\tau = 1000$ **(d)** $\tau = 2000$

Figure 7: Comparison of cost across various thresholds for different combination-selection algorithms.



(a) Representation rates **(b) Saturation points** **(c) Only synthetic data** **(d) Unfairness comparison.**

Figure 8: Experiment results for the Emotions dataset. Figure 8b shows the augmentation-performance saturation point for each emotion. In Figure 8d, unfairness (disparate performance) is reduced for the uncovered groups.

6.4 Extensions

To demonstrate the adaptability of Chameleon, in this experiment, we extend it for (a) textual data, (b) representation rate [19] for measuring inadequate coverage, and (c) sentiment analysis as the task. We modified our combination selection metric from a threshold-based coverage on the number of elements in each demographic group to a representation rate [19]. For this experiment, we utilized the Emotions dataset. Figure 8a presents emotions in the dataset and their representation rate. We set thresholds of 5%, 10%, and 15%, and augmented the under-represented groups using Chameleon. We adapted the masking and generating phases of Chameleon to handle textual data by selecting sentences with different sentiments and using the GPT-4 foundation model to change the sentiment of these sentences to the desired emotion.

Figure 8b illustrates the F1-score improvement for each group as the representation rate increases. The Chameleon augmentation process has enhanced classifier performance across all under-represented groups. The figure also reveals the saturation point for each emotion, where further augmentation with synthetic data leads to a decrease in model performance. Notably, this performance decline begins when the number of synthetic samples exceeds the number of real samples. Notice that we have augmented each emotion independently from the others to remove the effect of multiple augmentations on the results.

Based on the results in Figure 8b, we augmented each emotion to its peak performance point and compared the resulting disparities with those from Reweighing and SMOTE approaches. As shown in Figure 8d, the F1-disparity is lowest when using Chameleon. The Reweighing approach is almost ineffective, while SMOTE reduces the disparity but is not as effective as Chameleon. It is important to note that augmenting certain emotions is more challenging due to their inherent ambiguity. For example, the emotion *Love* can easily overlap with *Joy*, *Sadness*, or *Surprise* depending on the context, whereas *Fear* has a more distinct boundary from other emotions. This explains why the disparity for *Fear* is nearly zero after augmentation, while *Love* still exhibits some disparity.

In a separate experiment, we tasked GPT-4 with generating tweets for a specific sentiment purely from its own imagination, starting with a dataset that lacked the desired sentiment entirely. This scenario simulates the absence of a minority group and the necessity to generate it entirely based on the foundation model's imagination. Figure 8c illustrates the model's performance when no training data is provided for a particular group, relying solely on the foundation model. As observed, the F1-score significantly differs when comparing scenarios with and without training data. The average F1-score for minority groups with a 2% representation rate is approximately 0.60 when training data is available but drops to about 0.10 with only synthetic data. This disparity arises because the training and test data share the same distribution, whereas Chameleon-generated data relies entirely on the foundation model's imagination. Despite this, using Chameleon, we observe an improvement in the F1-score to up to 0.2 for minority groups, demonstrating a significant enhancement using solely synthetic data.

7 RELATED WORK

While data bias has been a long-standing concern in the statistical community [64], social data presents unique challenges due to its inherent complexity and sensitivity [9, 10, 32, 52, 65]. Issues of diversity and representativeness have been studied across various disciplines, including social science [12, 31, 78], political science [79], and information retrieval [3].

Efforts to trace machine bias back to its sources involve identifying different types [42, 58, 65] and sources [27, 29, 81] of biases in data. Existing work to meet *responsible data* requirements [62] extend throughout various stages of the data analysis pipeline, including data annotation [53, 55], data cleaning and repair [68, 69, 80], data imputation [57], entity resolution [38, 72], and data integration [61, 62, 73].

Data Coverage. The notion of data coverage has been proposed for detecting under-representation issues in a dataset [74]. Related work in this area can be divided into (a) lack of coverage detection and (b) lack of coverage resolution. Coverage detection in tabular data has been studied for both discrete [5] and continuous [6] attributes, whether in single or multiple relations [56], and recently for image dataset [60]. Efforts for addressing lack of coverage without additional data collection include query rewriting [1, 2, 76] and generating lack of representation warning [71].

Related work on improving the coverage of minorities falls under two general categories: (1) real data collection/integration [62] and (2) data augmentation with synthetic data. In the first category, Asudeh et al. [5] and Azzalini et al. [7] specify the minimal additional samples to collect (or remove) to resolve representation bias in data. On the other hand, Nargesian et al. [61] and Chang et al. [20] integrate additional data from a data lake to resolve under-representation issues. Papers in the second category add partially altered duplicates of already existing tuples or generate new synthetic entries from existing data [19, 23, 48, 75]. For example, Sharma et al. [75] duplicates the tuples from the majority group and perturbs them to generate samples from the minority group. More advanced works in this category train generative adversarial networks for data augmentation with synthetic data. Some of such works in the context of time series and healthcare data include [13, 36, 39, 59]. GANs have also been used for image data augmentation [43, 44, 88]. For more details on the detection and resolution approaches for representation bias on data with various modalities, refer to [74]. This paper belongs to the second category, i.e., data augmentation. To the best of our knowledge, none of the existing work uses foundation models for fairness-aware data augmentation to enhance the coverage of minorities in multi-modal datasets.

Foundation Models and Data Management. With recent advancements in LLMs [16, 28, 82] and foundation models, those have been widely used in various research communities for tasks such as Code Generation [24], Synthetic Image Generation [67], and Video Generation [8]. The current state of utilizing these models in the data management community reflects a growing recognition of their potential and challenges. Some of the recent works on utilizing generative AI for data management problems are as follows. LLMs have shown extraordinary performance in answering natural language queries [21, 85, 87]. Particularly, ThalamusDB enables answering complex natural language queries on multi-modal data [50]. LLMs and foundation models have also been utilized for challenging tasks such as dataset search [85], predicting data correlations [84], data-lake profiling [4], and anomaly detection in time-series [25]. Extending data management techniques for LLMs include [33, 34] that develop sampling-based and query rewriting approaches for improving the reliability and fairness of the LLMs. To the best of our knowledge, none of the existing work has utilized foundation models for fairness-aware multi-modal data augmentation.

8 DISCUSSIONS AND LIMITATIONS

8.1 Reliance on the existing models

Our system relies extensively on foundation models for synthetic data augmentation. Consequently, its performance is constrained by the capabilities, limitations, and inherent biases of these models.

Furthermore, our rejection test uses vector representations (embeddings) to represent the underlying dataset's distribution and determine if a generated tuple is an outlier. We assume these embeddings are accurate and that cosine similarity between embeddings reflects their semantic similarities. However, the accuracy and inherent biases of the embeddings are limitations of our approach.

8.2 Distribution preservation

The outlier detection test proposed in § 3.1 operates at the individual level, ensuring each synthetic tuple generated by the foundation model is not an outlier based on the underlying distribution ξ represented by the dataset $\mathcal{D}$. However, an individual-level out-of-distribution test cannot guarantee that a generated set collectively represents the original distribution ξ. Consequently, our test does not guarantee the preservation of the overall distribution.

Theorem 3 provides a bound on the distribution shift on the generated set. It, however, assumes that the foundation model does not significantly perturb the guide tuple. This is hard to measure/verify and may not always be correct.

Therefore, additional steps are needed to ensure the KL divergence between ξ and ξ', the distributions represented by $\mathcal{D}$ and the synthetically generated set, is bounded. Besides the outlier detection test, a stronger set-level distribution test should be applied before adding synthetic tuples to the dataset. If the KL divergence exceeds a predefined threshold, a subset of generated tuples should be removed and regenerated. Determining the minimal set of tuples for removal is an interesting problem, which we leave for future investigation and complexity analysis.

9 CONCLUSION

In this paper, we introduced Chameleon for fairness-aware data augmentation to reduce the under-representation of minority groups. Motivated by the recent advancements in the foundation models, our system efficiently utilizes them for data repair with minimum addition of synthetically generated data while ensuring the augmented data is of high quality and follows the underlying data distribution. Our experiment results demonstrated the effectiveness of our data-repair approach in reducing the unfairness of a downstream task. This motivates future work to extend the scope of fairness-aware data augmentation to other settings.

ACKNOWLEDGMENTS

This work was supported in part by NSF grants 2107290, 2348919, 2106176, and 2312931.

REFERENCES

[1] Chiara Accinelli, Barbara Catania, Giovanna Guerrini, and Simone Minisi. 2021. The impact of rewriting on coverage constraint satisfaction.. In *EDBT Workshops*.

[2] Chiara Accinelli, Simone Minisi, and Barbara Catania. 2020. Coverage-based Rewriting for Data Preparation. In *EDBT Workshops*.

[3] Rakesh Agrawal, Sreenivas Gollapudi, Alan Halverson, and Samuel Ieong. 2009. Diversifying search results. In *WSDM*. ACM, 5–14.

[4] Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. 2023. Language Models Enable Simple Systems for Generating Structured Views of Heterogeneous Data Lakes. *PVLDB* 17, 2 (2023), 97–105.

[5] Abolfazl Asudeh, Zhongjun Jin, and HV Jagadish. 2019. Assessing and remedying coverage for a given dataset. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 554–565.

[6] Abolfazl Asudeh, Nima Shahbazi, Zhongjun Jin, and H. V. Jagadish. 2021. Identifying Insufficient Data Coverage for Ordinal Continuous-Valued Attributes. In *SIGMOD*. ACM.

[7] Fabio Azzalini, Chiara Criscuolo, and Letizia Tanca. 2021. Functional Dependencies to Mitigate Data Bias. In *Proceedings of the 30th Italian Symposium on Advanced Database Systems*.

[8] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Yuanzhen Li, Tomer Michaeli, et al. 2024. Lumiere: A Space-Time Diffusion Model for Video Generation. *arXiv preprint arXiv:2401.12945* (2024).

[9] Solon Barocas, Moritz Hardt, and Arvind Narayanan. 2019. Fairness and machine learning: Limitations and opportunities. fairmlbook.org.

[10] Solon Barocas and Andrew D Selbst. 2016. Big data's disparate impact. *Calif. L. Rev.* 104 (2016), 671.

[11] Alejandro Bellogín, Pablo Castells, and Iván Cantador. 2017. Statistical biases in information retrieval metrics for recommender systems. *Information Retrieval Journal* 20 (2017), 606–634.

[12] Ellen Berrey. 2015. *The enigma of diversity: The language of race and the limits of racial justice*. University of Chicago Press.

[13] Rok Blagus and Lara Lusa. 2013. SMOTE for high-dimensional class-imbalanced data. *BMC bioinformatics* 14 (2013), 1–16.

[14] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258* (2021).

[15] Djallel Bouneffouf, Irina Rish, and Charu Aggarwal. 2020. Survey on applications of multi-armed and contextual bandits. In *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1–8.

[16] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.

[17] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712* (2023).

[18] Kuntai Cai, Xiaokui Xiao, and Graham Cormode. 2023. Privlava: synthesizing relational data with foreign keys under differential privacy. *Proceedings of the ACM on Management of Data* 1, 2 (2023), 1–25.

[19] L Elisa Celis, Vijay Keswani, and Nisheeth Vishnoi. 2020. Data preprocessing to mitigate bias: A maximum entropy based approach. In *ICML*. PMLR, 1349–1359.

[20] Jiwon Chang, Bohan Cui, Fatemeh Nargesian, Abolfazl Asudeh, and HV Jagadish. 2024. Data distribution tailoring revisited: cost-efficient integration of representative data. *The VLDB Journal* (2024), 1–24.

[21] Shuaichen Chang and Eric Fosler-Lussier. 2023. How to Prompt LLMs for Text-to-SQL: A Study in Zero-shot, Single-domain, and Cross-domain Settings. *arXiv preprint arXiv:2305.11853* (2023).

[22] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res.* 16 (2002), 321–357. https://doi.org/10.1613/jair.953

[23] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16 (2002), 321–357.

[24] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374* (2021).

[25] Yuhang Chen, Chaoyun Zhang, Minghua Ma, Yudong Liu, Ruomeng Ding, Bowen Li, Shilin He, Saravan Rajmohan, Qingwei Lin, and Dongmei Zhang. 2023. ImDiffusion: Imputed Diffusion Models for Multivariate Time Series Anomaly Detection. *Proceedings of the VLDB Endowment* 17, 3 (2023), 359–372.

[26] Alessio Corrado. 2019. Animals-10 Dataset. https://www.kaggle.com/datasets/alessiocorrado99/animals10 Accessed: 2024-05-16.

[27] Kate Crawford. 2013. The hidden biases in big data. *Harvard business review* 1, 4 (2013).

[28] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[29] Nicholas Diakopoulos. 2015. Algorithmic accountability: Journalistic investigation of computational power structures. *Digital journalism* 3, 3 (2015), 398–415.

[30] Wilfrid J Dixon and Frank J Massey Jr. 1951. Introduction to statistical analysis. (1951).

[31] Frank Dobbin and Alexandra Kalev. 2016. Why diversity programs fail and what works better. *Harvard Business Review* 94, 7-8 (2016), 52–60.

[32] Marina Drosou, HV Jagadish, Evaggelia Pitoura, and Julia Stoyanovich. 2017. Diversity in big data: A review. *Big data* 5, 2 (2017), 73–84.

[33] Sana Ebrahimi, Kaiwen Chen, Abolfazl Asudeh, Gautam Das, and Nick Koudas. 2024. AXOLOTL: Fairness through Assisted Self-Debiasing of Large Language Model Outputs. *arXiv preprint arXiv:2403.00198* (2024).

[34] Sana Ebrahimi, Nima Shahbazi, and Abolfazl Asudeh. 2024. REQUAL-LM: Reliability and Equity through Aggregation in Large Language Models. *arXiv preprint arXiv:2404.11782* (2024).

[35] Mahdi Erfanian, HV Jagadish, and Abolfazl Asudeh. 2024. Chameleon: Foundation Models for Fairness-aware Multi-modal Data Augmentation to Enhance Coverage of Minorities. *arXiv preprint arXiv:2402.01071* (2024).

[36] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. 2017. Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633* (2017).

[37] Ju Fan, Tongyu Liu, Guoliang Li, Junyou Chen, Yuwei Shen, and Xiaoyong Du. 2020. Relational data synthesis using generative adversarial networks: A design space exploration. *arXiv preprint arXiv:2008.12763* (2020).

[38] Nikolaos Fanourakis, Christos Kontousias, Vasilis Efthymiou, Vassilis Christophides, and Dimitris Plexousakis. 2023. FairER demo: Fairness-Aware and Explainable Entity Resolution. (2023).

[39] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. 2018. Data augmentation using synthetic data for time series classification with deep residual networks. *arXiv preprint arXiv:1808.02455* (2018).

[40] Yunhe Feng and Chirag Shah. 2022. Has CEO Gender Bias Really Been Fixed? Adversarial Attacking and Improving Gender Fairness in Image Search. (2022).

[41] Bernard D Flury. 1990. Acceptance–rejection sampling made easy. *Siam Review* 32, 3 (1990), 474–476.

[42] Batya Friedman and Helen Nissenbaum. 1996. Bias in computer systems. *TOIS* 14, 3 (1996), 330–347.

[43] Markos Georgopoulos, James Oldfield, Mihalis A Nicolaou, Yannis Panagakis, and Maja Pantic. 2021. Mitigating demographic bias in facial datasets with style-based multi-attribute transfer. *International Journal of Computer Vision* 129, 7 (2021), 2288–2307.

[44] Karan Goel, Albert Gu, Yixuan Li, and Christopher Ré. 2020. Model patching: Closing the subgroup performance gap with data augmentation. *arXiv preprint arXiv:2008.06775* (2020).

[45] John Hammersley. 2013. *Monte carlo methods*. Springer Science & Business Media.

[46] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. 2005. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing: International Conference on Intelligent Computing, ICIC 2005, Hefei, China, August 23-26, 2005, Proceedings, Part I 1*. Springer, 878–887.

[47] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. 2019. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*. 1314–1324.

[48] Vasileios Iosifidis and Eirini Ntoutsi. 2018. Dealing with bias via data augmentation in supervised learning scenarios. *Jo Bates Paul D. Clough Robert Jäschke* 24 (2018).

[49] Hosagrahar V Jagadish, Johannes Gehrke, Alexandros Labrinidis, Yannis Papakonstantinou, Jignesh M Patel, Raghu Ramakrishnan, and Cyrus Shahabi. 2014. Big data and its technical challenges. *Commun. ACM* 57, 7 (2014), 86–94.

[50] Saehan Jo and Immanuel Trummer. 2023. Demonstration of ThalamusDB: Answering Complex SQL Queries with Natural Language Predicates on Multi-Modal Data. In *Companion of the 2023 International Conference on Management of Data*. 179–182.

[51] Faisal Kamiran and Toon Calders. 2012. Data preprocessing techniques for classification without discrimination. *Knowledge and information systems* 33, 1 (2012), 1–33.

[52] Jon Kleinberg. 2019. Fairness, Rankings, and Behavioral Biases. FAT*.

[53] Simone Lazier, Saravanan Thirumuruganathan, and Hadis Anahideh. 2023. Fairness and Bias in Truth Discovery Algorithms: An Experimental Analysis. *arXiv preprint arXiv:2304.12573* (2023).

[54] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. 661–670.

[55] Yanying Li, Haipei Sun, and Wendy Hui Wang. 2020. Towards fair truth discovery from biased crowdsourced answers. In *SIGKDD*. 599–607.

[56] Yin Lin, Yifan Guan, Abolfazl Asudeh, and HV Jagadish. 2020. Identifying insufficient data coverage in databases with multiple relations. *Proceedings of the VLDB Endowment* 13, 12 (2020), 2229–2242.

[57] Fernando Martínez-Plumed, Cèsar Ferri, David Nieves, and José Hernández-Orallo. 2019. Fairness and missing values. *arXiv preprint arXiv:1905.12728* (2019).

[58] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. 2021. A survey on bias and fairness in machine learning. *ACM Computing Surveys (CSUR)* 54, 6 (2021), 1–35.

[59] Nicolo Micheletti, Raffaele Marchesi, Nicholas I-Hsien Kuo, Sebastiano Barbieri, Giuseppe Jurman, and Venet Osmani. 2023. Generative AI Mitigates Representation Bias Using Synthetic Health Data. *medRxiv* (2023), 2023–09.

[60] Melika Mousavi, Nima Shahbazi, and Abolfazl Asudeh. 2024. Data Coverage for Detecting Representation Bias in Image Datasets: A Crowdsourcing Approach. In *EDBT*. 47–60.

[61] Fatemeh Nargesian, Abolfazl Asudeh, and HV Jagadish. 2021. Tailoring data source distributions for fairness-aware data integration. *Proceedings of the VLDB Endowment* 14, 11 (2021), 2519–2532.

[62] Fatemeh Nargesian, Abolfazl Asudeh, and H. V. Jagadish. 2022. Responsible Data Integration: Next-generation Challenges. *SIGMOD* (2022).

[63] Nelgiriyewithana. 2023. Emotions Dataset. https://www.kaggle.com/datasets/nelgiriyewithana/emotions Accessed: 2024-05-16.

[64] Jerzy Neyman and Egon Sharpe Pearson. 1936. Contributions to the theory of testing statistical hypotheses. *Statistical Research Memoirs* (1936).

[65] Alexandra Olteanu, Carlos Castillo, Fernando Diaz, and Emre Kiciman. 2019. Social data: Biases, methodological pitfalls, and ethical boundaries. *Frontiers in Big Data* 2 (2019), 13.

[66] P Jonathon Phillips, Harry Wechsler, Jeffery Huang, and Patrick J Rauss. 1998. The FERET database and evaluation procedure for face-recognition algorithms. *Image and vision computing* 16, 5 (1998), 295–306.

[67] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. 2022. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125* 1, 2 (2022), 3.

[68] Babak Salimi, Bill Howe, and Dan Suciu. 2020. Database repair meets algorithmic fairness. *ACM SIGMOD Record* 49, 1 (2020), 34–41.

[69] Babak Salimi, Luke Rodriguez, Bill Howe, and Dan Suciu. 2019. Interventional Fairness: Causal Database Repair for Algorithmic Fairness. In *SIGMOD*. ACM, 793–810.

[70] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. 1999. Support vector method for novelty detection. *Advances in neural information processing systems* 12 (1999).

[71] Nima Shahbazi and Abolfazl Asudeh. 2024. Reliability evaluation of individual predictions: a data-centric approach. *The VLDB Journal* (2024), 1–28.

[72] Nima Shahbazi, Nikola Danevski, Fatemeh Nargesian, Abolfazl Asudeh, and Divesh Srivastava. 2023. Through the Fairness Lens: Experimental Analysis and Evaluation of Entity Matching. *Proceedings of the VLDB Endowment* 16, 11 (2023), 3279–3292.

[73] Nima Shahbazi, Mahdi Erfanian, and Abolfazl Asudeh. 2024. Coverage-based Data-centric Approaches for Responsible and Trustworthy AI. *IEEE Data Eng. Bull.* 47, 1 (2024), 3–17.

[74] Nima Shahbazi, Yin Lin, Abolfazl Asudeh, and HV Jagadish. 2023. Representation Bias in Data: A Survey on Identification and Resolution Techniques. *Comput. Surveys* (2023).

[75] Shubham Sharma, Yunfeng Zhang, Jesús M Ríos Aliaga, Djallel Bouneffouf, Vinod Muthusamy, and Kush R Varshney. 2020. Data augmentation for discrimination prevention and bias disambiguation. In *AIES*. 358–364.

[76] Suraj Shetiya, Ian P. Swift, Abolfazl Asudeh, and Gautam Das. 2022. Fairness-Aware Range Queries for Selecting Unbiased Data. In *ICDE*. IEEE.

[77] Mallory Simon. 2009. HP looking into claim webcams can't see black people. CNN.

[78] Edward H Simpson. 1949. Measurement of diversity. *Nature* 163, 4148 (1949).

[79] James Surowiecki. 2005. *The wisdom of crowds*. Anchor.

[80] Ki Hyun Tae, Yuji Roh, Young Hun Oh, Hyunsu Kim, and Steven Euijong Whang. 2019. Data cleaning for accurate, fair, and robust models: A big data-AI integration approach. In *DEEM workshop*. 1–4.

[81] Antonio Torralba and Alexei A Efros. 2011. Unbiased look at dataset bias. In *CVPR 2011*. IEEE, 1521–1528.

[82] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[83] Tess Townsend. 2017. Most engineers are white and so are the faces they use to train software. Recode.

[84] Immanuel Trummer. 2023. Can Large Language Models Predict Data Correlations from Column Names? *Proceedings of the VLDB Endowment* 16, 13 (2023), 4310–4323.

[85] Immanuel Trummer. 2023. Demonstrating GPT-DB: Generating Query-Specific and Customizable Code for SQL Processing with GPT-4. *Proceedings of the VLDB Endowment* 16, 12 (2023), 4098–4101.

[86] Aleksandra Urman, Mykola Makhortykh, and Roberto Ulloa. 2022. Auditing the representation of migrants in image web search results. *Humanities and Social Sciences Communications* 9, 1 (2022), 1–16.

[87] Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large language models are versatile decomposers: Decompose evidence and questions for table-based reasoning. *arXiv preprint arXiv:2301.13808* (2023).

[88] Seyma Yucer, Samet Akçay, Noura Al-Moubayed, and Toby P Breckon. 2020. Exploring racial bias within face recognition via per-subject adversarially-enabled data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 18–19.

[89] Zhifei Zhang, Yang Song, and Hairong Qi. 2017. Age Progression/Regression by Conditional Adversarial Autoencoder. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[90] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. 2023. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *arXiv preprint arXiv:2302.09419* (2023).