A SYNTHETIC GENERATION

In this section, we introduce a few additional findings on coverage redundant Simpson's paradoxes from a data generative perspective. We first discuss the generation of separate instances of Simpson's paradox in Section A.1. We then introduce the process of generating coverage redundant Simpson's paradoxes in Section A.2. Finally, we summarize the overall data synthesization procedure in Section A.3.

A.1 Generating Simpson's Paradoxes

Recall from Definition 2.2 that an association configuration (AC) $p = (s_1, s_2, X, Y)$ is a Simpson's paradox if:

$$\begin{aligned}&(1)\ P(Y|s_1) \leq P(Y|s_2);\ \text{and}\\ &(2)\ P(Y|s_1\langle X \shortrightarrow v\rangle) \geq P(Y|s_2\langle X \shortrightarrow v\rangle),\ \forall v \in \mathrm{Dom}(X).\end{aligned}$$

In particular, the frequency statistics $P(Y|s_j)$ for $j = 1, 2$ in condition (1), is obtained by weighted averaging their sub-population frequency statistics in condition (2), where the weights are determined by the relative coverage sizes of each sub-population. Specifically, we have that:

$$\begin{aligned}P(Y|s_j) &= \sum_{v \in \mathrm{Dom}(X)} \frac{|\,\mathrm{cov}(s_j\langle X \shortrightarrow v\rangle)|}{|\,\mathrm{cov}(s_j)|} \cdot P(Y|s_j\langle X \shortrightarrow v\rangle)\\ &= \boldsymbol{Q}(s_j|X) \cdot \boldsymbol{P}(s_j|Y,X)^\top,\end{aligned}$$

where

$$\boldsymbol{Q}(s_j|X) = \left[\frac{|\,\mathrm{cov}(s_j\langle X \shortrightarrow v_1\rangle)|}{|\,\mathrm{cov}(s_j)|}, \frac{|\,\mathrm{cov}(s_j\langle X \shortrightarrow v_2\rangle)|}{|\,\mathrm{cov}(s_j)|}, \ldots, \frac{|\,\mathrm{cov}(s_j\langle X \shortrightarrow v_{|\,\mathrm{Dom}(X)|}\rangle)|}{|\,\mathrm{cov}(s_j)|}\right]$$

is the sample distribution of s_j partitioned under X, and

$$\boldsymbol{P}(s_j|Y,X) = \left[P(Y|s_j\langle X \shortrightarrow v_1\rangle), P(Y|s_j\langle X \shortrightarrow v_2\rangle), \ldots, P(Y|s_j\langle X \shortrightarrow v_{|\,\mathrm{Dom}(X)|}\rangle)\right]$$

is the frequency statistics of s_j's sub-populations partitioned by X. With this, we can rephrase Definition 2.2 by substituting the terms, that (s_1, s_2, X, Y) is a Simpson's paradox if:

$$\boldsymbol{Q}(s_1|X) \cdot \boldsymbol{P}(s_1|Y,X)^\top < \boldsymbol{Q}(s_2|X) \cdot \boldsymbol{P}(s_2|Y,X)^\top;\ \text{and} \quad (1)$$

$$\boldsymbol{P}(s_1|Y,X)[j] > \boldsymbol{P}(s_2|Y,X)[j],\ 1 \leq j \leq |\,\mathrm{Dom}(X)|. \quad (2)$$

Therefore, the essence of generating an instance of Simpson's paradox is to *(a)* find the set of sub-population frequency statistics $\boldsymbol{P}(s_1|Y,X)$ and $\boldsymbol{P}(s_2|Y,X)$ that satisfy inequality (2); and *(b)* solve for the sample distributions $\boldsymbol{Q}(s_1|X)$ and $\boldsymbol{Q}(s_2|X)$ that satisfy inequality (1). We discuss each in the following paragraphs.

Sub-population Frequency Statistics. The first step of generating an instance of Simpson's paradox is to ensure that each sub-population in s_2, partitioned by X, has an frequency statistics value smaller than its sibling sub-population in s_1 (inequality (2)). A simple pattern that achieves this is:

$$\begin{aligned}\boldsymbol{P}(s_2|Y,X)[1] < \boldsymbol{P}(s_1|Y,X)[1] <\\ \boldsymbol{P}(s_2|Y,X)[2] < \boldsymbol{P}(s_1|Y,X)[2] <\\ \cdots\\ \boldsymbol{P}(s_2|Y,X)[|\,\mathrm{Dom}(X)|] <\\ \boldsymbol{P}(s_1|Y,X)[|\,\mathrm{Dom}(X)|].\end{aligned} \quad (3)$$

This pattern ensures that for any value v_j $(1 \leq j \leq |\,\mathrm{Dom}(X)|)$, we have $\boldsymbol{P}(s_1|Y,X)[j] > \boldsymbol{P}(s_2|Y,X)[j]$, satisfying inequality (2). With this pattern, we can now focus on finding sample distributions that satisfy inequality (1).

Sample Distributions. Given the sub-population frequency statistics pattern established above, we now need to solve for sample distributions $\boldsymbol{Q}(s_1|X)$ and $\boldsymbol{Q}(s_2|X)$ that satisfy inequality (1). To achieve this, we formulate the problem as a quadratic program:

$$\begin{aligned}\text{minimize}\quad & \sum_{j=1}^{2} \left\| \boldsymbol{Q}(s_j|X) - \frac{1}{|\,\mathrm{Dom}(X)|}\mathbf{1} \right\|_2^2\\ \text{subject to}\quad & \text{(i)}\ \sum_{k=1}^{|\,\mathrm{Dom}(X)|} \boldsymbol{Q}(s_j|X)[k] = 1, \quad j = 1, 2;\\ & \text{(ii)}\ \boldsymbol{Q}(s_j|X)[k] > 0, \quad j = 1, 2\\ & \quad \text{and } 1 \leq k \leq |\,\mathrm{Dom}(X)|;\\ & \text{(iii)}\ \boldsymbol{Q}(s_2|X) \cdot \boldsymbol{P}(s_2|Y,X)^\top\\ & \quad > \boldsymbol{Q}(s_1|X) \cdot \boldsymbol{P}(s_1|Y,X)^\top.\end{aligned} \quad (4)$$

We incorporate inequality (1) as a linear constraint specified in condition (iii). Moreover, the objective function of the QP aims to minimize the squared distance between each sample distribution and the uniform distribution $\frac{1}{|\,\mathrm{Dom}(X)|}\mathbf{1}$ to promote uniformity. Our synthetic generator also supports optimizing towards other distribution patterns such as normal or Zipfian distributions to better mimic the statistics of real-world data.

Synthesis of Simpson's Paradox. To summarize, generating an instance of Simpson's paradox requires establishing sub-population frequency statistics satisfying inequality (2) and solving for sample distributions that produce the reversal effect outlined in inequality (1). Algorithm 7 formalizes this generation procedure, taking as input an AC (s_1, s_2, X, Y) and crucially, a size parameter U that controls the number of records covered by the generated Simpson's paradox. The algorithm proceeds in three main steps: first, it generates the sub-population frequency statistics following the pattern in Equation (3); second, it solves the quadratic program in (4) to obtain optimal sample distributions; and third, it populates the output table with $2 \cdot U$ records for the Simpson's paradox (s_1, s_2, X, Y), distributing records across sub-populations according to the sample distributions and assigning label values according to probabilities given by the sub-population frequency statistics.

A.2 Realizing Coverage Redundancies

Having established a method for generating individual instances of Simpson's paradox, we now discuss creating (coverage) redundant Simpson's paradoxes. Building on the discussions in Section 3.1 and

Algorithm 7 Generate non-redundant Simpson's paradox.

Input: An AC (s_1, s_2, X, Y), paradox size U
Output: Data records T for the Simpson's paradox (s_1, s_2, X, Y)

Obtain the sub-population frequency statistics $\boldsymbol{P}(s_1|Y, X)$, $\boldsymbol{P}(s_2|Y, X)$ following Equation (3);
Obtain the sample distributions $\boldsymbol{Q}(s_1|X)$, $\boldsymbol{Q}(s_2|X)$ by solving the quadratic program (QP) in (4);
// Populate data records following the obtained sample distributions and sub-population aggregate statistics
for each $1 \le k \le |\,\mathrm{Dom}(X)|$ and each $j \in [1, 2]$ **do**
    // Find the number of records for each sub-population
    Let $U_{j,k} \leftarrow U \cdot \boldsymbol{Q}(s_j|X)[k]$;
    // Assign labels according to sub-population aggr. stats.
    Add $U_{j,k}$ copies of $s_j\langle X \to v_k\rangle$ as records to T and assign $U_{j,k} \cdot \boldsymbol{P}(s_j|Y, X)[k]$ of them with $(Y = 1)$;
return T.

Definition 3.8, (coverage) redundancies fundamentally arise when distinct populations have identical coverage within the data. To systematically realize (coverage) redundancies, we must understand the conditions under which populations would share the same coverage, as this property serves as the foundation for realizing both sibling child and separator equivalences. To this end, we remark on the following proposition.

Proposition A.1 (Impossibility of coverage identicality). *Let T be a base table with n categorical attributes $\{X_1, \ldots, X_n\}$. If the set of unique records in T corresponds exactly to all possible combinations of attribute values (*i.e., *the complete Cartesian product $\prod_{i=1}^{n} \mathrm{Dom}(X_i)$), then no two distinct populations of T share the same coverage.*

Proof. Assume, by contradiction, that there exist two distinct populations s and s' with $\mathrm{cov}(s) = \mathrm{cov}(s')$. Since $s \neq s'$, let X_{k_0} be an attribute for which $s[k_0] \neq s'[k_0]$. Without loss of generality, assume $s[k_0] = v$ for a fixed value $v \in \mathrm{Dom}(X_{k_0})$ and $s'[k_0] = *$.

Let us consider two records, r_1 and r_2, which are identical on all attributes except X_{k_0}. For r_1, let $r_1.X_{k_0} = v$, and for r_2, let $r_2.X_{k_0} = v'$ where $v' \neq v$ and $v' \in \mathrm{Dom}(X_{k_0})$. For all other attributes X_j where $j \neq k_0$, if $s[j] \neq *$, then $r_1.X_j = r_2.X_j = s[j]$. Observe that both records r_1 and r_2 must exist in T because T contains the complete Cartesian product of all attribute domains.

Now, r_1 is covered by both s and s'. However, record r_2 is covered by s' (because $s'[k_0] = *$) but not by s (because $s[k_0] = v$ but $r_2.X_{k_0} = v', v' \neq v$).

This means $r_2 \in \mathrm{cov}(s')$ but $r_2 \notin \mathrm{cov}(s)$, which contradicts our assumption that $\mathrm{cov}(s) = \mathrm{cov}(s')$. Therefore, no two distinct populations can share the same coverage when T contains the complete Cartesian product of all attribute domains. □

The contrapositive of Proposition A.1 implies that populations sharing identical coverage can only exist when the dataset contains a proper subset of the complete Cartesian product of attribute domains. Therefore, to facilitate the generation of coverage redundant Simpson's paradoxes, we impose a size threshold t that is significantly smaller than $\prod_{i=1}^{n} |\,\mathrm{Dom}(X_i)|$ to constrain the number of unique records in the generated dataset.

Having established the condition for realizing populations with identical coverage, we now proceed to develop methods for realizing each of the three types of coverage equivalences: sibling child equivalence, separator equivalence, and statistic equivalence.

Sibling Child Equivalence. Suppose we have a set of data records T that produces a Simpson's paradox $p_1 = (s_1, s_2, X, Y)$, where $s_1 = s\langle X_0 \to u_1\rangle$ and $s_2 = s\langle X_0 \to u_2\rangle$ are siblings from a common parent s. To realize sibling equivalence, our goal is update the records in T such that they also produce another Simpson's paradox $p_2 = (s_1', s_2', X, Y)$, where $s_1' = s'\langle X_0' \to v_1\rangle$ and $s_2' = s'\langle X_0' \to v_2\rangle$ are siblings from a common parent s', and that p_2 is sibling child equivalent to p_1. According to Definition 3.8, sibling child equivalence requires $\mathrm{cov}(s_1) = cov(s_1')$ and $\mathrm{cov}(s_2) = \mathrm{cov}(s_2')$. Based on the relationship between (s, s_1, s_2) and (s', s_1', s_2'), we have three scenarios:

(1) **Scenario 1**: $s \neq s'$, $X_0 = X_0'$, and $\{u_1, u_2\} = \{v_1, v_2\}$. In this case, sibling child equivalence is achieved by ensuring $\mathrm{cov}(s) = \mathrm{cov}(s')$. To this, for each categorical attribute X_k where $s[k] \neq *$ or $s'[k] \neq *$, we update every record r in $\mathrm{cov}(s)$ (within T), such that $r.X_k = s[k]$ or $r.X_k = s'[k]$. If $s[k] = s'[k] = *$, then no update is needed for $r.X_k$.
(2) **Scenario 2**: $s = s'$, $X_0 \neq X_0'$, and $\{u_1, u_2\} \neq \{v_1, v_2\}$. In this case, to achive sibling child equivalence, we establish a one-to-one mapping $f : \{u_1, u_2\} \mapsto \{v_1, v_2\}$ such that $f(u_1) = v_1$ and $f(u_2) = v_2$. For each record r in T, we set $r.X_0' = f(r.X_0)$ when $r.X_0 \in \{u_1, u_2\}$. This ensures $\mathrm{cov}(s\langle X_0 \to u_k\rangle) = \mathrm{cov}(s\langle X_0' \to f(u_k)\rangle)$ for $k = 1, 2$.
(3) **Scenario 3**: $s \neq s'$, $X_0 \neq X_0'$, and $\{u_1, u_2\} \neq \{v_1, v_2\}$. This combines the previous senarios. To achieve sibling child equivalence, we first ensure $\mathrm{cov}(s) = \mathrm{cov}(s')$ as in Scenario 1, then establish the one-to-one mapping as in Scenario 2.

Example A.2. Consider the data records in a slightly perturbed version of Table 2 where attribute values in D are randomized. Supposed the perturbed Table 2 is populated as a result of generating the Simpson's paradox $p_1 = ((*, b_1, *, *), (*, b_2, *, *), A, Y_1)$. To create a sibling equivalent Simpson's paradox $p_2 = ((*, *, *, d_1), (*, *, *, d_2), A, Y_1)$, we apply both Scenarios 1 and 2.

For Scenario 1, the parent population is $(*, *, *, *)$ for both p_1 and p_2, which are identical, so no adjustment to records in the perturbed table is needed.

For Scenario 2, we define a one-to-one mapping $f : \{b_1, b_2\} \to \{d_1, d_2\}$ where $f(b_1) = d_1$ and $f(b_2) = d_2$. We then update each record in the perturbed table with $B = b_1$ gets $D = d_1$ and each record with $B = b_2$ gets $D = d_2$. This establishes that $cov((*, b_1, *, *)) = cov((*, *, *, d_1))$ and $cov((*, b_2, *, *)) = cov((*, *, *, d_2))$.

In this way, we make p_1 and p_2 to be sibling equivalent, as verified in Example 3.2. □

Algorithm 8 formalizes this process of generating sibling-child-equivalent Simpson's paradoxes.

Separator Equivalence. Recall from Proposition 3.3 and Definition 3.8, Simpson's paradoxes $p_1 = (s_1, s_2, X_1, Y)$ and $p_2 = (s_1, s_2, X_1', Y)$ are separator equivalent if there exists a one-to-one mapping f between $\mathrm{Dom}(X_1)$ and $\mathrm{Dom}(X_1')$ such that for every

Algorithm 8 Realizing sibling child equivalence.

Input: Data records T producing the Simpson's paradox $p_1 = (s_1, s_2, X, Y)$ where $s_1 = s\langle X_0 \to u_1\rangle$ and $s_2 = s\langle X_0 \to u_2\rangle$, sibling populations (s_1, s_2) where $s_1' = s\langle X_0' \to v_1\rangle$ and $s_2' = s\langle X_0' \to v_2\rangle$
Output: Updated data records T producing a sibling-child-equivalent Simpson's paradox $p_2 = (s_1', s_2', X, Y)$

```
// Scenario 1: Ensure cov(s) = cov(s')
for each record r ∈ T s.t. r ∈ cov(s) do
    for each attribute X_k s.t. s[k] ≠ * or s'[k] ≠ * do
        Set r.X_k ← s[k] if s[k] ≠ *, else r.X_k ← s'[k];
// Scenario 2: Establish the one-to-one mapping
Establish the mapping f where f(u_j) = v_j for j = 1, 2;
for each record r ∈ T do
    Set r.X_0' ← f(r.X_0) if r.X_0 ∈ {u_1, u_2}.
```

$v \in \mathrm{Dom}(X_1)$ and $j = 1, 2$,

$$\mathrm{cov}(s_j\langle X_1 \to v\rangle) = \mathrm{cov}(s_j\langle X_1' \to f(v)\rangle).$$

To achieve this, for every record r in T, we set $r.X_1' = f(r.X_1)$, which we formalize the process in Algorithm 9.

Example A.3. Consider a perturbed version of Table 2 where attribute values in C are initially randomized. Suppose the perturbed Table 2 is populated as a result of generating the Simpson's paradox $p_1 = ((*, b_1, *, *), (*, b_2, *, *), A, Y_1)$. To create a separator equivalent Simpson's paradox $p_2 = ((*, b_1, *, *), (*, b_2, *, *), C, Y_1)$, we establish a one-to-one mapping f between $\mathrm{Dom}(A) = a_1, a_2$ and $\mathrm{Dom}(C) = c_1, c_2$ where $f(a_1) = c_1$ and $f(a_2) = c_2$. We then update each record in the perturbed table so that whenever $A = a_1$, we set $C = c_1$, and whenever $A = a_2$, we set $C = c_2$. This ensures that $\mathrm{cov}((*, b_1, *, *)\langle A \to a_k\rangle) = \mathrm{cov}((*, b_1, *, *)\langle C \to c_k\rangle)$ and $\mathrm{cov}((, b_2, *,)\langle A \to a_k\rangle) = \mathrm{cov}((, b_2, *, *)\langle C \to c_k\rangle)$ for $k \in \{1, 2\}$. As verified in Example 3.4, p_1 and p_2 are separator equivalent. □

Algorithm 9 Realizing separator equivalence.

Input: Set of data records T producing the Simpson's paradox (s_1, s_2, X_1, Y), a separator attribute X_1'
Output: Updated set of data records T producing a separator equivalent Simpson's paradox $p_2 = (s_1, s_2, X_1', Y)$

```
Let f : Dom(X_1) ↦ Dom(X_1') be the one-to-one map;
for each record r ∈ T do
    Set r.X_1' ← f(r.X_1).
```

Statistic Equivalence. Recall from Proposition 3.5 and Definition 3.8, Simpson's paradoxes $p_1 = (s_1, s_2, X, Y_2)$ and $p_2 = (s_1, s_2, X, Y_2')$ are statistic equivalent if for each s_j $(j = 1, 2)$ $P(Y_2|s_j) = P(Y_2'|s_j)$, and for every value $v \in \mathrm{Dom}(X)$, $P(Y_2|s_j\langle X \to v\rangle) = P(Y_2'|s_j\langle X \to v\rangle)$. To achieve this, we simply ensure that each record has identical values for both label attributes Y_2 and Y_2'. We formalize this process in Algorithm 10.

Example A.4. Consider a perturbed version of Table 2 where attribute values in Y_2 are initially randomized. Suppose the perturbed Table 2 is populated as a result of generating the Simpson's paradox $p_1 = ((*, b_1, *, *), (*, b_2, *, *), A, Y_1)$. To create a statistic equivalent Simpson's paradox $p_2 = ((*, b_1, *, *), (*, b_2, *, *), A, Y_2)$, we update each record in the perturbed table so that $Y_2 = Y_1$ for all records. This ensures that $P(Y_1|s_j) = P(Y_2|s_j)$ and $P(Y_1|s_j\langle A \to a_k\rangle) = P(Y_2|s_j\langle A \to a_k\rangle)$ for $j \in \{1, 2\}$ and $k \in \{1, 2\}$. As verified in Example 3.6, p_1 and p_2 are statistic equivalent. □

Algorithm 10 Realizing statistic equivalence.

Input: Data records T producing the Simpson's paradox $p_1 = (s_1, s_2, X, Y_2)$, a label attribute Y_2'
Output: Updated data records T producing a statistic equivalent Simpson's paradox $p_2 = (s_1, s_2, X, Y_2')$

```
for each record r ∈ T do
    Set r.Y_2' ← r.Y_2.
```

A.3 Data Generation Workflow

Algorithm 11 Generate redundant Simpson's paradoxes.

Input: Categorical attributes $\{X_i\}_{i=1}^n$, label attributes $\{Y_j\}_{j=1}^m$, size threshold $t \ll \prod_{i=1}^n |\mathrm{Dom}(X_i)|$
Output: Data table T (initially empty)

```
Let R ← ∅ to collect the set of unique data records;
Let P ← ∅ to collect the set of generated Simpson's paradoxes;
while |R| < t do
    Let p_1 = (s_1, s_2, X_1, Y_3) be an AC not in P;
    // Step 1: Generate distinct Simpson's paradox
    Populate T' for the Simpson's paradox p_1 using Alg. 7;
    // Step 2: Introduce coverage redundancies
    // Sibling child equivalence
    Apply Alg. 8 to T' to create a sibling-child-equivalent Simpson's paradox p_2 = (s_1', s_2', X_1, Y_2);
    // Separator equivalence
    Apply Alg. 9 to T' to create a separator equivalent Simpson's paradox p_3 = (s_1, s_2, X_1', Y_3);
    // Statistic equivalence
    Apply Alg. 10 to T' to create a statistic equivalent Simpson's paradox p_4 = (s_1, s_2, X_1, Y_2');
    Add p_1, p_2, p_3, and p_4 to P;
    Add T' to T and unique records of T' to R;
return T.
```

Building upon the techniques established in Sections A.1 and A.2, we formulate a systematic approach for synthetic data generation that integrates both individual Simpson's paradox generation and coverage redundancy realization. The process employs a two-phase strategy: first generating distinct instances of Simpson's paradoxes (Section A.1), then systematically introducing coverage redundancies through sibling child, separator, and statistics equivalences (Section A.2). These phases are iterated until reaching a specified threshold $t \ll \prod_{i=1}^n |\mathrm{Dom}(X_i)|$ of unique records populated, which per Proposition A.1 ensures the dataset contains populations with identical coverage necessary for redundancy.

Algorithm 11 formalizes this process, taking categorical attributes $\{X_i\}_{i=1}^n$, label attributes $\{Y_j\}_{j=1}^m$, and the size threshold t as input, and producing a synthetic data table containing groups of (coverage) redundant Simpson's paradoxes.