FULL-INFORMATION ESTIMATION FOR HIERARCHICAL DATA

RYAN CUMINGS-MENON

Abstract. The U.S. Census Bureau's 2020 Disclosure Avoidance System (DAS) bases its output on noisy measurements, which are population tabulations added to realizations of mean-zero random variables. These noisy measurements are observed in a set of hierarchical geographic units, *e.g.*, the U.S. as a whole, states, counties, census tracts, and census blocks. The noisy measurements from the 2020 Redistricting Data File and Demographic and Housing Characteristics File statistical data products are now public. The purpose of this paper is to describe a method to leverage the hierarchical structure within these noisy measurements to compute confidence intervals for arbitrary tabulations and in arbitrary geographic entities composed of census blocks. This method is based on computing a weighted least squares estimator (WLS) and its variance matrix. Due to the high dimension of this estimator, this operation is not feasible using the standard approach, since this would require evaluating products with the inverse of a dense matrix with several billion (or even several trillion) rows and columns. In contrast, the approach we describe in this paper computes the required estimate and its variance with a time complexity and memory requirement that scales linearly in the number of census blocks.

1. Introduction

As described by Abowd et al. (2022); Cumings-Menon et al. (2023) in more detail, the U.S. Census Bureau's 2020 Disclosure Avoidance System (DAS) uses formally private methods (Bun and Steinke, 2016; Dwork et al., 2006) to protect the confidentiality of 2020 Census respondents. One important advantage of formally private methods is that they allow for a much greater level of transparency than the methods the Census Bureau used for disclosure limitation in previous decennial censuses. For example, the Census Bureau has released noisy measurements that were used to create the tabulations found in the published 2020 Census statistical data products (U.S. Census Bureau, 2023b), which are called the noisy measurement files (NMFs). These scalar noisy measurements are defined as population tabulations for specific geographic units (*e.g.*, the count of respondents that identify as Black and Asian in Utah, the count of respondents that identify as Hispanic or Latino in the U.S. as a whole, *etc.*) added to realizations of mean-zero random variables. The geographic units used to define these noisy measurements are also defined hierarchically in a rooted tree, *e.g.*, the U.S. as a whole, states, counties, census tracts, and census blocks. The purpose of this paper is to describe a way to use these hierarchical NMFs to compute the best linear unbiased estimator (BLUE) and confidence intervals (CIs) for arbitrary cross tabulations and for arbitrary geographic regions.

After introducing notation in the next subsection, the remainder of this section describes our primary motivating use case, *i.e.*, computing point estimates and CIs using the publicly available NMFs, and the previous literature on methods related to the ones we describe here. To make this paper accessible for other use cases, when possible, we avoid terminology that is specific to the NMFs; one of the main requirements for the application of these methods to other use cases is that the input data is composed of observations for vertices in a rooted tree, as explained in more detail in the next section. Section 2 describes our proposed approach for estimating the full-information weighted least squares (WLS) estimator, which we call the Two-Pass Algorithm, since this algorithm performs operations recursively from the bottom of the tree upward and then, in a second stage, from the root of the tree downward. Section 3 describes a computationally efficient way to compute the covariance between the WLS estimates of arbitrary pairs of vertices of the tree. Section 4 describes how the proposed algorithms can be used to compute confidence intervals for user-defined queries in arbitrary geographic regions. Section 5 provides a numerical experiment of our proposed approach, to explore the accuracy of the resulting confidence interval estimates and the computational requirements of

Date: April, 2024.

The views expressed in this technical paper are those of the author and not those of the U.S. Census Bureau. The Census Bureau has reviewed this data product to ensure appropriate access, use, and disclosure avoidance protection of the confidential source data (Project No. 7502798, Disclosure Review Board (DRB) approval number: CBDRB-FY24-CED005-0002).

the proposed algorithms. Our proposed two-pass estimation approach is related to the approach described by Hay et al. (2010), which is described in Section 6 in more detail, and Section 7 concludes.

1.1. **Notation.** Throughout the paper we denote matrices using uppercase, and vectors using bold lowercase. Given the matrices $A, B \in \mathbb{R}^{N\times N}$ we use $A \leq B$ to denote the property that all eigenvalues of $B - A$ are real and the minimum eigenvalue of $B - A$ is non-negative. Given the random vectors $\boldsymbol{a} \in \mathbb{R}^M$ and $\boldsymbol{b} \in \mathbb{R}^N$, each with elements with finite variance, we use $\mathrm{Var}(\boldsymbol{a}) \in \mathbb{R}^{M\times M}$ to denote the variance matrix of $\boldsymbol{a}$ and $\mathrm{Cov}(\boldsymbol{a}, \boldsymbol{b}) \in \mathbb{R}^{M\times N}$ to denote the covariance matrix between $\boldsymbol{a}$ and $\boldsymbol{b}$. We denote the Kronecker product of real matrices A, B by $A \otimes B$. The length N column vector with each element equal to $c \in \mathbb{R}$ is denoted by $\boldsymbol{c}_N$ and an $M \times N$ matrix with each element equal to $C \in \mathbb{R}$ by $C_{M\times N}$. Also, when there is little risk of confusion, we omit the subscript and write $\boldsymbol{c}$ (respectively, C) instead. We denote the $n \times n$ identity matrix by I_n, the i^{th} row of $A \in \mathbb{R}^{m\times n}$ by $A[i, \cdot]$, and the i^{th} column of A by $A[\cdot, i]$. We also denote the set of nonnegative real numbers by $\mathbb{R}_+$, the positive reals by $\mathbb{R}_{++}$, and the nonnegative integers by $\mathbb{Z}_+$. We also denote the cardinality of the finite set S by $\mathrm{Card}(S)$.

1.2. **Setting.** We make use of some terms used by the Census Bureau to describe geographic entities, most of which can alternatively be translated into concepts from graph theory. For example, the Census Bureau's concept of a *geographic spine*, or *spine*, can alternatively be viewed as a rooted tree. Vertices in this rooted tree are called *geographic units*, or *geounits*, and each geounit is associated with a geographic region. For example, in the tabulation US spine, as defined for each decennial census by the Census Bureau's Geography Division, the root vertex is the US geounit, which has a geographic extent corresponding to all 50 states and Washington DC. Also, in the tabulation Puerto Rico (PR) spine, the root geounit is the PR geounit, corresponding to Puerto Rico as a whole. The geographic extent of each geounit on the spine corresponds to an element of a partition of the geographic extent of the root geounit. The collection of geounits corresponding to elements within the same partition is called a *geographic level*, or a *geolevel*. For example, in the tabulation US spine, the US geolevel contains only the US geounit, and the state geolevel consists of the children of the US geounit, *i.e.*, 51 geounits, each corresponding to either one of the 50 states or Washington DC. The other geolevels on the tabulation US spine are the county, census tract, block group, and block geolevels. Figure 1 provides a graphical representation of the tabulation US spine, including an example of a path in the 2020 vintage of this spine from a block to the US geounit.

While we use the terminology of the Census Bureau when describing the primary use case we have in mind, such as in the numerical experiment we present in Section 5, most often we use terminology from graph theory instead. Specifically, let the rooted tree(/spine) be denoted by $\mathcal{G}$. We use $g \in \mathcal{G}$ to denote vertex g of $\mathcal{G}$. To denote the subtree of $\mathcal{G}$ rooted at node $g \in \mathcal{G}$, *i.e.*, the subset of g that includes $g \in \mathcal{G}$ and all of its descendants, we use $\mathcal{G}_g$. We assume that $\mathcal{G}$ is defined so that the shortest path from any given leaf vertex to the root vertex all have a length equal to L, and for any level index $l \in \{0, \ldots, L\}$, we use $\mathrm{Level}(\mathcal{G}, l)$ to denote the set of vertices in level l. We also use $\mathrm{Children}(g)$ to denote the set of child vertices of $g \in \mathcal{G}$.

For each vertex $g \in \mathcal{G}$, we will denote the vector of (unknown) independent variables associated with the vertex by $\boldsymbol{\beta}(g) \in \mathbb{R}^n$. Throughout the paper, we assume that parent-child consistency holds; in other words, for any vertex $g \in \mathcal{G}$ that is not a leaf, we assume

$$\sum_{c\in\mathrm{Children}(g)} \boldsymbol{\beta}(c) = \boldsymbol{\beta}(g). \tag{1}$$

We also will associate with each vertex $g \in \mathcal{G}$ a coefficient matrix $S(g) \in \mathbb{R}^{m\times n}$, which we assume has full column rank. Note that we do not assume that the coefficient matrix of different vertices are identical.[1] For each vertex g, the vector of observed noisy measurements of g will be denoted by $\boldsymbol{y}(g) = S(g)\boldsymbol{\beta}(g) + \boldsymbol{u}(g)$, where each element of $\boldsymbol{u}(g) \in \mathbb{R}^m$ is an independently distributed random variable with zero mean and finite variance.

We perform various stacking operations on the attributes of the vertices below, *e.g.*, $\mathrm{stack}(\{S(g)\boldsymbol{\beta}(g)\}_{g\in\mathcal{G}})$, and these operations require that the vertices are ordered. Specifically, we assume this ordering ensures that,

[1] One implication of our use of m and n to denote the number of rows and columns of $S(g)$ throughout the paper is that this implicitly requires that these dimensions are the same for each vertex $g \in \mathcal{G}$; however, we actually only require the number of columns of $S(g)$, *i.e.*, $n \in \mathbb{N}$, to be the same for each vertex.

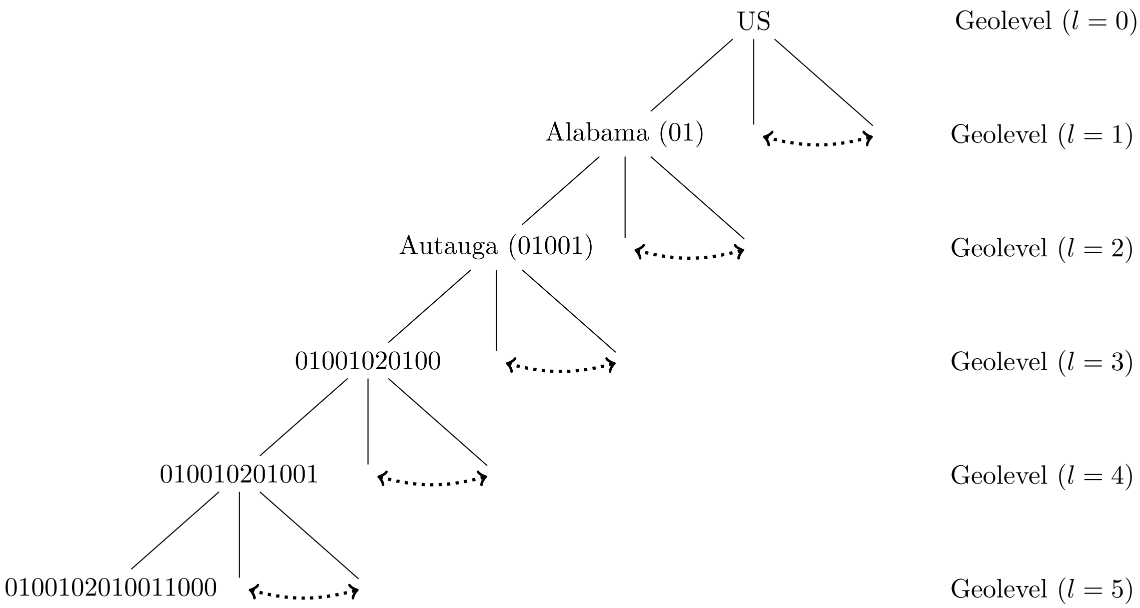


FIGURE 1. This is a graphical depiction of the 2020 tabulation US spine, as defined by the Geography Division of the Census Bureau. The right column provides the geolevel names and indices. The left side of the figure provides an example of a path from the US geounit to a block geounit. Specifically, in our ordering of geounits within each geolevel, this particular path passes through the first geounit in each geolevel.

if $0 \leq l < l' \leq L$, then each vertex in level l precedes vertices in level l' and also that the children of the u^{th} vertex of level l preceded the children of the $(u+1)^{\text{st}}$ vertex of level l. For example, the path provided in Figure 1 provides the first geounit in this ordering in each geolevel, along with the numerical geounit identifier codes, or *geocodes*, for these geounits in the 2020 tabulation US spine. These geocodes are formatted so that the first part of each geocode is the same as the geocode of the parent. Defining geocodes in this way is a straightforward way to define an ordering with both of our required properties, since this geounit ordering can be defined so that it corresponds to the ordering of the lexicographically-sorted list of these geocodes.[2]

It may also be helpful to see how these constructs fit into the use case that is the main motivation for this paper, *i.e.*, estimating arbitrary population cross tabulations for arbitrary geographic entities using the public census NMFs. In this setting, for each geounit $g \in \mathcal{G}$, $\boldsymbol{\beta}(g) \in \mathbb{Z}_+^n$ is defined as a vector of histogram counts, *i.e.*, a flattened fully saturated contingency table, $S(g) \in \{0,1\}^{m \times n}$ is called a strategy matrix, and $\boldsymbol{y}(g) := S(g)\boldsymbol{\beta}(g) + \boldsymbol{u}(g)$ is called the noisy measurements of geounit g. Also, in this use case, for each geounit $g \in \mathcal{G}$, $\boldsymbol{u}(g) \in \mathbb{Z}^m$ is a draw from the mean-zero discrete Gaussian distribution, as described by Canonne et al. (2020). One particular example we use in our numerical experiment in Section 5 is the case of the NMFs used for the persons universe of the Redistricting Data File. For each geounit, these tabulations are composed of cross products of the attributes of the schema of the histogram, which, as described by Abowd et al. (2022) in more detail, consist of the following attributes.

- Household or Group Quarters Type (HHGQ), 8 levels: Provides counts of those living in households, correctional facilities for adults, juvenile facilities, nursing facilities/skilled-nursing facilities, other institutional facilities, college/university student housing, military quarters, and those living in other noninstitutional facilities
- Voting Age (VOTING_AGE), 2 levels: Provides counts of individuals that are age 17 or younger and those that are age 18 or older

[2] More specifically, this ordering is defined by a sort operation that results in geocodes with a lower width preceding longer-width geocodes and geocodes with the same width, *i.e.*, geocodes of geounits within the same geolevel, being sorted lexicographically.

- Hispanic or Latino Origin (HISPANIC), 2 levels: Provides counts of individuals that are Hispanic/Latino and those that are not Hispanic or Latino
- Census Race (CENRACE), 63 levels: Provides counts of individuals that identify as each combination of Black/African American, American Indian/Native Alaskan, Asian, Native Hawaiian/Pacific Islander, White, and some other race, except "none of the above."

Note that U.S. Census Bureau (2023b) provides instructions for downloading the Redistricting Data File NMFs, both with unobserved histogram counts defined using 2010 and using 2020 data, as well as the NMFs for Demographic and Housing Characteristics (DHC) files DAS executions. The DHC NMFs use larger schemas than the Redistricting Data File schemas in order to support the estimation of more granular queries. For example, while the persons universe Redistricting Data File schema described above consists of $n = 2,016 = 8 \times 2 \times 2 \times 63$ detailed histogram cells for each geounit, the persons universe DHC schema consists of 1,227,744 detailed cells.

1.3. **Motivation: The Weighted Least Squares (WLS) Estimator.** In this section we will define the WLS estimator for $\{\boldsymbol{\beta}(g)\}_{g\in\mathcal{G}}$. To do so, note that our parent-child consistency assumption implies that, for any $g \in \mathcal{G}$, $S(g)\boldsymbol{\beta}(g)$ can be expressed as a linear function of $\{\boldsymbol{\beta}(g)\}_{g\in\text{Level}(\mathcal{G},L)}$. In other words, if we let $\boldsymbol{\beta} \in \mathbb{R}^N$ denote $\text{stack}(\{\boldsymbol{\beta}(g)\}_{g\in\text{Level}(\mathcal{G},L)})$, then there exists $S \in \mathbb{R}^{M\times N}$ such that the vector $\text{stack}(\{S(g)\boldsymbol{\beta}(g)\}_{g\in\mathcal{G}})$ is given by $S\boldsymbol{\beta}$. Similarly, let $\boldsymbol{u} \in \mathbb{R}^M$ denote the stacked error terms, *i.e.*, $\boldsymbol{u} := \text{stack}(\{\boldsymbol{u}(g)\}_{g\in\mathcal{G}})$, and let $\boldsymbol{y}$ denote the stacked vector of observed noisy measurements from all vertices, *i.e.*, $\boldsymbol{y} := \text{stack}(\{\boldsymbol{y}(g)\}_{g\in\mathcal{G}})$. Using this notation, we have $\boldsymbol{y} = S\boldsymbol{\beta} + \boldsymbol{u}$. Note that our assumptions in the previous section imply that, for every $g \in \mathcal{G}$, $S(g)$ has full column rank and that $\boldsymbol{u}(g)$ is a random variable such that $\text{Var}(\boldsymbol{u}(g))$ is a diagonal matrix with finite elements. The next example describes $\boldsymbol{u}, \boldsymbol{y}, \boldsymbol{\beta}$, and S for one case in which the tree $\mathcal{G}$ is particularly simple.

Example 1. *Suppose the tree $\mathcal{G}$ is composed of one root vertex, say g_0, and this vertex has two child vertices, say g_1 and g_2, respectively. In this case, we would have $\boldsymbol{u} = \mathit{stack}(\{\boldsymbol{u}(g_0), \boldsymbol{u}(g_1), \boldsymbol{u}(g_2)\})$, $\boldsymbol{y} = \mathit{stack}(\{\boldsymbol{y}(g_0), \boldsymbol{y}(g_1), \boldsymbol{y}(g_2)\})$, and $\boldsymbol{\beta} := \mathit{stack}(\{\boldsymbol{\beta}(g_1), \boldsymbol{\beta}(g_2)\})$. Also, S would be defined by,*

$$S = \begin{bmatrix} S(g_0) & S(g_0) \\ S(g_1) & 0 \\ 0 & S(g_2) \end{bmatrix}.$$

We can use this notation to define the full-information WLS estimator and its variance matrix as

$$\tilde{\boldsymbol{\beta}} := \arg\min_{\boldsymbol{\beta}} (S\boldsymbol{\beta} - \boldsymbol{y})^\top \text{Var}(\boldsymbol{u})^{-1}(S\boldsymbol{\beta} - \boldsymbol{y}) = \left(S^\top \text{Var}(\boldsymbol{u})^{-1} S\right)^{-1} S^\top \text{Var}(\boldsymbol{u})^{-1}\boldsymbol{y} \tag{2}$$

$$\text{Var}(\tilde{\boldsymbol{\beta}}) := \left(S^\top \text{Var}(\boldsymbol{u})^{-1} S\right)^{-1}. \tag{3}$$

The motivation for this paper is to describe a computationally efficient method that takes as input the observed noisy measurements $\boldsymbol{y} \in \mathbb{R}^M$, an arbitrary vector $\boldsymbol{q} \in \mathbb{R}^n$, and an arbitrary subset of leaf vertices $H \subset \text{Level}(\mathcal{G}, L)$, and then returns the best linear unbiased estimate of

$$\beta_{H,\boldsymbol{q}} := \sum_{g\in H} \boldsymbol{q}^\top \boldsymbol{\beta}(g), \tag{4}$$

along with the variance of this estimate and its CI. Note that for graphs with a particularly low number of leaf vertices, so that $\text{Var}(\tilde{\boldsymbol{\beta}})$ is not a high dimensional matrix, the standard approach to WLS estimation could be carried out in a straightforward manner. Specifically, if we define $\boldsymbol{h} \in \{0,1\}^{\text{Card}(\text{Level}(\mathcal{G},L))}$ so that $\boldsymbol{h}[i] = 1$ if and only if the i^{th} leaf vertex is in H, then this estimator and its variance would be given by

$$\tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}} = (\boldsymbol{h} \otimes \boldsymbol{q})^\top \tilde{\boldsymbol{\beta}} \tag{5}$$

$$\text{Var}(\tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}}) = (\boldsymbol{h} \otimes \boldsymbol{q})^\top \text{Var}(\tilde{\boldsymbol{\beta}})(\boldsymbol{h} \otimes \boldsymbol{q}), \tag{6}$$

These two values could in turn be used to estimate confidence intervals using a Gaussian approximation for $\tilde{\boldsymbol{\beta}}$.

In this paper, we describe how to compute $\tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}}$, $\mathrm{Var}(\tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}})$, and the corresponding CI, without explicitly computing and storing the matrix $\mathrm{Var}(\tilde{\boldsymbol{\beta}})$. The use cases we have in mind are ones in which $\mathrm{Var}(\tilde{\boldsymbol{\beta}})$ is too high dimensional to compute or store explicitly, primarily because $\mathcal{G}$ has a high number of leaves. For example, the Redistricting Data File NMFs contain noisy measurements for approximately six million blocks, so the outer inverse in (3) required to define $\mathrm{Var}(\tilde{\boldsymbol{\beta}})$ directly would not be possible to evaluate in practice, even when n, *i.e.*, the number of columns of each $\{S(g)\}_{g\in\mathcal{G}}$, is low. In contrast, our proposed approach does require that n is small enough to allow for the efficient evaluation of inverses of $n \times n$ matrices, since our proposed algorithm does evaluate inverses of matrices of this size explicitly. When using standard methods to evaluate matrix inverses that are used in practice, our proposed algorithm has a time complexity and a memory requirement that is $O(n^3 \sum_l \mathrm{Card}(\mathrm{Level}(\mathcal{G}, l)))$.[3] Our proposed approach works by precomputing a set of state variables for each vertex of $\mathcal{G}$ using the two-pass algorithm that we describe in the next section. We also describe algorithms that take these returned state variables as input to compute $\tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}}$, $\mathrm{Var}(\tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}})$, and the corresponding WLS-based $1-\alpha$ confidence level CI.

1.4. **Previous Literature.** Our proposed estimation algorithm is related to two existing algorithms. First, the approach described here is a generalization of the approach described by Hay et al. (2010), in the sense that these two approaches provide identical estimates, *i.e.*, $\tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}}$, when $m = n = 1$, $\mathrm{Var}(\boldsymbol{u}(g))$ is the same for each vertex $g \in \mathcal{G}$, and the number of children of each parent vertex is constant. More detail on this connection is provided in an example in Section 6. Note that, in addition to providing an algorithm for the WLS estimate itself, we also provide a method that computes the exact variance of the estimate, rather than a bound on this variance.

Second, there are also similarities between the approach described here and *forward-backward* algorithms described in the literature on hidden Markov models on trees; for example, Willsky (2002) describes several related approaches for WLS estimation using hierarchical data. At first, these approaches may seem only tangentially related because the most standard model in this literature does not satisfy our parent-child consistency assumption in equation (1), and instead assumes that, for each child $c \in \mathcal{G}$ with parent $g \in \mathcal{G}$, $\boldsymbol{\beta}(c)$ is given by $\boldsymbol{\beta}(c) = A(c)\boldsymbol{\beta}(g) + w(c)$, where $A(c) \in \mathbb{R}^{n\times n}$ and $w(c) \in \mathbb{R}^n$ is a random variable, which is known as the Markov property. In other words, while we define $\{\boldsymbol{\beta}(g)\}_{g\in\mathcal{G}}$ in a non-random and bottom-up manner, the hidden Markov model literature normally defines these vectors using a random process that proceeds in a top-down manner. However, Willsky (2002) describes Markov models on trees that are said to be *degenerate*, *i.e.*, models where the Markov property is not required to hold, and other models that are said to be *internal*, *i.e.*, models in which the histogram cell counts of each parent vertex g depends on $\{\boldsymbol{\beta}(c)\}_{c\in\mathrm{Children}(g)}$. Thus, the algorithm described in this paper can be viewed as an example of a forward-backward estimation approach from the hidden Markov model literature to estimate a model that is both degenerate and internal.

1.5. **A Preliminary Result.** We make use of the inverse-variance weighted mean for a vector at multiple points in this paper. The next Lemma provides the formula for the resulting vector and its variance estimate.

Lemma 1. *Suppose $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^n$ are realizations of random variables, each with finite variances and with a mean equal to $\boldsymbol{\psi} \in \mathbb{R}^n$. Then, the minimum variance unbiased estimate of $\boldsymbol{\psi}$ that is linear in $\boldsymbol{a}, \boldsymbol{b}$ is given by*

$$\hat{\boldsymbol{\psi}} = Var(\hat{\boldsymbol{\psi}}) \left(Var(\boldsymbol{a})^{-1}\boldsymbol{a} + Var(\boldsymbol{b})^{-1}\boldsymbol{b} \right)$$

where

$$Var(\hat{\boldsymbol{\psi}}) = \left(Var(\boldsymbol{a})^{-1} + Var(\boldsymbol{b})^{-1} \right)^{-1}$$

Proof. Let $\boldsymbol{c} := (\boldsymbol{a}^\top, \boldsymbol{b}^\top)^\top$, $X := \mathrm{stack}(\{I_n, I_n\})$, and let the block matrix Ω be defined as

$$\Omega := \begin{bmatrix} \mathrm{Var}(\boldsymbol{a}) & 0 \\ 0 & \mathrm{Var}(\boldsymbol{b}) \end{bmatrix}.$$

[3] Note that we describe a way that our proposed approaches can be used to estimate unbiased point estimates, variances, and confidence intervals of estimates for queries in the much larger DHC schemas, despite the fact that n is prohibitively large in these cases for our proposed methods to be applied directly, in Section 4.1.

Now consider applying a WLS estimator to a dataset with the i^{th} independent and dependent variables given by Y_i and $X_{i,\cdot}^\top$, respectively. The resulting estimator is equal to

$$\check{\boldsymbol{\psi}} = \left(X^\top \Omega^{-1} X\right)^{-1} X^\top \Omega^{-1} \boldsymbol{c},$$

which has variance given by

$$\mathrm{Var}(\check{\boldsymbol{\psi}}) = \left(X^\top \Omega^{-1} X\right)^{-1}.$$

Also, the Gauss-Markov theorem implies this estimator is the best linear unbiased estimator for $\boldsymbol{\psi}$; see for example, (Davidson et al., 2004). The final result follows from

$$\mathrm{Var}(\check{\boldsymbol{\psi}}) = \left(\mathrm{Var}(\boldsymbol{a})^{-1} + \mathrm{Var}(\boldsymbol{b})^{-1}\right)^{-1} = \mathrm{Var}(\hat{\boldsymbol{\psi}}),$$

and

$$\check{\boldsymbol{\psi}} = \left(X^\top \Omega^{-1} X\right)^{-1} X^\top \Omega^{-1} Y = \mathrm{Var}(\hat{\boldsymbol{\psi}}) \left(\mathrm{Var}(\boldsymbol{a})^{-1}\boldsymbol{a} + \mathrm{Var}(\boldsymbol{b})^{-1}\boldsymbol{b}\right) = \hat{\boldsymbol{\psi}}.$$

□

2. The Two-Pass Algorithm

Before describing the main algorithm in this section, it will be helpful to describe the primary data structures that we use. As described in the following subsections, our main proposed algorithm performs a series of updates on state variables within each vertex $g \in \mathcal{G}$, which includes an estimate of $\boldsymbol{\beta}(g)$ and its variance matrix. To reduce space requirements while running this algorithm, note that it is often possible to delete prior state variables after performing some of the updates described below. Table 1 summarizes some notation we use in this section, some of which is introduced in the following subsections.

Similar to the forward-backward algorithms described by Willsky (2002), the result of the two-pass algorithm described below is the BLUE for $\boldsymbol{\beta}(g)$ for each vertex $g \in \mathcal{G}$. This is because, for each vertex $g \in \mathcal{G}$, each of the three updates to the state variable estimate of $\beta(g)$ is given by the BLUE for a progressively larger information set, *i.e.*, a subset of the noisy measurements. Specifically, the first step of the algorithm is to define this state variable estimate of each vertex g as the WLS estimator based only on the noisy measurements of vertex g. Given our assumptions that $S(g)$ has full column rank and $\boldsymbol{u}(g)$ has a mean of zero and finite variance, the Gauss-Markov theorem implies that this initial WLS estimate is the BLUE for this particular information set; see for example, (Davidson et al., 2004). In the bottom-up pass, this state variable is updated as the inverse-variance-weighted mean of this initial estimate and an estimate defined as the sum over the bottom-up pass estimates of the children of vertex g. As described in Lemma 1, using the inverse-variance-weighted mean results in the linear unbiased estimator with the lowest possible variance, so the bottom-up pass update defines the estimate of $\boldsymbol{\beta}(g)$ as the BLUE for the information set given by the noisy measurements of the graph $\mathcal{G}_g$. Similarly, in the top-down pass, this information set is expanded further to also include the noisy measurements of higher levels of $\mathcal{G}$ by projecting the estimate from the bottom-up pass onto the column space of S, using a projection defined as the minimizer of the inverse-variance-weighted least squares objective function. This step also ensures the final estimate satisfies the same parent-child consistency property as the (unobserved) vector of histogram counts, *i.e.*, equation (1). As described in more detail in Theorem 1 below, the variance-minimizing projection is also used to define this final state variable update.

2.1. **Bottom-Up Pass: Initialization Step.** The first step of the algorithm is to initialize state variables of each vertex $g \in \mathcal{G}$ as

$$\hat{\boldsymbol{\beta}}(g|g) := \left(S(g)^\top \mathrm{Var}(\boldsymbol{u}(g))^{-1} S(g)\right)^{-1} S(g)^\top \mathrm{Var}(\boldsymbol{u}(g))^{-1} \boldsymbol{y}(g), \tag{7}$$

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}(g|g)) := \left(S(g)^\top \mathrm{Var}(\boldsymbol{u}(g))^{-1} S(g)\right)^{-1}. \tag{8}$$

For each of the leaf of the tree, *i.e.*, each vertex in $\mathrm{Level}(\mathcal{G}, L)$, also let

$$\hat{\boldsymbol{\beta}}(g|g-) := \hat{\boldsymbol{\beta}}(g|g) \tag{9}$$

$$\mathrm{Var}(\hat{\boldsymbol{\beta}}(g|g-)) := \mathrm{Var}(\hat{\boldsymbol{\beta}}(g|g)) \tag{10}$$

$\mathcal{G}$	The rooted tree, or, in our main motivating use case, the geographic spine
$\text{Children}(g)$	The set of vertices in $\mathcal{G}$ that are children of g
$\mathcal{G}_g$	The subtree of $\mathcal{G}$ rooted at vertex g
$S(g) \in \mathbb{R}^{m\times n}$	The coefficient matrix for vertex g
$\boldsymbol{y}(g) \in \mathbb{R}^m$	The observed noisy measurements of vertex g
$\boldsymbol{u}(g) \in \mathbb{R}^m$	The error component of the noisy measurements of vertex g
$\text{Var}(\boldsymbol{z})$	The variance matrix of the random variable $\boldsymbol{z}$
$\boldsymbol{\beta}(g) \in \mathbb{R}^n$	The (unobserved) vector of independent variables for vertex g
$\hat{\boldsymbol{\beta}}(g\vert g) \in \mathbb{R}^n$	Estimate for vertex g based on $\boldsymbol{y}(g)$
$\hat{\boldsymbol{\beta}}(g\vert g-) \in \mathbb{R}^n$	Estimate for vertex g based on noisy measurements in $\mathcal{G}_g$
$\tilde{\boldsymbol{\beta}}(g) \in \mathbb{R}^n$	Estimate for vertex g based on the observed random variables of all vertices in $\mathcal{G}$

TABLE 1. Some of the common notational conventions we use throughout the paper

2.2. **Bottom-Up Pass: Recursion Step.** The updates described in this section are performed first for vertices in $\text{Level}(\mathcal{G}, L-1)$, and afterward in each level moving up the tree. Specifically, given $\{\hat{\boldsymbol{\beta}}(c|c-)\}_{c\in\text{Children}(g)}$ and $\{\text{Var}(\hat{\boldsymbol{\beta}}(c|c-))\}_{c\in\text{Children}(g)}$, let

$$\hat{\boldsymbol{\beta}}(g|\text{Children}(g)-) := \sum_{c\in\text{Children}(g)} \hat{\boldsymbol{\beta}}(c|c-), \tag{11}$$

$$\text{Var}(\hat{\boldsymbol{\beta}}(g|\text{Children}(g)-)) := \sum_{c\in\text{Children}(g)} \text{Var}(\hat{\boldsymbol{\beta}}(c|c-)). \tag{12}$$

Now, since we have two independent estimates of $\boldsymbol{\beta}(g)$, *i.e.*, $\hat{\boldsymbol{\beta}}(g|g)$ and $\hat{\boldsymbol{\beta}}(g|\text{Children}(g)-)$, for each vertex g, we can define a new estimate as the linear combination of these estimates that has the lowest possible variance, which is the invariance-variance weighted mean of these two vector estimates, as described in Lemma 1. In other words, in this step, we update the state variables using

$$\text{Var}(\hat{\boldsymbol{\beta}}(g|g-)) := \Big(\text{Var}(\hat{\boldsymbol{\beta}}(g|g))^{-1} + \text{Var}(\hat{\boldsymbol{\beta}}(g|\text{Children}(g)-))^{-1}\Big)^{-1} \tag{13}$$

$$\hat{\boldsymbol{\beta}}(g|g-) := \text{Var}(\hat{\boldsymbol{\beta}}(g|g-))\Big(\text{Var}(\hat{\boldsymbol{\beta}}(g|g))^{-1}\hat{\boldsymbol{\beta}}(g|g) + \text{Var}(\hat{\boldsymbol{\beta}}(g|\text{Children}(g)-))^{-1}\hat{\boldsymbol{\beta}}(g|\text{Children}(g)-)\Big). \tag{14}$$

2.3. **Top-Down Pass: Initialization Step.** Prior to starting the top-down pass, we initialize the state variables of the root vertex g as

$$\tilde{\boldsymbol{\beta}}(g) := \hat{\boldsymbol{\beta}}(g|g-) \tag{15}$$

$$\text{Var}(\tilde{\boldsymbol{\beta}}(g)) := \text{Var}(\hat{\boldsymbol{\beta}}(g|g-)). \tag{16}$$

2.4. **Top-Down Pass: Recursion Step.** The updates in the top-down pass can be viewed as solutions to the optimization problem

$$\begin{aligned} \{\tilde{\boldsymbol{\beta}}(c)\}_{c\in\text{Children}(g)} = \mathop{\arg\min}_{\{\boldsymbol{\beta}(c)\}_{c\in\text{Children}(g)}} \sum_{c\in\text{Children}(g)} \|\text{Var}(\hat{\boldsymbol{\beta}}(c|c-))^{-1/2}\Big(\hat{\boldsymbol{\beta}}(c|c-) - \boldsymbol{\beta}(c)\Big)\|_2^2 \text{ such that:} \\ \sum_{c\in\text{Children}(g)} \boldsymbol{\beta}(c) = \tilde{\boldsymbol{\beta}}(g). \end{aligned} \tag{17}$$

The Karush–Kuhn–Tucker (KKT) conditions of this optimization problem are

$$\begin{aligned} \text{Var}(\hat{\boldsymbol{\beta}}(c|c-))^{-1}\Big(\tilde{\boldsymbol{\beta}}(c) - \hat{\boldsymbol{\beta}}(c|c-)\Big) = \boldsymbol{\lambda} \ \forall\, c \in \text{Children}(g) \\ \sum_{c\in\text{Children}(g)} \tilde{\boldsymbol{\beta}}(c) = \tilde{\boldsymbol{\beta}}(g). \end{aligned}$$

These two conditions imply that $\boldsymbol{\lambda}$ can be found using

$$\begin{aligned}\sum_{c\in\text{Children}(g)} \tilde{\boldsymbol{\beta}}(c) - \hat{\boldsymbol{\beta}}(c|c-) &= \sum_{c\in\text{Children}(g)} \text{Var}(\hat{\boldsymbol{\beta}}(c|c-))\boldsymbol{\lambda} \\ \implies \tilde{\boldsymbol{\beta}}(g) - \sum_{c\in\text{Children}(g)} \hat{\boldsymbol{\beta}}(c|c-) &= \sum_{c\in\text{Children}(g)} \text{Var}(\hat{\boldsymbol{\beta}}(c|c-))\boldsymbol{\lambda} \\ \implies \boldsymbol{\lambda} &= \left(\sum_{c\in\text{Children}(g)} \text{Var}(\hat{\boldsymbol{\beta}}(c|c-))\right)^{-1} \left(\tilde{\boldsymbol{\beta}}(g) - \sum_{c\in\text{Children}(g)} \hat{\boldsymbol{\beta}}(c|c-)\right).\end{aligned}$$

After substituting this value of $\boldsymbol{\lambda}$ into the first KKT condition, we have

$$\tilde{\boldsymbol{\beta}}(c) := \hat{\boldsymbol{\beta}}(c|c-) + A(c)\left(\tilde{\boldsymbol{\beta}}(g) - \sum_{c'\in\text{Children}(g)} \hat{\boldsymbol{\beta}}(c'|c'-)\right), \tag{18}$$

where

$$A(c) := \text{Var}(\hat{\boldsymbol{\beta}}(c|c-))\left(\sum_{c'\in\text{Children}(g)} \text{Var}(\hat{\boldsymbol{\beta}}(c'|c'-))\right)^{-1}. \tag{19}$$

Next we will derive the update to the variance matrix state variable, *i.e.*, $\text{Var}(\tilde{\boldsymbol{\beta}}(c))$. To do so, first let $\widetilde{Q}_g$ be defined so that $\tilde{\boldsymbol{\beta}}(g) = \tau + \widetilde{Q}_g \sum_{c\in\text{Children}(g)} \hat{\boldsymbol{\beta}}(c|c-)$, where τ is a random variable that is independent of $\{\hat{\boldsymbol{\beta}}(c|c-)\}_{c\in\text{Children}(g)}$, *i.e.*, τ is the component of $\tilde{\boldsymbol{\beta}}(g)$ that is a linear function of the noisy measurements of vertex g and the vertices that are ancestors of g. Also, let $B_c := \hat{\boldsymbol{\beta}}(c|c-) - A(c)\sum_{c'\in\text{Children}(g)} \hat{\boldsymbol{\beta}}(c'|c'-)$. To derive $\text{Var}(\tilde{\boldsymbol{\beta}}(c))$, we will first derive a few intermediate properties. First, the definition of $A(c)$ implies

$$A(c)\left(\sum_{c'\in\text{Children}(g)} \text{Var}(\hat{\boldsymbol{\beta}}(c'|c'-))\right) = \text{Var}(\hat{\boldsymbol{\beta}}(c|c-)). \tag{20}$$

Next, we have

$$\begin{aligned}\text{Cov}(B_c, \tilde{\boldsymbol{\beta}}(g)) &= \text{Cov}(\hat{\boldsymbol{\beta}}(c|c-) - A(c)\sum_{c'\in\text{Children}(g)} \hat{\boldsymbol{\beta}}(c'|c'-), \tau + \widetilde{Q}_g \sum_{c'\in\text{Children}(g)} \hat{\boldsymbol{\beta}}(c'|c'-)) \\ &= \text{Var}(\hat{\boldsymbol{\beta}}(c|c-))\widetilde{Q}_g^\top - A(c)\left(\sum_{c'\in\text{Children}(g)} \text{Var}(\hat{\boldsymbol{\beta}}(c'|c'-))\right)\widetilde{Q}_g^\top \\ &= \Big(\text{Var}(\hat{\boldsymbol{\beta}}(c|c-)) - \text{Var}(\hat{\boldsymbol{\beta}}(c|c-))\Big)\widetilde{Q}_g^\top = 0, \qquad (21)\end{aligned}$$

where the penultimate equality follows from (20).

Equation (21) implies that $\text{Var}(\tilde{\boldsymbol{\beta}}(c))$ can be written as

$$\begin{aligned}\text{Var}(\tilde{\boldsymbol{\beta}}(c)) &= \text{Cov}(\tilde{\boldsymbol{\beta}}(c), \tilde{\boldsymbol{\beta}}(c)) = \text{Cov}(B_c + A(c)\tilde{\boldsymbol{\beta}}(g), B_c + A(c)\tilde{\boldsymbol{\beta}}(g)) \\ &= \text{Cov}(B_c, B_c) + \text{Cov}(B_c, A(c)\tilde{\boldsymbol{\beta}}(g)) + \text{Cov}(A(c)\tilde{\boldsymbol{\beta}}(g), B_c) + \text{Cov}(A(c)\tilde{\boldsymbol{\beta}}(g), A(c)\tilde{\boldsymbol{\beta}}(g)) \\ &= \text{Var}(B_c) + A(c)\text{Var}(\tilde{\boldsymbol{\beta}}(g))A(c)^\top \\ &= \text{Var}(\hat{\boldsymbol{\beta}}(c|c-)) - \text{Var}(\hat{\boldsymbol{\beta}}(c|c-))A(c)^\top - A(c)\text{Var}(\hat{\boldsymbol{\beta}}(c|c-)) \\ &+ A(c)\left(\sum_{c'\in\text{Children}(g)} \text{Var}(\hat{\boldsymbol{\beta}}(c'|c'-))\right)A(c)^\top + A(c)\text{Var}(\tilde{\boldsymbol{\beta}}(g))A(c)^\top,\end{aligned}$$

so this and equation (20) imply

$$\begin{aligned}\mathrm{Var}(\tilde{\boldsymbol\beta}(c)) &= \mathrm{Var}(\hat{\boldsymbol\beta}(c|c-)) - \mathrm{Var}(\hat{\boldsymbol\beta}(c|c-))A(c)^\top - A(c)\mathrm{Var}(\hat{\boldsymbol\beta}(c|c-))\\ &+ \mathrm{Var}(\hat{\boldsymbol\beta}(c|c-))A(c)^\top + A(c)\mathrm{Var}(\tilde{\boldsymbol\beta}(g))A(c)^\top\\ &= \mathrm{Var}(\tilde{\boldsymbol\beta}(c)) := \mathrm{Var}(\hat{\boldsymbol\beta}(c|c-)) - A(c)\mathrm{Var}(\hat{\boldsymbol\beta}(c|c-)) + A(c)\mathrm{Var}(\tilde{\boldsymbol\beta}(g))A(c)^\top. \end{aligned}\tag{22}$$

The following theorem describes the way in which the projection defining the estimator $\tilde{\boldsymbol\beta}(c)$ is optimal. This theorem is proved in Appendix A.

Theorem 1. *Suppose that the following conditions hold.*

(a) $stack(\{\check{\boldsymbol\beta}(c)\}_{c\in Children(g)}) \in \mathbb{R}^{n\,Card(Children(g))}$ *is any unbiased estimator of* $stack(\{\boldsymbol\beta(c)\}_{c\in Children(g)})$

(b) $stack(\{\check{\boldsymbol\beta}(c)\}_{c\in Children(g)})$ *is linear in* $\tilde{\boldsymbol\beta}(g)$ *and each element of* $\{\hat{\boldsymbol\beta}(c|c-)\}_{c\in Children(g)}$

(c) $\sum_{c\in Children(g)} \check{\boldsymbol\beta}(c) = \tilde{\boldsymbol\beta}(g)$

Out of all such estimators, the estimator $\{\tilde{\boldsymbol\beta}(c)\}_{c\in Children(g)}$ *defined in (18), is optimal, in the sense that,*

$$Var(stack(\{\tilde{\boldsymbol\beta}(c)\}_{c\in Children(g)})) \le Var(stack(\{\check{\boldsymbol\beta}(c)\}_{c\in Children(g)})).$$

Algorithm 1 provides pseudocode that summarizes this two-pass algorithm, but a description of how this algorithm fits into the motivation for this paper, as described in Section 1.3, may also be helpful. Specifically, the final state variables $\{\tilde{\boldsymbol\beta}(u)\}_{u\in\mathcal{G}}$ and $\{\mathrm{Var}(\tilde{\boldsymbol\beta}(u))\}_{u\in\mathcal{G}}$ are directly related to the standard WLS estimate, $\tilde{\boldsymbol\beta}$, and its variance matrix, as defined in (2-3), because $\tilde{\boldsymbol\beta} = \mathrm{stack}(\{\tilde{\boldsymbol\beta}(u)\}_{u\in\mathrm{Level}(\mathcal{G},L)})$ and the diagonal blocks of $\mathrm{Var}(\tilde{\boldsymbol\beta})$ are given by $\{\mathrm{Var}(\tilde{\boldsymbol\beta}(u))\}_{u\in\mathrm{Level}(\mathcal{G},L)}$. This implies that, after running this two-pass algorithm, we can take as input $\boldsymbol{q}\in\mathbb{R}^n$ and $H\subset\mathrm{Level}(\mathcal{G},L)$ and output $\tilde\beta_{H,\boldsymbol q}$, as defined in (5). In cases in which H is the set of leaves that are descendants of a given vertex $u\in\mathcal{G}$, we could also compute $\mathrm{Var}(\tilde\beta_{H,\boldsymbol q})$ using $\boldsymbol q^\top\mathrm{Var}(\tilde{\boldsymbol\beta}(u))\boldsymbol q$, but we would not be able to compute $\mathrm{Var}(\tilde\beta_{H,\boldsymbol q})$ in more general cases. The next section describes how to compute $\mathrm{Cov}(\tilde{\boldsymbol\beta}(u),\tilde{\boldsymbol\beta}(v))$ for arbitrary $u,v\in\mathcal{G}$, which, as described in Section 4 below in more detail, can be used to compute $\mathrm{Var}(\tilde\beta_{H,\boldsymbol q})$ in all cases.

Algorithm 1: Two-Pass_Estimation: Returns matrices and vectors that can be used in Algorithm 3 to generate point estimates and confidence intervals of user-defined queries for an arbitrary set of leaves

input : $\mathcal{G}$: The rooted tree, with noisy measurements and coefficient matrices associated with each vertex

input : $\{\boldsymbol y(g), S(g), \mathrm{Var}(\boldsymbol u(g))\}_{g\in\mathcal{G}}$: The set of noisy measurements, coefficient matrices, and variance matrices of the noisy measurements for each vertex $g\in\mathcal{G}$

`//` For each vertex g in graph $\mathcal{G}$, use equations (7) and (8) to define $\hat{\boldsymbol\beta}(g|g)$ and $\mathrm{Var}(\hat{\boldsymbol\beta}(g|g))$, respectively. For each vertex $g\in\mathrm{Level}(\mathcal{G},L)$, let $\hat{\boldsymbol\beta}(g|g-)$ and $\mathrm{Var}(\hat{\boldsymbol\beta}(g|g-))$ be defined as $\hat{\boldsymbol\beta}(g|g)$ and $\mathrm{Var}(\hat{\boldsymbol\beta}(g|g))$, respectively, as described in equations (9) and (10).

`//` For each l in $L-1,\dots,0$ and each vertex $g\in\mathrm{Level}(\mathcal{G},l)$, use equations (14) and (13) to define $\hat{\boldsymbol\beta}(g|g-)$ and $\mathrm{Var}(\hat{\boldsymbol\beta}(g|g-))$.

`//` Initialize the top-down pass using (15) and (16), *i.e.*, for g given by the root vertex, let $\tilde{\boldsymbol\beta}(g)$ and $\mathrm{Var}(\tilde{\boldsymbol\beta}(g))$ be defined as $\hat{\boldsymbol\beta}(g|g-)$ and $\mathrm{Var}(\hat{\boldsymbol\beta}(g|g-))$, respectively.

`//` For each level l in $1,\dots,L$ and each vertex $c\in\mathrm{Level}(\mathcal{G},l)$, define $A(c),\tilde{\boldsymbol\beta}(c)$ and $\mathrm{Var}(\tilde{\boldsymbol\beta}(c))$ using (19), (18), and (22), respectively.

return: $\{\tilde{\boldsymbol\beta}(g), \mathrm{Var}(\tilde{\boldsymbol\beta}(g)), A(g), \mathrm{Var}(\hat{\boldsymbol\beta}(g|g-))\}_{g\in\mathcal{G}}$

3. Covariance Between Estimates of Arbitrary Vertices

Similar derivations that were used to derive $\mathrm{Var}(\tilde{\boldsymbol{\beta}}(c))$ in the preceding section can also be carried out to find the covariance $\mathrm{Cov}(\tilde{\boldsymbol{\beta}}(u), \tilde{\boldsymbol{\beta}}(v))$ for arbitrary $u, v \in \mathcal{G}$. Specifically, this involves recursively substituting instances of $\tilde{\boldsymbol{\beta}}(\cdot)$ in the covariance function inputs for its definition in (18), until the only instance of $\tilde{\boldsymbol{\beta}}(\cdot)$ is the closest common ancestor of u and v in the tree $\mathcal{G}$. Afterward, these functional forms can be simplified significantly using algebraic simplifications and the property (20). We provide these covariance matrices, rather than the underlying derivations in detail, because the derivations themselves are both straightforward and tedious. It turns out that the resulting covariance matrices are very similar to the ones described in Section IV.4 by Willsky (2002), as we describe in more detail below.

Before describing these covariance matrices, some additional notation will be helpful. Specifically, let $u \wedge v$ denote the vertex that is the closest common ancestor of u and v. Also, let $\omega(u, v)$ denote the shortest path from u to v. We also denote the t^{th} element (respectively, the t^{th} element from the end) of this path by $\omega(u, v)[t-1]$ (respectively, $\omega(u, v)[-t]$). In other words, for $u, v \in \mathcal{G}$, the shortest path from u to v is given by $\omega(u, v)[0](= u), \omega(u, v)[1], \ldots, \omega(u, v)[-2], \omega(u, v)[-1](= v)$. Also, we use the same notation defined in the last subsection for B_c, *i.e.*, $B_c := \hat{\boldsymbol{\beta}}(c|c-) - A(c) \sum_{c' \in \mathrm{Children}(u \wedge v)} \hat{\boldsymbol{\beta}}(c'|c'-)$.

Next we will derive the covariance matrix $\mathrm{Cov}(\tilde{\boldsymbol{\beta}}(u), \tilde{\boldsymbol{\beta}}(v))$ for two specific cases. First, we consider the case in which $u, v \in \mathcal{G}$ are sibling vertices. Note that, because of property (20), we have

$$
\begin{aligned}
&\mathrm{Cov}(B_u, B_v) = A(u) \left(\sum_{c \in \mathrm{Children}(u \wedge v)} \mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c-)) \right) A(v)^\top - \mathrm{Var}(\hat{\boldsymbol{\beta}}(u|u-)) A(v)^\top - A(u) \mathrm{Var}(\hat{\boldsymbol{\beta}}(v|v-)) \\
&= \mathrm{Var}(\hat{\boldsymbol{\beta}}(u|u-)) A(v)^\top - \mathrm{Var}(\hat{\boldsymbol{\beta}}(u|u-)) A(v)^\top - A(u) \mathrm{Var}(\hat{\boldsymbol{\beta}}(v|v-)). \\
&= -A(u) \mathrm{Var}(\hat{\boldsymbol{\beta}}(v|v-)),
\end{aligned}
$$

which implies

$$
\begin{aligned}
&\mathrm{Cov}(\tilde{\boldsymbol{\beta}}(u), \tilde{\boldsymbol{\beta}}(v)) = \mathrm{Cov}(B_u + A(u)\tilde{\boldsymbol{\beta}}(u \wedge v), B_v + A(v)\tilde{\boldsymbol{\beta}}(u \wedge v)) \\
&= \mathrm{Cov}(B_u, B_v) + \mathrm{Cov}(B_u, A(v)\tilde{\boldsymbol{\beta}}(u \wedge v)) \\
&+ \mathrm{Cov}(A(u)\tilde{\boldsymbol{\beta}}(u \wedge v), B_v) + \mathrm{Cov}(A(u)\tilde{\boldsymbol{\beta}}(u \wedge v), A(v)\tilde{\boldsymbol{\beta}}(u \wedge v)) \\
&= \mathrm{Cov}(\tilde{\boldsymbol{\beta}}(u), \tilde{\boldsymbol{\beta}}(v)) = A(u)\mathrm{Var}(\tilde{\boldsymbol{\beta}}(u \wedge v))A(v)^\top - A(u)\mathrm{Var}(\hat{\boldsymbol{\beta}}(v|v-)).
\end{aligned} \tag{23}
$$

Next we will consider the case in which $g, c \in \mathcal{G}$ and $c \in \mathrm{Children}(g)$. In this case we have

$$
\begin{aligned}
&\mathrm{Cov}(\tilde{\boldsymbol{\beta}}(c), \tilde{\boldsymbol{\beta}}(g)) = \mathrm{Cov}(B_c + A(c)\tilde{\boldsymbol{\beta}}(g), \tilde{\boldsymbol{\beta}}(g)) \\
&= \mathrm{Cov}(B_c, \tilde{\boldsymbol{\beta}}(g)) + A(c)\mathrm{Cov}(\tilde{\boldsymbol{\beta}}(g), \tilde{\boldsymbol{\beta}}(g)) \\
&= \mathrm{Cov}(\tilde{\boldsymbol{\beta}}(c), \tilde{\boldsymbol{\beta}}(g)) = A(c)\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g)).
\end{aligned} \tag{24}
$$

Equations (23-24) can also be generalized to describe the functional form of $\mathrm{Cov}(\tilde{\boldsymbol{\beta}}(u), \tilde{\boldsymbol{\beta}}(v))$ for arbitrary $u, v \in \mathcal{G}$. This functional form closely resembles the one provided in Willsky (2002) for the covariance between the values of a random process at two vertices on a rooted tree. The functional form provided by Willsky (2002) is based on the recursive equation describing the coarse-to-fine dynamics of the process, which in our case is (18), and can be viewed as a series of operations involving matrices associated with vertices along the shortest path from u to v in $\mathcal{G}$, *e.g.*, in our case $\mathrm{Var}(\tilde{\boldsymbol{\beta}}(\cdot)), \mathrm{Var}(\hat{\boldsymbol{\beta}}(\cdot| \cdot -))$, and $A(\cdot)$. The covariance matrix provided by Willsky (2002) can be derived by starting from $u \wedge v$ by initializing the covariance matrix as $\mathrm{Var}(\tilde{\boldsymbol{\beta}}(u \wedge v))$ and then recursively moving outward on the path $\omega(u, v)$ from $u \wedge v$, to account for the relationship between $\tilde{\boldsymbol{\beta}}(u \wedge v)$ and vertices further away from $u \wedge v$ along this path. Our functional forms are slightly different than the ones provided by Willsky (2002) because the inclusion of term $-A(c) \sum_{c' \in \mathrm{Children}(g)} \hat{\boldsymbol{\beta}}(c'|c'-)$ in (18) prevents the random process resulting from the two pass-approach described in the previous section from satisfying the Markov property exactly. Specifically, while the Markov property requires that, for any $u, v \in \mathcal{G}$, $\tilde{\boldsymbol{\beta}}(u)$ and $\tilde{\boldsymbol{\beta}}(v)$ are independent after conditioning on only $\tilde{\boldsymbol{\beta}}(u \wedge v)$, we instead have independence between any $\tilde{\boldsymbol{\beta}}(u)$ and $\tilde{\boldsymbol{\beta}}(v)$ after conditioning on $\{\tilde{\boldsymbol{\beta}}(u \wedge v)\} \cup \{\tilde{\boldsymbol{\beta}}(c)\}_{c \in \text{Children}(u \wedge v)}$. This difference between our setting and the model considered by Willsky (2002) requires some additional care when reasoning about the relationship between elements of $\{\tilde{\boldsymbol{\beta}}(u \wedge v)\} \cup \{\tilde{\boldsymbol{\beta}}(c)\}_{c \in \text{Children}(u \wedge v)}$, so we will use the covariance matrices in (23-24) directly when accounting for these relationships.

First we consider cases in which $v \in \mathcal{G}$ is an ancestor of $u \in \mathcal{G}$. In this case, the resulting covariance matrix is identical to the ones provided in Section IV.4 by Willsky (2002). Specifically,

$$\begin{aligned} \operatorname{Cov}(\tilde{\boldsymbol{\beta}}(u), \tilde{\boldsymbol{\beta}}(v)) &= \left(\prod_{k \in \omega(u,v)/\{v, \omega(u,v)[-2]\}} A(k) \right) \operatorname{Cov}(\omega(u,v)[-2], \tilde{\boldsymbol{\beta}}(v)) \\ &= \left(\prod_{k \in \omega(u,v)/v} A(k) \right) \operatorname{Var}(\tilde{\boldsymbol{\beta}}(v)). \end{aligned} \tag{25}$$

Next we consider the case in which $u, v \in \mathcal{G}$ are not ancestors of one another. In this case, we have

$$\begin{aligned} \operatorname{Cov}(\tilde{\boldsymbol{\beta}}(u), \tilde{\boldsymbol{\beta}}(v)) = &\left(\prod_{k \in \omega(u,u \wedge v)/\{\omega(u,u\wedge v)[-2], u \wedge v\}} A(k) \right) \operatorname{Cov}(\tilde{\boldsymbol{\beta}}(\omega(u, u \wedge v)[-2]), \tilde{\boldsymbol{\beta}}(\omega(u \wedge v, v)[1])) \\ &\left(\prod_{k \in \omega(u \wedge v, v)/\{\omega(u \wedge v, v)[1], u \wedge v\}} A(k)^\top \right). \end{aligned} \tag{26}$$

Algorithm 2: Compute_Covariance: Computes the matrix $\operatorname{Cov}(\tilde{\boldsymbol{\beta}}(u), \tilde{\boldsymbol{\beta}}(v))$ for arbitrary $u, v \in \mathcal{G}$

---

input : $\mathcal{G}$: The rooted tree
input : $\{\tilde{\boldsymbol{\beta}}(g), \operatorname{Var}(\tilde{\boldsymbol{\beta}}(g)), A(g), \operatorname{Var}(\hat{\boldsymbol{\beta}}(g|g-))\}_{g \in \mathcal{G}}$: the output of Algorithm 1
input : $u, v \in \mathcal{G}$: This algorithm returns $\operatorname{Cov}(\tilde{\boldsymbol{\beta}}(u), \tilde{\boldsymbol{\beta}}(v))$
if $u = v$ **then**
  return: $\operatorname{Var}(\tilde{\boldsymbol{\beta}}(u))$
end
if $v = u \wedge v$ **then**
  `// See equation (25) for this case:`
  return: $\left(\prod_{k \in \omega(u,v)/v} A(k)\right) \operatorname{Var}(\tilde{\boldsymbol{\beta}}(v))$
end
if $u = u \wedge v$ **then**
  `// Switch the inputs` u, v `and transpose the output of this function:`
  return: $\text{Compute\_Covariance}\left(\mathcal{G}, \{\tilde{\boldsymbol{\beta}}(g), \operatorname{Var}(\tilde{\boldsymbol{\beta}}(g)), A(g), \operatorname{Var}(\hat{\boldsymbol{\beta}}(g|g-))\}_{g \in \mathcal{G}}, v, u\right)^\top$
end
`// The remaining case follows from (26):`
$u' \leftarrow \omega(u, u \wedge v)[-2]$
$v' \leftarrow \omega(u \wedge v, v)[1]$
$\operatorname{Cov}(\tilde{\boldsymbol{\beta}}(u'), \tilde{\boldsymbol{\beta}}(v')) \leftarrow A(u') \operatorname{Var}(\tilde{\boldsymbol{\beta}}(u' \wedge v')) A(v')^\top - A(u') \operatorname{Var}(\hat{\boldsymbol{\beta}}(v'|v'-))$
return: $\left(\prod_{k \in \omega(u,u')/\{u'\}} A(k)\right) \operatorname{Cov}(\tilde{\boldsymbol{\beta}}(u'), \tilde{\boldsymbol{\beta}}(v')) \left(\prod_{k \in \omega(v',v)/\{v'\}} A(k)^\top\right)$

When $u, v \in \text{Level}(\mathcal{G}, L)$, the derivations in this section can be used to derive the off-diagonal blocks of $\operatorname{Var}(\tilde{\boldsymbol{\beta}})$, as defined in (3), so, combined with the derivation for the diagonal blocks provided in the previous section, these results allow for the computation of arbitrary elements of the variance matrix of the standard WLS estimate.

Algorithm 2 provides pseudocode that summarizes the methods described in this section. The next section describes how Algorithms 1 and 3 can be used to compute $\tilde{\beta}_{H,\boldsymbol{q}}$, as defined in (5), its variance, and confidence intervals for this estimate.

4. Confidence Interval Estimation

The previous two sections describe each of the state variable updates required for the two-pass estimation approach and the computation of $\mathrm{Cov}(\tilde{\boldsymbol{\beta}}(u), \tilde{\boldsymbol{\beta}}(v))$ for arbitrary $u, v \in \mathcal{G}$, as summarized in Algorithms 1 and 2, respectively. It is straightforward to use these two algorithms to derive confidence intervals centered at $\tilde{\beta}_{H,\boldsymbol{q}}$, which is done below in Algorithm 3.

Algorithm 3: Estimate_Confidence_Interval: Returns the full-information BLUE count estimate and confidence interval of a user-defined query evaluated in an arbitrary subset of the leaf vertices of $\mathcal{G}$

input : $\mathcal{G}$: The rooted tree
input : $\{\tilde{\boldsymbol{\beta}}(g), \mathrm{Var}(\tilde{\boldsymbol{\beta}}(g)), A(g), \mathrm{Var}(\hat{\boldsymbol{\beta}}(g|g-))\}_{g\in\mathcal{G}}$: the output of Algorithm 1
input : $\boldsymbol{q} \in \mathbb{R}^n$: The linear counting query of interest
input : $H \subset \mathrm{Level}(\mathcal{G}, L)$: The subset of the leaves of $\mathcal{G}$ for which to estimate the linear query
input : α : This algorithm will return the $1-\alpha$ confidence interval of $\beta_{H,\boldsymbol{q}}$
vertices $\leftarrow \emptyset$
`//` For each level l in $L-1,\dots,0$ and for each vertex $g \in \mathrm{Level}(\mathcal{G}, l)$, if all children of g are in H, remove $\mathrm{Children}(g)$ from H and add vertex g to H.
$\mathrm{Var}(\tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}}) \leftarrow 0$
$\tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}} \leftarrow 0$
for $u \in H$ **do**
    $\tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}} \leftarrow \tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}} + \boldsymbol{q}^\top \tilde{\boldsymbol{\beta}}(u)$
    for $v \in H$ **do**
        $\mathrm{Cov}(\tilde{\boldsymbol{\beta}}(u), \tilde{\boldsymbol{\beta}}(v)) \leftarrow \mathrm{Compute\_Covariance}\Big(\mathcal{G}, \{\tilde{\boldsymbol{\beta}}(g), \mathrm{Var}(\tilde{\boldsymbol{\beta}}(g)), A(g), \mathrm{Var}(\hat{\boldsymbol{\beta}}(g|g-))\}_{g\in\mathcal{G}}, u, v\Big)$
        $\mathrm{Var}(\tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}}) \leftarrow \mathrm{Var}(\tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}}) + \boldsymbol{q}^\top \mathrm{Cov}(\tilde{\boldsymbol{\beta}}(u), \tilde{\boldsymbol{\beta}}(v))\boldsymbol{q}$
    end
end
`//` $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution:
$c \leftarrow \sqrt{\mathrm{Var}(\tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}})}\Phi^{-1}(1-\alpha/2)$
return: $\tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}}, (\tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}} - c, \tilde{\boldsymbol{\beta}}_{H,\boldsymbol{q}} + c)$

A number of theoretical properties can also be shown for the estimate $\tilde{\beta}_{H,\boldsymbol{q}}$ and its confidence interval using standard results on the weighted least squares estimator. Recall that in Section 1.3 we assumed that $S \in \mathbb{R}^{M\times N}$ has full column rank and that $\boldsymbol{u} \in \mathbb{R}^M$ is a vector of independent mean-zero random variables, each with a known and finite variance. As a result of these assumptions, the Gauss-Markov theorem implies that the estimate $\tilde{\beta}_{H,\boldsymbol{q}}$ that is output from Algorithm 3 has the smallest possible variance out of all estimators that are unbiased and linear in the noisy measurements $\boldsymbol{y}$; see for example, (Davidson et al., 2004). In addition, if $\boldsymbol{u}(g)$ is normally distributed for each $g \in \mathcal{G}$, then $\tilde{\beta}_{H,\boldsymbol{q}}$ is also normally distributed, which can be shown using standard finite sample properties of the WLS estimator (Davidson et al., 2004). As a result, when this additional normality condition holds, this finite sample property implies that the confidence intervals output from Algorithm 2 are statistically valid in this case.

When $\boldsymbol{u}(g)$ is not normally distributed (but our other assumptions still hold), we could use standard asymptotic theory for regressions to show the same result, *i.e.*, normality of $\tilde{\beta}_{H,\boldsymbol{q}}$ along with its same implication of statistical validity of the confidence intervals output from Algorithm 2, holds in some limiting cases Davidson et al. (2004). However, these limiting cases are not realistic in our primary motivating use case, as they require the number of noisy measurements to diverge, so we will not expand on these further.

After describing our implementation of Algorithm 3 in the next subsection, we will provide a numerical experiment to assess the accuracy of these confidence intervals in practice for our motivating use case. While $\boldsymbol{u}$ is not distributed as a Gaussian in this motivating use case, there are still reasons to be optimistic about the accuracy of the confidence intervals output from Algorithm 3; for example, as described in Section 1.2, the distribution of $\boldsymbol{u}$ is closely related to the Gaussian distribution, *i.e.*, it is the discrete Gaussian distribution Canonne et al. (2020).

4.1. **Computational Considerations.** For many use cases, the limiting factor for the feasibility of the algorithms proposed in this paper will be the need to store many dense $n \times n$ matrices. For example, one possible use case would be to apply these methods to the NMFs for the Redistricting Data File, and, as described above, in this case n is equal to $2,016$. Since our choice of output for Algorithm 1, *i.e.*, $\{\tilde{\boldsymbol{\beta}}(g), \mathrm{Var}(\tilde{\boldsymbol{\beta}}(g)), A(g), \mathrm{Var}(\hat{\boldsymbol{\beta}}(g|g-))\}_{g\in\mathcal{G}}$, requires storing three dense $n \times n$ matrices for every vertex and since $\mathcal{G}$ would have approximately 6 million leaf vertices when using the production US DAS execution Redistricting Data File NMFs, at least several hundreds of terabytes would be required to store the output of Algorithm 1 without using methods to reduce storage space in this case. This is problematic because it would be ideal to store the output of Algorithm 1 after it is run only once, so that this output can be used to support the computation of confidence intervals of linear query answers in arbitrary set of leaves over an extended period of time. With this in mind, this section describes multiple ways to reduce the storage requirements of the output of Algorithm 1.

For the primary motivating use case we have in mind, one of the most straightforward and effective ways to keep space requirements of the output of Algorithm 1 low would be to reduce the size of the histogram schema by marginalizing the NMFs over certain levels of attributes in the original schema. For example, in the case of the NMFs for the Redistricting Data File, rather than running Algorithm 1 on the full schema HHGQ $\times$ HISPANIC $\times$ VOTING_AGE $\times$ CENRACE, we could marginalize the NMFs over all attributes other than HHGQ, save the output of Algorithm 1, and then repeat this process for each of the three remaining attributes. Given the number of levels of each of these attributes, as provided in Section 1.2, this alternative would reduce the space requirements for all of the outputs of Algorithm 1 by approximately a factor of $1,005.8(\approx 8 \times 2 \times 2 \times 63/(8+2+2+63))$, and would also substantially reduce the computational requirements of the executions of this algorithm. Likewise, calling Algorithm 1 after marginalizing over each possible pair of these four attributes would reduce the space requirements by a factor of approximately 14.2 relative to using all attributes simultaneously. The downside of this approach is that we could only support the computation of user-defined queries that depend on only a subset of the attributes; however, in the vast majority of use cases for decennial census data, the queries of interest do not depend on all attributes of the full histogram schema used to generate the NMFs. For this reason, in practice, we expect that use cases using decennial census NMFs will most often only require $n \leq 2,016$. Note that, unlike the NMFs for Redistricting Data File DAS executions, the schemas used to generate the NMFs for the persons and units universes of the DHC DAS executions have more than 1.2 million detailed histogram cells, so marginalizing over attributes is a requirement for computational feasibility of our proposed approach in these cases.

There are also several algorithmic modifications that can be considered to reduce the size of the output of Algorithm 1. First, storage of the matrices $\{A(g)\}_g$ can optionally be avoided entirely, since these matrices can be recomputed using $\{\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g)), \mathrm{Var}(\hat{\boldsymbol{\beta}}(g|g-))\}_{g\in\mathcal{G}}$ when they are required. Second, one can also leverage the fact that, in many use cases, some of the matrices output from Algorithm 1 will be the same, to avoid storing multiple instances of duplicate matrices. For example, for our primary use case of interest, vertices in the lowest level of $\mathcal{G}$ have the matrix $\mathrm{Var}(\hat{\boldsymbol{\beta}}(g|g-))$ given by one of only two possible values. Third, one can use the fact that the variance matrices in $\{\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g)), \mathrm{Var}(\hat{\boldsymbol{\beta}}(g|g-))\}_{g\in\mathcal{G}}$ are symmetric to reduce the storage requirement of these matrices by approximately one half. Fourth, the storage requirement of the output of Algorithm 1 can also be reduced using a compressed format. In the numerical experiments provided in the next section, we use these final two methods to reduce the space requirements for the output of Algorithm 1.

5. Numerical Experiment

In this section we provide a numerical experiment using a Python/PySpark implementation of the algorithms we proposed above. For the purposes of this numerical experiment, this implementation was executed on Amazon Web Services (AWS) using Elastic Map Reduce (EMR) version 6.12 and Apache Spark version

Table 2. CI Empirical Coverage: For each of the four CIs considered, and each of the three sets of queries considered, this table provides the CI coverage, which is the proportion of CIs that contain the true CEF-based count out of all queries in the given query set for all block groups in PR.

	CI Type			
Query	0.90 CI	0.95 CI	0.90 CI, Nonnegative	0.95 CI, Nonnegative
All Redistricting Tabulations	0.8994	0.9496	0.9473	0.9737
Total Population	0.8922	0.9363	0.8922	0.9363
CENRACE	0.9002	0.9498	0.9479	0.9739

3.4. The DAS is run on EMR clusters built from AWS r5.24xlarge virtual machines, in which each virtual machine has 96 cores and 768GiB of RAM. We configured the cluster so that it had one primary node and 5 core nodes, and configured Spark so that there were 5 executors, each with 40 executor cores.

Rather than using all noisy measurements generated by a full US DAS execution, in this numerical experiment, we used the publicly available NMFs from the persons universe Redistricting Data File Puerto Rico (PR) DAS execution of the 2010 Demonstration Data Product; instructions for downloading these NMFs are available at U.S. Census Bureau (2023b). This DAS execution used settings that were identical to that of the 2020 production DAS execution that was used to publish the 2020 PR persons tables in the Redistricting Data File, other than the fact that the 2010 Census data was used as input instead of the 2020 Census data. As described previously, unlike the tabulation US spine described in Figure 1, the geographic spine that is used internally within the DAS implementation for PR executions is rooted at the geounit corresponding to PR as a whole, which is included in the state geolevel, and subsequent geolevels include county, tract, optimized block group, and the block geolevels. Optimized block groups are distinct from the census block groups that are included in the tabulation spine described above, so the 2,543 PR tabulation census block groups are not on the optimized spine used internally within DAS; for more detail on the optimized spine, see Cumings-Menon et al. (2022). Since these PR tabulation census block groups are not in the spine used to generate the NMFs, they provide interesting off-spine geographic entities for the purpose of testing our proposed estimation approaches, so we will focus on the estimation of CIs of 2010 tabulations for PR census block groups.

We will consider CIs at the 0.9 and 0.95 confidence levels in particular. In the context of this numerical experiment, we will use *empirical coverage*, or *coverage*, of a set of CIs to refer to the proportion of the CIs in this set that contain the true population count. We will estimate the coverage of several CIs for several sets of queries in this section; note that a $1-\alpha$ confidence level CI should ideally have a coverage that is at least approximately $1-\alpha$. Note that one complication in this numerical example is that the point estimates (and therefor the CIs) are not entirely independent with one another. We do not expect the impact of this issue to be large, because the strength of the dependence between typical pairs of PR census block groups becomes weaker as the fanout values, *i.e.*, the number of child geounits of parent geounits, increases, and the fanout values in the spine are fairly high, especially at higher geolevels. For example, the county geolevel is composed of the 78 municipios of PR.

Since we know each true count is nonnegative, we can optionally increase the coverage of each CI, and reduce its width, by rounding the endpoints of the CI up to zero in the event they are negative, which we will refer to as a *nonnegative CI*. Tables 2 and 3 provide the empirical coverage and the average width, respectively, of the 0.9 and 0.95 CIs, both before and after rounding up to zero for three sets of queries. Specifically, we use “All Redistricting Tabulations” to refer to all persons Redistricting Data File tabulations published in 2020, *i.e.*, tables P1, P2, P3, P4, and P5, as described by U.S. Census Bureau (2023a), “Total Population” to refer to the total population, and “CENRACE” to refer to the collection of queries that provide the population counts in each of the 63 race combination categories of this attribute, as described in Section 1.2. For each of these three sets of queries, Tables 2 and 3 consider the CIs for each query in the set, evaluated in each 2010 vintage PR tabulation census block group.

Table 2 shows that coverage of the CIs before rounding up to zero, has a coverage that is fairly close to the confidence level of the CI considered, for each of the three sets of queries. The final two columns demonstrate that rounding these endpoints up to zero improves the coverage of these CIs, particularly for CIs of more granular queries that are more likely to have negative CI endpoints. Table 3 also shows that

TABLE 3. CI Averaged Widths: For each of the four CIs considered, and each of the three sets of queries considered, this table provides the CI width, averaged over all queries in the given query set and all block groups in PR.

	CI Type			
Query	0.90 CI	0.95 CI	0.90 CI, Nonnegative	0.95 CI, Nonnegative
All Redistricting Tabulations	100.8	120.1	54.84	65.74
Total Population	314.4	374.6	313.1	373.2
CENRACE	97.81	116.6	50.66	60.78

rounding up to zero can significantly narrow the widths of these same CIs of more granular queries. This later table also shows that the widths of these CIs are fairly wide on average. This is because census block groups were not on the optimized spine used internally in Redistricting Data file DAS executions. Note that one possibility to increase accuracy further would be to also leverage the information in the DHCP NMFs, after marginalizing over all cells other than the 2,016 detailed histogram cells in the Redistricting Data file schema, as described in Section 1.2. However, as described in Cumings-Menon et al. (2022), the DHC DAS executions used a different spine than the Redistricting Data file DAS executions, so if this is done, Algorithm 1 would need to be run separately on both sets of NMFs. Afterward, for each query and each geographic entity, the WLS-based point estimates and variances could be computed separately using each of the outputs of Algorithm 1. If these point estimates are denoted by $\tilde{\beta}_{H,\boldsymbol{q},1}, \tilde{\beta}_{H,\boldsymbol{q},2} \in \mathbb{R}$, then the inverse-variance weighted query estimate would be equal to

$$\check{\beta}_{H,\boldsymbol{q}} = \left(\tilde{\beta}_{H,\boldsymbol{q},1}/\mathrm{Var}(\tilde{\beta}_{H,\boldsymbol{q},1}) + \tilde{\beta}_{H,\boldsymbol{q},2}/\mathrm{Var}(\tilde{\beta}_{H,\boldsymbol{q},2})\right) / \left(1/\mathrm{Var}(\tilde{\beta}_{H,\boldsymbol{q},1}) + 1/\mathrm{Var}(\tilde{\beta}_{H,\boldsymbol{q},2})\right), \tag{27}$$

which has a variance given by

$$\mathrm{Var}(\check{\beta}_{H,\boldsymbol{q}}) = 1/\left(1/\mathrm{Var}(\tilde{\beta}_{H,\boldsymbol{q},1}) + 1/\mathrm{Var}(\tilde{\beta}_{H,\boldsymbol{q},2})\right), \tag{28}$$

and then the final confidence interval can be computed using the same approach as in Algorithm 3, *i.e.*, with endpoints given by $\check{\beta}_{H,\boldsymbol{q}} \pm \sqrt{\mathrm{Var}(\check{\beta}_{H,\boldsymbol{q}})}\Phi^{-1}(1-\alpha/2)$.

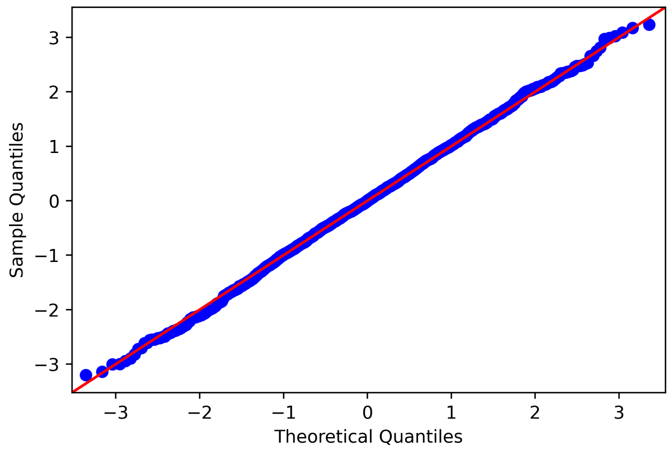
Note that the analyses above assess statistical validity of these CIs at only two confidence levels. To provide a way to assess statistical validity at all confidence levels, we also constructed the $Z-$scores of a set of estimates, which for each estimate $\tilde{\beta}_{H,\boldsymbol{q}}$ is defined as,

$$z_{H,\boldsymbol{q}} = \left(\tilde{\beta}_{H,\boldsymbol{q}} - \beta_{H,\boldsymbol{q}}\right) / \sqrt{\mathrm{Var}(\tilde{\beta}_{H,\boldsymbol{q}})}. \tag{29}$$

If $\boldsymbol{u}(g)$ were normally distributed for each $g \in \mathcal{G}$, then we would have $z_{H,\boldsymbol{q}} \sim N(0,1)$, so we can asses the error due to our normality modeling assumption by computing the $Z-$scores for a set of queries to see how closely these scores appear to follow a standard normal distribution. To do this, we computed the $Z-$scores for the total population query for each of the 2,543 census block groups in PR. Figure 2 provides a Q-Q plot based on these $Z-$scores. For each blue point, the horizontal axis provides the empirical quantiles of the 2,543 sample $Z-$scores and the vertical axis provides the corresponding theoretical quantile value based on a standard normal distribution. The red line provides the identity function. Since the blue points are very close to the identity function, this set of $Z-$scores appear to follow a Gaussian distribution quite closely. This plot also implies that, for any $\alpha \in (0,1)$, the $1-\alpha$ CIs for this set of queries would have empirical coverage close to $1-\alpha$.

To provide an idea of the computational requirements of our implementation of the proposed algorithms for the numerical experiment explored here, the total runtime of our implementation of Algorithm 1 was 8.6 hours, and estimating all of the CIs afterward took approximately 17.1 hours. Our implementation of Algorithm 1 stored its output in AWS S3 storage in a Apache Parquet format, and the size of this final output was approximately 795.6 GB for all geolevels in this compressed format, with the vast majority of this space (699.8 GB) being required to store the block geolevel matrices. The total number of blocks in PR in 2010 with either a housing unit or a GQ, *i.e.*, the universe of blocks we consider in this numerical example, was 51,937, which is about 1/113 of the total number of blocks in the spine of the 2020 US DAS

FIGURE 2. Q-Q Plot: The blue points in this plot can be viewed as points on a parametric curve, parameterized by the quantile $q \in (0,1)$. The horizontal axis provides the q quantile of the standard normal distribution and the vertical axis provides the sample quantile of the $Z-$score. The red line provides the identity function, $f(x) = x$.



optimized spine. Given this difference in the scale of the 2020 US and the 2010 PR DAS optimized spines and the computational requirements of this numerical experiment for the 2010 PR NMFs, a similar exercise may also be possible for the 2020 US production NMFs, albeit with a more significant computational cost. Also, the persons universe Redistricting Data File DAS schema likely has a number of detailed cells, *i.e.*, $n = 2,016$, near the highest value that would still allow for estimation in the 2020 US DAS optimized spines without marginalizing prior to estimation. As described previously, estimation using the 2020 US DHCP or DHCH NMFs would require marginalizing the NMFs to smaller(/coarser) histograms prior to estimation, since both of these two schemas have more than $n = 1.2$ million detailed cells for each geounit.

6. Comparison with Hay et al. (2010)

As described above, the approach proposed in this paper can be viewed as a generalization of the approach proposed by Hay et al. (2010), which we will describe in more detail in this section. To do so, we will assume we are in the setting described by Hay et al. (2010). In other words, we will suppose that, for every $g \in \mathcal{G}$, $S(g)$ is equal to the 1×1 matrix $[1]$, each vertex $g \in \mathcal{G}$ has $k \in \mathbb{N}$ children, and that the PLB allocated to each query, and therefore also the variance of each noisy measurement, is the same. Although it is not assumed by Hay et al. (2010), we will also suppose the variance of each query is equal to one specifically, since this variance does not appear in their algorithm and it simplifies the derivations below slightly.

In this setting Hay et al. (2010) show that the OLS estimator can be found using a two-step approach. For each $g \in \mathcal{G}$, before stating this theorem, they first define $z(g) \in \mathbb{R}$, which in our notation is given by

$$z(g) = \begin{cases} y(g) & g \in \text{Level}(\mathcal{G}, L) \\ \left(\frac{k^{L-l+1} - k^{L-l}}{k^{L-l+1} - 1}\right) y(g) + \left(\frac{k^{L-l} - 1}{k^{L-l+1} - 1}\right) \sum_{c \in \text{Children}(g)} z(c) & g \in \text{Level}(\mathcal{G}, l) \text{ and } l \neq L \end{cases}. \tag{30}$$

The relevant result from Hay et al. (2010) is stated below.

Theorem 2. *(Theorem 3 from Hay et al. (2010)) If, for every $g \in \mathcal{G}$, $S(g)$ is equal to the matrix $[1]$, each vertex $g \in \mathcal{G}$ has $k \in \mathbb{N}$ children, and the variance of each noisy measurement is one, then the WLS solution can be found using the recurrence relation*

$$\check{\boldsymbol{\beta}}(v) = \begin{cases} z(v) & v \in \mathit{Level}(\mathcal{G}, 0) \\ z(v) + \frac{1}{k}\left(\check{\boldsymbol{\beta}}(g) - \sum_{c \in \mathit{Children}(v)} z(c)\right) & v \notin \mathit{Level}(\mathcal{G}, 0) \text{ and } v \in \mathit{Children}(g) \end{cases}. \tag{31}$$

Below we show that our estimator is a special case of the estimator described by Hay et al. (2010).

Theorem 3. *If, for every $g \in \mathcal{G}$, $S(g)$ is equal to the matrix* [1], *each vertex $g \in \mathcal{G}$ has $k \in \mathbb{N}$ children, and the variance of each noisy measurement is one, then $\breve{\boldsymbol{\beta}}(g) = \tilde{\boldsymbol{\beta}}(g)$.*

Proof. First we will show that $z(g) = \hat{\boldsymbol{\beta}}(g|g-)$ under the assumptions of the theorem. To do so, for $g \in \text{Level}(\mathcal{G}, l)$, we will begin by deriving a convenient functional form for $v(l) := \text{Var}(\hat{\boldsymbol{\beta}}(g|g-))$ under the conditions of the theorem. Using the definition from Section 2.1 and the assumptions of this theorem, for any $g \in \text{Level}(\mathcal{G}, L)$, we have $v(L) = \text{Var}(\boldsymbol{y}(g)) = 1$. Using notation introduced in Section 2.2, we have $\text{Var}(\hat{\boldsymbol{\beta}}(g|\text{Children}(g)-)) = kv(l)$ for any $g \in \text{Level}(\mathcal{G}, l-1)$, so the variance of the inverse variance weighted mean implies

$$v(l-1) = \frac{1}{(1 + 1/(kv(l)))} = 1 - \frac{1}{1 + kv(l)}$$

After solving this recursive equation with the boundary condition that $v(L) = 1$, we have,

$$v(l) = \frac{(k-1)k^L}{k^{L+1} - k^l}$$

Note that, for $g \in \text{Level}(\mathcal{G}, l)$, equation (13) implies

$$\hat{\boldsymbol{\beta}}(g|g-) = v(l) \left(\boldsymbol{y}(g) + \frac{1}{kv(l+1)} \sum_{c \in \text{Children}(g)} \hat{\boldsymbol{\beta}}(c|c-) \right).$$

Note that, for any $g \in \text{Level}(\mathcal{G}, L)$, we clearly have $z(g) = \hat{\boldsymbol{\beta}}(g|g-)$. Also, in the case of vertices in levels above level L, the definition of $z(g)$ and $\hat{\boldsymbol{\beta}}(g|g-)$ follows a similar form, in the sense that, for any $g \in \text{Level}(\mathcal{G}, l)$ with $l < L$, both estimators are weighted means of $\boldsymbol{y}(g)$ and the sum over the estimates of the children of g. In the case of both $z(g)$ and $\hat{\boldsymbol{\beta}}(g|g-)$, the weights of these two terms sum to one, so we will establish that $z(g) = \hat{\boldsymbol{\beta}}(g|g-)$ for all $g \in \text{Level}(\mathcal{G}, l)$ with $l < L$ by showing that the ratio of these weights are the same for both of these estimators. In the case of $\hat{\boldsymbol{\beta}}(g|g-)$, this ratio of these weights is given by $kv(l+1)$, and in the case of $z(g)$ this ratio is given by

$$\frac{k^{L-l+1} - k^{L-l}}{k^{L-l} - 1}.$$

Thus, to establish that $z(g) = \hat{\boldsymbol{\beta}}(g|g-)$ for all $g \in \text{Level}(\mathcal{G}, l)$ with $l < L$, we need to show

$$\begin{aligned} \frac{k^{L-l+1} - k^{L-l}}{k^{L-l} - 1} = kv(l+1) &\iff \frac{k^{L-l+1} - k^{L-l}}{k^{L-l} - 1} = \frac{(k-1)k^{L+1}}{k^{L+1} - k^{l+1}} \\ &\iff \frac{k^L(k-1)}{k^L - k^l} = \frac{k^L(k-1)}{k^L - k^l}, \end{aligned}$$

which implies $z(g) = \hat{\boldsymbol{\beta}}(g|g-)$ for all $g \in \mathcal{G}$.

The final result follows from the fact that, after replacing $z(g)$ with $\hat{\boldsymbol{\beta}}(g|g-)$ in the recurrence relation in (31), this recursive formula is equal to the top-down pass described in Section 2. □

7. Conclusion

This paper describes a two-pass estimation approach for hierarchical data that is capable of providing WLS-based estimates and confidence intervals. We also provide a numerical exercise to demonstrate feasibility of the approach for our motivating use case. As described in the previous section, this two-pass approach can be viewed as a generalization of the approach described by Hay et al. (2010). This section concludes by describing some advantages of our proposed approach relative to another approach in the context of our primary motivating use case, *i.e.*, estimating tabulations and their corresponding confidence intervals using census NMFs, as described in Section 1.3.

In the context of our motivating use case, an alternative to using the WLS estimator described above is to simply derive point estimates from the NMFs directly. For example, one can estimate the total population of a geographic entity that corresponds to a geounit using the total population primitive DP answer of the geounit directly. An alternative estimator for this same query answer can be derived by summing over the total population noisy measurements of the children of this geounit. More generally, in this section, we will refer to all estimates that can be defined by adding and/or subtracting a subset of the scalars in the

NMFs as *naïve estimates* of a tabulation. For use cases that do not require high precision, and only require tabulation estimates for geographic entities that are geounits on the spine used to generate the NMFs, these naïve estimates may be adequate. However, unlike the naïve estimates, the full-information WLS estimator provides a unique estimate, with the added benefit of also providing the lowest possible variance out of all estimators that are unbiased linear functions of the noisy measurements.

These advantages of the two-pass WLS estimator are most obvious when estimating queries for a geographic entity that is far from the spine used to generate the noisy measurements, meaning many geounits would need to be added and subtracted from one another to derive the geographic entity. For example, when deriving a total population estimate for Navajo Nation, it would not be obvious how to choose a relatively small set of noisy measurements that provides an unbiased naïve estimate with a sufficiently low variance. Examples of approaches that are capable of constructing these subsets of the NMFs would include considering adding the NMFs at the block geolevel NMFs, considering the NMFs for the least-granular set of geounits that are entirely within a geounit on the spine, *i.e.*, the approach described in the first comment of Algorithm 3, or considering the NMFs of the smallest set of geounits that can be used to derive the off-spine geographic entity by adding and/or subtracting the geounits in this set.[4] As a result, data users confronted with this issue when using one of the naïve estimators would often choose different sets of noisy measurements and thus potentially arrive at dissimilar estimates. In contrast, the WLS estimator proposed here allows users to construct the unique minimum-variance point estimates, and their corresponding CIs using a normality approximation, even for geographic entities that are far from the geographic spine. We expect data users will view the combination of accuracy, approximate normality, and uniqueness provided by the WLS-based tabulation estimators to be advantageous.

It is also worth pointing out that the Census Bureau is developing an alternative method to compute confidence intervals, based on simulating the DAS using the 2020 production DAS execution outputs, *i.e.*, the microdata detailed files (MDFs), as the input of each simulation iteration. This simulation-based approach appears to perform well, with CIs that are typically more narrow than the WLS-based CIs of the approach described in this paper, since this simulation-based approach leverages the increased accuracy due to the inequality constraints used within DAS. The main advantage of the WLS-based approach described here over this simulation-based approach is that we avoid the assumption that DAS runs using the private census data as input have similar error distributions as DAS runs that use the 2020 MDFs as input. Since both approaches have advantages, which of these two approaches is best for a given application will typically be dependent on the specific goals of the use case.

References

Abowd, J. M., Ashmead, R., Cumings-Menon, R., Garfinkel, S., Heineck, M., Heiss, C., Johns, R., Kifer, D., Leclerc, P., Machanavajjhala, A., et al. (2022). The 2020 Census disclosure avoidance system topdown algorithm. *Harvard Data Science Review*, (Special Issue 2).

Bun, M. and Steinke, T. (2016). Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *Theory of Cryptography Conference*, pages 635–658. Springer.

Canonne, C. L., Kamath, G., and Steinke, T. (2020). The discrete Gaussian for differential privacy. *Advances in Neural Information Processing Systems*, 33:15676–15688.

Cumings-Menon, R., Abowd, J. M., Ashmead, R., Kifer, D., Leclerc, P., Ocker, J., Ratcliffe, M., and Zhuravlev, P. (2022). Geographic spines in the 2020 census disclosure avoidance system. *arXiv preprint arXiv:2203.16654*.

Cumings-Menon, R., Ashmead, R., Kifer, D., Leclerc, P., Spence, M., Zhuravlev, P., and Abowd, J. M. (2023). Disclosure avoidance for the 2020 Census Demographic and Housing Characteristics File. *arXiv preprint arXiv:2312.10863*.

Davidson, R., MacKinnon, J. G., et al. (2004). *Econometric theory and methods*, volume 5. Oxford University Press New York.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer.

Hay, M., Rastogi, V., Miklau, G., and Suciu, D. (2010). Boosting the accuracy of differentially private histograms through consistency. *Proceedings of the VLDB Endowment*, 3(1).

[4]This latter set of geounits is described in more detail in Cumings-Menon et al. (2022).

U.S. Census Bureau (2023a). Decennial Census P.L. 94-171 Redistricting Data.
U.S. Census Bureau (2023b). Developing the DAS: Demonstration Data and Progress Metrics.
Willsky, A. S. (2002). Multiresolution markov models for signal and image processing. *Proceedings of the IEEE*, 90(8):1396–1458.

Appendix A. Result on Optimality of Top-Down Pass Projection

This section provides the proof of Theorem 1. Before doing this, we first prove three lemmas that provide properties of the state variables for the parent vertex $g \in \mathcal{G}$. To do this, some additional notation will also be helpful. Specifically, for this parent vertex $g \in \mathcal{G}$, let $\kappa := \text{Card}(\text{Children}(g))$ and $E \in \mathbb{R}^{n\kappa \times n}$ be defined as

$$E := \begin{bmatrix} I_n \\ \vdots \\ I_n \end{bmatrix}.$$

Also, let $\tilde{\boldsymbol{\beta}}_g$ be defined as $\text{stack}(\{\tilde{\boldsymbol{\beta}}(c)\}_{c\in\text{Children}(g)})$, where $\tilde{\boldsymbol{\beta}}(c)$ is defined in (18), and let $\hat{\boldsymbol{\beta}}_g$ be defined as $\text{stack}(\{\hat{\boldsymbol{\beta}}(c|c-)\}_{c\in\text{Children}(g)})$, where $\hat{\boldsymbol{\beta}}(c|c-)$ is defined in (14).

The first lemma in this section provides the functional form for the matrix $\widetilde{Q}_g$, as introduced in Section 2.4.

Lemma 2. *In the top-down pass, for each parent vertex $g \in \mathcal{G}$, $\widetilde{Q}_g$ satisfies*

$$\widetilde{Q}_g = Var(\tilde{\boldsymbol{\beta}}(g)) \left(\sum_{c\in Children(g)} Var(\hat{\boldsymbol{\beta}}(c|c-)) \right)^{-1}. \tag{32}$$

Proof. The initialization step of the top-down pass implies that for the root vertex g, *i.e.*, equations (15)-(16), we have $\tilde{\boldsymbol{\beta}}(g) = \hat{\boldsymbol{\beta}}(g|g-)$, and

$$\text{Var}(\tilde{\boldsymbol{\beta}}(g)) = \text{Var}(\hat{\boldsymbol{\beta}}(g|g-)) = \left(\text{Var}(\hat{\boldsymbol{\beta}}(g|g))^{-1} + \text{Var}(\hat{\boldsymbol{\beta}}(g|\text{Children}(g)-))^{-1}\right)^{-1}.$$

Thus, equation (14) implies

$$\widetilde{Q}_g = \text{Var}(\hat{\boldsymbol{\beta}}(g|g-))\text{Var}(\hat{\boldsymbol{\beta}}(g|\text{Children}(g)-))^{-1} = \text{Var}(\tilde{\boldsymbol{\beta}}(g)) \left(\sum_{c\in\text{Children}(g)} \text{Var}(\hat{\boldsymbol{\beta}}(c|c-)) \right)^{-1},$$

so the property in the statement of this result, *i.e.*, equality (32), holds for the root vertex.

Now we will suppose equality in (32) holds for an arbitrary parent vertex $g \in \mathcal{G}$ and we show that this implies it also holds for the children of g. To do so, first recall that, for any vertex $p \in \text{Children}(g)$, equation (18) implies

$$\begin{aligned}
\tilde{\boldsymbol{\beta}}(p) &= \hat{\boldsymbol{\beta}}(p|p-) + A(p) \left(\tilde{\boldsymbol{\beta}}(g) - \sum_{p'\in\text{Children}(g)} \hat{\boldsymbol{\beta}}(p'|p'-) \right) \\
&= \hat{\boldsymbol{\beta}}(p|p-) + A(p) \left(\boldsymbol{\tau}_g + \widetilde{Q}_g \left(\sum_{p'\in\text{Children}(g)} \hat{\boldsymbol{\beta}}(p'|p'-) \right) - \sum_{p'\in\text{Children}(g)} \hat{\boldsymbol{\beta}}(p'|p'-) \right).
\end{aligned}$$

By (14), we can substitute each instance of $\hat{\boldsymbol{\beta}}(p|p-)$ with

$$\check{\boldsymbol{\tau}}_p + \text{Var}(\hat{\boldsymbol{\beta}}(p|p-)) \left(\sum_{c\in\text{Children}(p)} \text{Var}(\hat{\boldsymbol{\beta}}(c|c-)) \right)^{-1} \left(\sum_{c\in\text{Children}(p)} \hat{\boldsymbol{\beta}}(c|c-) \right),$$

where $\check{\boldsymbol{\tau}}_p$ is a random vector that is independent of $\hat{\boldsymbol{\beta}}(c|c-)$, for each $c \in \text{Children}(p)$. Thus, for $\boldsymbol{\tau}_p$ defined as another random vector that is independent of $\hat{\boldsymbol{\beta}}(c|c-)$, for each $c \in \text{Children}(p)$, we have

$$\begin{aligned}\tilde{\boldsymbol{\beta}}(p) =& \boldsymbol{\tau}_p + \text{Var}(\hat{\boldsymbol{\beta}}(p|p-)) \left(\sum_{c\in\text{Children}(p)} \text{Var}(\hat{\boldsymbol{\beta}}(c|c-)) \right)^{-1} \left(\sum_{c\in\text{Children}(p)} \hat{\boldsymbol{\beta}}(c|c-) \right) \\ &+ A(p) \left(\widetilde{Q}_g - I \right) \text{Var}(\hat{\boldsymbol{\beta}}(p|p-)) \left(\sum_{c\in\text{Children}(p)} \text{Var}(\hat{\boldsymbol{\beta}}(c|c-)) \right)^{-1} \left(\sum_{c\in\text{Children}(p)} \hat{\boldsymbol{\beta}}(c|c-) \right) \\ =& \boldsymbol{\tau}_p + \left(I + A(p) \left(\widetilde{Q}_g - I \right) \right) \text{Var}(\hat{\boldsymbol{\beta}}(p|p-)) \left(\sum_{c\in\text{Children}(p)} \text{Var}(\hat{\boldsymbol{\beta}}(c|c-)) \right)^{-1} \left(\sum_{c\in\text{Children}(p)} \hat{\boldsymbol{\beta}}(c|c-) \right).\end{aligned}$$

Since we supposed the equality (32) holds for vertex $g \in \mathcal{G}$, this implies

$$\widetilde{Q}_p = \left(I + A(p) \left(\text{Var}(\tilde{\boldsymbol{\beta}}(g)) \left(\sum_{p\in\text{Children}(g)} \text{Var}(\hat{\boldsymbol{\beta}}(p|p-)) \right)^{-1} - I \right) \right) \text{Var}(\hat{\boldsymbol{\beta}}(p|p-)) \left(\sum_{c\in\text{Children}(p)} \text{Var}(\hat{\boldsymbol{\beta}}(c|c-)) \right)^{-1},$$

so since this result requires that

$$\widetilde{Q}_p = \text{Var}(\tilde{\boldsymbol{\beta}}(p)) \left(\sum_{c\in\text{Children}(p)} \text{Var}(\hat{\boldsymbol{\beta}}(c|c-)) \right)^{-1},$$

we need to show,

$$\text{Var}(\tilde{\boldsymbol{\beta}}(p)) = \left(I + A(p) \left(\text{Var}(\tilde{\boldsymbol{\beta}}(g)) \left(\sum_{p\in\text{Children}(g)} \text{Var}(\hat{\boldsymbol{\beta}}(p|p-)) \right)^{-1} - I \right) \right) \text{Var}(\hat{\boldsymbol{\beta}}(p|p-)).$$

By (22), this equality holds if and only if

$$\begin{aligned}&\text{Var}(\hat{\boldsymbol{\beta}}(p|p-)) - A(p)\text{Var}(\hat{\boldsymbol{\beta}}(p|p-)) + A(p)\text{Var}(\tilde{\boldsymbol{\beta}}(g))A(p)^\top \\ &= \left(I + A(p) \left(\text{Var}(\tilde{\boldsymbol{\beta}}(g)) \left(\sum_{p\in\text{Children}(g)} \text{Var}(\hat{\boldsymbol{\beta}}(p|p-)) \right)^{-1} - I \right) \right) \text{Var}(\hat{\boldsymbol{\beta}}(p|p-)) \\ &\iff A(p)\text{Var}(\tilde{\boldsymbol{\beta}}(g))A(p)^\top = A(p)\text{Var}(\tilde{\boldsymbol{\beta}}(g)) \left(\sum_{p\in\text{Children}(g)} \text{Var}(\hat{\boldsymbol{\beta}}(p|p-)) \right)^{-1} \text{Var}(\hat{\boldsymbol{\beta}}(p|p-)) \\ &\iff A(p)^\top = \left(\sum_{p\in\text{Children}(g)} \text{Var}(\hat{\boldsymbol{\beta}}(p|p-)) \right)^{-1} \text{Var}(\hat{\boldsymbol{\beta}}(p|p-)),\end{aligned}$$

which is satisfied by the definition of $A(p)$, *i.e.*, (19). □

Lemma 3 provides an eigenvalue bound that will be useful in the proof of the main result of this section.

Lemma 3. *For any parent vertex $g \in \mathcal{G}$, we have $\sum_{c\in \text{Children}(g)} \text{Var}(\hat{\boldsymbol{\beta}}(c|c-)) \geq \text{Var}(\tilde{\boldsymbol{\beta}}(g))$.*

Proof. For arbitrary parent vertex $g \in \mathcal{G}$, let $s_g := \sum_{c\in\text{Children}(g)} \text{Var}(\hat{\boldsymbol{\beta}}(c|c-))$.

First, we prove the result for the root vertex $g \in \mathcal{G}$. In this case, (16) and (13) imply

$$\text{Var}(\tilde{\boldsymbol{\beta}}(g)) = \text{Var}(\hat{\boldsymbol{\beta}}(g|g-)) = \left(\text{Var}(\hat{\boldsymbol{\beta}}(g|g))^{-1} + s_g^{-1} \right)^{-1}$$

Thus, the result in this case follows from

$$s_g \geq \mathrm{Var}(\tilde{\boldsymbol{\beta}}(g)) \iff s_g \geq \left(\mathrm{Var}(\hat{\boldsymbol{\beta}}(g|g))^{-1} + s_g^{-1}\right)^{-1}$$
$$\iff \left(\mathrm{Var}(\hat{\boldsymbol{\beta}}(g|g))^{-1} + s_g^{-1}\right) s_g \geq I \iff \mathrm{Var}(\hat{\boldsymbol{\beta}}(g|g))^{-1} s_g \geq 0,$$

which is satisfied because both $\mathrm{Var}(\hat{\boldsymbol{\beta}}(g|g))^{-1}$ and s_g are positive definite by our assumptions on $\{S(g)\}_{g\in\mathcal{G}}$ and the variance of the mean zero errors $\{\boldsymbol{u}(g)\}_{g\in\mathcal{G}}$.

Second, we prove the result for the arbitrary parent vertex $c \in \mathcal{G}$, which has a parent vertex that we denote by $g \in \mathcal{G}$. Recall equation (22) is

$$\begin{aligned}
&\mathrm{Var}(\tilde{\boldsymbol{\beta}}(c)) = \mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c-)) - A(c)\mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c-)) + A(c)\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))A(c)^\top \\
&= \mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c-)) - \mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c-))s_g^{-1}\mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c-)) + \mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c-))s_g^{-1}\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))s_g^{-1}\mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c-)) \\
&= \left(\mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c))^{-1} + s_c^{-1}\right)^{-1} - \left(\mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c))^{-1} + s_c^{-1}\right)^{-1} s_g^{-1} \left(\mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c))^{-1} + s_c^{-1}\right)^{-1} \\
&+ \left(\mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c))^{-1} + s_c^{-1}\right)^{-1} s_g^{-1}\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))s_g^{-1} \left(\mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c))^{-1} + s_c^{-1}\right)^{-1},
\end{aligned}$$

where the final equality follows from (13). Thus, $s_c \geq \mathrm{Var}(\tilde{\boldsymbol{\beta}}(c))$ is satisfied if and only if

$$\begin{aligned}
&s_c \geq \left(\mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c))^{-1} + s_c^{-1}\right)^{-1} - \left(\mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c))^{-1} + s_c^{-1}\right)^{-1} s_g^{-1} \left(\mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c))^{-1} + s_c^{-1}\right)^{-1} \\
&+ \left(\mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c))^{-1} + s_c^{-1}\right)^{-1} s_g^{-1}\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))s_g^{-1} \left(\mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c))^{-1} + s_c^{-1}\right)^{-1} \\
&\iff \left(\mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c))^{-1} + s_c^{-1}\right) s_c \left(\mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c))^{-1} + s_c^{-1}\right) \geq \mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c))^{-1} \\
&+ s_c^{-1} - s_g^{-1} + s_g^{-1}\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))s_g^{-1}
\end{aligned}$$

$$\iff \mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c))^{-1} s_c \mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c))^{-1} + \mathrm{Var}(\hat{\boldsymbol{\beta}}(c|c))^{-1} + s_g^{-1} - s_g^{-1}\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))s_g^{-1} \geq 0. \tag{33}$$

A sufficient condition for (33) is

$$s_g^{-1} - s_g^{-1}\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))s_g^{-1} >= 0 \iff s_g >= \mathrm{Var}(\tilde{\boldsymbol{\beta}}(g)),$$

so, since this latter condition is satisfied by the root vertex, inequality (33), and thus also the final result, follows from induction. □

Theorem 1 shows that the projection used in the top-down pass, *i.e.*, the solution to the optimization problem in (17), provides the minimum-variance estimator in the family of estimators that are unbiased, linear in the input random variables, and that satisfy the parent-child consistency constraints. The next lemma provides the functional form of all such estimators that are in this family.

Lemma 4. *Suppose that the following conditions hold.*

(a) $\check{\boldsymbol{\beta}}_g := \mathit{stack}(\{\check{\boldsymbol{\beta}}(c)\}_{c\in \mathit{Children}(g)}) \in \mathbb{R}^{n\kappa}$ *is any unbiased estimator of* $\mathit{stack}(\{\boldsymbol{\beta}(c)\}_{c\in \mathit{Children}(g)})$

(b) $\check{\boldsymbol{\beta}}_g$ *is linear in* $\tilde{\boldsymbol{\beta}}(g)$ *and* $\hat{\boldsymbol{\beta}}_g := \mathit{stack}(\{\hat{\boldsymbol{\beta}}(c|c-)\}_{c\in \mathit{Children}(g)})$

(c) $E^\top\check{\boldsymbol{\beta}}_g = \sum_{c\in \mathit{Children}(g)} \check{\boldsymbol{\beta}}(c) = \tilde{\boldsymbol{\beta}}(g)$

Then, there exists $R \in \mathbb{R}^{n\kappa\times n}$ *such that*

$$\check{\boldsymbol{\beta}}_g = (I - RE^\top)\hat{\boldsymbol{\beta}}_g + R\tilde{\boldsymbol{\beta}}(g)$$

and also $E^\top R = I$.

Proof. By the linearity property of $\check{\boldsymbol{\beta}}_g$ that is assumed in the statement of the result, *i.e.*, assumption (b), there exists $Q \in \mathbb{R}^{n\kappa\times n\kappa}$ and $R \in \mathbb{R}^{n\kappa\times n}$ such that

$$\check{\boldsymbol{\beta}}_g = (I - Q)\hat{\boldsymbol{\beta}}_g + R\tilde{\boldsymbol{\beta}}(g). \tag{34}$$

After left multiplying both sides of this equality by $E^\top$, assumption (c) implies

$$\tilde{\boldsymbol{\beta}}(g) = (E^\top - E^\top Q)\hat{\boldsymbol{\beta}}_g + E^\top R\tilde{\boldsymbol{\beta}}(g).$$
$$\iff E^\top = E^\top Q \tag{35}$$
$$\text{and } E^\top R = I \tag{36}$$

After taking the expectation of both sides of (34), assumption (a) implies

$$E(\check{\boldsymbol{\beta}}_g) = \text{stack}(\{\boldsymbol{\beta}(c)\}_{c\in\text{Children}(g)}) = \left((I-Q) + Re^\top\right)\text{stack}(\{\boldsymbol{\beta}(c)\}_{c\in\text{Children}(g)})$$
$$\iff Q = Re^\top \tag{37}$$

□

The three lemmas above are used in the following proof of Theorem 1, which shows that the projection used in the top-down pass provides the best unbiased, linear estimate that satisfies the parent-child consistency constraints.

Theorem 1. *Suppose that the following conditions hold.*

(a) $\check{\boldsymbol{\beta}}_g := \textit{stack}(\{\check{\boldsymbol{\beta}}(c)\}_{c\in \textit{Children}(g)}) \in \mathbb{R}^{n\kappa}$ *is any unbiased estimator of* $\textit{stack}(\{\boldsymbol{\beta}(c)\}_{c\in \textit{Children}(g)})$
(b) $\check{\boldsymbol{\beta}}_g$ *is linear in* $\tilde{\boldsymbol{\beta}}(g)$ *and* $\hat{\boldsymbol{\beta}}_g := \textit{stack}(\{\hat{\boldsymbol{\beta}}(c|c-)\}_{c\in \textit{Children}(g)})$
(c) $\sum_{c\in \textit{Children}(g)} \check{\boldsymbol{\beta}}(c) = \tilde{\boldsymbol{\beta}}(g)$

If $\tilde{\boldsymbol{\beta}}(c)$ *is defined as in (18) and* $\tilde{\boldsymbol{\beta}}_g$ *is defined as* $\textit{stack}(\{\tilde{\boldsymbol{\beta}}(c)\}_{c\in \textit{Children}(g)})$*, then*

$$\textit{Var}(\tilde{\boldsymbol{\beta}}) \leq \textit{Var}(\check{\boldsymbol{\beta}}_g). \tag{38}$$

Proof. Equation (18) implies that $\tilde{\boldsymbol{\beta}}_g = (I - RE^\top)\hat{\boldsymbol{\beta}}_g + R\tilde{\boldsymbol{\beta}}(g)$, where $R := \text{stack}(\{A(c)\}_{c\in\text{Children}(g)})$. Since $E^\top R = I$ by definition of R, Lemma 4 implies that there exists $D \in \mathbb{R}^{n\kappa\times n}$ such that $E^\top D = 0$ and

$$\check{\boldsymbol{\beta}}_g = (I - (R+D)E^\top)\hat{\boldsymbol{\beta}}_g + (R+D)\tilde{\boldsymbol{\beta}}(g),$$

which implies

$$\begin{aligned}\text{Var}(\check{\boldsymbol{\beta}}_g) &= (I-(R+D)E^\top)\text{Var}(\hat{\boldsymbol{\beta}}_g)(I - E(R+D)^\top) + (R+D)\text{Var}(\tilde{\boldsymbol{\beta}}(g))(R+D)^\top \\ &+ (I-(R+D)E^\top)\text{Cov}(\hat{\boldsymbol{\beta}}_g, \tilde{\boldsymbol{\beta}}(g))(R+D)^\top + (R+D)\text{Cov}(\tilde{\boldsymbol{\beta}}(g), \hat{\boldsymbol{\beta}}_g)(I - E(R+D)^\top).\end{aligned} \tag{39}$$

After expanding, substituting $\text{Cov}(\hat{\boldsymbol{\beta}}_g, \tilde{\boldsymbol{\beta}}(g))$ for $\text{Cov}(\tilde{\boldsymbol{\beta}}(g), \hat{\boldsymbol{\beta}}_g)^\top$, and then using Lemma 2 to substitute $\text{Cov}(\tilde{\boldsymbol{\beta}}(g), \hat{\boldsymbol{\beta}}_g)$ for $E^\top\text{Var}(\hat{\boldsymbol{\beta}}_g)E\widetilde{Q}_g^\top$, (39) becomes

$$\begin{aligned}\text{Var}(\check{\boldsymbol{\beta}}_g) &= \text{Var}(\tilde{\boldsymbol{\beta}}_g) - (I-(R+D)E^\top)\text{Var}(\hat{\boldsymbol{\beta}}_g)ED^\top - DE^\top\text{Var}(\hat{\boldsymbol{\beta}}_g)(I - ER^\top) \\ &+ (R+D)\text{Var}(\tilde{\boldsymbol{\beta}}(g))D^\top + D\text{Var}(\tilde{\boldsymbol{\beta}}(g))R^\top \\ &+ (I-(R+D)E^\top)\text{Var}(\hat{\boldsymbol{\beta}}_g)E\widetilde{Q}_g^\top D^\top - DE^\top\text{Var}(\hat{\boldsymbol{\beta}}_g)E\widetilde{Q}_g^\top R^\top \\ &- (R+D)\widetilde{Q}_g E^\top\text{Var}(\hat{\boldsymbol{\beta}}_g)ED^\top + D\widetilde{Q}_g E^\top\text{Var}(\hat{\boldsymbol{\beta}}_g)(I - ER^\top)\end{aligned} \tag{40}$$

Since (21) implies $(I - RE^\top)\text{Var}(\hat{\boldsymbol{\beta}}_g)E = 0$, (40) can be written as

$$\begin{aligned}\text{Var}(\check{\boldsymbol{\beta}}_g) &= \text{Var}(\tilde{\boldsymbol{\beta}}_g) + DE^\top\text{Var}(\hat{\boldsymbol{\beta}}_g)ED^\top + (R+D)\text{Var}(\tilde{\boldsymbol{\beta}}(g))D^\top + D\text{Var}(\tilde{\boldsymbol{\beta}}(g))R^\top \\ &- DE^\top\text{Var}(\hat{\boldsymbol{\beta}}_g)E\widetilde{Q}_g^\top(R+D)^\top - (R+D)\widetilde{Q}_g E^\top\text{Var}(\hat{\boldsymbol{\beta}}_g)ED^\top\end{aligned} \tag{41}$$

Since Lemma 2 implies $\widetilde{Q}_g = \mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))\left(E^\top \mathrm{Var}(\hat{\boldsymbol{\beta}}_g)E\right)^{-1}$, (41) can be written as

$$\begin{aligned}
\mathrm{Var}(\check{\boldsymbol{\beta}}_g) &= \mathrm{Var}(\tilde{\boldsymbol{\beta}}_g) + DE^\top \mathrm{Var}(\hat{\boldsymbol{\beta}}_g)ED^\top + (R+D)\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))D^\top + D\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))R^\top \\
&\quad - DE^\top \mathrm{Var}(\hat{\boldsymbol{\beta}}_g)E\left(E^\top \mathrm{Var}(\hat{\boldsymbol{\beta}}_g)E\right)^{-1}\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))(R+D)^\top \\
&\quad - (R+D)\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))\left(E^\top \mathrm{Var}(\hat{\boldsymbol{\beta}}_g)E\right)^{-1}E^\top \mathrm{Var}(\hat{\boldsymbol{\beta}}_g)ED^\top \\
\iff \mathrm{Var}(\check{\boldsymbol{\beta}}_g) &= \mathrm{Var}(\tilde{\boldsymbol{\beta}}_g) + DE^\top \mathrm{Var}(\hat{\boldsymbol{\beta}}_g)ED^\top + (R+D)\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))D^\top + D\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))R^\top \\
&\quad - D\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))(R+D)^\top - (R+D)\mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))D^\top \\
\iff \mathrm{Var}(\check{\boldsymbol{\beta}}_g) &= \mathrm{Var}(\tilde{\boldsymbol{\beta}}_g) + D\left(E^\top \mathrm{Var}(\hat{\boldsymbol{\beta}}_g)E - \mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))\right)D^\top \qquad (42)
\end{aligned}$$

Since Lemma 3 implies $E^\top \mathrm{Var}(\hat{\boldsymbol{\beta}}_g)E - \mathrm{Var}(\tilde{\boldsymbol{\beta}}(g))$ is positive semi-definite, (42) implies $\mathrm{Var}(\check{\boldsymbol{\beta}}_g) \geq \mathrm{Var}(\tilde{\boldsymbol{\beta}}_g)$. $\square$