SUPPLEMENTARY MATERIALS: Confidence intervals for functionals in constrained inverse problems via data-adaptive sampling-based calibration

Michael Stanley*, Pau Batlle , Pratik Patil†, Houman Owhadi‡, and Mikael Kuusela§

This document serves as a supplement to the paper "Confidence intervals for functionals in constrained inverse problems via data-adaptive sampling-based calibration". Below we provide an organization for the supplement, followed by a summary of the main notation used in both the paper and the appendix.

Organization. A roadmap for the supplement is provided in Table SM1.

Table SM1: Roadmap of the supplement.

Appendix	Description
Section SM1	Proofs in Section 3 (Proofs of Lemma 3.1 and Corollary 3.2)
Section SM2	Proofs in Section 4 (Proof of Theorem 4.1)
Section SM3	Additional details and illustrations in Subsection 5.1 (VGS and Polytope samplers)
Section SM4	Additional details in Subsection 5.2 (quantile regression)
Section SM5	Additional details and illustrations in Section 6 (importance-like sampler)

Notation. An overview of the main notation used in this paper is provided in Table SM2.

SM1. Proofs in Section 3.

SM1.1. Proof of Lemma 3.1. We reproduce the original argument in [SM2], originally stated in terms of p-values, translated into our quantile setting. Fix any $\boldsymbol{x}^* \in \mathcal{X}$ and consider the sets:

- $A_1 = \{\boldsymbol{y} : B_\eta(\boldsymbol{y}) \ni \boldsymbol{x}^*\}$
- $A_2 = \{\boldsymbol{y} : \lambda(\mu^*, \boldsymbol{y}) \leq Q_{\boldsymbol{x}}(1-\gamma)\}$
- $A_3 = \{\boldsymbol{y} : \lambda(\mu^*, \boldsymbol{y}) \leq \bar{q}_{\gamma,\eta}(\mu^*)\} = \{\boldsymbol{y} : \lambda(\mu^*, \boldsymbol{y}) \leq \sup_{\boldsymbol{x}\in\mathcal{B}_\eta\cap\Phi_{\mu^*}} Q_{\boldsymbol{x}}(1-\gamma)\}$

We now have that $\mathbb{P}(\boldsymbol{y} \in A_1) = 1-\eta, \mathbb{P}(\boldsymbol{y} \in A_2) = 1-\gamma$, and that $(A_1 \cap A_2) \subset (A_1 \cap A_3)$. Therefore,

$$\mathbb{P}(\boldsymbol{y} \notin A_3) = \mathbb{P}(\boldsymbol{y} \notin A_3, \boldsymbol{y} \in A_1) + \mathbb{P}(\boldsymbol{y} \notin A_3, \boldsymbol{y} \notin A_1) \tag{SM1.1}$$

$$\leq \mathbb{P}(\boldsymbol{y} \notin A_2, \boldsymbol{y} \in A_1) + \mathbb{P}(\boldsymbol{y} \notin A_1) \tag{SM1.2}$$

*Analytical Mechanics Associates, USA
†Department of Statistics and Data Sciences, University of Texas at Austin, USA
‡Department of Computing and Mathematical Sciences, California Institute of Technology, USA
§Department of Statistics and Data Science, Carnegie Mellon University, USA

Table SM2: Summary of main notation used in the paper and the appendix.

Notation	Description
Non-bold lower or upper case	Denotes scalars (e.g., α, μ, Q).
Bold lower case	Denotes vectors (e.g., $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{h}$).
Bold upper case	Denotes matrices (e.g., $\boldsymbol{K}$, $\boldsymbol{I}$).
Calligraphic font	Denotes sets (e.g., $\mathcal{X}$, $\mathcal{C}$, $\mathcal{D}$).
$\mathbb{R}$	Set of real numbers.
$\mathbb{R}_+$	Set of non-negative real numbers.
$[n]$	Set $\{1, \ldots, n\}$ for a positive integer n.
$\|\boldsymbol{u}\|_2$	The ℓ_2 norm of vector $\boldsymbol{u}$.
$\|f\|_{L_2}$	The L_2 norm of function f.
$\boldsymbol{K}^\top$	The transpose of a matrix $\boldsymbol{K} \in \mathbb{R}^{m \times p}$.
$\boldsymbol{I}_m$ or $\boldsymbol{I}$	The $m \times m$ identity matrix.
$\boldsymbol{v} \leq \boldsymbol{u}$	Componentwise ordering for vectors $\boldsymbol{v}$ and $\boldsymbol{u}$.
$\boldsymbol{A} \preceq \boldsymbol{B}$	The Loewner ordering for symmetric matrices $\boldsymbol{A}$ and $\boldsymbol{B}$.
$\mathbf{1}\{A\}$	Indicator random variable associated with event A.
$Y = \mathcal{O}_\alpha(X)$	Deterministic big-O notation, indicating that Y is bounded by $\|Y\| \leq C_\alpha X$.
C_α	A numerical constant that may depend on the parameter α in context.
$\mathcal{O}_p$	Probabilistic big-O notation.
$\xrightarrow{\text{p}}$	Convergence in probability.
$X \succeq Y$	Stochastic dominance of X by Y, indicating that $\mathbb{P}(X \geq z) \geq \mathbb{P}(Y \geq z)$ for all $z \in \mathbb{R}$.

$$\leq \mathbb{P}(\boldsymbol{y} \notin A_2) + \mathbb{P}(\boldsymbol{y} \notin A_1) \tag{SM1.3}$$

$$= \gamma + \eta \tag{SM1.4}$$

so that $\mathbb{P}(\boldsymbol{y} \in A_3) \geq 1 - \gamma - \eta$. Imposing $1 - \gamma - \eta \geq 1 - \alpha$ gives the desired result.

SM1.2. Proof of Corollary 3.2. By definition, $\bar{q}^{\mu}_{\gamma,\eta} \leq \bar{q}_{\gamma,\eta}$ for all $\mu \in \mathbb{R}$. Therefore, $C^{\text{sl}}_\alpha(\boldsymbol{y}; \mathcal{B}_\eta) \subseteq C^{\text{gl}}_\alpha(\boldsymbol{y}; \mathcal{B}_\eta)$, and thus

$$\mathbb{P}_{\boldsymbol{x}^*}\big(\mu^* \in C^{\text{gl}}_\alpha(\boldsymbol{y}; \mathcal{B}_\eta)\big) \geq \mathbb{P}_{\boldsymbol{x}^*}\big(\mu^* \in C^{\text{sl}}_\alpha(\boldsymbol{y}; \mathcal{B}_\eta)\big) \geq 1 - \alpha. \tag{SM1.5}$$

SM2. Proof of Theorem 4.1. Throughout the proof, we make use of the following lemma:

Lemma SM2.1. *Let $f : \mathbb{R}^n \to \mathbb{R}$ and $\mathcal{X} \subset \mathbb{R}^n$ such that $\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})$ is achieved, and at least one of the minimizers $\boldsymbol{x}^*$ satisfies:*

1. *f is continuous at $\boldsymbol{x}^*$,*
2. *$\boldsymbol{x}^*$ is not an isolated point of $\mathcal{X}$ (i.e., $\forall \delta > 0, B_\delta(\boldsymbol{x}^*) \cap \mathcal{X} \neq \emptyset$).*

Let μ be a measure on $\mathcal{X}$ such that $\mu(B) > 0$ for all $B \subseteq \mathcal{X}$ such that $\lambda_{Leb}(B) > 0$ (where $\lambda_{Leb}(B)$ refers here to the Lebesgue measure of the set B). Let $Y_m = \min_{i=1,\ldots,m} f(\boldsymbol{x}_i)$, where $\boldsymbol{x}_i$

are i.i.d. samples from μ. Then $Y_m \xrightarrow{p} f(\boldsymbol{x}^*)$.

Proof. Fix $\varepsilon > 0$ and let us show that $\mathbb{P}(|Y_m - f(\boldsymbol{x}^*)| > \varepsilon) \to 0$. Since f is continuous at $\boldsymbol{x}^*$ there exists a $\delta > 0$ such that $f(B_\delta(\boldsymbol{x}^*)) \subset B_\varepsilon(f(\boldsymbol{x}^*))$ so that,

$$\mathbb{P}(|Y_m - f(\boldsymbol{x}^*)| \geq \varepsilon) \leq \mathbb{P}(\boldsymbol{x}_i \notin B_\delta(\boldsymbol{x}^*),\ \forall i = 1, \dots, m) = (\mathbb{P}_{\boldsymbol{x}\sim\mu}(\boldsymbol{x} \notin B_\delta(\boldsymbol{x}^*)))^m. \tag{SM2.1}$$

Since $\boldsymbol{x}^*$ is not an isolated point, we have $\lambda_{\text{Leb}}(B_\delta(\boldsymbol{x}^*) \cap \mathcal{X}) > 0$ and therefore $\mathbb{P}_{\boldsymbol{x}\sim\mu}(\boldsymbol{x} \notin B_\delta(\boldsymbol{x}^*)) < 1$ and $\mathbb{P}(|Y_m - f(\boldsymbol{x}^*)| \geq \varepsilon) \to 0$. ∎

We begin by proving that the empirical maximums of the quantiles obtained both by Algorithms 3.1 and 3.2 (assuming the quantile regressor is consistent) converge to the true max quantile. **Algorithm 3.1**

Let $\bar{q}^{\text{de}}_{\gamma,\eta} := \max_{i=1,\dots,M} \widehat{q}^i_\gamma(N)$, where we explicitly write the dependence with the number of samples and the index i refers to the quantile estimated at the i-th sampled point $\boldsymbol{x}_i$. We aim to show that $\bar{q}^{\text{de}}_{\gamma,\eta} \xrightarrow{\text{p}} \bar{q}_{\gamma,\eta}$. We know that $\widehat{q}^i_\gamma(N)$ converges in probability to $Q_{P_{\boldsymbol{x}_i}}(1-\gamma)$ as $N \to \infty$. We have

$$\left|\bar{q}^{\text{de}}_{\gamma,\eta} - \bar{q}_{\gamma,\eta}\right| \leq \left|\bar{q}^{\text{de}}_{\gamma,\eta} - \max_{i=1,\dots,M} Q_{P_{\boldsymbol{x}_i}}(1-\gamma)\right| + \left|\max_{i=1,\dots,M} Q_{P_{\boldsymbol{x}_i}}(1-\gamma) - \bar{q}_{\gamma,\eta}\right|. \tag{SM2.2}$$

The first term can be made smaller than $\varepsilon/2$ as $N \to \infty$ by convergence of the estimator, and the second term can be made smaller than $\varepsilon/2$ as $M \to \infty$ by application of Lemma SM2.1 to the quantile function (maximizing instead of minimizing).

Algorithm 3.2

The proof is identical to that of Algorithm 3.1, with the only difference of replacing the quantiles estimated via Monte Carlo sampling with those estimated by the quantile regression, and N to M_{tr}, the number of samples needed to train the quantile regression. Since the quantile regression is assumed to be consistent, the first term can be made arbitrarily small as M_{tr} grows, and the result follows.

Since identical convergence results apply for both algorithms, we will not explicitly distinguish. The proof is written in terms of N, which can be replaced by M_{tr}.

SM2.1. Proof of Statement 1 (Global Inverted). Recall,

$$C^{\text{gl}}_{\text{inv}}(\boldsymbol{y}) = \left[\min_{k\in\{1,\dots,M\}:\lambda(\varphi(\boldsymbol{x}_k),\boldsymbol{y})\leq\widehat{q}(N)} \varphi(\boldsymbol{x}_k), \max_{k:\lambda(\varphi(\boldsymbol{x}_k),\boldsymbol{y})\leq\widehat{q}(N)} \varphi(\boldsymbol{x}_k)\right] \tag{SM2.3}$$

$$= \left[\min_{k\in\{1,\dots,M\}:\lambda(\mu_k,\boldsymbol{y})\leq\widehat{q}(N)} \mu_k, \max_{k:\lambda(\mu_k,\boldsymbol{y})\leq\widehat{q}(N)} \mu_k\right] \tag{SM2.4}$$

where $\widehat{q}(N) := \max_{i=1,\dots,M} \widehat{q}^i_\gamma(N)$, the estimated quantiles of the sampled $\boldsymbol{x}_i \in \mathcal{B}_\eta$ and $\mu_i := \varphi(\boldsymbol{x}_i)$. Note that μ_i are samples in $\varphi(\mathcal{B}_\eta) \subset \mathbb{R}$. Also note that we have more explicitly written out the interval definition (i.e., Equation (3.8)) to emphasize clarity rather than presentation.

Consider the left extreme of the interval; a similar argument follows from the right extreme. Consider three quantities:

$$\mu_1(N, M) := \min \mu_k \quad \text{s.t.} \quad k = 1, \dots, M \text{ and } \lambda(\mu_k, \boldsymbol{y}) \leq \widehat{q}(N) \tag{SM2.5}$$

$$\mu_2(M) := \min \mu_k \quad \text{s.t.} \quad k = 1, \dots, M \text{ and } \lambda(\mu_k, \boldsymbol{y}) \leq \bar{q}_{\gamma,\eta} \tag{SM2.6}$$

$$\mu_3 := \min \mu \quad \text{s.t.} \quad \mu \in \varphi\left(\mathcal{B}_\eta\right) \text{ and } \lambda(\mu, \boldsymbol{y}) \leq \bar{q}_{\gamma,\eta} \tag{SM2.7}$$

Our goal is to show that as $N, M \to \infty$, $\mu_1 \xrightarrow{\text{p}} \mu_3$. Lemma SM2.1 shows that $\mu_2 \xrightarrow{\text{p}} \mu_3$. Indeed, the minimization over the indices k such that the condition is satisfied can be seen as a rejection sampling strategy in which all accepted samples are samples of the feasible region of the optimization in μ_3. As M grows, since the sampler eventually samples all areas of $\varphi(\mathcal{B}_\eta)$, some samples are guaranteed to be close to the optimum with high probability. Finally, for fixed M and N going to infinity, $\mu_1 \xrightarrow{\text{p}} \mu_2$. This follows from the continuity of the optimization problem with respect to the right-hand side of the constraint, and the fact that $\max_{i=1,\dots,M} q(\boldsymbol{x}_i) \xrightarrow{\text{p}} \bar{q}_{\gamma,\eta}$. It follows that as $N, M \to \infty$, $\mu_1 \xrightarrow{\text{p}} \mu_3$.

SM2.2. Proof of Statement 2 (Sliced Inverted). The proof technique is similar to the one of Statement 1, replacing $\widehat{q}(N) := \max_{i=1,\dots,M} \widehat{q}^i_\gamma(N)$ for $\widehat{q}^k_\gamma(N)$. Defined then, similarly to the previous proof:

$$\mu_1(N, M) := \min \mu_k \quad \text{s.t.} \quad k = 1, \dots, M \text{ and } \lambda(\mu_k, \boldsymbol{y}) \leq \widehat{q}^k_\gamma(N) \tag{SM2.8}$$

$$\mu_2(M) := \min \mu_k \quad \text{s.t.} \quad k = 1, \dots, M \text{ and } \lambda(\mu_k, \boldsymbol{y}) \leq \bar{q}_{\gamma,\eta}(\mu_k) \tag{SM2.9}$$

$$\mu_3 := \min \mu \quad \text{s.t.} \quad \mu \in \varphi\left(\mathcal{B}_\eta\right) \text{ and } \lambda(\mu, \boldsymbol{y}) \leq \bar{q}_{\gamma,\eta}(\mu) \tag{SM2.10}$$

where $\bar{q}_{\gamma,\eta}(\mu) = \max_{\boldsymbol{x} \in \Phi_\mu \cap \mathcal{B}_\eta} Q_{\boldsymbol{x}}(1-\gamma)$. As $N \to \infty$, $\widehat{q}^k_\gamma(N) \to Q_{\boldsymbol{x}_k}(1-\gamma)$. Lemma SM2.1 can be used to show $\mu_2 \xrightarrow{\text{p}} \mu_3$, because the sampler strategy used that first samples $\boldsymbol{x}_k$ and then accepts $\mu_k = \varphi(\boldsymbol{x}_k)$ as a sample if $\lambda(\varphi(\boldsymbol{x}_k), \boldsymbol{y}) < q^k_\gamma$, it can be shown that for every feasible point μ, there is eventually a sample close to it.

Finally, μ_1 will become arbitrarily close to μ_2 as M grows large, since both are taking the minimum over samples that are sampled densely from the feasible set, meaning accepted μ_k will eventually be close to μ_3 for both the case of μ_1 and the case of μ_2

SM2.3. Proof of Statement 3 (Global Optimized). We prove the continuity of the optimization problem,

$$\max_{\mu \in \varphi(\mathcal{B}_\eta)} \mu \text{ s.t. } \lambda(\mu, \boldsymbol{y}) \leq q, \tag{SM2.11}$$

as a function of q in the positive measure interval $(\lambda(\bar{\mu}, \boldsymbol{y}), \bar{q}_{\gamma,\eta}]$. Therefore, convergence in probability follows as we have convergence in probability to $\bar{q}_{\gamma,\eta}$ as $N, M \to \infty$, which in particular implies that the maximum quantile estimate is in the interval $(\lambda(\bar{\mu}, \boldsymbol{y}), \bar{q}_{\gamma,\eta}]$ almost surely. We do so by appealing to the maximum theorem [SM16, § E.3], which, in general, guarantees continuity of functions of the form $f^*(\theta) = \sup\{f(x, \theta) : x \in C(\theta)\}$ as long as f is continuous, C is a continuous compact-valued correspondence and $C(\theta)$ is non-empty for all $\theta \in \Theta$. The continuity of C comes from the continuity of the LLR, f is equal to the identity and the strict feasibility condition ensures C is non-empty in $\Theta := (\lambda(\bar{\mu}, \boldsymbol{y}), \bar{q}_{\gamma,\eta}]$.

SM2.4. Proof of Statement 3 (Sliced Optimized). We will prove sufficient conditions for convergence of $\inf_{\mu : \widehat{f}_k(\mu) \geq 0} \mu$ to $\inf_{\mu : f(\mu) \geq 0} \mu$ as $\widehat{f}_k$ converges to f, and the result will follow by

taking $\widehat{f}_k(\mu) = \widehat{m}_\gamma(\mu) - \lambda(\mu, \boldsymbol{y})$ and $f(\mu) = m_\gamma(\mu) - \lambda(\mu, \boldsymbol{y})$. A similar argument can be repeated for the supremum. Use the notation $\widehat{f}_k$ to indicate that k sampled points are used to estimate this function via the definition of $\widehat{m}_\gamma(\mu)$ (see Subsection 3.2).

Lemma SM2.2. *Let $f : \mathbb{R} \to \mathbb{R}$ be a function, and let f_k be a sequence of functions $f_k : \mathbb{R} \to \mathbb{R}$. Let $\mu^* = \inf_{f(\mu)\geq 0} \mu$ and $\mu_k = \inf_{f_k(\mu)\geq 0} \mu$. Let the sequence of functions $\{f_k\}$ be such that for all $\delta > 0$,*

$$\mathbb{P}\left(\sup_{\mu} |f_k(\mu) - f(\mu)| > \delta\right) \to 0 \text{ as } k \to \infty, \tag{SM2.12}$$

namely, f_k converges in probability uniformly to f. Furthermore, let f be such that for all $\varepsilon > 0$, there exists $\delta' > 0$ such that if $|\mu - \mu^*| \geq \varepsilon$, then $|f(\mu)| > \delta'$. Then, we have $\mu_k \xrightarrow{\mathrm{p}} \mu^*$.

Proof. Assume for the sake of contradiction that there exists $\varepsilon > 0$ such that $\mathbb{P}(|\mu_k - \mu^*| \geq \varepsilon)$ does not go to 0. Then, by the condition on f, it must follow that $\mathbb{P}(|f(\mu_k)| > \delta)$ does not go to 0. But,

$$\mathbb{P}(|f(\mu_k)| > \delta) \leq \mathbb{P}(|f(\mu_k) - f_k(\mu_k)| > \delta) + \mathbb{P}(|f_k(\mu_k)| > \delta), \tag{SM2.13}$$

and the right-hand side goes to 0 by uniform convergence in probability (first term) and by feasibility of μ_k (second term). ∎

SM3. Sampling the Berger-Boos set.

SM3.1. VGS Sampler for low-dimensional and full column rank settings. Under the linear-Gaussian assumptions, when $\boldsymbol{K}$ has full column rank, it can be shown that

$$\mathcal{B}_\eta = \left\{\boldsymbol{x} \in \mathcal{X} : (\boldsymbol{x} - \widehat{\boldsymbol{x}})^\top \boldsymbol{K}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{K} (\boldsymbol{x} - \widehat{\boldsymbol{x}}) \leq \chi^2_{n,\eta}\right\}, \tag{SM3.1}$$

where $\widehat{\boldsymbol{x}}$ is the Generalized Least-Squares (GLS) estimator. Equation (SM3.1) describes an ellipsoid in $\mathbb{R}^p$ intersected with $\mathcal{X}$ with axis directions and lengths determined by $\chi^2_{n,\eta}$ and the eigenvectors and eigenvalues of $\boldsymbol{K}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{K}$, respectively (see [SM17] for more details). An ellipsoid is nothing but a deformed ball. As such, we can sample uniformly at random from the ellipsoid in $\mathbb{R}^p$ using an algorithm sampling uniformly at random from a p-ball, followed by an appropriate linear transformation and translation. We can then include an additional accept-reject step to account for the constraint set $\mathcal{X}$. Note that all subsequent discussions of spheres and balls assume unit radii and centering at the origin. Also note that because the ellipsoid in (SM3.1) is centered around the GLS estimator, we can sample points first from within an ellipsoid of the same shape centered at the origin, and then translate those points by $\widehat{\boldsymbol{x}}$.

[SM23] propose a particularly efficient and clever algorithm to sample uniformly at random from the p-ball by proving a connection between uniform sampling on the $(p+1)$-sphere (i.e., the surface of the ball in $\mathbb{R}^{p+2}$) and uniform sampling within the p-ball (i.e., *within* the unit ball in $\mathbb{R}^p$). Namely, one can sample a Gaussian $\widetilde{\boldsymbol{z}} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{p+2})$ and normalize $\boldsymbol{z} := \widetilde{\boldsymbol{z}}/\|\widetilde{\boldsymbol{z}}\|_2$ to sample a point uniformly at random from the $(p+1)$-sphere. Next, one simply drops the last two elements of $\boldsymbol{z}$ to obtain a point that is uniformly sampled from within a p-ball. The

validity of this approach is substantiated by Lemma 1 and Theorem 1 in [SM23] and relies upon a distribution result about the ratio of chi-squared distributions and preservation of distribution under an orthogonal transformation. To denote sampling a point following this procedure, we use the notation $\boldsymbol{x} \sim \text{VGS}(p)$.

To sample from our desired ellipsoid in (SM3.1), first consider the eigendecomposition of $\boldsymbol{K}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{K} = \boldsymbol{P} \boldsymbol{\Omega} \boldsymbol{P}^\top$, where $\boldsymbol{\Omega} = \text{diag}\left(\omega_1^2, \omega_2^2, \ldots, \omega_p^2\right)$, ω_i is the i-th eigenvalue of $\boldsymbol{K}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{K}$ and $\boldsymbol{P}$ is an orthonormal matrix where the columns vectors are the eigenvectors of $\boldsymbol{K}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{K}$. Denote $\boldsymbol{\Omega}^{1/2} = \text{diag}\left(\omega_1, \omega_2, \ldots, \omega_p\right)$ and by extension, $\boldsymbol{\Omega}^{-1/2} = \text{diag}\left(\omega_1^{-1}, \omega_2^{-1}, \ldots, \omega_p^{-1}\right)$ when $\omega_i > 0$ for all i. These decompositions imply the correct transformation to apply to points sampled from the p-ball via $\boldsymbol{x} \sim \text{VGS}(p)$. Namely, define $\boldsymbol{w} := \sqrt{\chi^2_{n,\eta}} \boldsymbol{P} \boldsymbol{\Omega}^{-1/2} \boldsymbol{x}$. To know that $\boldsymbol{w}$ is sampled from the correct ellipsoid, it should be the case that $\boldsymbol{w}^\top \boldsymbol{K}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{K} \boldsymbol{w} \leq \chi^2_{n,\eta}$, as then we can simply make the update $\boldsymbol{w} \leftarrow \boldsymbol{w} + \widehat{\boldsymbol{x}}$ to ensure that we have a point sampled from (SM3.1). This guarantee is verified as follows:

$$\begin{aligned} \boldsymbol{w}^\top \boldsymbol{K}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{K} \boldsymbol{w} &= \boldsymbol{w}^\top \boldsymbol{P} \boldsymbol{\Omega}^{1/2} \boldsymbol{\Omega}^{1/2} \boldsymbol{P}^\top \boldsymbol{w} \\ &= \chi^2_{n,\eta} \cdot \boldsymbol{x}^\top \boldsymbol{\Omega}^{-1/2} \boldsymbol{P}^\top \boldsymbol{P} \boldsymbol{\Omega}^{1/2} \boldsymbol{\Omega}^{1/2} \boldsymbol{P}^\top \boldsymbol{P} \boldsymbol{\Omega}^{-1/2} \boldsymbol{x} \\ &= \chi^2_{n,\eta} \cdot \boldsymbol{x}^\top \boldsymbol{x} \leq \chi^2_{n,\eta}, \end{aligned} \tag{SM3.2}$$

where $\boldsymbol{P}^\top \boldsymbol{P} = \boldsymbol{I}$ by definition and the last line follows since we know $\boldsymbol{x}$ is sampled from the p-ball and therefore $\boldsymbol{x}^\top \boldsymbol{x} \leq 1$. Equation (SM3.2) justifies that the $\boldsymbol{w}$ samples lie within the desired ellipsoid, and their uniform distribution follows from the uniform distribution of $\boldsymbol{x}$ in the p-ball since the distribution is preserved under a linear transformation. Finally, we accept $\boldsymbol{w}$ if $\boldsymbol{w} \in \mathcal{X}$ and reject if $\boldsymbol{w} \notin \mathcal{X}$. This procedure for sampling at random from $\mathcal{B}_\eta$ is summarized in Algorithm SM3.1.

Algorithm SM3.1 VGS Sampler

Input: $p \in \mathbb{N}$, $M \in \mathbb{N}$, $\boldsymbol{P}$, $\boldsymbol{\Omega}^{-1/2}$, $\widehat{\boldsymbol{x}}$, $\chi^2_{n,\eta}$.

1: Define $\mathcal{S} := \{\}$ to be the initialized set in which the sampled points are to be placed.
2: **for** $k = 1, 2, \ldots, M$ **do**
3: **Sample from the p-ball using VGS:** $\boldsymbol{x}_k := \boldsymbol{z}_{1:p} / \|\boldsymbol{z}\|_2$, where $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_{p+2})$ and $\boldsymbol{z}_{1:p}$ denotes taking the first through p-th indices (inclusive).
4: **Transform VGS output:** $\boldsymbol{w}_k := \sqrt{\chi^2_{n,\eta}} \boldsymbol{P} \boldsymbol{\Omega}^{-1/2} \boldsymbol{x}_k$.
5: **Translate $\boldsymbol{w}_k$ by the GLS estimator:** $\boldsymbol{w}_k \leftarrow \boldsymbol{w}_k + \widehat{\boldsymbol{x}}$.
6: **Accept-Reject to incorporate constraints**: If $\boldsymbol{w}_k \in \mathcal{X}$, then add $\mathcal{S} \leftarrow \mathcal{S} \cup \{\boldsymbol{w}_k\}$, else start loop iteration k again.
7: **end for**

Output: $\mathcal{S}$ containing uniformly sampled points over $\mathcal{B}_\eta$ (as defined in (SM3.1)).

Generating samples using the VGS Sampler is efficient, but its feasibility diminishes in high dimensions because of the accept-reject step. To illustrate this point, consider the simple scenario where $\mathcal{X} = \mathbb{R}^p_+$, i.e., the non-negative orthant of $\mathbb{R}^p$ and suppose we sample from the p-ball intersected with $\mathbb{R}^p_+$ by sampling $\boldsymbol{x} \sim \text{VGS}(p)$ where $\boldsymbol{P} = \boldsymbol{\Omega} = \boldsymbol{I}$, $\widehat{\boldsymbol{x}} = \boldsymbol{0}$ and

we use 1 instead of $\chi^2_{n,\eta}$. Then, $\mathbb{P}(\boldsymbol{x} \in \mathbb{R}^p_+) = 2^{-p}$ and the acceptance probability of each sample goes to zero exponentially in p. To make this point slightly more general, we generate data $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{x}^*_p, \boldsymbol{I}_p)$ where $\boldsymbol{x}^*_p \in \mathbb{R}^p_+$ and is a vector of ones. For a collection of dimensions $p \in [2, 30]$, we sample $\boldsymbol{x} \sim \text{VGS}(p)$ with $\boldsymbol{P} = \boldsymbol{\Omega} = \boldsymbol{I}$ and $\widehat{\boldsymbol{x}} = \boldsymbol{y}$ to estimate the probability that $\boldsymbol{x}$ is in $\mathbb{R}^p_+$ and plot the results in the left panel of Figure SM1. Since the acceptance probability decays exponentially (under 10^{-4} for 30 dimensions), it becomes clear that this algorithm is inefficient in dimensions larger than 10, since the acceptance probability quickly becomes prohibitively small in regimes where a larger sample size is even more important.

Dimension is only one of two primary complicating factors. Not only is less of a (potentially) shifted p-ball's volume in the non-negative orthant as p grows but if our data are generated via $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{K}\boldsymbol{x}^*, \boldsymbol{I}_p)$ with $\boldsymbol{x}^* \in \mathbb{R}^p_+$ where the condition number of $\boldsymbol{K}$ is large, it is possible that even less of the ellipsoid from which the VGS Sampler draws points intersects with the parameter constraint. To illustrate this point, we generate a single observation from the aforementioned model using a $\boldsymbol{K} \in \mathbb{R}^{40\times 40}$ as described in Subsection 6.3. The $\boldsymbol{x}^* \in \mathbb{R}^{40}_+$ is created in the same way as that of Subsection 6.3.1. The right panel of Figure SM1 shows a computed probability mass function for the number of coordinates in a VGS Sampler draw (with $\boldsymbol{P}$ and $\boldsymbol{\Omega}$ defined such that $\boldsymbol{K}^\top\boldsymbol{K} = \boldsymbol{P}\boldsymbol{\Omega}\boldsymbol{P}^\top$) complying with the non-negativity parameter constraints. Equivalently, the right panel of Figure SM1 shows the computed probability mass function for the number of coordinates lying within the Berger–Boos set. For this particular setup, we critically note that out of 5×10^4 draws from the VGS Sampler, none of the draws had all coordinates comply with the non-negativity constraint. By contrast, for the aforementioned noise model where we more simply sample from the p-ball intersected with the non-negative orthant, the computed acceptance probability is approximately 3.35×10^{-6}, in both cases emphasizing the VGS Sampler's poor performance in high-dimensional and ill-conditioned forward model regimes.

SM3.2. Polytope sampler for general settings. In settings where the forward model is not linear and full column rank, Algorithm SM3.1 fails. The ineffectiveness of this algorithm increases if the condition number of the linear forward model is large, such that most of the pre-image ellipsoid defining the Berger–Boos set lies outside of the constraint set. Although these scenarios induce particular geometric challenges, in the linear-Gaussian case with convex $\mathcal{X}$, we are still fundamentally sampling a convex set for which there is a vast literature. For example, there is a vast sampling literature for computing Bayesian posteriors in high dimensions via nested sampling [SM18, SM1, SM4]. In particular, nested sampling has been successfully applied in high-dimensional cosmology settings using sophisticated approaches to strategically sample the parameter space by restricting prior sampling in various ways [SM4, SM14]. Although these approaches provide tools addressing a sampling setting similar to ours (i.e., the Berger–Boos set can be viewed as a portion of the parameter space defined by a cutoff on the likelihood) they are ultimately aimed at sampling from a particular distribution (i.e., the posterior), which is a stronger criterion than required here. For sampling general convex sets, simple algorithms like Hit-and-Run are available [SM19, SM10, SM11]. However, in the particular case considered here, more sophisticated and efficient algorithms can be devised. In particular, there exists a deep literature on random walks over polytopes such that the asymptotic stationary distribution of the walk is a uniform distribution over the polytope

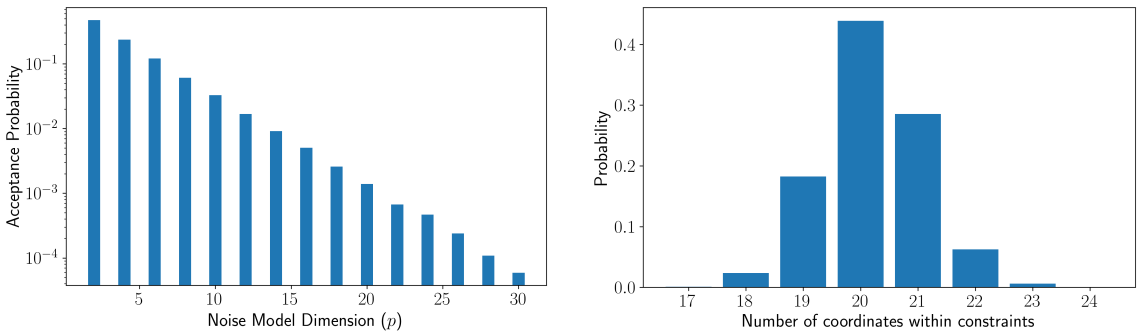

Figure SM1: Numerical illustrations of the VGS Sampler's infeasibility in high-dimensional regimes. The **left** panel shows the computed acceptance probability of a point drawn by the VGS Sampler with data generated from a non-negatively constrained Gaussian noise model. Crucially, at only 30 dimensions, the acceptance probability is already less than 10^{-4} for this particular setup. The **right** panel shows the computed probability mass function for the number of non-negative constraint complying coordinates of a VGS sample with data generated from a non-negatively constrained linear Gaussian model in 40 dimensions with a non-identity forward model. Since this is an example using a forward model with a large condition number ($\approx 1.6 \times 10^4$), we critically note that there is empirically zero probability of generating a sample within the non-negativity constraints.

of interest [SM6, SM15, SM5]. As such, we propose to first construct a bounding polytope, $\mathcal{P}^d$ composed of d half-spaces around $\mathcal{B}_\eta$, sample C random walks within the Berger–Boos set using the Vaidya walk as described in [SM5] each starting at a point from a collection of strategically chosen locations, and then combine the parallel chains to create the final sample set. The detailed algorithm can be seen in Algorithm SM3.2.

Although MCMC algorithms are typically evaluated using trace plots on individual dimensions of the parameter space, given the high-dimensionality of the problem and the final step combining several MCMC chains in the parameter space started from different positions, it is more meaningful to evaluate this algorithm's ability to sample fully from the functional space. Namely, we can solve for the largest and smallest values the functional can take within the Berger–Boos set as follows:

$$I_{\mathsf{BB}}(\boldsymbol{y}) := \left[\mu^l_{\mathsf{BB}}, \mu^u_{\mathsf{BB}}\right] = \left[\min_{\boldsymbol{x}\in\mathcal{B}_\eta} \varphi(\boldsymbol{x}), \max_{\boldsymbol{x}\in\mathcal{B}_\eta} \varphi(\boldsymbol{x})\right]. \tag{SM3.3}$$

When $\varphi(\boldsymbol{x}) = \boldsymbol{h}^\top \boldsymbol{x}$ for some $\boldsymbol{h} \in \mathbb{R}^p$, $I_{\mathsf{BB}}(\boldsymbol{y})$ corresponds to the SSB interval in [SM20]. Computing the sets $C^{\mathrm{sl}}_\alpha(\boldsymbol{y}; \mathcal{B}_\eta)$ and $C^{\mathrm{gl}}_\alpha(\boldsymbol{y}; \mathcal{B}_\eta)$ well is then contingent upon sampling functional values within $I_{\mathsf{BB}}(\boldsymbol{y})$ well, since a functional value can only be included in the inverted set if the sampler has a non-zero probability of sampling arbitrarily close to it. By "well", we informally mean that the sampled functional values range at least between the endpoints of $I_{\mathsf{BB}}(\boldsymbol{y})$ and that if we partition $I_{\mathsf{BB}}(\boldsymbol{y})$ into n sub-intervals, that most if not all of the sub-intervals contain at least one sample. In practice, it will often be the case that the sampled functional values can lie outside the endpoints of $I_{\mathsf{BB}}(\boldsymbol{y})$ since the MCMC chains are sampling a bounding polytope of the Berger–Boos set.

Choosing the bounding polytope. Any bounded polytope of $\mathcal{B}_\eta$ is the intersection of a finite number of half-spaces defined in $\mathbb{R}^p$. The task of constructing a bounding polytope is then equivalent to choosing a collection of d hyperplanes in $\mathbb{R}^p$ to construct a set $\mathcal{P}^d := \{\boldsymbol{x} \in \mathbb{R}^p : \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}\}$ such that $\mathcal{B}_\eta \subseteq \mathcal{P}^d$, where $\boldsymbol{A} \in \mathbb{R}^{d\times p}$ and $\boldsymbol{b} \in \mathbb{R}^d$. Let $\boldsymbol{a}_i^\top \in \mathbb{R}^p$ denote the i-th row vector of $\boldsymbol{A}$. We compute b_i (the i-th element of $\boldsymbol{b}$) as

$$b_i = \max_{\boldsymbol{x}\in\mathcal{B}_\eta} \boldsymbol{a}_i^\top \boldsymbol{x}. \tag{SM3.4}$$

This construction ensures the necessary inclusion. We consider three approaches to pick the vectors $\boldsymbol{a}_i$: (i) using the constraints $\mathcal{X}$, (ii) using the known eigenvectors defining the bounded directions of the pre-image ellipsoid, and (iii) randomly. In practice, we combine these approaches to ensure that we consider only parameter settings in agreement with our physical constraints, and to tighten the bounding Berger–Boos set polytope as much as possible. There is a tradeoff with respect to the latter consideration since the mixing time and computational cost for the Vaidya walk increase with the number of hyperplanes [SM5].

To incorporate the known parameter constraints, consider the non-negativity constraint used in Section 6, i.e., $\boldsymbol{x} \in \mathbb{R}^p_+$. To enforce non-negativity, we set $\boldsymbol{a}_i = -\boldsymbol{e}_i$ for $i = 1, \ldots, p$, where $\boldsymbol{e}_i$ is defined by its i-th element set to one and the rest of its elements set to zero, and $b_i = 0$. These choices produce p rows in $\boldsymbol{A}$ corresponding to the desired lower bounds (i.e., $x_i \geq 0$ for all i), but we can compute an additional p constraints using (SM3.4) with $\boldsymbol{a}_i := \boldsymbol{e}_i$ for $i = p+1, \ldots 2p$. These $2p$ constraints define a hyperrectangle enclosing the Berger–Boos set in the parameter space. To incorporate polytope constraints based upon the forward model, we use the ellipsoidal definition of the pre-image as shown in (SM3.1) and eigendecomposition of $\boldsymbol{K}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{K} = \boldsymbol{P}\boldsymbol{\Omega}\boldsymbol{P}^\top$ shown in (SM3.2). Note that this ellipsoid form is valid under the linear-Gaussian noise assumption and for all $\boldsymbol{K}$. In the event that $\boldsymbol{K}$ is not full column rank, the ellipsoid defined by Equation (SM3.1) is still defined via the pseudo-inverse of $\boldsymbol{K}^\top\boldsymbol{\Sigma}\boldsymbol{K}$, but where the ellipsoid is unbounded in some directions. The column vectors of $\boldsymbol{P}$ corresponding to the non-zero entries on the diagonal of $\boldsymbol{\Omega}$ are the eigenvectors corresponding to the bounded principal axes. As such, both $\boldsymbol{p}_i$ and $-\boldsymbol{p}_i$ for $i = 1, \ldots, p$ (the column vectors of $\boldsymbol{P}$) can be used as rows of $\boldsymbol{A}$ with their corresponding bounds defined by (SM3.4). Finally, to further tighten the polytope around the Berger–Boos set, we sample a multivariate Gaussian, i.e., $\boldsymbol{a}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_p)$ to include random hyperplanes. Note, it is possible that the unbounded directions of the ellipsoid defined by Equation (SM3.1) are not bounded when intersecting with the parameter constraint set, and hence there is no bounding polytope. In this case, our interval construction should be unbounded since there is not enough information to produce a bounded confidence set for the quantity of interest.

Once the components $(\boldsymbol{A}, \boldsymbol{b})$ have been defined using some or all of the above hyperplane generation strategies, the Vaidya walk can immediately be employed to perform a random walk around the polytope. The primary intuitive requirement we wish to satisfy with any sampling scheme in this context is that every region of the Berger–Boos set has a non-zero probability of being sampled. Asymptotically, the Vaidya walk samples the desired polytope uniformly at random, which satisfies a stronger requirement. In practice, the need to sample non-uniformly can arise if there are particularly meaningful parameter settings in regions that are difficult for the random walk to reach. We explore such a case in Subsection 6.2. Additionally, although

the asymptotic distribution of the Vaidya sampler is theoretically sufficient, in practice, it often has some difficulty reaching the corners of the generated polytope. Although MCMC chain mixing is typically evaluated by looking at trace plots, this diagnostic is insufficient here because the dimension is high, and the defined polytope can make different dimensions difficult to compare. Instead, we consider how well the sampler samples the functional values $\varphi(\boldsymbol{x})$. This motivates taking a collection of starting points between the SSB endpoints and the Chebyshev center of the polytope, as described in the following section.

Constructing the parallel chain starting points. Although the random walks defined in [SM5] asymptotically sample uniformly over $\mathcal{P}^d$, given the long and thin shape of the pre-image, running the Vaidya walk from even a "good" starting position does not consistently sample the functional space well. Instead, we use the following heuristic to construct several parallel chains that constitute a complete sample when combined. We define an even number of starting points, $C \in 2\mathbb{N}$, to roughly span the Berger–Boos set, run the Vaidya walk for M_p steps from each starting point to collect parameter settings $\{\widetilde{\boldsymbol{x}}_1, \ldots, \widetilde{\boldsymbol{x}}_{M_p}\}$, and combine the samples from each walk, resulting in $C \times M_p$ total samples. In practice, C and M_p are chosen to yield a total of M samples as desired for Algorithm 3.1 or Algorithm 3.2. Since we have designed this process to avoid the neccessity of any individual chain reaching its asymptotic distribution, we do not need a burn-in period for any chain as long as we are confident that the union of the sampled points sufficiently covers Interval (SM3.3) so that we may accept or reject any quantity of interest value in that interval. We define the collection of starting points using the line segments defined by the parameters generating the endpoints of $I_{\mathsf{BB}}(\boldsymbol{y})$ and the Chebyshev center of $\mathcal{P}^d$ as defined in [SM3], and denoted by $\boldsymbol{x}_c$. This point is the center of the largest ball contained within $\mathcal{P}^d$ and therefore acts as a way to characterize the center of $\mathcal{P}^d$. We denote the parameter settings generating the endpoints of $I_{\mathsf{BB}}(\boldsymbol{y})$ by $\widehat{\boldsymbol{x}}^l$ for the lower endpoint and $\widehat{\boldsymbol{x}}^u$ for the upper endpoint. We then define a uniform grid of values $\{\tau_l\}_{l=1}^{C/2}$ such that $\tau_l \in (0,1)$ and $\tau_l < \tau_{l+1}$ for all l. For each $k \in [C]$, define $k' := \lceil k/2 \rceil$ and set $\boldsymbol{x}_k^{start} := \tau_{k'}\widehat{\boldsymbol{x}}^l + (1-\tau_{k'})\boldsymbol{x}_c$ if k is odd and $\boldsymbol{x}_k^{start} := \tau_{k'}\widehat{\boldsymbol{x}}^u + (1-\tau_{k'})\boldsymbol{x}_c$ if k is even. Creating starting positions along the lines connecting these endpoints and the Chebyshev center accommodates the chosen polytope while helping ensure that samples are chosen spanning the range of possible functional values over the Berger–Boos set.

The empirical performance of the Polytope sampler can be seen in Figure SM2, showing (left) a histogram of the functional values sampled and (right) a trace plot for the sampled Vaidya walks starting from points constructed as described above. These plots are generated for one observation from the 80-dimensional ill-posed inverse problem in the coverage study performed in Subsection 6.3. Critically, the histogram shows that the Polytope sampler samples the functional space well, and the trace plot shows that our starting point construction heuristic performs well in practice.

Sampling from the Berger–Boos set. With $\mathcal{P}^d$ and the starting positions defined, we construct a sample $\mathcal{S} := \{\widetilde{\boldsymbol{x}}_1, \ldots, \widetilde{\boldsymbol{x}}_{C\times M_p}\}$. We leave the details of the Vaidya walk to the original paper [SM5], but note an essential radius tuning parameter of the algorithm that must be chosen. This radius impacts the spread of a Gaussian proposal distribution for the walk and thus affects a new proposed point's acceptance probability. Choosing a radius that is too large results in a low acceptance rate of proposed steps, thus creating a walk that does

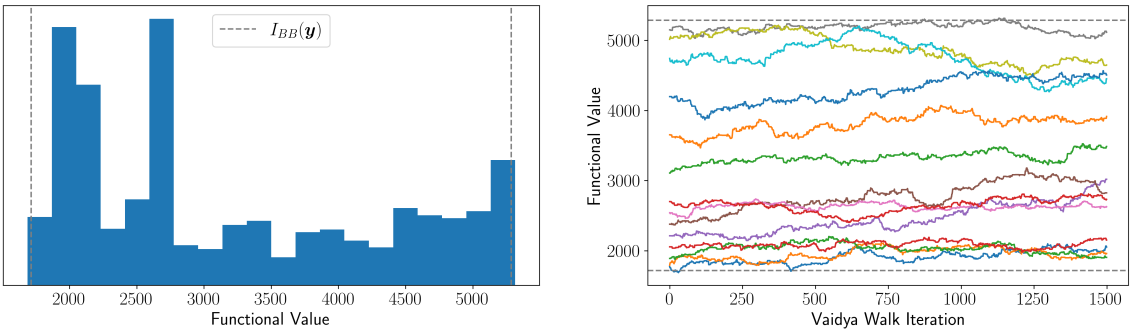

Figure SM2: Polytope sampler output for a realization of the 80-dimensional ill-posed inverse problem studied in Subsection 6.3. The **left** panel contains a histogram of sampled functional values which both span and cover well the range of $I_{\mathsf{BB}}(\boldsymbol{y})$ (shown by the dashed gray lines in both plots). The **right** panel contains trace plots of the 14 Vaidya walks (each indicated by a different color) which together constitute the full sample. Our heuristic for choosing starting points along the lines connecting the parameter settings generating the endpoints of $I_{\mathsf{BB}}(\boldsymbol{y})$ and the Chebyshev center of the polytope provides a good initial spread of starting functional values.

not mix well. In contrast, choosing a radius that is too small results in a high acceptance rate with relatively small step sizes. In practice, we find a radius setting of 0.5 works well as it produces an acceptance probability of $\approx 33.3\%$, producing a reasonable tradeoff between taking meaningful steps and not rejecting too many steps.

SM4. Quantile regression overview. Fundamentally, given a one-dimension random variable $z \sim P_{\boldsymbol{x}}$ that depends on the parameter $\boldsymbol{x} \in \mathbb{R}^p$, we are interested in the upper γ-quantile at every parameter setting, i.e.,

$$\mathbb{P}_{\boldsymbol{x}}(z > Q_{\boldsymbol{x}}(1-\gamma)) = \gamma. \tag{SM4.1}$$

We note that the quantile surface itself is not random, so we can use draws from the distribution $P_{\boldsymbol{x}}$ to estimate $Q_{\boldsymbol{x}}$ at a given parameter setting $\boldsymbol{x}$. However, as noted in Subsection 3.2, performing such an estimate is not always computationally feasible, and intuitively, we might expect quantiles to vary smoothly over the parameter space, which would imply that information about a quantile at $\boldsymbol{x}_1$ should be related to a quantile at $\boldsymbol{x}_2$ if these points are close. In statistics literature, estimating $Q_{\boldsymbol{x}}$ is framed as estimating a quantile function conditional on known covariates and is often thought of as a generalization of estimating the conditional median [SM9]. Just as conditional mean and conditional median estimation can be accomplished by using an appropriate loss function (sum of squares and absolute differences, respectively), estimating conditional quantiles can be accomplished by minimizing the pinball loss defined as follows:

$$L_\gamma(z,q) := \begin{cases} (1-\gamma)(q-z), & z < q, \\ -\gamma(z-q), & z \geq q. \end{cases} \tag{SM4.2}$$

[SM7, SM21]. As such, estimating the quantile surface can be framed as a risk-minimization problem, leaving only standard modeling choices to fill in for q in (SM4.2). Although initial

Algorithm SM3.2 Polytope sampler

Input: $M_p \in \mathbb{N}$ and $C \in 2\mathbb{N}$. $\boldsymbol{A} \in \mathbb{R}^{d\times p}$, $\boldsymbol{b} \in \mathbb{R}^d$. $\{\tau_l\}_{l=1}^{C/2}$, where $\tau_l \in (0,1)$ and $\tau_l < \tau_{l+1}$ for all l.

1: Let $\mathcal{S} := \{\}$ be the set in which we store all sampled parameter settings.
2: **Construct Chebyshev center:** Solve for $\boldsymbol{x}_c$ using the following optimization.

$$\begin{aligned} \underset{\boldsymbol{x}_c, r}{\text{maximize}} \quad & r \\ \text{subject to} \quad & \boldsymbol{a}_i^\top \boldsymbol{x}_c + r\|\boldsymbol{a}_i\|_2 \leq b_i, \quad i = 1, \ldots, d. \end{aligned}$$

3: **Compute φ extremes of Berger–Boos set:** Extreme points are computed with respect to the functional of interest:

$$\begin{aligned} \widehat{\boldsymbol{x}}^l &:= \text{argmin} \ \ \varphi(\boldsymbol{x}) \quad \text{subject to} \ \ \boldsymbol{x} \in \mathcal{B}_\eta, \\ \widehat{\boldsymbol{x}}^u &:= \text{argmax} \ \ \varphi(\boldsymbol{x}) \quad \text{subject to} \ \ \boldsymbol{x} \in \mathcal{B}_\eta. \end{aligned} \tag{SM3.5}$$

4: **for** $k = 1, 2, \ldots, C$ **do**
5: **Construct starting point:** Define $k' := \lceil k/2 \rceil$. Define $\boldsymbol{x}_k^{start} = \tau_{k'}\widehat{\boldsymbol{x}}^l + (1-\tau_{k'})\boldsymbol{x}_c$ if k is odd and $\boldsymbol{x}_k^{start} = \tau_{k'}\widehat{\boldsymbol{x}}^u + (1-\tau_{k'})\boldsymbol{x}_c$ if k is even.
6: **Run the Vaidya walk for M_p steps:** Collect samples $\{\widetilde{\boldsymbol{x}}_1, \ldots, \widetilde{\boldsymbol{x}}_{M_p}\}$ and add them to $\mathcal{S}$.
7: **end for**

Output: $\mathcal{S}$ containing sampled points over $\mathcal{B}_\eta$.

efforts were focused on linear parametric quantile regressors [SM8], in recent years, modeling efforts have focused on nonparametric varieties. [SM13] adapted random forests to quantile regression. [SM22] leveraged Reproducing Kernel Hilbert Spaces to construct smooth quantile regressors. Closer to our application, [SM12] used neural networks to optimize the pinball loss to learn the quantile surface for their application.

SM5. Additional details and illustrations in Section 6.

SM5.1. Importance-like sampler for the three-dimensional example in Subsection 6.2.

Since the parameter settings with quantiles meaningfully larger than $\chi^2_{1,\alpha}$ are located close to the constraint boundary, a sampling challenge is presented. Using the samplers as described in Subsection 5.1 results in under-sampling of this large-quantile region since both samplers provide uniform random samples over the Berger–Boos set. Algorithm SM5.1 presents a modified version of Algorithm 3.1, an importance-like sampler to increase the probability mass of samples close to the constraint boundary. Note, we say "importance-like" because we do not provide any theoretical guarantee regarding this sampler's ability to produce draws from a particular target distribution. We tailored Algorithm SM5.1 to settings with a non-negativity constraint and hand-tuned the length scale parameter to the particular three-dimensional

example in Subsection 6.2.

Algorithm SM5.1 Importance-like sampler for three-dimensional constrained Gaussian

Input: Number of samples: $M \in \mathbb{N}$, inverse length scale: γ_p, and order of norm: $q \in (0, 1)$.

Instantiate a list $\mathcal{S}$ of length M to store sampled points.
while $|\mathcal{S}| < M$ **do**
  Draw $M - |\mathcal{S}|$ realizations from Algorithm SM3.2: $\widetilde{\boldsymbol{x}}_1, \ldots, \widetilde{\boldsymbol{x}}_{M-|\mathcal{S}|}$
  for $k = 1, \ldots, M - |\mathcal{S}|$: **do**
    Compute the probability of accepting the k-th draw: $p_k := \exp\left(-\gamma_p \|\widetilde{\boldsymbol{x}}_k\|_q\right)$.
    Draw $z_k \sim \text{Bernoulli}(p_k)$.
    if $z_k = 1$ **then**
      $\mathcal{S}[k] \leftarrow \widetilde{\boldsymbol{x}}_k$
    end if
  end for
end while

Output: Sampled parameters in Berger–Boos set $\mathcal{S}$.

The key to Algorithm SM5.1 is the additional accept/reject step where the k-th sample is accepted with probability p_k. Additionally, by setting $q \in (0, 1)$, we prioritize retaining samples closer to the non-negativity boundary. This effect can be seen in Figure SM3, where the vast majority of samples are found closer to the constraint boundary. The ability of Algorithm SM5.1 to better sample the high-quantile regions of the Berger–Boos set can be seen in Figure SM4. In particular, for the functional values $\mu \in (-4, -2)$, the importance-like sampler is substantially more effective than the Polytope sampler at finding parameter settings with γ-quantiles greater than 1.1.

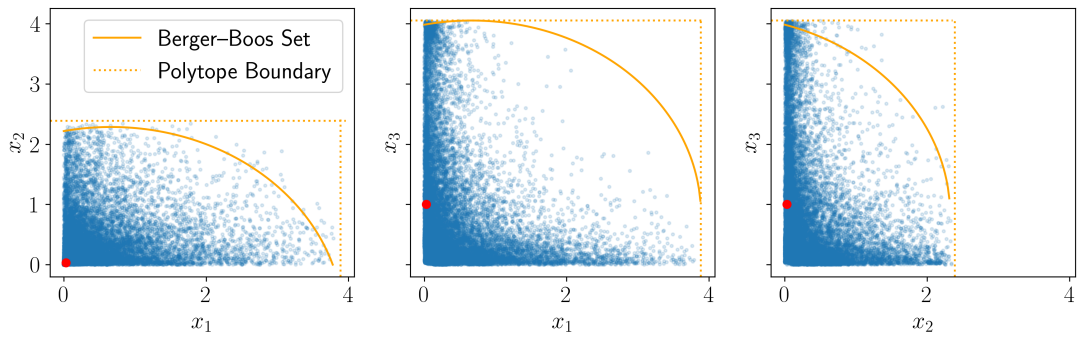

Figure SM3: When sampling points using Algorithm SM5.1, the vast majority of samples are found closer to the non-negativity constraint boundary. The true parameter setting is shown by the red point, while the parameter settings sampled by Algorithm SM5.1 are shown by the blue points. This sampling prioritization helps adequately sample the regions of the Berger–Boos set where the quantile surface is larger than $\chi^2_{1,\alpha}$. Furthermore, the vast majority of sampled points lie within the Berger–Boos set, with some lying outside within the bounding polytope.

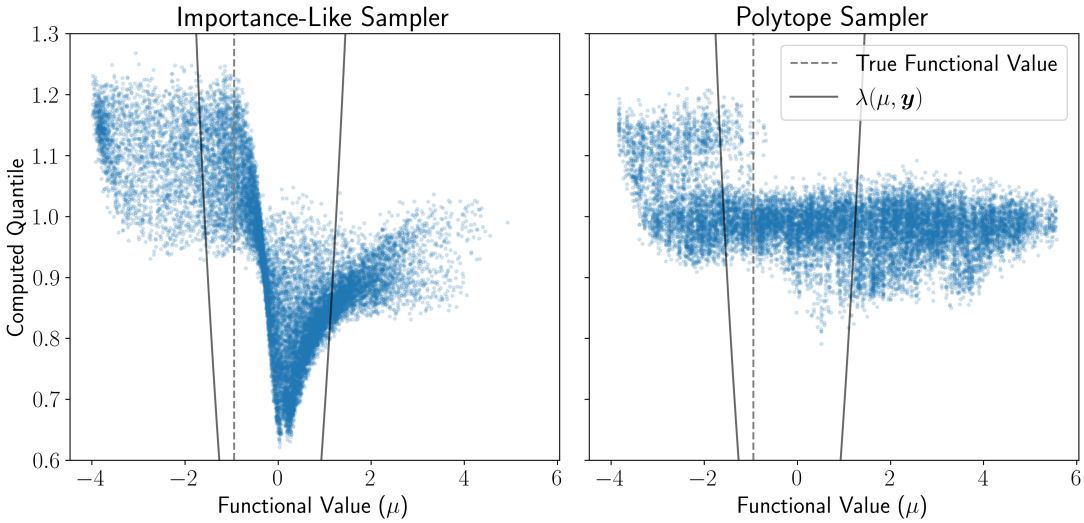

Figure SM4: The importance-like sampler described by Algorithm SM5.1 is more effective than the Polytope sampler described by Algorithm SM3.2 at sampling parameter settings with γ-quantile greater than 1.1. Each parameter setting sampled by Algorithm SM5.1 is shown by a blue point. This improved ability helps ensure the coverage guarantee shown in the left panel of Figure 4.

REFERENCES

[1] G. Ashton et al., *Nested sampling for physical scientists*, Nature Reviews Methods Primers, 2 (2022).

[2] R. L. Berger and D. D. Boos, *P values maximized over a confidence set for the nuisance parameter*, Journal of the American Statistical Association, 89 (1994), pp. 1012–1016.

[3] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[4] J. Buchner, *Nested sampling methods*, arXiv preprint arXiv:2101.09675v4, (2023).

[5] Y. Chen, R. Dwivedi, M. J. Wainwright, and B. Yu, *Fast mcmc sampling algorithms on polytopes*, Journal of Machine Learning Research, 19 (2018), pp. 1–86.

[6] R. Kannan and H. Narayanan, *Random walks on polytopes and an affine interior point method for linear programming*, Mathematics of Operations Research, 37 (2012), pp. 1 – 20.

[7] R. Koenker, *Quantile regression*, vol. 38, Cambridge university press, 2005.

[8] R. Koenker and G. Bassett Jr, *Regression quantiles*, Econometrica: journal of the Econometric Society, (1978), pp. 33–50.

[9] R. Koenker and K. F. Hallock, *Quantile regression*, Journal of Economic Perspectives, 15 (2001), pp. 143–156.

[10] L. Lovasz, *Hit-and-run mixes fast*, Mathematical Programming, 86 (1999), pp. 443 – 461.

[11] L. Lovasz and S. Vempala, *Hit-and-run from a corner*, SIAM Journal on Computing, 35 (2006), pp. 985 – 1005.

[12] L. Masserano, T. Dorigo, R. Izbicki, M. Kuusela, and A. Lee, *Simulation-based inference with waldo: Confidence regions by leveraging prediction algorithms or posterior estimators for inverse problems*, AISTATS, (2023).

[13] N. Meinshausen, *Quantile regression forests*, Journal of Machine Learning Research, 7 (2006), pp. 983–999.

[14] N. A. Montel, J. Alvey, and C. Weniger, *Scalable inference with autoregressive neural ratio estimation*, arXiv preprint arXiv:2308:08597v1, (2023).

[15] H. Narayanan, *Ranomized interior point methods for sampling and optimization*, The Annals of Applied Probability, 26 (2016), pp. 597–641.

[16] E. A. Ok, *Real Analysis with Economic Applications*, Econometric Society Monographs, Princeton University Press, Princeton, NJ, 1st ed., 2007, https://press.princeton.edu/books/hardcover/9780691129495/real-analysis-with-economic-applications. Includes bibliographical references and index.

[17] B. W. Rust and W. R. Burrus, *Mathematical Programming and the Numerical Solution of Linear Equations*, American Elsevier, 1972.

[18] J. Skilling, *Nested sampling*, AIP Conference Proceedings, 735 (2004), p. 295.

[19] R. L. Smith, *Efficient monte carlo procedures for generating points uniformly distributed over bounded regions*, Operations Research, 32 (1984), pp. 1296–1308.

[20] M. Stanley, P. Patil, and M. Kuusela, *Uncertainty quantification for wide-bin unfolding: one-at-a-time strict bounds and prior-optimized confidence intervals*, Journal of Instrumentation, 17 (2022), https://doi.org/https://doi.org/10.1088/1748-0221/17/10/P10013.

[21] I. Steinwart and A. Christmann, *Estimating conditional quantiles with the help of the pinball loss*, Bernoulli, 117 (2011), pp. 211–225.

[22] I. Takeuchi, Q. V. Le, T. D. Sears, and A. J. Smola, *Nonparametric quantile estimation*, Journal of Machine Learning, 7 (2006), pp. 1231–1264.

[23] A. R. Voelker, J. Gosmann, and T. C. Stewart, *Efficiently sampling vectors and coordinates from the n-sphere and n-ball*, tech. report, Centre for Theoretical Neuroscience, 2017.