Online Supplementary Materials for "Bayesian Safe Policy Learning with Chance Constrained Optimization: Application to Military Security Assessment during the Vietnam War" *Journal of the Royal Statistical Society, Series A (Statistics in Society)*

Zeyang Jia* Eli Ben-Michael† Kosuke Imai‡

July 14, 2025

*Ph.D. Student, Department of Statistics, Harvard University. 1 Oxford Street, Cambridge MA 02138, USA. Email: zeyangjia@fas.harvard.edu

†Assistant Professor, Department of Statistics & Data Science and Heinz College of Information Systems & Public Policy, Carnegie Mellon University, USA. 4800 Forbes Avenue, Hamburg Hall, Pittsburgh PA 15213. Email: ebenmichael@cmu.edu URL: ebenmichael.github.io

‡Professor, Department of Government and Department of Statistics, Harvard University. 1737 Cambridge Street, Institute for Quantitative Social Science, Cambridge MA 02138, USA. Email: imai@harvard.edu URL: https://imai.fas.harvard.edu

S1 Two-way and Three-way decision tables used in the HES

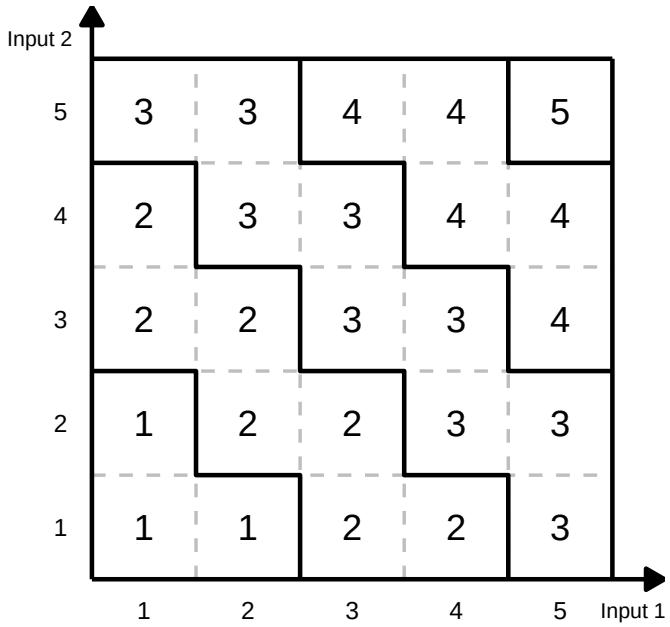


Figure S1: The two-way table used in the Hamlet Evaluation System for aggregating two input scores. The (i, j) element in the table above shows the output score when the first input is i and the second input is j.

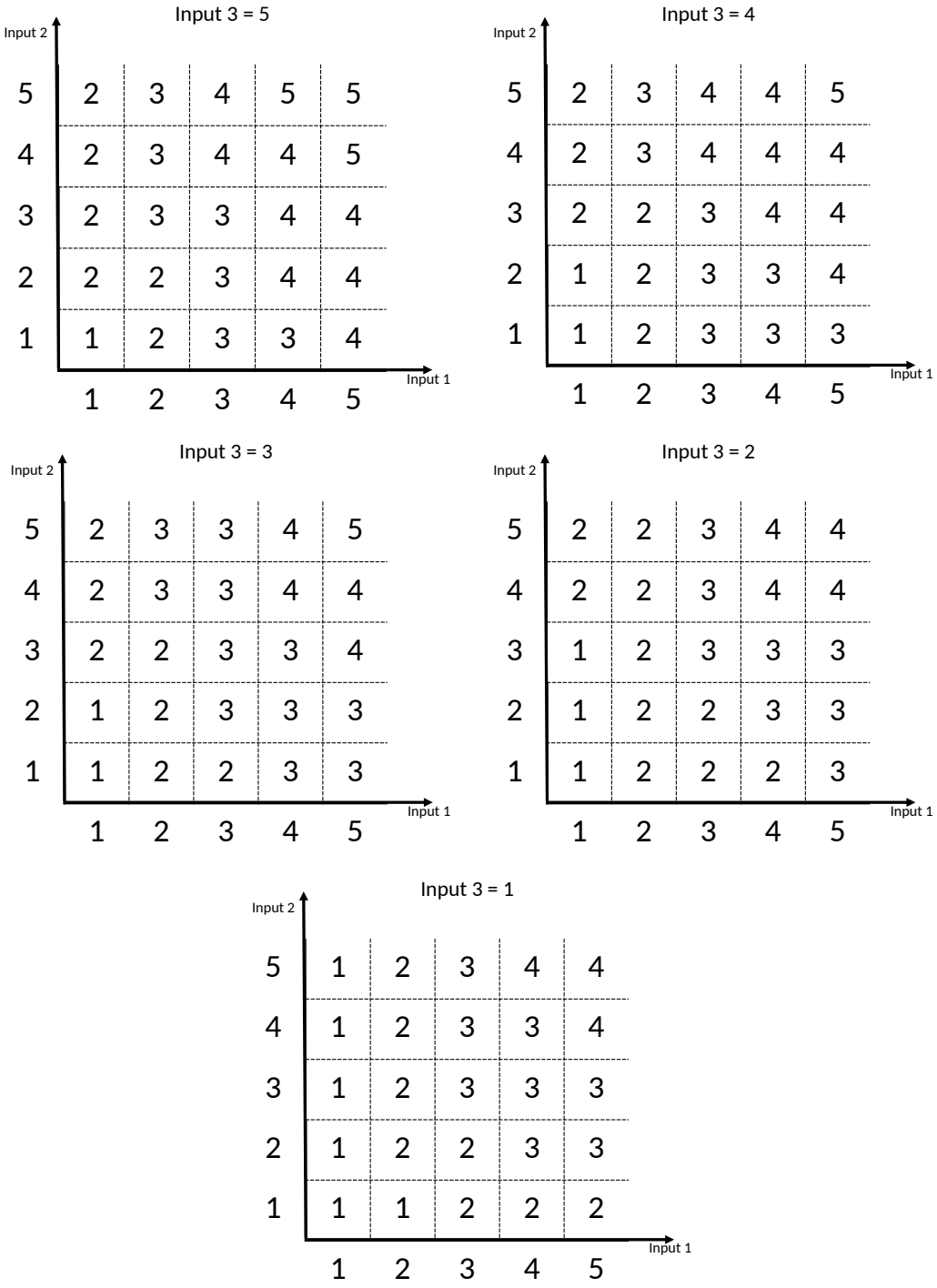


Figure S2: Three-way decision tables used in the HES. Each figure fixes the third input score and shows the output score for different combination of the first two inputs.

S2 An additional figure and table

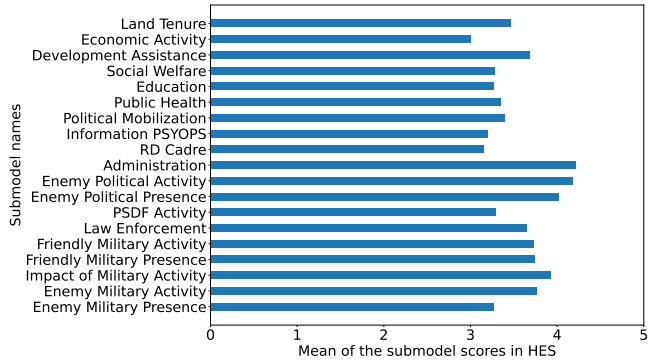


Figure S3: The mean of each sub-model score across the 1954 regions

Variable	Value
Number of regions	1954
Number of regions being attacked	1024
Average security score	3.34
Average regional safety outcome	0.53
Average regional economy outcome	0.51
Average civic society outcome	0.43

Table S1: Summary statistics on key measures

S3 BART and GP for Bayesian Inference on the CATE

Below, we consider BART and GP for Bayesian inference on these parameters $\{f_k\}_{k=0}^{K-1}$. Specifically, we sample from the posterior distribution of $\{f_k\}_{k=0}^{K-1}$ and apply Algorithm 1 to find a safe policy.

Bayesian Additive Regression Trees (BART). BART is a popular Bayesian nonparametric model that is commonly used for causal inference, especially to estimate the CATE (Taddy et al., 2016; Hahn et al., 2020). In general, BART excels in learning complex nonlinear relations while it is often poor at extrapolating. Thus, BART may be a suitable choice when there exists a substantial covariate overlap between treatment conditions.

We use a BART to model each f_k for $k \in \{0, 1, \ldots, K-1\}$ as the sum of L regression trees, i.e., $f_k(\boldsymbol{x}) = \sum_{\ell=1}^{L} g_{k\ell}(\boldsymbol{x}; T_{k\ell}, P_{k\ell})$ where $g_{k\ell}(\cdot)$ is the ℓ-th regression tree with parameter $T_{k\ell}$ and $P_{k\ell}$ denoting the structure of the regression tree and the parameters in the terminal nodes, respectively. Thus, the parameter $\boldsymbol{\Theta}$ consists of $\{T_{k\ell}, P_{k\ell}\}_{1\le\ell\le L, 0\le k\le K-1}$ as well as σ^2. We draw posterior samples for $\{T_{k\ell}, P_{k\ell}\}_{1\le\ell\le L}$ using an MCMC algorithm once a prior distribution is specified (Chipman et al., 2010).

Gaussian Process Regression. Another popular Bayesian nonparametric model is Gaussian Process regression (GP). GP has a greater degree of smoothness than BART, making it more suitable for extrapolation (Rasmussen and Williams, 2006; Branson et al., 2019). Therefore, we should consider using GP when the overlap of covariates between treatment conditions is poor.

As in the case of BART, we use a GP to model each f_k. Specifically, f_k is assumed to be a random function based on a collection of Gaussian processes. To conduct Bayesian inference on f_k, we specify a prior for f_k by giving the mean function $\mu_k(\cdot)$ and kernel function $K_k(\cdot, \cdot)$ and obtain posterior samples of f_k using the MCMC algorithm (Rasmussen and Williams, 2006; Branson et al., 2019).

When strong prior information is unavailable, we can set $\mu_k(\cdot) = 0$ for $k \ge 1$ which corresponds to

no treatment effect. For the kernel function, which determines the covariance between $f_k(\boldsymbol{x}_1), f_k(\boldsymbol{x}_2)$ for any $\boldsymbol{x}_0, \boldsymbol{x}_1 \in \mathcal{X}$, we can, for example, use Matern kernels:

$$K_{\text{Matern}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sigma_0^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}||\boldsymbol{x}_1 - \boldsymbol{x}_2||}{\ell} \right)^{\nu} B_\nu \left(\frac{\sqrt{2\nu}||\boldsymbol{x}_1 - \boldsymbol{x}_2||}{\ell} \right) \tag{S1}$$

where l is the scale parameter, σ_0^2 is the variance parameter, B_ν is the modified Bessel function of the second kind, and ν is the smoothness parameter. The hyperparameters in the Matern kernels can be selected based on prior knowledge about the smoothness of $f_k(\cdot)$. For example, to make f_k more smooth, we can increase the scale and smoothness parameters. In general, ℓ and ν determine the prior knowledge about the smoothness of f, while σ_0^2 determines the strength of this prior knowledge.

Extrapolation based on GP is similar to frequentist extrapolation methods that specify the model class by assuming a certain type of smoothness on the CATE under the robust optimization framework. For example, Ben-Michael et al. (2021) considers the case with two arms and assumes a Lipschitz constraint on the CATE, i.e., $|f_1(\boldsymbol{x}_1) - f_1(\boldsymbol{x}_2)| \leq c|\boldsymbol{x}_1 - \boldsymbol{x}_2|$. In our framework, if we specify the prior of $f_1(\boldsymbol{x})$ as a GP with mean function $m(\boldsymbol{x})$ that is c_1 Lipschitz, then the Matern kernel with scale parameter ℓ and smoothness parameter ν implies the following probabilistic Lipschitz condition:

$$\mathbb{P}\left(|f_1(\boldsymbol{x}_1) - f_1(\boldsymbol{x}_2)| > c_2||\boldsymbol{x}_1 - \boldsymbol{x}_2||\right) \leq \sigma_0^2 \left\{ \left(1 + \frac{1}{\nu - 1}\right) \frac{1}{c_2^2 l^2} + \frac{c_1^2}{c_2^2} \right\}$$

Thus, there exists a direct relationship between the prior hyperparameter of GP and the smoothness of the underlying model.

S4 Additional Simulation Results

Here we present additional detailed simulation results.

S4.1 Additional explanation on the regularization effect

In the main text, we discussed the way that the ACRisk acts as a form of regularization. Here we include more intuition about this regularization effect. Figure S4 shows the true CATE in the simulation setup in Section 5. The area in red indicates a positive CATE whereas the area in blue indicates a negative CATE. Without the ACRisk constraint, one may end up assigning the area with light blue color to the treatment condition due to finite sample error. If we impose the ACRisk constraint, however, we can reduce such errors because those areas have a large posterior uncertainty in determining the sign of the CATE.

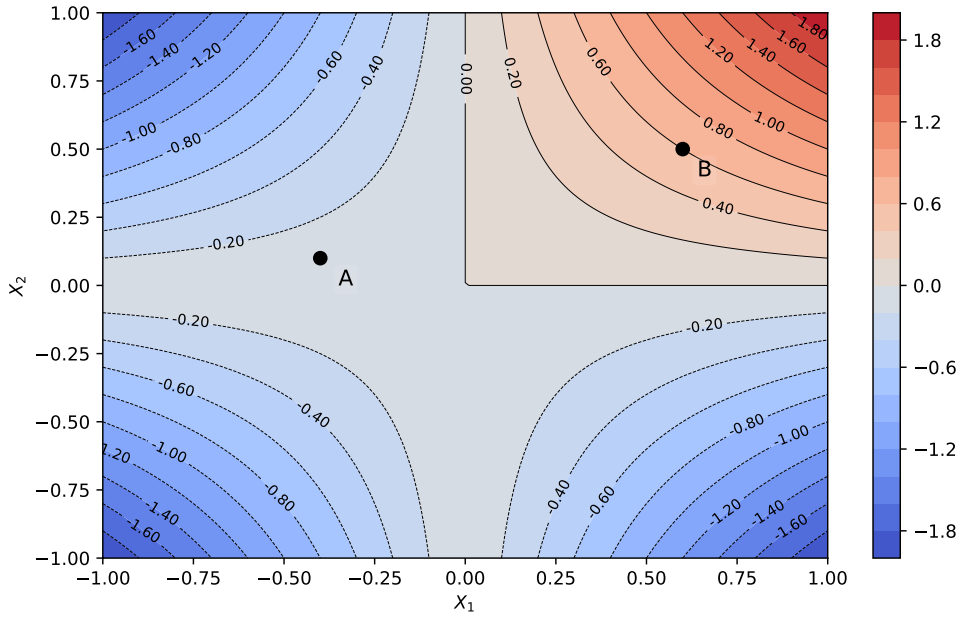
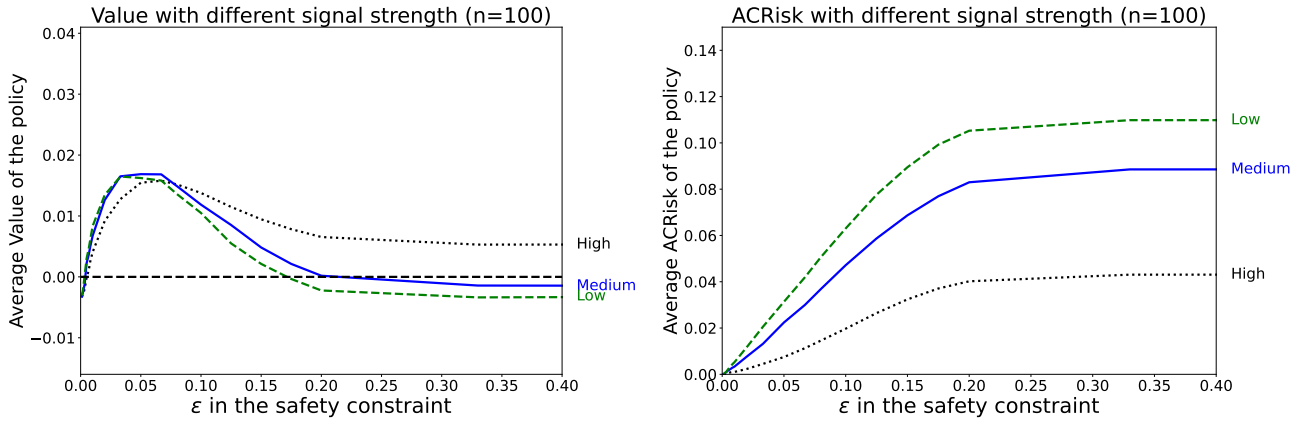


Figure S4: The true conditional average treatment effect (CATE) in the simulation study. The area with warm color indicates that the CATE is positive while the area with cold color means that the CATE is negative. The posterior ACRisk of giving the treatment at B is much smaller than the corresponding risk at A because the CATE at A is close to 0 and the uncertainty of determining its sign is large.

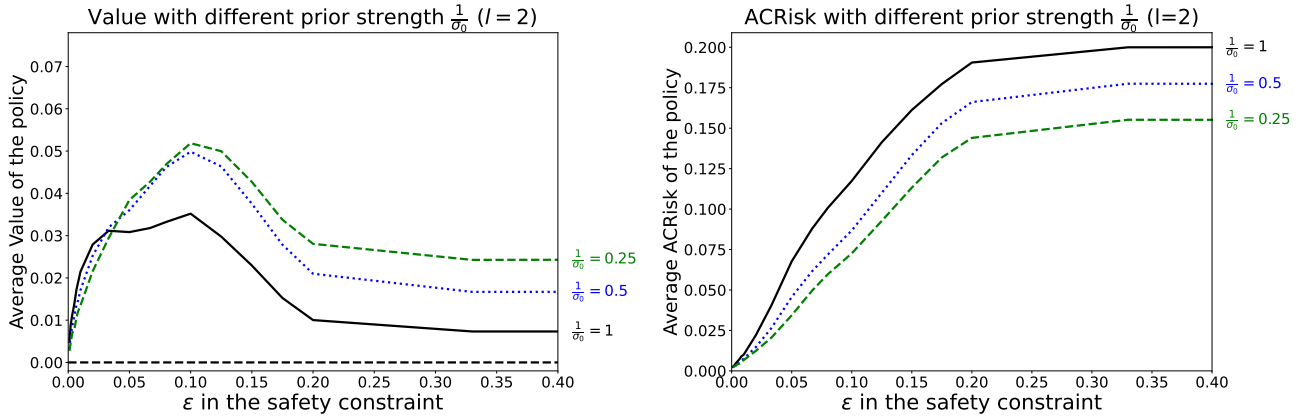
S4.2 Average Value and ACRisk with different signal strength and prior strength



(a) Average Value

(b) Average ACRisk

Figure S5: Average Value and ACRisk for learned policies using data with covariate overlap, varying the safety constraint and signal strength.



(a) Average Value

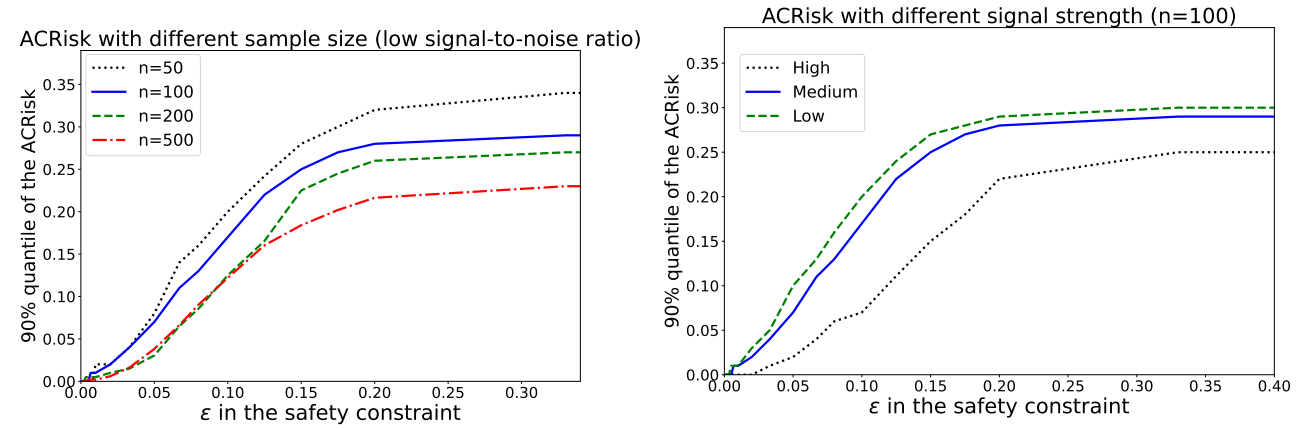
(b) Average ACRisk

Figure S6: Average Value and ACRisk for learned policies using data without covariate overlap, varying the safety constraint and prior strength for the CATE.

S4.3 The tail distribution of the empirical ACRisk

In the main text, we discuss the average ACRisk of the learned policy in multiple simulations. However, we may also be interested in the tail distribution for the ACRisk of learned policy. Therefore, we inspect the 90 percentile of the ACRisk for the learned policy across 2000 simulations. Figures S7 and S8 show how the 90 percentile of the ACRisk changes with the sample size, signal strength, smoothness for prior of the CATE, and the strength of the prior.

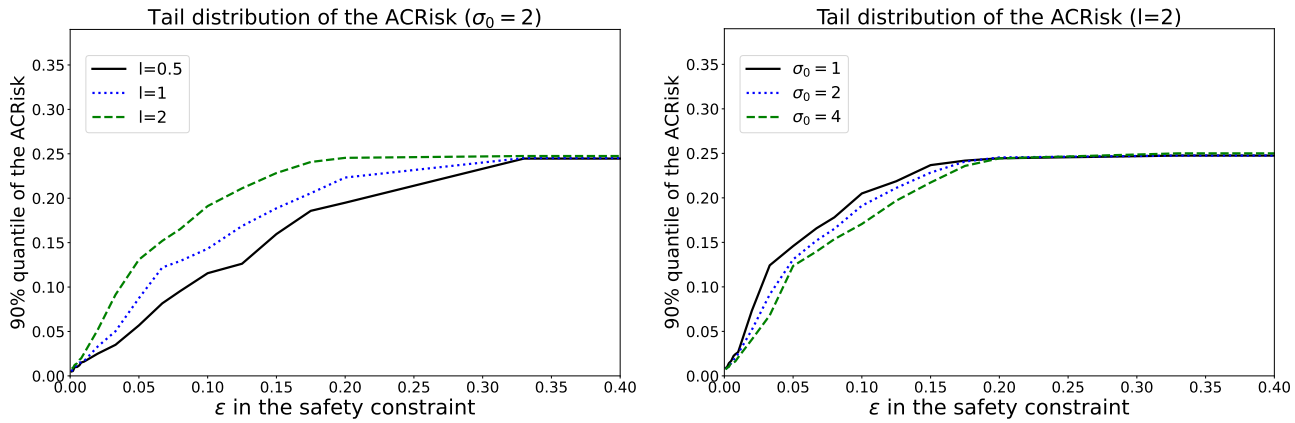
Overall, we observe similar results as the result for average ACRisk showed in the main text. The 90 percentile of the ACRisk increases as the safety constraint ϵ increases, until reaching a plateau that corresponds to the ACRisk obtained by maximizing the posterior expected utility with no constraint. A samller sample size or lower signal lead to a greater ACRisk, and a smoother or stronger prior increases the ACRisk because we estrapolate more aggressively.



(a) ACRisk with different sample size

(b) ACRisk with different signal strength

Figure S7: 90 % quantile of the ACRisk for learned policy among 2000 simulations. The CATE is estimated with BCF and covariates have overlap



(a) ACRisk with different smoothness

(b) ACRisk with different prior strength

Figure S8: 90% quantile of the ACRisk for learned policy among 2000 simulations. The CATE is estimated with a GP and there is no covariate overlap.

S4.4 Simulation results with binary outcomes

The previous simulation design focused on continuous outcomes. Here, we present simulation results when the outcome is binary. We use a similar setup as in Section 5 where the covariates $\boldsymbol{X} = (X_1, X_2)$ and $X_1, X_2 \overset{\text{i.i.d}}{\sim} Uniform[-1, 1]$. We use the same Scenario I (with covariate overlap) and Scenario II (without covariate overlap) for generating the decision D in the data. For the outcome, we let

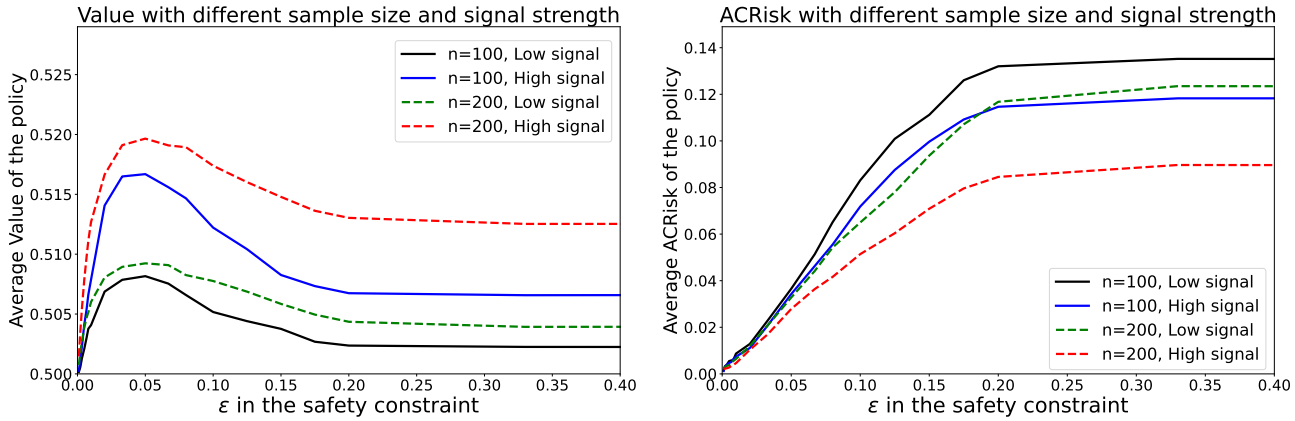
$$Y \mid \boldsymbol{X} \sim \text{Bernoulli}\left(\text{expit}\left(\frac{X_1}{2} + \frac{X_2}{2} + \gamma\{3I(X_1 > 0, X_2 > 0) - \frac{3}{2}\}D|X_1||X_2|\right)\right),$$

where we consider a strong signal case $\gamma = 2$ and a weak signal case $\gamma = 1$. We vary the number of observations $n \in \{50, 100, 200\}$, and we use a GP to model the CATE.

With covariate overlap. In the case with covariate overlap, there is no need for extrapolation. Therefore, we use a weak prior for the GP and specify the mean function $m(x) = 0$, the kernel function as a Matern kernel with $\sigma_0 = 4, l = 0.5$. We show how the average ACRisk and the value changes as a function of the safety constraint ϵ under different sample size and signal strength in Figure S9.

As shown in Figure S9, the average ACRisk increaseas as the ϵ in the safety constraint increases under all setups. Fixing the safety constraint ϵ, a larger sample size or stronger signal strength leads to a lower ACRisk. Similar to the results in the main text, we also observe a regularization effect of the safety constraint, where under appropriate safety constraint, the average value of the learned policy is higher than the value of the policy that maximizes the posterior expected utility without any safety constraint.

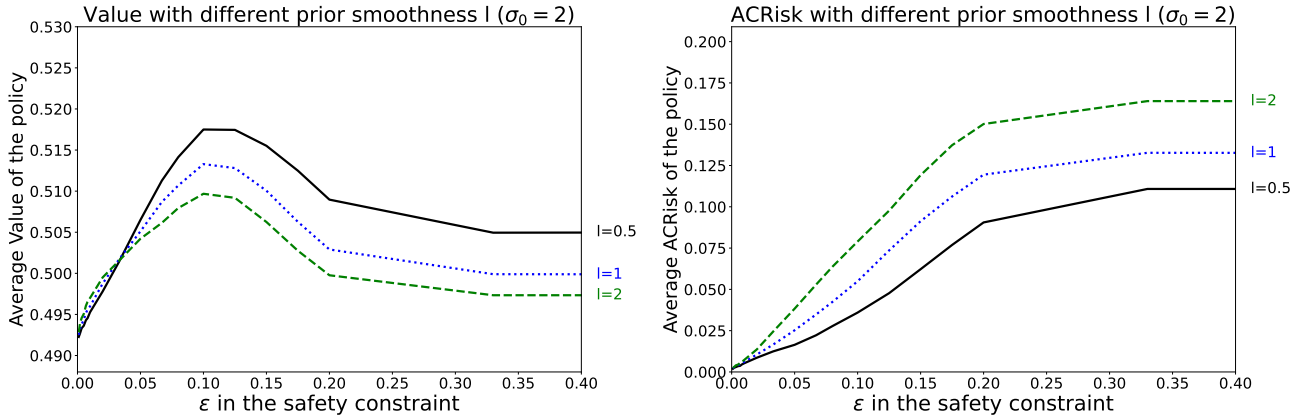
Without covariate overlap When there is no covariate overlap, the CATE must be extrapolated. We fix the sample size to 200 and the signal strength $\gamma = 2$ and investigate the average Value and ACRisk for learned policies with different priors. Figure S10 and Figure S11 shows the average Value



(a) Average Value with different sample size and signal strength

(b) Average ACRisk with different sample size and signal strength

Figure S9: Average Value and ACRisk for the learned policy among 2000 simulations. The CATE is estimated with GP and covariates have overlap

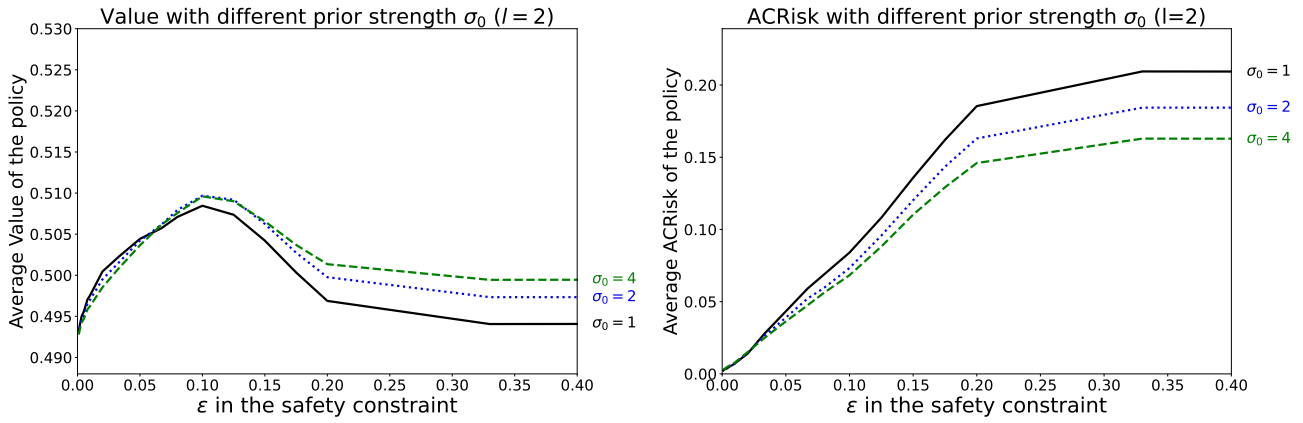


(a) Average Value

(b) Average ACRisk

Figure S10: Average Value and ACRisk for learned policies using data without covariate overlap, varying the safety constraint and prior smoothness for the CATE.

and ACRisk of the learned policy as a function of the safety constraint ϵ under different prior smoothness and prior strength. We observe that with a smoother or stronger prior, the learned has a higher average ACRisk as the extrapolation is more aggressive.



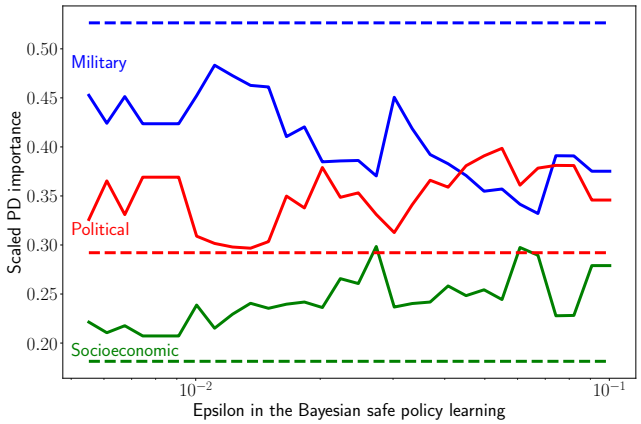
(a) Average Value

(b) Average ACRisk

Figure S11: Average Value and ACRisk for learned policies using data without covariate overlap, varying the safety constraint and prior strength for the CATE.

S5 Additional application results

S5.1 Scaled PD importance of level-3 scores

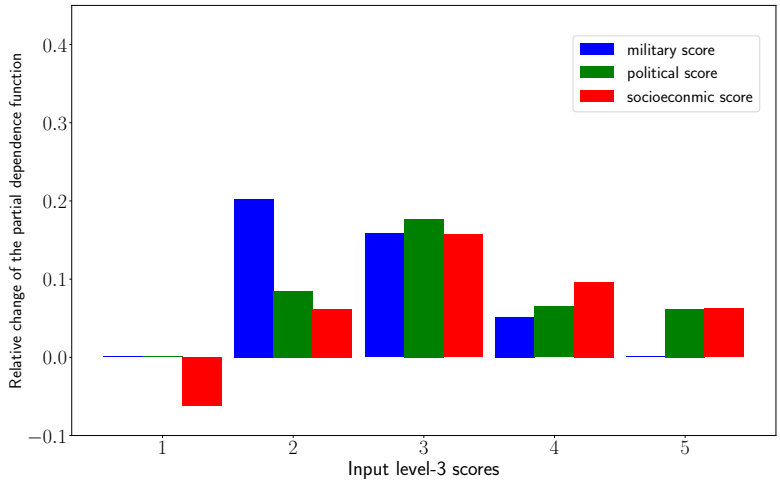


(a) Regional civic society as outcome

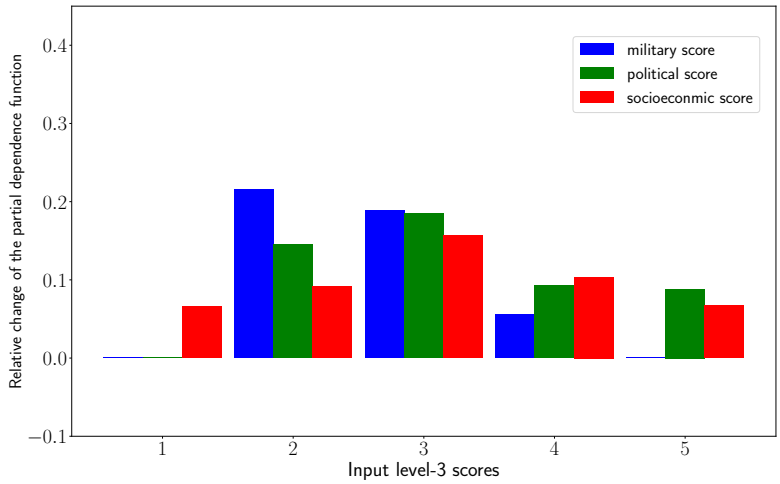
S5.2 Sensitivity analysis for prior hyperparameters

In the main text, we use a GP to estimate the CATE, relying on the GP prior for extrapolation. Here, we present a sensitivity analysis for the smoothness of the GP prior, which determines the degree of extrapolation. Other than the original setup where $l = 1$, we also consider the setups with $l = 0.5$ (less extrapolation on the CATE), $l = 2$ (more extrapolation on the CATE).

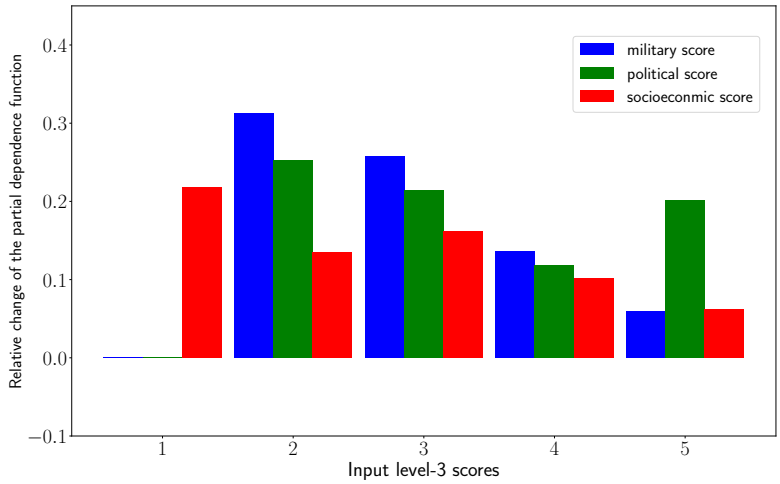
In Figure S13, we plot the partial dependence of the output security score as a function of the input level-3 score, under learned policies with different values of the prior smoothness parameter l. We use the regional safety as the outcome for policy learning, and set the safety parameter $\epsilon = 0.1$. As shown in Figure S13, with greater values of the prior smoothness parameter l for extrapolation, the obtained policies tend to systematically increase the output security score. This is consistent to



(a) $l = 0.5$

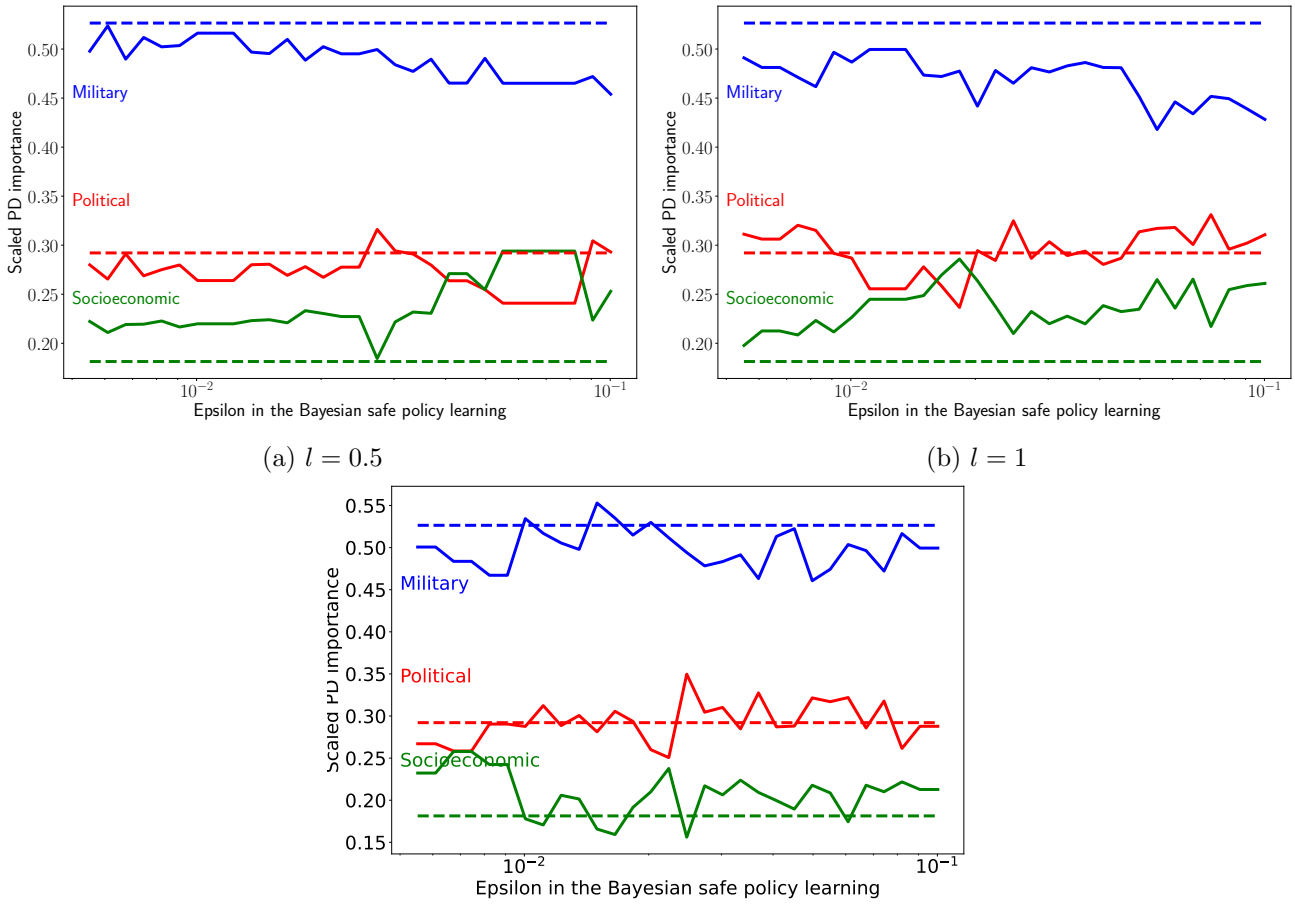


(b) $l = 1$



(c) $l = 2$

Figure S13: The relative change of the PD function from the baseline policy to the learned policy for $\epsilon = 0.1$. Each block corresponds to the different input of the PD function, and different colors corresponds to different level-3 scores.



(a) $l = 0.5$ (b) $l = 1$

(c) $l = 2$

Figure S14: The scaled Partial Dependence (PD) importance of level-3 scores of the learned policy with regional safety outcomes, as a function of the ϵ. The solid line corresponds to the learned policy, and the dashed line indicates the baseline policy. Lines with different colors shows the PD importance of different level-3 scores.

the conclusion of Dell and Querubin (2018): airstrikes increased the communist insurgency activities and decreases regional safety. Specifically, the change of the partial dependence is larger when $l = 2$, which corresponds to a stronger smoothness prior and more extrapolation. When $l = 0.5$, there is less extrapolation and the change of the partial dependence is smaller.

We further plot the relative partial dependence importance of each level-3 scores as a function of the safety constraint ϵ, varying the prior smoothness parameter l. In Figure S13, under all the prior smoothness parameter, the learned policy downweights the military sub-model scores and upweights the socioeconomic sub-model scores.

S6 Optimization for Monotonic Decision Tables

In this section, we develop an optimization algorithm applicable to monotonic decision tables where the output of a decision table is non-decreasing in each input dimension.

S6.1 Problem definition

We define the *monotonic decision table* as below:

Definition S1 (Monotonic decision tables)**.** Suppose $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_p, \mathcal{Y}$ are finite totally ordered sets, T is a function that maps $x \in \mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times ... \times \mathcal{X}_p$ to $\mathcal{Y}$. Then, T is a monotonic decision table if and only if the following condition holds,

$$\forall\ x = \{x_i\}_{i=1}^p, x' = \{x'_i\}_{i=1}^p \in \mathcal{X} \text{ where } x_i \leq x'_i \text{ for } 1 \leq i \leq p,\ T(x_1, ..., x_p) \leq T(x'_1, ..., x'_p)$$

We consider the following general optimization problem over monotonic decision tables.

Definition S2 (Optimization with monotonic decision tables)**.** Suppose f, g are functions that map a monotonic decision table T into a real-valued output. We consider the problem of finding an optimal monotonic decision table T_{opt} as defined below:

$$T_{opt} := \underset{T}{\operatorname{argmin}}\, f(T) \text{ subject to } g(T) \geq 0 \tag{S2}$$

In general, optimization over monotonic decision tables with the form given in Equation (S2) is difficult to solve. Although one could enumerate all possible monotonic decision tables and their corresponding $f(T), g(T)$, the number of enumerations is equal to $\mathcal{Y}^{|\mathcal{X}|} = \mathcal{Y}^{|\mathcal{X}_1| \times |\mathcal{X}_2| \times \cdots \times |\mathcal{X}_p|}$, which grows exponentially as the size of the decision table increases. In our application, we wish to simultaneously learn one two-way decision table and one three-way decision table, yielding a total of $5^{25} \times 5^{125} = 5^{150}$ enumerations. We would like to avoid enumerating these many possibilities.

Therefore, we use a Markov chain Monte Carlo (MCMC)-based stochastic algorithm for optimizing over monotonic decision tables by adopting ideas from the graph theory. Specifically, we represent a monotonic decision tables as an equivalent directed acyclic graph (DAG) where the directed edges indicates the monotonicity conditions, and optimize over monotonic decision tables by sampling the DAGs using an MCMC algorithm.

S6.2 Graph representation for monotonic decision tables

Monotonic decision tables can be equivalently represented as directed acyclic graphs (DAGs). We represent different inputs of decision tables as vertices of the DAG, and the monotonicity constraint on the decision table as directed edges in the graph. For example, Figure S15 shows the graph representation for the original two-way decision tables in the HES (Figure S1). The two-way decision table has $5 \times 5 = 25$ different inputs, which corresponds to the 25 vertices in the graph. The edges in the DAG indicate the monotonicity constraint that the output should satisfy according to the decision table. For example, in Figure S15, there is a directed edge from vertex $[1, 1]$ to vertex $[1, 2]$, which means that the output of the decision table for input $[1, 1]$ should be no greater than the output for input $[1, 2]$.

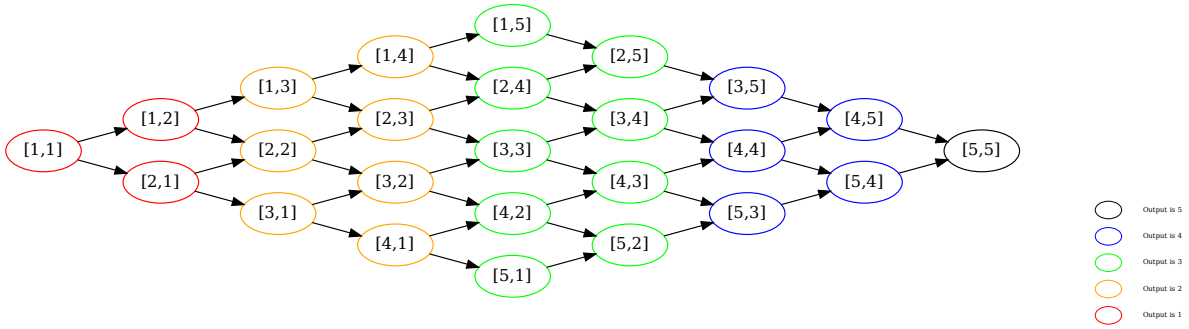


Figure S15: The DAG representation of the original 2-way decision table in the HES. Each vertice corresponds to an input of the 2-way table, and directed edges indicate the relative order the corresponding output should satisfy based on the monotonicity constraint. The color of the nodes indicate the output of the 2-way decision table in the original HES.

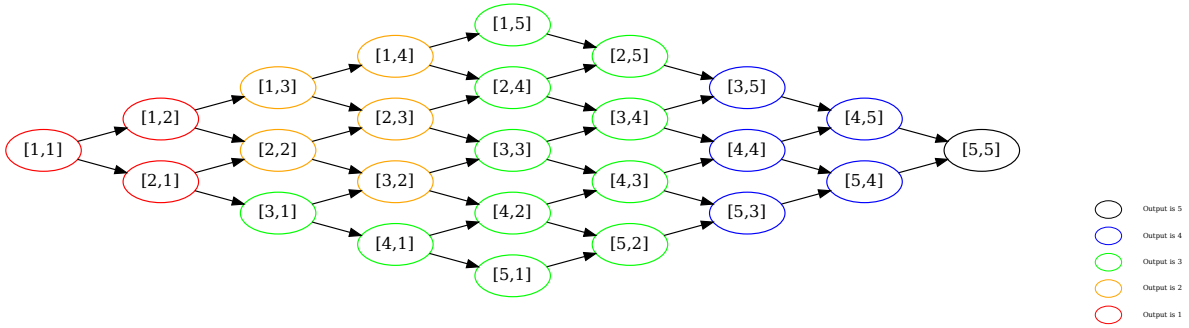


Figure S16: An example of partition for a decision table that does not satisfy the monotonicity constraint. There is a directed edge from $[3, 1]$ to $[3, 2]$, but the output for $[3, 1]$ is larger than the output for $[3, 2]$.

With this representation, finding a decision table from $\mathcal{X}$ to $\mathcal{Y}$ is equivalent to finding a graph partition that separate the DAG into $|\mathcal{Y}|$ areas where vertices in the same area have the same output scores. A monotonicity constraint on the decision table is equivalently translated to an *acyclic* constraint on the partition, where we forbid any directed edges from a node with larger output to a node with a smaller outcome (Herrmann et al., 2017, 2019).

Figure S15 shows the graph partition corresponding to the original two-way decision table in the HES, where the color of a node indicates the partition. This satisfies the monotonicity constraint. In contrast, Figure S16 shows a different partition that does not satisfy the monotonicity constraint. There is a directed edge from $[3, 1]$ to $[3, 2]$, but the output for $[3, 1]$ is larger than the output for $[3, 2]$.

S6.3 Optimization by sampling partitions of the DAGs

We optimize over monotonic decision tables by finding optimal acyclic partitions of the corresponding DAGs. We propose an MCMC-based stochastic algorithm for sampling acyclic partitions of the DAGs and optimize over it. To do this, we first sample a *topological sort* of a DAG (Karzanov and Khachiyan,

Algorithm 1: Stochastic Optimization Algorithm for Monotonic Decision Tables

Data: A DAG $\mathcal{G}$ that contains all the potential inputs of the decision table and relations from monotonicity constraints; a Markov chain $\mathcal{M}$ that generates an acyclic partial partition of $\mathcal{G}$ with a uniform stationary distribution; An initial monotonic decision table T_0 and corresponding state S_0 of $\mathcal{M}$.

for $0 \leq r \leq R$ **do**
  $S_{r0} \leftarrow S_r$
  $T_{r0} \leftarrow T_r$
  for $1 \leq k \leq K$ **do**
    $S_{rk} \leftarrow \mathcal{M}(S_{r(k-1)})$
    $T_{rk} \leftarrow$ Decision table that corresponds to S_{rk}
  end for
  $S_{r+1} \leftarrow \arg\min_{T \in \{T_{r0}, T_{r1}, \ldots, T_{rK}\}} f(T)$ subject to $g(T) \geq 0$
  $T_{r+1} \leftarrow$ Decision table corresponds to S_{r+1};
end for
Return T_{R+1}

1990) and then segment the topological sort into $|\mathcal{Y}|$ pieces to obtain a partition of the DAG.

Definition S3. (Topological sorts on DAGs) A topological sort of a directed acyclic graph is a linear ordering of its vertices such that for every directed edge uv from vertex u to vertex v, u comes before v in the ordering.

We take advantage of the fact that any segmentation of a topological sort corresponds to an acyclic partition and produces a decision table satisfying the monotonicity constraint. Conversely, for any decision table satisfying the monotonicity constraint, there will be a corresponding topological sort and segmentation that produces the decision table. We use the MCMC algorithm of Karzanov and Khachiyan (1990) to sample a topological sort from the DAG and a random walk to sample segmentations of the topological sorts. This gives us a Markov chain that generates random acyclic partitions. Finally, we use a *short-burst* algorithm to use the MCMC sampling algorithm for optimization (Cannon et al., 2023). Algorithm 1 formally describes our optimization procedure.

In the application, we run the above algorithm 2000 times to obtain 2000 corresponding decision tables. Then we choose the one that achieves the highest posterior expected value.

References

Ben-Michael, E., Greiner, D. J., Imai, K., and Jiang, Z. (2021). Safe policy learning through extrapolation: Application to pre-trial risk assessment. *arXiv preprint arXiv:2109.11679*.

Branson, Z., Rischard, M., Bornn, L., and Miratrix, L. W. (2019). A nonparametric bayesian methodology for regression discontinuity designs. *Journal of Statistical Planning and Inference*, 202:14–30.

Cannon, S., Goldbloom-Helzner, A., Gupta, V., Matthews, J., and Suwal, B. (2023). Voting rights, markov chains, and optimization by short bursts. *Methodology and Computing in Applied Probability*, 25(1):36.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2010). Bart: Bayesian additive regression trees.

Dell, M. and Querubin, P. (2018). Nation building through foreign intervention: Evidence from discontinuities in military strategies. *The Quarterly Journal of Economics*, 133(2):701–764.

Hahn, P. R., Murray, J. S., and Carvalho, C. M. (2020). Bayesian regression tree models for causal inference: Regularization, confounding, and heterogeneous effects (with discussion). *Bayesian Analysis*, 15(3):965–1056.

Herrmann, J., Kho, J., Uçar, B., Kaya, K., and Çatalyürek, Ü. V. (2017). Acyclic partitioning of large directed acyclic graphs. In *2017 17th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGRID)*, pages 371–380. IEEE.

Herrmann, J., Özkaya, M. Y., Uçar, B., Kaya, K., and Çatalyürek, U. V. (2019). Multilevel algorithms for acyclic partitioning of directed acyclic graphs. *SIAM Journal on Scientific Computing*, 41(4):A2117–A2145.

Karzanov, A. and Khachiyan, L. (1990). *On the conductance of order Markov chains*. Rutgers University, Department of Computer Science, Laboratory for Computer ....

Rasmussen, C. E. and Williams, C. K. I. (2006). Gaussian processes for machine learning. The MIT Press.

Taddy, M., Gardner, M., Chen, L., and Draper, D. (2016). A nonparametric bayesian analysis of heterogenous treatment effects in digital experimentation. *Journal of Business & Economic Statistics*, 34(4):661–672.