Matchings, Predictions and Counterfactual Harm in Refugee Resettlement Processes

Seungeon Lee[*] MARINEARCHON159@GMAIL.COM
Korea Advanced Institute of Science and Technology

Nina Corvelo-Benz NINACOBE@MPI-SWS.ORG
Suhas Thejaswi THEJASWI@MPI-SWS.ORG
Manuel Gomez-Rodriguez MANUEL@MPI-SWS.ORG
Max Planck Institute for Software Systems

Editors: Biwei Huang and Mathias Drton

Abstract

Resettlement agencies have started to adopt data-driven algorithmic matching to match refugees to locations using employment rate as a measure of utility. Given a pool of refugees, data-driven algorithmic matching utilizes a classifier to predict the probability that each refugee would find employment at any given location. Then, it uses the predicted probabilities to estimate the expected utility of all possible placement decisions. Finally, it finds the placement decisions that maximize the predicted utility by solving a maximum weight bipartite matching problem. In this work, we argue that, using existing solutions, there may be pools of refugees for which data-driven algorithmic matching is (counterfactually) harmful—it would have achieved lower utility than a given default policy used in the past, had it been used. Then, we develop a post-processing algorithm that, given placement decisions made by a default policy on a pool of refugees and the employment outcomes of the refugees in the pool, solves an inverse matching problem to minimally modify the predictions made by a given classifier. Under these modified predictions, the optimal matching policy that maximizes predicted utility on the pool is guaranteed to be not harmful. Further, we introduce a Transformer model that, given placement decisions made by a default policy on multiple pools of refugees and the employment outcomes of the refugees in these pools, learns to modify the predictions made by a given classifier so that the optimal matching policy that maximizes predicted utility under the modified predictions on an unseen pool of refugees is less likely to be harmful than under the original predictions. Experiments on simulated resettlement processes using synthetic refugee data created from a variety of publicly available data from international organizations, including the United Nations Refugee Agency (UNHCR), suggest that our methodology may be effective in making algorithmic placement decisions that are less likely to be harmful than existing solutions.

Keywords: Refuge Resettlement, Counterfactual Harm, Inverse Matching Problem

1. Introduction

In recent years, there is an increasing excitement in the potential of data-driven algorithmic matching to improve matching decisions in a wide variety of high-stakes application domains. Examples of such matching decisions include: matching refugees to locations (Bansak et al., 2018; Ahani et al., 2021, 2023; Freund et al., 2023); matching patients to appointments in health clinics (Salah and Srinivas, 2022); or matching blood/organ donations to recipients (McElfresh et al., 2023; Aziz et al.,

[*] The author contributed to this work during an internship at the Max Planck Institute for Software Systems.

2021). In all these cases, a central authority needs to distribute a limited set of resources—locations, appointments, or donations—among a group of individuals—refugees, patients, or recipients.

In this work, we focus on data-driven algorithmic matching in refugee resettlement processes, a specific application domain where matching decisions impact a particularly vulnerable group of individuals, data-driven algorithmic matching has been already deployed by a large resettlement agency in the United States[1], and may be soon deployed by others in the United States and elsewhere[2]. Given a pool of refugees, data-driven algorithmic matching aims to optimize the overall utility of placement decisions for these refugees. Because employment has been argued to play an important factor in the success of integration, e.g., by enabling self-sufficiency and ties to local residents (Ager and Strang, 2008), this utility is typically measured as the number of refugees that find employment soon after relocation. If we knew beforehand in which locations refugees would find employment, then we could find the optimal decision policy by solving a maximum weight bipartite matching problem (Tanimoto et al., 1978; Lau et al., 2011). In this matching problem, nodes would represent refugees as well as locations and edge weights would represent employment outcomes, *i.e.*, whether each refugee would (1) or would not (0) find employment at each location, and the goal would be to find a matching that maximizes the sum of edge weights in the matching. Unfortunately, at the time placement decisions are made, we cannot know for sure in which locations refugees would find employment—there is uncertainty on the value of the employment outcomes. To overcome this challenge, previous work on data-driven algorithmic matching leverages machine learning classifiers to predict the probability that a refugee would find employment at each location and then uses these predicted probabilities as edge weights in the above matching problem (Bansak et al., 2018; Ahani et al., 2021, 2023; Freund et al., 2023). As a result, given a pool of refugees, the resulting algorithmic decision policy is guaranteed to maximize predicted utility—the expected utility over the predicted employment probabilities.

However, previous work on data-driven algorithmic matching does not only utilize predicted utility to make algorithmic placement decisions but also to evaluate the quality of these decisions in comparison with decisions made by a default policy used in the past. In doing so, they implicitly assume that, for each location, the predicted employment probabilities are well-calibrated estimates of the true employment probabilities (Gneiting et al., 2007). Unfortunately, such an assumption is likely to be violated because every placement decision policy induces a different distribution of refugees across locations—it induces a different distribution shift (Quinonero-Candela et al., 2009). As a consequence, using existing solutions, any claim of superiority of algorithmic placement decisions over decisions made by default policy used in the past based on predicted utility is questionable. In fact, one cannot even tell to what extent algorithmic matching implements the principle of "first, do no harm", a principle that has been recently argued to be applicable to machine learning systems for decision support (Richens et al., 2022; Beckers et al., 2022; Li et al., 2023; Beckers et al., 2023).[3]

In this context, the most widely accepted definition of harm is arguably the counterfactual comparative account of harm (in short, counterfactual harm) (Feinberg, 1986; Hanser, 2008; Klocksiem, 2012). Under this definition, an action is harmful to an individual if they would have been in a worse state had the action been taken. Building upon this definition, in our work, we say that data-driven algorithmic matching is harmful to a pool of refugees if it would have worsened their employment

[1] https://www.refugees.ai/

[2] https://rematch-eu.org/about-matching/

[3] The European Unions' AI act mentions the term "harm" more than 35 times and points out that, its crucial role in the design of algorithmic systems must be defined carefully.

outcomes in comparison with a default decision policy used in the past, had it been used. Then, our goal is to minimize how frequently data-driven algorithmic matching causes harm.

Our Contributions. We start by formally characterizing resettlement processes in terms of a structural causal model (SCM) (Pearl, 2009), as illustrated in Figure 1. Using this characterization, we first show that, if the decisions made by an algorithmic decision policy satisfy a counterfactual condition with respect to the placement decisions made by a given default policy on a pool of refugees, then the algorithmic decision policy is not harmful to this pool—it would have achieved equal or higher utility than the default policy, had it been used. Building on this counterfactual condition, we make the following contributions:

(1) We develop an algorithm that, given placement decisions made by a default policy on a pool of refugees and the (corresponding) realized employment outcomes, solves an inverse bipartite matching problem to minimally modify the refugees' predicted employment probabilities provided by a given classifier. Under these modified probabilities, the placement decisions made by an algorithmic matching policy that maximizes the predicted utility provably satisfy the above counterfactual condition on the pool.

(2) Given placement decisions made by a default policy on multiple pools of refugees and the (corresponding) realized employment outcomes, we use the predicted employment probabilities provided by a classifier and the minimally modified predicted employment probabilities provided by the previously introduced algorithm to train a Transformer model. This model is able to minimally modify the predicted employment probabilities of an unseen pool so that it approximately satisfies the above counterfactual condition with respect to this pool.

Finally, we validate our methodological contributions on simulated resettlement processes using synthetic refugee data created using publicly available aggregated data from a variety of international organizations, including the United Nations Refugee Agency (UNHCR). The results show that our methodology may be effective in making algorithmic placement decisions that are less likely to be harmful than existing solutions. [4]

Further related work. Our work builds upon further related work on causal inference, bipartite matching, and critiques of prediction optimization.

Standard causal inference techniques have been widely applied to address distribution shift and causal inference in various domains. Methods such as instrumental variables (Angrist et al., 1996), inverse propensity scoring (IPS) (Robins et al., 1994), and doubly robust (DR) estimation (Dudík et al., 2011) are representative examples. However, a direct application of existing techniques to data-driven algorithmic matching presents challenges. For example, identifying valid instrumental variables is difficult in refugee resettlement because the available data may lack sufficient additional information to satisfy the necessary conditions for these instruments. Similarly, IPS and DR methods require estimating the distribution shift induced by the default policy across all potential matches within each pool—a task that becomes computationally infeasible due to the vast number of possible assignments.

Bipartite matching problems, or more generally assignment problems, have multifaceted applications in various domains and thus a rich, extensive literature (Gale and Shapley, 1962; Crawford and Knoer, 1981; Lovász and Plummer, 2009; Gibbons, 1985; Stelmakh et al., 2021; McElfresh et al., 2023; Aziz et al., 2021). Less common and not well known are inverse assignment problems (De-

[4] The code and data used in our experiments are available at https://github.com/Networks-Learning/counterfactually-harmless-matching.

mange and Monnot, 2014; Heuberger, 2004; Lee and Terekhov, 2020). Whereas assignment problems aim to find an optimal assignment that maximizes a desired parameterized objective function, inverse assignment problems aim to find the minimal change to those parameters such that a desired assignment becomes an optimal assignment (Liu and Zhang, 2003; Bérczi et al., 2023). Inverse assignment problems have been recently proven useful in the context of counterfactual explanations (Korikov et al., 2021; Korikov and Beck, 2021). In particular, finding the nearest counterfactual explanation for algorithmic assignments has been reduced to an inverse problem. Within the literature on inverse assignment problems, the work most closely related to ours is by Yang and Zhang (2007), which introduces an algorithmic framework to solve partial inverse assignment problems where the desired assignment is only partially given. More specifically, in our first methodological contribution, we adapt their framework to minimally modify the refugees' predicted employment probabilities by a given classifier (in their framework, the parameters) so that, under these modified probabilities, the optimal placement decisions (the optimal assignment) match a subset of those made by a default policy (the desired partial assignment).

Prediction optimization has been recently put into question in the context of decision support (van Amsterdam et al., 2023; Wang et al., 2024; Liu et al., 2024). More specifically, it has been argued that optimizing decision support systems to improve prediction accuracy does not always translate to better decision-making. Our work aligns with this critique since we argue that, by maximizing predictive utility, one cannot rule out the possibility that data-driven algorithmic matching causes harm.

2. A Causal Model for Refugee Resettlement Processes

We consider a resettlement process where, for each realization of the process, a decision maker receives a pool $\mathcal{I}$ of n refugees with features $\boldsymbol{x} = (x_i)_{i\in\mathcal{I}} \in \mathcal{X}^n$, matches each refugee $i \in \mathcal{I}$ to a location $\boldsymbol{l} = (l_i)_{i\in\mathcal{I}} \in \mathcal{L}^n$ out of k locations, and eventually receives a utility $u(\boldsymbol{y}) \geq 0$, where $\boldsymbol{y} = (y_i)_{i\in\mathcal{I}} \in \{0,1\}^n$ are outcome variables specifying whether a refugee i finds a job ($y_i = 1$) or does not find a job ($y_i = 0$) soon after relocation. Here, we assume that each location $l \in \mathcal{L}$ has a maximum capacity c_l to host refugees and the utility $u(\boldsymbol{y}) = \sum_i u(y_i)$ is separable. Without loss of generality, we further assume that $u(y_i) = y_i$[5].

Next, we characterize the matching process using a structural causal model (SCM) (Pearl, 2009), which we denote as $\mathcal{M}$. The SCM $\mathcal{M}$, which entails a distribution $P^{\mathcal{M}}$, is defined by the following set of assignments[6]:

$$X_i = f_X(D_i), \quad Y_i = f_Y(D_i, V_{i,L_i}) \quad \forall i \in \mathcal{I}, \quad \boldsymbol{L} = \tilde{\pi}(\boldsymbol{X}, W) \quad \text{and} \quad U = \mathbf{1}^T \boldsymbol{Y}, \tag{1}$$

where $D_i \sim P(D)$, $V_{i,l} \sim P(V \mid L = l)$ and $W \sim P(W)$ are independent exogenous random variables, often called exogenous noise variables, which characterize the refugee's individual characteristics[7], the synergies between locations and refugees, and the decision maker's individual characteristics, respectively. Further, we have that f_X and f_Y are unknown causal functions, and $\tilde{\pi}$ is

[5] We note that our approach is agnostic to the choice of utility function.

[6] Random variables are denoted with capital letters and realizations of random variables with lower case letters.

[7] To allow for features X_i that are causal and anticausal to the outcome variable Y_i, the noise variable D_i is a parent of both X_i and Y_i, as discussed elsewhere (Schölkopf et al., 2012). If we allow only for causal features and there are no hidden confounders, we could just write $Y_i = f_Y(X_i, V_{i,L_i})$.

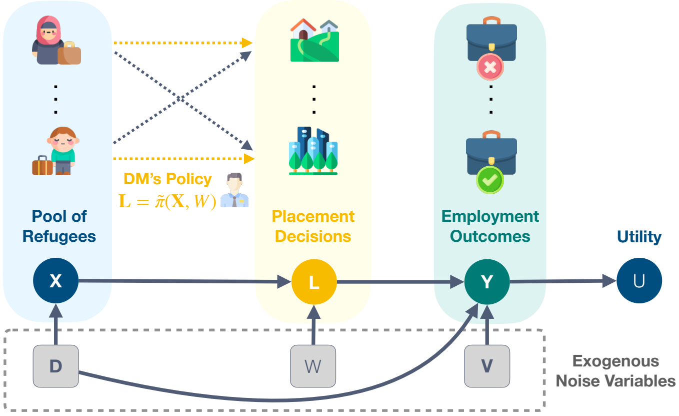


Figure 1: Our structural causal model $\mathcal{M}$. Circles represent endogenous random variables and boxes represent exogenous random variables. The value of each endogenous variable is given by a function of the values of its ancestors in the structural causal model, as defined by Eq. 1. The value of each exogenous variable is sampled independently from a given distribution.

the decision maker's default matching policy (in short, the default policy)[8]. Figure 1 shows a visual representation of our SCM $\mathcal{M}$.

Moreover, we assume the decision maker has access to a classifier $g : \mathcal{L} \times \mathcal{X} \rightarrow [0,1]^{|\mathcal{L}|}$ that, for each location $l \in \mathcal{L}$, maps a refugee's feature vector $x \in \mathcal{X}$ to a predicted employment probability $g_l(x)$. Here, the higher the predicted employment probability $g_l(x)$, the more the classifier g believes the candidate will find a job at location l, and we denote the predicted probabilities for all refugees in the pool $\mathcal{I}$ as $\boldsymbol{g} = (g_l(x_i))_{i \in \mathcal{I}, l \in \mathcal{L}}$. Further, let $\Pi(\boldsymbol{G})$ be the class of algorithmic matching policies that assign refugees to locations based on the predicted employment probabilities $\boldsymbol{G}$ and satisfy the capacity constraints. Then, we can view the implementation of an algorithmic matching policy $\pi(\boldsymbol{G}) \in \Pi(\boldsymbol{G})$ as an intervention $\text{do}(\boldsymbol{L} = \pi(\boldsymbol{G}))$ in the SCM $\mathcal{M}$. The intervened SCM $\mathcal{M}^{\text{do}(\boldsymbol{L}=\pi(\boldsymbol{G}))}$, which entails a distribution $P^{\mathcal{M}\,;\,\text{do}(\boldsymbol{L}=\pi(\boldsymbol{G}))}$, is defined by:

$$X_i = f_X(D_i),\ Y_i = f_Y(D_i, V_{i,L_i}),\ G_{i,l} = g_l(X_i)\ \forall i \in \mathcal{I}, \forall l \in \mathcal{L},\ \boldsymbol{L} = \pi(\boldsymbol{G}) \text{ and } U = \boldsymbol{1}^T \boldsymbol{Y}. \tag{2}$$

In addition, given placement decisions $\boldsymbol{l}$ made by a default policy $\tilde{\pi}$ on a pool of refugees with features $\boldsymbol{x}$ and the (corresponding) realized employment outcomes $\boldsymbol{y}$, we define a counterfactual SCM $\mathcal{M}_{\boldsymbol{X}=\boldsymbol{x},\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{L}=\boldsymbol{l}}$ where the noise variables are distributed according to the posterior distribution $P(\boldsymbol{D}, \boldsymbol{V}, W \,|\, \boldsymbol{X} = \boldsymbol{x}, \boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{L} = \boldsymbol{l})$ and we denote the resulting distribution entailed by this SCM using $P^{\mathcal{M}\,|\,\boldsymbol{X}=\boldsymbol{x},\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{L}=\boldsymbol{l}}$. Finally, we characterize counterfactual statements comprising the algorithmic policy $\pi(\boldsymbol{G})$ using the intervened counterfactual SCM $\mathcal{M}^{\text{do}(\boldsymbol{L}=\pi(\boldsymbol{g}))}_{\boldsymbol{X}=\boldsymbol{x},\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{L}=\boldsymbol{l}}$ and denote its entailed distribution using $P^{\mathcal{M}\,|\,\boldsymbol{X}=\boldsymbol{x},\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{L}=\boldsymbol{l}\,;\,\text{do}(\boldsymbol{L}=\pi(\boldsymbol{g}))}$.

In the next section, we will use the above characterization to reason about the conditions the predicted employment probabilities $\boldsymbol{G}$ should satisfy so that the optimal matching policy that maximizes predicted utility also maximizes the expected utility $\mathbb{E}[U \,|\, \boldsymbol{X}]$.

[8] For ease of presentation, we assume a constant pool size n and capacities c_l. However, all theoretical results and algorithms can be easily adapted to settings where the pool size and capacities change across matching processes.

3. On the Optimality of Algorithmic Matching

Given a pool $\mathcal{I}$ of n refugees with features $\boldsymbol{x}$, existing approaches to data-driven algorithmic matching equate the problem of finding an optimal algorithmic matching policy that maximizes (conditional) expected utility, *i.e.*,

$$\pi^*(\boldsymbol{g}) \in \Pi^*(\boldsymbol{g}) = \underset{\pi(\boldsymbol{g}) \in \Pi(\boldsymbol{g})}{\operatorname{argmax}} \; \mathbb{E}_{\boldsymbol{Y} \sim P^{\mathcal{M}\,;\,\text{do}(\boldsymbol{L}=\pi(\boldsymbol{g}))}}[\mathbf{1}^T \boldsymbol{Y} \mid \boldsymbol{X} = \boldsymbol{x}], \tag{3}$$

to the much easier problem of finding an algorithmic matching policy that maximizes (conditional) predicted utility, *i.e.*,

$$\hat{\pi}(\boldsymbol{g}) \in \hat{\Pi}(\boldsymbol{g}) = \underset{\pi(\boldsymbol{g}) \in \Pi(\boldsymbol{g})}{\operatorname{argmax}} \sum_{i \in \mathcal{I}} g_{\pi_i(\boldsymbol{g})}(x_i), \tag{4}$$

where $\pi_i(\boldsymbol{g}) \in \mathcal{L}$ denotes the location assigned by policy $\pi(\boldsymbol{g})$ to refugee $i \in \mathcal{I}$. This latter problem can be viewed as a maximum weight bipartite matching problem (Tanimoto et al., 1978; Lau et al., 2011) and thus its solution, which may be non unique, can be recovered from the solution to the following linear program:

$$\begin{aligned}
\text{maximize} \quad & \sum_{i \in \mathcal{I}, l \in \mathcal{L}} g_l(x_i)\, z_{il} \\
\text{subject to} \quad & \sum_{l \in \mathcal{L}} z_{il} \leq 1 \quad \forall i \in \mathcal{I}, \\
& \sum_{i \in \mathcal{I}} z_{il} \leq c_l \quad \forall l \in \mathcal{L}, \\
& z_{il} \in [0,1] \quad \forall i \in \mathcal{I}, l \in \mathcal{L}.
\end{aligned} \tag{5}$$

In particular, it holds that $\hat{\pi}(\boldsymbol{g}) = (\operatorname{argmax}_{l \in \mathcal{L}} \hat{z}_{il})_{i \in \mathcal{I}}$, where $\hat{\boldsymbol{z}} = (\hat{z}_{il})_{i \in \mathcal{I}, l \in \mathcal{L}}$ is an optimal integral solution to the above linear program, as shown elsewhere (Bansak et al., 2018; Ahani et al., 2021). However, existing approaches do not really investigate the sufficient conditions under which the predicted employment probabilities $\boldsymbol{g}$ should satisfy for both policies $\pi^*(\boldsymbol{g})$ and $\hat{\pi}(\boldsymbol{g})$ to offer the same expected utility.

The following proposition shows that, if the predicted employment probabilities $\boldsymbol{g}$ are perfectly calibrated, then policies in $\hat{\Pi}(\boldsymbol{g})$ and $\Pi^*(\boldsymbol{g})$ must offer the same expected utility.[9]

Proposition 1 *For any $\boldsymbol{x} \sim P^{\mathcal{M}}(\boldsymbol{X})$, if $g_l(x_i) = P^{\mathcal{M}\,;\,do(L_i=l)}(Y_i = 1 \mid X_i = x_i)$ for all $l \in \mathcal{L}$ and $i \in \mathcal{I}$, then, for any $\hat{\pi}(\boldsymbol{g}) \in \hat{\Pi}(\boldsymbol{g})$, it holds that $\hat{\pi}(\boldsymbol{g}) \in \Pi^*(\boldsymbol{g})$.*

Unfortunately, the above condition is unlikely to hold in practice. This is due to the fact that, if the distribution of conditional probability values $P^{\mathcal{M}\,;\,do(\boldsymbol{L}=\boldsymbol{l})}(Y_i = 1 \mid X_i)$ induced by $P^{\mathcal{M}}(X_i)$ is nonatomic, even if we are able to sample data from $P^{\mathcal{M}\,;\,do(\boldsymbol{L}=\boldsymbol{l})}$, finding the individual probabilities $P^{\mathcal{M}\,;\,do(L_i=l)}(Y_i = 1 \mid X_i = x_i)$ from this data is not possible without distributional assumptions, even asymptotically (Barber, 2020; Gupta et al., 2020). Consequently, we cannot say whether the use of existing approaches to data-driven algorithmic matching over a given default policy for a particular pool of refugees will truly increase utility or cause potential harm. With this in mind, over the next two sections, we introduce an alternative approach to data-driven algorithmic matching with the goal of reducing harmful placement decisions with respect to the default policy.

[9] All proofs can be found in Appendix A.

4. Algorithmic Matching with Counterfactual Guarantees

Given the placement decisions $\boldsymbol{l}$ made by a default policy $\tilde{\pi}$ on a pool of refugees $\mathcal{I}$ with features $\boldsymbol{x}$ and the (corresponding) realized employment outcomes $\boldsymbol{y}$, we first identify a class of algorithmic decision policies that are counterfactually harmless—they would have achieved at least the same utility as the default policy, had they been used:

Proposition 2 *Given $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{l} \sim P^{\mathcal{M}}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{L})$, let the class of algorithmic decision policies $\Pi_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{l}}$ be defined as*

$$\Pi_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{l}} = \{\pi(\boldsymbol{g}) \,|\, \pi(\boldsymbol{g}) \in \Pi(\boldsymbol{g}) \wedge \pi_i(\boldsymbol{g}) = l_i \ \forall i \in \mathcal{I} \text{ such that } y_i = 1\}. \tag{6}$$

Then, for any $\pi(\boldsymbol{g}) \in \Pi_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{l}}$, it holds that

$$\mathbf{1}^T \boldsymbol{Y} \geq \mathbf{1}^T \boldsymbol{y} \quad \text{for all} \quad \boldsymbol{Y} \sim P^{\mathcal{M} \,|\, \boldsymbol{X}=\boldsymbol{x}, \boldsymbol{Y}=\boldsymbol{y}, \boldsymbol{L}=\boldsymbol{l} \,;\, do(\boldsymbol{L}=\pi(\boldsymbol{g}))}(\boldsymbol{Y}), \tag{7}$$

where $P^{\mathcal{M} \,|\, \boldsymbol{X}=\boldsymbol{x}, \boldsymbol{Y}=\boldsymbol{y}, \boldsymbol{L}=\boldsymbol{l} \,;\, do(\boldsymbol{L}=\pi(\boldsymbol{g}))}(\boldsymbol{Y})$ denotes the intervened counterfactual distribution of the employment outcomes $\boldsymbol{Y}$ entailed by the counterfactual SCM $\mathcal{M}^{do(\boldsymbol{L}=\pi(\boldsymbol{g}))}_{\boldsymbol{X}=\boldsymbol{x}, \boldsymbol{Y}=\boldsymbol{y}, \boldsymbol{L}=\boldsymbol{l}}$.

Here, it is important to note that the expected utility achieved by any given algorithmic policy with respect to all possible pools of refugees can be rewritten as an average over counterfactual utilities, as formalized by the following proposition:

Proposition 3 *For any resettlement process $\mathcal{M}$ satisfying Eq. 1 and algorithmic policy $\pi \in \Pi(\boldsymbol{G})$, the following equality holds:*

$$\begin{aligned}&\mathbb{E}_{\boldsymbol{Y} \sim P^{\mathcal{M} \,;\, do(\boldsymbol{L}=\pi(\boldsymbol{G}))}(\boldsymbol{Y})}[\mathbf{1}^T \boldsymbol{Y}] \\ &\qquad = \mathbb{E}_{\boldsymbol{X}', \boldsymbol{Y}', \boldsymbol{L}' \sim P^{\mathcal{M}}(\boldsymbol{X}', \boldsymbol{Y}', \boldsymbol{L}')}\left[\mathbb{E}_{\boldsymbol{Y} \sim P^{\mathcal{M} \,|\, \boldsymbol{X}=\boldsymbol{X}', \boldsymbol{Y}=\boldsymbol{Y}', \boldsymbol{L}=\boldsymbol{L}' \,;\, do(\boldsymbol{L}=\pi(\boldsymbol{G}))}(\boldsymbol{Y})}[\mathbf{1}^T \boldsymbol{Y}]\right]\end{aligned} \tag{8}$$

Then, from Propositions 2 and 3, we can immediately conclude that, if an algorithmic policy π is counterfactually harmless for *any* pool of refugees $\mathcal{I}$, then, π will achieve equal or higher expected utility than the default policy $\tilde{\pi}$, *i.e.*, if $\pi(\boldsymbol{g}) \in \Pi_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{l}}$ for any $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{l} \sim P^{\mathcal{M}}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{L})$, then, it holds that

$$\mathbb{E}_{\boldsymbol{Y} \sim P^{\mathcal{M} \,;\, \mathrm{do}(\boldsymbol{L}=\pi(\boldsymbol{G}))}(\boldsymbol{Y})}[\mathbf{1}^T \boldsymbol{Y}] \geq \mathbb{E}_{\boldsymbol{Y} \sim P^{\mathcal{M}}(\boldsymbol{Y})}[\mathbf{1}^T \boldsymbol{Y}]. \tag{9}$$

In the next section, we will build upon the above theoretical results to develop a practical post-processing framework to minimally modify the predicted employment probabilities $\boldsymbol{g} = g(\boldsymbol{x})$ provided by a given classifier g so that the algorithmic matching policy $\hat{\pi}(\breve{\boldsymbol{g}})$ that maximizes (conditional) predicted utility under the minimally modified probabilities $\breve{\boldsymbol{g}}$ is less likely to be counterfactually harmful than the algorithmic matching policy $\hat{\pi}(\boldsymbol{g})$ that maximizes (conditional) predicted utility under the original probabilities $\boldsymbol{g}$.

5. A Practical Post-Processing Framework

Given retrospective data about multiple pools of refugees under the default policy $\tilde{\pi}$, our framework first finds, for each of the above pools, the minimally modified predicted probabilities $\breve{\boldsymbol{g}}$ under which any $\hat{\pi}(\breve{\boldsymbol{g}}) \in \hat{\Pi}(\breve{\boldsymbol{g}})$ is provably counterfactually harmless with respect to $\tilde{\pi}$, in hindsight. Then, it uses the original and modified predicted probabilities of all the above pools to train a transformer model h that, given the original predicted probabilities $\boldsymbol{g}$ of an *unseen* pool $\mathcal{I}$, predicts the minimally modified predicted probabilities $\breve{\boldsymbol{g}}$.[10] Figure 2 illustrates our overall post-processing framework.

[10] Our framework allows for the modified predicted probabilities $\breve{\boldsymbol{g}}$ to be larger than 1.

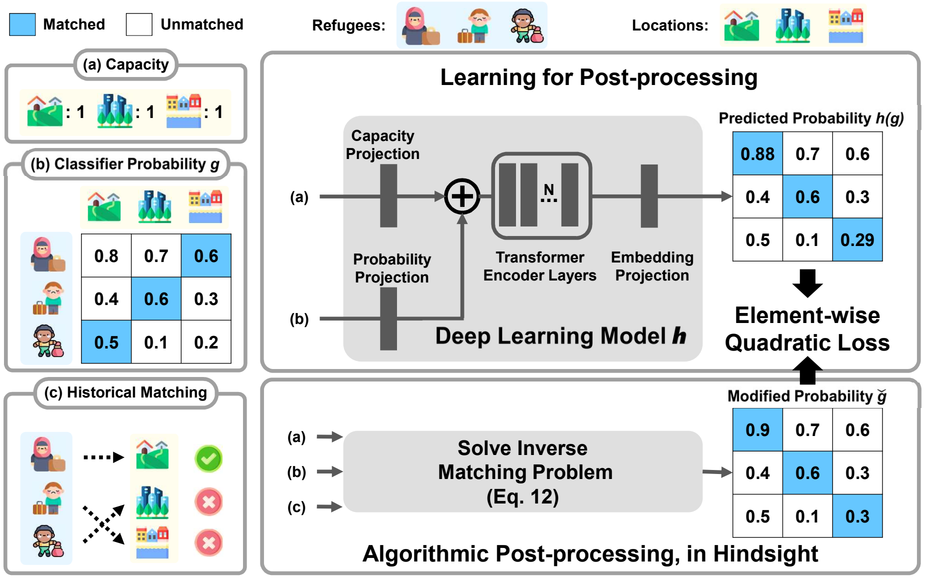


Figure 2: Illustration of our post-processing framework. The matrices indicate the scores used in algorithmic matching, while blue cells in score matrix denotes the matching result of maximum weight matching. In this illustrative example, ϵ is set to 0.1.

Avoiding Counterfactual Harm, In Hindsight. Given $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{l} \sim P^{\mathcal{M}}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{L})$, we formulate the problem of finding the minimally modified predicted employment probabilities $\breve{\boldsymbol{g}}$ under which any $\hat{\pi}(\breve{\boldsymbol{g}}) \in \hat{\Pi}(\breve{\boldsymbol{g}})$ is provably counterfactually harmless with respect to $\tilde{\pi}$ as follows:

$$\begin{aligned} &\text{minimize} \quad && ||\breve{\boldsymbol{g}} - \boldsymbol{g}||_1 = \sum_{i \in \mathcal{I}, l \in \mathcal{L}} |\breve{g}_l(x_i) - g_l(x_i)|, \\ &\text{subject to} \quad && \hat{\Pi}(\breve{\boldsymbol{g}}) \subseteq \Pi_{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{l}}. \end{aligned} \tag{10}$$

To solve the above problem, we build on the algorithmic framework of Yang and Zhang (2007). Let $\pi'(\boldsymbol{g}) \in \Pi_{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{l}}$ be a policy that maximizes the (conditional) predicted utility over the refugees $i \in \mathcal{I}' = \{i \in \mathcal{I} \,|\, l_i = 0\}$ and note that, for any given $\boldsymbol{g}$, $\pi'(\boldsymbol{g})$ can be recovered from the solution to the following linear program, similarly as in Eqs. 4 and 5:

$$\begin{aligned} &\text{maximize} \quad && \sum_{i \in \mathcal{I}', l \in \mathcal{L}} g_l(x_i)\, z_{il} \\ &\text{subject to} \quad && \sum_{l \in \mathcal{L}} z_{il} \leq 1 \quad \forall i \in \mathcal{I}', \\ & && \sum_{i \in \mathcal{I}'} z_{il} \leq c'_l \quad \forall l \in \mathcal{L}, \\ & && z_{il} \in \{0, 1\} \quad \forall i \in \mathcal{I}', l \in \mathcal{L}, \end{aligned} \tag{11}$$

where $c'_l = c_l - |\{i \in \mathcal{I} \,|\, l_i = l \wedge y_i = 1\}|$. In particular, we have that $\pi'_i(\boldsymbol{g}) = \text{argmax}_{l \in \mathcal{L}}\, z'_{il}$ if $i \in \mathcal{I}'$, where $\boldsymbol{z}' = (z'_{il})_{i \in \mathcal{I}', l \in \mathcal{L}}$ is an optimal integral solution to the above linear program, and $\pi'_i(\boldsymbol{g}) = l_i$ otherwise.

Then, using the above policy $\pi'(\boldsymbol{g})$ and the optimal solution to the dual of the linear program defined by Eq. 5, we find the minimally modified predicted employment probabilities $\breve{\boldsymbol{g}}$, as formalized by the following Theorem:

Theorem 4 *Given $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{l} \sim P^{\mathcal{M}}(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{L})$, the minimally modified employment probabilities $\breve{\boldsymbol{g}}$ under which any algorithmic policy $\hat{\pi}(\breve{\boldsymbol{g}}) \in \hat{\Pi}(\breve{\boldsymbol{g}})$ is provably counterfactually harmless with respect to the default policy $\tilde{\pi}$, as defined in Eq. 10, is given by:*

$$\breve{g}_l(x_i) = \begin{cases} \hat{u}_i + \hat{v}_l + \epsilon & \textit{if } \pi'_i(\boldsymbol{g}) = l \\ g_l(x_i) & \textit{otherwise}, \end{cases} \tag{12}$$

where $\epsilon > 0$ is an infinitesimally small constant and $\hat{\boldsymbol{u}} = (\hat{u}_i)_{i \in \mathcal{I}}$ and $\hat{\boldsymbol{v}} = (\hat{v}_l)_{l \in \mathcal{L}}$ are the optimal solution to the dual of the linear program defined by Eq. 5 under predicted probabilities $\boldsymbol{g}$, i.e.*,*

$$\begin{aligned} \textit{minimize} \quad & \sum_{i \in \mathcal{I}} u_i + \sum_{l \in \mathcal{L}} c_l v_l, \\ \textit{subject to} \quad & u_i + v_l \geq g_l(x_i) \quad \forall i \in \mathcal{I}, l \in \mathcal{L}, \\ & u_i \geq 0, v_l \geq 0 \quad \forall i \in \mathcal{I}, l \in \mathcal{L}. \end{aligned} \tag{13}$$

Learning to Avoid Counterfactual Harm. In the previous section, we have derived an expression to compute the minimally modified employment probabilities $\breve{\boldsymbol{g}}$ under which any $\hat{\pi}(\breve{\boldsymbol{g}}) \in \hat{\Pi}(\breve{\boldsymbol{g}})$ is provably counterfactually harmless with respect to the default policy $\tilde{\pi}$. Unfortunately, the expression requires retrospective data about the pool of interest, *i.e.*, the placement decisions $\boldsymbol{l}$ made by a default policy $\tilde{\pi}$ and the (corresponding) realized employment outcomes $\boldsymbol{y}$. As a consequence, we cannot directly use it to compute the minimally modified employment probabilities $\breve{\boldsymbol{g}}$ of unseen pools.

However, we can use the above expression on retrospective data from multiple pools of refugees $\{\mathcal{I}_j\}$ to train a deep learning model $h : [0,1]^{k \times n} \to \mathbb{R}_+^{k \times n}$ that, given the original predicted probabilities $\boldsymbol{g}$ of an unseen pool $\mathcal{I}$, approximately predicts the minimally modified predicted probabilities $h(\boldsymbol{g}) \approx \breve{\boldsymbol{g}}$. Our model consists of four modules: (i) a prediction probability projection layer, (ii) a capacity projection layer, (iii) N layers of Transformer encoders, and (iv) an embedding projection layer. The projection layers comprise two linear projections with ReLU activation. The layers of Transformer encoders exclude positional encoding so that they become agnostic to the number or order of the refugees in each pool. Moreover, in our experiments, we train our model using a quadratic loss, *i.e.*, $\mathbb{E}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{l} \sim P^{\mathcal{M}}(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{L})} \left[\sum_{i \in \mathcal{I}, l \in \mathcal{L}} \left(h_{l,i}(\boldsymbol{g}) - \breve{g}_l(x_i) \right)^2 \right]$, where $h_{l,i}(\boldsymbol{g})$ denotes the approximately minimally modified predicted probability for $l \in \mathcal{L}$ and $i \in \mathcal{I}$. For further implementation details used in our experiments, refer to Appendix C.

6. Experiments

In this section, we use publicly available aggregated data from a variety of international organizations, including the United Nations Refugee Agency (UNHCR), to generate synthetic refugee data. Then, we use this synthetic data to simulate and compare the outcome of multiple resettlement processes under a default policy and several algorithmic policies. Here, note that, by using synthetic refugee data, rather than real refugee data, we can calculate the (true) expected utility achieved by any

Noise Level (w)	0	0.125	0.25	0.375	0.5	0.625	0.75	0.875	1
$\hat{\pi}(\boldsymbol{p})$	0	33.2	23.6	17.6	13.6	11.4	9.2	8.2	6
$\hat{\pi}(\boldsymbol{g})$	75	64.6	55	48.2	41	33.4	24.8	23.4	15.6
$\hat{\pi}(h(\boldsymbol{g}))$	**69.2**	**58.9**	**45.4**	**38.0**	**33.9**	**30.6**	**22.9**	23.4	16.0

Table 1: Percentage (%) of pools counterfactually harmed by the algorithmic policies $\hat{\pi}(\boldsymbol{p})$, $\hat{\pi}(\boldsymbol{g})$, and $\hat{\pi}(h(\boldsymbol{g}))$ in the test set under various noise levels w and $\beta = 0.6$. Lower numbers indicate better performance. The policy $\hat{\pi}(\breve{\boldsymbol{g}})$ consistently achieves 0% harm, as expected from Eq. 9, and thus it is not presented. Bold numbers indicate that $\hat{\pi}(h(\boldsymbol{g}))$ counterfactually harms fewer pools than $\hat{\pi}(\boldsymbol{g})$, with results averaged over 5 runs. More quantitative results, including standard deviations, are available in Appendix D.

algorithmic policy and, given retrospective data under a default policy, we can calculate the (true) counterfactual utility achieved by any algorithmic policy.[11]

Experimental Settings. We create 5,000 synthetic pools of refugees to be resettled to $k = 10$ locations. Each pool $\mathcal{I}$ contains $n = 100$ synthetic refugees, and each refugee $i \in \mathcal{I}$ is represented by 4-dimensional feature vector x_i and $k = 10$ labels $\{y_i(l)\}_{l\in\mathcal{L}}$ indicating whether the refugee would find employment (or not) at each state l soon after relocation. Each location corresponds to one US State and the features contain demographic information about a refugee's age, country of origin, sex, and educational attainment. The features and labels are sampled from distributions $P^{\mathcal{M}}(X)$ and $P^{\mathcal{M}\,;\,\text{do}(L_i=l)}(Y \mid X = x_i)$ informed by aggregate statistics from UNHCR, the World Bank, the U.S. Census, the U.S. Bureau of Labor Statistics, and Migration Policy Institute. Refer to Appendix B for more details on the synthetic data generation process. The classifier g overestimates the value of the true employment probability for half of the locations, picked at random, and underestimates its value for the remaining half. Whenever g overestimates the employment probability, it predicts $g_l(x) = p \cdot (1 + \beta)$ and, whenever g underestimates the employment probability, it predicts $g_l(x) = p \cdot (1 - \beta)$, where $p = P^{\mathcal{M}\,;\,\text{do}(L_i=l)}(Y_i = 1 \mid X_i = x_i)$ and β is a given parameter.

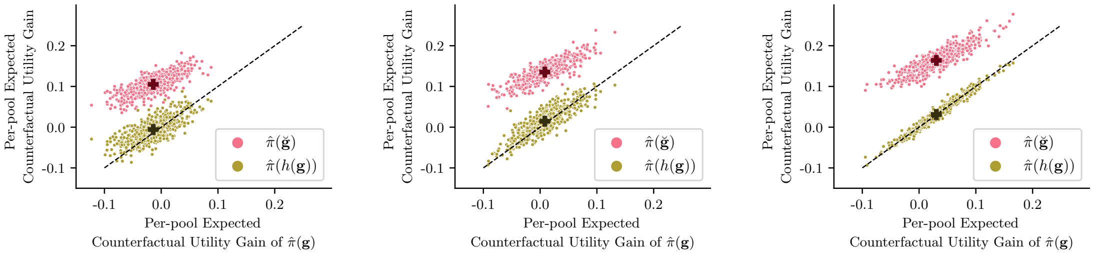
In our experiments, we randomly split the pools into a training set with 4,000 pools, which we use to train the deep learning model h, a validation set with 500 pools, which we use for searching the best model across multiple epochs as well as the best value of ϵ for each default policy $\tilde{\pi}(\boldsymbol{x}, w)$, and a test set with 500 pools, which we use for evaluation[12]. Moreover, we implement and compare the performance of the following policies:

(i) A default policy $\tilde{\pi}(\boldsymbol{x}, w)$ that first finds the placement decisions that maximize the (conditional) expected utility under the true employment probabilities $\boldsymbol{p}$ and then picks a ratio w of these placement decisions and shuffles them with each other.

(ii) An algorithmic policy $\hat{\pi}(\boldsymbol{p})$ that makes placement decisions that maximize the (conditional) expected utility under the true employment probabilities $\boldsymbol{p}$. This policy is unrealizable in practice, as discussed in Section 3. Here, note that $\hat{\pi}(\boldsymbol{p}) = \tilde{\pi}(\boldsymbol{x}, 0)$.

[11] We are not aware of any publicly available dataset with real refugee data. Unfortunately, we were unable to obtain access to real refugee data used in previous studies (Bansak et al., 2018; Ahani et al., 2021, 2023; Freund et al., 2023).

[12] Refer to Appendix C for additional implementation details regarding the deep learning model h.



(a) Low noise level

(b) Medium noise level

(c) High noise level

Figure 3: Per-pool expected counterfactual utility gain achieved by the proposed algorithmic policies $\hat{\pi}(\breve{\boldsymbol{g}})$ and $\hat{\pi}(h(\boldsymbol{g}))$ with respect to the policy $\hat{\pi}(\boldsymbol{g})$ in the test set for $\beta = 0.6$ under low ($w = 0.125$), medium ($w = 0.5$) and high ($w = 0.875$) noise level. The cross markers indicate the expected counterfactual utility gain across all pools in the test set. Pools above the (dashed) identity line (*i.e.*, $y = x$) show increase in counterfactual utility gain compared to policy $\hat{\pi}(\boldsymbol{g})$.

(iii) An algorithmic policy $\hat{\pi}(\boldsymbol{g})$ that makes placement decisions that maximize the (conditional) predicted utility under the predicted employment probabilities $\boldsymbol{g}$.

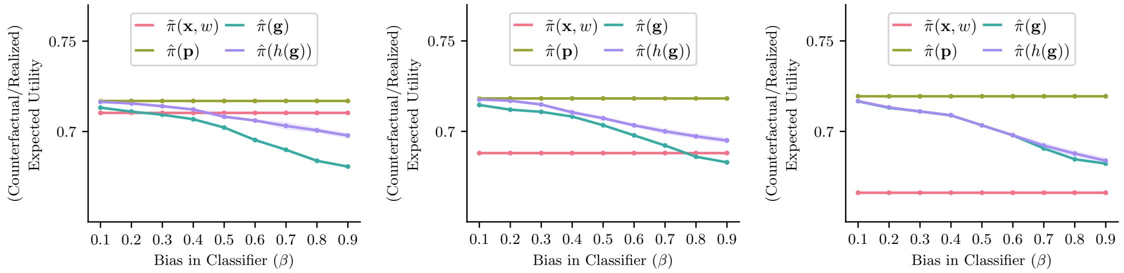
(iv) An algorithmic policy $\hat{\pi}(\breve{\boldsymbol{g}})$ that makes placement decisions that maximize the (conditional) predicted utility under the modified predicted employment probabilities $\breve{\boldsymbol{g}}$ given by Eq. 12. This policy is unrealizable in practice since it uses retrospective data about the pool of interest.

(v) An algorithmic policy $\hat{\pi}(h(\boldsymbol{g}))$ that makes placement decisions that maximize the (conditional) predicted utility under the postprocessed predicted employment probabilities $h(\boldsymbol{g})$, where h is the deep learning model described in Section 5.

To compare the performance achieved by the above policies, we use the percentage of pools in the test set that are counterfactually harmed by each algorithmic policy $\hat{\pi}(\cdot)$ and the expected counterfactual utility $\mathbb{E}_{\boldsymbol{Y} \sim P^{\mathcal{M}\,|\,\boldsymbol{X}=\boldsymbol{x}, \boldsymbol{Y}=\boldsymbol{y}, \boldsymbol{L}=\boldsymbol{l}\,;\,\text{do}(\boldsymbol{L}=\hat{\pi}(\cdot))}(\boldsymbol{Y})}[\mathbf{1}^T \boldsymbol{Y}]$ achieved by each algorithmic policy $\hat{\pi}(\cdot)$ in comparison with the realized utility achieved by the default policy $\tilde{\pi}(\boldsymbol{x}, w)$ across pools of refugees in the test set. In Appendix D, we also report the (realized) utility achieved by each policy across pools of refugees in the test set.

Results. We first calculate the percentage of pools in the test set that are counterfactually harmed by each algorithmic policy under different noise levels w. Table 1 presents the results for $\beta = 0.6$, which offer several insights. We find that, by postprocessing the predicted employment probabilities $\boldsymbol{g}$, the algorithmic policy $\hat{\pi}(h(\boldsymbol{g}))$ counterfactually harms fewer (similar) pools than the algorithmic policy $\hat{\pi}(\boldsymbol{g})$ for low (high) noise levels, *i.e.*, $w \leq 0.75$ ($w > 0.75$). This suggest that, as the level of noise of the default policy $\tilde{\pi}(\boldsymbol{x}, w)$ increases, it is more difficult for our framework to learn to avoid harm from past placement decisions made by the default policy and the (corresponding) realized employment outcomes.

Next, we compare the per-pool expected counterfactual utility gain achieved by the algorithmic policies $\hat{\pi}(\breve{\boldsymbol{g}})$ and $\hat{\pi}(h(\boldsymbol{g}))$, which are designed using our framework, in comparison to the counterfactual utility gain achieved by algorithmic policy $\hat{\pi}(\boldsymbol{g})$ under low ($w = 0.125$), medium ($w = 0.5$) and high ($w = 0.875$) noise level. Figure 3 summarizes the results for $\beta = 0.6$, which further supports



(a) Low noise level (b) Medium noise level (c) High noise level

Figure 4: Expected counterfactual utility achieved by the algorithmic policies $\hat{\pi}(\boldsymbol{p})$, $\hat{\pi}(\boldsymbol{g})$ and $\hat{\pi}(h(\boldsymbol{g}))$ in comparison with the expected realized utility achieved by the default policy $\tilde{\pi}(\boldsymbol{x}, w)$ across all pools in the test set for different β values under low ($w = 0.125$), medium ($w = 0.5$) and high ($w = 0.875$) noise level. For $\hat{\pi}(h(\boldsymbol{g}))$, the results are averaged over 5 runs, where the error bands represent standard deviations.

the findings derived from Table 1. As expected from Proposition 2, we find that the unrealizable algorithmic policy $\hat{\pi}(\breve{\boldsymbol{g}})$ always achieves a positive expected counterfactual gain for every pool. Further, for low and medium noise levels, we find that the algorithmic policy $\hat{\pi}(h(\boldsymbol{g}))$ typically offers a greater per-pool expected counterfactual utility gain than the algorithmic policy $\hat{\pi}(\boldsymbol{g})$ and it achieves a positive expected counterfactual utility averaged across pools. For high noise levels, we find that both algorithmic policies $\hat{\pi}(h(\boldsymbol{g}))$ and $\hat{\pi}(\boldsymbol{g})$ offer a comparable expected counterfactual utility gain per pool.

Finally, we compare the expected counterfactual utility achieved by the algorithmic policies $\hat{\pi}(\breve{\boldsymbol{g}})$ and $\hat{\pi}(h(\boldsymbol{g}))$ in comparison with the expected realized utility achieved by the default policy $\tilde{\pi}(\boldsymbol{x}, w)$ for different β values under low ($w = 0.125$), medium ($w = 0.5$) and high ($w = 0.875$) noise level. Figure 4 summarizes the results, which show that, for low and medium noise levels, the algorithmic policy $\hat{\pi}(h(\boldsymbol{g}))$ offers greater expected counterfactual utility than $\hat{\pi}(\boldsymbol{g})$ across all β values and, for high noise levels, both offer a comparable expected counterfactual utility.

7. Conclusions

In this work, we have initiated the study of (counterfactual) harm in data-driven algorithmic matching. We have developed a postprocessing framework that, given retrospective data under a given default policy, postprocesses the predictions made by the classifiers used in existing solutions to avoid causing harm. Further, using synthetic refugee data created using a variety of publicly available data, we have empirically shown that our framework may be effective in making algorithmic placement decisions that are less likely to be harmful than existing solutions. Our work opens up many interesting avenues for future work. For example, our work considers a setting in which data-driven algorithmic matching replaces a human decision maker. However, it would be interesting to consider a setting in which data-driven algorithmic matching supports, rather than replace, the human decision maker. Moreover, it would be important to evaluate our methodological contributions on real retrospective refugee data comprising of placement decisions and realized employment outcomes. Additionally, our approach can be extended to account for preferences of refugees to specific locations, particularly when they have family ties in one or multiple locations. If these preferences are to be considered in the matching process, our approach can be adapted by restricting the set of feasible matchings to prioritize proximity, without increasing computational complexity. Further, we have focused on reducing the overall amount of harm caused by data-driven algorithmic matching. However, this may lead significant disparities across demographic groups, as shown recently (Freund et al., 2023; Körtner and Bach, 2023; Zezulka and Genin, 2024), and thus it would be important to extend our methodology to account for fairness considerations. Finally, investigating decision-focused learning (Mandi et al., 2024) for counterfactual harm minimization is a promising direction for future research.

Acknowledgments

We would like to thank Evimaria Terzi for fruitful discussions at an early stage of the project. Gomez-Rodriguez acknowledges support from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 945719).

References

Alastair Ager and Alison Strang. Understanding integration: A conceptual framework. *Journal of Refugee Studies*, 21:166, 2008.

Narges Ahani, Tommy Andersson, Alessandro Martinello, Alexander Teytelboym, and Andrew C Trapp. Placement optimization in refugee resettlement. *Operations Research*, 69(5):1468–1486, 2021.

Narges Ahani, Paul Gölz, Ariel D. Procaccia, Alexander Teytelboym, and Andrew C. Trapp. Dynamic placement in refugee resettlement. *Operations Research*, September 2023.

Joshua D Angrist, Guido W Imbens, and Donald B Rubin. Identification of causal effects using instrumental variables. *Journal of the American statistical Association*, 91(434):444–455, 1996.

Haris Aziz, Agnes Cseh, John P Dickerson, and Duncan C McElfresh. Optimal kidney exchange with immunosuppressants. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 21–29. AAAI Press, 2021.

Kirk Bansak, Jeremy Ferwerda, Jens Hainmueller, Andrea Dillon, Dominik Hangartner, Duncan Lawrence, and Jeremy Weinstein. Improving refugee integration through data-driven algorithmic assignment. *Science*, 359(6373):325–329, 2018.

Rina Foygel Barber. Is distribution-free inference possible for binary regression? *Electronic Journal of Statistics*, 14(2):3487–3524, 2020.

Sander Beckers, Hana Chockler, and Joseph Halpern. A causal analysis of harm. In *Advances in Neural Information Processing Systems*, volume 35, pages 2365–2376. Neural Information Processing Systems Foundation, 2022.

Sander Beckers, Hana Chockler, and Joseph Y Halpern. Quantifying harm. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 363–371. IJCAI, 2023.

Kristóf Bérczi, Lydia Mirabel Mendoza-Cadena, and Kitti Varga. Inverse optimization problems with multiple weight functions. *Discrete Applied Mathematics*, 327:134–147, 2023. ISSN 0166-218X.

Bureau of Labor Statistics. Local area unemployment statistics. https://www.bls.gov/lau/table14full22.htm, 2024. Online; accessed 20 May 2024.

Census Bureau. Educational attainment. https://data.census.gov/table/ACSST1Y2022.S1501, 2024. Online; accessed 20 May 2024.

Vincent P Crawford and Elsie Marie Knoer. Job matching with heterogeneous firms and workers. *Econometrica: Journal of the Econometric Society*, pages 437–450, 1981.

Marc Demange and Jérôme Monnot. An introduction to inverse combinatorial problems. *Paradigms of Combinatorial Optimization: Problems and New Approaches*, pages 547–586, 2014.

Miroslav Dudík, John Langford, and Lihong Li. Doubly robust policy evaluation and learning. In *International Conference on Machine Learning*, pages 1097–1104, 2011.

Joel Feinberg. Wrongful life and the counterfactual element in harming. *Social Philosophy and Policy*, 4(1):145–178, 1986.

Daniel Freund, Thodoris Lykouris, Elisabeth Paulson, Bradley Sturt, and Wentao Weng. Group fairness in dynamic refugee assignment. In *Proceedings of the ACM Conference on Economics and Computation*, page 701. ACM, 2023.

David Gale and Lloyd S Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

Alan Gibbons. *Algorithmic graph theory*. Cambridge University Press, 1985.

Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 69(2): 243–268, 2007.

Chirag Gupta, Aleksandr Podkopaev, and Aaditya Ramdas. Distribution-free binary classification: prediction sets, confidence intervals and calibration. In *Advances in Neural Information Processing Systems*, volume 33, pages 3711–3723. Neural Information Processing Systems Foundation, 2020.

Matthew Hanser. The metaphysics of harm. *Philosophy and Phenomenological Research*, 77(2): 421–450, 2008.

Clemens Heuberger. Inverse combinatorial optimization: A survey on problems, methods, and results. *Journal of combinatorial optimization*, 8:329–361, 2004.

Justin Klocksiem. A defense of the counterfactual comparative account of harm. *American Philosophical Quarterly*, 49(4):285–300, 2012.

Anton Korikov and J Christopher Beck. Counterfactual explanations via inverse constraint programming. In *International Conference on Principles and Practice of Constraint Programming*. Dagstuhl, 2021.

Anton Korikov, Alexander Shleyfman, and Chris Beck. Counterfactual explanations for optimization-based decisions in the context of the gdpr. In *Proceedings of the ICAPS Workshop on Explainable AI Planning*, 2021.

John Körtner and Ruben Bach. Inequality-averse outcome-based matching. 2023.

Lap Chi Lau, Ramamoorthi Ravi, and Mohit Singh. *Iterative Methods in Combinatorial Optimization*, volume 46. Cambridge University Press, 2011.

Taewoo Lee and Daria Terekhov. *Encyclopedia of Optimization: Inverse Optimization*, pages 1–6. Springer, 2020.

Haoxuan Li, Chunyuan Zheng, Yixiao Cao, Zhi Geng, Yue Liu, and Peng Wu. Trustworthy policy learning under the counterfactual no-harm criterion. In *Proceedings of the International Conference on Machine Learning*. PMLR, 2023.

Lydia T Liu, Solon Barocas, Jon Kleinberg, and Karen Levy. On the actionability of outcome prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 22240–22249, 2024.

Zhenhong Liu and Jianzhong Zhang. On inverse problems of optimum perfect matching. *Journal of combinatorial optimization*, 7:215–228, 2003.

László Lovász and Michael D Plummer. *Matching theory*. American Mathematical Society, 2009.

Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *Journal of Artificial Intelligence Research*, 80:1623–1701, 2024.

Duncan C. McElfresh, Christian Kroer, Sergey Pupyrev, Eric Sodomka, Karthik Sankararaman, Zack Chauvin, Neil Dexter, and John P. Dickerson. Matching algorithms for blood donation. *Nature Machine Intelligence*, 5(10):1108–1118, October 2023.

Migration Policy Institute. State immigration data profiles. https://www.migrationpolicy.org/programs/data-hub/state-immigration-data-profiles, 2024. Online; accessed 20 May 2024.

Judea Pearl. *Causality*. Cambridge University Press, 2009.

Joaquin Quinonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D Lawrence. *Dataset Shift in Machine Learning*. MIT Press, 2009.

Jonathan Richens, Rory Beard, and Daniel H Thompson. Counterfactual harm. In *Advances in Neural Information Processing Systems*, volume 35, pages 36350–36365. Neural Information Processing Systems Foundation, 2022.

James M Robins, Andrea Rotnitzky, and Lue Ping Zhao. Estimation of regression coefficients when some regressors are not always observed. *Journal of the American statistical Association*, 89(427): 846–866, 1994.

Haya Salah and Sharan Srinivas. Predict, then schedule: Prescriptive analytics approach for machine learning-enabled sequential clinical scheduling. *Computers & Industrial Engineering*, 169:108270, 2022.

Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. In *Proceedings of the International Conference on Machine Learning*, pages 459–466. Omnipress, 2012.

Ivan Stelmakh, Nihar Shah, and Aarti Singh. Peerreview4all: Fair and accurate reviewer assignment in peer review. *Journal of Machine Learning Research*, 22(163):1–66, 2021.

Steven L Tanimoto, Alon Itai, and Michael Rodeh. Some matching problems for bipartite graphs. *Journal of the ACM*, 25(4):517–525, 1978.

UNHCR. Refugee data finder. https://www.unhcr.org/refugee-statistics/, 2022. Online; accessed 20 May 2024.

Wouter AC van Amsterdam, Nan van Geloven, Jesse H Krijthe, Rajesh Ranganath, and Giovanni Ciná. When accurate prediction models yield harmful self-fulfilling prophecies. *arXiv preprint arXiv:2312.01210*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, volume 30, 2017.

Angelina Wang, Sayash Kapoor, Solon Barocas, and Arvind Narayanan. Against predictive optimization: On the legitimacy of decision-making algorithms that optimize predictive accuracy. *ACM Journal of Responsible Computing*, 1(1), March 2024.

World Bank. World bank open data. https://databank.worldbank.org/databases, 2024. Online; accessed 20 May 2024.

Xiaoguang Yang and Jianzhong Zhang. Partial inverse assignment problems under ℓ_1 norm. *Operations research letters*, 35(1):23–28, 2007.

Sebastian Zezulka and Konstantin Genin. From the fair distribution of predictions to the fair distribution of social goods: Evaluating the impact of fair machine learning on long-term unemployment. In *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, pages 1984–2006, 2024.

Appendix A. Proofs

A.1. Proof of Proposition 1

We prove this by contra-position. For a given $\boldsymbol{X} = \boldsymbol{x}$, if the algorithmic matching policy $\hat{\pi}(\boldsymbol{g}) = (\text{argmax}_{l\in\mathcal{L}}\, \hat{z}_{il})_{i\in\mathcal{I}}$, where $\hat{\boldsymbol{z}} = (\hat{z}_{il})_{i\in\mathcal{I},l\in\mathcal{L}}$ is an optimal integral solution to the linear program defined by Eq. 5, does not correspond to an optimal policy $\pi^*(\boldsymbol{g})$, then we must have that

$$\mathbb{E}_{\boldsymbol{Y}\sim P^{\mathcal{M}\,;\,\text{do}(\boldsymbol{L}=\pi^*(\boldsymbol{g}))}}[\mathbf{1}^T\boldsymbol{Y} \mid \boldsymbol{X} = \boldsymbol{x}] > \mathbb{E}_{\boldsymbol{Y}\sim P^{\mathcal{M}\,;\,\text{do}(\boldsymbol{L}=\hat{\pi}(\boldsymbol{g}))}}[\mathbf{1}^T\boldsymbol{Y} \mid \boldsymbol{X} = \boldsymbol{x}]. \tag{14}$$

Given any algorithmic policy $\pi(\boldsymbol{g})$, let $\boldsymbol{z} = (z_{il})_{i\in\mathcal{I},l\in\mathcal{L}}$ with $z_{il} = \mathbb{1}[\pi_i(\boldsymbol{g}) = l]$, where note that z_{il} has to satisfy the constraints of the matching problem defined in Eq. 5 for $\pi(\boldsymbol{g})$ to be a valid algorithmic matching policy. That is, each refugee is assigned to at most one location and each location l is assigned at most c_l refugees. Further, we have that:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{Y}\sim P^{\mathcal{M}\,;\,\text{do}(\boldsymbol{L}=\pi(\boldsymbol{g}))}}[\mathbf{1}^T\boldsymbol{Y} \mid \boldsymbol{X} = \boldsymbol{x}] &= \sum_{i\in\mathcal{I}} P^{\mathcal{M}\,;\,do(L_i=(\pi_i(\boldsymbol{g})))}(Y_i = 1 \mid X_i = x_i) \\
&= \sum_{i\in\mathcal{I},l\in\mathcal{L}} P^{\mathcal{M}\,;\,do(L_i=l)}(Y_i = 1 \mid X_i = x_i)\mathbb{1}[\pi_i(g(\boldsymbol{x})) = l] \\
&= \sum_{i\in\mathcal{I},l\in\mathcal{L}} g_l(x_i) z_{il}
\end{aligned}$$

Hence, we can rewrite Eq. 14 as

$$\sum_{i\in\mathcal{I},l\in\mathcal{L}} g_l(x_i) z^*_{il} > \sum_{i\in\mathcal{I},l\in\mathcal{L}} g_l(x_i) \hat{z}_{il}$$

where $z^*_{il} = \mathbb{1}[\pi^*_i(\boldsymbol{g}) = l]$. However, this leads to a contradiction since the solution $\hat{\boldsymbol{z}}$ cannot be the optimal solution to the linear program defined by Eq. 5 if $\boldsymbol{z}^*$ is a feasible solution and has greater objective function value. As a consequence, it must hold that

$$\mathbb{E}_{\boldsymbol{Y}\sim P^{\mathcal{M}\,;\,\text{do}(\boldsymbol{L}=\pi^*(\boldsymbol{g}))}}[\mathbf{1}^T\boldsymbol{Y} \mid \boldsymbol{X} = \boldsymbol{x}] \leq \mathbb{E}_{\boldsymbol{Y}\sim P^{\mathcal{M}\,;\,\text{do}(\boldsymbol{L}=\hat{\pi}(\boldsymbol{g}))}}[\mathbf{1}^T\boldsymbol{Y} \mid \boldsymbol{X} = \boldsymbol{x}]$$

and thus $\hat{\pi}(\boldsymbol{g}) \in \Pi^*(\boldsymbol{g})$.

A.2. Proof of Proposition 2

We begin by writing the counterfactual probability of $Y_i = 1$, for $i \in \mathcal{I}$, in terms of the probability density of D_i, V_{i,l_i} and $V_{i,\pi_i(\boldsymbol{g})}$ as an expectation:

$$\begin{aligned}
&P^{\mathcal{M}\,|\,\boldsymbol{X}=\boldsymbol{x},\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{L}=\boldsymbol{l}\,;\,\text{do}(\boldsymbol{L}=\pi(\boldsymbol{g}))}(Y_i = 1) = \mathbb{E}_{Y_i\sim P^{\mathcal{M}\,|\,\boldsymbol{X}=\boldsymbol{x},\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{L}=\boldsymbol{l}\,;\,\text{do}(\boldsymbol{L}=\pi(\boldsymbol{g}))}}[Y_i] \\
&= \int_{d,v,v'} P(D_i = d, V_{i,l_i} = v, V_{i,\pi_i(\boldsymbol{g})} = v' \mid \boldsymbol{X} = \boldsymbol{x}, \boldsymbol{Y} = \boldsymbol{y}, \boldsymbol{L} = \boldsymbol{l}) \cdot f_Y(d, v')\, dd\, dv\, dv'. \quad (15)
\end{aligned}$$

Since $V_{i,\pi_i(\boldsymbol{g})}$ is independent of D_i, X_i and Y_i conditioned on V_{i,l_i}, we can rewrite the probability density function in Eq. 15 as

$$\begin{aligned}
&P(D_i = d, V_{i,l_i} = v \mid X_i = x_i, Y_i = y_i, L_i = l_i) \cdot P(V_{i,\pi_i(\boldsymbol{g})} = v' \mid V_{i,l_i} = v) \\
&\quad = P(D_i = d, V_{i,l_i} = v \mid f_X(D_i) = x_i, f_Y(D_i, V_{i,l_i}) = y_i) \cdot P(V_{i,\pi_i(\boldsymbol{g})} = v' \mid V_{i,l_i} = v), \quad (16)
\end{aligned}$$

Further, it is straight forward to see that

$$P(V_{i,\pi_i(\boldsymbol{g})} = v' | V_{i,l_i} = v) = \begin{cases} 1 & \text{if } \pi_i(\boldsymbol{g}) = l_i \wedge v' = v \\ 0 & \text{if } \pi_i(\boldsymbol{g}) = l_i \wedge v' \neq v \\ P(V_{i,\pi_i(\boldsymbol{g})} = v') = P(V = v' \,|\, L_i = \pi_i(\boldsymbol{g})) & \text{if } \pi_i(\boldsymbol{g}) \neq l_i, \end{cases} \tag{17}$$

where the last case follows from the definition of SCM $\mathcal{M}$ since V_{i,l_i} and $V_{i,\pi_i(\boldsymbol{g})}$ are independent when $\pi_i(\boldsymbol{g}) \neq l_i$. Thus, when $\pi_i(\boldsymbol{g}) = l_i$, it follows from Eqs. 16 and 17 that the right hand side of Eq. 15 is equivalent to

$$\int_{d,v} P(D_i = d, V_{i,l_i} = v \,|\, f_X(D_i) = x_i, f_Y(D_i, V_{i,l_i}) = y_i) \cdot f_Y(d, v) \, dd \, dv$$
$$= y_i \cdot \int_{d,v} P(D_i = d, V_{i,l_i} = v \,|\, f_X(D_i) = x_i, f_Y(D_i, V_{i,l_i}) = y_i) \, dd \, dv = y_i, \tag{18}$$

where the equality stems from the fact that density function $P(D_i = d, V_{i,l_i} = v \,|\, f_X(D_i) = x_i, f_Y(D_i, V_{i,l_i}) = y_i)$ can only be positive when $f_Y(d, v) = y_i$. Since $y_i \in \{0, 1\}$, it follows from Eq. 18 that for $\boldsymbol{Y} \sim P^{\mathcal{M}\,|\,\boldsymbol{X}=\boldsymbol{x},\boldsymbol{Y}=\boldsymbol{y},\boldsymbol{L}=\boldsymbol{l}\,;\,\mathrm{do}(\boldsymbol{L}=\pi(\boldsymbol{g}))}$ must hold that

$$Y_i = y_i \quad \text{for all } i \in \mathcal{I} \text{ such that } \pi_i(\boldsymbol{g}) = l_i. \tag{19}$$

Now, let $\pi(\boldsymbol{g})$ be a policy in $\Pi_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{l}}$. By definition of $\Pi_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{l}}$, we have that, for any $i \in \mathcal{I}$, $y_i = 1$ implies $\pi_i(\boldsymbol{g}) = l_i$. Hence, we can conclude that

$$\boldsymbol{1}^T \boldsymbol{y} = \sum_{i \in \mathcal{I}} y_i = \sum_{i \in \mathcal{I}} y_i \cdot \mathbb{1}[\pi_i(\boldsymbol{g}) = l_i] \overset{(i)}{=} \sum_{i \in \mathcal{I}} Y_i \cdot \mathbb{1}[\pi_i(\boldsymbol{g}) = l_i] \leq \boldsymbol{1}^T \boldsymbol{Y}.$$

where (i) follows from Eq. 19.

A.3. Proof of Proposition 3

Using linearity of expectation, we can rewrite the right hand side of Eq. 8 as

$$\mathbb{E}_{\boldsymbol{X}',\boldsymbol{Y}',\boldsymbol{L}' \sim P^{\mathcal{M}}} \left[\mathbb{E}_{\boldsymbol{Y} \sim P^{\mathcal{M}\,|\,\boldsymbol{X}=\boldsymbol{X}',\boldsymbol{Y}=\boldsymbol{Y}',\boldsymbol{L}=\boldsymbol{L}'\,;\,\mathrm{do}(\boldsymbol{L}=\pi(\boldsymbol{G}))}} [\boldsymbol{1}^T \boldsymbol{Y}] \right]$$
$$= \int \sum_{\boldsymbol{y}'} \sum_{\boldsymbol{l}'} P^{\mathcal{M}}(\boldsymbol{X}' = \boldsymbol{x}', \boldsymbol{Y}' = \boldsymbol{y}', \boldsymbol{L}' = \boldsymbol{l}') \left[\mathbb{E}_{\boldsymbol{Y} \sim P^{\mathcal{M}\,|\,\boldsymbol{X}=\boldsymbol{X}',\boldsymbol{Y}=\boldsymbol{Y}',\boldsymbol{L}=\boldsymbol{L}'\,;\,\mathrm{do}(\boldsymbol{L}=\pi(\boldsymbol{G}))}} [\boldsymbol{1}^T \boldsymbol{Y}] \right] \, d\boldsymbol{x}' \tag{20}$$

Using Eq. 15 and the fact that $\boldsymbol{X}' \overset{d}{=} \boldsymbol{X}$ and $\boldsymbol{Y}' \overset{d}{=} \boldsymbol{Y}$, we have that

$$P^{\mathcal{M}}(\boldsymbol{X}' = \boldsymbol{x}', \boldsymbol{Y}' = \boldsymbol{y}', \boldsymbol{L}' = \boldsymbol{l}') \left[\mathbb{E}_{\boldsymbol{Y} \sim P^{\mathcal{M}\,|\,\boldsymbol{X}=\boldsymbol{X}',\boldsymbol{Y}=\boldsymbol{Y}',\boldsymbol{L}=\boldsymbol{L}'\,;\,\mathrm{do}(\boldsymbol{L}=\pi(\boldsymbol{G}))}} [\boldsymbol{1}^T \boldsymbol{Y}] \right]$$
$$= \int P^{\mathcal{M}}(\boldsymbol{L}' = \boldsymbol{l}' \,|\, \boldsymbol{X}' = \boldsymbol{x}') \times P(\boldsymbol{D} = \boldsymbol{d}', \boldsymbol{V}_{:,\boldsymbol{L}'} = \boldsymbol{v}', \boldsymbol{V}_{:,\pi(\boldsymbol{G})} = \boldsymbol{v}, \boldsymbol{X}' = \boldsymbol{x}', \boldsymbol{Y}' = \boldsymbol{y}' \,|\, \boldsymbol{L}' = \boldsymbol{l}')$$
$$\times \left[\sum_{i \in \mathcal{I}} f_Y(d'_i, v_i) \right] d\boldsymbol{d}' \, d\boldsymbol{v}' \, d\boldsymbol{v} \tag{21}$$

where $\boldsymbol{V}_{:,\boldsymbol{L}'} = (V_{i,L'_i})_{i\in\mathcal{I}}$ and $\boldsymbol{V}_{:,\pi(\boldsymbol{G})} = (V_{i,\pi_i(\boldsymbol{G})})_{i\in\mathcal{I}}$ are vectors of exogeneous noise variables $V_{i,l}$ with $i \in \mathcal{I}, l \in \mathcal{L}$.

Note that, when combining Eq. 21 with Eq. 20, we can omit random variables $\boldsymbol{X}'$ and $\boldsymbol{Y}'$ from the joint distribution $P(\boldsymbol{D}, \boldsymbol{V}_{:,\boldsymbol{L}'}, \boldsymbol{V}_{:,\pi(\boldsymbol{G})}, \boldsymbol{X}', \boldsymbol{Y}' \,|\, \boldsymbol{L}')$ and substitute it from the conditional distribution $P^{\mathcal{M}}(\boldsymbol{L}' = \boldsymbol{l}' \,|\, \boldsymbol{X}' = \boldsymbol{x}')$ since the outcome of both variables can be deduced deterministically from exogenous variables $\boldsymbol{D}$ and $\boldsymbol{V}_{:,\boldsymbol{L}'}$. We obtain following expression

$$\begin{aligned}
&\mathbb{E}_{\boldsymbol{X}',\boldsymbol{Y}',\boldsymbol{L}'\sim P^{\mathcal{M}}}\left[\mathbb{E}_{\boldsymbol{Y}\sim P^{\mathcal{M}\,|\,\boldsymbol{X}=\boldsymbol{X}',\boldsymbol{Y}=\boldsymbol{Y}',\boldsymbol{L}=\boldsymbol{L}'\,;\,\mathrm{do}(\boldsymbol{L}=\pi(\boldsymbol{G}))}}[\boldsymbol{1}^T\boldsymbol{Y}]\right] \\
&\quad = \sum_{\boldsymbol{l}'} \int P^{\mathcal{M}}(\boldsymbol{L}' = \boldsymbol{l}' \,|\, \boldsymbol{D}' = \boldsymbol{d}') \cdot P(\boldsymbol{D} = \boldsymbol{d}', \boldsymbol{V}_{:,\boldsymbol{L}'} = \boldsymbol{v}', \boldsymbol{V}_{:,\pi(\boldsymbol{G})} = \boldsymbol{v} \,|\, \boldsymbol{L}' = \boldsymbol{l}') \\
&\qquad\qquad \times \left[\sum_{i\in\mathcal{I}} f_Y(d'_i, v_i)\right] d\boldsymbol{d}'\, d\boldsymbol{v}'\, d\boldsymbol{v}
\end{aligned} \tag{22}$$

Using that $\boldsymbol{D}, \boldsymbol{V}_{:,\pi(\boldsymbol{G})}$ are independent from $\boldsymbol{L}'$ and rearranging summations and integrals in the above equation, we have that

$$\begin{aligned}
&\int_{\boldsymbol{d}',\boldsymbol{v}} P(\boldsymbol{D} = \boldsymbol{d}', \boldsymbol{V}_{:,\pi(\boldsymbol{G})} = \boldsymbol{v}) \left[\sum_{i\in\mathcal{I}} f_Y(d'_i, v_i)\right] \times \sum_{\boldsymbol{l}'} P^{\mathcal{M}}(\boldsymbol{L}' = \boldsymbol{l}' \,|\, \boldsymbol{D}' = \boldsymbol{d}') \\
&\qquad \times \int_{\boldsymbol{v}'} P(\boldsymbol{V}_{:,\boldsymbol{L}'} = \boldsymbol{v}' \,|\, \boldsymbol{L}' = \boldsymbol{l}', \boldsymbol{V}_{:,\pi(\boldsymbol{G})} = \boldsymbol{v})\, d\boldsymbol{v}'\, d\boldsymbol{d}' d\boldsymbol{v} \\
&\qquad \overset{(i)}{=} \int_{d',v} P(\boldsymbol{D} = \boldsymbol{d}') \cdot P(\boldsymbol{V}_{:,\pi(\boldsymbol{G})} = \boldsymbol{v}) \cdot \sum_{i\in\mathcal{I}} f_Y(d'_i, v_i)\, d\boldsymbol{d}' d\boldsymbol{v} \\
&\qquad \overset{(ii)}{=} \mathbb{E}_{\boldsymbol{Y}\sim P^{\mathcal{M}\,;\,\mathrm{do}(\boldsymbol{L}=\pi(\boldsymbol{G}))}}[\boldsymbol{1}^T\boldsymbol{Y}]
\end{aligned}$$

where (i) follows from $\sum_{\boldsymbol{l}'} P^{\mathcal{M}}(\boldsymbol{L}' = \boldsymbol{l}' \,|\, \boldsymbol{D}' = \boldsymbol{d}') = 1$ and the fact that $\int_{\boldsymbol{v}'} P(\boldsymbol{V}_{:,\boldsymbol{L}'} = \boldsymbol{v}' \,|\, \boldsymbol{L}' = \boldsymbol{l}', \boldsymbol{V}_{:,\pi(\boldsymbol{G})} = \boldsymbol{v})\, d\boldsymbol{v}' = 1$ and (ii) follows from the definition of intervened SCM $\mathcal{M}^{\mathrm{do}(\boldsymbol{L}=\pi(\boldsymbol{G}))}$.

A.4. Proof of Theorem 4

We first find the minimally modified predicted probabilities $\boldsymbol{g}'$ under which the counterfactually harmless policy $\pi'(\boldsymbol{g}) \in \Pi_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{l}}$ that maximizes the (conditional) predicted utility over the refugees in $\mathcal{I}' = \{i \in \mathcal{I} \,|\, l_i = 0\}$ with respect to $\boldsymbol{g}$ also maximizes the (conditional) predicted utility over refugees in $\mathcal{I}$ with respect to $\boldsymbol{g}'$, *i.e.*,

$$\begin{aligned}
&\text{minimize} && ||\boldsymbol{g}' - \boldsymbol{g}||_1 \\
&\text{subject to} && \pi'(\boldsymbol{g}) \in \hat{\Pi}(\boldsymbol{g}'),
\end{aligned} \tag{23}$$

where note that, under the minimally modified predicted probabilities $\boldsymbol{g}'$, there might exist other $\hat{\pi}(\boldsymbol{g}') \in \hat{\Pi}(\boldsymbol{g}')$ such that $\hat{\pi}(\boldsymbol{g}') \notin \Pi_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{l}}$. In what follows, we assume that the pool size $n = \sum_{l\in\mathcal{L}} c_l$ without loss of generality.[13]

[13] If $n < \sum_{l\in\mathcal{L}} c_l$, we can introduce a (dummy) refugee set $\mathcal{I}'$ of size $\sum_{l\in\mathcal{L}} c_l - n$ with predicted probabilities $g_l(x_i) = 0$ for every $i \in \mathcal{I}'$ and $l \in \mathcal{L}$. If $\sum_{l\in\mathcal{L}} c_l < n$, we can introduce a (dummy) location l' with capacity $c_{l'} = n - \sum_{l\in\mathcal{L}} c_l$ with predicted probabilities $g_{l'}(x_i) = 0$ for every $i \in \mathcal{I}$. In both cases, the optimal solution $\boldsymbol{g}'$ does not change.

To solve the above problem, we apply the algorithmic framework by Yang and Zhang (2007). First, we claim that, for all $i \in \mathcal{I}$ and $l \in \mathcal{L}$, the minimally modified predicted probability $g'_l(x_i)$ must satisfy that

$$\begin{aligned} g'_l(x_i) &\geq g_l(x_i) \text{ if } \exists \hat{\pi}(\boldsymbol{g}') \in \hat{\Pi}(\boldsymbol{g}') \text{ such that } \hat{\pi}_i(\boldsymbol{g}') = l \\ g'_l(x_i) &\leq g_l(x_i) \text{ if } \nexists \hat{\pi}(\boldsymbol{g}') \in \hat{\Pi}(\boldsymbol{g}') \text{ such that } \hat{\pi}_i(\boldsymbol{g}') = l, \end{aligned} \tag{24}$$

We establish this claim by contradiction. Assume there exists $\hat{\pi}(\boldsymbol{g}') \in \hat{\Pi}(\boldsymbol{g}')$ with at least one $i \in \mathcal{I}$ such that $\hat{\pi}_i(\boldsymbol{g}') = l$ and $g'_l(x_i) < g_l(x_i)$. This contradicts the optimality of $\boldsymbol{g}'$ as a solution for Eq. 23, since maintaining the original predicted probability $g_l(x_i)$ yields a better solution. Assume there exists at least one $i \in \mathcal{I}$ such that $\nexists \hat{\pi}(\boldsymbol{g}') \in \hat{\Pi}(\boldsymbol{g}')$ with $\hat{\pi}_i(\boldsymbol{g}') = l$ and $g'_l(x_i) > g_l(x_i)$. This also contradicts the optimality of $\boldsymbol{g}'$ as maintaining the original predicted probability $g_l(x_i)$ for $i \in \mathcal{I}$ also yields a better solution.

Using the above claim, we can derive a lower bound on the objective value achieved by the minimally modified predicted probabilities $\boldsymbol{g}'$, as formalized by the following Lemma:

Lemma 5 *The minimally modified predicted probabilities* $\boldsymbol{g}'$ *under which* $\pi'(\boldsymbol{g}) \in \hat{\Pi}(\boldsymbol{g}')$ *satisfy that*

$$||\boldsymbol{g}' - \boldsymbol{g}||_1 \geq \sum_{i \in \mathcal{I}} \left[g_{\hat{\pi}_i(\boldsymbol{g})}(x_i) - g_{\hat{\pi}_i(\boldsymbol{g}')}(x_i) \right]$$

for any $\hat{\pi}(\boldsymbol{g}) \in \hat{\Pi}(\boldsymbol{g})$ *and* $\hat{\pi}(\boldsymbol{g}') \in \hat{\Pi}(\boldsymbol{g}')$.

Further, we can use the dual of the linear program used to find an algorithmic matching policy $\hat{\pi}(\boldsymbol{g}) \in \hat{\Pi}(\boldsymbol{g})$, defined in Eq. 13, to show that the above lower bound is essentially tight, as formalized by the following Proposition:

Proposition 6 *The minimally modified predicted probabilities* $\boldsymbol{g}'$ *under which* $\pi'(\boldsymbol{g}) \in \hat{\Pi}(\boldsymbol{g}')$ *satisfy that:*

$$||\boldsymbol{g}' - \boldsymbol{g}||_1 = \sum_{i \in \mathcal{I}} \left[g_{\hat{\pi}_i(\boldsymbol{g})}(x_i) - g_{\hat{\pi}_i(\boldsymbol{g}')}(x_i) \right]$$

for any $\hat{\pi}(\boldsymbol{g}) \in \hat{\Pi}(\boldsymbol{g})$ *and* $\hat{\pi}(\boldsymbol{g}') \in \hat{\Pi}(\boldsymbol{g}')$.

Importantly, the proof of the above proposition directly gives us an explicit solution to the problem of finding the minimally modified predicted probabilities $\boldsymbol{g}'$ under which $\pi'(\boldsymbol{g}) \in \hat{\Pi}(\boldsymbol{g}')$:

$$g'_l(x_i) = \begin{cases} \hat{u}_i + \hat{v}_l & \text{if } \pi'_i(\boldsymbol{g}) = l \\ g_l(x_i) & \text{otherwise,} \end{cases}$$

where $\hat{\boldsymbol{u}} = (\hat{u}_i)_{i \in \mathcal{I}}$ and $\hat{\boldsymbol{v}} = (\hat{v}_l)_{l \in \mathcal{L}}$ is the optimal solution to the dual of the linear program used to find an algorithmic matching policy $\hat{\pi}(\boldsymbol{g}) \in \hat{\Pi}(\boldsymbol{g})$.

Further, we can show that $\boldsymbol{g}'$ as defined above is also a solution to the problem of finding the minimally modified predicted probabilities such that there exists $\hat{\pi}(\boldsymbol{g}') \in \hat{\Pi}(\boldsymbol{g}')$ with $\hat{\pi}(\boldsymbol{g}') \in \Pi_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{l}}$. This claim is proven by contradiction. Assume there exists $\pi''(\boldsymbol{g}) \in \Pi_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{l}}$ such that, for the minimally modified predicted probabilities $\boldsymbol{g}''$ such that $\pi''(\boldsymbol{g}) \in \hat{\Pi}(\boldsymbol{g}'')$, it holds that $||\boldsymbol{g}'' - \boldsymbol{g}||_1 < ||\boldsymbol{g}' - \boldsymbol{g}||_1$. Since both $\pi''(\boldsymbol{g})$ and $\pi'(\boldsymbol{g})$ are counterfactually harmless, we have that $\pi''_i(\boldsymbol{g}) = \pi'_i(\boldsymbol{g})$ for all $i \in \mathcal{I} \setminus \mathcal{I}'$. Hence, using Proposition 6, it must follow that $\sum_{i \in \mathcal{I}'} g_{\pi'_i(\boldsymbol{g})}(x_i) < \sum_{i \in \mathcal{I}'} g_{\pi''_i(\boldsymbol{g})}(x_i)$.

This directly contradicts the optimality of $\pi'(\boldsymbol{g})$ as the policy that maximizes the (conditional) predicted utility over refugees in $\mathcal{I}'$ described in Eq. 11.

Then, we can immediately conclude that the minimally modified employment probabilities $\breve{\boldsymbol{g}}$ under which any $\hat{\pi}(\breve{\boldsymbol{g}}) \in \hat{\Pi}(\breve{\boldsymbol{g}})$ is provably counterfactually harmless as defined in Eq. 10 is given by:

$$\breve{g}_l(x_i) = \begin{cases} g'_l(x_i) + \epsilon & \text{if } \pi'_i(\boldsymbol{g}) = l \\ g'_l(x_i) & \text{otherwise,} \end{cases}$$

where $\epsilon > 0$ is an infinitesimally small constant that rules out the possibility that, under the minimally modified predicted probabilities $\breve{\boldsymbol{g}}$, there exists $\hat{\pi}(\breve{\boldsymbol{g}}) \in \hat{\Pi}(\breve{\boldsymbol{g}})$ such that $\hat{\pi}(\breve{\boldsymbol{g}}) \notin \Pi_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{l}}$. This concludes the proof.

A.5. Proof of Lemma 5

For any $\hat{\pi}(\boldsymbol{g}) \in \hat{\Pi}(\boldsymbol{g})$ and $\hat{\pi}(\boldsymbol{g}') \in \hat{\Pi}(\boldsymbol{g}')$, we have that:

$$\begin{aligned}
&\sum_{i \in \mathcal{I}, l \in \mathcal{L}} |g'_l(x_i) - g_l(x_i)| \\
&\geq \sum_{i \in \mathcal{I}} \Big[|g'_{\hat{\pi}_i(\boldsymbol{g}')}(x_i) - g_{\hat{\pi}_i(\boldsymbol{g}')}(x_i)| + \mathbb{1}(\hat{\pi}_i(\boldsymbol{g}) \neq \hat{\pi}_i(\boldsymbol{g}')) |g'_{\hat{\pi}_i(\boldsymbol{g})}(x_i) - g_{\hat{\pi}_i(\boldsymbol{g})}(x_i)| \Big] \\
&\overset{(i)}{=} \sum_{i \in \mathcal{I}} \Big[(g'_{\hat{\pi}_i(\boldsymbol{g}')}(x_i) - g_{\hat{\pi}_i(\boldsymbol{g}')}(x_i)) + \mathbb{1}(\hat{\pi}_i(\boldsymbol{g}) \neq \hat{\pi}_i(\boldsymbol{g}')) (g_{\hat{\pi}_i(\boldsymbol{g})}(x_i) - g'_{\hat{\pi}_i(\boldsymbol{g})}(x_i)) \Big] \\
&= \sum_{i \in \mathcal{I}} \Big[g'_{\hat{\pi}_i(\boldsymbol{g}')}(x_i) - g'_{\hat{\pi}_i(\boldsymbol{g})}(x_i) + g_{\hat{\pi}_i(\boldsymbol{g})}(x_i) - g_{\hat{\pi}_i(\boldsymbol{g}')}(x_i) \\
&\qquad\qquad + \mathbb{1}(\hat{\pi}_i(\boldsymbol{g}) = \hat{\pi}_i(\boldsymbol{g}')) (g'_{\hat{\pi}_i(\boldsymbol{g})}(x_i) - g_{\hat{\pi}_i(\boldsymbol{g})}(x_i)) \Big] \\
&\overset{(ii)}{\geq} \sum_{i \in \mathcal{I}} \left[g_{\hat{\pi}_i(\boldsymbol{g})}(x_i) - g_{\hat{\pi}_i(\boldsymbol{g}')}(x_i) \right],
\end{aligned}$$

where (i) follows from both inequalities in Eq. 24 and (ii) follows from the definition of $\hat{\pi}_i(\boldsymbol{g}')$ and the first inequality in Eq. 24.

A.6. Proof of Proposition 6

Let $\hat{z}(\boldsymbol{g})$ denote an optimal solution to the linear program defined by Eq. 5 under the predicted probabilities $\boldsymbol{g}$, and $\hat{\boldsymbol{u}}(\boldsymbol{g})$ and $\hat{\boldsymbol{v}}(\boldsymbol{g})$ denote the optimal solution to the dual of the same linear program also under $\boldsymbol{g}$.

From the complementary slackness conditions, for every $i \in \mathcal{I}$ and $l \in \mathcal{L}$, it holds that:

$$\begin{aligned}
\hat{u}_i(\boldsymbol{g}) + \hat{v}_l(\boldsymbol{g}) &= g_l(x_i) \quad \text{if } \hat{\pi}_i(\boldsymbol{g}) = l \\
\hat{u}_i(\boldsymbol{g}) + \hat{v}_l(\boldsymbol{g}) &\geq g_l(x_i) \quad \text{otherwise.}
\end{aligned}$$

Let $\hat{\boldsymbol{g}} = \{\hat{g}_l(x_i)\}_{i \in \mathcal{I}, l \in \mathcal{L}}$ where, for each $i \in \mathcal{I}$ and $l \in \mathcal{L}$, we set the value of $\hat{g}_l(x_i)$ as follows:

$$\hat{g}_l(x_i) = \begin{cases} \hat{u}_i(\boldsymbol{g}) + \hat{v}_l(\boldsymbol{g}) & \text{if } \pi'_i(\boldsymbol{g}) = l \\ g_l(x_i) & \text{otherwise.} \end{cases} \tag{25}$$

Now, note that, for all $i \in \mathcal{I}$ and $l \in \mathcal{L}$, it holds that $\hat{u}_i(\boldsymbol{g}) + \hat{v}_l(\boldsymbol{g}) \geq \hat{g}_l(x_i)$ and thus $\hat{\boldsymbol{u}}(\boldsymbol{g})$ and $\hat{\boldsymbol{v}}(\boldsymbol{g})$ is a feasible dual solution to the linear program defined by Eq. 5 under $\hat{\boldsymbol{g}}$. Moreover, for any $i \in \mathcal{I}$ and $l \in \mathcal{L}$ such that $\hat{z}_{il}(\boldsymbol{g}) = 1$, it holds that $\hat{u}_i(\boldsymbol{g}) + \hat{v}_l(\boldsymbol{g}) = \hat{g}_l(x_i)$ independently of the value of $\hat{z}_{il}(\hat{\boldsymbol{g}})$. As a direct consequence, we have that $\hat{\boldsymbol{z}}(\boldsymbol{g})$ is *still* an optimal solution to the linear program defined by Eq. 5 under $\hat{\boldsymbol{g}}$. Moreover, from complementary slackness, $\hat{\boldsymbol{u}}(\boldsymbol{g})$ and $\hat{\boldsymbol{v}}(\boldsymbol{g})$ are also an optimal solution to the dual of the linear program under $\hat{\boldsymbol{g}}$. Thus, by strong duality, we have that:

$$\sum_{i\in\mathcal{I},l\in\mathcal{L}} \hat{g}_l(x_i)\,\hat{z}_{il}(\boldsymbol{g}) = \sum_{i\in\mathcal{I}} \hat{u}_i(\boldsymbol{g}) + \sum_{l\in\mathcal{L}} c_l \cdot \hat{v}_l(\boldsymbol{g}) = \sum_{i\in\mathcal{I},l\in\mathcal{L}} g_l(x_i)\,\hat{z}_{il}(\boldsymbol{g}) \tag{26}$$

Next, let $\boldsymbol{z}' = (z'_{il})_{i\in\mathcal{I},l\in|Lcal}$ be defined as $z'_{il} = \mathbb{1}[\pi'_i(\boldsymbol{g}) = l]$. Then, using Eqs. 25 and 26 and the fact that $n = \sum_{l\in\mathcal{L}} c_l$, we have that:

$$\begin{aligned}
\sum_{i\in\mathcal{I},l\in\mathcal{L}} \hat{g}_l(x_i)\,z'_{il} &= \sum_{i\in\mathcal{I},l\in\mathcal{L}} \left[\mathbb{1}[\pi'_i(\boldsymbol{g}) = l]\,(\hat{u}_i(\boldsymbol{g}) + \hat{v}_l(\boldsymbol{g}))\,z'_{il}(\boldsymbol{g}) + \mathbb{1}[\pi'_i(\boldsymbol{g}) \neq l] \cdot g_l(x_i) z'_{il}(\boldsymbol{g})\right] \\
&= \sum_{i\in\mathcal{I},l\in\mathcal{L}} \mathbb{1}[\pi'_i(\boldsymbol{g}) = l]\,(\hat{u}_i(\boldsymbol{g}) + \hat{v}_l(\boldsymbol{g}))\,z'_{il}(\boldsymbol{g}) \\
&= \sum_{i\in\mathcal{I}} \hat{u}_i(\boldsymbol{g}) \left[\sum_{l\in\mathcal{L}} \mathbb{1}[\pi'_i(\boldsymbol{g}) = l] \cdot z'_{il}\right] + \sum_{l\in\mathcal{L}} \hat{v}_l(\boldsymbol{g}) \left[\sum_{i\in\mathcal{I}} \mathbb{1}[\pi'_i(\boldsymbol{g}) = l] \cdot z'_{il}\right] \\
&= \sum_{i\in\mathcal{I}} \hat{u}_i(\boldsymbol{g}) + \sum_{l\in\mathcal{L}} c_l \cdot \hat{v}_l(\boldsymbol{g}) \\
&= \sum_{i\in\mathcal{I},l\in\mathcal{L}} \hat{g}_l(x_i)\,\hat{z}_{il}(\boldsymbol{g}) \\
&= \sum_{i\in\mathcal{I},l\in\mathcal{L}} g_l(x_i)\,\hat{z}_{il}(\boldsymbol{g}).
\end{aligned}$$

As a direct consequence, we have that $\boldsymbol{z}'$ is an optimal solution to the linear program defined by Eq. 5 under $\boldsymbol{g}$ and thus also under $\hat{\boldsymbol{g}}$. Moreover, using the definition of $\hat{\pi}(\boldsymbol{g})$, we also have that:

$$\sum_{i\in\mathcal{I}} \hat{g}_{\pi'_i(\boldsymbol{g})}(x_i) = \sum_{i\in\mathcal{I},l\in\mathcal{L}} \hat{g}_l(x_i)\,z'_{il} = \sum_{i\in\mathcal{I},l\in\mathcal{L}} g_l(x_i)\,\hat{z}_{il}(\boldsymbol{g}) = \sum_{i\in\mathcal{I}} g_{\hat{\pi}_i(\boldsymbol{g})}(x_i) \tag{27}$$

Further, using the definition of $\hat{\boldsymbol{g}}$, the complementary slackness conditions and Eq. 27, we have that:

$$\sum_{i\in\mathcal{I},l\in\mathcal{L}} |\hat{g}_l(x_i) - g_l(x_i)| = \sum_{i\in\mathcal{I}} \left[\hat{g}_{\pi'_i(\boldsymbol{g})}(x_i) - g_{\pi'_i(\boldsymbol{g})}(x_i)\right] = \sum_{i\in\mathcal{I}} \left[g_{\hat{\pi}_i(\boldsymbol{g})}(x_i) - g_{\pi'_i(\boldsymbol{g})}(x_i)\right]$$

Since $\pi'(\boldsymbol{g}) \in \hat{\Pi}(\hat{\boldsymbol{g}})$, this directly implies that, for any $\hat{\pi}(\boldsymbol{g}) \in \hat{\Pi}(\boldsymbol{g})$ and $\hat{\pi}(\hat{\boldsymbol{g}}) \in \hat{\Pi}(\hat{\boldsymbol{g}})$, it holds that

$$\sum_{i\in\mathcal{I},l\in\mathcal{L}} |\hat{g}_l(x_i) - g_l(x_i)| = \sum_{i\in\mathcal{I}} \left[g_{\hat{\pi}_i(\boldsymbol{g})}(x_i) - g_{\hat{\pi}_i(\hat{\boldsymbol{g}})}(x_i)\right]$$

Thus, from Lemma 5, it follows that the predicted probabilities $\hat{\boldsymbol{g}}$ must be the minimally modified predicted probabilities $\boldsymbol{g}'$ under which $\pi'(\boldsymbol{g}) \in \hat{\Pi}(\boldsymbol{g}')$. This concludes the proof.

Appendix B. Additional Details about the Synthetic Data Generation

In this section, we give additional details regarding the procedure we follow to generate synthetic refugee from aggregate statistics on demographic features of refugees and countries from various international organizations.

B.1. Aggregate Statistical Data

We aggregate statistics about refugees who migrated to the United States in 2022 from the UNHCR database (UNHCR, 2022), including country of origin, sex, age groups, and number of refugees allocated across ten states, namely California, Florida, Illinois, Maryland, Massachusetts, New Jersey, New York, Pennsylvania, Texas and Virginia. We aggregate statistics about the population of the above ten US states from the U.S. Census (Census Bureau, 2024), including age, sex, and education. We aggregate statistics about employment in the above ten US states for different age groups and sex from the U.S. Department of Labor (Bureau of Labor Statistics, 2024). We aggregate statistics about the foreign-born population in the above ten US states from the Migration Policy Institute (Migration Policy Institute, 2024), including sex, country of origin and employment. Additionally, we aggregate statistics about the population of different countries from the World Bank (World Bank, 2024), including age, sex, and education. We standardize statistics on age groups, country of origin, and education levels across data sources. The age groups are *20-24*, *25-34*, *35-44*, *45-64*, and *65-100*. The levels of education are *Primary or less*, *Secondary*, and *Tertiary*. The country of origin is classified into the following regions: *Africa*, *Asia*, *Europe*, *Latin America*, *Northern America*, and *Oceania*. Table 2 summarizes all the quantities used in our data generation process, which are derived from the above aggregate statistics.

Symbol	Meaning
$\tau(c, s)$	The proportion of refugees with country of origin c and sex s
$\tau(a \mid c)$	The proportion of the population in age group a in country of origin c
$\tau(e \mid c)$	The proportion of the population with level of education e in country of origin c
$n(a, e, s \mid l)$	The number of people in state l in age group a, with level of educational e, and sex s
$\tau(a \mid l, w = 1)$	The proportion of the employed population in state l in age group a
$\tau(s \mid l, w = 1)$	The proportion of the employed population in state l of sex s
$\tau(w = 1 \mid l, f = 1)$	The proportion of the foreign-born population in state l employed.
$\tau(c \mid l, f = 1)$	The proportion of the foreign-born population in state l from country of origin c
$\tau(c \mid l, w = 1, f = 1)$	The proportion of the employed, foreign-born population in state l from country of origin c
$\tau(e \mid l, f = 1)$	The proportion of the foreign-born population in state l with level of education e
$\tau(e \mid l, w = 1, f = 1)$	The proportion of the employed, foreign-born population in state l with level of education e
$\tau(s \mid l, f = 1)$	The proportion of the foreign-born population in state l of sex s

Table 2: Quantities used in our data generation process, which are derived from publicly available aggregate statistics. The symbol $w = 1$ indicates that the given statistics pertain to the employed population, and $f = 1$ indicates that the statistics pertain to the foreign-born population. If both $w = 1$ and $f = 1$ are present, the statistics refer to the employed foreign-born population.

B.2. Data Generation Process

The attributes of each refugee i — age group $a_i \in \mathcal{A}$, country of origin $c_i \in \mathcal{C}$, level of education $e_i \in \mathcal{E}$ and sex $s_i \in \mathcal{S}$—are drawn according to different categorical distributions. First, we draw the country of origin c_i and the sex s_i based on the joint categorical distribution parameterized by proportions $\tau(c, s)$. The age group $A = a_i$ and level of education $E = e_i$ are then drawn per individual depending on their country of origin c_i:

$$c_i, s_i \sim Cat(\tau(c, s)), \quad a_i \sim Cat(\tau(a \,|\, c_i)), \quad e_i \sim Cat(\tau(e \,|\, c_i)) \tag{28}$$

Employment probability $p_{i,l}$ of refugee i in location l follows a beta distribution $\mathcal{B}_i$ with a mean of $\mu_l(a_i, c_i, e_i, s_i)$—the average employment probability of refugees with features a_i, c_i, e_i, s_i in state l—and a fixed variance σ^2. The variance σ^2 is set to 0.001 and the mean $\mu_l(a, c, e, s)$ is chosen to match the marginal statistics $\mu_l(a)$, $\mu_l(c)$, $\mu_l(e)$ and $\mu_l(s)$ of the general population in the US by solving the following optimization problem:

$$\begin{aligned}
\min_{\{\mu_l(a,c,e,s)\}} \quad & \frac{1}{|\mathcal{C} \times \mathcal{E} \times \mathcal{S}|} \sum_{c',e',s'} (\mu_l(a, c', e', s') - \mu_l(a))^2 \\
& + \frac{1}{|\mathcal{A} \times \mathcal{E} \times \mathcal{S}|} \sum_{a',e',s'} (\mu_l(a', c, e', s') - \mu_l(c))^2 \\
& + \frac{1}{|\mathcal{A} \times \mathcal{C} \times \mathcal{S}|} \sum_{a',c',s'} (\mu_l(a', c', e, s') - \mu_l(e))^2 \\
& + \frac{1}{|\mathcal{A} \times \mathcal{C} \times \mathcal{E}|} \sum_{a',c',e'} (\mu_l(a', c', e', s) - \mu_l(s))^2 \\
\text{subject to} \quad & \mu_l(a)(1 - \rho_a) \leq \sum_{c',e',s'} \mu_l(a, c', e', s') \cdot \tau(c', e', s' \,|\, l, f = 1) \leq \mu_l(a)(1 + \rho_a) \\
& \mu_l(c)(1 - \rho_c) \leq \sum_{a',e',s'} \mu_l(a', c, e', s') \cdot \tau(a', e', s' \,|\, l, f = 1) \leq \mu_l(c)(1 + \rho_c) \\
& \mu_l(e)(1 - \rho_e) \leq \sum_{a',c',s'} \mu_l(a', c', e, s') \cdot \tau(a', c', s' \,|\, l, f = 1) \leq \mu_l(e)(1 + \rho_e) \\
& \mu_l(s)(1 - \rho_s) \leq \sum_{a',c',e'} \mu_l(a', c', e', s) \cdot \tau(a', c', e' \,|\, l, f = 1) \leq \mu_l(s)(1 + \rho_s) \\
& \mu_l^L(a, c, e, s)(1 - \rho_b) \leq \mu_l(a, c, e, s) \leq \mu_l^U(a, c, e, s)(1 + \rho_b).
\end{aligned} \tag{29}$$

Here, $\tau(a, c, e \,|\, l, f = 1)$ denotes the proportion of the foreign-born population in age group a, from country of origin c, and with level of education e in location l. The proportions $\tau(a, c, s \,|\, l, f = 1)$, $\tau(a, e, s \,|\, l, f = 1)$, and $\tau(c, e, s \,|\, l, f = 1)$ are defined analogously. Under some assumptions, we can derive the above proportions from the aggregate statistics we gathered:

- $\tau(c, e, s \,|\, l, f = 1) = \tau(c \,|\, l, f = 1) \cdot \tau(e, s \,|\, l)$
- $\tau(a, e, s \,|\, l, f = 1) = \tau(a, e, s \,|\, l)$
- $\tau(a, c, s \,|\, l, f = 1) = \tau(c \,|\, l, f = 1) \cdot \tau(a, s \,|\, l)$

- $\tau(a, c, e \,|\, l, f = 1) = \tau(c \,|\, l, f = 1) \cdot \tau(a, e \,|\, l)$

The first assumption is that the joint distribution of the employed population in age group a, with level of education e, and sex s would be similar to that of the overall population with the same features across the states. The second assumption is that the country of origin c is independent of a, e and s for the employed population across the states.

In Eq. 29, the hyperparameters ρ_a, ρ_c, ρ_e, ρ_s control the "looseness" of the conditions that make $\mu_l(a, c, e, s)$ follow the marginal statistics $\mu_l(a)$, $\mu_l(c)$, $\mu_l(e)$, $\mu_l(s)$, respectively, based on the proportion of each refugee group. We set $\rho_a = 0.5$, $\rho_c = 0.0$, $\rho_e = 0.1$, and $\rho_s = 0.0$. The possible range of the solution is decided by another hyperparameter ρ_b as well as $\mu_l^L(a, c, e, s) = \max(\mu_l(a), \mu_l(c), \mu_l(e), \mu_l(s))$ and $\mu_l^U(a, c, e, s) = \min(\mu_l(a), \mu_l(c), \mu_l(e), \mu_l(s))$. This prevents the solver finding an extreme solution for $\mu_l(a, c, e, s)$ that is far from marginal statistics. In our generation process ρ_b is set to 0.6.

Marginal statistics $\mu_l(a)$, $\mu_l(c)$, $\mu_l(e)$ and $\mu_l(s)$ are computed based on the available aggregate statistics as follows. Marginal statistics $\mu_l(a)$ (*i.e.*, the proportion of the foreign-born population in state l in age group a employed) is computed as follows:

$$\begin{aligned}\mu_l(a) = \tau(w = 1|a, l, f = 1) &= \frac{\tau(a \,|\, l, w = 1, f = 1) \cdot \tau(w = 1 \,|\, l, f = 1)}{\tau(a \,|\, l, f = 1)} \\ &= \frac{\tau(a \,|\, l, w = 1) \cdot \tau(w = 1 \,|\, l, f = 1)}{\tau(a \,|\, l)},\end{aligned}$$

where $\tau(a \,|\, l) = \frac{\sum_{e' \in \mathcal{E}, s' \in \mathcal{S}} n(a, e', s' \,|\, l)}{\sum_{a' \in \mathcal{A}, e' \in \mathcal{E}, s' \in \mathcal{S}} n(a', e', s' \,|\, l)}$. Here, we assume that the age distribution of the employed, foreign-born population would be similar to that of total employed population across the states (*i.e.*, $\tau(a \,|\, l, w = 1, f = 1) \approx \tau(a \,|\, l, w = 1)$), and the age distribution of the foreign-born population would be similar to that of the total population across the states (*i.e.*, $\tau(a \,|\, l, f = 1) \approx \tau(a \,|\, l)$). Marginal statistics $\mu_l(c)$ (*i.e.*, the proportion of the foreign-born population in state l from country of origin c employed) is computed as follows:

$$\mu_l(c) = \tau(w = 1|c, l, f = 1) = \frac{\tau(c \,|\, l, w = 1, f = 1) \cdot \tau(w = 1 \,|\, l, f = 1)}{\tau(c \,|\, l, f = 1)}.$$

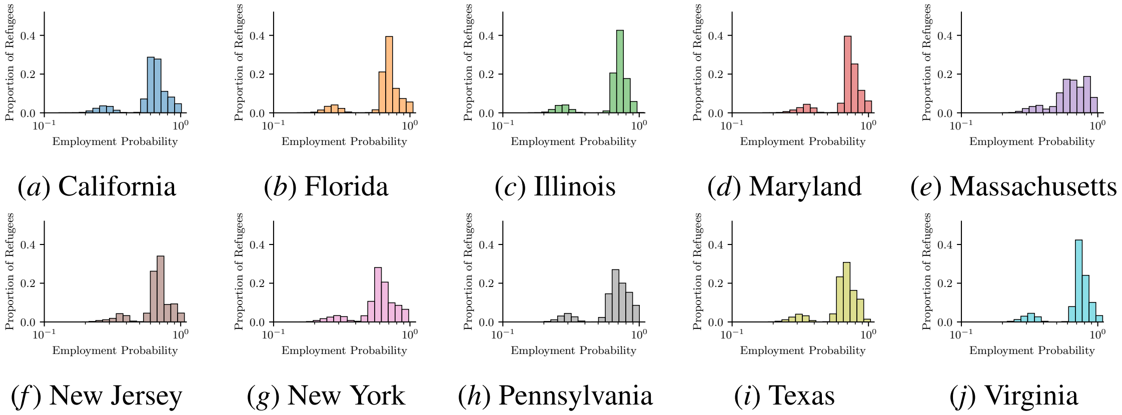
Marginal statistics $\mu_l(e)$ (*i.e.*, the proportion of the foreign-born population in state l with level of education e employed) is computed as follows:

$$\mu_l(e) = \tau(w = 1|e, l, f = 1) = \frac{\tau(e \,|\, l, w = 1, f = 1) \cdot \tau(w = 1 \,|\, l, f = 1)}{\tau(e \,|\, l, f = 1)}.$$

Marginal statistics $\mu_l(s)$ (*i.e.*, the proportion of the foreign-born population in state l of sex s employed) is computed as follows:

$$\begin{aligned}\mu_l(g) = \tau(w = 1|s, l, f = 1) &= \frac{\tau(s \,|\, l, w = 1, f = 1) \cdot \tau(w = 1 \,|\, l, f = 1)}{\tau(s \,|\, l, f = 1)} \\ &= \frac{\tau(s \,|\, l, w = 1) \cdot \tau(w = 1 \,|\, l, f = 1)}{\tau(s \,|\, l, f = 1)},\end{aligned}$$

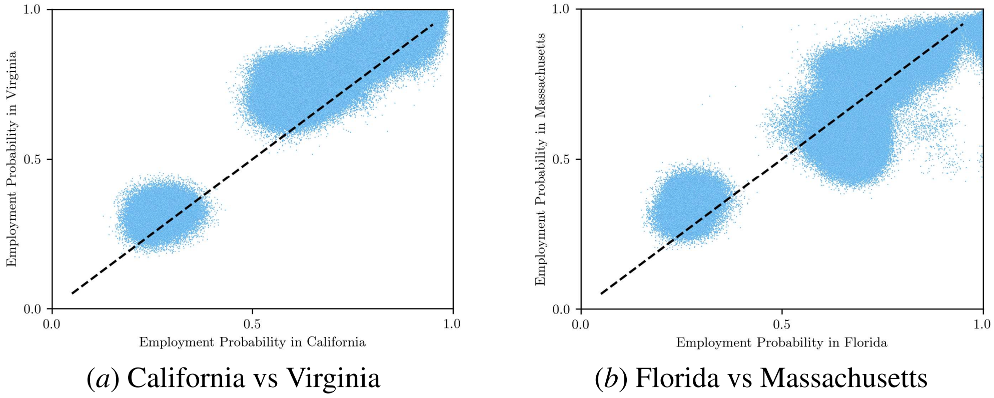
where we assume that the sex distribution of the employed, foreign-born population would be similar to that of total employed population across the states (*i.e.*, $\tau(s \,|\, l, w = 1, f = 1) = \tau(s \,|\, l, w = 1)$).



(*a*) California (*b*) Florida (*c*) Illinois (*d*) Maryland (*e*) Massachusetts

(*f*) New Jersey (*g*) New York (*h*) Pennsylvania (*i*) Texas (*j*) Virginia

Figure 5: Empirical distribution of employment probability of refugees for all states, computed using 500,000 synthesized refugees. The distributions are plotted using exponential binning.



(*a*) California vs Virginia (*b*) Florida vs Massachusetts

Figure 6: Per-refugee employment probability for two pairs of locations and $500,000$ synthesized refugees. The (dashed) identity line (*i.e.*, $y = x$) indicates equal probabilities between two states.

B.3. Distribution of Generated Data

Using the generation process described in Appendix B.2, we create $5{,}000$ synthetic pools of refugees to be resettled to $k = 10$ locations. Each pool contains $n = 100$ synthetic refugees and the employment probability of each refugee is sampled from the beta distribution derived from their demographic features. Figure 5 shows the empirical distribution of employment probability for all states, computed using the resulting $500{,}000$ synthesized refugees. While these empirical distributions are similar, the employment probability per refugee varies across locations, as shown in Figure 6 for two pairs of states.

Appendix C. Additional Implementation Details

In this section, we offer additional implementation details about deep learning model h used in our postprocessing framework.

Training Data. The minimally modified employment probabilities $\breve{g}$ are computed from the given placement decisions made by the default policy $\tilde{\pi}(\boldsymbol{x}, w)$ and the predicted employment probabilities $\boldsymbol{g}$ by solving Eq. 13 using the linear program solver in `Scipy` package.

Architecture of the Postprocessing Deep Learning Model h. Our deep learning model h consists of a prediction probability projection layer, a capacity projection layer, N layers of Transformer encoders, and an embedding projection layer. Note that all projection layers consist of two linear projections with ReLU activation. A prediction probability projection layer projects $\boldsymbol{g}$, the expected employment probability of each refugee in $k = 10$ locations, to hidden dimension $d = 128$. Similarly, a capacity projection layer projects the capacity of all locations to the same hidden dimension. This projected capacity information is then added to the projected employment probabilities for each refugee. The added data pass through $N = 2$ layers of Transformer encoder (Vaswani et al., 2017), where each Transformer encoder layer has 1 head and 128 hidden dimensions. Note that, the layers of Transformer encoder exclude positional encoding so that they become agnostic to the number or order of the refugees in each pool. The processed data is projected onto a $k = 10$ dimensional space to predict the difference $\breve{\boldsymbol{g}} - \boldsymbol{g}$. The final output of the model $h(\boldsymbol{g})$ is obtained by adding this predicted difference to the original model input $\boldsymbol{g}$.

Training. In all our experiments, we employ quadratic loss and the AdamW optimizer for the optimization process. Our training batches consist of 16 pools of refugees, and the training duration is set for 50 epochs. We start with an initial learning rate of 0.001 and apply an exponential learning rate scheduler with a decay factor of $\gamma = 0.9$ to adjust the learning rate over time. It takes 7 seconds of training time per epoch in our setting.

Hardware Setup. We use a GPU server equipped with Intel Xeon Platinum 8268 CPU @ 2.90GHz, 376 GB memory and NVIDIA A100 GPUs. A single GPU is used in each experiment.

Noise Level (w)	$\tilde{\pi}(\boldsymbol{x}, w)$	$\hat{\pi}(\boldsymbol{p})$	$\hat{\pi}(\boldsymbol{g})$	$\hat{\pi}(\breve{\boldsymbol{g}})$	$\hat{\pi}(h(\boldsymbol{g}))$
0.000	0.7200	0.7200	0.6976	0.8161	**0.7077±0.0009**
0.125	0.7129	0.7200	0.6976	0.8180	**0.7074±0.0003**
0.250	0.7032	0.7200	0.6976	0.8188	**0.7072±0.0012**
0.375	0.6956	0.7200	0.6976	0.8195	**0.7071±0.0015**
0.500	0.6892	0.7200	0.6976	0.8242	**0.7038±0.0012**
0.625	0.6794	0.7200	0.6976	0.8262	**0.7007±0.0012**
0.750	0.6715	0.7200	0.6976	0.8288	**0.7002±0.0006**
0.875	0.6659	0.7200	0.6976	0.8293	**0.6981±0.0005**
1.000	0.6532	0.7200	0.6976	0.8319	0.6975±0.0005

Table 3: Comparison of the per-pool average realized utility of algorithmic policies $\tilde{\pi}(\boldsymbol{x}, w)$, $\hat{\pi}(\boldsymbol{p})$, $\hat{\pi}(\boldsymbol{g})$, $\hat{\pi}(\breve{\boldsymbol{g}})$, and $\hat{\pi}(h(\boldsymbol{g}))$ against the default policy $\tilde{\pi}(\boldsymbol{x}, w)$ across 500 test set pools under various noise levels w (higher numbers indicate better performance). The numbers of $\hat{\pi}(h(\boldsymbol{g}))$ are averaged results over 5 runs with their standard deviation. Bold numbers indicate increase in average realized utility by $\hat{\pi}(h(\boldsymbol{g}))$ compared to $\hat{\pi}(\boldsymbol{g})$, while underlined numbers indicates the higher average realized utility of $\hat{\pi}(h(\boldsymbol{g}))$ compared to $\tilde{\pi}(\boldsymbol{x}, w)$.

Appendix D. Additional Experimental Results

In this section, we provide additional experimental results that are omitted from the main paper due to space limitations.

D.1. Experimental Results on Realized Utility

In Section 6, we could verify that the algorithmic policy $\hat{\pi}(\breve{\boldsymbol{g}})$ reduces the risk of harm compared to algorithmic policy $\hat{\pi}(\boldsymbol{g})$. Table 3 summarizes the per-pool average realized utility of the algorithmic policies given labels $\{y_i(l)\}_{l \in \mathcal{L}, i \in \mathcal{I}}$ indicating whether a refugee would find employment at each state l soon after relocation. As expected from Eq. 9, we observe that the unrealizable algorithmic policy $\hat{\pi}(\breve{\boldsymbol{g}})$ achieves higher average realized utility compared to default policy $\tilde{\pi}(\boldsymbol{x}, w)$ across all noise levels. Additionally, the postprocessing algorithmic policy $\hat{\pi}(h(\boldsymbol{g}))$ successfully achieves higher average realized utility compared to the algorithmic policy $\hat{\pi}(\boldsymbol{g})$ that maximizes the predicted utility under the predicted employment probabilities $\boldsymbol{g}$.

Number of Refugees (n)	Post-processing (Eq. 12)		Training Deep Learning Model h	
	Time (s)	Memory (MB)	Time (s)	Memory (MB)
10	0.006±0.000	0.229	0.063±0.001	0.006
100	0.034±0.001	0.285	0.065±0.003	0.023
1000	1.443±0.134	0.881	0.080±0.003	0.272

Table 4: Per-pool time and memory cost for solving the post-processing problem in Eq. 12 and training the deep learning model h with varying numbers of refugees in a pool (n) while the number of locations is fixed to 10. The running times are averaged over 5 runs, with standard deviations reported.

Number of Locations (m)	Post-processing (Eq. 12)		Training Deep Learning Model h	
	Time (s)	Memory (MB)	Time (s)	Memory (MB)
2	0.012±0.000	0.231	0.064±0.003	0.023
5	0.019±0.000	0.270	0.064±0.002	0.023
10	0.034±0.001	0.285	0.065±0.003	0.023

Table 5: Per-pool time and memory cost for solving the post-processing problem in Eq. 12 and training the deep learning model h with varying numbers of locations (m) while the number of refugees in a pool is fixed to 100. The running times are averaged over 5 runs, with standard deviations reported.

D.2. Empirical Running Time and Peak-Memory Usage of Post-Processing Framework

Table 4 and Table 5 present empirical running time and memory consumption per pool of refugees for finding the minimally modified predicted employment probabilities in Eq. 12 and training the deep learning model h. These results are shown for varying numbers of refugees in a pool (n) and locations (m), respectively. We observe that the number of refugees in a pool affects the time and memory costs; however, these costs remain manageable, even with a large number of refugees, such as $n = 1000$. The number of locations does not significantly influence either the time or memory costs within our framework. Notably, our method is not resource intensive and requires minimal computational resources. It maintains a minimal memory foot print, using less than one gigabyte of memory across all experiments, and completes both solving the linear program and updating the deep learning model in under two seconds per instance.

Noise Level (w)	$\hat{\pi}(\boldsymbol{p})$	$\hat{\pi}(\boldsymbol{g})$	$\hat{\pi}(\breve{\boldsymbol{g}})$	$\hat{\pi}(h(\boldsymbol{g}))$
0.000	0.00	75.00	0.00	**69.24±0.96**
0.125	33.20	64.60	0.00	**58.88±0.63**
0.250	23.60	55.00	0.00	**45.44±0.59**
0.375	17.60	48.20	0.00	**38.00±1.25**
0.500	13.60	41.00	0.00	**33.92±0.68**
0.625	11.40	33.40	0.00	**30.64±1.04**
0.750	9.20	24.80	0.00	**22.88±0.56**
0.875	8.20	23.40	0.00	23.40±0.77
1.000	6.00	15.60	0.00	16.04±0.50

Table 6: Comparison of the percentage (%) of pools counterfactually harmed by algorithmic policies $\hat{\pi}(\boldsymbol{p})$, $\hat{\pi}(\boldsymbol{g})$, $\hat{\pi}(\breve{\boldsymbol{g}})$, and $\hat{\pi}(h(\boldsymbol{g}))$ against the default policy $\tilde{\pi}(\boldsymbol{x}, w)$ across 500 test set pools under various noise levels w (lower numbers indicate better performance). The numbers of $\hat{\pi}(h(\boldsymbol{g}))$ are averaged results over 5 runs with their standard deviation. Bold numbers indicate reductions in counterfactually harmed pools by $\hat{\pi}(h(\boldsymbol{g}))$ compared to $\hat{\pi}(\boldsymbol{g})$.

Noise Level (w)	$\tilde{\pi}(\boldsymbol{x}, w)$	$\hat{\pi}(\boldsymbol{p})$	$\hat{\pi}(\boldsymbol{g})$	$\hat{\pi}(\breve{\boldsymbol{g}})$	$\hat{\pi}(h(\boldsymbol{g}))$
0.000	0.7200	0.7200	0.6991	0.8156	**0.7078±0.0004**
0.125	0.7129	0.7201	0.6989	0.8179	**0.7072±0.0003**
0.250	0.7032	0.7192	0.6976	0.8192	$\underline{\mathbf{0.7074 \pm 0.0003}}$
0.375	0.6956	0.7202	0.6984	0.8220	$\underline{\mathbf{0.7064 \pm 0.0008}}$
0.500	0.6892	0.7210	0.6974	0.8244	$\underline{\mathbf{0.7044 \pm 0.0005}}$
0.625	0.6794	0.7180	0.6967	0.8248	$\underline{\mathbf{0.7005 \pm 0.0008}}$
0.750	0.6715	0.7201	0.6982	0.8290	$\underline{\mathbf{0.7001 \pm 0.0004}}$
0.875	0.6659	0.7187	0.6964	0.8298	$\underline{\mathbf{0.6969 \pm 0.0001}}$
1.000	0.6532	0.7184	0.6967	0.8309	$\underline{0.6965 \pm 0.0001}$

Table 7: Comparison of the expected counterfactual utility of algorithmic policies $\tilde{\pi}(\boldsymbol{x}, w)$, $\hat{\pi}(\boldsymbol{p})$, $\hat{\pi}(\boldsymbol{g})$, $\hat{\pi}(\breve{\boldsymbol{g}})$, and $\hat{\pi}(h(\boldsymbol{g}))$ against the default policy $\tilde{\pi}(\boldsymbol{x}, w)$ across 500 test set pools under various noise levels w (higher numbers indicate better performance). The numbers of $\hat{\pi}(h(\boldsymbol{g}))$ are averaged results over 5 runs with their standard deviation. Bold numbers indicate increase in expected counterfactual utility by $\hat{\pi}(h(\boldsymbol{g}))$ compared to $\hat{\pi}(\boldsymbol{g})$, while underlined numbers indicates the higher expected counterfactual utility of $\hat{\pi}(h(\boldsymbol{g}))$ compared to $\tilde{\pi}(\boldsymbol{x}, w)$.

D.3. Full Experimental Results on Counterfactual Utility

Table 6 presents the full results for the percentage (%) of pools in the test that are counterfactually harmed by each algorithmic policy under different noise levels w, and Table 7 presents the full results for the per-pool expected counterfactual utilities of each algorithmic policy under different noise levels w. The numbers of $\hat{\pi}(h(\boldsymbol{g}))$ are average results over 5 runs with their standard deviation.

β	Noise Level (w)	0.000	0.125	0.250	0.375	0.500	0.625	0.750	0.875	1.000
	$\tilde{\pi}(\boldsymbol{x}, w)$	0.7200	0.7129	0.7032	0.6956	0.6892	0.6794	0.6715	0.6659	0.6532
	$\hat{\pi}(\boldsymbol{p})$	0.7200	0.7201	0.7192	0.7202	0.7210	0.7180	0.7201	0.7187	0.7184
0.1	$\hat{\pi}(\boldsymbol{g})$	0.7168	0.7166	0.7157	0.7169	0.7169	0.7155	0.7163	0.7145	0.7149
	$\hat{\pi}(\breve{\boldsymbol{g}})$	0.7558	0.7686	0.7803	0.7913	0.8032	0.8106	0.8223	0.8306	0.8389
	$\hat{\pi}(h(\boldsymbol{g}))$	**0.7197**	**0.7194**	**0.7192**	**0.7196**	**0.7199**	**0.7172**	**0.7170**	**0.7150**	**0.7153**
0.2	$\hat{\pi}(\boldsymbol{g})$	0.7146	0.7145	0.7137	0.7144	0.7147	0.7135	0.7135	0.7117	0.7126
	$\hat{\pi}(\breve{\boldsymbol{g}})$	0.7797	0.7867	0.7946	0.8018	0.8101	0.8163	0.8242	0.8299	0.8371
	$\hat{\pi}(h(\boldsymbol{g}))$	**0.7188**	**0.7190**	**0.7183**	**0.7191**	**0.7193**	**0.7163**	**0.7146**	**0.7118**	0.7126
0.3	$\hat{\pi}(\boldsymbol{g})$	0.7125	0.7122	0.7117	0.7125	0.7130	0.7116	0.7121	0.7098	0.7103
	$\hat{\pi}(\breve{\boldsymbol{g}})$	0.7905	0.7953	0.8018	0.8084	0.8140	0.8192	0.8258	0.8300	0.8356
	$\hat{\pi}(h(\boldsymbol{g}))$	**0.7170**	**0.7169**	**0.7163**	**0.7165**	**0.7169**	**0.7132**	**0.7123**	**0.7099**	0.7103
0.4	$\hat{\pi}(\boldsymbol{g})$	0.7106	0.7104	0.7086	0.7105	0.7098	0.7089	0.7090	0.7081	0.7084
	$\hat{\pi}(\breve{\boldsymbol{g}})$	0.7984	0.8029	0.8070	0.8122	0.8180	0.8218	0.8276	0.8300	0.8350
	$\hat{\pi}(h(\boldsymbol{g}))$	**0.7140**	**0.7144**	**0.7133**	**0.7129**	**0.7127**	**0.7092**	**0.7094**	**0.7082**	**0.7086**
0.5	$\hat{\pi}(\boldsymbol{g})$	0.7050	0.7046	0.7037	0.7048	0.7044	0.7032	0.7048	0.7029	0.7026
	$\hat{\pi}(\breve{\boldsymbol{g}})$	0.8069	0.8098	0.8129	0.8165	0.8207	0.8230	0.8283	0.8302	0.8335
	$\hat{\pi}(h(\boldsymbol{g}))$	**0.7108**	**0.7104**	**0.7096**	**0.7098**	**0.7090**	**0.7055**	**0.7050**	0.7029	**0.7028**
0.6	$\hat{\pi}(\boldsymbol{g})$	0.6991	0.6989	0.6976	0.6984	0.6974	0.6967	0.6982	0.6964	0.6967
	$\hat{\pi}(\breve{\boldsymbol{g}})$	0.8156	0.8179	0.8192	0.8220	0.8244	0.8248	0.8290	0.8298	0.8309
	$\hat{\pi}(h(\boldsymbol{g}))$	**0.7078**	**0.7072**	**0.7074**	**0.7064**	**0.7044**	**0.7005**	**0.7001**	**0.6969**	0.6965
0.7	$\hat{\pi}(\boldsymbol{g})$	0.6914	0.6908	0.6901	0.6908	0.6913	0.6896	0.6906	0.6887	0.6899
	$\hat{\pi}(\breve{\boldsymbol{g}})$	0.8240	0.8253	0.8247	0.8269	0.8282	0.8270	0.8292	0.8288	0.8284
	$\hat{\pi}(h(\boldsymbol{g}))$	**0.7050**	**0.7041**	**0.7037**	**0.7021**	**0.7004**	**0.6974**	**0.6960**	**0.6898**	0.6892
0.8	$\hat{\pi}(\boldsymbol{g})$	0.6862	0.6857	0.6850	0.6852	0.6860	0.6839	0.6853	0.6832	0.6839
	$\hat{\pi}(\breve{\boldsymbol{g}})$	0.8297	0.8292	0.8284	0.8290	0.8295	0.8275	0.8289	0.8280	0.8266
	$\hat{\pi}(h(\boldsymbol{g}))$	**0.7025**	**0.7020**	**0.7008**	**0.6990**	**0.6976**	**0.6951**	**0.6925**	**0.6859**	**0.6846**
0.9	$\hat{\pi}(\boldsymbol{g})$	0.6839	0.6835	0.6827	0.6827	0.6835	0.6814	0.6827	0.6808	0.6817
	$\hat{\pi}(\breve{\boldsymbol{g}})$	0.8309	0.8302	0.8290	0.8295	0.8296	0.8270	0.8281	0.8275	0.8257
	$\hat{\pi}(h(\boldsymbol{g}))$	**0.6996**	**0.6988**	**0.6972**	**0.6969**	**0.6954**	**0.6888**	**0.6852**	**0.6826**	0.6816

Table 8: Expected counterfactual utility achieved by the algorithmic policies $\hat{\pi}(\boldsymbol{p})$, $\hat{\pi}(\boldsymbol{g})$ and $\hat{\pi}(h(\boldsymbol{g}))$ in comparison with the expected realized utility achieved by the default policy $\tilde{\pi}(\boldsymbol{x}, w)$ across all pools in the test set for different β values under varying noise levels. For $\hat{\pi}(h(\boldsymbol{g}))$, the results are averaged over 5 runs.

In both tables, we can observe that the algorithmic policy that uses our postprocessing algorithm, $\hat{\pi}(h(\boldsymbol{g}))$, reduces (maintains) harm in high (low) noise levels compared to the algorithmic policy that uses the predicted employment probability, $\hat{\pi}(\boldsymbol{g})$. In Table 6, the percentage of pools counterfactually harmed by $\hat{\pi}(\breve{\boldsymbol{g}})$ is 0% for all noise level, as expected from Eq. 9. In Table 7, $\hat{\pi}(\breve{\boldsymbol{g}})$ consistently exhibits higher expected counterfactual utility across all noise levels. Furthermore, $\hat{\pi}(h(\boldsymbol{g}))$ shows higher expected counterfactual utility compared to the default policy $\tilde{\pi}(\boldsymbol{x}, w)$ for high noise levels (*i.e.*, $w \geq 0.25$).

D.4. Full Experimental Results of Hyperparameter Analysis

Table 8 shows the full results of expected counterfactual utility achieved by the algorithmic policies $\hat{\pi}(\boldsymbol{p})$, $\hat{\pi}(\boldsymbol{g})$, $\hat{\pi}(\breve{\boldsymbol{g}})$ and $\hat{\pi}(h(\boldsymbol{g}))$ in comparison with the expected realized utility achieved by the default policy $\tilde{\pi}(\boldsymbol{x}, w)$ for different β values under varying noise levels. We can observe that the postprocessing algorithmic policy offers greater expected counterfactual utility than $\hat{\pi}(\boldsymbol{g})$ in most of the cases.