Efficient Public Health Intervention Planning Using Decomposition-Based Decision-Focused Learning

Sanket Shah*
Harvard University
sanketshah@g.harvard.edu

Arun Suggala
Google Research India
arunss@google.com

Milind Tambe
Google Research
milindtambe@google.com

Aparna Taneja
Google Research India
aparnataneja@google.com

ABSTRACT

The declining participation of beneficiaries over time is a key concern in public health programs. A popular strategy for improving retention is to have health workers 'intervene' on beneficiaries at risk of dropping out. However, the availability and time of these health workers are limited resources. As a result, there has been a line of research on optimizing these limited intervention resources using Restless Multi-Armed Bandits (RMABs). The key technical barrier to using this framework in practice lies in estimating the beneficiaries' RMAB parameters from historical data. Recent research on Decision-Focused Learning (DFL) has shown that estimating parameters that maximize beneficiaries' cumulative returns rather than predictive accuracy, is essential to good performance.

Unfortunately, these gains come at a high computational cost because of the need to solve and evaluate the RMAB in each DFL training step. Consequently, past approaches may not be sustainable for the NGOs that manage such programs in the long run, given that they operate under resource constraints. In this paper, we provide a principled way to exploit the structure of RMABs to speed up DFL by decoupling intervention planning for different beneficiaries. We use real-world data from an Indian NGO, ARMMAN, to show that our approach is up to two orders of magnitude faster than the state-of-the-art approach while also yielding superior model performance. This enables computationally efficient solutions, giving NGOs the ability to deploy such solutions to serve potentially millions of mothers, ultimately advancing progress toward UNSDG 3.1.

KEYWORDS

AI for Social Good, Public Health, Predict-Then-Optimize, Decision-Focused Learning, Restless Multi-Armed Bandits, Optimization

ACM Reference Format:
Sanket Shah, Arun Suggala, Milind Tambe, and Aparna Taneja. 2024. Efficient Public Health Intervention Planning Using Decomposition-Based Decision-Focused Learning. In *Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), Auckland, New Zealand, May 6 – 10, 2024*, IFAAMAS, 12 pages.

1 INTRODUCTION

A pervasive challenge faced by public health programs is one of beneficiary retention. To combat the declining engagement of beneficiaries over time, a common strategy has been to use 'interventions' (e.g., personalized service calls) to encourage participation and address concerns. This has been employed in a variety of domains such as medication adherence [20], chronic illness management [17], treatment prioritization [7], and mobile health [21]. However, despite their effectiveness, such interventions are expensive and, thus, effectively limited resources. Consequently, optimizing the selection of beneficiaries for these interventions is crucial.

Towards this end, there has been a recent line of research on using Restless Multi-Armed Bandits (RMABs) [16, 30, 31] to optimize intervention resources in these domains. In the RMAB framework, each beneficiary's adherence to the program is modeled as a Markov Decision Process (MDP). The goal, then, is to design policies that choose K out of N beneficiaries for health worker intervention in each timestep such that the overall adherence of all beneficiaries is maximized. However, the key technical barrier to using this framework in practice lies in estimating the beneficiaries' MDP parameters, which are essential for determining these intervention policies. To address this gap, past work relies on *predicting* these parameters using historical data and beneficiary demographics.

An essential component of an effective predictive pipeline in the public health domain involves using 'Decision-Focused Learning' (DFL) [11, 19, 32], a way to incorporate intervention planning into the training loop in order to create models that maximize beneficiary adherence directly (cf. predictive accuracy). Both simulated experiments [18, 29] and a field study [25] have shown that models trained using DFL outperform those trained using traditional supervised learning pipelines. However, the improved performance of DFL comes at a heavy computational cost—incorporating decision-making into the training pipeline requires solving, evaluating, and differentiating through intervention planning at every training step.

To reduce the computational overhead of using DFL, the state-of-the-art approach [29] uses the popular Whittle Index heuristic [30] to simplify intervention planning. This heuristic decomposes the task of creating a good policy for *all the beneficiaries* to one of deciding whether to act on *individual beneficiaries* in a simplified version of the RMAB problem. However, while this speeds up the *planning* of a good policy, *evaluating* the resulting policy requires repeatedly simulating the outcome of the policy. Yet, such evaluation is a crucial aspect of the DFL training pipeline. Indeed, as we show in Section 5, this either results in evaluations with high variance and, as a result, suboptimal learning (for a small number of simulations), or high cost (for a large number of simulations).

Instead, in this paper, we create a decomposition-based DFL approach that extends the ideas from the RMAB planning literature [14, 30] to both create *and evaluate* policies efficiently, *without the need for any simulations*. Specifically, we begin in Section 4.1 by showing how using the approach from Hawkins [14] to create decomposed policies leads to budget constraint violations in the

*Work done as an intern at Google Research India

Proc. of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2024), N. Alechina, V. Dignum, M. Dastani, J.S. Sichman (eds.), May 6 – 10, 2024, Auckland, New Zealand.

Figure 1: An mMitra beneficiary (courtesy of ARMMAN)

DFL setting. Rather, in Section 4.2, we propose an alternative approach and show how optimizing over a richer class of policies allows us to provably estimate the optimal beneficiary parameters in this setting. Finally, in Section 4.3, we show how to efficiently (in $O(N)$ time) incorporate this approach into the DFL pipeline by building on techniques from the DFL literature [3, 4].

To evaluate our approach, we use real-world data from ARMMAN, an Indian NGO, that leverages mobile health (mHealth) technology to promote healthy pregnancies. Specifically, we use secondary data from their mMitra program [6], which has successfully delivered vital preventive care information to *2.9 million* women, to build our domain. Notably, DFL [25] has been currently deployed for intervention planning in mMitra and has served around 250,000 beneficiaries so far. Then, in Section 5, we present the results of how our approach does against this existing approach (based on Wang et al. [29]) on both the real-world domain and a synthetic domain.

We show that our proposed method is up to 500x faster than the currently deployed approach, while also producing better-performing models (Table 1). Practically, this means that models that would take more than a day to train in the past can now be trained in minutes *with no loss in quality*. All in all, we believe that our contribution will allow more scalable learning for RMABs, and hopefully help ARMMAN and other NGOs move us one step closer to UN Sustainable Development Goal 3.1.

2 BACKGROUND

ARMMAN's mMitra Program. The UN Sustainable Development Goal (SDG) 3.1 aims to reduce the global maternal mortality ratio to below 70 per 100,000 live births by 2030. In line with this goal, ARMMAN uses mHealth technology to combat maternal and neonatal mortality in underprivileged communities across India. Specifically, ARMMAN's mMitra program delivers preventive care information on maternal and infant health through free automated voice calls to beneficiaries. Notably, ≈90% of mothers in the program fall below the World Bank's international poverty line [26]. Consequently, these weekly calls provide vital and timely information that would otherwise remain inaccessible to these women. However, despite the program's success, engagement wanes over time, with 22% of beneficiaries dropping out within just three months of enrollment [26]. To combat this, ARMMAN deploys health workers to conduct live service calls to encourage participation and address concerns. In the context of the mMitra program, our goal is to determine which subset of beneficiaries to select for these service calls on a weekly basis so as to maximize engagement.

Restless Multi-Armed Bandits. RMABs are an extension of the well-known multi-armed bandit framework to the case where the states of different arms evolve over time *regardless of whether they are pulled or not*. Concretely, each arm $i \in [N]$ of the RMAB is modeled as an MDP that is defined by the tuple $(\mathcal{S}_i, \mathcal{A}_i, T_i, R_i, \gamma)$ where $\mathcal{S}_i$ is the state space, $\mathcal{A}_i$ is the action space, $T_i, R_i : \mathcal{S}_i \times \mathcal{A}_i \times \mathcal{S}_i \to \mathbb{R}$ are the transition and reward functions, and γ is the discount factor.

Although the results presented in this paper extend to all RMABs, we make the following simplifications for ease of exposition:

- $\mathcal{S}_i := \mathcal{S} = \{0, \ldots, |\mathcal{S}|-1\}$ that denotes the degree of engagement with the public health program.
- $R_i := R(s) = \frac{s}{|\mathcal{S}|-1}$, the reward is directly proportional to the degree of engagement with the program.
- $\mathcal{A}_i = \mathcal{A} := \{0, 1\}$ that denotes whether a beneficiary is intervened on (1) or not (0).

An important point to note here is that, in large-scale public health interventions, we typically do not have enough data to estimate complex per-arm models, especially for the intervention action. As a result, each per-arm MDP (i.e., $|\mathcal{S}|$) is typically small.

The solution concept for RMABs is a policy $\pi : \mathcal{S}^N \to \mathcal{A}^N$ that satisfies a budget constraint $\sum_i \pi_i(s_i) \leq B$ where B is our budget. The optimal policy $\pi^\star$ for transitions $\boldsymbol{T}$ can then be written as:

$$\boldsymbol{\pi}^\star(\boldsymbol{T}) = \arg\max_{\boldsymbol{\pi}} J_{\boldsymbol{T}}(\boldsymbol{\pi}) \quad s.t. \quad \sum_{i=1}^{N} \pi_i(s_i) \leq B, \ \forall \boldsymbol{s} \in \mathcal{S}^N \tag{1}$$

where $J_{\boldsymbol{T}}(\boldsymbol{\pi}) = \mathbb{E}_{\tau \sim \boldsymbol{\pi}, \boldsymbol{T}}[R(s) + \gamma R(s') + \gamma^2 R(s'') + \ldots]$ is the expected return for trajectories τ generated using policy $\boldsymbol{\pi}$ and transitions $\boldsymbol{T}$.

In the RMAB above, *the only thing that is unknown* is the transition matrix $\boldsymbol{T}$ that determines beneficiaries' engagement and response to interventions. The challenge, then, is estimating $\boldsymbol{T}$.

Decision-Focused Learning (DFL). While the parameters in bandit problems are sometimes learned online, in public health settings this can be impractical because the programs are short and feedback infrequent. For example, ARMMAN's mMitra program runs for 72 weeks and beneficiaries are only called once a week. Moreover, we want to be able to intervene *as early as possible* to prevent beneficiaries from dropping out of the program. As a result, we instead estimate the transition matrices T from historical data, *offline*.

This has been modeled as a Predict-then-Optimize (PtO) problem in past work [25, 29] and involves three steps:

(1) **Predict Step:** First, we use the demographic features $\boldsymbol{x} = [x_1, \ldots, x_N]$ associated with each of the N beneficiaries (arms) to *predict* their transition matrices $\hat{T} = [\hat{T}_1, \ldots, \hat{T}_N] = [M_\theta(x_1), \ldots, M_\theta(x_N)]$ using a predictive model M_θ.
(2) **Optimize/Planning Step:** Next, we use these predicted transition matrices $\hat{T}$ to compute the optimal policy $\boldsymbol{\pi}^\star(\hat{T}) = \max_{\boldsymbol{\pi}} J_{\hat{T}}(\boldsymbol{\pi})$, where J is the expected return under policy $\boldsymbol{\pi}$.
(3) **Evaluation Step** Finally, we evaluate the policy $\boldsymbol{\pi}^\star(\hat{T})$ on the *true* historical transition probabilities T, i.e., $J_T(\boldsymbol{\pi}^\star(\hat{T}))$. We call this value the 'Decision Quality' (DQ) of the prediction $\hat{T}$.

The overall goal for DFL, then, is to learn a set of parameters $\theta^\star$ for the predictive model M_θ such that the final decision quality is maximized. With a slight abuse of notation where $M_\theta(\boldsymbol{x}) = [M_\theta(x_1), \ldots, M_\theta(x_N)]$, this can be written as:

$$\theta^\star = \arg\max_{\theta} \ \mathbb{E}_{\boldsymbol{x}, T \sim \mathcal{D}} \underbrace{\left[J_T(\boldsymbol{\pi}^\star(M_\theta(\boldsymbol{x}))) \right]}_{\ell_{\text{DFL}}(M_\theta(\boldsymbol{x}), T)} \tag{2}$$

This is different from a typical supervised learning problem in which the goal is to minimize a "standard" loss function, e.g., MSE:

$$\theta^\star = \arg\max_\theta \mathbb{E}_{\boldsymbol{x},T\sim\mathcal{D}} \underbrace{\left[||M_\theta(\boldsymbol{x}) - T||_2^2\right]}_{\ell_{\text{MSE}}(M_\theta(\boldsymbol{x}),T)}$$

3 RELATED WORK

DFL for RMABs. The closest related branch of the literature on solving problems similar to Eq. (2) is that of decision-focused model-based reinforcement learning [12, 13, 22, 28]. There, the goal is to estimate MDP parameters that lead to good downstream policies. However, while these approaches can *technically* be applied to the RMAB domain, the state space of RMABs is combinatorial in the number of arms N and known to be PSPACE-Hard [23] to solve.

To make solving Eq. (2) computationally tractable for RMABs, Wang et al. [29] propose an efficient approximate approach to planning that uses the popular Whittle Index-based policy [30]:

$$\boldsymbol{\pi}^\star \approx \boldsymbol{\pi}^{\text{WI}}(s) = \begin{cases} 1 & \text{WI}^i(s^i) \in \text{Top-B}(\text{WI}(\boldsymbol{s})) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

However, while $\boldsymbol{\pi}^{\text{WI}}$ only depends on the Whittle Indexes (that are calculated independently per arm), the Top-B policy still acts on the combinatorial state space $\boldsymbol{s} = [s_1, \ldots, s_N]$. As a result, evaluating $\boldsymbol{\pi}^{\text{WI}}$ requires using expensive Monte Carlo simulations. Instead, in this paper, we propose a novel and significantly cheaper way to approximate both policy creation *and evaluation*.

Decomposed RMAB Evaluation. The solution we present in Section 4 builds on foundational work in the planning literature [14, 30]. The Whittle Index heuristic itself is based on a relaxation of Eq. (1) that decomposes the combinatorial problem into N per-arm problems. However, in Section 4.1, we describe why existing methods lead to constraint violations in our DFL setting. Then, in Section 4.2, we show how to modify these ideas so that they *are* applicable and derive a novel solution method for the resulting formulation.

Multi-Model MDPs. Our solution in Section 4.2 requires coming up with a policy that maximizes the return with respect to one MDP (A) while having a bounded return with respect to a different MDP (B). This is a generalization of the popular "Constrained MDPs" framework [2] to the case where the MDPs A and B have different transition matrices in addition to different reward functions. The most directly related work to this is that of "Concurrent MDPs" [10] or "Multi-model MDPs" [24], which show that solving for such policies is NP-Hard and provide Mixed Integer Programming-based solutions. Instead, in this paper, we use the fact that per-arm MDPs for public health RMABs are typically small to create an efficient alternate approach that is also easily differentiable.

4 DECOMPOSED RMAB EVALUATION

Our high-level idea for speeding up DFL involves coming up with a good policy $\boldsymbol{\pi}^{\text{DEC}}$ that has the following properties:

- **Decomposable:** If we can come up with a good policy $\boldsymbol{\pi}^{\text{DEC}} = [\pi_1^{\text{DEC}}(s_1), \ldots, \pi_N^{\text{DEC}}(s_N)]$ that acts on different beneficiaries independently, we can also *evaluate* it in a decomposed manner:

$$J_T(\boldsymbol{\pi}^{\text{DEC}}) = \sum_i J_{T_i}(\pi_i^{\text{DEC}})$$

  Specifically, we can evaluate the per-arm returns by solving the Bellman Equations (Algorithm 3 in Appendix A) *without the need for simulations*, because the number of states in each per-arm MDP is typically small in RMAB formulations for public health.
- **Differentiable:** If the algorithm for estimating $\boldsymbol{\pi}^{\text{DEC}}$ is differentiable, we can simply substitute $\boldsymbol{\pi}^\star$ with $\boldsymbol{\pi}^{\text{DEC}}$ in Eq. (2) to get the following decomposed estimator for the predictive model:

$$\theta^\star = \arg\max_\theta \mathbb{E}_{\boldsymbol{x},T\sim\mathcal{D}} \left[J_T(\boldsymbol{\pi}^{\text{DEC}}(M_\theta(\boldsymbol{x})))\right] \tag{4}$$

Importantly, the Whittle Index policy $\boldsymbol{\pi}^{\text{WI}}$ in Eq. (3) that is used by Wang et al. [29] is *not* decomposable because we need to know the states $\boldsymbol{s}$ of all beneficiaries to determine Top-B(WI($\boldsymbol{s}$)) (Eq. (3)).

In the remainder of this section, we begin by showing why past approaches for calculating $\boldsymbol{\pi}^{\text{DEC}}$ lead to bad estimators of $\theta^\star$ and hence bad estimates of $\hat{T}$ in Section 4.1. Then, in Section 4.2, we propose an alternate problem formulation that leads to provably good estimation. Finally, we show how to efficiently solve for $\boldsymbol{\pi}^{\text{DEC}}$ in this alternative formulation by extending techniques from the DFL literature in Section 4.3.

4.1 Limitations of Past Work in Estimating $\theta^\star$

To create a policy that does not depend on the joint state $\boldsymbol{s} = [s_1, \ldots, s_N]$ of all the beneficiaries but rather on each beneficiary individually, past work [14, 30] relaxes the per-state budget constraint in Eq. (1) to a constraint over the amount of budget used *in expectation*. This results in the following relaxed problem:

$$\boldsymbol{\pi}^{\hat{T}\text{-DEC}}(\hat{T}) = \arg\max_{\boldsymbol{\pi}} J_{\hat{T}}(\boldsymbol{\pi}) \quad s.t.\ \bar{J}_{\hat{T}}(\boldsymbol{\pi}) \leq \frac{B}{1-\gamma} \tag{5}$$

where, $\bar{J}$ is the expected return of an MDP with transitions $\hat{T}$, but a different reward $\bar{R}(\boldsymbol{s},\boldsymbol{a}) = \sum_{i\in[N]} a_i$. $\bar{J}$ keeps track of how many interventions the policy $\boldsymbol{\pi}$ performs, and the constraint makes sure that this value is bounded by the (infinite-horizon discounted) budget $\frac{B}{1-\gamma}$. Then, Hawkins [14] shows an efficient way to solve the dual reformulation of this problem to get a decomposable policy.

However, while all the planning literature only focuses on calculating a good policy for a *single* fixed transition matrix $\boldsymbol{T}$, there are actually two sets of transition matrices in our DFL setting—the predicted transition matrices $\hat{T}$, and the true transition matrices T. As a result, if we use Eq. (5) to plan for the optimal policy $\boldsymbol{\pi}^\star(\hat{T}) \approx \boldsymbol{\pi}^{\hat{T}\text{-DEC}}(\hat{T})$ in the DFL pipeline, we would only satisfy the budget constraint with respect to the *predicted* transitions, not the true transitions. As a result, if $\hat{T} \neq T$ it could lead to (possibly large) constraint violations:

Example 4.1. Below, we describe what may go wrong in the simplest possible parameter estimation problem—predicting the parameters of an RMAB with only one arm, i.e., a single 2-state MDP's transition matrix $\hat{T}$. Consider the prediction:

$$\hat{T}^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ \hat{T}^1 = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix},\ \text{WI}_{\hat{T}} = \begin{bmatrix} \frac{\gamma}{1-\gamma} \\ 0 \end{bmatrix} \tag{6}$$

where, an entry of the matrix $\hat{T}^a_{s,s'}$ represents the probability $P(s'|s,a)$ of transitioning to state s' when in state s and taking action a, and $\text{WI}_{\hat{T}}$ contains the whittle indices of each state.

This MDP has the highest possible Whittle Index (action effect) for state 0—if you don't act, you'll always stay in state 0 and accumulate no reward, but if you act on the arm *just once*, you will transition to state 1 where you can passively collect a reward of 1 in every timestep without ever needing to act again. Because you only need to act once to get the benefits, the optimal policy uses only 1 unit of budget in comparison to the $\frac{B}{1-\gamma}$ units that are available (the $1-\gamma$ factor comes from the infinite-horizon discounting). As a result, as long as our budget $B \geq 1-\gamma$, the optimal policy $\pi^{\hat{T}\text{-DEC}}(\hat{T})$ according to Eq. (5) will be to act in state 0.

However, in the DFL context, this policy must be evaluated not on the predicted transition matrix $\hat{T}$, but on the *true* transition matrix that could be completely different. For example, consider the following true transition matrix T:

$$T^0 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \; T^1 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}, \; \text{WI}_T = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

For this transition matrix, we will always stay in state 0 (or move there, if we start in state 1). Applying the policy $\pi^{\hat{T}\text{-DEC}}(\hat{T})$ from above, that chooses to act in state 0, we will expend a discounted budget of $\approx \frac{1}{1-\gamma}$ because we will act in every timestep. As a result, if our true budget is only $1-\gamma$, we will overshoot our budget by a factor of $\frac{\text{used budget}}{\text{true budget}} = \frac{\frac{1}{1-\gamma}}{1-\gamma} = \frac{1}{(1-\gamma)^2}$, which is **100x** for a standard discount factor of $\gamma = 0.9$. △

The example above shows that there exists a combination of predictions $\hat{T}$ and true matrices T for which Eq. (5) leads to budget violations. However, the goal is not to solve for good policies, but rather to estimate parameters by using Eq. (5) in the DFL pipeline. So, do these budget violations lead to bad parameter estimation? In the theorem below, we show that using Eq. (5) to perform parameter estimation leads to spurious minima in the DFL setting.

Theorem 1. *Predicting $\hat{T} = T$ is **not** always a maximizer of the Predict-Then-Optimize problem below:*

$$\hat{T}^\star = \arg\max_{\hat{T}} \; J_T(\boldsymbol{\pi}^{\hat{T}\text{-DEC}}(\hat{T}))$$

Proof Sketch. The intuition for this claim is that, along the lines of Example 4.1, one can "buy" more budget by predicting a transition matrix $\tilde{\boldsymbol{T}}$ that uses less budget than the true transitions T. To prove this, we provide a proof by counterexample where:

$$J_T(\boldsymbol{\pi}^{\hat{T}\text{-DEC}}(\tilde{\boldsymbol{T}})) > J_T(\boldsymbol{\pi}^{\hat{T}\text{-DEC}}(T)) \qquad \square$$

Moreover, our choice of T in the counter-example is not special, making bad parameter estimation the norm, and not an exception.

4.2 Our Approach: DEC-DFL

In this section, we begin by proposing Eq. (7), an alternative to to Eq. (5), that leads to provably good parameter estimation (Theorem 2). Then, to solve Eq. (7), we propose a series of approximations that exploit the properties of Theorem 2 and the fact that per-arm MDPs in public health-based RMABs are small, to get Algorithm 1.

Then, we begin this section by first defining an alternative to Eq. (5) that ensures budget feasibility:

$$\boldsymbol{\pi}^{T\text{-DEC}}(\hat{T}) = \arg\max_{\boldsymbol{\pi}} \; J_{\hat{T}}(\boldsymbol{\pi}) \quad s.t. \;\; \bar{J}_T(\boldsymbol{\pi}) \leq \frac{B}{1-\gamma} \tag{7}$$

where the only difference is that the budget constraint must now be satisfied with respect to *true transition matrix* T. Then, we can show that $\boldsymbol{\pi}^{T\text{-DEC}}$ leads to good DFL parameter estimation:

Theorem 2. *Predicting $\hat{T} = T$ is always a maximizer of the Predict-Then-Optimize problem below:*

$$\hat{T}^\star = \arg\max_{\hat{T}} \; J_T(\boldsymbol{\pi}^{T\text{-DEC}}(\hat{T})) \tag{8}$$

Proof. We begin by noting that the input to J_T in Eq. (8) is the output of Eq. (7). As a result, any such input policy must satisfy the constraint that $\bar{J}_T(\boldsymbol{\pi}) \leq \frac{B}{1-\gamma}$. Then, the optimal solution to Eq. (8) across all possible policies is $\boldsymbol{\pi}^\star = \arg\max_{\bar{J}_T(\boldsymbol{\pi}) \leq \frac{B}{1-\gamma}} J_T(\boldsymbol{\pi})$, which is (by definition) exactly the solution to $\boldsymbol{\pi}^{T\text{-DEC}}(T)$! Therefore, any prediction $\hat{T}$ can only ever do as well as $\boldsymbol{\pi}^{T\text{-DEC}}(T)$:

$$J_T(\boldsymbol{\pi}^{T\text{-DEC}}(T)) \geq J_T(\boldsymbol{\pi}^{T\text{-DEC}}(\hat{T})), \quad \forall \hat{T} \qquad \square$$

Solving the problem in Eq. (7), however, is significantly more challenging than solving Eq. (5) because, unlike in Hawkins [14], the dual reformulation of Eq. (7) cannot be efficiently solved (see 'Multi-Model MDPs' in Section 3). Instead, in this paper, we use a different set of approximations that rely on two observations:

- **Theorem 2 holds regardless of the domain of $\boldsymbol{\pi}$:** Our argument only relies on the fact that $\boldsymbol{\pi}^{T\text{-DEC}}(T)$ maximizes $\arg\max_{\boldsymbol{\pi}} J_T(\boldsymbol{\pi})$. However, this is true regardless of whether $\boldsymbol{\pi}$ is a deterministic policy, a randomized policy, or even some mixture of these. As a result, we will have good parameter estimation regardless of the class of policies that we optimize over.
- **We do not have to solve Eq. (7) exactly**: Given that our only use of $\boldsymbol{\pi}^{\text{DEC}}$ is to estimate good parameters $\boldsymbol{\theta}^\star$, we do not have to restrict ourselves to using practically implementable policies. Instead, we can choose a different policy space that is easier to optimize over. This is similar to minimizing the MSE as an easy-to-optimize surrogate for the "0-1" loss.

So, while in practice we may want to optimize over the class of deterministic policies that contains, for e.g., the Whittle Index policy $\boldsymbol{\pi}^{\text{WI}}$, we can instead optimize over a richer class—a mixture of deterministic policies Z such that $\boldsymbol{\pi} \sim Z$. Then, we use two facts to simplify our optimization. First, we use the following theorem to show that optimizing over this space is equivalent to optimizing over the space of *decomposable* deterministic policies $\boldsymbol{\pi} \sim Z^{\text{DEC}}$.

Theorem 3. *Let Ω be the set of all distributions over deterministic policies, and Ω^{DEC} be the set of all distributions over deterministic* decomposable *policies. Consider the following optimization problems:*

$$\max_{Z \in \Omega} \quad \mathbb{E}_{\boldsymbol{\pi} \sim Z}[J_T(\boldsymbol{\pi})], \quad s.t. \;\; \mathbb{E}_{\boldsymbol{\pi} \sim Z}[\bar{J}_T(\boldsymbol{\pi})] \leq \frac{B}{1-\gamma}$$

$$\max_{Z \in \Omega^{\text{DEC}}} \quad \mathbb{E}_{\boldsymbol{\pi} \sim Z}[J_T(\boldsymbol{\pi})], \quad s.t. \;\; \mathbb{E}_{\boldsymbol{\pi} \sim Z}[\bar{J}_T(\boldsymbol{\pi})] \leq \frac{B}{1-\gamma}$$

Then, any maximizer of the latter is also a maximizer of the former.

Second, we use the fact that each per-arm MDP is typically small in public health-based RMAB formulations (just two states in our real-world domain). Combining these two, we can enumerate all $2^{|\mathcal{S}|}$ deterministic per-arm policies, and then solve for the optimal

mixture over them using the following optimization problem:

$$Z^\star(J_{\hat{T}}, \bar{J}_T) = \underset{0 \le Z_{ij} \le 1}{\arg\max} \quad \sum_{i=1}^{N} \sum_{j=1}^{2^{|\mathcal{S}|}} Z_{ij} J_{\hat{T}_i}(\pi^j) + \Phi(Z)$$
$$s.t. \quad \sum_{j=1}^{2^{|\mathcal{S}|}} Z_{ij} = 1, \quad \forall i$$
$$\sum_{i=1}^{N} \sum_{j=1}^{2^{|\mathcal{S}|}} Z_{ij} \bar{J}_{T_i}(\pi^j) \le \frac{B}{1-\gamma} \tag{9}$$

where each variable $Z^\star_{ij}$ in the solution is the probability of acting on arm i using policy π^j. $\Phi(Z)$ is a regularization term that is added to make the solution differentiable with respect to $\hat{T}$ (discussed in more detail below). Our overall algorithm for the decomposed evaluation of a set of predictions $\hat{T}$ is then described in Algorithm 1.

Differentiability. From the perspective of the optimization problem, $J_{\hat{T}_i}(\pi^j)$ and $\bar{J}_{T_i}(\pi^j)$ are constants. As a result, if we set $\Phi(Z) = 0$, solving Eq. (9) reduces to a linear program. However, it has been shown that the solutions of linear programs are not differentiable with respect to their inputs [11, 32] because similar predictions almost always lead to the same decisions. To make the solutions of Eq. (9) vary smoothly as $\hat{T}$ changes, we add a regularization term Φ (e.g., the L_2 norm $||Z||_2$ of the variables or the entropy $H(Z)$) to the objective of the optimization problem.

4.3 Efficiently Solving Equation (9)

The previous section provided a way to create good decomposable RMAB policies using an approximation to Eq. (7). However, the crux of the solution, Algorithm 1, involves incorporating the optimization problem in Eq. (9) into the DFL pipeline. One way to do this would be to use differentiable optimization packages like Cvxpylayers [1] (DEC-DFL), but this can be slow. Instead, in this section, we use the fact that all the arms are tied together only by the budget constraint to speed up Algorithm 1 and create our final *'Fast DEC-DFL'* method for RMAB parameter estimation using DFL.

Algorithm 1 Calculation of $\ell_{\text{DEC-DFL}}$ using $\boldsymbol{\pi}^{T\text{-DEC}}(\hat{T})$

Input: Predicted transition matrices $\hat{T}$
Parameter: True transition matrices T
Output: $\ell_{\text{DEC-DFL}}(\hat{T}, T)$

1: **for all** $i \in [N]$ and $\pi^j \in 2^{|\mathcal{S}|}$ **do** ▷*Ideally, in parallel*
2: Get return of "reward" MDP and predicted transitions $\hat{T}_i$:
$$J_{\hat{T}_i}(\pi^j) \leftarrow \textsc{GetReturns}(\hat{T}_i, R, \pi^j)$$
3: Get return of "reward" MDP and true transitions T_i:
$$J_{T_i}(\pi^j) \leftarrow \textsc{GetReturns}(T_i, R, \pi^j)$$
4: Get return of "budget" MDP and true transitions T_i:
$$\bar{J}_{T_i}(\pi^j) \leftarrow \textsc{GetReturns}(T_i, \bar{R}, \pi^j)$$
5: **end for**
6: Solve Eq. (9) using returns $J_{\hat{T}}$ and $\bar{J}_T$ calculated above:
$$Z^\star \leftarrow Z^\star(J_{\hat{T}}, \bar{J}_T)$$
7: **return** $\ell_{\text{DEC-DFL}} = \sum_i \sum_j Z^\star_{ij} \cdot J_{T_i}(\pi^j)$

Forward Pass. To solve Eq. (9), we first observe that the only thing tying together different arms is a single constraint, i.e., $\sum_{i,j} Z_{ij} \bar{J}_{T_i}(\pi^j) \le \frac{B}{1-\gamma}$. Moreover, Eq. (9) is a convex optimization problem that is strictly feasible as long as the budget $B > 0$. Then, because of strong duality via Slater's condition [8], we can instead solve the following primal-dual problem:

$$\min_{\lambda \ge 0} \ \underset{0 \le Z_{ij} \le 1}{\arg\max} \quad \sum_{i=1}^{N} \sum_{j=1}^{2^{|\mathcal{S}|}} Z_{ij} [J_{\hat{T}_i}(\pi^j) - \lambda \bar{J}_{T_i}(\pi^j)] + \alpha H(Z) + \lambda \frac{B}{1-\gamma}$$
$$s.t. \quad \sum_{j=1}^{2^{|\mathcal{S}|}} Z_{ij} = 1, \quad \forall i$$

where, $H(Z) = -\sum_i \sum_j Z_{ij} \log Z_{ij}$ is the entropy of the distribution Z_i over the different possible policies π^j and α is the weight of the regularization. Then, the solution to the inner maximization problem is given by the softmax function [4, 15]. Therefore we can simplify our reformulated optimization problem as:

$$\min_{\lambda \ge 0} \sum_{i=1}^{N} \sum_{j=1}^{2^{|\mathcal{S}|}} \tilde{Z}^\star_{ij}(\lambda) [J_{\hat{T}_i}(\pi^j) - \lambda \bar{J}_{T_i}(\pi^j)] + \lambda \frac{B}{1-\gamma}$$
$$\text{where, } \tilde{Z}^\star_{ij}(\lambda) = \text{softmax}_{\pi^j} \left(\frac{J_{\hat{T}_i}(\pi^j) - \lambda \bar{J}_{T_i}(\pi^j)}{\alpha} \right) \tag{10}$$

Now, to solve for the optimal value of the dual variable $\lambda^\star$, we rely on KKT conditions. In particular, it is well known that $\lambda^\star$ satisfies the complementary slackness [8] condition in Eq. (11). Then, to solve Eq. (10), we use a numerical root-finding algorithm to find the value of $\lambda^\star$ that leads to exactly satisfying the budget constraint. Algorithm 2 describes this procedure, and the following theorem proves that it does indeed return the optimal dual variable.

Theorem 4. *Algorithm 2 solves for the optimal dual variable $\lambda^\star$*

Proof. Based on KKT conditions, we know that any $\lambda^* \ge 0$ satisfying the following condition is an optimal solution to Eq. (9):

$$\lambda^\star \left(\sum_{i=1}^{N} \sum_{j=1}^{2^{|\mathcal{S}|}} Z^\star_{ij}(\lambda) \bar{J}_{T_i}(\pi^j) - \frac{B}{1-\gamma} \right) = 0 \tag{11}$$

First, observe that $\sum_{i=1}^{N} \sum_{j=1}^{2^{|\mathcal{S}|}} Z^\star_{ij}(\lambda) \bar{J}_{T_i}(\pi^j)$ decreases monotonically in λ. This follows from Eq. (10) and the properties of softmax (see Proposition D.1 for a proof). Intuitively, λ can be thought of as the "cost of acting". Then, as $\lambda \to \infty$ you will never act because the cost is too high, and if $\lambda \to -\infty$ you are incentivized to always act.

Now consider the following equation: $\sum_{i=1}^{N} \sum_{j=1}^{2^{|\mathcal{S}|}} Z^\star_{ij}(\lambda) \bar{J}_{T_i}(\pi^j) - B/(1-\gamma) = 0$. Because of the strict monotonicity of $Z^\star_{ij}(\lambda)$ the equation has a unique root. If this root is positive, then it satisfies the KKT condition in Equation (11) and is hence an optimizer. In this case, the budget constraint is tight. On the other hand, if the root is negative, then the budget constraint has a slack and the unique optimal solution is $\lambda^\star = 0$. □

Algorithm 2 exploits the monotonicity of $\sum_{i,j} Z^\star_{ij}(\lambda) \bar{J}_{T_i}(\pi^j)$ to efficiently find a root. It uses bisection method [9] and requires at

Algorithm 2 ForwardPass

Inputs: The Expected Returns $J_{\hat{T}}$ and $\bar{J}_T$
Parameter: Error tolerance ϵ, Budget B, Max reward R_{max}
Output: Distribution $Z^\star$ over arms $i \in [N]$ and policies π^j

procedure EvalLambda(λ)
  Compute $\tilde{Z}^\star(\lambda) \leftarrow \text{softmax}_{\pi^j}([J_{\hat{T}_i} - \lambda \bar{J}_{T_i}]), \forall i$
  return $\sum_{ij} \tilde{Z}^\star_{ij}(\lambda) \bar{J}_{T_i}(\pi^j) - \frac{B}{1-\gamma}$
end procedure

Set interval to $I = [-\frac{R_{\max}}{1-\gamma}, \frac{R_{\max}}{1-\gamma}]$ ▷ $\frac{R_{max}}{1-\gamma} = \max(J_{\hat{T}})$
Run the root-finding algorithm to get the optimal penalty $\lambda^\star$:

$$\lambda^\star \leftarrow \text{RootFinder}(\text{EvalLambda}, I, \epsilon)$$

Ignore constraint if $\lambda^\star < 0$, i.e., the constraint is not violated:

$$\lambda^\star \leftarrow \max(\lambda^\star, 0)$$

return $\lambda^\star, Z^\star \leftarrow \text{softmax}_{\pi^j}([J_{\hat{T}_i} - \lambda^\star \bar{J}_{T_i}]), \forall i$

most $\log \epsilon^{-1}$ calls to EvalLambda to find an ϵ-approximate root. Consequently, the forward pass takes $O(N \cdot 2^{|\mathcal{S}|} \cdot \log \epsilon^{-1})$ time because each call to EvalLambda takes $O(N \cdot 2^{|\mathcal{S}|})$ time.

Backward Pass. The goal of the backward pass is to find the derivatives of the minimizer $Z^\star$ with respect to its inputs, i.e., $\nabla_{J_{\hat{T}}} Z^\star$ and $\nabla_{\bar{J}_T} Z^\star$. To do this, we differentiate through the KKT conditions of Equation (9) and solve the resulting set of linear equations [3]. Specifically, for a convex program of the form:

$$\max_z \quad q^\top z + H(z)$$
$$s.t. \quad Az = b, Gz \le h$$

we get the following set of linear equations:

$$\begin{bmatrix} \text{diag}(\frac{-1}{z^\star}) & A^\top & G^\top \text{diag}(\lambda^\star) \\ A & 0 & 0 \\ G & 0 & -\text{diag}(Gz^\star - h) \end{bmatrix} \begin{bmatrix} d_z \\ d_\nu \\ d_\lambda \end{bmatrix} = \begin{bmatrix} \frac{\partial \ell}{\partial z^\star} \\ 0 \\ 0 \end{bmatrix} \tag{12}$$

where (1) $[d_z, d_\nu, d_\lambda]$ are intermediate variables that relate to the gradients of ℓ with respect to the parameters of the optimization problem, and (2) $\frac{\partial \ell}{\partial z^\star}$ is the derivative of the evaluation function with respect to the minimizer $z^\star$ and is the input to the backward pass. Then, given the solution to the set of linear equations above, we can extract the derivatives of interest as follows:

$$\nabla_q \ell = \nabla_{J_{\hat{T}}} \ell = d_z$$
$$\nabla_G \ell = \nabla_{\bar{J}_T} \ell = \lambda^\star (d_z - d_\lambda z^\star)$$

The key challenge in the backward pass is in efficiently solving the set of linear equations in Eq. (12). Given that there are $N \cdot 2^{|\mathcal{S}|} + N + 1$ variables, naively solving these equations would be order $O(N^3)$. However, given the sparsity of the matrix, we can use Gaussian elimination to derive a closed-form solution to Eq. (12).

To do this, we begin by considering the simpler case, where there is no budget constraint. The set of equations in Eq. (12) can then be completely decomposed into the following per-arm equations:

$$\begin{bmatrix} \text{diag}(\frac{-1}{Z_i^\star}) & \mathbf{1}_{2^{|\mathcal{S}|}} \\ \mathbf{1}^\top_{2^{|\mathcal{S}|}} & 0 \end{bmatrix} \begin{bmatrix} d_{z_i} \\ d_{\nu_i} \end{bmatrix} = \begin{bmatrix} \frac{\partial \ell}{\partial Z_i^\star} \\ 0 \end{bmatrix} = \begin{bmatrix} J_{T_i} \\ 0 \end{bmatrix}$$

and the reduced row-echelon form of the augmented matrix is:

$$\left[\begin{array}{cc|c} \text{diag}(\frac{-1}{Z_i^\star}) & \mathbf{0}_{2^{|\mathcal{S}|}} & J_{T_i} - J_{T_i}^\top Z_i^\star \\ \mathbf{0}^\top_{2^{|\mathcal{S}|}} & 1 & J_{T_i}^\top Z_i^\star \end{array}\right]$$

Next, we put the budget constraint back in and rewrite the system of equations as the following augmented matrix:

$$\left[\begin{array}{ccc|c} \text{diag}(\frac{-1}{Z^\star}) & \mathbf{0}_{N \cdot 2^{|\mathcal{S}|} \times N} & \lambda^\star \bar{J}_T & \begin{bmatrix} \vdots \\ J_{T_i} - J_{T_i}^\top Z_i^\star \\ \vdots \end{bmatrix} \\ \mathbf{0}_{N \times N \cdot 2^{|\mathcal{S}|}} & Id_{N \times N} & \mathbf{0}_N & \begin{bmatrix} \vdots \\ J_{T_i}^\top Z_i^\star \\ \vdots \end{bmatrix} \\ \bar{J}_T^\top & \mathbf{0}_N^\top & \xi & 0 \end{array}\right]$$

where $\xi = \frac{B}{1-\gamma} - \sum_{ij} \tilde{Z}^\star_{ij} \bar{J}_{T_i}(\pi^j)$ is the amount of "slack" budget left over. We can then perform Gaussian elimination on the budget constraint and back-substitute to get the values of $[d_z, d_\nu, d_\lambda]$; we do not show the exact calculations here because they're clunky, but this can easily be solved algorithmically. In addition, given that we're performing a constant number of operations on $O(N \cdot 2^{|\mathcal{S}|})$ variables, our backward pass has an $O(N)$ complexity.

5 EXPERIMENTS

In this section, we empirically test our proposed approach on two domains and compare it to baselines from the literature.

Real-World Dataset. This is the same dataset used by Wang et al. [29]. We use the data from a large-scale anonymized quality improvement study performed by ARMMAN for 7 weeks [21] with beneficiary consent. We choose the cohort that received randomized interventions and randomly split it into 60 training, 20 validation, and 20 test sub-cohorts. Each sub-cohort has $N = 76$ beneficiaries and a budget of $B = 3$. For the features $\boldsymbol{x}$, we use 44 categorical demographic features captured during program intake, e.g., age, education, and income level. For the transitions, we first create trajectories for each beneficiary from their historical listenership. We do this by discretizing engagement into 2 states—an engaging beneficiary listens to the weekly automated voice message (average length 60 seconds) for more than 30 seconds—and sequencing them to create an array $(s_0, a_0, s_1, \ldots)$. Then, to get the transition matrix for beneficiary i, we combine the observed transitions with P_{pop}, a prior created by pooling all the beneficiaries' trajectories together:

$$T_i(s, a, s') = P_i(s'|s, a) = \frac{\alpha P_{\text{pop}}(s'|s, a) + N(s, a, s')}{\sum_{x \in \mathcal{S}} \alpha P_{\text{pop}}(x|s, a) + N(s, a, x)}$$

where $N(s, a, s')$ is the number of times the sub-sequence s, a, s' occurs in the trajectory, and $\alpha = 5$ is the strength of the prior.

Synthetic Dataset. We also create a synthetic dataset for which it's easier to control for important hyperparameters, e.g., the number of states $|\mathcal{S}|$ in the per-beneficiary MDP. Here, we generate the transition matrices T uniformly at random. We also generate trajectories of 10 timesteps based on these transition matrices. Then, to create the features, we pass the transition matrices through a randomly initialized 8–layer feedforward network with a hidden dimension of 1000. We then generate 100 cohorts of $N = 100$ beneficiaries

Table 1: Decision Quality Results. We document the performance of linear models trained using various loss functions in the table below. The values in **bold** represent the highest entries in the column, and those in *italics* are those that are in the 95% confidence interval of the maximum value. We find that our proposed loss functions consistently outperform the baselines from the literature.

	Loss	Normalized Joint Test DQ (↑)			Normalized Decomposed Test DQ (↑)		
		Real-World	Synthetic (2-State)	Synthetic (5-State)	Real-World	Synthetic (2-State)	Synthetic (5-State)
2-Stage	NLL	0.04 ± 0.06	0.59 ± 0.13	0.16 ± 0.04	-0.27 ± 0.11	0.64 ± 0.18	0.15 ± 0.07
	MSE	0.10 ± 0.06	0.83 ± 0.03	**0.38 ± 0.03**	-0.19 ± 0.10	0.86 ± 0.04	*0.37 ± 0.03*
Wang et al. [29]	SIM-DFL (1 trajectory)	-0.05 ± 0.07	0.78 ± 0.03	0.15 ± 0.02	-0.33 ± 0.08	0.82 ± 0.06	0.16 ± 0.04
	SIM-DFL (10 trajectories)	0.34 ± 0.15	0.79 ± 0.03	0.16 ± 0.03	0.21 ± 0.20	0.84 ± 0.06	0.15 ± 0.02
	SIM-DFL (100 trajectories)	0.26 ± 0.10	0.80 ± 0.02	0.19 ± 0.03	0.15 ± 0.11	0.86 ± 0.03	0.19 ± 0.04
	SIM-DFL (1000 trajectories)	*Timeout*	0.80 ± 0.02	0.18 ± 0.03	*Timeout*	0.84 ± 0.04	0.20 ± 0.05
Ours	DEC-DFL (L2)	*0.58 ± 0.04*	**0.86 ± 0.02**	*0.34 ± 0.04*	*0.50 ± 0.04*	**0.91 ± 0.01**	**0.38 ± 0.03**
	DEC-DFL (Entropy)	**0.62 ± 0.05**	**0.86 ± 0.02**	*0.33 ± 0.04*	**0.52 ± 0.05**	**0.91 ± 0.01**	*0.35 ± 0.03*
	Fast DEC-DFL (Entropy)	*0.57 ± 0.12*	**0.86 ± 0.02**	*0.33 ± 0.04*	*0.45 ± 0.14*	**0.91 ± 0.01**	*0.35 ± 0.03*

with a budget of $B = 10$ per cohort. We split these cohorts into 20 train, 20 validation, and 60 test sub-cohorts.

Baselines. Broadly, we compare against two sets of baselines—(1) "standard" regression loss functions that focus on predictive accuracy, and (2) the DFL approach proposed by Wang et al. [29]. For the first, we use the Mean Squared Error between the predicted and true transition matrices (used by Mate et al. [21]), and the Negative Log Likelihood (NLL) that the predicted transition matrices generate the observed trajectories (used as a baseline by Wang et al. [29]). For the second, we use Wang et al. [29]'s SIM-DFL approach and vary the number of *simulated trajectories* used to evaluate the Whittle Index policy, to show the trade-off between cost and learned model quality. We compare these baselines to our proposed approach (Algorithm 1), in which we solve Eq. (9) using either the Cvxpylayers library [1] (DEC-DFL) or the strategy in Section 4.3 (Fast DEC-DFL). We use these different approaches to train a linear predictive model.

Evaluation Metrics. We evaluate the quality of our learned models M_θ using the predict-then-optimize framework (Eq. (2)):

$$\text{DQ}(M_\theta) = \mathbb{E}_{\boldsymbol{x}, T\sim\mathcal{D}}\left[J_T(\boldsymbol{\pi}^\star(M_\theta(\boldsymbol{x})))\right]$$

where DQ is the "decision quality" of the model. We approximate the value of the expectation using samples from the test set, resulting in the 'Test DQ'. In addition, we make the following modifications:

- **Policy Approximation:** As discussed in Section 3, calculating $\boldsymbol{\pi}^\star$ is PSPACE-Hard and so we either evaluate the models using $\boldsymbol{\pi}^\star \approx \boldsymbol{\pi}^{\text{WI}}$ as in past work [29] to get the "Joint DQ", or $\boldsymbol{\pi}^\star \approx \boldsymbol{\pi}^{T\text{-DEC}}$ to get the "Decomposed DQ". We use 1000 trajectories to evaluate the Joint Test DQ in the experiments below.
- **Normalization:** In order to ensure that we're focusing on the *intervention effect*, we linearly re-scale the decision quality such that 0 corresponds to the DQ of never acting and 1 corresponds to acting based on perfect predictions.

Putting these together we get our metrics of interest, i.e., the 'Normalized Joint Test DQ' and the 'Normalized Decomposed Test DQ'. The policy used in practice is $\boldsymbol{\pi}^{\text{WI}}$, and so the former metric is a good representation of how well the learned models would do if deployed. The latter is the surrogate we introduce in this paper; measuring this allows us to empirically verify that our proposed objective is well-correlated with the true objective of interest.

Hyperparameter Tuning. For our experiments, we vary the learning rate $\text{lr} = \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$ and for our approach we also vary the regularization constant $\alpha = \{1, 0.1\}$. All our results are averaged over 10 random train-val-test splits and 5 random model initializations per split. We then choose the hyperparameter value which leads to the lowest loss on the validation set. The final results are presented as "mean ± 1 standard error of the mean".

5.1 Overall Results

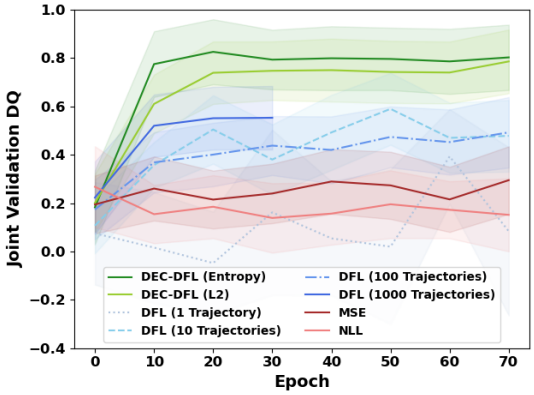
In this section, we analyze the results of our experiments, presented in Table 1 and Fig. 2. Overall, we find that our 'Fast DEC-DFL' approach described in Section 4.3 yields a speed-up of up to 500x over past work while also achieving comparable model performance.

We now look more closely at our decision quality results in Table 1:

- *DFL is important in the real-world domain:* The 2-stage methods do significantly worse than both SIM-DFL and DEC-DFL in the real-world domain. This is consistent with past work [25, 29].
- *DFL is less useful in the simulated domain:* In the 2-state domain, we find that DEC-DFL performs only slightly better than MSE, and in the 5-stage domain, this difference disappears almost completely. We believe that this is because the true data-generating process is a lot noisier than the one we use to create our synthetic domain, and that is where DFL is particularly useful.
- *Decomposed Test DQ mirrors Joint Test DQ:* Broadly, we find that the ordering of methods according to the Decomposed DQ mirrors the ordering according to the Joint DQ, which is used in practice. This suggests that our decomposed evaluation method is a good way to measure decision quality.
- *DEC-DFL consistently does better than SIM-DFL:* This is true regardless of the number of trajectories used in SIM-DFL. This, combined with the fact that we continue to see an improvement in DQ as the number of trajectories increases in Fig. 2b, suggests that even 1000 trajectories are not enough for accurate simulation-based evaluations.

Loss	Time Per Epoch In Seconds (↓)		
	Real-World	Synthetic (2-State)	Synthetic (5-State)
NLL	0.69 ± 0.08	0.22 ± 0.05	0.24 ± 0.07
MSE	0.49 ± 0.03	0.15 ± 0.01	0.18 ± 0.06
SIM-DFL (1 trajectory)	18.20 ± 2.78	6.51 ± 1.01	18.18 ± 0.96
SIM-DFL (10 trajectories)	21.34 ± 1.90	8.83 ± 1.77	19.97 ± 2.08
SIM-DFL (100 trajectories)	51.10 ± 1.57	29.33 ± 11.20	34.07 ± 2.00
SIM-DFL (1000 trajectories)	503.24 ± 32.16	305.69 ± 130.77	246.48 ± 57.47
DEC-DFL (L2)	4.63 ± 0.15	2.20 ± 0.40	46.28 ± 4.00
DEC-DFL (Entropy)	19.51 ± 2.30	11.61 ± 0.87	289.62 ± 49.61
Fast DEC-DFL (Entropy)	1.07 ± 0.10	0.39 ± 0.03	0.70 ± 0.16

(a) Time taken by different methods for a single training epoch.



(b) Validation DQ vs. Epoch on Real-World Dataset.

Figure 2: Computational Cost Results. In (a), we find that our proposed "Fast DEC-DFL" loss is roughly 500x faster than the "SIM-DFL (1000 Trajectories)" loss proposed by Wang et al. [29]. In (b), we show that this speed-up does not come at any cost in terms of the rate of convergence. In fact, "DEC-DFL" outperforms even "SIM-DFL (1000 Trajectories)" till the latter times out.

We now analyze the computational time results in Fig. 2.

- *Fast DEC-DFL is 500x faster than SIM-DFL (1000 trajectories):* In addition, Fast DEC-DFL even has better performance than SIM-DFL, as seen in Fig. 2b.
- *DEC-DFL does not scale well in $|\mathcal{S}|$:* We see that in going from 2 to 5 states, the computational cost of DEC-DFL increases by ≈20x, which is even higher than the $\frac{2^5}{2^2} = 8$ increase in number of policies required to solve Eq. (9). This is because naively solving Eq. (12) requires inverting a matrix of dimension $O(N \cdot 2^{|\mathcal{S}|})$.
- *The convergence rate is similar for all methods:* We see in Fig. 2b that all the methods seem to converge after a similar number of epochs. This suggests that the per-epoch difference in computational cost from Fig. 2a extends to the overall computational cost of training predictive models using the different methods.

5.2 Sensitivity To Model Capacity

In Section 5.1 our results are presented for linear predictive models M_θ. Here, we show that our findings hold *even if we use more complex predictive models*. In Table 2 we find that:

- *Increasing model capacity helps 2-stage and SIM-DFL:* We find that increasing model capacity from 'small' to 'medium' or 'large' seems to boost performance when using the 'MSE' or 'SIM-DFL (1 Trajectory)' losses. However, even a *linear* model trained using DEC-DFL outperforms all other baselines.
- *Model capacity does not affect DEC-DFL:* Increasing model capacity does not seem to help when using the DEC-DFL approach. In Appendix E, we visualize the predictions of different approaches and show that DEC-DFL finds beneficiaries that would benefit from interventions even with limited model capacity.

Table 2: Sensitivity to Model Capacity. We report the performance of models of varying sizes in the table below.

Loss	Normalized Joint Test DQ (↑)		
	Small (Linear)	Medium (2-Layer, 64 Dim)	Large (4-Layer, 500 Dim)
NLL	0.04 ± 0.06	0.01 ± 0.07	0.04 ± 0.06
MSE	0.10 ± 0.06	0.35 ± 0.12	0.34 ± 0.11
SIM-DFL (1 trajectory)	-0.05 ± 0.07	0.36 ± 0.07	0.30 ± 0.18
SIM-DFL (10 trajectories)	0.34 ± 0.15	0.47 ± 0.20	0.36 ± 0.27
SIM-DFL (100 trajectories)	0.26 ± 0.10	0.44 ± 0.21	0.33 ± 0.28
DEC-DFL (L2 Reg)	0.58 ± 0.04	0.59 ± 0.04	0.61 ± 0.03
DEC-DFL (Entropy)	**0.62 ± 0.05**	**0.60 ± 0.06**	**0.62 ± 0.04**
Fast DEC-DFL (Entropy)	0.57 ± 0.12	**0.60 ± 0.04**	0.61 ± 0.04

6 CONCLUSION AND FUTURE WORK

Overall, we propose a novel approach, 'Fast DEC-DFL', for solving RMABs in the DFL setting. Our approach efficiently calculates decomposable policies that are cheap to evaluate. This results in a **500x** speedup over state-of-the-art methods on real-world data from ARMMAN, while also improving model performance. Concretely, where past work ('SIM-DFL (1000) Trajectories') can take more than a day to train for our dataset with ≈ 5000 beneficiaries, Fast DEC-DFL takes minutes. This gain in speed with the added benefit of improved accuracy paves the way for DFL-based RMAB models to be deployed more widely and at larger scale. For example, this could potentially help ARMMAN with their ongoing efforts to boost engagement in their Kilkari program—the largest maternal mHealth program in the world [5], with 3 million active subscribers.

ACKNOWLEDGMENTS

This material is based upon work supported by the NSF under Grant No. IIS-2229881. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF.

7 ETHICS STATEMENT

Secondary Analysis and Data Usage. The experiments with the ARMMAN dataset fall into the category of secondary analysis of the aforementioned dataset. We use previously collected listenership trajectories of beneficiaries enrolled in the mMitra program. The dataset is anonymized and contains no personally identifiable information. The dataset is owned by ARMMAN and only they can share it further.

Consent for Data Collection and Sharing. Consent for collecting data is obtained from each participant in the service call program. The data collection process is carefully explained to the participants before collecting the data. Data exchange and use were regulated through clearly defined exchange protocols including anonymization by ARMMAN, read-only access to researchers, restricted use of the data for research purposes only, and approval by ARMMAN's ethics review committee.

Universal Accessibility of Health Information. This study focuses on improving the effectiveness of only the *live service calls*. All participants will receive the same weekly health information by automated message regardless of whether they are scheduled to receive service calls or not. The service call program does not withhold any information from the participants nor conduct any experimentation on the health information. Moreover, all participants can request service calls via a free missed call.

Road To Deployment. The next steps involve testing our algorithm on more recent data to make sure our algorithm continues to show gains, and to run an equity audit to make sure that our algorithm prioritizes vulnerable subgroups. We then plan to conduct a randomized field trial to evaluate the accuracy of the algorithm and verify the computational gains over the currently deployed DFL pipeline. We hope for such a model to potentially showcase its strengths in applying DFL in a cost-effective way at such a massive scale. We must highlight, that all the above steps will be conducted with constant collaboration with ARMMAN; with ARMMAN ultimately being in charge of the actual deployment.

REFERENCES

[1] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and Z. Kolter. 2019. Differentiable Convex Optimization Layers. *Advances in Neural Information Processing Systems* (2019).

[2] Eitan Altman. 1999. *Constrained Markov decision processes*. Vol. 7. CRC press.

[3] Brandon Amos and J Zico Kolter. 2017. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*. PMLR, 136–145.

[4] Brandon Amos, Vladlen Koltun, and J Zico Kolter. 2019. The limited multi-label projection layer. *arXiv preprint arXiv:1906.08707* (2019).

[5] ARMMAN. 2023. *Kilkari*. https://www.armman.org/kilkari/

[6] ARMMAN. 2023. *mMitra*. https://www.armman.org/mmitra/

[7] Turgay Ayer, Can Zhang, Anthony Bonifonte, Anne C Spaulding, and Jagpreet Chhatwal. 2019. Prioritizing hepatitis C treatment in US prisons. *Operations Research* 67, 3 (2019), 853–873.

[8] Stephen P Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

[9] Richard P Brent. 2013. *Algorithms for minimization without derivatives*. Courier Corporation.

[10] Peter Buchholz and Dimitri Scheftelowitsch. 2019. Computation of weighted sums of rewards for concurrent MDPs. *Mathematical Methods of Operations Research* 89 (2019), 1–42.

[11] Adam N Elmachtoub and Paul Grigas. 2022. Smart "predict, then optimize". *Management Science* 68, 1 (2022), 9–26.

[12] Amir-massoud Farahmand, Andre Barreto, and Daniel Nikovski. 2017. Value-aware loss function for model-based reinforcement learning. In *Artificial Intelligence and Statistics*. PMLR, 1486–1494.

[13] Joseph Futoma, Michael C Hughes, and Finale Doshi-Velez. 2020. Popcorn: Partially observed prediction constrained reinforcement learning. *arXiv preprint arXiv:2001.04032* (2020).

[14] Jeffrey Thomas Hawkins. 2003. *A Langrangian decomposition approach to weakly coupled dynamic optimization problems and its applications*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[15] Ya-Ping Hsieh, Chen Liu, and Volkan Cevher. 2019. Finding mixed nash equilibria of generative adversarial networks. In *International Conference on Machine Learning*. PMLR, 2810–2819.

[16] Young Hun Jung and Ambuj Tewari. 2019. Regret bounds for thompson sampling in episodic restless bandit problems. *Advances in Neural Information Processing Systems* 32 (2019).

[17] Jackson A Killian, Manish Jain, Yugang Jia, Jonathan Amar, Erich Huang, and Milind Tambe. 2023. Equitable Restless Multi-Armed Bandits: A General Framework Inspired By Digital Health. *arXiv preprint arXiv:2308.09726* (2023).

[18] Jackson A Killian, Bryan Wilder, Amit Sharma, Vinod Choudhary, Bistra Dilkina, and Milind Tambe. 2019. Learning to prescribe interventions for tuberculosis patients using digital adherence data. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2430–2438.

[19] Jayanta Mandi, Vıctor Bucarey, Maxime Mulamba Ke Tchomba, and Tias Guns. 2022. Decision-focused learning: through the lens of learning to rank. In *International Conference on Machine Learning*. PMLR, 14935–14947.

[20] Aditya Mate, Jackson Killian, Haifeng Xu, Andrew Perrault, and Milind Tambe. 2020. Collapsing bandits and their application to public health intervention. *Advances in Neural Information Processing Systems* 33 (2020), 15639–15650.

[21] Aditya Mate, Lovish Madaan, Aparna Taneja, Neha Madhiwalla, Shresth Verma, Gargi Singh, Aparna Hegde, Pradeep Varakantham, and Milind Tambe. 2022. Field study in deploying restless multi-armed bandits: Assisting non-profits in improving maternal and child health. *Proceedings of the AAAI Conference on Artificial Intelligence* 36 (2022), 12017–12025.

[22] Evgenii Nikishin, Romina Abachi, Rishabh Agarwal, and Pierre-Luc Bacon. 2022. Control-oriented model-based reinforcement learning with implicit differentiation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36. 7886–7894.

[23] Christos H Papadimitriou and John N Tsitsiklis. 1994. The complexity of optimal queueing network control. *Proceedings of IEEE 9th annual conference on structure in complexity Theory* (1994), 318–322.

[24] Lauren N Steimle, David L Kaufman, and Brian T Denton. 2021. Multi-model Markov decision processes. *IISE Transactions* 53, 10 (2021), 1124–1139.

[25] Shresth Verma, Aditya Mate, Kai Wang, Neha Madhiwalla, Aparna Hegde, Aparna Taneja, and Milind Tambe. 2023. Restless Multi-Armed Bandits for Maternal and Child Health: Results from Decision-Focused Learning. *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems* (2023), 1312–1320.

[26] Shresth Verma, Gargi Singh, Aditya Mate, Paritosh Verma, Sruthi Gorantla, Neha Madhiwalla, Aparna Hegde, Divy Thakkar, Manish Jain, Milind Tambe, et al. 2023. Increasing impact of mobile health programs: SAHELI for maternal and child care. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 15594–15602.

[27] John Von Neumann and Oskar Morgenstern. 1947. Theory of games and economic behavior, 2nd rev. (1947).

[28] Kai Wang, Sanket Shah, Haipeng Chen, Andrew Perrault, Finale Doshi-Velez, and Milind Tambe. 2021. Learning mdps from features: Predict-then-optimize for sequential decision making by reinforcement learning. *Advances in Neural Information Processing Systems* 34 (2021), 8795–8806.

[29] Kai Wang, Shresth Verma, Aditya Mate, Sanket Shah, Aparna Taneja, Neha Madhiwalla, Aparna Hegde, and Milind Tambe. 2023. Scalable decision-focused learning in restless multi-armed bandits with application to maternal and child health. *Proceedings of the AAAI Conference on Artificial Intelligence* 37 (2023), 12138–12146.

[30] Richard R Weber and Gideon Weiss. 1990. On an index policy for restless bandits. *Journal of applied probability* 27, 3 (1990), 637–648.

[31] Peter Whittle. 1988. Restless bandits: Activity allocation in a changing world. *Journal of applied probability* 25, A (1988), 287–298.

[32] Bryan Wilder, Bistra Dilkina, and Milind Tambe. 2019. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1658–1665.

[33] EB Yanovskaya. 1974. Infinite zero-sum two-person games. *Journal of Soviet Mathematics* 2, 5 (1974), 520–541.

A EFFICIENTLY CALCULATING THE RETURNS OF DECOMPOSED POLICY

Algorithm 3 GetReturns

Input: Transition matrices T, Rewards R, Policy π
Output: Expected return $J_T(\pi)$

1: Get the markov transitions induced by the policy π:

$$T_\pi(s, s') \leftarrow T(s, \pi(s), s')$$

2: Get the corresponding value function:

$$V \leftarrow (I - \gamma T_\pi)^{-1} R$$

3: Multiply V with the initial state distribution:

$$J_T(\pi) \leftarrow \mathbb{E}_{s_0}[V(s_0)]$$

4: **return** $J_T(\pi)$

B PROOF OF THEOREM 1

For the sake of clarity, we restate the Theorem 1 below.

Theorem. *Predicting $\hat{T} = T$ is **not** always a maximizer of the Predict-Then-Optimize problem below:*

$$\hat{T}^\star = \arg\max_{\hat{T}} J_T(\boldsymbol{\pi}^{\hat{T}\text{-DEC}}(\hat{T}))$$

Proof. Consider a 2-state RMAB with 2 arms, $\gamma = 0.9$, and a budget $B = \frac{1}{1+\gamma}$ (i.e., expected budget $\frac{B}{1-\gamma} = \frac{1}{1-\gamma^2}$). One arm has a transition matrix described by T^{good} and the other by T^{bad}:

$$T^{\text{good}} = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{array}\right], \; T^{\text{bad}} = \left[\begin{array}{cc|cc} 1 & 0 & 0.5 & 0.5 \\ 1 & 0 & 1 & 0 \end{array}\right]$$

Now, acting in state 1 for either T^{good} or T^{bad} (lower row) doesn't make sense because there's no difference in the transition probabilities whether you act or not. Acting in state 0 of T^{good} uses an expected budget of $\bar{J} = \frac{1}{1-\gamma^2}$ and increases the expected return $\Delta J_{T^{\text{good}}}$ by $\frac{\gamma}{1-\gamma^2}$. Acting in state 0 of T^{bad} uses an expected budget of $\bar{J} = \frac{2}{2-\gamma-\gamma^2}$ and increases the expected return $\Delta J_{T^{\text{good}}}$ by $\frac{\gamma}{2-\gamma-\gamma^2}$. So, if we solve for $\boldsymbol{\pi}^{\hat{T}\text{-DEC}}([T^{\text{good}}, T^{\text{bad}}])$, the policy we get will be to only act in state 0 of T^{good}, because (a) it has a higher ratio of $\frac{\Delta J}{\bar{J}}$ than acting in state 0 of T^{bad}, and (b) uses up all the budget.

However, if we'd instead predicted the "best-case" transition matrix T^{OPT} as defined in Eq. (6), we could do better. As discussed in Example 4.1, acting in state 0 of T^{OPT} only uses an expected budget of $\bar{J} = 1$. Therefore, solving for $\boldsymbol{\pi}^{\hat{T}\text{-DEC}}([T^{\text{OPT}}, T^{\text{OPT}}])$ results in a policy for acting in state 0 for both T^{good} and T^{bad} (as long as $2 < \frac{1}{1-\gamma^2}$, which is satisfied for $\gamma = 0.9$). This is strictly better than $\boldsymbol{\pi}^{\hat{T}\text{-DEC}}([T^{\text{good}}, T^{\text{bad}}])$ which only acts in state 0 of T^{good}. Therefore:

$$J_T(\boldsymbol{\pi}^{\hat{T}\text{-DEC}}([T^{\text{OPT}}, T^{\text{OPT}}]) > J_T(\boldsymbol{\pi}^{\hat{T}\text{-DEC}}([T^{\text{good}}, T^{\text{bad}}])$$ □

Note that there isn't anything special about our choice of T; we just chose values that simplify the exposition. We could, however, repeat this sort of argument for almost any choice of T where acting is better than not acting!

C PROOF OF THEOREM 3

For the sake of clarity, we restate the Theorem 3 below.

Theorem. *Let Ω be the set of all distributions over deterministic policies, and let Ω^{DEC} be the set of all distributions over deterministic, decomposable policies. Consider the following optimization problems*

$$\max_{Z \in \Omega} \quad \mathbb{E}_{\boldsymbol{\pi} \sim Z}[J_T(\boldsymbol{\pi})], \quad \text{s.t.} \quad \mathbb{E}_{\boldsymbol{\pi} \sim Z}[\bar{J}_T(\boldsymbol{\pi})] \leq \frac{B}{1-\gamma} \tag{13}$$

$$\max_{Z \in \Omega^{\text{DEC}}} \quad \mathbb{E}_{\boldsymbol{\pi} \sim Z}[J_T(\boldsymbol{\pi})], \quad \text{s.t.} \quad \mathbb{E}_{\boldsymbol{\pi} \sim Z}[\bar{J}_T(\boldsymbol{\pi})] \leq \frac{B}{1-\gamma} \tag{14}$$

Then, any maximizer of optimization problem (14) *is also a maximizer of optimization problem* (13).

Proof. The Lagrangian of the first optimization problem is given by

$$\max_{Z\in\Omega}\min_{\lambda\geq 0}\ \mathbb{E}_{\boldsymbol{\pi}\sim Z}[J_T(\boldsymbol{\pi})]+\lambda\left(\frac{B}{1-\gamma}-\mathbb{E}_{\boldsymbol{\pi}\sim Z}[\tilde{J}_T(\boldsymbol{\pi})]\right)$$

Note that the above objective is linear in λ and Z. Using popular minimax theorems we can swap the ordering of min and max and obtain the following equivalent problem [27, 33]

$$\min_{\lambda\geq 0}\max_{Z\in\Omega}\ \mathbb{E}_{\boldsymbol{\pi}\sim Z}[J_T(\boldsymbol{\pi})]+\lambda\left(\frac{B}{1-\gamma}-\mathbb{E}_{\boldsymbol{\pi}\sim Z}[\tilde{J}_T(\boldsymbol{\pi})]\right).$$

Observe that for any fixed λ, the inner optimization decomposes across the N arms. Using this observation, it is easy to see that there exists an optimal Z that decomposes across the arms. So, the above problem can be equivalently written as

$$\min_{\lambda\geq 0}\max_{Z\in\Omega^{\text{DEC}}}\ \mathbb{E}_{\boldsymbol{\pi}\sim Z}[J_T(\boldsymbol{\pi})]+\lambda\left(\frac{B}{1-\gamma}-\mathbb{E}_{\boldsymbol{\pi}\sim Z}[\tilde{J}_T(\boldsymbol{\pi})]\right).$$

By appealing to minimax theorems we again swap the ordering of min and max and obtain the following equivalent problem

$$\max_{Z\in\Omega^{\text{DEC}}}\min_{\lambda\geq 0}\ \mathbb{E}_{\boldsymbol{\pi}\sim Z}[J_T(\boldsymbol{\pi})]+\lambda\left(\frac{B}{1-\gamma}-\mathbb{E}_{\boldsymbol{\pi}\sim Z}[\tilde{J}_T(\boldsymbol{\pi})]\right).$$

Note that this is equivalent to the second problem in Equation (14). This shows that any optimizer of Equation (14) is also an optimizer of Equation (13). □

D ADDITIONAL RESULTS

Proposition D.1. *Let $f(\lambda)=\frac{\sum_{i=}^{N} v_i e^{-\lambda v_i}}{\sum_{i=1}^{N} e^{-\lambda v_i}}$. Then f is a monotonically decreasing function of λ.*

Proof. The first derivative of f is given by

$$f'(\lambda)=-\frac{\sum_{i=1}^{N} v_i^2 e^{-\lambda v_i}}{\sum_{i=1}^{N} e^{-\lambda v_i}}+\frac{(\sum_{i=1}^{N} v_i e^{-\lambda v_i})^2}{(\sum_{i=1}^{N} e^{-\lambda v_i})^2}.$$

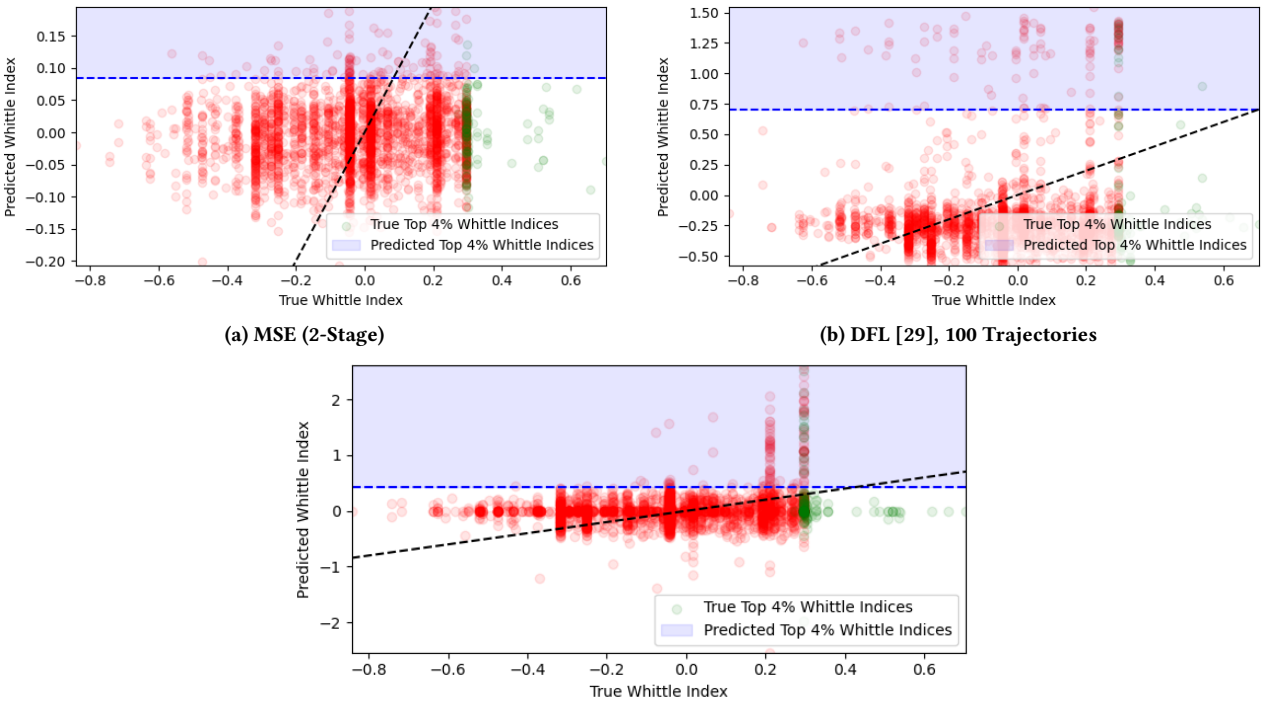
From the definition of $f(\lambda)$, the derivative can be rewritten as

$$f'(\lambda)=-\frac{\sum_{i=1}^{N}(v_i-f(\lambda))^2 e^{-\lambda v_i}}{\sum_{i=1}^{N} e^{-\lambda v_i}}$$

This shows that $f'(\lambda)\leq 0$. Consequently, f' is a decreasing function of λ. If at least one v_i is different from others, then $f'(\lambda)<0$ and f is a strictly decreasing function of λ. □

E VISUALIZING THE LEARNED MODELS

We visualize the predictions of the learned models in Fig. 3. We plot the predicted Whittle Index versus the true Whittle Index. In Fig. 3a, there isn't much difference between the Whittle Index distribution in the blue shaded region versus the population, highlighting that the model is not able to isolate beneficiaries for whom the action effect would be high. Conversely, in Fig. 3c, we see that the Whittle indices in the blue region have high true values, implying good model performance. We find that the model in Fig. 3b has performance somewhere in between (a) and (c). This shows that our approach is able to effectively find subsets of the population.



(a) MSE (2-Stage)

(b) DFL [29], 100 Trajectories

(c) DEC-DFL (Ours)

Figure 3: Visualization of Predictions on Real-World Domain. We plot the *predicted* (y-axis) versus *true* (x-axis) Whittle indices induced by different loss functions. A good loss function is one for which the top-B predicted Whittle indices (blue shaded region) are actually high, i.e., the *true* action effect is high when we predict a high action effect.