Improving Causal Transplant Outcomes through Dynamic Organ Offer Estimation

Alessandro Marchese
Vrije Universiteit Brussel
Brussels, Belgium
Alessandro.Marchese@vub.be

Hans de Ferrante
Eurotransplant International Foundation
Leiden, the Netherlands
h.deferrante@eurotransplant.org

Jeroen Berrevoets
University of Cambridge
Cambridge, England
jeroen.berrevoets@maths.cam.ac.uk

Sam Verboven
Vrije Universiteit Brussel
Brussels, Belgium
Sam.Verboven@vub.be

Abstract

Matching donor organs to patients in need is a difficult but important problem. A crucial factor in transplant outcomes is the cold ischemic time of the organ, which increases every time an organ offer is rejected. Despite this, acceptance dynamics have so far been neglected in favour of purely outcome driven offers. As a first alternative, we propose DynamITE, a novel causal organ allocation methodology that explicitly takes into account the acceptance behaviour over sequences of offers. DynamITE dynamically updates organ acceptance estimates, cold ischemic times (CIT) and causal effects throughout the matching process. We demonstrate that DynamITE improves early organ acceptance and maximizes patient life expectancy compared to current policies.

1 Introduction

Organ transplantation is the only curative therapy for patients with end-stage organ disease. However, due to a persistent donor shortage in the United States 17 patients die every day while waiting for a transplant [12]. Simultaneously, almost 20 percent of the organs recovered for transplantation in the United States remain unused [13]. There is an urgent need for organ allocation systems that efficiently place the limited supply of available organs to meet the increasing demand for transplantation and ultimately improve clinical outcomes.

Turndowns of organ offers degrade organ quality and lead to organ nonuse. Transplant centers frequently decline offered organs on behalf of their patients, for instance because of logistical or donor quality reasons [20, 10]. In fact, in the United States organ offer acceptance rates are only 1% for kidneys, 3% for livers, and 5% for hearts [22]. Moreover, most transplanted organs are declined by multiple patients before transplantation. Such repeated turndowns of an organ and delays in offer acceptance decisions mean that the organ has to be kept longer in a hypothermic state [16, 28]. These extended Cold Ischemia Times (CIT) are detrimental to the quality of the organ, and negatively affect transplant outcomes [8]. Additionally, delays in acceptance decisions and repeated turndowns lead to organ nonuse [19, 27].

Existing allocation policies ignore offer turndowns. Current organ allocation systems prioritize candidates based on scores for medical urgency [29] or anticipated transplant benefit [1, 9]. Lists of waiting candidates ranked in this way are referred to as *match-runs*. These match-runs determine the

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

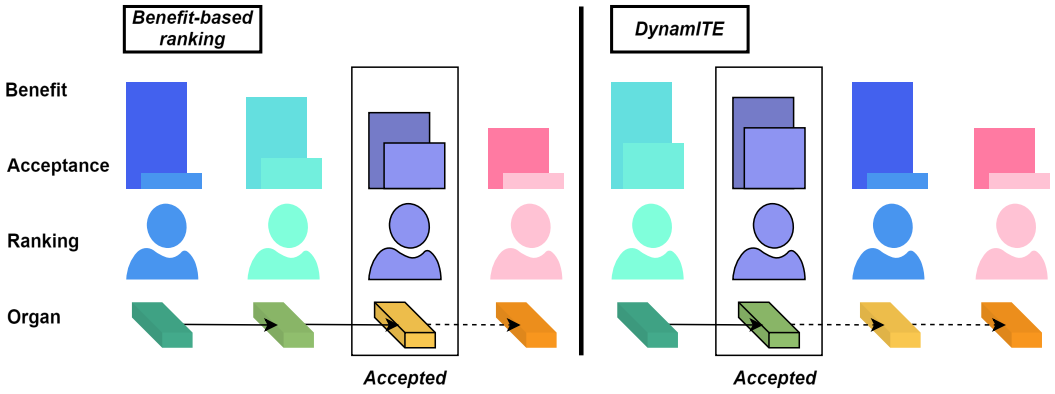

Figure 1: **Problem setting.** The benefit of an organ diminishes over time. Consequently, the more an organ is rejected and passed down, the lower the ultimate benefit for the patient who eventually accepts it. DynamITE takes into account decision time and the probability of acceptance, enhancing the overall benefit by promoting earlier acceptance of organs. In contrast, a policy that disregards acceptance probability (left) will yield a reduced benefit compared to DynamITE (right).

order in which candidates are offered the organ until a candidate accepts the organ for transplantation. Currently, a candidate's likelihood of accepting the graft offer does not affect their position on the match-run. As shown in figure 1, this may lead to repeated turndowns, extend CITs, and consequently lead to worse transplant outcomes.

Challenges in optimizing match-runs. Incorporating offer organ acceptance behavior and its impact on CIT into the design of organ matching policies is inherently complex for several reasons. First, precise predictions are needed regarding how long patients will live after receiving a particular organ. This coincides with the ITE (individualized treatment effects) estimation challenge in organ transplantation [4, 3]. Second, organ acceptance behavior is highly stochastic, and depends on extensive sets of patient and donor characteristics [30]. Finally, the sequential nature of the organ allocation process introduces a sequential bias to ITE estimation [11]. This is because acceptance behavior upstream in the allocation process directly affects which organ offers downstream candidates receive, and negatively impacts quality of the organ through CIT.

Correcting for this bias is particularly difficult, as each decision influences not only the quality of the organ (due to cold ischemic time) but also the potential benefits for downstream patients. These dynamics make optimizing match-runs a fundamentally different and more challenging problem than what existing policies are equipped to address.

Our contribution: DynamITE. In this paper, we present DynamITE, a novel organ allocation methodology that optimizes organ-to-patient matching by incorporating ITEs while accounting for the sequential and time-sensitive nature of organ offers. Unlike existing models that assume matches are always and instantaneously accepted, DynamITE uses predicted offer acceptance probabilities and decision times to dynamically update organ offers, taking into account adjusted CIT with each successive offer. This innovation ensures more timely acceptance decisions, reducing CIT and preventing organ nonuse. We demonstrate DynamITE's effectiveness through a series of experiments on semi-synthetic data, showing substantial improvements in patient life expectancy and reduced organ wastage compared to other policies.

2 Related work.

Organ allocation policies. Currently used match-runs in organ transplantation prioritize candidates based on medical urgency or anticipated benefit. For instance, MELD [18] or MELD-Na [15] are scoring functions for medical urgency, which have been used to prioritize candidates for liver transplantation. MELD-based match-runs fail to capture complex interactions between organs and other patients that influence patient outcomes. Policies have been proposed based on a causal

approach, relying on potential outcomes and ITE to rank patients [21, 31]. Additionally, some policies take operational aspects of transplantation into consideration, such as arrival rates of other organs and patients [4, 3] or the transport distance [2, 6]. Thus far, no policies have been proposed that explicitly take the operational aspect of acceptance uncertainty into consideration, leading DynamITE to be the first policy to explicitly model acceptance uncertainty and its effect on CIT.

Offer acceptance modeling. Discrete event simulators are routinely used for policy evaluation in transplantation [24, 25, 26]. The goal of such simulators is to mimic the complete organ allocation process, which includes waitlist survival, generation of match-runs, and simulation of post-transplant survival [23]. A key component of such simulators are modules that predict organ offer acceptance decisions based on patient and donor characteristics [30]. The widely used models developed by the Scientific Registry of Transplant Recipients (SRTR) rely on logistic models for predicting offer acceptance predictions [25, 24, 26]. Others have explored basing these prediction models on traditional ML methods [14, 5], or more complex conditional acceptance rules based on organ-patient compatibility [32]. Importantly, organs are treated as static objects in such simulations, ignoring the deterioriation of donor quality through increased CITs due to repeated turndowns. In DynamITE, spillover effects of repeated offer turndowns on organ quality are accounted for, as well as explicitly optimized over.

3 Problem formulation.

The organ allocation process consists of an organ being repeatedly offered to candidates on a waitlist until a candidate accepts the organ. We model three aspects of this allocation process: the order (match-run) in which an organ is offered to waitlist candidates, how candidates respond to an organ offer, and how long waitlist candidates and transpant recipients survive. A summary of all introduced notation can be found in table 3 in appendix A.

Responding to offers. Let $\mathcal{X} \subset \mathbb{R}^d$ be the space of all possible patients and $\mathcal{O} \subset \mathbb{R}^e$ the space of all possible organs. When an organ $\mathbf{O} \in \mathcal{O} \cup \{\emptyset\}$ is offered to a patient $\mathbf{X} \in \mathcal{X}$, we observe a response $A \in \{0, 1\}$, where $A = 1$ if the offer is accepted, and $A = 0$ if it is refused. The time between offer and acceptance is denoted $T \in \mathbb{R}_+$.

Sequential offers. Consider $\mathcal{X}_Q \subset \mathcal{P}(\mathcal{X})$ as a waiting list of size $K = |\mathcal{X}_Q|$. As an organ is offered sequentially to different patients and deteriorates over time, consider $\mathbf{O}_k$ as the organ offered to $\mathbf{X}_k \in \mathcal{X}_Q$ where $k \in \mathbb{Z}^+$ represents a sequence number and $1 \leq k \leq K$. Similarly, A_k and T_k represent the answer and decision time of patient $\mathbf{X}_k$ when faced with the offer of organ $\mathbf{O}_k$. When a donor organ becomes available this initial organ $\mathbf{O}_1 \in \mathcal{O}$ must be matched with a patient in the waiting list. Consider $\mathbf{R} \in \mathrm{Sym}(\mathcal{X}_Q)$ as a ranking of wait-listed patients, representing a match-run. When a donor organ becomes available, a policy π attributes a rank k to each patient in $\mathcal{X}_Q$ such that $\mathbf{R} = (\mathbf{X}_k)_{k=1}^K$.

A patient $\mathbf{X}_P$ who accepts the offer $A_P = 1$ will receive the transplant. Organs are offered to patients following the ranking $\mathbf{R}$ in ascending order until a patient $\mathbf{X}_P$ for which $A_P = 1$ is found, meaning that all patients before rank P refused the offer: $\forall k < P : A_k = 0$ and all patients after rank P do not receive an offer $\forall k > P : \mathbf{O}_k = \emptyset$

Transplant outcomes. Following [4], we define $Y^{\mathbf{o}}$ as the potential life expectancy for a patient $\mathbf{X}$ when they receive the organ from an accepted offer $\mathbf{o} \in \mathcal{O} \cup \{\emptyset\}$. If $\mathbf{o} = \emptyset$, it signifies the scenario where the patient does not receive an organ and thus dies before a transplant and let $Y = Y^{\mathbf{O}}$ be the resulting observed outcome. An accepting patient $\mathbf{X}_P$ receives the organ $\mathbf{O}_{P+1}$ that would have been offered to the next patient $\mathbf{X}_{P+1}$, as T_P affects the received organ where the expectation is taken over the acceptance rates and the decision times. The organ matched with a patient $\mathbf{X}$ under policy π is denoted as $\mathbf{O}_{\mathbf{X}}^{\pi} \in \mathcal{O} \cup \{\emptyset\}$.

Let $\tau(\mathbf{X}, \mathbf{O}) = \mathbb{E}[Y^{\mathbf{O}} - Y^{\emptyset} \mid \mathbf{X}]$ be the ITE or benefit of a patient $\mathbf{X}$ who receives the organ $\mathbf{O} \in \mathcal{O}$ and let $\tau_k = \tau(\mathbf{X}_k, \mathbf{O}_k)$. We define the benefit of a ranking $\mathbf{R}$ as the benefit of the patient who accepts $\tau(\mathbf{R}, \mathbf{O}_1) = \sum_{k=1}^K \tau_k \mathbb{I}[A_k = 1]$ which simplifies to $\tau(\mathbf{R}, \mathbf{O}_1) = \tau_P$ for observed runs. Finally, consider $\kappa(\mathbf{R}, \mathbf{O}_1) = \mathbb{E}[\tau(\mathbf{R}, \mathbf{O}_1)]$ as the expected benefit from a ranking and an initial organ.

Training data. We use a dataset containing offers for donor organs alongside with patients' acceptance. A set of offers is observed for each new organ. Formally, $\mathcal{D} = \{((\mathbf{x}_k, \mathbf{o}_k, a_k, t_k) : k = 1, ..., P_i) : i = 1, ..., N\}$ where N is the total number of initial offers and each tuple $(\mathbf{x}_k, \mathbf{o}_k, a_k, t_k)$ is generated by the patient distribution $p(\mathbf{X})$, the initial organ distribution $p(\mathbf{O}_1)$ and the executed policy π_{obs}.

Objective. The primary objective is to develop and validate a policy that optimizes the allocation and acceptance of organ offers in a dynamic and time-sensitive context. Specifically, given a dataset $\mathcal{D}$ that includes historical organ offers, patient characteristics, and their responses, the goal is to identify a ranking policy $\hat{\pi} : \mathcal{X}_Q \times \mathcal{O} \to Sym(\mathcal{X}_Q)$ that maximizes the expected benefit $\mathbb{E}[\kappa]$. The expectation $\mathbb{E}[\kappa]$ is taken over the distributions of patient attributes $p(\mathbf{X})$ and organ offers $p(\mathbf{O})$. The challenge lies in ensuring that the policy not only ranks patients in a way that maximizes their individual treatment effects (ITE) but also takes into account the stochastic nature of patient decisions. The performance of a policy is measured (in experiment 5) based on the following measures: the expected ITE, the expected rank of the accepting patient, the expected final CIT of the organ and the percentage of organ nonuse.

Assumptions. In the context of using an ITE estimator to test different policies, the three core assumptions of causal inference will have to be made: positivity and unconfoundedness [4], and the Stable Unit Treatment Value Assumption (SUTVA).

Sequential bias, which originates from compounding decision times that affect CIT, violates the SUTVA [11]. SUTVA assumes that the potential outcomes for one unit (patient) are independent of the treatment assignments of other units. In this case, however, the decision time of one patient impacts the organ offer for others, leading to dependent outcomes. Modeling sequential offers considering the organ as a static object would violate SUTVA, resulting in spillover effects which would lead to inaccurate ITE estimators. In section 4.1 we describe how we respect this assumption in the context of sequential organ offers.

4 DynamITE.

DynamITE is composed of four key components: a mechanism that dynamically updates organ offers, a decision time estimator, an offer acceptance estimator, and an ITE estimator. By integrating these components, DynamITE is able to create a feasible search space that facilitates the identification of the optimal ranking for organ allocation.

4.1 Updating organ offers.

Due to the sequential nature of the matching process, SUTVA is violated, causing dependencies between outcomes, and patients and organs previous to the outcomes in question. Specifically, we want the path from previous offers, patients and outcomes to be closed, ensuring the necessary conditional independence:

$$A_k, T_k \perp\!\!\!\perp \{\mathbf{X}_n, \mathbf{O}_n : n \in [1; k-1]\} \mid \mathbf{X}_k, \mathbf{O}_k \tag{1}$$

To achieve this, we rely on a set U_F of update rules to keep the offers up to date in the simulations, and ensure that outcomes are solely dependent on the patient receiving the offer and the offer itself:

$$U_F := \{\forall f \in F : \mathbf{O}_{k+1}[f] := g_f(\mathbf{O}_k[f], \hat{T}_k)\}, \tag{2}$$

where F consists of a set of features that are time dependent, f is a feature, $\hat{T}_k$ is the estimated decision time of patient $\mathbf{X}_k$ for offer $\mathbf{O}_k$ and $g_f(\mathbf{O}_k[f], \hat{T}_k)$ is a function that models how feature f depends on time. Consider u_F as a function applying this set of updates, we can then define and assume the structural causal relationship:

$$\mathbf{O}_{k+1} := u_F(\mathbf{X}_k, \mathbf{O}_k) \tag{3}$$

Applying u_F updates the time dependent features of an organ offer. Recursively applying u_F will result in further $\mathbf{O}_k$ estimates, for example:

$$\mathbf{O}_3 := u_F(\mathbf{X}_2, u_F(\mathbf{X}_1, \mathbf{O}_1)). \tag{4}$$

Conditioning on the updated offer, the path from previous offers, patients and outcomes is closed, ensuring the necessary conditional independence 1. Specific information about the considered features and update functions can be found in D.2.

4.2 PatientNet.

In order to update the organ after each offer and to estimate offer acceptance we train a model that jointly estimates decision time and the acceptance probability in a multi-task fashion. Consider PatientNet, shown in figure 2, as an estimator with parameters θ_Φ, θ_T and θ_A for: the shared representation, $\phi = \Phi_{\theta_\Phi}(\mathbf{X}, \mathbf{O})$; the decision time, $\hat{T} = T_{\theta_T}(\phi)$; the acceptance probability, $\hat{A} = A_{\theta_A}(\phi)$. We define a Mean Squared Error loss ($\mathcal{L}_{MSE}$) for $\hat{T}$ and a binary cross-entropy loss ($\mathcal{L}_{BCE}$) for $\hat{A}$ such that the total loss becomes: $\mathcal{L}_{PatNet}(\theta_\Phi, \theta_T, \theta_A) := \mathcal{L}_{MSE}(\theta_\Phi, \theta_T) + \gamma \mathcal{L}_{BCE}(\theta_\Phi, \theta_A)$, where $\gamma \in [0, 1]$ is a hyperparameter controlling the trade-off between the different prediction tasks. We seek the saddle point $(\theta_\Phi^*, \theta_T^*, \theta_A^*)$ by solving the following optimisation problem: $(\theta_\Phi^*, \theta_T^*, \theta_A^*) := \arg\min_{\theta_\Phi, \theta_T, \theta_A} \mathcal{L}_{PatNet}(\theta_\Phi, \theta_T, \theta_A)$. These parameters are learned by minimizing $\mathcal{L}_{PatNet}(\theta_\Phi, \theta_T, \theta_A)$ using stochastic gradient descent-based optimization.

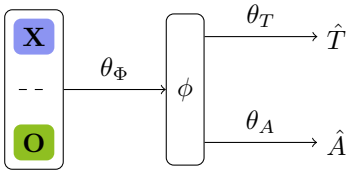

Figure 2: PatientNet Architecture: the patient $\mathbf{X}$ and organ $\mathbf{O}$ vectors are concatenated. A representation ϕ is learnt across both estimation tasks. The output of the model is the estimated decision time $\hat{T}$ and the estimated acceptance probability $\hat{A}$. Patient and organ color codes are the same as the match made by DynamITE in figure 1.

4.3 The DynamITE policy.

The optimization problem. Consider $\xi(\mathbf{R}, k)$ as a a function that returns $\mathbf{X}_k$ in $\mathbf{R}$. Suppose that τ can be estimated by any ITE estimator: $\hat{\tau} = \tau_{\theta_\tau}(\mathbf{X}, \mathbf{O})$. We can define DynamITE $\hat{\pi}(\mathcal{X}_Q, \mathbf{O}_1)$, a policy that maximizes a closed-form approximation of κ, for which a derivation is provided in appendix B, as follows:

$$\arg\max_{\mathbf{R} \in Sym(\mathcal{X}_Q)} \quad \sum_{k=1}^{K} \hat{\tau}_k \hat{A}_k \prod_{n=1}^{k-1} (1 - \hat{A}_n) \tag{5}$$

$$\text{s.t.} \quad \hat{A}_k = A_{\theta_A}(\Phi_{\theta_\Phi}(\mathbf{X}_k, \hat{\mathbf{O}}_k)), \qquad \forall k \tag{6}$$

$$\hat{\tau}_k = \tau_{\theta_\tau}(\mathbf{X}_k, \hat{\mathbf{O}}_{k+1}), \qquad \forall k \tag{7}$$

$$\hat{\mathbf{O}}_{k+1} = u_F(\mathbf{X}_k, \hat{\mathbf{O}}_k), \qquad \forall k \tag{8}$$

$$\mathbf{X}_k \neq \mathbf{X}_{k'}, \qquad \forall k \neq k' \tag{9}$$

$$\mathbf{X}_k = \xi(\mathbf{R}, k), \qquad \forall k \tag{10}$$

$$\hat{\mathbf{O}}_1 = \mathbf{O}_1 \tag{11}$$

Here, constraint 6 models the acceptance probabilities using an updated organ estimate, constraint 7 models the estimated ITE using an updated organ estimate and constraint 8 is responsible for the organ updates. Constraints 9 and 10 represent ranking constraints. Constraint 9 ensures that no two patients in the ranking are assigned the same ranks while constraint 10 enforces that the ranking of patients is consistent with the order dictated by the policy.

Solving the optimization problem. To solve the optimization problem 5, we rely on a heuristic initialization to provide a high-quality starting point for a local search algorithm. DynamITE initialises

a ranking by ranking the patients in ascending order using simple Dynamics Aware (Dyna) scores:

$$\text{Dyna}(\mathbf{X}, \mathbf{O}_1) := \frac{A_{\theta_A}(\Phi_{\theta_\Phi}(\mathbf{X}, \mathbf{O}_1))^{\alpha_A}}{T_{\theta_T}(\Phi_{\theta_\Phi}(\mathbf{X}, \mathbf{O}_1))^{\alpha_T}} \tau_{\theta_\tau}(\mathbf{X}, \mathbf{O}_1)^{\alpha_\tau}, \tag{12}$$

where $\alpha_A, \alpha_T, \alpha_\tau$ are tunable hyperparameters; representing the respective importance of acceptance, decision time and ITE. The initial ranking is then further improved using any local search algorithm. For our experiments in section 5, we use a local search based on adjacent swaps.

5 Evaluation.

Supplemental materials regarding robustness, the search, synthetic functions, hyperparameters, scenario generation, DynamITE, data and code availability can be found in appendix C and D.

5.1 PatientNet: Acceptance and decision time estimation.

Experimental setup. We create semi-synthetic training set $\mathcal{D}$ using real liver-patient pair data from the UNOS dataset [7] with synthetic outcomes. Similarly to [4], we train two kernel density estimators (KDEs), one on patients and one on organs, and use them to generate new patient and organ vectors. We first generate a set of initial organs, meaning their CIT is set to 0, and for every organ we generate a set of patients which represents $\mathcal{X}_Q$. Since we are using data for livers, we rank patients according to their MELD scores. We construct a synthetic acceptance function based on organ-patient compatibility conditions [32]. These conditions, together with the numerical value of CIT and MELD will result in a score $\beta(\mathbf{X}, \mathbf{O})$ which will be turned into an acceptance probability: $A = \sigma(\beta(\mathbf{X}, \mathbf{O}))$ [26]. We also construct a decision time function $T = \lambda A(1-A)\exp(-(A-\frac{1}{2})^2) + \mathcal{N}_T$. Both λ and the hyperparameters in β are tuned such that the average CIT of accepted organs and the average acceptance rates coincide with real statistics. Each simulated and observed tuple $(\mathbf{x}_k, \mathbf{o}_k, a_k, t_k)$ is added to $\mathcal{D}$. We then split $\mathcal{D}$ into a train and test set: $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{test}}$ respectively. Finally, we train PatientNet on $\mathcal{D}_{\text{train}}$ such that it is able to predict acceptance probabilities and decision times.

Benchmarks. The methods we use as benchmarks are traditional ML methods that have been previously used to model acceptance (as discussed in section 2): linear regression, logistic regression [14, 26], Support Vector Machine (SVM), AdaBoost and Random Forests (RF) [5]. For each benchmark, 2 separate models are trained (if possible): one for acceptance and one for decision time. We train and compare different ML models on their performance in predicting the acceptance rate and decision time for a given organ-patient pair. As these models could be used to rank patients, we also report the AUROC for the acceptance estimations.

Results. In table 1 we report the results which show that within the benchmarks, both Logistic Regression and Random Forests seem to be the best choices. However, PatientNet significantly outperforms all benchmarks on the acceptance metrics while being on par with Random Forests for predicting decision times.

Table 1: Performance metrics on $\mathcal{D}_{\text{test}}$ for acceptance (BCE, AUROC, AUPRC, and Brier score) and decision time (MSE) for different ML models. Standard deviation is reported in brackets next to each score.

Models	BCE	AUROC	AUPRC	Brier score	MSE
Linear Regression	n.a.	n.a.	n.a.	n.a.	.176 (.486)
Logistic Regression	.124 (.506)	.777	0.169	.029 (.144)	n.a.
SVM	.142 (.529)	.563	0.066	.031 (.157)	.192 (.698)
AdaBoost	.192 (.248)	.819	0.214	.039 (.101)	.103 (.254)
RF	.154 (1.613)	.889	0.419	.023 (.115)	.037 (.201)
PatientNet	**.067** (.298)	**.966**	**0.478**	**.019** (.090)	**.036** (.123)

5.2 DynamITE: Ranking patients.

ITE estimator. We use an OrganITE model [4] as ITE estimator and train real organ-patient pair data from the UNOS dataset [7] with synthetic outcomes. However, we make sure that CIT has a

negative effect on patient outcomes. We will use the same OrganITE model for all policies that make use of an ITE estimator to keep the comparisons fair.

Benchmarks. We formulate the following benchmarks, $\pi(\mathcal{X}_Q, \mathbf{O}_1)$: (i) MELD [18]; (ii) MELD-Na [15]; (iii) maximal acceptance (MaxAcc), which ranks the patients based on their estimated acceptance rate: $A_{\theta_A}(\Phi_{\theta_\Phi}((\mathbf{X}_k, \mathbf{O}_1))$ in descending order; (iv) minimal decision time (MinTime), which ranks the patients based on their estimated decision time: $T_{\theta_T}(\theta_\Phi(\mathbf{X}_k, \mathbf{O}_1)$ in ascending order, (v) TransplantBenefit (TB) [21], which ranks the patients based on their ITE: $\tau(\mathbf{X}_k, \mathbf{O}_1)$ in descending order. We compare the benchmarks with DynamITE, which searches for a ranking that maximizes κ. It is important to note that MELD and MELD-Na are organ invariant. While MaxAcc, MinTime and TB consider organ-specific features, they still only consider its initial features.

Experimental setup. We use kernel density estimators (KDEs) to generate initial organs and for each organ, a waiting list $\mathcal{X}_Q$. Next, we test each policy for each generated scenario and compute the average rank of the accepting patient (APTR), CIT, ITE and the percentage of nonused organs for each policy. Each benchmark has access to the trained OrganITE and PatientNet models however the mentioned metrics are obtained using the synthetic acceptance, decision time and OrganITE.

Results. Table 2 shows the performance of the policies across the different metrics. The MaxAcc policy, maximizing acceptance, significantly reduces APTR, CIT and nonuse. However, this strategy is not effective in terms of ITE compared the ITE-based policies: TB and DynamITE. Paradoxically, MinTime results in the highest APTR, CIT and Nonuse. This is because, on average, patients who refuse offers answer faster than patients who would accept, which leads to a naive prioritisation of patients who refuse, letting the organ deteriorate. While TB reports the highest ITE from the benchmarks, its other metrics suggest room for optimisation. This is where DynamITE capitalizes on and achieves higher ITEs with lower APTR and CIT.

Table 2: Average ITE, acceptance rank, cold ischemic time and nonuse percentage for policies tested over 10 different runs of each of the 1000 KDE generated scenarios. Standard deviation is reported in brackets next to each score.

Policies	**ITE (days gained)**	**APTR**	**CIT**	**Nonuse**
MELD	926 (34)	9.36 (.22)	7.22 (.11)	21.8% (1.1%)
MELD-Na	939 (33)	9.66 (.32)	7.39 (.13)	22.7% (.7%)
MaxAcc	1097 (35)	**2.43** (.12)	**5.32** (.16)	**14.5%** (1.1%)
MinTime	684 (41)	26.07 (.64)	10.64 (.16)	31.4% (.9%)
TB (w/ OrganITE)	3429 (61)	9.86 (.25)	7.44 (.08)	22.5% (1%)
DynamITE	3515 (108)	7.71 (.32)	6.91 (.12)	21.3% (.7%)
DynamITE (w/ search)	**3582** (53)	7.87 (.26)	7 (.11)	21.9% (.9%)

6 Discussion

DynamITE through the lens of ML. DynamITE represents a novel approach in organ allocation that incorporates both patient acceptance, decision timing, and dynamic updates to organ quality. In general, moving toward computational models that more closely reflect the complexities of real-world operational and clinical environments represents a significant step forward. Future work could explore strategic decision-making by patients based on historic acceptance data and extend DynamITE to all relevant organ types.

DynamITE through a clinical lens. Late and repeated turndowns of organ offers negatively impact organ quality by increasing cold ischemia time (CIT). DynamITE shows that explicitly modelling offer acceptance and decision time dynamics can increase both the placement efficiency and overall benefit of match-runs. This offers new perspectives for reducing the reliance on out-of-sequence offering, and thereby make allocation more just and transparent. Finally, reducing the average rank of the accepting patient eases the operational burden on the healthcare system by decreasing the number of organ matches that need to be evaluated.

References

[1] Elisa Allen, Rhiannon Taylor, Alexander Gimson, and Douglas Thorburn. Transplant benefit-based offering of deceased donor livers in the United Kingdom. *Journal of Hepatology*, 81(3):471–478, September 2024. Publisher: Elsevier BV.

[2] Barış Ata, Anton Skaro, and Sridhar Tayur. Organjet: Overcoming geographical disparities in access to deceased donor kidneys in the united states. *Management Science*, 63(9):2776–2794, 2017.

[3] Jeroen Berrevoets, Ahmed Alaa, Zhaozhi Qian, James Jordon, Alexander ES Gimson, and Mihaela Van Der Schaar. Learning queueing policies for organ transplantation allocation using interpretable counterfactual survival analysis. In *International Conference on Machine Learning*, pages 792–802. PMLR, 2021.

[4] Jeroen Berrevoets, James Jordon, Ioana Bica, Mihaela van der Schaar, et al. Organite: Optimal transplant donor organ offering using an individual treatment effect. *Advances in neural information processing systems*, 33:20037–20050, 2020.

[5] Dimitris Bertsimas, Jerry Kung, Nikolaos Trichakis, David Wojciechowski, and Parsia A Vagefi. Accept or decline? an analytics-based decision tool for kidney offer evaluation. *Transplantation*, 101(12):2898–2904, 2017.

[6] Dimitris Bertsimas, Theodore Papalexopoulos, Nikolaos Trichakis, Yuchen Wang, Ryutaro Hirose, and Parsia A Vagefi. Balancing efficiency and fairness in liver transplant access: tradeoff curves for the assessment of organ distribution policies. *Transplantation*, 104(5):981–987, 2020.

[7] John M. Cecka. The unos scientific renal transplant registry–2000. *Clinical Transplants*, pages 1–18, 2000.

[8] Agnes Debout, Yohann Foucher, Katy Trébern-Launay, Christophe Legendre, Henri Kreis, Georges Mourad, Valérie Garrigue, Emmanuel Morelon, Fanny Buron, Lionel Rostaing, Nassim Kamar, Michèle Kessler, Marc Ladrière, Alexandra Poignas, Amina Blidi, Jean-Paul Soulillou, Magali Giral, and Etienne Dantan. Each additional hour of cold ischemia time significantly increases the risk of graft failure and mortality following renal transplantation. *Kidney International*, 87(2):343–349, February 2015. Publisher: Elsevier BV.

[9] T.M. Egan, S. Murray, R.T. Bustami, T.H. Shearon, K.P. McCullough, L.B. Edwards, M.A. Coke, E.R. Garrity, S.C. Sweet, D.A. Heiney, and F.L. Grover. Development of the new lung allocation system in the united states. *American Journal of Transplantation*, 6(5):1212–1227, May 2006. Publisher: Elsevier BV.

[10] David S. Goldberg, Benjamin French, James D. Lewis, Frank I. Scott, Ronac Mamtani, Richard Gilroy, Scott D. Halpern, and Peter L. Abt. Liver transplant center variability in accepting organ offers and its impact on patient survival. *Journal of Hepatology*, 64(4):843–851, April 2016.

[11] Juan C David Gomez, Amy L Cochran, and Gabriel Zayas-Caban. Unveiling bias in sequential decision making: A causal inference approach for stochastic service systems. *arXiv preprint arXiv:2307.07879*, 2023.

[12] HRSA. Organ Donation Statistics. Technical report, HRSA (Health Resources and Services Administration), 2023.

[13] Ajay K. Israni, David A. Zaun, Katrina Gauntt, Cory R. Schaffhausen, Cinthia Lozano, Warren T. McKinney, Jonathan M. Miller, and Jon J. Snyder. OPTN/SRTR 2022 annual data report: Deceased organ donation. *American Journal of Transplantation*, 24(2):S457–S488, February 2024. Publisher: Elsevier BV.

[14] Sang-Phil Kim, Diwakar Gupta, Ajay K Israni, and Bertram L Kasiske. Accept/decline decision module for the liver simulated allocation model. *Health care management science*, 18:35–57, 2015.

[15] W Ray Kim, Scott W Biggins, Walter K Kremers, Russell H Wiesner, Patrick S Kamath, Joanne T Benson, Erick Edwards, and Terry M Therneau. Hyponatremia and mortality among patients on the liver-transplant waiting list. *New England Journal of Medicine*, 359(10):1018–1026, 2008.

[16] Kristen L. King, Sulemon G. Chaudhry, Lloyd E. Ratner, David J. Cohen, S. Ali Husain, and Sumit Mohan. Declined offers for deceased donor kidneys are not an independent reflection of organ quality. *Kidney360*, 2(11):1807–1818, November 2021. Publisher: Ovid Technologies (Wolters Kluwer Health).

[17] Xun Luo, Joseph Leanza, Allan B Massie, Jacqueline M Garonzik-Wang, Christine E Haugen, Sommer E Gentry, Shane E Ottmann, and Dorry L Segev. Meld as a metric for survival benefit of liver transplantation. *American Journal of Transplantation*, 18(5):1231–1237, 2018.

[18] Michael Malinchoc, Patrick S Kamath, Fredric D Gordon, Craig J Peine, Jeffrey Rank, and Pieter CJ Ter Borg. A model to predict poor survival in patients undergoing transjugular intrahepatic portosystemic shunts. *Hepatology*, 31(4):864–871, 2000.

[19] Sumit Mohan, Mariana C. Chiles, Rachel E. Patzer, Stephen O. Pastan, S. Ali Husain, Dustin J. Carpenter, Geoffrey K. Dube, R. John Crew, Lloyd E. Ratner, and David J. Cohen. Factors leading to the discard of deceased donor kidneys in the United States. *Kidney International*, 94(1):187–198, July 2018. Publisher: Elsevier BV.

[20] Michael S. Mulvihill, Hui J. Lee, Jeremy Weber, Ashley Y. Choi, Morgan L. Cox, Babatunde A. Yerokun, Muath A. Bishawi, Jacob Klapper, Maragatha Kuchibhatla, and Matthew G. Hartwig. Variability in donor organ offer acceptance and lung transplantation survival. *The Journal of Heart and Lung Transplantation*, 39(4):353–362, April 2020. Publisher: Elsevier BV.

[21] James Neuberger, Alex Gimson, Mervyn Davies, Murat Akyol, John O'Grady, Andrew Burroughs, Mark Hudson, UK Blood, et al. Selection of patients for liver transplantation and allocation of donated livers in the uk. *Gut*, 57(2):252–257, 2008.

[22] OPTN and SRTR. Optn/srtr 2022 annual data report. Technical report, U.S. Department of Health and Human Services, Health Resources and Services Administration, 2024. Accessed: 2024-08-22.

[23] Burhaneddin Sandikci, Sait Tunc, and Bekir Tanriover. A new simulation model for kidney transplantation in the united states. In *2019 Winter Simulation Conference (WSC)*. IEEE, December 2019.

[24] SRTR. *KPSAM 2015 User Guide*, 2015. Accessed: 2024-08-31.

[25] SRTR. *TSAM 2015 User Guide*, 2015. Accessed: 2024-08-31.

[26] SRTR. *LSAM 2019 User Guide*, 2019. Accessed: 2024-08-31.

[27] Darren E. Stewart, Victoria C. Garcia, John D. Rosendale, David K. Klassen, and Bob J. Carrico. Diagnosing the decades-long rise in the deceased donor kidney discard rate in the united states. *Transplantation*, 101(3):575–587, March 2017. Publisher: Ovid Technologies (Wolters Kluwer Health).

[28] Andrew Wey, Nicholas Salkowski, Bertram L. Kasiske, Ajay K. Israni, and Jon J. Snyder. Influence of kidney offer acceptance behavior on metrics of allocation efficiency. *Clinical Transplantation*, 31(9), August 2017. Publisher: Wiley.

[29] Russell Wiesner, Erick Edwards, Richard Freeman, Ann Harper, Ray Kim, Patrick Kamath, Walter Kremers, John Lake, Todd Howard, Robert M. Merion, Robert A. Wolfe, and Ruud Krom. Model for end-stage liver disease (MELD) and allocation of donor livers. *Gastroenterology*, 124(1):91–96, January 2003. Publisher: Elsevier BV.

[30] Nicholas L. Wood, Douglas B. Mogul, Emily R. Perito, Douglas VanDerwerken, George V. Mazariegos, Evelyn K. Hsu, Dorry L. Segev, and Sommer E. Gentry. Liver simulated allocation model does not effectively predict organ offer decisions for pediatric liver transplant candidates. *American Journal of Transplantation: Official Journal of the American Society of Transplantation and the American Society of Transplant Surgeons*, 21(9):3157–3162, September 2021.

[31] Jinsung Yoon, Ahmed M Alaa, Martin Cadeiras, and Mihaela Van Der Schaar. Personalized donor-recipient matching for organ transplantation. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[32] Yunwei Zhang, Anne Hu, Yingxin Lin, Yue Cao, Samuel Muller, Germaine Wong, and Jean Yee Hwa Yang. simkap: simulation framework for the kidney allocation process with decision making model. *Scientific Reports*, 13(1):16367, 2023.

A Summary of Notations

Table 3: Summary of notations used in section 3

Symbol	Definition	Description
$\mathcal{X}$	Patient space	Set of all possible patients, $\mathcal{X} \subset \mathbb{R}^d$
$\mathcal{O}$	Organ space	Set of all possible organs, $\mathcal{O} \subset \mathbb{R}^e$
$\mathbf{X}$	Patient vector	A patient, $\mathbf{X} \in \mathcal{X}$
$\mathbf{O}$	Organ vector	An organ, $\mathbf{O} \in \mathcal{O} \cup \{\emptyset\}$; $\emptyset$ represents no organ
A	Offer response	Patient's response to an organ offer, $A \in \{0, 1\}$; $A = 1$ if accepted, $A = 0$ if refused
T	Decision time	Time between offer and acceptance, $T \in \mathbb{R}_+$
$\mathcal{P}(\mathcal{X})$	Powerset of $\mathcal{X}$	Set of all subsets of $\mathcal{X}$
$\mathcal{X}_Q$	Waiting list	A set of patients on the waitlist, $\mathcal{X}_Q \subset \mathcal{P}(\mathcal{X})$
K	Waitlist size	Number of patients on the waitlist, $K = \lvert\mathcal{X}_Q\rvert$
$\mathrm{Sym}(\mathcal{X}_Q)$	Symmetric group	Set of all permutations (rankings) of $\mathcal{X}_Q$
$\mathbf{R}$	Patient ranking	A ranking of waitlisted patients, $\mathbf{R} \in \mathrm{Sym}(\mathcal{X}_Q)$
k	Sequence number	Position in the ranking, $k \in \mathbb{Z}^+$, $1 \leq k \leq K$
$\mathbf{X}_k$	Patient at rank k	The k-th patient in the ranking, $\mathbf{X}_k \in \mathcal{X}_Q$
$\mathbf{O}_k$	Organ at step k	Organ offered to $\mathbf{X}_k$
A_k	Response at rank k	Acceptance ($A_k = 1$) or refusal ($A_k = 0$) by $\mathbf{X}_k$
T_k	Decision time at rank k	Time taken by $\mathbf{X}_k$ to respond
P	Acceptance rank	Rank at which the organ is accepted, $A_P = 1$
$\mathbf{X}_P$	Accepting patient	Patient who accepts the organ offer
$Y^{\mathbf{o}}$	Potential outcome	Life expectancy for $\mathbf{X}$ when receiving organ $\mathbf{o}$
Y	Observed outcome	Actual life expectancy, $Y = Y^{\mathbf{O}}$
$\tau(\mathbf{X}, \mathbf{O})$	ITE	Expected benefit of $\mathbf{X}$ receiving $\mathbf{O}$, $\tau(\mathbf{X}, \mathbf{O}) = \mathbb{E}[Y^{\mathbf{O}} - Y^{\emptyset} \mid \mathbf{X}]$
τ_k	ITE at rank k	ITE for patient $\mathbf{X}_k$, $\tau_k = \tau(\mathbf{X}_k, \mathbf{O}_k)$
$\tau(\mathbf{R}, \mathbf{O}_1)$	Ranking benefit	Actual benefit of ranking $\mathbf{R}$, $\tau(\mathbf{R}, \mathbf{O}_1) = \sum_{k=1}^{K} \tau_k \mathbb{I}[A_k = 1]$
$\kappa(\mathbf{R}, \mathbf{O}_1)$	Expected benefit	Expected benefit from ranking $\mathbf{R}$ and initial organ $\mathbf{O}_1$, $\kappa(\mathbf{R}, \mathbf{O}_1) = \mathbb{E}[\tau(\mathbf{R}, \mathbf{O}_1)]$
$\mathcal{D}$	Dataset	Set of historical offers and responses, $\mathcal{D} = \{((\mathbf{x}_k, \mathbf{o}_k, a_k, t_k) : k = 1, \ldots, P_i) : i = 1, \ldots, N\}$
N	Number of offers	Total number of initial organ offers
$p(\mathbf{X})$	Patient distribution	Probability distribution over patient attributes
$p(\mathbf{O}_1)$	Initial organ distribution	Probability distribution over initial organ attributes
π	Policy	A function defining the patient ranking
π_{obs}	Observed policy	Policy used to generate the dataset $\mathcal{D}$
$\hat{\pi}$	Estimated policy	The policy to be identified, $\hat{\pi} : \mathcal{X}_Q \times \mathcal{O} \to \mathrm{Sym}(\mathcal{X}_Q)$
$\mathbb{E}[\kappa]$	Expected benefit	Expected value of κ, averaged over $p(\mathbf{X})$ and $p(\mathbf{O})$
$\mathbb{I}[\cdot]$	Indicator function	Equals 1 if the condition is true, 0 otherwise

B DynamITE's objective

The objective. In section 3, we defined $\kappa(\mathbf{R}, \mathbf{O}_1)$ as the expected τ corresponding to a match-run. Now, given that each patient has a certain probability of accepting the offered organ, we can achieve a closed-form expression for $\kappa(\mathbf{R}, \mathbf{O}_1)$:

$$\kappa(\mathbf{R}, \mathbf{O}_1) = \sum_{k=1}^{K} \tau_k P(A_k = 1 | \mathbf{R}, \mathbf{O}_1) \tag{13}$$

The probability of a proposed transplant taking place $P(A_k = 1|\mathbf{R}, \mathbf{O}_1)$ is further broken down into two factors: (i) the probability of the organ being offered to the patient, and (ii) the probability of the

patient accepting the organ offer:

$$P(A_k = 1|\mathbf{X}_k, \mathbf{O}_k) \prod_{n=1}^{k-1} P(A_n = 0|\mathbf{X}_n, \mathbf{O}_n) \tag{14}$$

These acceptance probabilities, and the decision times of the patients, can then be jointly estimated using a PatientNet model and τ can be estimated by any ITE estimator: $\hat{\tau} = \tau_{\theta_\tau}(\mathbf{X}, \mathbf{O})$. Using the estimated values $\hat{\tau}_k$ and $\hat{A}_k$, we approximate $\kappa(\mathbf{R}, \mathbf{O}_1)$ as follows:

$$\kappa(\mathbf{R}, \mathbf{O}_1) \approx \sum_{k=1}^{K} \hat{\tau}_k \hat{A}_k \prod_{n=1}^{k-1} (1 - \hat{A}_n) \tag{15}$$

C Additional experiments

C.1 Uncertainty importance.

Experimental setup. One of our key contributions lies in the deliberate incorporation of acceptance and decision timing within DynamITE. To explore the role of acceptance in the DynamITE policy, we examine how varying levels of uncertainty in patient-side decisions impact performance. The acceptance probabilities can be adjusted as follows: $A = \sigma(\frac{\beta(\mathbf{X},\mathbf{O})}{u})$, where $u \in (0, \infty)$ represents the uncertainty parameter. By setting higher values of u, the acceptance probabilities tend to shift closer to 0.5, thereby simulating more uncertain decisions. With the use of u, we evaluate all policies similarly to the patient ranking experiments described in Section 5.

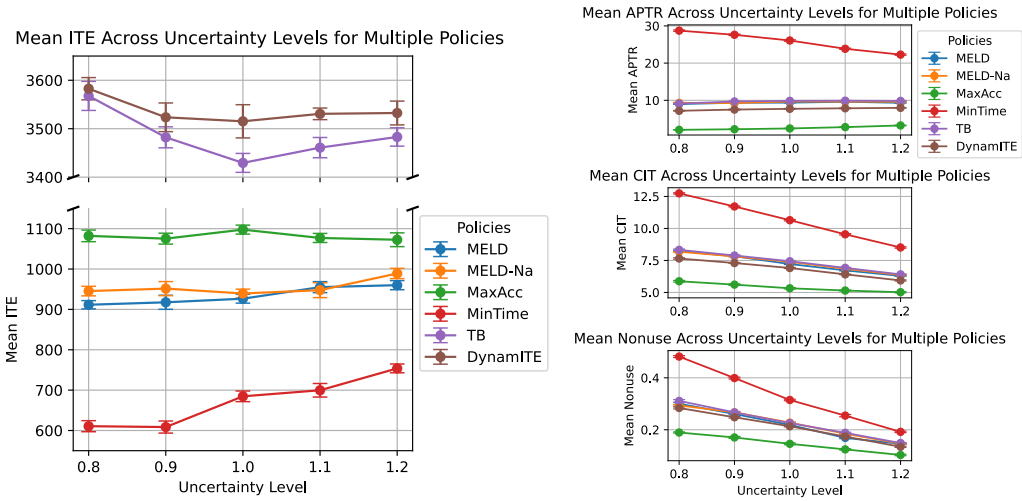

Figure 3: ITE-based performance of policies for different levels of uncertainty, over 10 different runs of each KDE generated scenario (left). Performance of policies across different levels of uncertainty in terms of APTR, CIT and Nonuse (right). Error bars are 95% confidence intervals.

Results. Figure 3 shows policy performance in function of uncertainty. Given the low acceptance rate, a higher uncertainty will, on average, increase acceptance rates, while a lower uncertainty will have the opposite effect. For uncertainty values which are not too far from the base value of 1, DynamITE outperforms other policies. However, in further regions, DynamITE starts to lose its edge as its acceptance estimator becomes less accurate.

APTR, CIT and Nonuse in function of uncertainty. In figure 3 we report performance of the different policies on APTR, CIT and Nonuse. For APTR, uncertainty shifts seem to only significantly affect MinTime, as the uncertainty increases MinTime becomes a better policy overall. In terms of both CIT and Nonuse, all policies improve significantly with a higher uncertainty.

Given that the average acceptance rate is low, increasing uncertainty will increase this acceptance rate but it will also increase decision times. For this reason, it is not evident what to expect from these experiments. However it seems that while the performance of non-ITE-based policies increases with an increase of uncertainty, ITE-based policies tend to perform better when uncertainty levels are low.

In figure 4 we report the correlations between metrics across different uncertainty levels for all policies. For both DynamITE and TB, there is a strong negative correlation between ITE and APTR. CIT and nonuse are positively correlated in all policies, especially MaxAcc and MELD, highlighting the importance of minimizing delays. All policies exhibit a negative correlation between APTR and nonuse. DynamITE shows negative correlations between ITE and CIT (-0.57) and between ITE and APTR (-0.81), indicating that its strategy effectively prioritizes patients who both accept quickly and derive significant benefit, reducing CIT and improving outcomes.

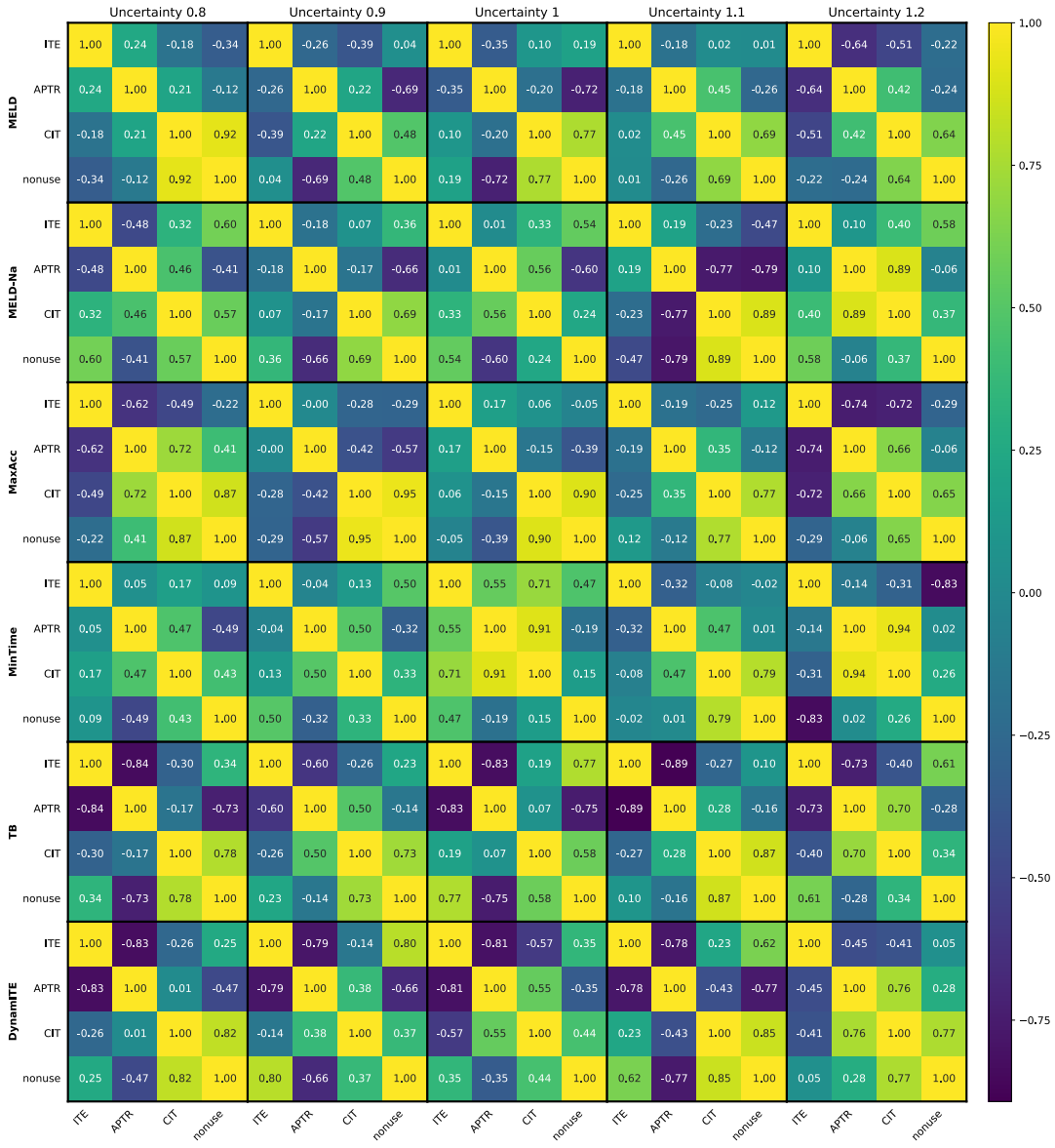

Figure 4: Correlations between ITE, APTR, CIT and Nonuse across different uncertainty levels for all policies

C.2 Search algorithm importance.

Experimental setup. DynamITE uses a search to optimize 5. In our experiments, we investigated how different configurations of the local search algorithm affect the performance of DynamITE. Specifically, we tested various combinations of the parameters $iterations$, top_k, and $accuracy$ on 100 different KDE generated scenarios. The parameter $iterations$ controls how many times DynamITE iterates over the ranking to perform adjacent swaps. The top_k parameter determines the number of top-ranked patients considered for optimization, and $accuracy$ defines the fraction of the ranking used to approximate the objective function $\kappa(\mathbf{R}, \mathbf{O}_1)$. A more detailed description of these search parameters can be found in appendix D.4. The local search is run on an Intel Core i7-1265U processor.

Results. Table 4 summarizes the results of these experiments, with the configuration parameters presented as individual columns. We report the average run time, the average estimated κ (days gained) as calculated by DynamITE using its internal estimations, and the average true κ (days gained) computed using the synthetic functions. The configuration with $iterations = 0$ corresponds to no search (i.e., the initial ranking without any local search optimization).

Table 4: Performance of DynamITE with different local search configurations over 100 KDE generated scenarios. The first row ($iterations = 0$) represents the baseline without search. For each tested configuration tuple ($iterations$, top_k, $accuracy$) the run time, estimated κ, and true κ are shown.

Iterations	**Top_k**	**Accuracy**	**Run Time (s)**	**Estimated κ (days)**	**True κ (days)**
0	0	.00	0.55	n.a.	2853.58
1	5	.30	**3.34**	2695.41	2902.07
1	15	.30	8.52	2701.25	2902.48
1	5	.40	3.66	2710.54	2902.77
1	15	.40	9.33	2724.00	2902.99
2	5	.30	6.71	2709.73	2919.14
2	5	.40	6.85	2724.89	2919.48
2	15	.30	18.48	2721.57	2922.43
2	15	.40	18.87	2745.26	2922.48
3	5	.30	7.30	2714.44	2924.45
3	5	.40	7.35	2729.60	2924.79
3	15	.30	22.22	2729.68	2930.04
3	15	.40	22.69	**2753.28**	**2930.50**

The results demonstrate that more extensive search configurations generally lead to improved true ITE gains. For instance, increasing the number of iterations and the top_k value tends to result in higher ITE, albeit at the cost of increased computational time. Notably, the configuration with $iterations = 3$, $top\_k = 15$, and $accuracy = 0.40$ achieves the highest average true ITE of 2930.50 days but also requires the longest average run time of approximately 22.69 seconds.

These findings indicate that while a more exhaustive search can enhance the performance of DynamITE by finding better patient rankings, there is a trade-off between computational efficiency and the quality of the solution. In practice, the choice of search parameters can be tailored based on the available computational resources and the acceptable time constraints for the allocation process.

D Experimental details

D.1 Data and code availability.

The organ-patient pair data reported in sections 5 and C have been supplied by the United Network for Organ Sharing as the contractor for the Organ Procurement and Transplantation Network. The interpretation and reporting of these data are the responsibility of the author(s) and in no way should be seen as an official policy of or interpretation by the OPTN or the U.S.Government.

The code used to generate the synthetic data and for the experiments in sections 5 and C is made available on `https://github.com/AlessandroMarchese/DynamITE`.

D.2 Updating organs.

We only consider CIT as feature in F as it is the main time dependent feature of an organ. We set $g_{\text{CIT}}(\mathbf{O}_k[\text{CIT}], \hat{T}_k) := \mathbf{O}_k[\text{CIT}] + \hat{T}_k$, this will result in the CIT of an organ to be updated, after each patient answer, throughout a sequence of offers.

D.3 Acceptance and decision time estimation.

Synthetic acceptance function. In section 5.2 we define how we use the function $\beta(\mathbf{X}, \mathbf{O})$ to model acceptance probabilities. Specifically, we define function $\beta(\mathbf{X}, \mathbf{O})$ as follows:

$$\begin{aligned}\beta(\mathbf{X}, \mathbf{O}) := h_1 \text{MELD}(\mathbf{X}) - h_2 \mathbf{O}[\text{CIT}] \\ - h_3 \mathbb{I}[|\mathbf{X}[\text{AGE}] - \mathbf{O}[\text{AGE}]| \leq 30] \\ - h_4 \mathbb{I}[|\mathbf{X}[\text{BMI}] - \mathbf{O}[\text{BMI}]| \leq 5] + \mathcal{N}_A\end{aligned} \tag{16}$$

Here, $\text{MELD}(\mathbf{X})$ returns the MELD score of patient $\mathbf{X}$, capped between 0 and 40. We took into consideration conditions based on the age and BMI differences between donor organ and patient. Hyperparameters h_1, h_2, h_3, h_4 have been tuned manually together with hyperparameters from the decision time function such that observed acceptance rates and cold ischemic times would coincide with the actual statistics. We find the following hyperparameters to model acceptance appropriately and resulting in the statistics: $h_1 = 0.01, h_2 = 0.15, h_3 = 0.22, h_4 = 0$. So we find it is best to leave BMI compatibility out of the acceptance function.

In our acceptance function, we take the MELD score as an urgency metric [17], we model the impact of CIT on acceptance and take some organ-patient compatibility (AGE and BMI) factors into account.

PatientNet Hyperparameters. We tuned the models in Table 1 and trained our PatientNet model with the hyperparameters listed in Table 5.

Table 5: PatientNet Hyperparameters

Component	Hyperparameters
Shared Representation $\Phi_{\theta_\Phi}(\mathbf{X}, \mathbf{O})$	Dense(64, L2 Regularization), LeakyReLU Dense(32, L2 Regularization), LeakyReLU
Acceptance Head $A_{\theta_A}(\phi)$	Dense(1), Sigmoid Activation
Decision Time Head $T_{\theta_T}(\phi)$	Dense(1), ReLU Activation
Task Loss Weight γ	1
Training Parameters	Maximum Epochs: 1000 Early Stopping Patience: 50

D.4 Patient ranking policies.

Generated scenarios. For the results in table 2 we generate 1000 initial organs using KDEs and setting $\mathbf{O}[\text{CIT}] := 0$. Next for each of these organs we generate a waitlist $\mathcal{X}_Q$ of size 50 containing KDE generated patients. Each organ-waitlist pair $(\mathbf{O}_1, \mathcal{X}_Q)$ represents a generated scenario. We let each policy handle each scenario for 10 different runs.

Organ nonuse. In our experiments, a nonused organ is an organ that is refused by the last patient of the ranking. In that case, the organ will not result in an ITE.

Dyna scores hyperparameters. In expression 4.3 we introduced the dyna scores. We manually tuned hyperparameters $\alpha_A, \alpha_T, \alpha_\tau$ on a different set of generated organ-waitlist pairs. For experiments 5 and 5.2 we used $\alpha_A = 0.6, \alpha_T = 1, \alpha_\tau = 0.7$.

DynamITE. Results for DynamITE in table 2 are shown with and without the search algorithm. For the search algorithm we use a local search based on local swaps. This local search must constantly approximate κ as it is optimizing it as objective. However, approximating $\kappa(\mathbf{R}, \mathbf{O}_1)$ is very computationally demanding as it requires 3 estimations ($\hat{A}_k$, $\hat{\tau}_k$ and $\hat{\mathbf{O}}_{k+1}$) for each patient in $\mathcal{X}_Q$. Moreover, these estimations are interdependent as the estimations for the patient at rank k can only be done once we already have the estimations for the patient at rank $k-1$, inhibiting approaches that would exploit parallelism.

For this reason, we relax the search by adding tunable search parameters:

- $iterations$: how many times DynamITE goes over the ranking in ascending order and tries to perform adjacent swaps. In experiment 5, this value is set to 1.
- top_k: determines over how many patients DynamITE should optimize. These will be the patients with ranks 1 up to top_k. We consider patients starting from the top as a change at the top in the ranking is expected to have a bigger impact than a change further down the ranking. This is because a swap between two patients at ranks k and $k+1$ will change acceptance probabilities, decision times and ITE of all patients at ranks larger than $k+1$, while leaving patients with ranks smaller than k completely unaffected. In experiment 5, this value is set to 5.
- $accuracy$: up to which patient $\kappa(\mathbf{R}, \mathbf{O}_1)$ is computed. In expression 15, K will be set to the closest integer to $accuracy * K$ resulting in an incomplete estimation of $\kappa(\mathbf{R}, \mathbf{O}_1)$. In experiment 5 we set this value to 0.4, resulting in $\kappa(\mathbf{R}, \mathbf{O}_1)$ being estimated for only the first 40% of patients.

These parameters have been set low for computational reasons. We let DynamITE's local search algorithm run for 4 seconds on an Intel Core i7-1265U processor, resulting in a 67 days improvement in ITE across the whole population. Nonetheless, this simple search significantly improves DynamITE's performance as shown in table 2.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: DynamITE, on average, results in better benefits and earlier acceptance in our policy related experiments: table 2 and figure 3.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The key limitation, the use of semi-synthetic data instead of fully real, historical data, is proposed as future work in the discussion section 6. Technical limitations such as causal inference assumptions and the computational limitations have been addressed in 3 and D.4 respectively. A violation of the SUTVA assumption would lead to an inaccurate estimation of DynamITE's objective function, which in its turn would lead to sub-optimal decision-making. This is implicitly tested in experiment 5, as all policies, with the exception of DynamITE with the search, do not consider organ updates when estimating ITEs. Finally, important fairness factors that are typically incorporated in organ allocation policies, such as waiting times, are not taken into account as DynamITE solely optimizes for benefits. Policies that only aim at optimizing benefits could lead to unfair allocations with regards to patients' age groups, prioritizing younger patients. However we underline that these factors can be explicitly taken into account by adjusting the DynamITE policy 4.3.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Dataset and causal inference assumptions have been addressed in 3 and 3. A derivation of our approximation for κ has been provided in B.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimental setups in 5.2, 5.1 and C.1 are made clear with separate paragraphs for each experiment. They include information about the data, synthetic functions and models we used. Additional information regarding hyperparameters is also available in D.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All information regarding data and code availability is provided in D.1.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All relevant details about data, synthetic functions and hyperparameters are provided in the experimental setups in 5.2, 5.1 and C.1 and the appendix D. For more information the code can be directly consulted through D.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Standard deviations and 95% CI error bars are shown in tables 2 and 1, and figure 3. The numpy function std() is used to compute standard deviations and the matplotlib.pyplot function errorbars() is used to plot confidence intervals.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The only relevant computational limitations are regarding the search algorithm and are described in D.4 together with the relevant system information and time of execution. Compute resources regarding other experiments are not reported as they did not present any resource related challenges.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Assumptions, limitations and goals of DynamITE are clearly stated. The research utilizes semi-synthetic data derived from the UNOS (United Network for Organ Sharing) dataset, ensuring that no real patient identifiers are exposed or used without appropriate permissions. The paper includes a statement about how the policy should not be interpreted as an official policy D.1.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper includes a discussion paragraph in which the clinical impact of DynamITE is summarized 6. By stating DynamITE's goal to directly optimize for outcomes we assume that a reader understands what this implies in terms of fairness of the policy, which has not been directly addressed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: There is no need for safeguards, no data or models are released. There are no real misuse possibilities besides explicit changes in DynamITE's objective function that would cause it to prioritize bad organ-patient matches (which would only be relevant if considered as an actual policy D.1).

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We adequately refer to the creators and providers of the used models 5.2 and data D.1.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code for PatientNet, MaxAcc, MinTime and DynamITE is provided D.1.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Our work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.