informs
http://pubsonline.informs.org/journal/msom

MANUFACTURING & SERVICE OPERATIONS MANAGEMENT
Vol. 00, No. 0, Xxxxx 0000, pp. 000–000
ISSN 1523-4614, EISSN 1526-5498

Submitted to *Manufacturing & Service Operations Management*

Queueing Causal Models: Comparative Analytics in Queueing Systems

Opher Baron
Rotman School of Management, University of Toronto, opher.baron@rotman.utoronto.ca

Dmitry Krass
Rotman School of Management, University of Toronto, dmitry.krass@rotman.utoronto.ca

Arik Senderovich
School of Information Technology, York University, sariks@yorku.ca

Mark van der Laan
Department of Statistics, University of California, Berkeley,laan@stat.berkeley.edu

Zhenghang Xu
Rotman School of Management, University of Toronto, zhenghang.xu@rotman.utoronto.ca

Abstract. Problem Definition Much of the focus of queueing theory (QT) is on performance evaluation that supports comparative analytics, i.e., comparing performance measures under different interventions. However, closed-form queueing models are very sensitive to assumptions. We develop a data-driven Structured Causal Queueing Model (SCQM) - a form of structural causal model that automatically adapts to the data generating process of queuing systems, finds causal relations, and supports comparative analytics. Numerical experiments show that the accuracy of SCQM is competitive with QT even for examples where analytical queueing solutions are available. **Methodology** We employ structured causal modeling methodology that uses queueing-relevant features to develop a simulator that replicates the data generating process of the original system without any prior knowledge of process dynamics. We apply Machine Learning (ML) models for identifying the parent sets and causal relations. We then provide intervention analysis using Monte Carlo simulation. **Managerial Implications** We use queueing knowledge to develop an accurate self-adapting data-driven performance evaluator for congested systems that requires no prior knowledge of the system dynamics. Using this method, companies can perform comparative analytics of interventions for queueing systems that may not be analytically solvable.

Key words: causal inference, queueing theory, machine learning, data-driven simulation

History: This paper was first submitted on .

1. Introduction

The availability of ample operational data recorded by modern information systems carries the promise of enabling data-driven analysis of congested systems such as hospitals, call centers, manufacturing floors, and public transportation systems (Kim 2014). Such systems are extensively studied and analyzed by *Queueing Theorists* (QTs). One focus of QTs is on comparative analytics (Baron 2021), i.e., evaluation of system performance measures, such as the expected customer waiting time, the queue-length distribution, the probability of delay, etc., under different managerial interventions. Typical examples of managerial interventions include adding resources to the system when wait times are excessive or changing service priorities and rates for different customer segments.

One prominent example of comparative analytics is to quantify the impact of changes in service rates on waiting times. For instance, Deo and Jain (2019) consider the impact of speeding up urgent patients to improve their chances of positive outcomes.

The traditional queueing theory (QT) approach is to model the underlying system via elementary building blocks (e.g., the arrival and service processes), assuming knowledge of the relations among these building blocks (e.g., single-channel or multi-channel queue), their parametric or non-parametric distributions, and the (dynamic) performance of the queueing system. The system is represented as one of many queueing theory models for which closed-form, approximate, or numerical estimates for performance measures of interest may be obtained. Using available data to estimate model parameters, interventions are modeled as changes to system parameters or dynamics, representing each intervention as a scenario of parameter values substituted into the appropriate formulas to estimate system performance.

A key challenge of this approach, in addition to its reliance on queueing expertise, is that adapting models to complex or non-standard systems often requires a substantial modeling effort, as queueing models are sensitive to assumptions and may require switching between analytical and simulation tools. For example, consider speed-up intervention for a fraction of customers in a single-station system. If we assume the system is a single-channel queue with Markovian arrivals and service (under both regular and speed-up conditions), the system can be represented as an M/G/1 queue with service times that are a mixture of two exponential distributions. In this case, closed-form expressions of performance measures are available (see Section 3). However, if arrivals do not follow a Poisson process, the model becomes a G/G/1 queue, for which only approximate expressions or numerical methods are available. If the arrival rate decreases in response to longer waiting time, an entirely different queueing model would need to be used, for which deriving even approximate expressions may be challenging. Thus, the QT approach assumes that the causal interconnections between various system components are captured and modeled based upon the QT's expertise. In contrast, the data-driven approach we propose adapts flexibly to changes in the data-generating process, without requiring explicit re-specification of a new model.

Data scientists (DS) use Machine Learning tools to analyze complex systems using available data (Kohavi et al. 2002). Their approach to the problem above would be to employ general-purpose techniques that are not specific to queueing theory and that would automatically adapt to the available data. One family of such techniques specifically designed to analyze causal relationships in dynamic systems are Structural Causal Models (SCMs) (Pearl 2009). While many successful applications of SCMs have been reported across several domains, there are significant challenges when the causal structure of the underlying system is not known and has to be discovered from data.

In this paper we introduce a class of SCMs - Structural Causal Queueing Models (SCQM) - that are specifically adapted to data generated by queueing systems. We describe the key system components and the conditions under which time ordering of events automatically implies a default causal structure that is rich enough to encompass many possible relationships between the system components (including the dependence of arrivals on past waiting times). We then apply ML-based refinement mechanism to automatically uncover the underlying dynamic structure in the data feed produced by the process, and use it to estimate the relationships between system variables (via a second ML algorithm). Applying Monte Carlo simulation to the resulting SCM, we obtain estimates of system performance measures under different interventions. We present the first systematic application of SCMs to queueing models. We demonstrate SCQM enables DS to carry out comparative analysis of complex queueing systems, even in the absence of detailed modeling expertise. SCQM complements traditional QT approaches by offering a flexible alternative that learns from data. Moreover, the resulting model automatically adapts to the dynamics of the underlying data-generating process by learning both the causal structure and functional relationships directly from observed data, enabling it to capture complex or nonstandard system behaviors without prior modeling assumptions.

Using the example of an M/M/1 queue with speed-ups, we show that a DS employing SCQM can achieve accuracy comparable to that of a QT using closed-form analytical solutions. While the QT benefits from exact expressions, their application still requires parameter estimation from data—just as the DS must estimate the structural causal model. In practice, the estimation uncertainties faced by both approaches often lead to similar levels of performance, highlighting the practical value of SCQM in settings where traditional analytical tools are not readily available.

When the example above is generalized to settings such as non-Poisson arrivals or multiple servers, classical queueing analysis has to rely on approximation or numerical methods while the (numerical) SCQM approach naturally adapts to such complexities without chang ing its methodology. Our numerical experiments show that the SCQM estimates are on par with the most accurate approximation methods.

Finally, we further generalize the M/M/1 example to the case where speed-ups are state-dependent, placing it beyond the realm of any known analytical or approximation QT results that we are aware of. At the same time, SCQM adapts effortlessly to this setting, yielding accurate counterfactual estimates of system performance. To summarize, our main contributions are:

1. We introduce Structural Causal Queueing Models (SCQM), a framework for comparative analytics in queueing systems. Our data-driven estimation methodology relies on natural structural assumptions about the data-generating process, which is general for many queues and requires no additional knowledge of queueing theory. This methodology consists of unsupervised and supervised machine learning models and a Monte Carlo simulator. This framework shifts the paradigm from working with a specialized queueing model that requires knowledge of both the system and queueing theory to SCQM, which DS can apply without the latter.
2. We demonstrate the DS approach by specifying and estimating an SCQM in estimating the post intervention, "interventional", waiting time in M/M/1 and GI/M/m models with speed-ups. We contrast this DS approach that can be applied to general queueing systems with a QT approach. Here, the data-driven QT approach estimates the queueing primitives of the arrival and service rates and plug these into a closed-form expression.
3. We evaluate our techniques numerically. The key takeaway is that the DS approach yields as accurate results as the QT one. Moreover, in the GI/M/2 with speed-ups and GI/M/1 with speed-ups examples, while QT approximations may underperform in certain settings, QT can rely on simulation to achieve high accuracy. However, simulation typically requires a well-specified model, whereas our approach learns both the structure and behavior from data. We view SCQM as a novel tool designed to extend the capabilities of QT, providing a complementary, data-driven approach to classical methodologies.

The rest of the paper is organized as follows. After reviewing the relevant literature in Section 2, we introduce the QT solution for the M/M/1 with speed-ups, and formally define the data-driven QT approach based on the exact analysis by Pollaczek-Khinchine formula in Section 3. In Section 4, we introduce SCMs, SCQM structure, and intervention analysis based on Monte Carlo simulation for a known SCQM. Section 5 outlines the estimation techniques within the proposed inference framework when the underlying causal graph is known. Section 6 establishes the default causal structure, and discussed how to refine the default causal graph to uncover the causal relations in the system; overview of the SCQM approach is provided in Section 6.3. Section 7 demonstrates the usefulness of our approach with numerical results in various settings. Conclusions and directions for future research are presented in Section 8.

2. Literature Review

We review relevant literature on causal inference, feature selection, queueing theory, and simulation.

Structural Causal Models and Causal Inference and with Longitudinal Data There is a rich body of literature on Structural Causal Models and Causal Inference. We refer the reader to a comprehensive volume by Peters et al. (2017) and references therein. We employ their basic framework in Section 4. Our interest lies in applications to data generated by queueing systems, where event logs have event timestamps (such as arrivals and service starts). There are several approaches for analyzing causal effects within such

longitudinal data structure, such as G-computation and targeted minimum loss estimation (TMLE). Readers can refer to Van der Laan et al. (2011), Van der Laan and Rose (2018) for a comprehensive review. Our work appears to be the first use of the G-computation in the queueing context.

Queueing Theory We refer the reader to Shortle et al. (2018) and references therein for some basic results in queueing theory, including the ones we use in Section 3; approximation formulas used in Section 7.2 can be found in see Buzacott and Shanthikumar (1993), Whitt (1983), Whitt and You (2022). Investigation of statistical estimates of queueing model parameters, which are essential for data-driven methods in queueing, have a long tradition starting from Cox (1955), and extending to more recent work Cruz et al. (2021) and a recent survey Asanjarani et al. (2021). In contrast to our approach, these methods assume a pre-specified queueing model.

Machine Learning in Queues. The use of ML for queueing models has been gaining interest, particularly for cases where no closed-form solutions or accurate approximations exist, see e.g., Nii et al. (2020), Baron et al. (2023), Sherzer et al. (2025) for use of ML to evaluate performance of a queue with a known structure. Several studies consider learning queueing dynamics from data. E.g., Palomo and Pender (2020) showed that Lindley's recursion by Lindley (1952), which describes waiting time dynamics of the First In First Out (FIFO) single-server queue, can be accurately learned from the data generated by a G/G/1 system. Palomo and Pender (2021) extended the analysis to systems with multiple queues and demonstrate that the dynamics of tandem queues can be learned from data as well. Kyritsis and Deriaz (2019) show that ML models can be trained to estimate average waiting time from real-life data (event log) consisting of only customer arrival and departure time stamps. Our approach, which also relies on ML techniques, is more ambitious as we seek to reconstruct full set of causal relations from the event log data (making very few assumptions about the data generating process) and use the resulting model for dynamic simulation of interventions on a system.

Queue Mining. An alternative approach to extract queueing models automatically from event logs using ML techniques is known as *queue mining*, see Senderovich et al. (2016). Queue mining starts from a family of queueing models (such as G/G/1, G/G/c) and aims to find the model that best fits the observed event log; discrete event simulation can then be applied to evaluate interventions. In contrast, while our causal model uses queueing features it does not assume a particular queueing family as the data generating mechanism, and inherently allows for time and state dependencies between elements of the underlying system.

Data-Driven Simulation. An active area of research focuses on calibration of accurate simulation models with data for estimating out-of-sample performance, focusing on capturing the dynamics of the underlying system, e.g., Carter and Blake (2004), Elalouf and Wachtel (2021). Another important direction is quantifying input uncertainty of the simulation model, see Corlu et al. (2020) for a comprehensive survey. Papers mentioned above assume the simulator itself has been pre-specified. In contrast, our approach seeks to automate the process of data-driven estimation of system dynamics and automatic generation of the simulation model. A somewhat similar goal is pursued by Camargo et al. (2020), who develop a data-driven automated

simulation discovery from event logs. However, similar to queue mining, they assume a pre-specified family of models, while our approach does not make this assumption.

3. Data-driven Analysis of Queueing Systems under Interventions: The M/M/1 system with speed-ups

We consider the problem of estimating the performance of a stationary queueing system with exponential inter-arrival and service times under different speed-up policies, which we use as our running example throughout the paper. We start by describing the problem, as well as the queueing theorist's (QT's) approach to analyzing this analytically tractable system. While queueing theory also encompasses numerical and simulation-based methods, we focus here on the analytical route to highlight the contrast with our data-driven approach.

Customers arrive to a *single server, First In First Out (FIFO)* queue according to a homogeneous Poisson process with rate λ. Customers are indexed with $i \in \{1, 2, \ldots, n\}$, in the order of their arrivals. Before service starts, a binary *action* $A_i \in \{0, 1\}$ is taken to either *speed-up* ($A_i = 1$) customer i or perform *regular* service ($A_i = 0$). Service times follow an exponential distribution with the rate μ_1 if $A_i = 1$ and μ_0 if $A_i = 0$, where $\mu_1 > \mu_0$ (i.e., the service time given $A_i = 1$ is stochastically shorter).

Actions A_i are independently and identically distributed (i.i.d.) following a Bernoulli distribution with a speed-up probability $p \in [0, 1]$, i.e., $P(A_i = 1) = p$. For convenience we represent queueing primitives (λ, μ_0, μ_1) as the vector $\theta := (\tau_T, \tau_{S,0}, \tau_{S,1})$, where $\tau_T = 1/\lambda, \tau_{S,0} = 1/\mu_0, \tau_{S,1} = 1/\mu_1$. We refer to this system as an *M/M/1 with speed-ups*.

The available data consists of n observations of customer visits, with the record of the visit of customer $i \in 1, \ldots, n$ comprised of (1) the inter-arrival time T_i, denoting the time between the arrivals of customer $i - 1$ and customer i, where $T_0 = 0$, (2) the waiting time W_i, (3) the speed-up action A_i, and (4) the service time S_i. Thus, the data record for customer i is given by $O_i := \{T_i, W_i, A_i, S_i\}$, and the overall data includes n observations of O_i that are generated under unknown θ and p.

Based on data we answer the following comparative question: what is the expected waiting time under the speed-up probability of interest $p^* \neq p$? While we consider the average waiting time as the performance metric, the methodology we present is applicable to the analysis of other metrics.

3.1. The Data-Driven Queueing Theory Approach

Here we describe an analytical QT approach for a data-driven analysis of the M/M/1 with speed-ups. When θ and p^* are known, the service time follows the hyper-exponential distribution, with parameters (p^*, μ_0, μ_1). Thus, M/M/1 with speed-ups is an M/G/1, and the QT applies the Pollaczek-Khinchine formula to obtain a closed-form expression for the expected waiting time. Let $\nu^2_{S,a}$ be the second moment of service time, conditional on action $a \in \{0, 1\}$ and observe that the conditional service time is exponential, $\nu^2_{S,a} = 2\tau^2_{S,a}$. Thus, assuming that utilization is less than one:

$$\rho^* := \frac{(1 - p^*)\tau_{S,0} + p^*\tau_{S,1}}{\tau_T} < 1, \tag{1}$$

the expected waiting time is (see the Pollaczek-Khinchine formula in (6.2), p .257 in Shortle et al. (2018)):

$$W_{\theta,p^*} = \frac{(1-p^*)\tau_{S,0}^2 + p^*\tau_{S,1}^2}{\tau_T - ((1-p^*)\tau_{S,0} + p^*\tau_{S,1})}. \tag{2}$$

Therefore, the QT knows the waiting time given p^* (when (1) does not hold, $W_{\theta,p^*} = \infty$). The QT estimates θ from the available data, e.g., using Maximum Likelihood Estimation (MLE), as:

$$\hat{\theta} = (\hat{\tau}_T, \hat{\tau}_{S,0}, \hat{\tau}_{S,1}) = \left(\frac{\sum_{i=1}^n T_i}{n}, \frac{\sum_{i=1}^n (1-A_i)S_i}{n - \sum_{i=1}^n A_i}, \frac{\sum_{i=1}^n A_i S_i}{\sum_{i=1}^n A_i} \right). \tag{3}$$

Substituting $\hat{\theta}$ for θ in (2) yields the ***QT solution*** for the counterfactual estimate, W_{θ,p^*}, for p^* satisfying (1):

$$W_{\hat{\theta},p^*} = \frac{(1-p^*)\hat{\tau}_{S,0}^2 + p^*\hat{\tau}_{S,1}^2}{\hat{\tau}_T - ((1-p^*)\hat{\tau}_{S,0} + p^*\hat{\tau}_{S,1})}. \tag{4}$$

We note that while we could have used an example based on a more complex queueing model, the analytical queueing theory approach is notoriously fragile: even a minor tweak to the model may lead to major changes in the analytical approach; this is illustrated when the arrival distribution is generalized to "G" in Section 7.2. In contrast, the DS approach we develop in the following sections is quite robust.

4. Structural Causal Models in Queueing Systems

In this part, we start by defining structural causal models (SCMs) following standard notation from Peters et al. (2017). We next adapt this definition to data generated by a queueing systems, defining Structural Causal Queueing Model (SCQM) and show how one can use queueing features to employ SCQMs together with Monte Carlo simulation to analyze the impact of interventions on various performance metrics.

We consider a discrete set of random variables $\mathcal{O} = \{V_i : i = 1, \ldots, \infty\}$ that may be observed in a system, such as inter-arrival times, service times, server identifiers, etc. We use uppercase letters for random variables (or set of random variables) and lowercase letters for their realizations. An *event log* $\bar{O}_d = \{V_1, \ldots, V_d\} \subset \mathcal{O}$ tracks the values of first d variables, where $\bar{o}_d = \{v_1, v_2, \ldots, v_d\}$ is the realization of $\bar{O}_d$. The following definition is based on (Peters et al. 2017, Ch.6).

Definition 1 (Structural Causal Model (SCM)) *An SCM is a triple $\mathcal{Q} = (\bar{O}_d, \mathcal{G}, \mathcal{F})$, where:*

- *$\bar{O}_d \in \mathcal{O}$ are the d variables in the event log.*
- *$\mathcal{G}$ is an acyclic structural causal graph (SCG) with nodes as variables in $\bar{O}_d$ and arcs representing causal relationships. Parents $PA(V_i)$ of $V_i \in \bar{O}_d$ are variables causing changes in V_i.*
- *$\mathcal{F}$ is a set of d assignment functions:*

$$V_i := f_i(PA(V_i), \epsilon_i), \quad i = 1, \ldots, d,$$

where $\{\epsilon_1, \ldots, \epsilon_d\}$ are jointly independent noise terms, $\epsilon_j \sim U[0,1]$ for $j = 1, \ldots, d$.

We assume that the terms $\epsilon_j, j = 1, \ldots, d$ are uniform without loss of generality since any alternative i.i.d distribution can be incorporated into the $\mathcal{F}$ functions using the inverse of the CDF. We also note that we could have assumed that only the parent sets are given, and then have constructed $\mathcal{G}$ from these parent sets.

Following Peters et al. (2017), we only deal with SCMs whose SCG is acyclic, i.e., there are no direct or indirect causal loops between the variables. Such a SCM induces a unique joint distribution $P^{\mathcal{Q}}$ over $\bar{O}_d$ (Peters et al. 2017, Proposition 6.3); we will refer to $P^{\mathcal{Q}}$ as the *observational* distribution. The distribution $P^{\mathcal{Q}}$ satisfies the Markov property showing that $P^{\mathcal{Q}}$ can be factorized into probabilities of each variable conditioned on its parents:

PROPOSITION 1. *(Peters et al. 2017, Proposition 6.31)*
Given an SCM $\mathcal{Q}$ with an acyclic SCG $\mathcal{G}$ and a joint distribution $P^{\mathcal{Q}}$, this distribution satisfies the **Markov property** *with respect to $\mathcal{G}$:*

$$P^{\mathcal{Q}}(V_1, V_2, \ldots, V_d) = \prod_{j=1}^{d} P(V_j | PA(V_j)). \tag{5}$$

When all elements of an SCM are known, it can be thought of as a dynamic system that provides the framework for causal analysis, where the causal relationships are specified by the graph $\mathcal{G}$.

4.1. Structural Causal Queueing Model (SCQM)

In this section we adapt the general SCM Definition 1 to the data stream that is generated by a queueing system. We start with an important assumption that observation times are available for all variables, as well as defining the "observed prior to" relationship. An implicit assumption we make is that all relevant queueing features are included, in or can be calculated from, data in the SCM.

ASSUMPTION 1 **(Observation Time and Observational Order)**. *For any variable $V \in \bar{O}_d$, a non-negative function $t(V)$ representing the observation time of V is available. For $U, V \in \bar{O}_d$ we say that "U is* observed prior *to V", denoted as $t(U) \prec t(V)$, if (a) $t(U) \leq t(V)$ and (b) $P(t(U) < t(V)) > 0$.*

Intuitively, $t(U) \prec t(V)$ implies that, while the observational times of U and V may coincide in some realizations, there is a positive probability of observing a realization $\bar{o}_d$ where $t(U) < t(V)$.

We next state the common (and intuitive) assumption that causation can only move forward in time, sometimes referred to as "non-anticipation property."

ASSUMPTION 2 **(Non-Anticipation)**. *Suppose $U \in PA(V)$, i.e., variable U is included in the parent set of V. Then, U must be observed prior to V, i.e., $t(U) \prec t(V)$.*

Intuitively, if U is observed prior to V then the realization of U is available before V is realized and thus U can be a causal parent of V. The previous two assumptions ensure that causal relationships are "predictive," i.e., based only on information available before a given variable is realized. This leads to the following key property.

PROPOSITION 2. *Suppose Assumptions 1 and 2 hold for some SCM $\mathcal{Q} = (\bar{O}_d, \mathcal{G}, \mathcal{F})$. Then, the causal graph $\mathcal{G}$ is acyclic.*

Proposition 2 holds since if there is a directed path from U to V in causal graph $\mathcal{G}$ then, by Assumption 2, U must be observed prior to V (since the "observed prior to" relationship is transitive). Thus, there cannot be a directed path from V to U. Note that as long as the policy is non-anticipatory, even queues with feedback do not violate the acyclic property, as the feedback only depends on past information.

Next, we define the structural properties of an SCM generated by a queueing system by ensuring that: (1) the event log comprises of customer visits, (2) key components are present, and (3) Proposition 2 holds.

Definition 2 (SCQM) *An SCQM is an SCM generated by a data stream originating from a queueing system. It satisfies Assumptions 1 and 2 well as the following properties:*

1. Customer Visit Record: *The variables in the event log are organized into $k-$dimensional subsets called "customer visit records", with the visit of customer i represented by $O_i = \{V_i^1, \ldots, V_i^k\} \subseteq \mathcal{O}$. The event log consists of n customer visit records, i.e., $d = kn$ and $\bar{O}_d = \cup_{i=1}^n O_i$. The composition of customer visit records is stationary, i.e., for $j \neq i$ and $m \in \{1, \ldots, k\}$, if $V_j^m \in O_j$ then $V_i^m \in O_i$.*
2. Queueing Components: *Each customer visit record O_i must include non-negative variables T_i, W_i, S_i representing, respectively, the inter-arrival, waiting, and service times of customer $i \geq 1$. Variable T_1 is the arrival time of the first customer 1. If customers are intervened on, O_i must also contain an intervention variable $A_i \in O_i$. Customer visit record may also include additional variables such as customer types, server characteristics, other relevant queueing features, etc.*
3. Arrival Ordering: *Customer visits are time-ordered with respect to customer arrivals. Formally, for $i \in \{1, \ldots, n\}$ the arrival time of customer i is denoted by $t(T_i)$ and is given by*

$$t(T_1) = T_1,\ t(T_i) = \sum_{k=1}^{i} T_i = t(T_{i-1}) + T_i \ \text{ for } i = 2, \ldots, n. \tag{6}$$

   Thus, $t(T_i) \prec t(T_j)$ whenever $i < j$.
4. Observational Ordering: *Within each visit record $O_i, i = 1, \ldots, n$ the observations times $t(T_i), t(W_i), t(S_i)$ are given by*

$$t(W_i) = t(T_i) + W_i,\ \ t(S_i) = t(T_i) + W_i + S_i, \tag{7}$$

   implying that $t(T_i) \prec t(W_i) \prec t(S_i)$. We refer to $t(W_i)$ as the service start time *and $t(S_i)$ as the* service end time *or* departure time *of customer i. The inter-arrival time T_i is observed prior to the service start time $t(W_i)$, which is observed prior to the departure time $t(S_i)$.*
5. Co-Observational Partition: *We assume any other component $V_i \in O_i$ (including the intervention indicator A_i) must be co-observed with either T_i, W_i or S_i, i.e., the observation time $t(V_i) \in$*

$\{t(T_i), t(W_i), t(S_i)\}$. We denote the corresponding co-observational sets by $\mathcal{C}(T_i), \mathcal{C}(W_i)$, and $\mathcal{C}(S_i)$; these sets form a partition of the customer visit record O_i. These co-observational sets preserve the "prior to" relationship of the underlying variables, i.e., any $U \in \mathcal{C}(T_i)$ is observed prior to any $V \in \mathcal{C}(W_i)$, which is in turn observed prior to any $W \in \mathcal{C}(S_i)$. Note that $t(V_i)$ belongs to the same co-observational set as V_i. In particular, $t(X) \in \mathcal{C}(X)$, where $X \in \{T_i, W_i, S_i\}$.

Properties (1.)-(5.) of Definition 2 ensure that the event log contains the essential components of any queueing system: inter-arrival, waiting and service times, in addition to intervention indicators; that the visit records corresponding to different customers are ordered by customer arrivals; and that there is a natural time ordering of all variables within each customer's visit record. These properties are quite intuitive and should hold for data generated by most queueing systems. For example, property (2.) allows for batch arrivals by setting $T_i = 0$ for all arrivals within a batch. Property (5.) can be relaxed to allow certain variables (such as actions A_i) to have their own observational times that may not coincide with those of T_i, W_i or S_i (e.g., we could have $t(A_i) \in [t(T_i), t(W_i)]$), as long as the "prior to" relations for all variables within a customer visit record O_i are well-defined. We assume property (5.) holds for simplicity.

For any element of $V_i^k \in O_i$ of the $i-th$ customer visit, Assumption 2 implies that the parent set $PA(V_i^k)$ must be contained in the *time-limited event log*, i.e.,

$$PA(V_i^k) \subset \bar{O}(t(V_i^k)) = \{U \in \bar{O}_d | t(U) \prec t(V_i^k)\}. \tag{8}$$

It is important to note that, in general, the time-limited log, $\bar{O}(t)$, may include elements of different customer visit records. For example, we may observe inter-arrival times of many customers before observing the service completion time of even the first customer. Definition 2 imposes only a partial order on the observational sequence of variables in $\bar{O}(t)$. For example, parts (4.) and (5.) imply that if $T_i \in \bar{O}(t)$ then $\mathcal{C}(T_j) \in \bar{O}(t)$ for all $j < i$. Similarly, if $S_i \in \bar{O}(t)$ then $\mathcal{C}(W_i), \mathcal{C}(T_i) \in \bar{O}(t)$. If, in addition to Definition 2, SCQM is known to possess some additional properties, other observational relationships may hold as well.

We illustrate the framework defined in the preceding two sections with our running example. Note that here the observations times $t(T_i), t(W_i), t(S_i)$ are sufficient queueing features for the analysis.

EXAMPLE 1 (M/M/1 WITH SPEED-UPS). We assume that each customer visit record includes the inter-arrival time, waiting time, speed-up action, and service time of customer i: $O_i := \{T_i, W_i, A_i, S_i\}$, with $k = |O_i| = 4$. The event log $\bar{O}_d = \bigcup_{i=1}^n O_i$ consists of n customer visits, with $d = 4n$.

The functional relationships $\mathcal{F}$ of Definition 1 are specified by

$$T_i = -\frac{1}{\lambda}\log(1-\epsilon_{T_i}) \qquad i \in \{1,\dots,n\} \tag{9}$$

$$A_i = \mathbb{1}\left(\epsilon_{A_i} \leq p\right) \qquad i \in \{1,\dots,n\} \tag{10}$$

$$S_i = (1-A_i)\left(-\frac{1}{\mu_0}\log(1-\epsilon_{S_i})\right) + A_i\left(-\frac{1}{\mu_1}\log(1-\epsilon_{S_i})\right) \qquad i \in \{1,\dots,n\} \tag{11}$$

$$W_i = \max(0, W_{i-1} + S_{i-1} - T_i) \qquad i \in \{1,\dots,n\}, \tag{12}$$

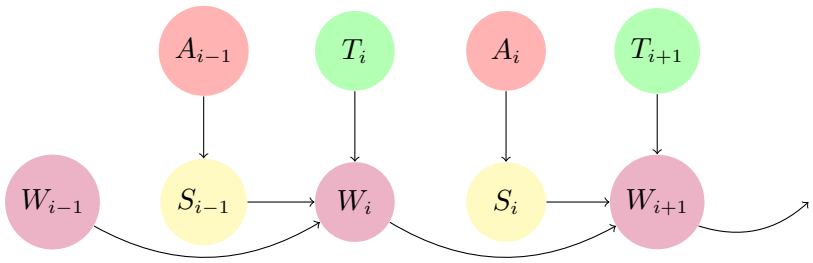


Figure 1 The causal graph of the M/M/1 with speed-ups

where $\mathbb{1}(\cdot)$ is the binary indicator function and $\epsilon_{T_i}, \epsilon_{A_i}, \epsilon_{S_i}$ are uniform noise terms.

These equations formalize the model described in Section 3 above. Equations (9,11) specify the exponential inter-arrival and the conditional service distributions given A_i. The action in (10) is a speed-up probability following a Bernoulli(p) random variable. Finally, (12) is known as the "Lindley recursion" (Lindley 1952) representing the queueing dynamics by linking W_i to W_{i-1}, S_{i-1}, and T_i. Note that the deterministic service discipline implies that the noise term ϵ_{W_i} is not required in (12).

The relations above induce the following parent relationships:

$$PA(S_i) = \{A_i\},\ i \in \{1,\dots,n\}, \text{where } PA(S_0) = \emptyset \tag{13}$$

$$PA(W_i) = \{W_{i-1}, S_{i-1}, T_i\},\ i \in \{1,\dots,n\}, \text{where } W_0 = S_0 = 0, PA(W_0) = \emptyset \tag{14}$$

$$PA(T_i) = P(A_i) = \emptyset,\ \ i \in \{1,\dots,n\}. \tag{15}$$

We note that in the terminology of Definition 2, action A_i is assumed to be co-observed with the service entry time W_i, i.e., $A_i \in \mathcal{C}(W_i)$, and thus equation (13) agrees with part (5.) of the definition. On the other hand, (14) assumes that the service entry time W_{i-1} and service end time S_{i-1} of the previous customer are observed prior to W_i, which does not follow from Definition 2 - this relationship is based on the additional information that the data generating process is $M/M/1$. While in the current section we assume that this information is given, in Section 6 we show how it can be learned from the event log data.

Since the model described above is an instance of SCQM, Proposition 2 ensures that the causal graph $\mathcal{G}$ is acyclic. This graph is depicted on Figure 1.

We can easily generalize the model above in a variety of ways e.g., by replacing (9) and (11) with:

$$T_i = F_T(\epsilon_{T_i}) \qquad i \in \{1,\dots,n\} \tag{16}$$

$$S_i = (1 - A_i)F_S^0(\epsilon_{S_i}) + A_i F_S^1(\epsilon_{S_i}) \qquad i \in \{1,\dots,n\}, \tag{17}$$

where $F_T(\cdot)$ is the inverse of the inter-arrival time distribution, and $F_S^j(\cdot), j = 0,1$ is the inverse of the service time distribution under $A_i = 0$ and $A_i = 1$, respectively, with $F_S^0(\cdot)$ stochastically dominated by $F_S^1(\cdot)$. This yields a *G/G/1 with speed-ups*, where these distributions may be estimated from data. The graph $\mathcal{G}$ and the parent sets (13-15) are not affected.

4.2. Monte Carlo Performance Evaluation for a Given SCM

In this and the following subsection we discuss performance evaluation (with and without interventions) for a specified causal model. This discussion applies to general SCMs for which the Markov property (Proposition 1) holds, and do not require the specific SCQM structure of the previous section.

Definition 3 (Performance Evaluation) *Consider an SCM $\mathcal{Q} = (\bar{O}_d, \mathcal{G}, \mathcal{F})$ and an m-dimensional function $\psi : \mathcal{O}_d \to \mathbb{R}^m$, where $\mathcal{O}_d$ refers to the support of $\bar{O}_d$. We refer to the m components of ψ as "performance metrics" and evaluate the expected performance of $\mathcal{Q}$ under ψ by:*

$$\Psi(\bar{O}_d) = \mathbb{E}_{P^{\mathcal{Q}}}(\psi(\bar{O}_d)) = \int_{\bar{o}_d} \psi(\bar{o}_d) dP^{\mathcal{Q}}(\bar{o}_d), \tag{18}$$

where $P^{\mathcal{Q}}$ is the observational distribution of $\mathcal{Q}$.

In some (very simple) cases, the distribution $P^{\mathcal{Q}}$ can be obtained analytically, and (18) can be evaluated in closed form. More generally, we can use Proposition 1 to evaluate (18) numerically via Monte Carlo simulation, as described in Algorithm 1 in Appendix A. The algorithm generates M independent observations of event logs, generated by following the causal graph $\mathcal{G}$ from root nodes (i.e., variables V_i that have empty parent sets) to leaf nodes. Each simulation yields an observation of the performance measure ψ. The number of observations M can be adjusted to reach the desired accuracy level. Note that this process does not assume any time-ordering with respect to customer visit records, allowing for very general causal dependencies between various elements of the system.

EXAMPLE 2 (WAITING TIME EVALUATION FOR M/M/1 WITH SPEED-UPS). Using the framework of Example 1 above, the expected waiting time is:

$$\psi(\bar{O}_d) = \frac{1}{n}\sum_{i=1}^{n} W_i, \;\; d = 4n. \tag{19}$$

We use $\psi(\bar{O}_d)$ as our performance measure.

We start by deriving the observational distribution $P_p^{\mathcal{Q}}$ induced by the speed-up policy p. Since the induced causal graph is acyclic, we can use the Markov factorization in Proposition 1, as well as structural assignments (9) to (12) to obtain the close-form expression (as before, $d = 4n$) :

$$\begin{aligned}
&P_p^{\mathcal{Q}}(\bar{O}_d = \bar{o}_d) \\
&= \prod_{i=1}^{n} P_p^{\mathcal{Q}}(T_i = t_i) P_p^{\mathcal{Q}}(W_i = w_i | W_{i-1} = w_{i-1}, S_{i-1} = s_{i-1}, T_i = t_i) P_p^{\mathcal{Q}}(A_i = a_i) P_p^{\mathcal{Q}}(S_i = s_i | A_i = a_i) \\
&= \begin{cases} \prod_{i=1}^{n} \left(\lambda \mathrm{e}^{-\lambda t_i}\right) \cdot \left(p^{a_i}(1-p)^{1-a_i}\right) \cdot \left((1-a_i)\mu_0 \mathrm{e}^{-\mu_0 s_i} + a_i \mu_1 \mathrm{e}^{-\mu_1 s_i}\right) \\ \qquad\qquad \text{if } w_i = \max(0, w_{i-1} + s_{i-1} - t_i), \forall i \in \{1, 2, \ldots, n\}, \\ 0 \qquad\qquad \text{otherwise.} \end{cases}
\end{aligned} \tag{20}$$

Equation (20) implies that the observational probability density non-zero only if the components $w_1, w_2, \ldots, w_n$ of $\bar{o}_d$ exactly follow the dynamics in (12).

Conceptually, we can now use the previous expression in (18) to compute the expected waiting time under policy p. However, this approach is not practical, as it requires generation of all possible waiting times trajectories along which the system may evolve; numerical evaluation is more natural. For this example, Algorithm 1 simplifies to Algorithm 2 (see Appendix A). It is important to note that, provided the causal graph and assignment functions are correct, a DS using this algorithm gets *completely accurate numerical evaluation* of the selected performance measure (here, the expected waiting time) under policy p.

4.3. Intervention Analysis

While the SCMs can be used to estimate the observational distribution and performance measures, their true power lies in their ability to provide a framework for causal reasoning and support intervention analysis, i.e., evaluation of the performance measures that would be realized if specific changes to the elements of the SCM are made. We formally define interventions based on Definition 6.8 in (Peters et al. 2017, Ch.6).

Definition 4 (Intervention) *Consider an SCM $\mathcal{Q} = (\bar{O}_d, \mathcal{G}, \mathcal{F})$ and its observational distribution $(V_1, V_2, \ldots, V_d) \sim P^{\mathcal{Q}}$. Assume that for some $i \in \{1, \ldots, d\}$ we replace the assignment for V_i by:*

$$V_i := \tilde{f}_i(\widetilde{PA}_i, \epsilon_i). \tag{21}$$

This replacement is called an "intervention" *and denoted with the* do operator*: $do(V_i := \tilde{f}_i(\widetilde{PA}_i, \epsilon_i))$; we say that variable V_i has been intervened on. The resulting new SCM is denoted as $\tilde{\mathcal{Q}} = (\bar{O}_d, \tilde{\mathcal{G}}, \tilde{\mathcal{F}})$ and the entailed distribution of $\tilde{\mathcal{Q}}$ is called and* interventional distribution*, represented by:*

$$P^{\tilde{\mathcal{Q}}} := P^{\mathcal{Q}; do(V_i := \tilde{f}_i(\widetilde{PA}_i, \epsilon_i))}. \tag{22}$$

We note that interventions defined with (21) are very flexible may and include changes to both, the parent set $\widetilde{PA}_i \neq PA_i$ (i.e., changes to the causal relationships) or the assignment function, $\tilde{f}_i(\widetilde{PA}_i, \epsilon_i) \neq f_i(PA_i, \epsilon_i)$. The latter may include changes to the distributions of the stochastic terms, even if the parent set remains invariant (since assignment functions incorporate inverse cdf's to transform the $U[0,1]$ noise terms to the desired distribution).

Definition 4 naturally extends to interventions over several variables. Let $I \subseteq \{1, 2, \ldots, d\}$ be *a set of intervened variables*. For each $i \in I$, we will correspondingly have (21) and hence the *group intervention* and the *group interventional distributions* are:

$$V_i := \tilde{f}_i(\widetilde{PA}_i, \epsilon_i) \quad \forall i \in I,\ P^{\tilde{\mathcal{Q}}} := P^{\mathcal{Q}; do(V_i := \tilde{f}_i(\widetilde{PA}_i, \epsilon_i), i \in I)}. \tag{23}$$

Our goal is to compute the interventional distribution $P^{\tilde{\mathcal{Q}}}$ and evaluate the performance metric with respect to this distribution. The challenge, that will become more apparent when we discuss data-driven

estimated in Section 5 below, is that we have never seen data from $P^{\tilde{\mathcal{Q}}}$ and we want to estimate it based on the data from the observational distribution $P^{\mathcal{Q}}$. To resolve this problem, Robins (1986) developed G-computation (also known as truncated factorization or manipulation theorem) as a standard method to represent $P^{\tilde{\mathcal{Q}}}$ with the elements of $P^{\mathcal{Q}}$ combined with the intervention specifications.

THEOREM 1 **(G-computation)**. *Consider an SCM $\mathcal{Q} = (\bar{O}_d, \mathcal{G}, \mathcal{F})$ and an intervention set $I \subset \{1, \ldots, d\}$. The SCM $\tilde{\mathcal{Q}}$ evolves from $\mathcal{Q}$ after the intervention $do(V_i := \tilde{N}_i)$, where $\tilde{N}_i := \tilde{f}_i(\widetilde{PA}_i, \epsilon_i)$ for $i \in I$, assuming that $(\tilde{N}_i, i \in I)$ follow joint density $\tilde{P}$. Then, the interventional distribution $P^{\tilde{\mathcal{Q}}} = P^{\mathcal{Q};do(V_i:=\tilde{N}_i)}$ can be written as:*

$$\begin{aligned} P^{\mathcal{Q};do(V_i:=\tilde{N}_i,i\in I)}(V_1,\ldots,V_d) &= \prod_{i\notin I} P^{\mathcal{Q};do(V_i:=\tilde{N}_i,i\in I)}(V_j|PA_j) \prod_{i\in I} P^{\mathcal{Q};do(V_i:=\tilde{N}_i,i\in I)}(\tilde{N}_i) \\ &= \tilde{P}(V_i, i\in I) \prod_{i\notin I} P^{\mathcal{Q}}(V_i|PA_i). \end{aligned} \tag{24}$$

The first equality from (24) follows from the Markovian decomposition (5) in Proposition 1, while the second one holds since for $i \in I$ and $j \notin I$ we have $P^{\mathcal{Q};do(V_i:=\tilde{N}_i)}(V_j|PA_j) = P^{\mathcal{Q}}(V_j|PA_j)$ because the distribution of V_j conditioned on its parent PA_j is not affected by the intervention on V_i. This limited effect of the intervention is known as the principle of independent mechanisms that guarantees that changing one of the causal mechanisms does not affect the others, see (Peters et al. 2017, Ch.2).

When it comes to evaluating the performance measure $\Psi^{do(V_k:=\tilde{N}_k,k\in I)}(\bar{O}_d)$ under the intervention, the situation is similar to the non-intervention case: while we could use (24) in place of the observational distribution P^Q in (18) and try to analytically evaluate the resulting expression, this approach is not fruitful except in simple cases. It is more practical to adapt the Monte Carlo simulation approach (i.e., our Algorithm 1 in Appendix 1) to the intervention case. Indeed, from Theorem 1 it follows that the only adjustment needed is to use the intervened assignment function (23) in place of the original assignment function when generating an observation for an intervened variable $v_i, i \in I$; see Algorithm 3 in Appendix 1 for details.

EXAMPLE 3 (THE M/M/1 WITH SPEED-UPS UNDER DIFFERENT POLICIES). Continuing from Example 2, recall that (10) specifies that action A_i speeds up the service for customer i with probability p. We would like to consider an intervention where p is replaced by $p^* \neq p$. This merely entails changing the assignment function in (10) to:

$$A_i = \mathbb{1}\left(\epsilon_{A_i} \leq p^*\right) \qquad \forall i \in \{1, 2, \ldots, n\}. \tag{25}$$

Note that this does not affect $PA(A_i)$, which is empty in both the original and the intervened SCM. The preceding expression leads directly to the interventional distribution

$$P^{\tilde{\mathcal{Q}}}_{p^*} = P^{\mathcal{Q};do\left(A_i=\mathbb{1}\left(\epsilon_{A_i}\leq p^*\right),\forall i\right)} \tag{26}$$

Applying Theorem 1 to obtain $P^{\tilde{\mathcal{Q}}}_{p^*}$ only entails replacing the distribution of A_i in (20) with the interventional distribution of A_i given by

$$\tilde{P}_{\theta,p^*}(A_i = a_i) = (p^*)^{a_i}(1-p^*)^{a_i}. \tag{27}$$

Similarly, adapting Algorithm 1 to obtain the Monte Carlo estimate of the interventional value of $\Psi^{do(A_i=\mathbb{1}(\epsilon_{A_i}\leq p^*),\forall i)}(\bar{O}_d)$ only requires replacing (10) with (25) in the Algorithm.

REMARK 1. An important condition underlying the intervention analysis described above is that there are no "unobserved confounders," i.e., no exogenous factors affecting the assignment functions of both the intervened variable and the variables affected by the intervened variable. In Example 3 this would be violated if some exogenous factor (e.g., gender of the customer) affected both $f_i(A_i)$ and $f_i(S_i)$ (since $PA(S_i) = A_i$). Essentially, we assume that the parent sets capture all factors affecting the distributions in our model. Technically, this assumption only needs to hold for SCQM components that influence the measure of interest Ψ.

5. Data-Driven Estimates with Known SCQM Structure

In this section we partially relax the assumption that all components of the SCQM, including the values of all the parameters in the assignment functions, are known. While we still assume that we know the *structure* of the SCQM (i.e., the correct causal graph $\mathcal{G}$ and the induced parent sets), we no longer assume that the specifications of the assignment functions $\mathcal{F}$ are known, and we thus estimate these from data. We focus on the general case, where no assumptions are made about the forms of the assignment functions (the *non-parametric* case). The simplified estimation procedures for the special *paramrtric* case where the forms of all assignment functions are assumed to be specified up to a set of finite parameters are discussed in Appendix B.

Since we need to estimate the forms of the assignment functions $f_i, i = 1,\ldots,d$ from data, the number of functional forms could be equal to the size of the available data $\bar{o}_d$, making statistical estimation not tenable. Thus, we leverage SCQM structure to make the following assumption:

ASSUMPTION 3 **(Bounded Stationary Form)**. *For each customer visit record $O_i = \{V_i^1,\ldots,V_i^k\}$, $i \in \{1,\ldots,n\}$, the forms of the assignment functions are independent of i, i.e.,*

$$V_i^j = f^j(PA(V_i^j), \epsilon_i^j), \;\; j = 1,\ldots,k.$$

Note that Assumption 3 implicitly requires that the size of all parent sets is bounded:

$$|PA(V_i^j)| \leq M \;\text{ for all } j \in \{1,\ldots,k\} \text{ for some } M \geq 0.$$

This is because unbounded parent sets, e.g., parent sets whose cardinality is increasing with i, would violate the finite number k of different assignment functions. In fact, since the parent sets could depend on all of

observed history, it could be growing with n, increasing the complexity of the assignment functions even as $n \to \infty$. Hence the need for the "bounded" part of Assumption 3.

Assumption 3 ensures that, as long as sufficient number of customer visits n is observed in the data, the number of functions to be estimated, k, is much smaller then n, thus making statistical estimation of these functions tenable. As in any data-driven method, in order to learn the assignment function from data, we need to to ensure that sufficient number of replications of each f^j exist within the event log.

While Assumption 3 may appear restrictive, it allows for a great deal of flexibility since visit components may include time series terms, and/or have parent sets depending on many previous events. Assumption 3 can also accommodate time-varying queues. For instance, although the assignment function of inter-arrival times remains invariant across customers, the arrival process may become non-stationary when parent sets include past event logs.

Note that under Assumption 3, estimating the assignment functions from observed data reduces to a supervised machine learning problem. Specifically, let $\tilde{\mathcal{F}}$ denote a class of conditional generative models. For each $j \in \{1, \ldots, k\}$, we estimate

$$\hat{f}^j \in \arg\max_{f_j \in \tilde{\mathcal{F}}} \frac{1}{n} \sum_{i=1}^{n} \log p_{f_j} \left(v_i^j \mid \overline{PA}(V_i^j) \right), \tag{28}$$

where $\overline{PA}(V_i^j)$ denotes the realizations of the parent set $PA(V_i^j)$, and $p_{f_j}(v \mid \overline{PA}(v))$ is the conditional density induced by the model f_j.

There are various approaches to modeling the conditional distribution $p_{f_j}(v \mid \overline{PA}(v))$, depending on the structure of the parent set $PA(V_i^j)$. Our general approach is to employ a *normalizing flow* (NF) framework (Papamakarios et al. 2021), which transforms a simple base distribution (typically standard Gaussian) into the target distribution through a sequence of invertible, differentiable transformations. Our implementation uses a masked autoregressive flow (MAF) architecture with 20 transformation layers, each implemented as a multi-layer perceptron (64 hidden units, ReLU activations) that conditions on the parent set $\overline{PA}(v)$. This design enables efficient density estimation and sampling while capturing complex dependencies.

There are two cases where we find simpler alternatives to the NF approach to be more effective. The first occurs when the parent set of a certain variable is either empty (e.g., variable T_i in Example 1) or involves only a small number of distinct values (e.g., variable S_i in the same example). As shown in Appendix F, in these cases Empirical Distribution Function (EDF) is a simpler and more accurate alternative to the NF. The second case occurs when the underlying relationship is deterministic (e.g., the Lindley recursion (12) for W_i). In this case the "pure" NF estimator, which assumes absolutely continuous density, fails to converge. While convergence can be achieved by adding random noise to the data and slowing the learning rate, we find that a fully-connected multi-layer perceptron (MLP) provides more accurate and sample efficient identification in this case - see Appendix C.

We thus suggest the following general estimation procedure:

(1) Use EDF for variables whose parent sets are empty or involve only a few (e.g. 2) possible values. For certain other variables the form of the assignment function may be deterministic and known a priori (e.g., time variables $t(V_i^k)$ given by (6, 7)), in which case no estimation is necessary.

(2) For all other variables fit MLP and evaluate cross-validation error. If the fit is nearly perfect (we use threshold $\delta = .01$) assume the relationship is deterministic and use MLP to represent the assignment function.

(3) For all remaining cases, fit NF. If the model fails to converge (because the underlying density is not continuous), adjust the procedure as described in Appendix C and/or revert to using the MLP from step (2) as an estimator.

We demonstrate this process using our running example:

EXAMPLE 4 (LEARNING STRUCTURAL ASSIGNMENTS FOR THE M/M/1 WITH SPEED-UPS). Given $O_i = \{T_i, W_i, S_i, A_i\}$, we assume that the parent sets in (13–15) are known and given in Example 1, but the functional forms (9–11) are unknown and must be estimated from the data.

Since A_i is the intervention variable, its assignment function f_A is given by (25) and does not require estimation. The variable T_i has an empty parent set, while S_i depends only on the binary variable A_i. This variables thus belong to case (1) above. We set the estimated assignment function $\hat{f}_T$ to be the inverse EDF of the inter-arrival times, and let $\hat{f}_{S|A=0}$ and $\hat{f}_{S|A=1}$ be the inverse EDFs of the service times for subgroups with $a_i = 0$ and $a_i = 1$, respectively.

For W_i, the parent set includes continuous variables W_{i-1}, S_{i-1}, and T_i. Thus we are in case (2). We use MLP to estimate $\hat{f}_W$ and find cross-validation error to be well under our threshold of $\delta = .01$ - see Appendix C for details. This correctly identifies the underlying relationship to be deterministic - it is, in fact, the Lindley recursion. In this simple example no variables require the NF estimator. However, this estimator is needed for more complex systems - see Appendix F.

To summarize, we obtain the estimated SCQM consisting of (25) and the following learned assignments:

$$\begin{aligned} T_i &= \hat{f}_T(\epsilon_{T_i}), && i \in \{1, \dots, n\}, \\ S_i &= (1 - A_i)\hat{f}_{S|A=0}(\epsilon_{S_i}) + A_i \hat{f}_{S|A=1}(\epsilon_{S_i}), && i \in \{1, \dots, n\}, \\ W_i &= \hat{f}_W(W_{i-1}, S_{i-1}, T_i), && i \in \{1, \dots, n\}. \end{aligned}$$

With these estimated assignment functions, we can replace the structural equations (9)–(12) with their learned counterparts, enabling us to perform Monte Carlo evaluation and counterfactual intervention evaluation as described in the preceding sections.

REMARK 2. In Remark 1 we discussed a necessary condition for the the SCQM-based intervention analysis to be applicable. The previous example reveals another condition: the observed event log should

provide sufficient coverage of the support of the intervened variable, which is a standard positivity assumption in causal inference. In the previous example this implies that we should observe a sufficient number of instances with actions $a_i = 1$ and $a_i = 0$. For example, if speed up was never observed (i.e., $a_i = 0$ for all cases), then $\hat{f}_{S|A=1}$ cannot be estimated. The same consideration applies to the QT approach.

6. Learning SCQM Causal Structure

In this section we complete the description of the Data Science approach by discussing how the causal graph $\mathcal{G}$ of an SCQM can be learned from the event log data. While a number of approaches for causal learning have been proposed, they all require additional assumptions, since, in general, the structure of $\mathcal{G}$ is not identifiable, see (Peters et al. 2017, Proposition 4.1). Fortunately, when SCQM Definition 2 along with the Bounded Stationary Form Assumption 3 hold, the default structure of the causal graph is automatically specified, ensuring that identifiability always holds, as shown in Section 6.1. However, this default structure is likely to contain many redundant elements in the parent sets, leading to loss of estimation accuracy, as well as obscuring the managerial insights. Thus, Section 6.2 describes the refinement step to uncover the "actual" (i.e., without redundancies) pattern of causal impacts thus reducing the complexity of $\mathcal{G}$. We note that this refinement is likely to be useful for many standard queueing systems, because of their relatively simple causal structure.

6.1. Default Parent Sets in Structural Causal Queueing Model (SCQM)

Consider an SCQM model and some variable $V \in \bar{O}_d$. As observed in (8), the parent set of V must belong to the time-limited event log: $PA(V) \subset \bar{O}(t) = \{U \in \bar{O}_d | t(U) \prec t(V)\}$ for $t = t(V)$ (i.e., the set of all variables observed prior to V). Note that the conditional probabilities in (5) do not change when some irrelevant information is added to the parent set. Thus, if $\bar{O}(t)$ contains some variables that do not change the value of the assignment function $V = f(PA(V), \epsilon_V)$ in Definition 1, their inclusion cannot affect any observational or interventional distribution. This leads to the following observation.

PROPOSITION 3. *Suppose Definition 2 holds. Then, for any variable $V \in \bar{O}_d$, we can assume that $PA(V) = \bar{O}(t(V))$.*

This result immediately yields the *default parent sets* for the SCQM. However, the time-limited event log, defined above, may be somewhat difficult to work with: in order to define the assignment functions that satisfy the stationary form Assumption 3 above, we need to know exactly which variables are included in the parent set for each variable. Moreover, including every possible combination of variables that may appear in a time-limited event log leads to the number of assignment functions growing with the size of the event log - violating Assumption 3. Thus, we need a more transparent structure of the default parent sets. To this end we start with the following definition.

Definition 5 (The Index Variables) *For $t > 0$ define the "index variables" $IT(t), IW(t), IS(t)$ as the number of inter-arrival times, waiting times, and service times observed by time t, respectively:*

$$IT(t) = \sum_{j=1}^{n} \mathbb{1}\left(t(T_j) \le t\right), \quad IW(t) = \sum_{j=1}^{n} \mathbb{1}\left(t(W_j) \le t\right), \quad IS(t) = \sum_{j=1}^{n} \mathbb{1}\left(t(S_j) \le t\right),$$

where $\mathbb{1}(\cdot)$ denotes binary indicator variable.

The index variables can be updated recursively with every new observation, as shown in the following proposition (recall that $\mathcal{C}(X)$ refers to co-observational set of X).

PROPOSITION 4. *Suppose Definition 2 holds. For some $t > 0$, let $IT(t)$, $IW(t)$, $IS(t)$ be the current values of the index variable. Let $t^+ > t$ be the first time period after t at which some variable V is observed (where $t(V) = t^+$). Since $V \in \mathcal{C}(T_i) \cup \mathcal{C}(W_i) \cup \mathcal{C}(S_i)$ for some i, we have:*

$$IT(t^+) = \begin{cases} IT(t)+1 & \text{if } V \in \mathcal{C}(T_i) \\ IT(t) & \text{otherwise} \end{cases} \quad IW(t^+) = \begin{cases} IW(t)+1 & \text{if } V \in \mathcal{C}(W_i) \\ IW(t) & \text{otherwise} \end{cases} \quad IS(t^+) = \begin{cases} IS(t)+1 & \text{if } V \in \mathcal{C}(S_i) \\ IS(t) & \text{otherwise} \end{cases} \tag{29}$$

Proposition 4 is based on observing that, by part (5.) of Definition 2, every variable $V \in O_i$ for some i, and therefore must be co-observed with either T_i, W_i or S_i. Thus, depending on which variable we observe, we simply increment the corresponding index variable by 1. We can regard the updating equations above as the deterministic (i.e., noise-free) assignment functions, and thus assume that the index variables are automatically included in the event log.

For a given $t > 0$, the index variables tell us exactly *how many* inter-arrival, waiting and service times are included in $\bar{O}(t)$, however we do not necessarily know *which* variables are included. This is because, by Definition 2, the index i of the customer visit record O_i refers to the order of arrivals. Thus, if $IT(t) = i$ we know that arrivals $T_1, \ldots, T_i$ have been observed (implying that $IT(t(T_i)) = i$). However, the ordering of other variables may be different. For example $IS(t) = i$ does not necessarily imply that variables $S_1, \ldots, S_k$ have been observed, since a later-arriving customer may overtake earlier arrivals during service and thus $IS(t(S_i))$ may not equal i. To address this uncertainty we define the observational order with respect to a specified variable below.

Definition 6 (Observational Order) *Consider a variable V. We use the notation $V_{[i]}$ to designate the i-th observed value of this V (e.g., $S_{[5]}$ is the 5-th observed service time). Similarly $[i](V)$ refers to the i-th customer when ordering is with respect to observed values of V, rather than arrivals (e.g., $[5](S) = 10$ implies that the 5-th customer to complete service was the 10-th arrival in the system). Note that for any $i = 1, 2, \ldots$ we have $T_{[i]} = T_i$, and $IW(t(W_{[i]})) = IS(t(S_{[i]})) = i$.*

We note if the data log was generated by the First Come First Serve (FCFS) queue, overtakes can only occur during service, and thus $[i](W) = i$. If, in addition, the queue is First In First Out (FIFO), e.g., a FCFS single server queue, then $[i](S) = i$ as well. Both conditions are easy to check from the observed event log.

In view of definitions 5 and 6, we represent time-limited event log as follows:

$$\bar{O}(t) = \bigcup_{j=1}^{IT(t)} \mathcal{C}(T_j) \cup \bigcup_{j=1}^{IW(t)} \mathcal{C}(W_{[j]}) \cup \bigcup_{j=1}^{IS(t)} \mathcal{C}(S_{[j]}). \tag{30}$$

This form is a much more convenient representation of the default parent set $\bar{O}(t(V))$ from Proposition 3 since it specifies exactly which variables are included and since the index variables are easily updated using (29) for every new entry in the observed event log $\bar{o}_d$. We illustrate the previous definition with a small example.

EXAMPLE 5. Suppose that at some time t the observed event log is given by $\bar{o}(t) = \{t_1, w_1, t_2, t_3, t_4, t_5, s_1, w_2, s_2, w_3\}$, where $t = t(w_3)$ (note that by Definition 2, we must have $t(w_3) = t(t_3) + w_3 = t_1 + t_2 + t_3 + w_3$, i.e., the observation times are recoverable from the observed event log). By time $t(w_3)$ we have observed 3 waiting times, 2 service completions and 5 arrivals, implying $IT(t) = 5, IW(t) = 3, IS(t) = 2$.

Unfortunately, the time-limited event log grows with the size of the data. As discussed earlier, this implicitly violates Assumption 3. We thus make an additional assumption that ensures that only finite history is required for each assignment function.

ASSUMPTION 4 **(History-Finite Relevant Information)**. *For customer index $i \in \{1, \ldots, n\}$, $t \in \{t(T_i), t(W_i), t(S_i)\}$ and an integer constant $K > 0$, let the* history-limited event log *be*

$$\bar{O}_i^K(t) = O_i(t) \cup \begin{cases} \bigcup_{j=1}^{K} \mathcal{C}(T_{IT(t)-j}) \cup \bigcup_{j=0}^{K-1} \mathcal{C}(W_{[IW(t)-j]}) \cup \bigcup_{j=0}^{K-1} \mathcal{C}(S_{[IS(t)-j]}) & \text{if } t = t(T_i) \\ \bigcup_{j=0}^{K-1} \mathcal{C}(T_{IT(t)-j}) \cup \bigcup_{j=1}^{K} \mathcal{C}(W_{[IW(t)-j]}) \cup \bigcup_{j=0}^{K-1} \mathcal{C}(S_{[IS(t)-j]}) & \text{if } t = t(W_i) \\ \bigcup_{j=0}^{K-1} \mathcal{C}(T_{IT(t)-j}) \cup \bigcup_{j=0}^{K-1} \mathcal{C}(W_{[IW(t)-j]}) \cup \bigcup_{j=1}^{K} \mathcal{C}(S_{[IS(t)-j]}) & \text{if } t = t(S_i) \end{cases}, \tag{31}$$

with $O_i(t)$ representing previously observed components of customer visit record, defined by:

$$O_i(t) := \begin{cases} \emptyset & \text{if } t = t(T_i) \\ \mathcal{C}(T_i) & \text{if } t = t(W_i) \\ \mathcal{C}(T_i) \cup \mathcal{C}(W_i) & \text{if } t = t(S_i) \end{cases}. \tag{32}$$

An SCQM satisfies the finite relevant information assumption *if there exists $K > 0$ such that any $V \in O_i, i \in \{1, \ldots, n\}$ with $t = t(V)$, is conditionally independent of $\bar{O}(t)/\bar{O}_i^K(t)$ given $\bar{O}_i^K(t)$, where "/" represents the set difference.*

As noted earlier, the index variables, as well as the timing variables $t(V)$ that are contained in $\bar{O}_i^K(t)$ are recursively updated via deterministic relationships based on the last observed value of the respective variable, and thus, recalling that the cardinality of each customer visit record $|O_i| = k$, the size of the history-limited event log is bounded by:

$$|\bar{O}_i^K(t)| \leq (K+1)(|\mathcal{C}(T_i)| + |\mathcal{C}(W_i)| + |\mathcal{C}(S_i)|) = (K+1)|O_i| = (K+1)k. \tag{33}$$

This observation, together with Proposition 3 immediately lead to the following corollary:

Corollary 1 *Suppose an SCQM satisfies Assumption 4. Then, for any $i \in \{1,\dots,n\}$ and $V_i \in O_i$ the finite default parent set is given by $PA(V_i) = \bar{O}_i^K(t)$, with $t = t(V_i)$.*

EXAMPLE 6. Suppose Definition 2 and Assumption 4 hold for our M/M/1 example with $K = 5$, i.e., we know that the observed event log data was generated by SCQM system and that the finite history assumption with history depth of at most 5 holds. Recall that each customer visit record $O_i = \{T_i, W_i, A_i, S_i\}$. As before, we assume that the action $A_i \in \mathcal{C}(W_i)$ i.e., it is co-observed with W_i. (i.e., the speed-up action is applied as customer enters service).

By (31), the default parent set for the inter-arrival time T_i is:

$$PA(T_i) = \bigcup_{j=1}^{5}\{T_{i-j}\} \cup \bigcup_{j=0}^{4}\{W_{[IW(t)-j]}, A_{[IW(t)-j]}\} \cup \bigcup_{j=0}^{4}\{S_{[IS(t)-j]}\}, \tag{34}$$

where all terms with non-positive indices are ignored. In this expression, the first term contains the last $K = 5$ observed inter-arrival times (which must be the inter-arrival items for customers $i-1,\dots,i-5$), the second term are the last 5 observed waiting times and co-observed actions (ordered by the observed waiting times), and the third term contains the 5 last observed service times. Note that the timings of the previous $K = 5$ events (for each of the arrival, service entry, and service completion) can be derived from this parent set, and thus are implicitly included in the parent set as well. Similarly, for the other variables in O_i we have:

$$PA(W_i) = PA(A_i) = \bigcup_{j=0}^{4}\{T_{IT(t)-j}\} \cup \bigcup_{j=1}^{5}\{W_{[IW(t)-j]}, A_{[IW(t)-j]}\} \cup \bigcup_{j=0}^{4}\{S_{[IS(t)-j]}\} \cup \{T_i\}, \tag{35}$$

$$PA(S_i) = \bigcup_{j=0}^{4}\{T_{IT(t)-j}\} \cup \bigcup_{j=0}^{4}\{W_{[IW(t)-j]}, A_{[IW(t)-j]}\} \cup \bigcup_{j=1}^{5}\{S_{[IS(t)-j]}\} \cup \{T_i, W_i, A_i\}. \tag{36}$$

We note that additional information about the underlying data-generation process is easy to incorporate into the default parent sets above. For example, if data came from a FCFS queue, then $[\cdot]$ can be dropped from the subscripts of all terms containing W_i and A_i variables, since their ordering will coincide with that of arrivals.

While Assumption 4 may appear restrictive, the next result shows that it holds for a wide range of queueing models, including the (FCFS) $G/G/m$ queue with possibly time-dependent arrival and service processes.

PROPOSITION 5. *Consider an SCQM with event-log data generated by a $G/G/m$ queue (general arrival and service distributions with m parallel servers and a FCFS queue) where arrival and service processes are autoregressive of order $r \le m$. Then, Assumption 4 holds with $K = m$. Moreover, the assignment function for the waiting time is deterministically given by*

$$W_i = \max\{t(T_i), t(S_{[i-m]})\} - t(T_i), \tag{37}$$

where $t(S_{[j]}) = t(T_j) = 0$ for $j \le 0$.

Proposition 5 holds since for inter-arrival and service variables (and their co-observational sets) Assumption 4 holds by hypothesis. Equation (37) holds by Krivulin's recursion (Krivulin 1995), and equation (7), implying that for $V \in \mathcal{C}(W_i)$ we have $PA(V) = \{T_i, S_{[i-m]}\}$, satisfying Assumption 4 with $K = m$. We further emphasize that being able to specify the features for Equation (37) requires a QT.

The intuition behind Krivulin's recursion is that every customer i who joins a $G/G/m$ queue faces one of two cases. Either the customer service starts at time $t(T_i)$ with a waiting time $W_i = t(T_i) - t(T_i) = 0$. Or customer i joins a busy system and enters service exactly then $i - m$-th service completion occurs (as then one of the servers frees up). The entry to service is exactly at this time because under FCFS there is no overtaking in the *queue*, i.e., during the waiting process, then $[i](W) = i$. Hence, in this case, $W_i = t(S_{[i-m]}) - t(T_i)$. We note that the assignment function in (37) is deterministic. Thus, while we may not know a priori the number of servers in the underlying data-generating process, assuming this number is constant over time and that an upper bound K is available, it is easy to discover by trying different values of m until (37) holds.

Proposition 5 can be extended to many other queueing-based data-generating systems, including limited-buffer queues, queues with customer balking, as well as queueing networks with stations operating as $G/G/m$ queues networks (in the latter case, equation (37) can be extended as in Krivulin (2012)). For this reason, Assumption 4 is not particularly limiting.

We also note that for the 1-server (i.e., $m = 1$) case, Krivulin's recursion (37) is equivalent to the Lindley recursion (12). Thus, if in Example 6 we assume that we know that the data came from a FIFO queue with $m = 1$ (which also implies FCFS queue), then (35) simplifies to

$$PA(W_i) = PA(A_i) = \{T_i, S_{i-1}, W_{i-1}\} = \{t(T_i), t(S_{i-1})\}.$$

6.2. Refinement of the Parent Sets in SCQM

As discussed earlier, the default parent sets derived in the previous section likely contain many variables that may not affect the assignment functions and can thus be eliminated to create "refined" parent sets. This refinement is equivalent to dimension reduction in the machine learning literature, for which numerous techniques are available. We focus on the techniques developed for identifying Granger causality (Granger 1969), as they are particularly appropriate for queueing settings due to time-ordering.

Consider an l-dimensional time-series process $\mathbf{X}_t = (X_{t1}, \ldots, X_{tl})$ specified by

$$\mathbf{X}_t = g(X_{<t1}, \ldots, X_{<tl}, \epsilon_t) \tag{38}$$

where $X_{<ti} = (\ldots, X_{(t-2)i}, X_{(t-1)i})$ are the past values of X_{it}, $g(\cdot)$ is some vector-valued function representing process dynamics, and the random noise ϵ_t is $U[0, 1]$. Component $X_{(t-r)j}$ is said to be *Granger non-causal* for X_{ti} if $g_i(X_{<t1}, \ldots X_{<tl}, \epsilon_t)$ is invariant with respect to $X_{(t-r)j}$, where $1 \leq r < t$ and $i, j \in \{1, \ldots, l\}$.

Equation (38) is closely related to the default parent sets defined in the previous section. Indeed, since SCQM consists of customer visit records O_i ordered by customer arrivals, it can be viewed as a, stationary, vector-valued time series process. Consider some component of the i-th customer visit $V \in O_i$, with the assignment function

$$V = f(PA^d(V), \epsilon_i), \tag{39}$$

where $PA^d(V)$ is the default parent set defined in the previous section. Note that we do not use subscript i for V or $f(\cdot)$ in view of Assumption 3, but $f(\cdot)$ could be different for various V. For instance, inter-arrival time and service time could have different assignment functions, as f_T, f_S. To simplify the notation from Assumption 3, we drop the superscript j in $V, f(\cdot)$ as well. If $f(\cdot)$ is invariant with respect to some $U \in PA^d(V)$, then $PA^d(V)$ can be "refined" by removing U. We see that refining the default parent set is essentially equivalent to identifying elements of $PA^d(V)$ that are Granger non-causal with respect to V; we call the resulting refined parent set $PA^r(V)$

To identify non-causal variables we adopt the techniques developed by Tank et al. (2021). In our setting, this involves modeling $f(\cdot)$ in (39) as $L-$hidden layer MLP, similar to Section 5. The input layer consists of all elements of $PA^d(V)$. Suppose H is number of neurons in the first hidden layer. Let $\mathbf{\Theta}^1 \in \mathbb{R}^{H \cdot |PA^d(V)|}$ be a vector of weights for arcs connecting the input layer with the first hidden layer. The key observation is that if some component $\mathbf{\Theta}^1_U = 0$, then variable U can be removed as an input without affecting the estimated assignment function $f(\cdot)$. We thus obtain

$$PA^R(V) = PA^d(V) - \{U \in PA^d(V) | \mathbf{\Theta}^1_U = 0\} \tag{40}$$

Following Tank et al. (2021), we use lasso regularization to drive components of $\mathbf{\Theta}^1$ to 0. Specifically, we solve the following optimization problem for each $V \in O_i$:

$$\min_{\mathbf{\Theta}} \left\{ \sum_{i=1}^{n} \left(v_i - \hat{f}^R(\overline{PA}^d(v_i)) \right)^2 + \eta \sum_{U \in PA^d(V)} \Omega\left(\mathbf{\Theta}^1_U\right) \right\}, \tag{41}$$

where v_i and $\overline{PA}^d(v_i)$ are the i-th realization of V and all the variables in $PA^d(V)$, $\hat{f}^R(\cdot)$ is the estimated MLP-based assignment function described above, η is the lasso regularization parameter optimized during model tuning, and $\Omega(\mathbf{\Theta}^1_U)$ are the penalty terms applied to components of $\mathbf{\Theta}^1_U$; the structure of $\Omega(\cdot)$ is described below.

A few comments regarding (41) are in order. First, note that $\hat{f}^R$ is an estimate of the assignment function in (39), which can be viewed as an alternative to $\hat{f}$ in (28) earlier. We use subscript R to distinguish these two estimates. There is a subtle, but crucial difference between them: while (28) is a *supervised learning* problem, where clearly defined target v_i is available, the presence of $\Omega(\cdot)$ term in (41) makes it an *unsupervised learning* problem, as the true value of this term is not known. In particular, if the regularization weight

η is set to sufficiently high value, the optimal solution is $\mathbf{\Theta}^1_U = 0$ and $PA^R(V) = \emptyset$. Thus we seek some reasonable balance between the accuracy of $\hat{f}^R$ estimator and the second term. Our approach to this problem is described below. While using $\hat{f}^R$ is possible, we find that using $\hat{f}^R$ only in the parent set refinement step described in the current section, and subsequently applying (28) to the refined parent sets to obtain $\hat{f}$ as described in Section 5 above results in more accurate estimates. Second, observe that the regularization penalty term in (41) is applied only to the first-layer coefficients (because the primary goal in this step is to eliminate as many unnecessary components from $PA(V)$ as possible, rather than estimate the most accurate model).

To complete the specification of (41) we need to specify the activation function for the MLP (we use ReLU, as suggested by Tank et al. (2021)), the penalty term, and the strategy for selecting η. The simplest suggested form for the penalty term is

$$\Omega\left(\mathbf{\Theta}^1_U\right) = \| \mathbf{\Theta}^1_U \|_F, \tag{42}$$

where $\| \cdot \|_F$ is the Frobenius norm.

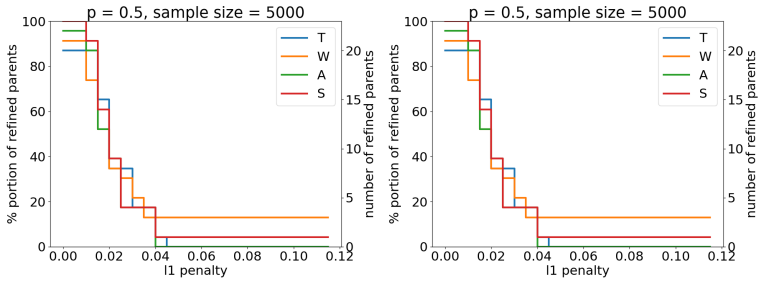
A suggested alternative, which we adopt, is to penalize all observations of a given variable that are included in $PA^d(V)$, with older observations receiving higher penalty weights. This results in eliminating older values (more distant lags) before more current values and in eliminating all observed values of a given variable - leading to more intuitive refined parent sets. Specifically, for $U \in PA^d(V_i)$ we denote by $U_< = \{U_j | U_j \in PA^d(V_i)\}$ the set of previously observed values of U in the default parent set. By Assumption 4, $|U_<| \in \{K, K+1\}$ depending on whether $U \in \mathcal{C}(T_i) \cup \mathcal{C}(W_i)$ or $U \in \mathcal{C}(S_i)$. E.g., in Example 6, with $V_i = W_i$ and $t = t(W_i)$, the set $T_< = \{T_{IT(t)}, T_{IT(t)-1}, \ldots, T_{IT(t)-4}, T_i\}$ with $|T_<| = 6$. We order the elements of $U_<$ by the observation order and set

$$\Omega\left(\mathbf{\Theta}^1_{U_<}\right) = \sum_{r=1}^{|U_<|} \| \mathbf{\Theta}^{1,r}_U, \ldots, \mathbf{\Theta}^{1,|U_<|}_U \|_F, \tag{43}$$

where $\mathbf{\Theta}^{1r}_U \in \mathbb{R}^H$ represent the layer 1 weight of the r-th observed value of U. Expression (43) applies group penalty to U and all of its lags, with older lags receiving higher penalties.

Perhaps the most important component of solving (41) is the selection of η. As observed earlier, by setting η high enough the optimum is achieved at $\mathbf{\Theta}^1 = 0$, removing all variables from $PA(V)$. Conversely, by setting $\eta = 0$, the penalty term is removed and no refinement is likely to take place. Neither is a useful solution.

We thus test a range of values by increasing η from zero until the value where $PA(V) = \emptyset$. Our expectation is that when the refined parent set matches with the "ground truth" (i.e., a given component of $PA^d(V)$ does not carry much information about V), the number of zero or near-zero components of $\Theta^1_U, U \in PA^d(V)$ will remain stable over a large range of η. We call this region of stability a *plateau* and the corresponding range of η the *length of plateau*. We thus select $PA^R(V)$ corresponding to the longest



(a) M/M/1 with Speed-ups **(b)** GI/M/2 with Speed-ups

Figure 2 **Proportion of refined parent set with respect to η. Left figure shows M/M/1 with speed-ups under $p = 0.5, n = 5000$. Right figure shows G/M/2 with speed-ups under $p = 0.5, n = 5000$. Vertical axes denote the proportion (left) and the number (right) of variables in the refined parent set.**

plateau. When no well-defined plateau emerges (i.e., $PA^R(V) = \emptyset$ is reached over a small range of η), we set $PA^R(V) = \emptyset$. We illustrate this approach in Example 7.

EXAMPLE 7 (M/M/1 WITH SPEED-UPS AND G/M/2 WITH SPEED-UPS: PARENT SET REFINEMENT). We illustrate the methodology described above for our running example. We used $\lambda = 1, p = 0.5, \mu_0 = 1.1, \mu_1 = 2$ to generate an event log with $n = 5000$ observations of O_i from M/M/1 with speed-ups.

To generate data-driven refined parent sets, we first assume that Assumption 4 holds and thus the default parent sets are given in Example 6 above with history length of $K = 5$.

For each variable T_i, W_i, A_i, S_i, We solved (41) with the penalty term (43). We used MLP with one hidden layer with 100 neurons. Each problem was re-solved with different values of η according to our plateau strategy described above. The results are presented in Figure 2a. The horizontal axis shows values of $\eta \in \{0, 0.005, 0.01, \cdots, 0.115\}$.

We see that for W (orange curve) the size of the parent set stabilizes at $|PA^R(W)| = 2$ (representing 9.52% of the default parent set size of 21). Similarly, for S the plateau is reached at $|PA^R(S)| = 1$ (i.e., $1/23 = 4\%$ of the default set size). For T and A no well-defined plateau emerges, and we conclude that $|PA^R(T)| = |PA^R(A)| = 0$. In all cases, the plateau is reached at, or before, $\eta = .05$ and persists until at least $\eta = 0.12$.

Picking an arbitrary value of η within the plateau region, e.g. $\eta = .06$ and examining the components of $\boldsymbol{\Theta}^1$ in (41) that have been driven to 0, we observe that

$$PA^R(W_i) = \{W_{i-1}, S_{i-1}, T_i\}, PA^R(S_i) = \{A_i\}, PA^R(A_i) = PA^R(T_i) = \emptyset,$$

and thus our procedure recovered the exact structure of the "ground truth" parent sets (see Example 1) in our data with only 5000 observations. The result is robust to different random seeds, as we consistently recover the ground truth within 25 independent replications.

Similarly, we run the experiments for GI/M/2 with speed-ups, where there are two servers and inter-arrival time distribution is a mix between exponential and uniform (see more details from Section 7.2). The

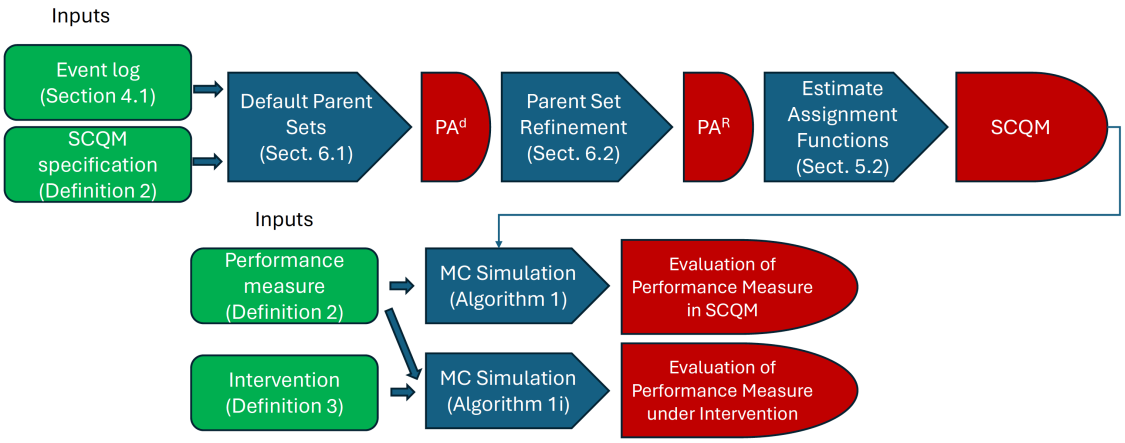


Figure 3 **Summary of SCQM appproach. Top panel: step (I), model specification and estimation. Bottom panel: step (II), model use for performance evaluation. Green rectangles represent inputs, half-ovals are outputs, and arrow-shaped boxes are applications of specified algorithms.**

results are shown from Figure 2b. We see that for W (orange curve) the size of the parent set stabilizes at $|PA(W)| = 2$ and $PA^R(W_i) = \{t(S_{[i-m]}), t(T_i)\}$ (representing 9.52% of the default set size of 21). For T, A, S, the results are similar to what we have in M/M/1 with speed-ups. In all cases, the plateau is reached at, or before, $\eta = .05$ and persists until at least $\eta = 0.12$. Under $K = 5$, we can successfully recover the ground truth causal parents of T_i, W_i, A_i, S_i.

The refined parent sets can now be passed to the procedure in Section 5 to estimate the assignment functions and then simulate the system performance under various interventions (i.e., different speed-up policies). The detailed results of these numerical experiments are described in Section 7 below.

To close this section, we note that Assumption 4 requires us to specify the exact value of K - the maximum history length parameter in SCQM. In practice this may not be easy; it is best to view K as the model hyper-parameter whose value is tuned during the training stage. We suggest starting with some initial value K. If, during the refinement stage, we find all lags of size K are eliminated, this was a good assumption, If not, we can increase K and repeat the process.

6.3. SCQM Estimation and Performance Evaluation: Summary

In the preceding sections we described the key components of our data-driven approach to SCQM. We summarize the main components and the logic flow in the current section. The summary is displayed on Figure 3. There are two main steps: (I) model specification and estimation, illustrated on the top panel, and (II) applying the estimated model to evaluate the specified performance measure with and without the specified intervention, shown on the bottom panel. The key stages of each step are indicated. We also provide references to the Section where each stage is described.

It is important to note that the available event-log data is used only for step (I). Once the components (i.e., parent sets and assignment functions) are estimated and passed to step (II), only Monte Carlo simulation is used to generate the synthetic data for evaluation of system performance under the specified intervention.

Thus, while step (I) is limited by the available data, step (II) is not: simulations can be run until the desired convergence criteria are met.

We also note that step (I) requires a number of decisions, including how the underlying ML models are structured, how the available data is subdivided into training and validation parts, etc. We address these issues in the following section where we discuss several numerical experiments.

7. Numerical Results: QT vs SCQM in Different Settings

In this Section we evaluate the empirical performance of SCQM-based DS approach through simulation studies under a range of settings that are increasingly more challenging for the QT: starting with the analytically tractable model of M/M/1 queue with "Bernoulli" speed-ups, progressing to a G/M/2 model with speed-ups for which approximation methods are available, though no exact analytical solutions exist, and finally presenting an M/M/1 queue with state-dependent speed-ups for which we are not aware of any QT analytical or approximation methods in the literature. In all three cases, SCQM effortlessly adjusts to new settings and delivers high-quality estimates.

7.1. M/M/1 with Speed-ups

We begin by applying our methodology to the running example of M/M/1 with speed-ups, comparing the numerical performance of QT and DS solutions. We generate event log data with parameters $\lambda = 1$, $\mu_0 = 1.1$, $\mu_1 = 2$, and varying speed-up probabilities:

$$p \in \mathcal{P} = \{0.01, 0.1, 0.5, 0.9, 0.99\}.$$

For each p, we simulate event logs of sizes $n_1 = 5000$ and $n_2 = 50000$ customer visits. For each pair $p, p^* \in \mathcal{P}$, we treat p as the factual setting (used for model estimation) and p^* as the counterfactual intervention (used for model testing). Our main performance metric is the expected waiting time in queue.

For the QT approach, the estimated waiting time W_{QT} is computed using Equation (4). For the DS approach, we consider two estimators. The *"full" DS estimator* (*DSF*) does not assume any knowledge of the underlying system and follows the methodology described on Figure 3. For comparison, we also use the parametric DS estimator (*DSP*), which assumes queueing knowledge that includes the causal graph and the form of the assignment functions (i.e., that inter-arrival and service times are exponential), employing data-driven estimates of system parameters - see Appendix B for details.

We assess estimator accuracy via relative error against the ground truth W_{GT} (computed via (2), using true parameters). For any $\hat{W} \in \{W_{QT}, W_{DSP}, W_{DSF}\}$, we define:

$$\text{Rel. Err.}(\hat{W}) = \frac{|\hat{W} - W_{GT}|}{W_{GT}}. \tag{44}$$

For each $(p, p^*) \in \mathcal{P} \times \mathcal{P}$ and $n \in \{5000, 50000\}$, we generate 25 replications and report average relative errors. Results are summarized in Table 1, with more complete results provided in Appendix D.

Table 1 Relative absolute error in percentage among QT, DSP and DSF in M/M/1 with speed-ups, $n \in \{5000, 50000\}$ and $p, p^* \in \{0.01, 0.5, 0.99\}$.

n	5000									50000								
p	0.01			0.5			0.99			0.01			0.5			0.99		
p^*	0.01	0.5	0.99	0.01	0.5	0.99	0.01	0.5	0.99	0.01	0.5	0.99	0.01	0.5	0.99	0.01	0.5	0.99
QT	16.0	15.0	33.5	23.9	6.6	6.9	132.6	32.0	3.6	5.9	4.3	10.3	5.2	1.6	1.7	35.8	12.8	1.2
DSP	15.3	15.0	33.4	23.0	6.8	7.1	79.6	31.6	3.7	5.8	4.6	10.2	5.2	1.6	1.6	35.3	12.9	1.2
DSF	15.2	15.0	35.5	22.7	7.1	7.9	82.2	33.7	4.0	6.1	4.3	10.9	5.5	1.8	2.0	35.6	12.7	1.3

There are several important takeaways:

(1) Despite the availability of a closed-form QT solution, SCQM-based methods (especially DSF) achieve similar accuracy. The parametric DSP estimator provides no substantial improvement over DSF. All three methods require estimation of system parameters (explicit or implicit), and these estimation errors dominate simulation variance. Increasing n reduces errors across all methods, underscoring the importance of having sufficient training data for accuracy of data-driven estimators.

(2) Error magnitudes grow when the factual and counterfactual policies differ significantly. Note that large values of the policy parameter (p and p^*) imply large number of speed-ups, resulting in low system utilization, and vice versa. Largest errors occur when the system utilization for training data is very different from the utilization under the counterfactual estimate, i.e., the difference $|p - p^*|$ is large. In such cases even $n = 50000$ observations in the training data may not assure high accuracy.

(3) QT has a notable advantage in computational efficiency. Its estimates are near-instantaneous, while DSF and DSP require simulation (and training for DSF). For $p^* = 0.01$, DSF runtimes ranged from 100 to 1000 seconds, DSP from 1 to 6 seconds (on a 4GB CPU laptop). While QT's accuracy may not exceed DS, its efficiency remains a key strength when analytical solutions are available.

7.2. GI/M/2 with Speed-ups

We now extend the analysis to a GI/M/2 queue with speed-ups, introducing two servers and non-exponential arrivals. This setting is analytically intractable. Thus the QT methods require approximaitons or simulation. In contrast, the SCQM approach remains applicable without any additional structural assumptions or adjustments to the basic methodology.

We model inter-arrival times as a mixture of exponential and uniform distributions with a mixture weight $\pi \in [0, 1]$. Service rates are split evenly across two servers: $\mu_1 = \mu_2 = \mu/2$. As no exact QT solution exists for GI/M/2 with speed-ups, we compare six approximations: BS1–BS3 from Buzacott and Shanthikumar (1993), and W83, W89, and W93 from Whitt (1983, 1989, 1993). Details of these methods can be found in Appendix E.1. The DSF procedure is unchanged from the previous section and follows the methodology on Figure 3. The refinement step successfully recovers the system's causal structure as in the M/M/1 case; see Appendix E.2 for details.

We compare the relative errors for QT approximations and DSF across different π and n values (higher π represents larger departure from the Poisson arrivals); we also used different (p, p^*) combinations as in

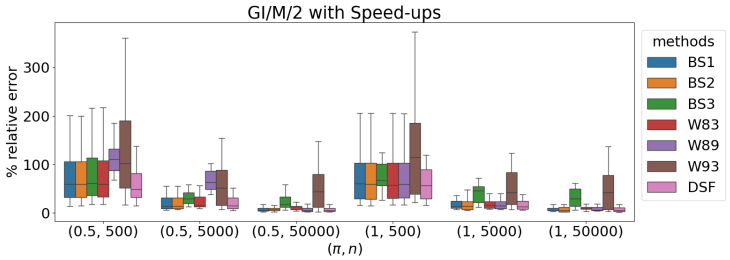


Figure 4 Boxplot of relative absolute errors (percentage) for QT approximations and DSF in GI/M/2 with speed-ups, $\pi \in \{0.5, 1\}, n \in \{500, 5000, 50000\}$. Outliers suppressed.

the previous section. Ground truth is computed via simulation (2 million customers). Summarized results are shown in Figure 4; detailed results are in Table 5 in Appednix E.2.

We find that DSF consistently delivers top-tier accuracy across all experimental conditions (varying p, p^*, π, and n). BS1, BS2, and W83 are competitive with DSF; BS3, W89, and W93 are not. Larger samples and smaller $|p - p^*|$ gaps improve accuracy, as in the M/M/1 case. The effect of π is modest.

To summarize, this section demonstrates the versatility of the DSF approach. Unlike QT-based techniques that had to be completely changed from the setting of Section 7, the DSF approach was identical - automatically adapting to the new data setting, and achieving high level of accuracy across all experimental conditions.

7.3. M/M/1 With State-Dependent Speed-ups

Here we consider another variation on our running M/M/1 example by changing the speed-ups from being based on Bernoulli policy (as in Example 1) to a more realistic state-dependent policy. Specifically, we assume that the assignment function (10) is replaced by

$$A_i = \mathbb{1}\left(T_i \leq \delta\right), \tag{45}$$

for a specified $\delta > 0$. Thus, the speed-up action is applied whenever there is a "burst" or arrivals, i.e., the inter-arrival time T_i between customers i and $i-1$ falls below the threshold δ, which defines the policy.

We generate the event log by simulating the system with the same parameters as in Section 7.1, except that the speed up probability p is replaced with $\delta = \{.5, 1, 2\}$ (note that lower value of δ implies less frequent speed-ups, and thus higher system utilization).

We follow the steps on the top panel of Figure 3 to obtain the SCQM estimator DSF (without making any changes to the estimation process). As a result of the parent set refinement procedure described in Section 6.2 we successfully recover all the underlying parent sets, including identification of $T_i \in PA(A_i)$ (see Figure 6 in Appendix E.3).

Next, we "intervene" by replacing δ with $\delta^* \in \{.5, 1, 2\}$ and obtain the counterfactual estimates following the process on the lower panel of Figure 3. We note that this intervention assumes that the decision-maker

n	5000									50000								
δ	0.5			1			2			0.5			1			2		
δ^*	0.5	1	2	0.5	1	2	0.5	1	2	0.5	1	2	0.5	1	2	0.5	1	2
DSP	9.1	5.6	5.3	8.0	5.3	4.0	8.3	4.1	2.9	2.3	2.9	3.4	2.2	2.4	2.3	4.4	2.5	2.1
DSF	8.7	6.7	6.0	8.1	4.6	3.7	8.9	5.3	3.7	2.3	2.8	3.5	2.5	2.5	2.5	4.6	2.6	2.2

Table 2 Relative absolute error in percentage between DSP and DSF in M/M/1 with state-dependent speed-ups, $n \in \{5000, 50000\}$ and $\delta, \delta^* \in \{0.5, 1, 2\}$.

is aware of (45), which seems reasonable: in order to intervene, one should know the direct impact of the policy; we note that an alternative intervention would be on the distribution of T_i, which does not require knowledge of (45).

For comparison, we also obtain the parametric estimator DSP (as in Section 7.1). We generate the "ground truth" by simulating the underlying system. As noted earlier, there is no QT estimate that we could use for comparison. The results are displayed on Table 2. We observe that, similarly to Section 7.1, (1) DSP estimate, that requires full knowledge of the system other than the parameter values, does not provide a significant advantage over the DSF, and (2) higher accuracy requires more training data. Interestingly, in this case the largest errors are associated with estimating the waiting time of a more congested system (i.e., small δ^*), rather then the $|\delta - \delta^*|$ gap. Of course, the main takeaway from this case is that SCQM approach is capable of effortlessly adapting to systems with complex dependencies which are very difficult to represent using the classical QT tools (note that one could have introduced a state-dependent control similar to (45) that depends on the waiting time W_i instead of T_i or on a larger set of previously observed variables. A control based on W_i would resemble classical queue-length-dependent policies, such as granting speed-up service when the queue length exceeds a certain threshold).

8. Conclusions and Future Work

We develop a novel framework based on causal modeling for comparative analysis of intervention policies in queueing systems. While the basic framework is quite general, our approach requires the data generating process to satisfy certain structural assumption resulting in Structural Causal Queueing Model (SCQM). These assumptions are quite natural for many data processes generated by queueing systems. SCQM approach yields a set of machine learning models and a coupled Monte Carlo simulator that can be used to estimate counterfactual effects of various interventions on the underlying system.

We demonstrate that the accuracy of SCQM-based estimates is competitive with those based on queueing theory (QT) even for the simple systems closed-form solution to the underlying queueing model is available. For the case where only approximate solutions are available, SCQM estimates are on par with the most accurate approximations. Of course, the real advantage of the SCQM approach is its ability to tackle systems with complex inter-dependencies for which no closed-form or approximate solutions are available while maintaining high accuracy via a purely data-driven approach.

We do not mean to suggest that SCQM should displace QT. Indeed, we feel that QT is far easier to work with when closed-form solutions are available - these formulas are both insightful, and useful for system optimization, in addition to being computationally efficient. On the other hand, we do recommend SCQM over QT when the exact knowledge of the queueing system is unavailable due to e.g., (1) unknown arrival and service process (unknown distributions, correlations among inter-arrival times and service times), (2) unknown queueing dynamics and (3) no available analytical or solutions or well-developed approximations. In such cases QT's method will be restrictive, while the SCQM approach automatically adapts, providing data-driven solutions for comparative analytics in queues. Our work opens several avenues for future research, including:

Characterizing Identifiable Interventions Perhaps the main open question is identifying which counterfactual intervention policies can be reliably estimated from observational data by SCQM. While necessary conditions are discussed in Remarks 1 and 2, sufficient conditions remain largely unexplored.

Improvement of SCQM Approach Both sample efficiency and computational efficiency of SCQM can likely be improved. For example one could explore integrating parent set refinement and assignment function estimation steps into a unified machine learning model, such as Long-Short-Term-Memory (LSTM) or highly adaptive LASSO by Benkeser and Van Der Laan (2016).

Queueing Structures that Violate the Finite Memory Assumption 4. This assumption is violated in certain queueing systems, particularly those with non-FCFS disciplines such as priority or last-come-first-served (LIFO) queues. In these settings, the last customer to begin service before customer i may arrive after i, introducing dependencies on future events. As a result, SCQM under Assumption 4 serves only as an approximation. To handle such cases, both the regularization structure and the choice of the memory parameter K may require adjustment.

Queuing Networks Most real-world applications involve networks of queues (possibly with re-entrant flows and complex customer-dependent routings), rather than a single station. Applying SCQM to this setting requires adding station-dependent information (arrival, routing, waiting and service times) to the event log of Definition 2. More importantly, it requires learning the assignment function for routing, which may be challenging. On the other hand, SCQM holds the promise of being able to handle complex system interdependencies. We present preliminary results for a tandem queue with dependent service in Appendix F.

SCM for Other Operations Management Applications The success of SCQM highlights the broad potential of applying causal approaches for comparative analytics in operations management. For example, we could treat a simple inventory system as a black box and learn the causal structure from the relevant data. By applying the SCM-based techniques, we can automate the discovery of data-driven simulation for inventory data, enabling the evaluation of counterfactual inventory policies.

References

Asanjarani A, Nazarathy Y, Taylor P (2021) A survey of parameter and state estimation in queues. *Queueing Systems* 97:39–80.

Baron O (2021) Business analytics in service operations—lessons from healthcare operations. *Naval Research Logistics* 68(5):517–533.

Baron O, Krass D, Senderovich A, Sherzer E (2023) Supervised ML for solving the GI/GI/1 queue. *INFORMS J. on Computing* .

Benkeser D, Van Der Laan M (2016) The highly adaptive lasso estimator. *2016 IEEE international conference on data science and advanced analytics (DSAA)*, 689–696 (IEEE).

Buzacott J, Shanthikumar J (1993) *Stochastic Models of Manufacturing Systems*. Prentice-Hall international series in industrial and systems engineering (Prentice Hall), ISBN 9780138475673.

Camargo M, Dumas M, González-Rojas O (2020) Automated discovery of business process simulation models from event logs. *Decision Support Systems* 134:113284.

Carter MW, Blake JT (2004) Using simulation in an acute-care hospital: easier said than done. *Operations research and health care: A handbook of methods and applications* 191–215.

Corlu CG, Akcay A, Xie W (2020) Stochastic simulation under input uncertainty: A review. *Operations Research Perspectives* 7:100162.

Cox DR (1955) The statistical analysis of congestion. *Journal of the Royal Statistical Society. Series A (General)* 118(3):324–335.

Cruz FR, Santos MA, Oliveira F, Quinino R (2021) Estimation in a general bulk-arrival markovian multi-server finite queue. *Operational Research* 21(1):73–89.

Deo S, Jain A (2019) Slow first, fast later: Temporal speed-up in service episodes of finite duration. *Production and operations management* 28(5):1061–1081.

Elalouf A, Wachtel G (2021) Queueing problems in emergency departments: a review of practical approaches and research methodologies. *Operations Research Forum*, volume 3, 2 (Springer).

Granger CW (1969) Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society* 424–438.

Kim SHH (2014) *Data-driven decisions in service systems* (Columbia University).

Kohavi R, Rothleder NJ, Simoudis E (2002) Emerging trends in business analytics. *Communications of the ACM* 45(8):45–48.

Krivulin NK (1995) A max-algebra approach to modeling and simulation of tandem queueing systems. *Mathematical and Computer Modelling* 22(3):25–37.

Krivulin NK (2012) Max-plus algebra models of queueing networks. *arXiv preprint arXiv:1212.0578* .

Kyritsis AI, Deriaz M (2019) A machine learning approach to waiting time prediction in queueing scenarios. *2019 Second International Conference on Artificial Intelligence for Industries (AI4I)*, 17–21 (IEEE).

Lindley DV (1952) The theory of queues with a single server. *Mathematical proceedings of the Cambridge philosophical society*, volume 48, 277–289 (Cambridge University Press).

Nii S, Okudal T, Wakita T (2020) A performance evaluation of queueing systems by machine learning. *2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan)*, 1–2 (IEEE).

Palomo S, Pender J (2020) Learning lindley's recursion. *2020 Winter Simulation Conference*, 644–655 (IEEE).

Palomo S, Pender J (2021) Learning the tandem network lindley recursion. *2021 Winter Simulation Conference*, 1–12 (IEEE).

Papamakarios G, Nalisnick E, Rezende DJ, Mohamed S, Lakshminarayanan B (2021) Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research* 22(1):2617–2680.

Pearl J (2009) *Causality* (Cambridge university press).

Peters J, Janzing D, Schölkopf B (2017) *Elements of causal inference: foundations and learning algorithms* (The MIT Press).

Robins J (1986) A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical modelling* 7(9-12):1393–1512.

Senderovich A, Weidlich M, Yedidsion L, Gal A, Mandelbaum A, Kadish S, Bunnell CA (2016) Conformance checking and performance improvement in scheduled processes: A queueing-network perspective. *Information Systems* 62:185–206.

Sherzer E, Baron O, Krass D, Resheff Y (2025) Approximating g (t)/gi/1 queues with deep learning. *European Journal of Operational Research* 322(3):889–907.

Shortle JF, Thompson JM, Gross D, Harris CM (2018) *Fundamentals of queueing theory*, volume 399 (John Wiley & Sons).

Tank A, Covert I, Foti N, Shojaie A, Fox EB (2021) Neural granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44(8):4267–4279.

Van der Laan MJ, Rose S (2018) *Targeted learning in data science* (Springer).

Van der Laan MJ, Rose S, et al. (2011) *Targeted learning: causal inference for observational and experimental data*, volume 4 (Springer).

Whitt W (1983) The queueing network analyzer. *The bell system technical journal* 62(9):2779–2815.

Whitt W (1989) An interpolation approximation for the mean workload in a gi/g/1 queue. *Operations Research* 37(6):936–952.

Whitt W (1993) Approximations for the gi/g/m queue. *Production and Operations Management* 2(2):114–161.

Whitt W, You W (2022) A robust queueing network analyzer based on indices of dispersion. *Naval Research Logistics* 69(1):36–56.

Appendix. (Online Supplement)

A. Performance Evaluation Algorithms

When the distribution $P^{\mathcal{Q}}$ can be obtained analytically, we can use Proposition 1 to evaluate (18) numerically via Monte Carlo simulation, as indicated in Algorithm 1. The algorithm generates M independent observations of event logs and uses them to obtain M samples of the performance measure ψ. Each event log is generated by following the causal graph $\mathcal{G}$ from root nodes (i.e., variables V_i that have empty parent sets) to leaf nodes. The number of observations M can be adjusted to reach the desired accuracy level. Note that the algorithm does not assume any time-ordering with respect to customer visit records, allowing for very general causal dependencies between various elements, both within each record and among records of different customers.

Algorithm 1 Performance Evaluation Under SCM

1: **for** $m = 1$ to M **do**: ▷ Major iteration
2: Initialize $S = \emptyset, \bar{o}_d = \emptyset$
3: **repeat** ▷ Generate an event log
4: Find $i \in \{1, \ldots, d\}$ such that $i \notin S$ and $PA(V_i) \subset \{V_j, j \in S\}$.
5: Draw an observation ϵ_i from $U[0,1]$ distribution.
6: Compute $v_i = f_i(\overline{PA}(v_i), \epsilon_i)$, where $\overline{PA}(v_i)$ are the realizations of
7: variables in $PA(V_i)$.
8: $S \leftarrow S \cup \{i\},\ \bar{o}_d \leftarrow \bar{o}_d \cup \{v_i\}$
9: **until** $S = \{1, \ldots, d\}$ ▷ Event log generated
10: Compute the realization of the performance measure $\psi_m(\bar{o}_d)$.
11: **end for** ▷ End of major iteration
12: Compute $\hat{\Psi} = \frac{1}{M}\sum_{m=1}^{M} \psi_m(\bar{o}_d)$ to estimate $\mathbb{E}_{\bar{O}_d \sim P^{\mathcal{Q}}}(\psi(\bar{O}_d))$.

in step 6 of Algorithm 1, which is now replaced by

$$6'.\ \text{Compute: } \begin{cases} \text{For } i \notin I,\ v_i = f_i(\overline{PA}(v_i), \epsilon_i) \\ \text{For } i \in I,\ v_i = \tilde{f}_i(\widetilde{\overline{PA}}(v_i), \epsilon_i), \end{cases} \tag{46}$$

where the $\overline{PA}$ and $\widetilde{\overline{PA}}(v_i)$ represent the realization of variables in the respective parent sets. We refer to this adaptation as "Algorithm 1i" for estimating the performance measure under the intervention set I.

A.1. Estimating the Number of Iterations for the Monte Carlo Algorithm

We determine the number of observations M by first setting an accuracy level $\epsilon > 0$ and confidence level $\alpha \in (0,1)$. By collecting M samples, the half-width of $100(1-\alpha)\%$ confidence interval for Ψ will be at most $\epsilon\hat{\Psi}$. In other words, with probability $1-\alpha$, the relative error from estimator $\hat{\Psi}$ towards ground truth Ψ is controlled within ϵ. To obtain M, we first collect M_0 samples of the performance measure, denoted as $\psi_1, \psi_2, \cdots, \psi_{M_0}$. We denote $\bar{\psi}_{M_0}$ as the sample average and $\hat{\sigma}^2_{\psi,M_0}$ as the sample variance of M_0 samples. Then, we compute,

$$M' = \min\{M : z_{1-\alpha/2}\hat{\sigma}_{\psi,M_0}/\sqrt{M} \le \epsilon\bar{\psi}_{M_0}\} = (z_{1-\alpha/2}\hat{\sigma}_{\psi,M_0}/(\epsilon\bar{\psi}_{M_0}))^2,$$

Algorithm 2 Performance Evaluation for M/M/1 With Speed-ups

1: **for** $m = 1$ to M **do**:
2: **for** $i = 1$ to n **do**
3: Draw independent observations $\epsilon_{T_i}, \epsilon_{A_i}, \epsilon_{S_i}$ from $U[0,1]$ distribution.
4: Use (9, 10, 11) to generate the realizations of t_i, a_i, s_i.
5: Plug these realizations into (12) to obtain the waiting time vector $\bar{w}_i = (w_1, \ldots, w_i)$.
6: **end for**
7: Use $\bar{w}_n$ and (19) to compute the realization of the performance measure $\psi_m(\bar{o}_d),\ d = 4n$.
8: **end for**
9: Compute $\hat{\Psi} = \frac{1}{M}\sum_{m=1}^{M} \psi_m(\bar{o}_d)$ to obtain an estimate of $\mathbb{E}_{\bar{O}_d \sim P^{\mathcal{Q}}}(\psi(\bar{O}_d))$.

Algorithm 3 Performance Evaluation Under an Intervention

Follow Algorithm 1, replacing Step 6 with the following:

6′:

$$\text{Compute} \begin{cases} \text{For } i \notin I,\ v_i = f_i(\overline{PA}(v_i), \epsilon_i) \\ \text{For } i \in I,\ v_i = \tilde{f}_i(\widetilde{\overline{PA}}(v_i), \epsilon_i), \end{cases}$$

where M' refers to the minimal sample size to achieve relative error ϵ with confidence level $100(1-\alpha)\%$ based on the estimation from first M_0 runs. $z_{1-\alpha/2}$ refers to $(1-\alpha/2)$th quantile of the standard normal random variable. If $M' \leq M_0$, then we stop and return $\hat{\Psi} = \bar{\psi}_{M_0}$; otherwise, we continue to run until we collect M' samples and report the average of M' samples. In the experiments shown from main content, we set $M_0 = 20, \alpha = 0.05, \epsilon = 0.02$.

B. Data-Driven Estimation in the Parametric Case

Suppose that the assignment functions $\mathcal{F}$ are parameterised by a vector $\xi = (\xi_1, \xi_2, \ldots, \xi_l)$. The set of assignment functions thus falls within a parametric family, $\{\mathcal{F}_\xi : \xi \in \Xi\}$, where $\Xi \subseteq \mathbb{R}^l$ is the parameter space - a finite-dimensional subset of Euclidean space. Evaluating the joint density for the observed realization of the event log $\bar{o}_d = (v_1, v_2, \ldots, v_d)$ yields the observational distribution $P_\xi^{\mathcal{Q}}(v_1, v_2, \ldots, v_d)$, which is the parametric likelihood function. We can now apply the Maximum Likelihood Estimation (MLE) to find the values of model parameters that maximize this likelihood function over parameter space, that is

$$\hat{\xi} \in \arg\max_{\xi \in \Xi} P_\xi^{\mathcal{Q}}(v_1, v_2, \ldots, v_d). \tag{47}$$

We note that an implicit assumption in obtaining statistically reliable parameter estimates is that $l << d$, i.e., the size of the available event log is much larger than the number of parameter values to be estimated. The specific value $\hat{\xi} \in \Xi$, called the *MLE estimate*, yields a plug-in estimate for the observational distribution via $P_{\hat{\xi}}^{\mathcal{Q}}$ in (1), (3), and for the assignment functions in Algorithm 1.

Under the intervention $do(V_k := \tilde{N}_k)$ where $\tilde{N}_k$ follows density $\tilde{P}$, we can estimate the interventional distribution using Theorem 1:

$$P_{\hat{\xi}}^{\mathcal{Q};do(V_k:=\tilde{N}_k)}(V_1, V_2, \ldots, V_d) = \prod_{j \neq k} P_{\hat{\xi}}^{\mathcal{Q}}(V_j | PA_j)\tilde{P}(\tilde{N}_k). \tag{48}$$

As before, we can evaluate the performance metric under the specified intervention using Algorithm $1i$. We demonstrate this parametric case for the M/M/1 with speed-ups.

EXAMPLE 8 (PARAMETRIC ESTIMATION FOR THE M/M/1 WITH SPEED-UPS). We assume that the structure of the SCQM is the same as in Example 1, however the parameter vector $\hat{\xi} = (\hat{\lambda}, \hat{\mu}_0, \hat{\mu}_1, \hat{p})$ now has to be estimated from the event log data. We maximize the likelihood function to obtain:

$$(\hat{\lambda}, \hat{\mu}_0, \hat{\mu}_1, \hat{p}) \in \underset{\lambda,\mu_0,\mu_1,p}{\arg\max} \prod_{i=1}^{n} \left(\lambda \mathrm{e}^{-\lambda t_i}\right) \cdot \left(p^{a_i}(1-p)^{1-a_i}\right) \cdot \left((1-a_i)\mu_0 \mathrm{e}^{-\mu_0 s_i} + a_i \mu_1 \mathrm{e}^{-\mu_1 s_i}\right). \tag{49}$$

The above optimization problem results in the standard estimates of arrival and service rates, as $\hat{\lambda} = 1/\hat{\tau}_T, \hat{\mu}_0 = 1/\hat{\tau}_{S,0}, \hat{\mu}_1 = 1/\hat{\tau}_{S,1}$, where, as in (3), $\hat{\tau}_T, \hat{\tau}_{S,0}, \hat{\tau}_{S,1}$ are given by

$$(\hat{\tau}_T, \hat{\tau}_{S,0}, \hat{\tau}_{S,1}) = \left(\frac{\sum_{i=1}^n T_i}{n}, \frac{\sum_{i=1}^n (1-A_i) S_i}{n - \sum_{i=1}^n A_i}, \frac{\sum_{i=1}^n A_i S_i}{\sum_{i=1}^n A_i}\right).$$

Next, we plug the estimated parameter vector $\hat{\theta} = (\hat{\lambda}, \hat{\mu}_0, \hat{\mu}_1)$ and intervened speed-up probability p^* into the likelihood function to estimate the interventional distribution(as before $d = 4n$):

$$P_{p^*}^{\tilde{\mathcal{Q}}} = P_{\hat{\theta},p^*}(\bar{O}_d = \bar{o}_d) = \begin{cases} \prod_{i=1}^{n} \left(\hat{\lambda}\mathrm{e}^{-\hat{\lambda} t_i}\right) \cdot \left((p^*)^{a_i}(1-p^*)^{1-a_i}\right) \cdot \left((1-a_i)\hat{\mu}_0 \mathrm{e}^{-\hat{\mu}_0 s_i} + a_i \hat{\mu}_1 \mathrm{e}^{-\hat{\mu}_1 s_i}\right) \\ \qquad\qquad \text{if } w_i = \max(w_{i-1} + s_{i-1} - t_i, 0), \forall i \in \{1, 2, \ldots, n\} \\ 0 \qquad\qquad \text{otherwise.} \end{cases} \tag{50}$$

To evaluate the waiting time performance metric specified in Example 2, we apply Algorithm 1 directly replacing the true parameters λ, μ_0, μ_1, p with their sample estimates and the intervened speed-up probability $\hat{\lambda}, \hat{\mu}_0, \hat{\mu}_1, p^*$ in the structural equations (9, 10, 11) in Step 4. This yields the Monte Carlo estimate of the interventional value of our performance measure $\Psi_{\hat{\theta},p^*}(\bar{O}_d)$, as:

$$\hat{W}_{\hat{\theta},p^*} := \frac{1}{n}\sum_{i=1}^{n} W_i \quad \text{with } \bar{O}_d \sim P_{\hat{\theta},p^*} \tag{51}$$

Note the similarity of this example to the data-driven estimation of the QT in Section 3. The main difference is that while the QT can plug in the obtained estimates directly into the closed-form formula (4) to obtain the waiting time under the intervention policy p^*, the DS has to use the Monte Carlo simulation Algorithm 1. We note that the estimator $\hat{W}_{\hat{\theta},p^*}$ requires the domain knowledge of the system as well: it requires to know the inter-arrival times and service times follow exponential distribution. Besides, it also relies on the knowledge of causal structure of M/M/1 with speed-ups, which are Lindley's recursion and the intervention mechanism. Not surprisingly, as discussed in Section 7 below, with sufficient amount of event log data the two approaches yield very similar results.

C. Approximation of Lindley's Recursion

We experiment with alternative network widths by varying the number of hidden neurons in the one-layer architectures (e.g., 1, 2, 4, 8, …, 512), and find via cross-validation that networks with 128 hidden units yield the lowest approximation error. Hence, we report results only for the 128-hidden-unit networks in what follows. We compare the approximation performance of a one-layer multi-layer perceptron (MLP) and a one-layer conditional normalizing flow (NF), each with 128 hidden units, in approximating the output of Lindley's recursion under various sample sizes and speed-up probability settings. The NF model consistently fails to train across all settings: its training loss diverges

to infinity and yields undefined (NaN) in-sample log-likelihoods. This failure stems from the fact that NF is designed to model absolutely continuous densities, and thus cannot represent degenerate (i.e., deterministic) distributions, such as those arising from Lindley's recursion. As the model attempts to assign increasingly high density to deterministic outputs, it encounters numerical instability and collapses during optimization. To mitigate this issue, we introduce a small amount of stochasticity by adding exponential noise with mean 10^{-6} to the output, which prevents divergence and enables training to proceed. Despite this fix, the NF still yields significantly worse performance than MLPs, particularly in the low-sample regime. These observations are quantitatively confirmed in Table 3, where the NF's mean squared errors remain orders of magnitude higher than those of the MLP when $n = 500$.

In contrast, the MLP model demonstrates stable and reliable performance across all configurations, corroborating the findings of Palomo and Pender (2020), who showed that a simple neural network can accurately learn Lindley's recursion. For all tested combinations of sample size n and speed-up probability pairs (p, p^*), the average mean squared error remains well below the predefined threshold $\delta = 0.01$. This indicates that the MLP is capable of capturing the deterministic mapping induced by Lindley's recursion with high accuracy. Moreover, increasing the sample size consistently improves the approximation quality, with error margins decreasing notably as n grows from 500 to 50,000. These results confirm that standard regression architectures like MLPs are well-suited for learning deterministic mappings, even in the presence of distributional shifts between training and testing (i.e., when $p \neq p^*$).

(p, p^*)	$n = 500$	$n = 5000$	$n = 50000$
$(0.1, 0.1)$	$4.8 \times 10^{-4} (\pm 3.0 \times 10^{-5})$	$1.8 \times 10^{-5} (\pm 2.4 \times 10^{-6})$	$1.8 \times 10^{-4} (\pm 2.1 \times 10^{-5})$
$(0.1, 0.5)$	$2.6 \times 10^{-4} (\pm 6.6 \times 10^{-6})$	$1.6 \times 10^{-5} (\pm 6.1 \times 10^{-7})$	$4.1 \times 10^{-5} (\pm 2.2 \times 10^{-6})$
$(0.1, 0.9)$	$1.6 \times 10^{-4} (\pm 4.2 \times 10^{-6})$	$1.2 \times 10^{-5} (\pm 5.0 \times 10^{-7})$	$1.7 \times 10^{-5} (\pm 5.2 \times 10^{-7})$
$(0.5, 0.1)$	$1.9 \times 10^{-3} (\pm 4.0 \times 10^{-4})$	$7.6 \times 10^{-5} (\pm 1.9 \times 10^{-5})$	$2.2 \times 10^{-5} (\pm 2.3 \times 10^{-6})$
$(0.5, 0.5)$	$2.7 \times 10^{-4} (\pm 2.0 \times 10^{-5})$	$1.1 \times 10^{-5} (\pm 7.4 \times 10^{-7})$	$7.8 \times 10^{-6} (\pm 2.6 \times 10^{-7})$
$(0.5, 0.9)$	$9.1 \times 10^{-5} (\pm 3.0 \times 10^{-6})$	$6.1 \times 10^{-6} (\pm 4.5 \times 10^{-7})$	$4.0 \times 10^{-6} (\pm 1.2 \times 10^{-7})$
$(0.9, 0.1)$	$7.5 \times 10^{-3} (\pm 1.7 \times 10^{-3})$	$5.7 \times 10^{-4} (\pm 8.3 \times 10^{-5})$	$6.6 \times 10^{-4} (\pm 2.0 \times 10^{-4})$
$(0.9, 0.5)$	$4.1 \times 10^{-4} (\pm 7.8 \times 10^{-5})$	$5.7 \times 10^{-5} (\pm 6.8 \times 10^{-6})$	$3.9 \times 10^{-5} (\pm 4.4 \times 10^{-6})$
$(0.9, 0.9)$	$5.0 \times 10^{-5} (\pm 9.3 \times 10^{-6})$	$8.2 \times 10^{-6} (\pm 1.1 \times 10^{-6})$	$1.1 \times 10^{-5} (\pm 4.7 \times 10^{-7})$

(a) Mean squared error for multi-layer perceptron (MLP).

(p, p^*)	$n = 500$	$n = 5000$	$n = 50000$
$(0.1, 0.1)$	$5.7 \times 10^{0} (\pm 2.8 \times 10^{0})$	$5.5 \times 10^{-2} (\pm 1.2 \times 10^{-2})$	$1.1 \times 10^{-1} (\pm 1.2 \times 10^{-2})$
$(0.1, 0.5)$	$1.3 \times 10^{-1} (\pm 7.6 \times 10^{-2})$	$1.8 \times 10^{-2} (\pm 7.9 \times 10^{-4})$	$3.5 \times 10^{-2} (\pm 2.3 \times 10^{-3})$
$(0.1, 0.9)$	$1.4 \times 10^{-2} (\pm 3.8 \times 10^{-4})$	$9.3 \times 10^{-3} (\pm 2.3 \times 10^{-4})$	$1.5 \times 10^{-2} (\pm 4.7 \times 10^{-4})$
$(0.5, 0.1)$	$4.3 \times 10^{0} (\pm 2.2 \times 10^{0})$	$2.0 \times 10^{0} (\pm 1.2 \times 10^{0})$	$4.9 \times 10^{-1} (\pm 3.4 \times 10^{-1})$
$(0.5, 0.5)$	$1.3 \times 10^{-1} (\pm 5.3 \times 10^{-2})$	$2.5 \times 10^{-2} (\pm 1.9 \times 10^{-2})$	$1.7 \times 10^{-2} (\pm 2.4 \times 10^{-3})$
$(0.5, 0.9)$	$4.0 \times 10^{-2} (\pm 9.4 \times 10^{-4})$	$4.4 \times 10^{-3} (\pm 1.0 \times 10^{-4})$	$6.8 \times 10^{-3} (\pm 2.0 \times 10^{-4})$
$(0.9, 0.1)$	$1.6 \times 10^{1} (\pm 6.3 \times 10^{0})$	$7.6 \times 10^{0} (\pm 3.7 \times 10^{0})$	$4.3 \times 10^{0} (\pm 2.2 \times 10^{0})$
$(0.9, 0.5)$	$6.2 \times 10^{-1} (\pm 2.7 \times 10^{-1})$	$1.6 \times 10^{-1} (\pm 1.0 \times 10^{-1})$	$8.1 \times 10^{-2} (\pm 5.4 \times 10^{-2})$
$(0.9, 0.9)$	$1.8 \times 10^{-2} (\pm 1.5 \times 10^{-3})$	$4.8 \times 10^{-3} (\pm 2.2 \times 10^{-4})$	$5.4 \times 10^{-3} (\pm 1.7 \times 10^{-4})$

(b) Mean squared error for normalizing flow (NF).

Table 3 Approximation of Lindley's recursion: mean squared error for multi-layer perceptron (MLP) and normalizing flow (NF). We report the average mean squared error over 25 independent replications, with the 95% confidence interval in parentheses, under various (n, p, p^*) configurations.

D. Detailed Results for M/M/1 with Speed-ups

Here we provide details of our computational experiments for our running example: the M/M/1 model with speed-ups. We compare the queuing theory-based estimate described in Section 3 (QT) with two SCQM estimates: the "full" non-parametric estimate constructed following the methodology in Section 6.3 (DSF) and the parametric estimate described in Appendix B (DSP).

For both DSP and DSF methods we obtain the corresponding average waiting time estimates W_{DSP}, W_{DSF} from the Monte Carlo simulation Algorithm 1. This step requires us to estimate the number of iterations M of the algorithm. This is accomplished as follows: we first perform M_0 iterations (we used $M_0 = 20$) to estimate the standard deviation of our waiting time estimates; we then compute M to insure that the width of the 95% Confidence Interval is within $\epsilon = .02$. The details of this procedure are in Appendix A.1.

Results for $p \in \{0.01, 0.5, 0.99\}$ and $n = 5000, 50000$ are presented on Table 4, with the two panels corresponding to the two event log sizes. We see that the performance of all three estimators is quite similar under all conditions. In the middle column, $p = 0.5$ and all three estimators benefit from having a larger data sample, with errors for $n = 50000$ case several times lower than the corresponding results for $n = 5000$. This improve accuracy points to the sensitivity of any data-driven estimate to errors made in estimating process parameters (for QT and DSP) or process assignment functions (DSF). The gaps between the accuracy of different estimators are quite small: less than 1% for $n = 5000$ and below 0.3% for $n = 50000$. This is perhaps to be expected for W_{QT} vs W_{DSP}, since the only difference between these two estimates is that the former relies on the closed-form expression (4), while the latter uses Monte Carlo simulation to obtain the same quantity. However, the excellent accuracy of W_{DSF} is surprising, since it indicates that knowledge of queueing theory does not impart a significant advantage in this case.

While the middle row on Table 4 represents the "good" observational data case, since it uses $p = 0.5$, ensuring that both actions $a = 0$ and $a = 1$ are frequent in the data, the left and right columns from Table 4 presents the results for the "poor" observational data cases, where $p = 0.01$ (indicating a very low percentage of sped-up customers in the training data) and $p = 0.99$ (nearly every customer is sped up). While the results are broadly similar to the previous case, we do observe some differences. First, the estimation errors for all three methods can be very large when the differences between p and p^* are extreme. This is intuitive: when the system is observed under one extreme, and we are estimating the counter-factual performance under the other extreme (e.g. speed-ups switch from being very common to very rare), estimating accuracy can be quite poor. This poor accuracy is particularly true for the $p = 0.99, p^* \in \{.01, .1\}$ where data comes from lightly-utilized system and the intervention results in a heavily-utilized one. Surprisingly, the QT method is impacted the worst by these extreme conditions, while the performance of DS methods (DSP and DSF) is much more robust. We summarize our findings above with the following observation.

Observation 1 *(1) The performance of the SCQM-based methods is comparable to QT under observational conditions with various combinations of p and p^* values and various observational sample sizes n. Moreover, knowledge of parametric structure of the assignment functions (DSP) does not impart any advantage over the "generic" technique (DSF).*

(2) Large differences between the observational policy p and the counterfactual policy p^* may lead to very large estimation errors, particularly when p^* implies a heavily utilized system.

p \ p^*	0.01	0.1	0.5	0.9	0.99
0.01	16.0 15.3 **15.2**	12.2 11.8 **11.9**	15.0 15.0 **15.0**	27.8 27.6 **28.9**	33.5 33.4 **35.5**
0.1	27.0 24.7 **25.5**	18.1 16.6 **17.9**	9.0 9.2 **9.2**	10.2 10.1 **10.1**	11.6 11.6 **12.2**
0.5	23.9 23.0 **22.7**	15.7 15.7 **16.0**	6.6 6.8 **7.1**	6.0 5.9 **6.7**	6.9 7.1 **7.9**
0.9	536.0 70.0 **68.7**	44.6 41.9 **40.5**	14.0 13.9 **13.5**	5.3 5.3 **5.2**	4.8 4.9 **5.2**
0.99	132.6 79.6 **82.2**	102.8 86.5 **86.9**	32.0 31.6 **33.7**	7.8 7.6 **8.8**	3.6 3.7 **4.0**

(a) $n = 5000$

p \ p^*	0.01	0.1	0.5	0.9	0.99
0.01	5.9 5.8 **6.1**	4.0 4.0 **4.1**	4.3 4.6 **4.3**	8.5 8.5 **8.8**	10.3 10.2 **10.9**
0.1	5.1 5.3 **5.5**	3.7 3.8 **4.1**	2.4 2.3 **2.4**	3.1 3.1 **3.1**	3.6 3.7 **3.7**
0.5	5.2 5.2 **5.5**	3.7 3.9 **3.9**	1.6 1.6 **1.8**	1.5 1.5 **1.4**	1.7 1.6 **2.0**
0.9	12.9 12.8 **12.8**	9.3 9.6 **9.4**	3.6 3.7 **3.8**	1.3 1.4 **1.6**	1.2 1.2 **1.4**
0.99	35.8 35.3 **35.6**	60.4 58.8 **57.5**	12.8 12.9 **12.7**	3.1 3.3 **3.1**	1.2 1.2 **1.3**

(b) $n = 50000$

Table 4 Relative absolute error (in percentage) of QT and DS in M/M/1 with speed-ups, $n \in \{5000, 50000\}$. Each cell presents the relative errors (in %) for three estimators. The first row refers to QT (queueing theorist) and DSP (data science - parametric) estimates. The second row (in bold) refers to DSF (data science full) estimates.

E. Additional Results for GI/M/m with Speed-ups

E.1. Queueing Theorist Approach

The arrival process of GI/M/1 with speed-ups is given in Section 7.2. The service process is the same with M/M/1 with speed-up example. Since non-exponential arrivals together with speed-ups lead to a GI/GI/m queue, for which no analytical solutions are available, the QT must employ some approximation formulae. A number of approximations have been proposed in the literature. We consider five such approximations.

We start with approximations of expected waiting time in GI/GI/1 and later show how to extend them into GI/GI/m with appropriate scaling. Buzacott and Shanthikumar (1993) (Section 3.4.4, pp. 74, 75) propose three different approximations, which we refer to as BS1, BS2 and BS3. The approximation formulae are provided as follows. Let as denote $C_T^2 := \mathbb{V}(T)/(\mathbb{E}(T))^2, C_S^2 := \mathbb{V}(S)/(\mathbb{E}(S))^2$ as the squared coefficient of variation for inter-arrival time and service time, respectively. Then the expected waiting time of G/G/1 queue is approximated by

$$W_{BS1} = \frac{\rho^2(1+C_S^2)}{1+\rho^2 C_S^2} \cdot \frac{C_T^2 + \rho^2 C_S^2}{2\lambda(1-\rho)} \qquad W_{BS2} = \frac{\rho(1+C_S^2)}{2-\rho+\rho C_S^2} \cdot \frac{\rho(2-\rho)C_T^2 + \rho^2 C_S^2}{2\lambda(1-\rho)} \tag{52}$$

$$W_{BS3} = \frac{\rho C_T^2(1-(1-\rho)C_T^2) + \rho^2 C_S^2}{2\lambda(1-\rho)} \tag{53}$$

All three use squared coefficient of variations of arrivals C_T^2 and, according to Buzacott and Shanthikumar (1993), all three should work well when $C_T^2 \leq 1$. This is clearly the case in our example since the inter-arrival time is drawn from uniform distribution (with $C_T^2 = 1/3$) and exponential distribution (with $C_T^2 = 1$). An alternative popular approximation is proposed in Whitt (1983), equation (44). We refer to these as W83 and W89 respectively. We herein provide the details on computing W_{W83}. The formula of W_{W83} is shown as:

$$W_{W83} = \begin{cases} \frac{\tau\rho(C_T^2+C_S^2)}{2(1-\rho)} \cdot \exp\left[-\frac{2(1-\rho)}{3\rho} \cdot \frac{(1-C_T^2)^2}{C_T^2+C_S^2}\right], & \text{if } C_T^2 < 1 \\ \frac{\tau\rho(C_T^2+C_S^2)}{2(1-\rho)}, & \text{if } C_T^2 \geq 1. \end{cases} \tag{54}$$

We can notice that all approximations mentioned above rely on the first two moments of inter-arrival time and service. In contrast, Whitt (1989) considers including the third moment of inter-arrival time for approximation, referred as W_{89}. We implement it under the instruction from (Whitt 1989, Section 5) and omit the details here.

Moreover, Buzacott and Shanthikumar (1993) mentioned that the approximation of G/G/1 can be extended to approximation of G/G/m with proper scaling. Let $W \in \{W_{BS1}, W_{BS2}, W_{BS3}, W_{W83}, W_{89}\}$ be one of the approximation of G/G/1 expected waiting time. Let $W_{M/M/m}$ denote the expected waiting time under the same utilization in an M/M/m queue. Then we can have:

$$W^{(m)} = \frac{W_{M/M/m}}{W_{M/M/1}} \cdot W \qquad W \in \{W_{BS1}, W_{BS2}, W_{BS3}, W_{W83}, W_{89}\}. \tag{55}$$

$W^{(m)}$ denotes the extension of approximation W in a G/G/c queue. We include a superscript (m) to highlight that approximation is applied to a multi-server system with m servers. In the special case of $m = 1$, it is clear that $W^{(1)} = W$ and it means we do not need any adaption for a single-server system.

For GI/GI/m system, there is an additional approximation given by Whitt (1993). The formula is shown as:

$$W_{W93}^{(m)} = \frac{1}{\mu} \cdot \phi \cdot \frac{C_T^2 + C_S^2}{2} \cdot W_{M/M/m}. \tag{56}$$

The coefficient ϕ is constructed as follows. We first denote

$$\gamma := \{0.24, (1-\rho)(m-1)((4+5m)^{1/2} - 2)/(16m\rho)\}$$

and $\phi_1 = 1 + \gamma, \phi_2 = 1 - 4\gamma$. Furthermore, we can compute $\phi_3 = \phi_2 \exp(-2(1-\rho)/3\rho)$ and $\phi_4 = \min\{1, (\phi_1 + \phi_3)/2\}$. Let

$$\Psi = \begin{cases} 1, & \text{if } (C_T^2 + C_S^2)/2 \geq 1 \\ \phi_4^{2(1-(C_T^2+C_S^2)/2)}, & \text{if } 0 \leq (C_T^2 + C_S^2)/2 \leq 1. \end{cases}$$

Then we can express coefficient ϕ as:

$$\phi = \begin{cases} \frac{4(C_T^2 - C_S^2)}{4C_T^2 - 3C_S^2}\phi_1 + \frac{C_S^2}{4C_T^2 - 3C_S^2}\Psi, & \text{if } C_T^2 \geq C_S^2 \\ \frac{C_S^2 - C_T^2}{2C_T^2 + 2C_S^2}\phi_3 + \frac{C_S^2 + 3C_T^2}{2C_T^2 + 2C_S^2}\Psi, & \text{if } C_T^2 \leq C_S^2. \end{cases} \tag{57}$$

Hence, we can compute $W_{W93}^{(m)}$.

To employ the approximations above is a data-driven format, the QT first uses the data to obtains estimates of the first two moments of inter-arrival times, along with service rate under each action, and then plugs them into the respective formula. In the numerical experiment, queueing theorist will approximate the expected waiting time from $\{W_{BS1}^{(m)}, W_{BS2}^{(m)}, W_{BS3}^{(m)}, W_{W83}^{(m)}, W_{89}^{(m)}, W_{W93}^{(m)}\}$ and we will compare the best approximation with DS solution.

E.2. GI/M/2 with Speed-ups

For the data scientist (DS), the process to estimate SCQM in GI/M/2 with speed-ups involves three main steps: refining the parent sets (Section 6.2), estimating the assignment functions (Section 5), and running the Monte Carlo simulation (Algorithm 1). We first discuss the estimation of SCQM, which contains parent set refinement and assignment function estimation.

- **Parent sets for T_i, A_i, and S_i:** These can be recovered from data since their causal structure matches the M/M/1 example. T_i and S_i are fitted using their empirical distributions, while A_i follows a Bernoulli distribution with success probability p^*.

- **Parent set for W_i:** The parent set of W_i can be refined as $\{t(S_{[IS(t(W_i))]}), t(T_i)\}$, since they are the only relevant components in Krivulin's recursion from default parent set $PA(W_i)$ given by Corollary 1. Moreover, DS can observe that $t(S_{[IS(t(W_i))]}) = t(S_{[i-m]})$ for some $m \in \mathbb{N}_+$ and for all $i \in \{m+1, m+2, \ldots, n\}$. Observation holds as data are generated from GI/G/2 queues. By Krivulin's recursion, before customer i starts service, the last observed service completion before customer i starts service is exactly the $i-2$-th observed service completion. Solving for m is straightforward. DS can enumerate customer $i \in \{1, 2, \ldots, n\}$ until DS finds a customer i^* such that for all customer $i \geq i^*$, the difference between $IS(t(W_i))$ and i does not change with respect to i. DS will solve c as $i^* - 1$. Next, DS will estimate the structural assignment functions on W_i by solving (28). Namely, DS fits a function mapping $\hat{f}_W$ between the inputs $\{t(S_{[IS(t(W_i))]}), t(T_i)\}$ and output W_i.

The estimation of SCQM is completed. The last step of DS is to simulate the queueing data under policy p^* with Algorithm 1. Simulation of T_i, A_i, S_i is easy as their likelihood has been specified. DS can collect all realizations of T_i, A_i, S_i before simulating W_i. To simulate W_i for all $i < i^*$, DS will assign $W_i = 0$ and obtain $t(S_i) = t(T_i) + S_i$. To simulate W_i for all $i \geq i^*$, DS will do it recursively. DS starts from customer i^* and first needs to pick up $t(S_{[i^*-c]}) = t(S_{[1]})$ as the timestamp of first service completion event before W_i gets simulated. Then DS will take $\{t(S_{[1]}), t(T_{i^*})\}$ as the input into $\hat{f}_W$ which returns W_{i^*}. DS will update the service completion time of customer i^* by $t(S_{i^*}) = t(T_{i^*}) + W_{i^*} + S_{i^*}$. By doing so recursively, at customer $i \geq i^*$, DS will use the timestamp of $1 + i - i^*$-th simulated service completion event and arrival timestamp of customer i to predict W_i and update the service completion time of customer i.

We visualize the performance of DS under $p = 0.5$ in following Figure 5. Parent set refinement results has been covered in in Figure 2. Left column presents the relative errors for the $\pi = 0.5$ and $p = 0.5$ case, i.e., when inter-arrival times are not very far from the exponential case. We see that under various choices of p^*, the error of the DS approach is much lower than that achieved by the best of the five approximation. Right column in Figure 5 presents the errors under $\pi = 1$, where the inter-arrival process further deviates from exponential. We observe that in this case, DSF significantly outperforms all approximations as well. Comparing the results for the five approximations, we observe that there is no clear winner, with all methods making large errors in some cases. The W93 appear particularly unsafe, leading to extremely large errors in some cases, while BS1, BS2, W83 are generally more accurate. The percentage errors of QT and DS for other $p \in \mathcal{P}$ are shown in Table 5.

E.3. Additional results for M/M/1 With State-Dependent Speed-Ups

Figure 6 shows the results of the parent set refinement procedure.

F. Additional Results for Tandem System of Two Single-Server Queues

SCQM structure defined above (see Definition 2) assumes a single-station queueing system, since each customer record O_i includes only one observation of T_i, W_i, and S_i variables. Many real-life systems consist of networks of queues with possibly complex structures (e.g., fork-join and re-entrant networks). An important aspect of such networks is routing - i.e., transitions between different stations, which may be customer and/or context-dependent.

While SCQM can be applied separately to each station in the network, to represent the overall system we need to generalize Definition 2 to include arrival, waiting and service information for each station in each customer record O_i. We also need to include structural assignment function that governs transitions between stations. While a full

p \ p^*	0.01	0.1	0.5	0.9	0.99
0.01	92.2 92.1 91.4 91.8 118.8 183.9 **70.2**	45.3 45.2 45.0 45.2 76.8 101.9 **44.0**	229.5 229.3 229.8 229.0 266.3 441.5 **136.8**	106.0 106.0 113.7 108.0 128.8 169.0 **103.7**	135.2 135.8 147.1 139.2 143.2 213.3 **137.7**
0.1	200.5 200.5 201.1 200.7 148.2 427.5 **81.3**	80.0 80.1 81.8 80.9 100.2 189.9 **70.2**	32.5 32.6 36.6 33.8 87.1 71.2 **32.4**	41.9 42.5 52.8 45.3 122.3 57.6 **42.1**	47.4 48.1 60.9 51.4 134.2 60.5 **48.4**
0.5	59.4 59.3 58.9 59.2 82.8 113.0 **54.8**	52.2 52.1 51.8 52.0 82.2 101.5 **48.2**	19.1 18.9 17.8 18.3 68.4 33.0 **20.0**	18.7 18.2 23.2 18.5 91.1 22.0 **19.9**	21.0 21.0 29.4 22.6 100.0 25.3 **20.7**
0.9	71.9 71.8 70.7 71.4 79.7 105.9 **84.5**	261.1 261.0 260.8 261.0 93.3 497.9 **76.5**	33.2 33.1 32.9 33.2 87.0 51.3 **47.5**	14.5 15.1 26.9 17.8 111.0 20.0 **18.9**	14.0 14.5 32.5 18.0 122.7 16.5 **14.7**
0.99	91.7 91.7 90.4 91.5 100.0 88.2 **93.7**	217.4 217.3 216.4 217.3 185.7 361.5 **85.8**	146.8 146.7 148.2 148.1 132.1 246.4 **56.7**	75.3 75.6 83.4 79.7 152.6 102.2 **42.7**	22.0 22.2 36.8 24.9 111.9 25.7 **19.9**

(a) $\pi = 0.5, n = 500$

p \ p^*	0.01	0.1	0.5	0.9	0.99
0.01	92.2 92.1 91.4 91.8 118.8 183.9 **70.2**	45.3 45.2 45.0 45.2 76.8 101.9 **44.0**	229.5 229.3 229.8 229.0 266.3 441.5 **136.8**	106.0 106.0 113.7 108.0 128.8 169.0 **103.7**	135.2 135.8 147.1 139.2 143.2 213.3 **137.7**
0.1	200.5 200.5 201.1 200.7 148.2 427.5 **81.3**	80.0 80.1 81.8 80.9 100.2 189.9 **70.2**	32.5 32.6 36.6 33.8 87.1 71.2 **32.4**	41.9 42.5 52.8 45.3 122.3 57.6 **42.1**	47.4 48.1 60.9 51.4 134.2 60.5 **48.4**
0.5	59.4 59.3 58.9 59.2 82.8 113.0 **54.8**	52.2 52.1 51.8 52.0 82.2 101.5 **48.2**	19.1 18.9 17.8 18.3 68.4 33.0 **20.0**	18.7 18.2 23.2 18.5 91.1 22.0 **19.9**	21.0 21.0 29.4 22.6 100.0 25.3 **20.7**
0.9	71.9 71.8 70.7 71.4 79.7 105.9 **84.5**	261.1 261.0 260.8 261.0 93.3 497.9 **76.5**	33.2 33.1 32.9 33.2 87.0 51.3 **47.5**	14.5 15.1 26.9 17.8 111.0 20.0 **18.9**	14.0 14.5 32.5 18.0 122.7 16.5 **14.7**
0.99	91.7 91.7 90.4 91.5 100.0 88.2 **93.7**	217.4 217.3 216.4 217.3 185.7 361.5 **85.8**	146.8 146.7 148.2 148.1 132.1 246.4 **56.7**	75.3 75.6 83.4 79.7 152.6 102.2 **42.7**	22.0 22.2 36.8 24.9 111.9 25.7 **19.9**

(b) $\pi = 1, n = 500$

p \ p^*	0.01	0.1	0.5	0.9	0.99
0.01	14.1 14.1 15.1 14.5 48.5 87.8 **13.7**	11.6 11.6 12.7 12.0 47.9 79.3 **10.9**	14.3 14.3 19.4 15.0 59.2 51.3 **14.9**	31.6 31.7 42.2 33.6 91.1 44.5 **31.5**	39.1 39.4 50.7 41.9 102.5 51.9 **39.7**
0.1	22.7 22.6 22.3 22.5 49.3 89.7 **21.6**	15.4 15.4 15.5 15.3 45.0 76.6 **15.2**	7.4 7.5 13.8 8.7 49.3 42.2 **7.9**	10.2 11.1 29.4 15.2 69.4 16.1 **10.0**	11.8 12.8 33.5 17.3 76.1 14.2 **11.3**
0.5	27.4 27.4 27.9 27.6 46.8 102.1 **25.9**	18.6 18.6 19.3 18.9 39.9 84.4 **17.6**	8.6 8.6 14.9 9.5 38.4 44.9 **9.0**	6.3 7.3 29.0 12.7 49.2 11.5 **5.8**	6.6 7.4 32.4 12.4 53.4 7.3 **7.1**
0.9	139.1 139.1 141.1 140.0 157.0 327.2 **92.4**	55.1 55.2 58.3 56.5 87.6 154.9 **51.3**	17.7 18.3 27.0 21.7 73.7 59.5 **16.6**	7.0 8.4 30.8 14.1 91.8 13.6 **7.0**	6.3 7.3 32.3 12.1 100.0 6.9 **5.6**
0.99	78.4 78.3 77.6 78.1 86.6 136.1 **67.8**	275.1 275.0 275.0 275.1 69.1 546.5 **109.5**	49.2 49.2 52.1 50.4 72.8 88.7 **48.4**	10.7 11.5 29.0 15.7 61.4 16.5 **11.2**	6.3 6.8 32.1 12.0 62.9 7.2 **6.1**

(c) $\pi = 0.5, n = 5000$

p \ p^*	0.01	0.1	0.5	0.9	0.99
0.01	14.1 14.1 15.1 14.5 48.5 87.8 **13.7**	11.6 11.6 12.7 12.0 47.9 79.3 **10.9**	14.3 14.3 19.4 15.0 59.2 51.3 **14.9**	31.6 31.7 42.2 33.6 91.1 44.5 **31.5**	39.1 39.4 50.7 41.9 102.5 51.9 **39.7**
0.1	22.7 22.6 22.3 22.5 49.3 89.7 **21.6**	15.4 15.4 15.5 15.3 45.0 76.6 **15.2**	7.4 7.5 13.8 8.7 49.3 42.2 **7.9**	10.2 11.1 29.4 15.2 69.4 16.1 **10.0**	11.8 12.8 33.5 17.3 76.1 14.2 **11.3**
0.5	27.4 27.4 27.9 27.6 46.8 102.1 **25.9**	18.6 18.6 19.3 18.9 39.9 84.4 **17.6**	8.6 8.6 14.9 9.5 38.4 44.9 **9.0**	6.3 7.3 29.0 12.7 49.2 11.5 **5.8**	6.6 7.4 32.4 12.4 53.4 7.3 **7.1**
0.9	139.1 139.1 141.1 140.0 157.0 327.2 **92.4**	55.1 55.2 58.3 56.5 87.6 154.9 **51.3**	17.7 18.3 27.0 21.7 73.7 59.5 **16.6**	7.0 8.4 30.8 14.1 91.8 13.6 **7.0**	6.3 7.3 32.3 12.1 100.0 6.9 **5.6**
0.99	78.4 78.3 77.6 78.1 86.6 136.1 **67.8**	275.1 275.0 275.0 275.1 69.1 546.5 **109.5**	49.2 49.2 52.1 50.4 72.8 88.7 **48.4**	10.7 11.5 29.0 15.7 61.4 16.5 **11.2**	6.3 6.8 32.1 12.0 62.9 7.2 **6.1**

(d) $\pi = 1, n = 5000$

p \ p^*	0.01	0.1	0.5	0.9	0.99
0.01	4.7 4.7 5.6 5.0 4.7 83.0 **4.9**	4.3 4.3 6.3 4.7 4.4 75.8 **4.4**	7.3 7.3 15.6 9.3 7.3 46.0 **7.8**	14.9 15.5 33.3 19.5 14.9 22.3 **15.1**	17.9 18.6 38.4 22.6 18.3 22.6 **18.1**
0.1	5.2 5.2 6.3 5.6 5.3 84.0 **5.3**	3.8 3.9 6.4 4.5 4.0 75.7 **3.9**	2.9 3.2 14.4 7.0 3.4 43.6 **3.4**	5.6 7.1 30.2 13.6 5.0 12.2 **4.7**	6.1 7.7 34.4 13.9 5.1 6.4 **4.6**
0.5	7.6 7.7 8.4 7.9 7.8 84.6 **6.9**	5.4 5.4 6.9 5.9 5.5 75.8 **6.8**	2.2 2.6 14.0 6.7 2.8 43.0 **2.8**	3.7 5.9 29.2 12.7 2.8 10.9 **2.5**	3.7 6.0 33.1 12.7 2.4 2.5 **2.5**
0.9	14.4 14.4 15.3 14.7 14.5 91.6 **13.1**	10.1 10.2 12.0 10.7 10.2 79.9 **9.8**	4.2 4.6 14.6 8.1 4.6 44.0 **3.8**	4.1 6.1 29.4 12.8 3.2 11.1 **2.0**	3.8 6.0 33.1 12.7 2.2 2.0 **2.8**
0.99	58.8 58.7 58.9 58.7 58.8 147.5 **48.8**	32.7 32.7 33.3 32.9 32.8 97.2 **32.7**	12.7 12.8 18.0 14.5 13.0 46.2 **12.9**	5.2 7.0 30.0 13.6 4.2 12.1 **3.6**	4.7 6.9 33.9 13.6 2.4 3.2 **2.6**

(e) $\pi = 0.5, n = 50000$

p \ p^*	0.01	0.1	0.5	0.9	0.99
0.01	4.7 4.7 5.6 5.0 4.7 83.0 **4.9**	4.3 4.3 6.3 4.7 4.4 75.8 **4.4**	7.3 7.3 15.6 9.3 7.3 46.0 **7.8**	14.9 15.5 33.3 19.5 14.9 22.3 **15.1**	17.9 18.6 38.4 22.6 18.3 22.6 **18.1**
0.1	5.2 5.2 6.3 5.6 5.3 84.0 **5.3**	3.8 3.9 6.4 4.5 4.0 75.7 **3.9**	2.9 3.2 14.4 7.0 3.4 43.6 **3.4**	5.6 7.1 30.2 13.6 5.0 12.2 **4.7**	6.1 7.7 34.4 13.9 5.1 6.4 **4.6**
0.5	7.6 7.7 8.4 7.9 7.8 84.6 **6.9**	5.4 5.4 6.9 5.9 5.5 75.8 **6.8**	2.2 2.6 14.0 6.7 2.8 43.0 **2.8**	3.7 5.9 29.2 12.7 2.8 10.9 **2.5**	3.7 6.0 33.1 12.7 2.4 2.5 **2.5**
0.9	14.4 14.4 15.3 14.7 14.5 91.6 **13.1**	10.1 10.2 12.0 10.7 10.2 79.9 **9.8**	4.2 4.6 14.6 8.1 4.6 44.0 **3.8**	4.1 6.1 29.4 12.8 3.2 11.1 **2.0**	3.8 6.0 33.1 12.7 2.2 2.0 **2.8**
0.99	58.8 58.7 58.9 58.7 58.8 147.5 **48.8**	32.7 32.7 33.3 32.9 32.8 97.2 **32.7**	12.7 12.8 18.0 14.5 13.0 46.2 **12.9**	5.2 7.0 30.0 13.6 4.2 12.1 **3.6**	4.7 6.9 33.9 13.6 2.4 3.2 **2.6**

(f) $\pi = 1, n = 50000$

Table 5 Relative absolute error for QT and DSF for GI/M/2 with speed-ups, $\pi \in \{0.5, 1\}, p \in \mathcal{P}$. Each cell presents the errors under different (p, p^*). The first row refers to BS1, BS2. The second row refers to BS3, W83. The third row refers to W89, W93. The last row refers to DSF (highlighted in bold face).

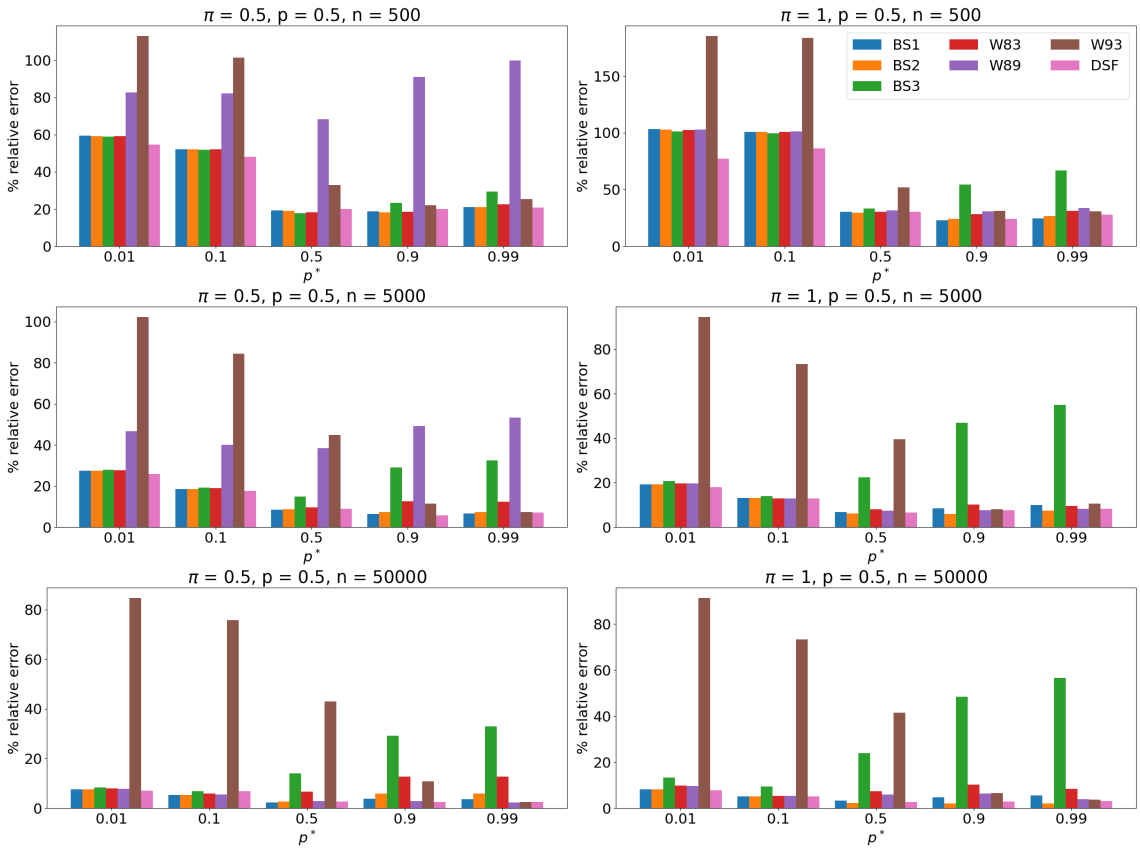


Figure 5 **Relative absolute error for QT and DSF for GI/M/2 with speed-ups,** $\pi \in \{0.5, 1\}, p = 0.5$**.**

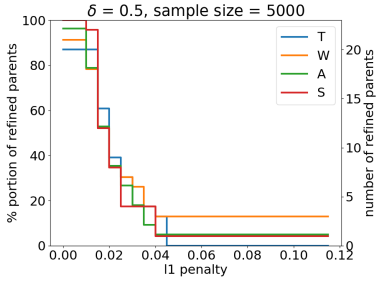


Figure 6 **Proportion of refined parent set with respect to** η **under** $\delta = 0.5, n = 5000$**.**

development of such an extension will be the subject of future work, in this section we illustrate the application of the generalized SCQM to a simple special case - a tandem network of two M/M/1 stations with speed-up and dependent service times. Even for this simple network, the introduction of dependencies between the two stations makes analytical or even approximate analysis problematic for QT. However, as seen below, SCQM approach easily extends to this setting.

F.1. SCQM for Tandem Network with Dependent Service Times.

Customers arrive to the first station of the tandem queue according to a Poisson process with rate λ. Each station has two service modes, with exponential service times at rate μ_0 or μ_1, determined by a binary action, as in the M/M/1 case. To extend SCQM to this system we use the extended observatio a record $O_i := \{(T_{i,q}, W_{i,q}, A_{i,q}, S_{i,q}) : q = 1, 2\}$,

where i denotes the customer and $q \in \{1,2\}$ indexes the station. We assume the following dependency between service times:

$$S_{i,2} = \alpha S_{i,1} + (1-\alpha)\tilde{S}_{i,2}, \quad \alpha \in (0,1), \tag{58}$$

where, $S_{i,1}$ is the service time of customer i at the upstream station and $\tilde{S}_{i,2}$ follows exponential distribution with either rate μ_0 or μ_1, depending on action $A_{i,2}$. Thus, $S_{i,2}$ depends on $S_{i,1}$ (and thus the speed-up action $A_{i,1}$(whenever $\alpha > 0$, while the service times are independent when $\alpha = 0$.

For this example, we assume the DS using SCQM knows that the event log came from the tandem queue system and thus do not try to discover the routing function. However, every other component of the SCQM is discovered and estimated from the training data, following the steps on Figure 3.

The knowledge of the network structure implies that parent set refinement at the first (i.e., upstream) station proceeds as in Example 6. For the second station, Assumption 4 yields the following default parent sets:

$$PA(T_{i,2}) = \bigcup_{j=1}^{K}\bigcup_{q=1}^{2}\{T_{i-j+2-q,q}, W_{i-j+2-q,q}, A_{i-j+2-q,q}, S_{i-j+2-q,q}\}, \tag{59}$$

$$PA(W_{i,2}) = PA(A_{i,2}) = PA(T_{i,2}) \cup \{T_{i,2}\}, \tag{60}$$

$$PA(S_{i,2}) = PA(W_{i,2}) \cup \{W_{i,2}, A_{i,2}\}. \tag{61}$$

Tandem queue structure also implies that the assignment function for inter-arrival times at station 2 is given by

$$T_{i,2} = t(S_{i,1}) - t(S_{i-1,1}) = T_{i,1} + W_{i,1} + S_{i,1} - W_{i-1,1} - S_{i-1,1}, \tag{62}$$

where $t(S_{j,1}) = 0$ for $j \leq 0$.

We next apply the refinement step (with $K = 5$) to the default parent sets - the results are shown in Figure 7a. The left panel shows the results for $\alpha = 0$, the independent service case, which identifies the following parent sets:

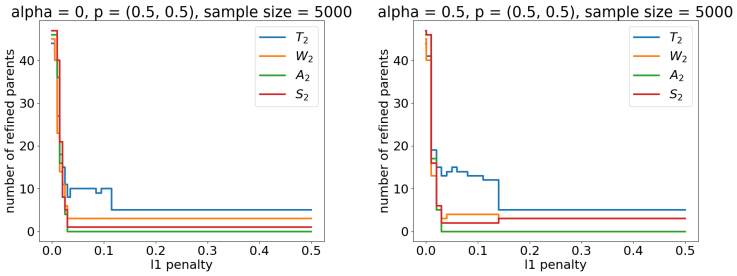
$$PA^R(W_{i,2}) = \{T_{i,2}, W_{i-1,2}, S_{i-1,2}\}, \quad PA^R(A_{i,2}) = \emptyset, \quad PA^R(S_{i,2}) = \{A_{i,2}\},$$
$$PA^R(T_{i,2}) = \{T_{i,1}, W_{i,1}, S_{i,1}, W_{i-1,1}, S_{i-1,1}\},$$

matching the true causal structure. The right panel of Figure 7a corresponds to $\alpha = 0.5$, i.e., the dependent service case. The only difference from $\alpha = 0$ is that $PA^R(S_{i,2}) = \{S_{i,1}, A_{i,2}\}$ in this case, which is also correctly identified by the refinement procedure.

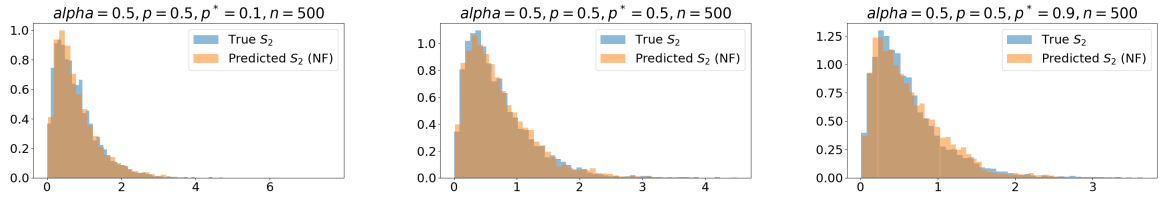
The estimation of assignment functions for station 1 is identical to Example 7. For station 2, the assignment function for $T_{i,2}$ is given by (62 and need not be estimated. The assignment function for $W_{i,2}$ (the Lindley recursion) is estimated by MLP as before. To estimate the assignment function that maps from $PA^R(S_{i,2})$ to $S_{i,2}$, we compare three different estimators: (1) a conditional NF (normalizing flow) with 10 hidden layers and 64 neurons per layer using 500 training samples, (2) the MLP estimator described ealier, and (3) the EDF estimator for the case when $\alpha = 0$ (notice that this estimator is not available for the $\alpha > 0$ case.

We compare the out-of-sample Wasserstein distance between the predicted and true service times over 5,000 test samples. Results are reported in Table 6 for the $\alpha = 0$ case and on Table 7 for the $\alpha = .5$ case.

For $\alpha = 0$, we observe that EDF consistently achieves lower Wasserstein distances than NF across nearly all (p, p^*) configurations and sample sizes (both EDF and NF outperform the MLP estimator, which is not shown). This suggests that, under independence, the empirical distribution estimator is better suited for approximating the conditional distribution than a flexible generative model like NF, possibly due to the simplicity of the conditional structure.



(a) Parent set refinement at station 2 under $\alpha \in \{0, 0.5\}$.



(b) Predicted distribution of $S_{i,2}$ when $\alpha = 0.5$.

Figure 7 **Numerical results for tandem system. Figure 7a shows sizes of refined parent sets at the second station of a tandem system as a function of service dependency α, with $p = 0.5$ at both stations and $n = 5000$. Lines in different colors correspond to different queueing variables $T_{i,2}$, $W_{i,2}$, $A_{i,2}$, and $S_{i,2}$. Figure 7b compares the histogram predicted service time $S_{i,2}$ with the ground truth. The vertical axis presents the empirical density.**

When $\alpha = 0.5$, Table 7 shows that NF significantly and consistently outperforms MLP across all sample sizes and combinations of speed-up probabilities (p, p^*), achieving dramatically lower Wasserstein distances. These results underscore the importance of modeling the full conditional distribution when dependencies exist between service times (MLP models only the conditional mean of the distribution).

Finally, on Figure 7b we compare the distribution of the estimated service times at Station 2 to the simulation-based "ground truth" results for the underlying system. We set $p = .5$ (at both stations), $n = 500$ and $\alpha = .5$. We use different counterfactual policies $p^* \in \{0.1, 0.5, 0.9\}$ (same policy at both stations). The results show that the predicted service time distributions closely match the true ones, in all cases, indicating that SCQM estimator was able to recover the counterfactual service time distribution accurately in spite of dependency in service times between the stations.

This example highlights the robustness of the SCQM approach in recovering causal structure and supporting counterfactual estimation for a systems for which we are not aware of no analytical or even approximate QT solutions and shows the promise of this approach for more general queueing networks.

	$n = 500$		$n = 5000$		$n = 50000$	
(p, p^*)	EDF	NF	EDF	NF	EDF	NF
$(0.1, 0.1)$	$0.35(\pm 0.00)$	$0.36(\pm 0.01)$	$0.36(\pm 0.01)$	$0.29(\pm 0.01)$	$0.32(\pm 0.01)$	$0.39(\pm 0.01)$
$(0.1, 0.5)$	$0.19(\pm 0.00)$	$0.26(\pm 0.01)$	$0.19(\pm 0.01)$	$0.23(\pm 0.01)$	$0.16(\pm 0.01)$	$0.30(\pm 0.01)$
$(0.1, 0.9)$	$0.06(\pm 0.00)$	$0.16(\pm 0.01)$	$0.03(\pm 0.00)$	$0.16(\pm 0.01)$	$0.02(\pm 0.00)$	$0.23(\pm 0.01)$
$(0.5, 0.1)$	$0.28(\pm 0.01)$	$0.42(\pm 0.01)$	$0.32(\pm 0.01)$	$0.28(\pm 0.01)$	$0.33(\pm 0.00)$	$0.37(\pm 0.01)$
$(0.5, 0.5)$	$0.11(\pm 0.01)$	$0.30(\pm 0.01)$	$0.16(\pm 0.01)$	$0.25(\pm 0.01)$	$0.16(\pm 0.01)$	$0.28(\pm 0.01)$
$(0.5, 0.9)$	$0.06(\pm 0.00)$	$0.20(\pm 0.01)$	$0.02(\pm 0.00)$	$0.23(\pm 0.01)$	$0.02(\pm 0.00)$	$0.23(\pm 0.01)$
$(0.9, 0.1)$	$0.28(\pm 0.01)$	$0.43(\pm 0.01)$	$0.34(\pm 0.01)$	$0.38(\pm 0.01)$	$0.33(\pm 0.01)$	$0.33(\pm 0.01)$
$(0.9, 0.5)$	$0.11(\pm 0.01)$	$0.33(\pm 0.01)$	$0.18(\pm 0.00)$	$0.31(\pm 0.01)$	$0.16(\pm 0.00)$	$0.25(\pm 0.01)$
$(0.9, 0.9)$	$0.05(\pm 0.00)$	$0.27(\pm 0.01)$	$0.02(\pm 0.00)$	$0.25(\pm 0.01)$	$0.01(\pm 0.00)$	$0.22(\pm 0.01)$

Table 6 $\alpha = 0$**: Wasserstein distance for empirical distribution function (EDF) and normalizing flow (NF). For both models, we run 25 independent replications to evaluate the Wasserstein distance under various** (n, p, p^*)**. The number out of the parenthesis denotes average Wasserstein distance and the value inside parenthesis denotes width of 95% confidence interval.**

	$n = 500$		$n = 5000$		$n = 50000$	
(p, p^*)	MLP	NF	MLP	NF	MLP	NF
$(0.1, 0.1)$	$0.13(\pm 0.00)$	$0.02(\pm 0.00)$	$0.16(\pm 0.00)$	$0.02(\pm 0.00)$	$0.17(\pm 0.00)$	$0.02(\pm 0.00)$
$(0.1, 0.5)$	$0.08(\pm 0.00)$	$0.02(\pm 0.00)$	$0.11(\pm 0.00)$	$0.02(\pm 0.00)$	$0.12(\pm 0.00)$	$0.02(\pm 0.00)$
$(0.1, 0.9)$	$0.08(\pm 0.00)$	$0.03(\pm 0.00)$	$0.08(\pm 0.00)$	$0.02(\pm 0.00)$	$0.09(\pm 0.00)$	$0.02(\pm 0.00)$
$(0.5, 0.1)$	$0.18(\pm 0.00)$	$0.02(\pm 0.00)$	$0.20(\pm 0.00)$	$0.02(\pm 0.00)$	$0.15(\pm 0.00)$	$0.02(\pm 0.00)$
$(0.5, 0.5)$	$0.13(\pm 0.00)$	$0.02(\pm 0.00)$	$0.14(\pm 0.00)$	$0.01(\pm 0.00)$	$0.11(\pm 0.00)$	$0.02(\pm 0.00)$
$(0.5, 0.9)$	$0.10(\pm 0.00)$	$0.03(\pm 0.00)$	$0.10(\pm 0.00)$	$0.01(\pm 0.00)$	$0.08(\pm 0.00)$	$0.01(\pm 0.00)$
$(0.9, 0.1)$	$0.13(\pm 0.00)$	$0.04(\pm 0.00)$	$0.18(\pm 0.00)$	$0.04(\pm 0.00)$	$0.19(\pm 0.00)$	$0.02(\pm 0.00)$
$(0.9, 0.5)$	$0.08(\pm 0.00)$	$0.02(\pm 0.00)$	$0.13(\pm 0.00)$	$0.02(\pm 0.00)$	$0.14(\pm 0.00)$	$0.02(\pm 0.00)$
$(0.9, 0.9)$	$0.06(\pm 0.00)$	$0.02(\pm 0.00)$	$0.08(\pm 0.00)$	$0.01(\pm 0.00)$	$0.10(\pm 0.00)$	$0.01(\pm 0.00)$

Table 7 $\alpha = 0.5$**: Wasserstein distance for multi-layer perceptron (MLP) and normalizing flow (NF). For both models, we run 25 independent replications to evaluate the Wasserstein distance under various** (n, p, p^*)**. The number out of the parenthesis denotes average Wasserstein distance and the value inside parenthesis denotes width of 95% confidence interval.**