Proceedings of the 2023 Winter Simulation Conference
C. G. Corlu, S. R. Hunter, H. Lam, B. S. Onggo, J. Shortle, and B. Biller, eds.

A PRELIMINARY STUDY OF REGULARIZATION FRAMEWORK FOR CONSTRUCTING TASK-SPECIFIC SIMULATORS

Dilara Aykanat
Zeyu Zheng

Department of Industrial Engineering
and Operations Research
University of California, Berkeley
Berkeley, CA 94720, USA

Nian Si

Booth School of Business
The University of Chicago
Chicago, IL 60615, USA

ABSTRACT

One approach to construct or calibrate simulators, when representative real data exist, is to ensure that the synthetic data generated by the simulated match the empirical distribution of the real data. However, such approach to construct simulators does not take into consideration where the constructed simulators will be used. For some applications, there are clear tasks (such as performance evaluation of different decisions) in users' mind where the simulated data will serve as input to the tasks. In this work, we propose an approach to use the knowledge of these tasks to guide the construction of simulators, in addition to the distribution match of simulated data and real data by regularizing the objective function with a task related penalty. We conduct a preliminary numerical study of this approach to illustrate the effectiveness compared to not taking into consideration the specific tasks of the simulators.

1 INTRODUCTION

Simulators that have the capability of effectively generating synthetic data that can represent real or hypothetical scenarios are useful in multiple domains, including service systems, financial markets, supply chain systems and healthcare systems. Constructing such simulators can be challenging. While there are multiple ways to construct a simulator, one way is to leverage historical real data to build the underlying distribution of a simulator. For example, we may want to construct a simulator that can generate independent copies of future random daily demands, where we have actually collected daily demand data from last month. In this case, it is useful to construct the simulator so that the underlying distribution matches that of the real data from last month, and then use the simulator to generate more representative samples. Such simulator is often in need to facilitate what-if analysis; see Nelson (2016). For another example, a hospital may want to share their large-scale health data to their academic partner in order to collaborate on new approaches to benefit patients. However, such data often contain private information that prohibits sharing. A simulator can be useful in this case to generate synthetic data that on one hand erases private information and, on the other hand, preserves useful population-level distribution information for academic research needs.

Simulators are useful also because they can facilitate various downstream tasks. For example, for service systems, simulated demand data are often used as input to evaluate the expected performance of different potential service plans. For finance related applications, simulated data are often used to evaluate the risks of portfolios. For private health data that are shared in the form of simulated synthetic data, the simulated data are then processed by feasible statistical learning tools for preliminary analysis. A piece of thought that motivates our work is as follows. If we know specifically what the downstream task is, how may we use the knowledge of the downstream task when we construct the simulator?

979-8-3503-6966-3/23/$31.00 ©2023 IEEE

Certainly, there is no unique answer about how to use the downstream task to guide the construction of simulators. Our work aims at providing one viable approach to utilize the knowledge of downstream task to guide the simulator construction. In particular, we consider a simple downstream tasks — expected performance evaluation of some given functional that takes the simulated data as input. With all else equal, we aim at constructing simulators such that the downstream task has close outcome for the two scenarios: (i) real data are fed into the downstream task, and (ii) simulated data are fed into the downstream task.

A brief summary of our contribution and limitation is as follows. We introduce a regularization framework that can be used to create task-specific simulators. Our approach involves designing the simulator while taking into consideration the downstream task, such that the simulator not only matches the distribution of real data but also recovers performance of downstream tasks if fed by the real data. We apply our methods to two different settings: the standard parametric simulator and the Wasserstein generative adversarial networks (Arjovsky et al. 2017). We conclude by showcasing the effectiveness of our framework on a real-world call center dataset. One current limitation of our work is the lack of theoretical analysis. We plan to propose the method in this work, accompanied with a numerical study, and leave the theoretical guarantees of the proposed regularization approach for future research.

1.1 Literature Review

Our work is connected to the literature of using downstream tasks to guide parameter estimation. One line of work is referred to as operational statistics and was introduced in Liyanage and Shanthikumar (2005) and extended in Zhu et al. (2008). In Liyanage and Shanthikumar (2005), an optimal order quantity in the single period newsvendor inventory control problem is directly estimated from the data, which is assumed to be an exponential distribution, rather than the true distribution. Zhu et al. (2008) find the optimal operational statistic using Bayesian analysis. Our approach differs from theirs in that their studies deal with inventory control and solve it to optimality in closed form, whereas our goal is to build a generator for general operations.

Minimum distance estimation involves minimizing the distance between the model distribution μ_θ and the empirical distribution $\hat{\mu}_n$, over the parameter space $\theta \in \mathscr{H}$. Bernton et al. (2019) utilized the Wasserstein distance to develop two distinct point estimators as particular examples of minimum distance estimators. They also evaluated the theoretical properties of these estimators and demonstrated their robustness in the face of misspecified settings, which are common in complex real-world systems. The Wasserstein distance is preferred over other probability distances in various applications, including Wasserstein Generative Adversarial Networks (WGANs) proposed by Arjovsky et al. (2017). WGANs approximate the Wasserstein distance as a cost function, leading to smoother gradients and more stable training, improving generalization performance. Different WGAN variants, such as Max-Sliced WGANs (Deshpande et al. 2019) that approximate the distance by projecting onto low-dimensional subspaces, have been developed to enhance estimation quality and performance in high-dimensional problems. Wasserstein distances have a wide range of applications in data science and operations research, such as image processing (Rubner et al. 2000), financial engineering (Dolinsky and Soner 2014), and distributionally robust optimization (Blanchet and Murthy 2019; Mohajerin Esfahani and Kuhn 2018; Gao and Kleywegt 2022; Mohajerin Esfahani and Kuhn 2018).

The use of neural network to assist the construction of simulators has been adopted (Cen et al. 2020), (Wang et al. 2020), (Zheng et al. 2023), (Cen and Haas 2022) and (Zhu et al. 2023). This line of work develops different approaches that utilize neural networks to construct simulators. Our work additionally utilizes the knowledge of downstream task to build neural network-assisted simulators. In particular, we extend the Doubly Stochastic WGAN (DS-WGAN) method developed in (Zheng et al. 2023).

2 BACKGROUND ON SIMULATORS AND DOWNSTREAM TASKS

2.1 Simulator Models

In this paper, we consider a simulator, which is a mapping from q-dimensional vector of random noise to a p-dimensional vector of random variables. We let $\mathbf{U} = (U_1, U_2, \ldots, U_q)$ to denote the input noise vector and without loss of generality, we assume there are independent and identically distributed (i.i.d.) uniform (0,1) random variables. We denote the output of the simulator as

$$\mathbf{G}(\mathbf{U}, \theta) = (G_1(\mathbf{U}, \theta), G_2(\mathbf{U}, \theta), \ldots, G_p(\mathbf{U}, \theta)),$$

where $\theta \in \Theta$ is a (possibly multi-dimensional) parameter for the simulator.

Now, suppose that there are n i.i.d. real data samples $\{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n\}$ from a population distribution $\nu_{\mathbf{X}}$. We denote the associated empirical distribution as

$$\hat{\nu}_{\mathbf{X}} = \frac{1}{n} \sum_{i=1}^{n} \delta_{\mathbf{X}_i}.$$

The goal is to construct a simulator that is able to generate an output distribution that is close to $\nu_{\mathbf{X}}$. In this paper, we measure the closeness between probability measures using the p-Wasserstein distance, i.e., $\mathscr{W}_p$ defined as:

$$\mathscr{W}_p(P, Q) = \inf_{\gamma \in \Gamma(P,Q)} \left(\int_{\mathscr{Y} \times \mathscr{Y}} \|x - y\|_2^p d\gamma(x, y) \right)^{1/p},$$

where $\Gamma(P, Q)$ is the set of probability measures on $\mathscr{Y} \times \mathscr{Y}$ with marginals P and Q. Therefore, to construct the simulator, ideally, we need to solve

$$\min_{\theta \in \Theta} \mathscr{W}_p^p(\mathbf{G}(\mathbf{U}, \theta), \nu_{\mathbf{X}}), \tag{1}$$

where we slightly abuse the notation to use $\mathbf{G}(\mathbf{U}, \theta)$ to denote the probability distribution of the random variable $\mathbf{G}(\mathbf{U}, \theta)$.

Since $\nu_{\mathbf{X}}$ is not accessible and $\mathscr{W}_p^p(\mathbf{G}(\mathbf{U}, \theta), \hat{\nu}_{\mathbf{X}})$ is hard to compute for an arbitrary $\mathbf{G}(\mathbf{U}, \theta)$, we usually generate m i.i.d. outputs of $\mathbf{Y}_i = \mathbf{G}(\mathbf{U}_i, \theta)$ for $i = 1, 2, \ldots, m$. The corresponding empirical distribution $\hat{\nu}_{\mathbf{Y},\theta}$ is used to approximate $\mathbf{G}(\mathbf{U}, \theta)$ and we use $\hat{\nu}_{\mathbf{X}}$ to approximate $\nu_{\mathbf{X}}$. Then, we solve the problem

$$\min_{\theta \in \Theta} \mathscr{W}_p^p(\hat{\nu}_{\mathbf{Y},\theta}, \hat{\nu}_{\mathbf{X}}). \tag{2}$$

Earlier, it was mentioned that a simulator denoted as $\mathbf{G}(\mathbf{U}, \theta)$ would be employed to evaluate the system's performance using certain important metrics $h(\mathbf{G}(\mathbf{U}, \theta))$. The conventional method involves identifying the optimal $\mathbf{G}$ without taking into account the impact of h and subsequently selecting the downstream task based on h. As a result, the metrics determined by h on the simulator may differ from those obtained from real-world data.

2.2 Downstream Tasks

Once the simulator is built, certain downstream tasks are optimized based on specific metrics represented by h. These downstream tasks can be denoted as follows.

$$\mathbb{E}\left[\min_{\pi \in \Pi} h(\pi, \mathbf{G}(\mathbf{U}, \theta))\right],$$

where π is a possible policy. As demonstrated in Section 4.2 using the call center application as an example, π represents a staffing plan that may involve random service time, while h is indicative of the queue lengths at certain time steps.

Typically, a nominal policy is already implemented in a system in practical settings, which we refer to as π_0. When constructing a simulator, it is essential to ensure that the performance metrics under this policy closely resemble those observed in the actual data, i.e.,

$$\mathbb{E}[h(\pi_0, \mathbf{G}(\mathbf{U}, \theta))] \approx \mathbb{E}_{\nu_{\mathbf{X}}}[h(\pi_0, X)].$$

To ease the notation, we reload the notation h as

$$h(\mathbf{G}(\mathbf{U}, \theta)) = h(\pi_0, \mathbf{G}(\mathbf{U}, \theta)),$$

where $h(\mathbf{G}(\mathbf{U}, \theta))$ could possibly be a random function. In Section 3, we will delve into the techniques that involve explicitly integrating the downstream task metrics in the process of simulator construction.

3 REGULARIZED FRAMEWORK FOR CONSTRUCTING SIMULATORS GIVEN DATA

In this section, we present our methods which incorporate the downstream task information into the simulator construction.

3.1 Building General Simulators With Regularization

Instead of solving Problem (1), we introduce the following regularized problem:

$$\min_{\theta \in \Theta} \mathscr{W}_p^p(\mathbf{G}(\mathbf{U}, \theta), \nu_{\mathbf{X}}) + c \cdot d(\mathbb{E}[h(\mathbf{G}(\mathbf{U}, \theta))], \mathbb{E}[h(\mathbf{X})]), \tag{3}$$

where $d(\cdot, \cdot)$ is some distance function and c is a regularizer coefficient. Since the population distributions cannot be accessed, Problem (3) cannot be solved directly. Therefore, we will address the following problem, which is an analog to Problem (2):

$$\min_{\theta \in \Theta} \mathscr{W}_p^p(\hat{\nu}_{\mathbf{Y},\theta}, \hat{\nu}_{\mathbf{X}}) + c \cdot d(\mathbb{E}_{\hat{\nu}_{\mathbf{Y},\theta}}[h(\mathbf{Y})], \mathbb{E}_{\hat{\nu}_{\mathbf{X}}}[h(\mathbf{X})]).$$

The function h can encompass various metrics, including sample means, or it may be a complex function that relies on specific factors, as discussed in 2.2. c is a tuning parameter, and d is a function that measures the distance between the simulator metrics and the real-data metrics. A natural choice of d is the square function $d(x, y) = \|x - y\|_2^2$.

Exact computation and optimization of the Wasserstein distance $\mathscr{W}_p$ is feasible for low-dimensional probability distributions and small sample sizes. In such cases, calculating the Wasserstein distance reduces to an assignment problem, and specialized linear programming approaches can prove useful. However, when the number of dimensions and sample sizes increase, computing the Wasserstein distance becomes computationally demanding. Regrettably, the data utilized in our analysis is typically high-dimensional, necessitating the use of various approximation methods to estimate it. This drives us to leverage the techniques employed in Wasserstein generative adversarial networks (Arjovsky et al. 2017).

3.2 Simulators Based on Wasserstein Generative Adversarial Networks (WGAN)

The focus of this section is on the utilization of a neural network simulator, $\mathbf{G}(\mathbf{U}, \theta)$, along with another neural network f, to approximate the computation of the Wasserstein distance $\mathscr{W}_p$. This gives rise to a WGAN setting (Arjovsky et al. 2017), where the generator $\mathbf{G}(\mathbf{U}, \theta)$ and discriminator f engage in a zero-sum game, vying against each other to accurately estimate $\mathscr{W}_p$.

WGANs are motivated by the Kantorovich-Rubinstein duality of the Wasserstein distance (Arjovsky et al. 2017). Specifically, we have

$$\mathscr{W}_1(P,Q) = \inf_{f \text{ is a 1-Lipschitz function}} \mathbb{E}_P[f(X)] - \mathbb{E}_Q[f(Y)].$$

Utilizing this duality result, Arjovsky et al. (2017) uses a neural network class $\mathscr{F}_{NN}$ to parameterize the 1-Lipschitz function class. Then, the WGAN performs the following optimization problem

$$\min_{G(\cdot,\theta)\in\mathscr{G}_{\text{NN}}} \max_{f_w\in\mathscr{F}_{\text{NN}}} \mathbb{E}[f_w(G(\mathbf{U},\theta))] - \frac{1}{n}\sum_{i=1}^{n} f_w(\mathbf{X}_i)$$

Generative models like WGAN have diverse applications, ranging from image processing to finance. It is particularly useful when there is a requirement to simulate a complex, high-dimensional distribution with intricate correlation structures, where the underlying properties are not known. An example of this is the work of Zheng et al. (2023), who employed WGAN to create a doubly stochastic arrival process. In their approach, each dimension of the process corresponds to a specific time interval of the day, and the arrival rates (λ_i) are correlated with each other.

By incorporating downstream tasks into the WGAN framework, we can derive (4), which includes a regularizer term that enables the generator training to account for the ultimate cost structure. The generative model obtained through this approach is referred to as R-WGAN.

$$\min_{G(\cdot,\theta)\in\mathscr{G}_{\text{NN}}} \max_{f_w\in\mathscr{F}_{\text{NN}}} \mathbb{E}[f_w(G(\mathbf{U},\theta))] - \frac{1}{n}\sum_{i=1}^{n} f_w(\mathbf{X}_i) + c\cdot d(\mathbb{E}_{\hat{\nu}_{\mathbf{Y},\theta}}[h(\mathbf{Y})], \mathbb{E}_{\hat{\nu}_{\mathbf{X}}}[h(\mathbf{X})]). \tag{4}$$

4 NUMERICAL EXAMPLES

In this section, we detail our numerical experiments that illustrate the effect of regularization on parameter estimation. In Section 4.1, We start with a simple parametric generator for multivariate lognormal distribution where we calculate the exact 2-Wasserstein distance between two samples of equal sizes to estimate the parameters and continue with some examples of R-WGAN, where no assumption for the true model is needed. In Section 4.2, we test the performance of R-WGAN on a real dataset of call center arrivals.

4.1 Parametric Distribution - Multivariate Lognormal

In this subsection, we present a two-dimensional parametric setting where we use the simulator to approximate samples $\mathbf{X} = (X_1, X_2)^\top \in \mathbb{R}^2$ generated from the following joint lognormal distribution:

$$(\log X_1, \log X_2)^\top \sim N((\mu_1^*, \mu_2^*)^\top, \Sigma^*)).$$

We consider the simulator $\mathbf{Y} = \mathbf{G}(\mathbf{U},\theta)$ defined in the following way:

$$Y_i = \exp((\mu_1,\mu_2)^\top + \begin{bmatrix} a_{11} & a_{12} \\ 0 & a_{22} \end{bmatrix} (\Phi^{-1}(\mathbf{U}_{i,1}), \Phi^{-1}(\mathbf{U}_{i,2})))^\top,$$

where $\Phi(\cdot)$ denotes the CDF of the standard Gaussian random variable and $\theta = (\mu_1,\mu_2,a_{11},a_{12},a_{22})^\top$.

Note that

$$\Sigma = \text{Cov}(\log(\mathbf{Y})) = \begin{bmatrix} a_{11}^2 + a_{12}^2 & a_{12}a_{22} \\ a_{12}a_{22} & a_{22}^2 \end{bmatrix}.$$

Our aim is to build a simulator based on a sample of size n that consists of i.i.d vectors $\mathbf{X}_1,\ldots,\mathbf{X}_n$ via subgradient method. Let $\mathbf{Y}_1,\ldots,\mathbf{Y}_m$ be the i.i.d. random vectors generated from the simulator $\mathbf{G}(\mathbf{U},\theta)$.

We consider simple mean metrics $h(\mathbf{X}) = \mathbf{X}$. Therefore the combined loss function with the regularizer coefficient c becomes:

$$\min_{\theta \in \Theta} \mathscr{W}_2^2(\hat{\nu}_{\mathbf{Y},\theta}, \hat{\nu}_{\mathbf{X}}) + c \cdot \|\mathbb{E}_{\mathbf{G}(\mathbf{U},\theta)}[\mathbf{Y}] - \mathbb{E}_{\hat{\nu}_{\mathbf{X}}}[\mathbf{X}]\|_2^2. \tag{5}$$

We note that

$$\mathbb{E}_{\mathbf{G}(\mathbf{U},\theta)}[\mathbf{Y}] = \left(\exp\left(\mu_1 + \frac{a_{11}^2 + a_{12}^2}{2}\right), \exp\left(\mu_2 + \frac{a_{22}^2}{2}\right)\right)^\top.$$

We experimented with 5000 iterations, $n = 10000$ initial sample size, 0.3 learning rate. We generated $m = 500$ random vectors at each iteration and we let $c \in \{0.0, 0.1\}$, $\mu_1^* = 1, \mu_2^* = 2, a_{11}^* = 0.5, a_{12}^* = 0.6, a_{22}^* = 0.7$ and started the search from $\mu_1^0 = 0, \mu_2^0 = 0, a_{11}^0 = 0.2, a_{12}^0 = 0.3, a_{22}^0 = 0.4$ where $c = 0.0$ corresponds to the *Unregularized* scenario and similarly a strictly positive c represents the *Regularized* version. We see that the regularized loss function shown by a blue line leads to a faster convergence; it results in a lower absolute deviation from the true parameter as depicted in Figure 1. The plot shows the absolute errors of the simulator parameters for $\{\mu_1, \mu_2, a_{11}, a_{12}, a_{22}\}$ throughout 5000 iterations, and the bottom right plot indicates the change in $\|\mathbb{E}_{\mathbf{G}(\mathbf{U},\theta)}[\mathbf{Y}] - \mathbb{E}_{\hat{\nu}_{\mathbf{X}}}[\mathbf{X}]\|_2^2$. Since we use the subgradient method that is not a descent method, the loss function does not decrease at every step, hence the oscillations of the absolute errors in Figure 1. It can also be seen that the regularized objective is more stable for a_{11}, a_{12} and a_{22}.

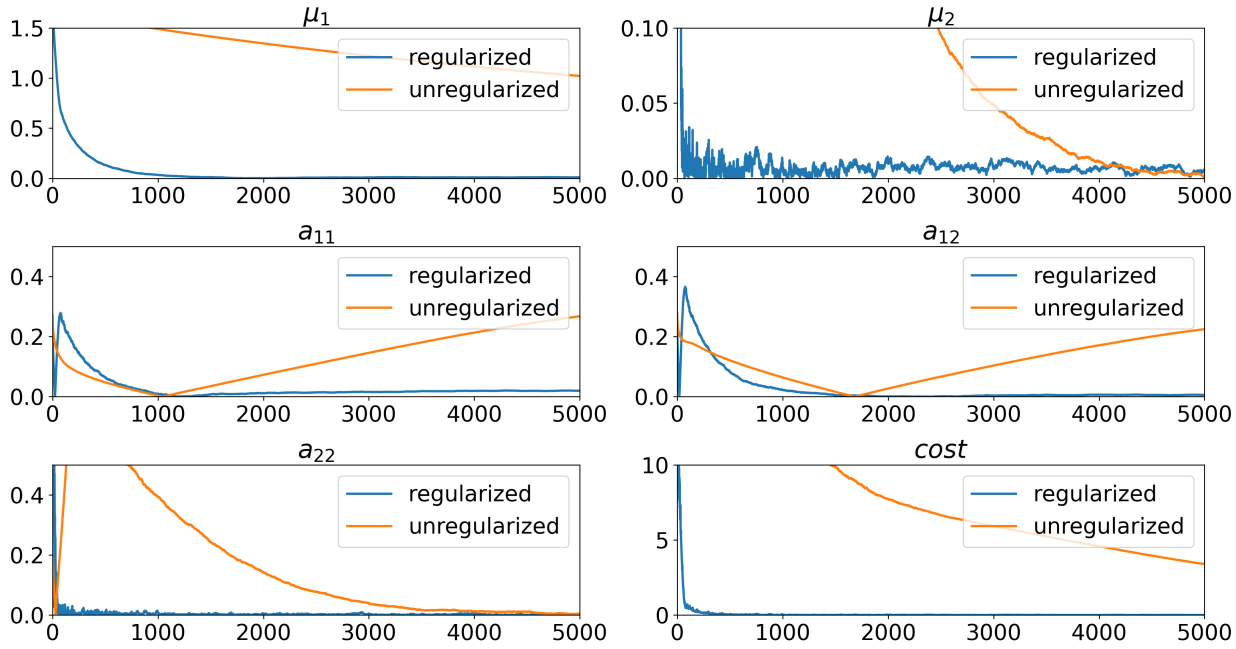


Figure 1: The absolute errors of the simulator parameters for $\{\mu_1, \mu_2, a_{11}, a_{12}, a_{22}\}$ through 5000 iterations in unregularized and regularized subgradient method.

Statistics of 10 replications in terms of $\|\mathbb{E}_{\mathbf{G}(\mathbf{U},\theta)}[\mathbf{Y}] - \mathbb{E}_{\hat{\nu}_{\mathbf{X}}}[\mathbf{X}]\|_2^2$ and $\|\mu_i^* - \hat{\mu}_i\|_2^2, i = 1, 2$ are given in Table 1, where $\hat{\mu}_i$ is the estimated parameters from the simulator. Similar to Figure 1, Table 1 shows that the regularized version yields more accurate parameter estimations.

4.2 Call Center Dataset

To evaluate the effectiveness of R-WGAN, we utilized actual arrival data gathered from a call center located in Oakland (L'Ecuyer et al. 2018). The call center operates from 8 am to midnight, 7 days a week, providing us with arrival count data per hour for each of the 16 hours it is open ($p = 16$). We obtained data for a

Table 1: Comparison of $\|\mu_i^* - \hat{\mu}_i\|_2^2$, $i = 1,2$ and $\|\mathbb{E}_{\mathbf{G}(\mathbf{U},\theta)}[\mathbf{Y}] - \mathbb{E}_{\hat{\nu}_\mathbf{X}}[\mathbf{X}]\|_2^2$ given by regularized and unregularized gradient descents with 10 replications at the end of 5000 iterations.

	$\|\mu_1^* - \hat{\mu}_1\|_2^2$		$\|\mu_2^* - \hat{\mu}_2\|_2^2$		$\|\mathbb{E}_{\mathbf{G}(\mathbf{U},\theta)}[\mathbf{Y}] - \mathbb{E}_{\hat{\nu}_\mathbf{X}}[\mathbf{X}]\|_2^2$	
	Unreg	Reg	Unreg	Reg	Unreg	Reg
mean	1.039	**4.012e-05**	3.285e-05	**1.1207e-05**	3.4167	**0.0139**
median	1.055	**4.506e-05**	2.540e-05	**4.1475e-06**	3.4499	**0.0062**
min	0.929	**6.638e-06**	7.847e-06	**2.4484e-07**	2.6038	**0.0015**
max	1.089	**6.664e-05**	7.989e-05	**3.8026e-05**	3.8127	**0.0422**
range	0.160	**6.001e-05**	7.205e-05	**3.7781e-05**	1.2088	**0.0407**

total of 2333 days, with 778 of those days allocated to the test set. To ensure the robustness of our results, we randomly shuffled the dataset 20 times, resulting in 20 distinct training and test set partitions.

For the nominal policy π_0, we defined the benchmark service rate S_j for each dimension j as the sum of the sample mean and sample standard deviation of each time interval $S_j = \bar{X}_j + \bar{\sigma}_j$ as shown in Table 2. We consider the queue length at each time step as our evaluation metrics, i.e., $h : \mathbb{R}^p \to \mathbb{R}^p$ is defined as

$$h(\mathbf{X}) = \mathbf{X}' := \left\{X_1', X_2', \ldots, X_p'\right\},$$

where $X_{j+1}' = \max(0, X_j + X_{j-1}' - S_j)$ with $X_0' = 0$, for $j = 1,2,\ldots,p$.

Table 2: Assumed service rates S, sample mean $\bar{X}$ and sample standard deviations $\bar{\sigma}$ of the Oakland Call center data.

dimension	$\bar{X}$	$\bar{\sigma}$	S
1	9.8675	5.4489	15.3164
2	12.6532	5.7889	18.4422
3	13.1928	5.7641	18.9570
4	13.0810	5.5997	18.6807
5	11.5426	5.1198	16.6625
6	11.5962	5.0214	16.6177
7	12.0771	5.0720	17.1492
8	11.4873	5.0951	16.5824
9	8.8114	4.5434	13.3548
10	8.0227	4.2932	12.3159
11	8.5696	4.1759	12.7455
12	8.9138	4.2311	13.1450
13	10.1140	4.4926	14.6066
14	10.3866	4.7907	15.1773
15	9.2576	4.6719	13.9295
16	9.1491	4.7015	13.8507

We consider the R-WGAN method:

$$\min_{G(\cdot,\theta)\in\mathscr{G}_{\mathrm{NN}}} \max_{f_w\in\mathscr{F}_{\mathrm{NN}}} \mathbb{E}[f_w(G(\mathbf{U},\theta))] - \frac{1}{n}\sum_{i=1}^{n} f_w(\mathbf{X}_i) + \frac{c}{p}\cdot\|\mathbb{E}_{\hat{\nu}_{\mathbf{Y},\theta}}[h(\mathbf{Y})] - \mathbb{E}_{\hat{\nu}_\mathbf{X}}[h(\mathbf{X})]\|_2^2.$$

We adopt the concept of correlation of past and future arrival count used by Oreshkin et al. (2016) as in Zheng et al. (2023). For an arrival count vector $\mathbf{X} = (X_1, X_2, \ldots, X_p)$, we define

$$T_{j:j+d-1} = X_j + \ldots + X_{j+d-1}, \text{ for } j \geq 1 \text{ and } d \leq p - j + 1,$$

which is the total count of arrivals in d successive time intervals starting from j-th time interval. The correlation between past and future arrival counts at the end of the j-th time interval is determined by calculating the correlation between the total arrivals in the first j periods $(T_{1:j})$ and the total arrivals in the remaining $p-j$ periods $(T_{j+1:p})$. This correlation is referred to as $\text{Corr}(\mathbf{T}_{1:j}, \mathbf{T}_{j+1:p})$, and is computed for the entire dataset as a function of j. The resulting correlation values are plotted as a solid blue line in Figure 2c.

Additionally, we generated 20 samples of size 2333 using WGAN and R-WGAN with $c = 1, 10$ trained on 20 distinct training sets. The mean correlations for each model were then plotted in Figure 2c. As the value of c increases, the correlations captured by R-WGAN diverge from the solid blue line that represents the actual correlations. Interestingly, R-WGAN with $c = 1$ (represented by the dashed purple line) was able to capture correlations more effectively than WGAN (represented by the dotted green line).

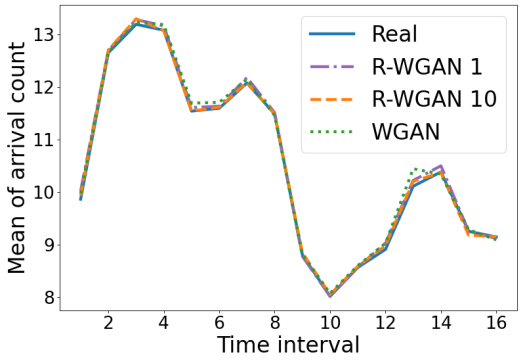
Figures 2a and 2b display the means and variances of each time interval j generated by WGAN, R-WGAN with $c = 1, 10$. As in Figure 2c, the mean results of 20 replications are shown. We can observe that the generators have preserved the actual means and variances, as evidenced by the overlapping lines.

Table 3 presents the results, which indicate that R-WGAN outperforms WGAN once again. Moreover, increasing the value of c from 1 to 10 results in even lower values of h. However, as previously discussed, high values of c can disrupt the actual correlation structure, as demonstrated in Figure 2c. Thus, there exists a trade-off between accurately estimating the actual costs and capturing the correlations between the arrival counts of different time intervals in a day.

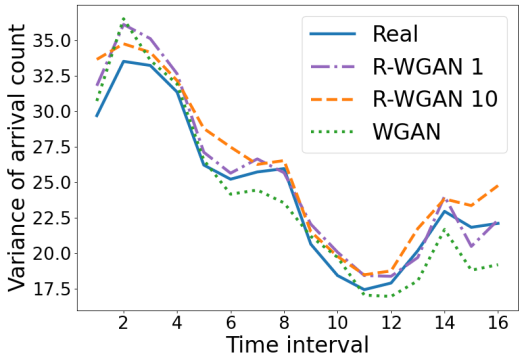
Table 3: $\|\mathbb{E}_{\hat{\nu}_{\mathbf{Y},\theta}}[h(\mathbf{Y})] - \mathbb{E}_{\hat{\nu}_{\mathbf{X}}}[h(\mathbf{X})]\|_2^2$ statistics of Oakland Call center arrival simulation with 20 replications.

Model	Epochs	Mean	Min	Max	Median
WGAN	4000	0.1275	0.0264	0.5456	0.0554
	5000	0.0555	0.0191	0.0852	0.0504
	6000	0.0294	0.0124	0.0472	0.0308
R-WGAN $c = 1$	4000	0.0204	0.0069	0.0358	0.0180
	5000	0.0185	0.0065	0.0509	0.0173
	6000	0.0189	0.0081	0.0440	0.0173
R-WGAN $c = 2.5$	4000	0.0126	0.0051	0.0206	0.0128
	5000	0.0167	0.0038	0.0379	0.0155
	6000	0.0149	0.0055	0.0230	0.0150
R-WGAN $c = 5$	4000	0.0114	0.0043	0.0219	**0.0106**
	5000	0.0135	0.0047	0.0293	0.0135
	6000	0.0132	0.0067	0.0206	0.0118
R-WGAN $c = 10$	4000	0.0121	0.0064	**0.0193**	0.0111
	5000	**0.0112**	**0.0036**	0.0219	0.0111
	6000	0.0125	0.0070	0.0221	0.0115

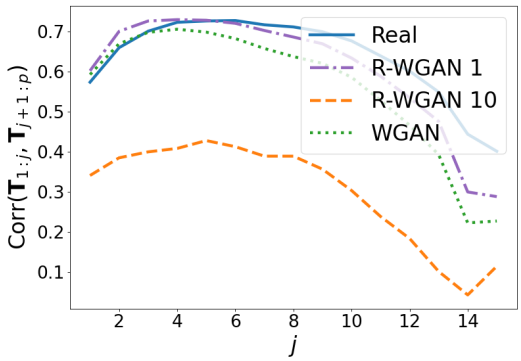
Finally, Figure 3 shows the histograms of generated samples for dimensions 1, 4, and 11. While choosing $c = 10$ over $c = 1$ may result in different outcomes in terms of costs and correlations, they both successfully capture the shape of the real samples for each dimension.



(a) Mean of arrival count in each time interval.



(b) Variance of arrival count in each time interval.



(c) The correlation of past and future arrival counts.

Figure 2: Mean performance on a call center dataset with 20 replications.

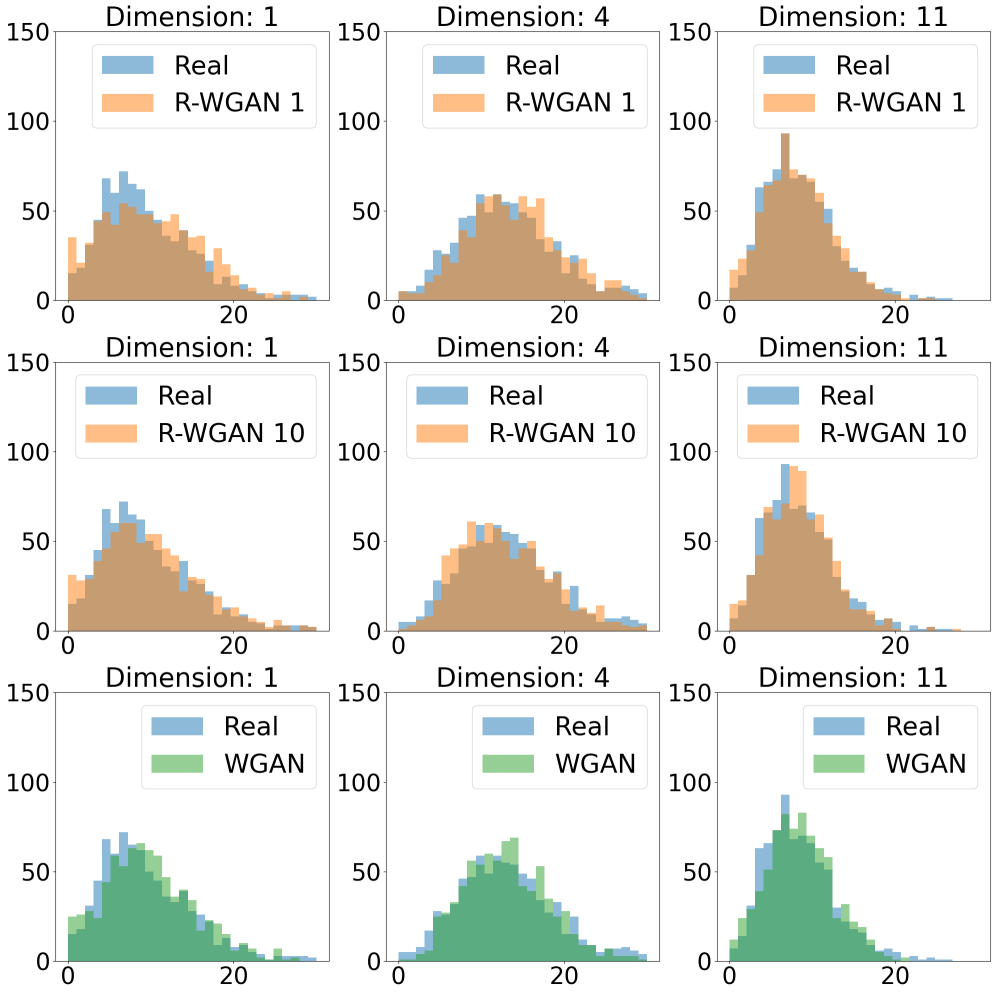


Figure 3: Oakland Call center arrival simulation, histogram of generated samples with R-WGAN $c = 1, 10$ and WGAN.

5 CONCLUSION

Many decisions in real life depend on stochastic inputs and necessitate the use of simulators in what-if scenarios under different assessment metrics. We propose a regularization framework that informs the training process of generative models of the downstream cost structure to obtain statistically better generators under the final evaluations. Our framework does not require a specific class of either cost functions or of simulators. Numerical experiments with both parametric models and WGANs (regularized GANs or R-WGANs) suggest that this framework can lead to more accurate generators with a moderate size of data, which can be appealing in the context of operational management. One current limitation of our work is the lack of theoretical analysis, especially regarding the selection process of the regularization coefficient c. We plan to leave the theoretical guarantees of the proposed regularization approach and further analysis of regularization coefficient for future research.

6 ACKNOWLEDGMENTS

We would like to thank all the anonymous reviewers for their detailed and useful comments. Their valuable suggestions were very helpful in improving our work.

REFERENCES

Arjovsky, M., S. Chintala, and L. Bottou. 2017. "Wasserstein Generative Adversarial Networks". In *Proceedings of the 34th International Conference on Machine Learning*, 214–223. Sydney, NSW, Australia: PMLR.

Bernton, E., P. E. Jacob, M. Gerber, and C. P. Robert. 2019. "On Parameter Estimation with the Wasserstein Distance". *Information and Inference: A Journal of the IMA* 8(4):657–676.

Blanchet, J., and K. Murthy. 2019. "Quantifying Distributional Model Risk via Optimal Transport". *Mathematics of Operations Research* 44(2):565–600.

Cen, W., and P. J. Haas. 2022. "NIM: Generative Neural Networks for Automated Modeling and Generation of Simulation Inputs". *ACM Transactions on Modeling and Computer Simulation* 33(3):Article number 3592790.

Cen, W., E. A. Herbert, and P. J. Haas. 2020. "NIM: Modeling and Generation of Simulation Inputs via Generative Neural Networks". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K.-H. G. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 584–595. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Deshpande, I., Y.-T. Hu, R. Sun, A. Pyrros, N. Siddiqui, S. Koyejo, Z. Zhao, D. Forsyth, and A. G. Schwing. 2019. "Max-Sliced Wasserstein Distance and Its Use for GANs". In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, edited by W. Brendel and M. R. Amer, 10640–10648. Los Alamitos, CA, USA: Institute of Electrical and Electronics Engineers Computer Society.

Dolinsky, Y., and H. M. Soner. 2014. "Martingale Optimal Transport and Robust Hedging in Continuous Time". *Probability Theory and Related Fields* 160(1-2):391–427.

Gao, R., and A. Kleywegt. 2022. "Distributionally Robust Stochastic Optimization with Wasserstein Distance". *Mathematics of Operations Research* 48(2):603–1211.

Liyanage, L. H., and J. Shanthikumar. 2005. "A Practical Inventory Control Policy Using Operational Statistics". *Operations Research Letters* 33(4):341–348.

L'Ecuyer, P., K. Gustavsson, and L. Olsson. 2018. "Modeling Bursts in the Arrival Process to an Emergency Call Center". In *Proceedings of the 2018 Winter Simulation Conference*, edited by M. Rabe, A. A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, 525–536. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Mohajerin Esfahani, P., and D. Kuhn. 2018. "Data-driven Distributionally Robust Optimization Using the Wasserstein Metric: Performance Guarantees and Tractable Reformulations". *Mathematical Programming* 171(1-2):115–166.

Nelson, B. L. 2016. "Some Tactical Problems in Digital Simulation for the Next 10 Years". *Journal of Simulation* 10(1):2–11.

Oreshkin, B., N. Réegnard, and P. L'Ecuyer. 2016. "Rate-Based Daily Arrival Process Models with Application to Call Centers". *Operations Research* 64:510–527.

Rubner, Y., C. Tomasi, and L. J. Guibas. 2000. "The Earth Mover's Distance as a Metric for Image Retrieval". *International journal of computer vision* 40(2):99.

Wang, R., P. Jaiwal, and H. Honnappa. 2020. "Estimating Stochastic Poisson Intensities Using Deep Latent Models". In *Proceedings of the 2020 Winter Simulation Conference*, edited by K.-H. G. Bae, B. Feng, S. Kim, S. Lazarova-Molnar, Z. Zheng, T. Roeder, and R. Thiesing, 596–607. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers, Inc.

Zheng, Y., Z. Zheng, and T. Zhu. 2023. "A Doubly Stochastic Simulator with Applications in Arrivals Modeling and Simulation". https://arxiv.org/abs/2012.13940, accessed 9th June.

Zhu, L., J. Shanthikumar, and M. Shen. 2008. "Solving Operational Statistics via a Bayesian Analysis". *Operations Research Letters* 36:110–116.

Zhu, T., H. Liu, and Z. Zheng. 2023. "Learning to Simulate Sequentially Generated Data via Neural Networks and Wasserstein Training". *ACM Transactions on Modeling and Computer Simulation* 33(3):1–34.

AUTHOR BIOGRAPHIES

DILARA AYKANAT is a Ph.D. student in the Department of Industrial Engineering & Operations Research at the University of California Berkeley. She has done research in simulation, machine learning, and data analytics. Her email address is dilara_aykanat@berkeley.edu.

NIAN SI is a postdoctoral principal researcher at the University of Chicago Booth School of Business. He obtained his Ph.D. in Management Science and Engineering from Stanford University, and a B.A. in Economics and a B.S. in Mathematics and Applied Mathematics both from Peking University. He has done research in applied probability and simulation. He is also interested in real-world problems arising from online platforms, including recommendation systems, online advertising, and A/B tests. His email address is niansi@chicagobooth.edu and his website is http://niansi.me.

ZEYU ZHENG is an assistant professor in the Department of Industrial Engineering & Operations Research at University of California Berkeley. He received his Ph.D. in Management Science and Engineering, Ph.D. minor in Statistics and M.A. in economics from Stanford University, and a B.S. in Mathematics from Peking University. He has done research in simulation and stochastic modeling. His email address is zyzheng@berkeley.edu and his website can be found at https://zheng.ieor.berkeley.edu/.