Adversarial Likelihood-Free Inference on Black-Box Generator

Dongjun Kim1 Weonyoung Joo1 Seungjae Shin1 Il-Chul Moon1

Abstract

Generative Adversarial Network (GAN) can be viewed as an implicit estimator of a data distribution, and this perspective motivates using GAN in the true parameter estimation under a complex black-box generative model. While previous works investigated how to backpropagate gradients through the black-box model, this paper suggests an augmented neural structure to perform a likelihood-free inference on the black-box model. Specifically, we suggest a new adversarial framework, Adversarial Likelihood-Free Inference (ALFI), with the beta-estimation network, that assumes a probabilistic model on the discriminator whose outputs are sampled from a stochastic process. Through the adversarial learning and the beta-estimation network learning, ALFI is able to find the posterior distribution of the parameter for the black-box generator model. We experimented ALFI with diverse simulation models as well as deconvolutional model, and we identified ALFI achieves the best parameter estimation accuracy with a limited simulation budget.

1. Introduction

Generative Adversarial Network (GAN) is highlighted recently for its success on estimating the data distribution, implicitly. In GAN, the generator is jointly learned with the discriminator, so the generator becomes a trainable and fine-tunable model under the adversarial learning mechanism. In contrast to such generators integrated into the adversarial framework, there has been an endeavor to incorporate the implicit data estimation capability of GAN into generators that are not integrable to the adversarial framework. For example, a simulation model can be considered as a generator that is not integrated into the GAN learning structure, and researchers are interested in the estimation on the true parameter of the simulation model with a snapshot of a validation observation (Meeds & Welling, 2015; Gutmann & Corander, 2016; Louppe et al., 2019).

Before we move on, we define a black-box generative model to clearly present our interested generator type. The black-box generative model indicates a generative model that has three properties. First, the internal process of the black-box model is designed at the previous stage, i.e. by domain experts or as another statistical model. Second, the coefficients of the internal process do not change since the coefficients are obtained by the domain specific knowledge or through a separate learning process. Third, the internal process contains the inherent stochasticity that forms an unknown distribution of the generated output. For example, continuous, discrete, and agent based simulation models can be a black-box generator that produces a stochastic trajectory of modeled states. As another example, a pre-trained and fixed de-convolutional neural network can be another practical case of the black-box generators.

The research on the black-box generator emphasizes the regeneration of the observation, which is formulated by inferring the posterior distribution $p(\theta|x_{obs})$, where $\theta \in \mathbb{R}^d$ is the model input parameter, and $x_{obs} \in \mathbb{R}^p$ is the observed data to reconstruct. However, this Bayesian inference problem suffers from the intractable likelihood $p(x_{obs}|\theta)$, where the intractability arises from the black-box nature that the nuisance variable $u \in \mathbb{R}^m$ is inaccessible in the inference procedure. Here, the nuisance variable $u$ is an unobservable control variable representing for the inherent stochasticity.

On top of the inference difficulty by the inaccessibility to $u$, the derivative of the posterior likelihood cannot be calculated, subsequently. This inability to calculate the derivative of the likelihood $p(x_{obs}|\theta)$ results in the inability to backpropagate the final signal feedback. The inability to backpropagate prevents the utilization of the black-box generators in conjunction with neural network models. To overcome the backpropagation problem, Louppe et al. (2019) proposes Adversarial Variational Optimization (AVO), an optimization algorithm of the Likelihood-Free Inference for black-box models in an adversarial setting. Louppe et al. (2019) suggests a proposal distribution $p(\theta)$ as the input parameter distribution of the black-box models to solve the backpropagation problem. However, in this solution concept, AVO assumes the posterior distribution to be the proposal distribution, which cannot be guaranteed, so AVO optimizes the parameters under a potentially incorrect distribution in its backpropagation.

This paper starts from the potential problems of AVO when only a sparse observation is given, and we suggest a new
Adversarial Likelihood-Free Inference on Black-Box Generator

adversarial framework, Adversarial Likelihood-Free Inference (ALFI), for the inference on black-box generative models. ALFI intends to expand the boundary of the adversarial framework from the implicit neural network generative models with a large amount of a dataset to the generic black-box generative models with a sparse observation, which originates from other domains. Instead of using the proposal distribution \( p_u(\theta) \), ALFI uses an auxiliary structure, called the beta-estimation network, to avoid the backpropagation problem. The beta-estimation network assumes a probabilistic model on the discriminator whose outputs are sampled from a stochastic process. We test ALFI with the baseline algorithms to estimate the underlying true parameter of the black-box generative models. As a result, we show that ALFI achieves the best estimation accuracy with a limited simulation budget.

2. Previous Research

Approximate Bayesian Computation: Approximate Bayesian Computation (ABC) (Tavaré et al., 1997; Fu & Li, 1997; Marjoram et al., 2003; Sisson et al., 2007; Beaumont et al., 2009) estimates the intractable likelihood \( p(x_{obs} | \theta) \) through Monte-Carlo simulations. The exact likelihood is given by

\[
p(x_{obs} | \theta) = \int p(x_{obs} | u, \theta) p_U(u) du = \int \delta(d(s(g(\theta, u)), s(x_{obs}))) p_U(u) du,
\]

where \( u \in \mathbb{R}^m \) is the nuisance latent variable that allows a generator \( g(\theta, u) \) to be deterministic by placing the stochasticity in \( u \); so the stochasticity of the generator is represented by an unobservable variable \( u \) (Meads & Welling, 2015; Ikonomov & Gutmann, 2019). Here, \( s : \mathbb{R}^p \to \mathbb{R}^q \) is the summary statistics from either generated instance or real-world instance, and \( d : \mathbb{R}^q \to \mathbb{R} \) is the discrepancy measure between two sets of the summary statistics. An example is a reconstruction error that can be formulated as \( (s, d) = (1, \| \cdot \|_2) \). A non-trivial example of the summary statistics is the average of the simulated results. We assume the summary statistics are embedded in the generator throughout the paper, so the output of the generator is the summary statistics of the generated raw data.

The pointwise exact likelihood in Eq. 1 cannot be estimated, since the distribution of the nuisance random variable \( p_U(u) \) and the set \( \{ u | d(g(\theta, u), x_{obs}) = 0 \} \) are generally unknown in a black-box generator. Thus, Tavaré et al. (1997) introduces an auxiliary tolerance hyperparameter \( \epsilon > 0 \) to calculate the likelihood \( p(x_{obs} | \theta) \) by the Monte-Carlo samplings. With given hyperparameter \( \epsilon \), we define a kernel function \( K_{\epsilon} \) that is a mollifier function converging to \( \delta \) as \( \epsilon \to 0 \). Then, Eq. 1 is transformed to the expectation over \( u \), which can be calculated through the Monte-Carlo evaluations.

\[
p(x_{obs} | \theta) = \lim_{\epsilon \to 0} E_{u \sim p_U}[K_{\epsilon}(u; \theta)].
\]

A case of this ABC approach is Rejection ABC (Tavaré et al., 1997; Fu & Li, 1997) that uses the boxcar kernel (Stein & Shakarchi, 2011), where the likelihood approximation becomes

\[
p(x_{obs} | \theta) \approx \frac{1}{|B^\theta_{\epsilon}(x_{obs})|} E_u[1_{B^\theta_{\epsilon}(x_{obs})}(u)],
\]

\[
B^\theta_{\epsilon}(x_{obs}) = \{ u : d(g(\theta, u), x_{obs}) < \epsilon \},
\]

with \( \epsilon \)-ball being defined on \( \epsilon > 0 \). Rejection ABC assumes \( |B^\theta_{\epsilon}(x_{obs})| \) to be invariant with \( \theta \), which leads to

\[
p(x_{obs} | \theta) \propto E_u[1_{B^\theta_{\epsilon}(x_{obs})}(u)].
\]

Rejection ABC becomes practically implemented as follows: first, we sample \( \theta \) from the prior distribution \( p(\theta) \); second, we check whether the simulation result, \( g(\theta, u) \), is within the \( \epsilon \)-ball; and third, if \( g(\theta, u) \) is within the \( \epsilon \)-ball, then we accept the parameter \( \theta \) as the sample from the posterior distribution \( p(\theta | x_{obs}) \). After total \( N \) samples are accepted, Rejection ABC estimates the posterior distribution through a density estimation method. Rejection ABC is generally used as the reference model, since the model estimates the posterior distribution accurately as \( \epsilon \to 0 \). However, Rejection ABC requires the exponentially increasing number of simulations when the parameter space on \( \theta \) is high dimensional.

Synthetic Likelihood: Synthetic Likelihood (SL) (Wood, 2010; Price et al., 2018; Ong et al., 2018) estimates the intractable likelihood under the Gaussianity assumption imposed on the summary statistics, as the below.

\[
p(x_{obs} | \theta) \approx N(x_{obs}; \hat{\mu}(\theta), \hat{\Sigma}(\theta)).
\]

Here, \( \hat{\cdot} \) is the empirical unbiased estimator based on samples. Although the SL approach does not rely on the tolerance hyperparameter \( \epsilon \) and the discrepancy measure \( d \), the SL approach assumes the explicit distribution that can be a source of Bias. It should be noted that the Gaussianity condition is valid only if the summary statistics are designed as the aggregation of i.i.d variables (Hartig et al., 2011). In spite of such a potential disadvantage, the Bayesian inference with the SL approach is recently investigated in Järvenpää et al. (2019), since the SL approach finds a high likelihood region with a relatively small number of simulations.

Bayesian Optimization Likelihood-Free Inference: With given \( d \) and \( s \), Bayesian Optimization Likelihood-Free Inference (BOLFI) (Gutmann & Corander, 2016) specifies the learning process to estimate the likelihood of \( \theta \) with a static set of simulation results. BOLFI uses the Gaussian Process Regression (GPR) with a training dataset, \( D_{disc} = \{ \theta_i, d(g(\theta_i, u_i), x_{obs}) \} \), where \( \theta_i \) are sequentially selected from the Bayesian optimization.
3.2. Two Potential Problems of Proposal Distribution

Louppe et al. (2019) mention that AVO is not suitable when the observation is a single instance. This is a valuable point, since the circumstance of having a sparse dataset becomes more common in diverse machine learning sub-fields, recently. We point out two potential problems under the sparse observation. First, Eq. 3 is transformed to

\[ V(\psi, \theta) = \log d_{\phi^*}(x) + E_{\theta \sim p_{\theta}(\theta)}[\log (1 - d_{\phi}(x))] \]

since \( p_r(x) = \delta_{x_{\text{obs}}}(x) \), and this value function is maximized when the discriminator has \( d_{\phi^*}(x) = 0 \), for all \( x \) in the support of the generator distribution \( p_g(x) \). Therefore, if the generator distribution is a continuous distribution, then the virtual training criterion \( C(\psi) = V(\psi, \phi^*) \), defined in Goodfellow et al. (2014), becomes a constant function \( C(\psi) = \log d_{\phi^*}(x_{\text{obs}}) \) with respect to the proposal parameter \( \psi \). Subsequently, the constant criterion has zero value for the derivative \( \frac{dC}{d\psi} \), which gives no feedback on the updates of the proposal parameters under the optimal discriminator \( \phi^* \). The first problem arises from the optimality of the discriminator.

On the other hand, the second problem comes from the optimal generator. The second problem of AVO is the implicit relationship between the proposal distribution \( p_\psi(\theta) \) and the posterior distribution \( p(\theta|x_{\text{obs}}) \). It is well-known (Goodfellow et al., 2014) that the optimal generator is equivalent with the real-world data distribution, i.e. \( p_g(x) = p_r(x) \), in the adversarial framework. Having assumed that the generator distribution is equal to the data distribution, i.e. \( p_g(x) = p_r(x) \), the marginal equivalence of the proposal distribution and the posterior distribution

\[ \int p(x|\theta)p_\psi(\theta)d\theta = \int p(x|\theta)p(\theta|x_{\text{obs}})d\theta \]

(4)

holds, where the left-hand-side is the generator distribution \( p_g(x) \) and the right-hand-side is the real-world data distribution \( p_r(x) \). However, the marginal equivalence does not guarantee the equivalence between the proposal \( p_\psi(\theta) \) and the posterior \( p(\theta|x_{\text{obs}}) \), see Supplementary Section C.

Figure 1 demonstrates an illustrative example that the implicit proposal distribution in AVO fails to match the posterior distribution. In particular, the multi-modality of the posterior distribution is not captured in the proposal distribution, which demonstrates that the proposal approach, under the adversarial framework, suffers from the mode collapse.

Moreover, the discriminator parameter converges to the optimal parameter \( \phi^* \) before the proposal distribution learns the optimal parameter \( \theta \) for all black-box models in Section 5. Consequently, AVO with a single observation cannot find the optimal parameter in our experiments in Section 5.
4. Adversarial Likelihood-Free Inference

This paper proposes Adversarial Likelihood-Free Inference (ALFI), a Bayesian inference that can operate in the adversarial setting with a black-box generator. Under the assumption of using a neural network or other types of black-box generators, such as stochastic simulations, ALFI suggests a sampling-based inference on the posterior \( p(\theta|x_{\text{obs}}) \) by the Metropolis-Hastings algorithm.

4.1. Motivation of Developing ALFI

The original GAN utilizes a neural network as a generator \( g_\omega(\theta) \) to model the data distribution \( p_r(x) \) with a prior distribution \( p(\theta) \) that is often assumed to be the standard Gaussian distribution. Here, the neural network-based generator has no stochasticity besides of \( p(\theta) \), so the nuisance stochastic variable \( u \) does not exist in the original GAN. Instead, the generator parameters, \( \omega \), can be optimized to map \( p(\theta) \) to \( p_r(x) \) in the GAN literature.

This paper hypothesizes the expansion of \( g_\omega(\theta) \). There could be a stochastic generator, such as various simulations with noises, that has a meaningful noise nuisance variable \( u \). In particular, these black-box generators may have a fixed model structure as a constant \( \omega \); i.e. a stochastic simulator \( g(\theta, u|\omega) \) modeling physical phenomena with a fixed coefficient \( \omega \) and a random noise \( u \). This hypothesis leads that \( g(\theta, u|\omega) \) can only produce \( p_r(x) \) by inferring the input distribution \( p(\theta) \). This is different from the original GAN because this hypothesis requires the shape of \( p(\theta) \) to be learned whereas \( p(\theta) \) in GAN is fixed as the standard Gaussian distribution.

AVO provides a method of parametrizing the prior distribution by an implicit proposal distribution \( p_\psi(\theta) \). This parameterization enables AVO to attain \( \nabla_\psi V(\psi, \phi) \) by the REINFORCE algorithm. Without the prior distribution, the input parameter \( \theta \) is not optimizable with the gradient descent method. In spite that AVO introduced the joint consideration of the black-box model with the adversarial framework, AVO has two major disadvantages: one from the optimality of the discriminator, and the other from the optimality of the generator. Therefore, we need a new methodology for the Bayesian inference in the adversarial setting.

4.2. Inference Procedure in ALFI

Figure 2-(c) depicts the network structure of ALFI, which consists of three major components: a black-box generator \( g: \mathbb{R}^d \times \mathbb{R}^m \rightarrow \mathbb{R}^q \), a discriminator network \( d_\phi : \mathbb{R}^q \rightarrow [0, 1] \), and a beta-estimation network \( b_\psi \).

The beta-estimation network \( b_\psi \) estimates a stochastic process where the random variable of the model input parameter \( \theta \), is assumed to follow the beta distribution. In other words, a surrogate function with a slice on \( \theta \) is the beta distribution, and the beta-estimation network estimates the response curve of \( d_\phi(g(\theta, u)) \) by an amortized inference on the beta distribution shape parameters \( b_\psi^r(\theta) \) and \( b_\psi^s(\theta) \). The estimated surrogate function is used to calculate the likelihood ratio \( \frac{p(x_{\text{obs}}|\theta)}{p(x_{\text{obs}}|\theta)} \), as described in Section 4.3. Then, the Metropolis-Hastings algorithm samples the next input parameters from the stationary distribution after iterations. It should be noted that the response curve of the stochastic process \( d_\phi(g(\theta, u)) \) converges to the static response curve with an optimal discriminator \( d_\phi^*(g(\theta, u)) \).

We emphasize that ALFI does not suffer from the negative aspects of AVO. First, AVO with optimal discriminator hinders the proposal distribution to find the optimal parameter \( \psi^* \), whereas the optimal discriminator in ALFI stabilizes the underlying response curve \( d_\phi(g(\theta, u)) \) into \( d_\phi^*(g(\theta, u)) \), and this stabilized response curve \( d_\phi^*(g(\theta, u)) \) is now a static stochastic process. After obtaining the discriminator optimality, the beta-estimation network learns the static stochastic process. Therefore, the optimal discriminator in ALFI helps the network to find the optimal parameter \( \psi^* \). In addition, the second problem in AVO is irrelevant in ALFI, since ALFI directly estimates the posterior distribution without introducing the proposal distribution.

4.3. Likelihood Ratio Approximation

This section calculates the approximation of the likelihood ratio \( \frac{p(x_{\text{obs}}|\theta)}{p(x_{\text{obs}}|\theta)} \). To obtain the approximation, ALFI adopts an explicit assumption on the shape of distribution. Even if the synthetic likelihood approach, in Eq. 2, suffers from the bias that comes from the explicit assumption, the explicit assumption enjoys the advantage that it significantly reduces the number of required simulations.

ALFI introduces a beta distribution assumption on the discriminator output. However, the output of the discriminator can be expanded to be multi-dimensional or unbounded, and such cases will require a different distribution assumption, other than the beta distribution. Even in such cases, ALFI provides the fundamental structure to infer such distributions, see Supplementary Section A.

The explicit beta assumption enables calculating the intractable likelihood, under the following arguments. The likelihood \( p(x_{\text{obs}}|\theta) \) is the marginal distribution that integrates out the joint distribution \( p(x_{\text{obs}}, u|\theta) \) by nuisance variable \( u \).

\[
p(x_{\text{obs}}|\theta) = \int p(x_{\text{obs}}|\theta, u) p_U(u) du \tag{5}
\]

The conditional likelihood \( p(x_{\text{obs}}|\theta, u) \), given \( u \), is cal-
Adversarial Likelihood-Free Inference on Black-Box Generator

(a) GAN
(b) AVO
(c) ALFI

Figure 2. Network structure comparisons for GAN, AVO, and ALFI. (a) Vanilla GAN generator is replaced with a generic black-box generator, the generator internal coefficients $\omega$ are fixed throughout the adversarial learning. (b) AVO suggests a proposal network $p_\psi(\theta)$ that learns the prior distribution of model parameter $\theta$. AVO learns the proposal network with variational optimization, which enables the backpropagation through generator via REINFORCE algorithm. (c) ALFI suggests a beta-estimation that forms a surrogate function for the discriminator output. ALFI infers the posterior distribution by Metropolis-Hastings algorithm with the acceptance ratio, calculated by the beta-estimation output.

At this moment, we define the random variable $Y_\theta : \mathbb{R}^m \to \mathbb{R}$ to be $Y_\theta(u) = d_\beta(g(\theta, u))$, which is the slice of the stochastic process $d_\alpha(g(\theta, u))$. Now, the change of variable, from $u \in \mathbb{R}^m$ to $y \in \mathbb{R}$ (see Supplementary Section A), transforms the intractable distribution $p_U(u)$ into the product of the 1-dimensional distribution $p_{\psi}(y)$ with an intractable auxiliary term $c$,

$$p(x_{\text{obs}}|\theta) = \lim_{\epsilon \to 0} \frac{1}{2\epsilon} \int_0^1 \int_{\mathbb{R}} \int_{\mathbb{R}} p_U(u) \, dy \, du,$$

(6)

Assuming that the boxcar kernel is the mollifiers to approximate the delta measure with the mollifiers $p(|d_\alpha(g(\theta, u)) - d_\gamma(x_{\text{obs}})| < \epsilon)$ whose limit is the delta distribution. Then, Eq. 5 is transformed to

$$p(x_{\text{obs}}|\theta) = \lim_{\epsilon \to 0} \int p(|d_\alpha(g(\theta, u)) - d_\gamma(x_{\text{obs}})| < \epsilon) p_U(u) \, du.$$

(7)

Unlike in Rejection ABC, where the measure of $\epsilon$-ball was intractable, in ALFI, the $\epsilon$-ball is centered at the 1-dimensional point $d_\gamma(x_{\text{obs}})$, and the measure of the $\epsilon$-ball becomes exactly $2\epsilon$.

$$p(x_{\text{obs}}|\theta) = \lim_{\epsilon \to 0} \frac{1}{2\epsilon} \int 1_{B_\epsilon(d_\gamma(x_{\text{obs}}))}(d_\alpha(g(\theta, u))) p_U(u) \, du.$$

(8)

Here, the nuisance variable $u$ is a $m$-dimensional variable $(u_1, ..., u_m)$. The auxiliary term $c$ in Eq. 7 is intractable, since the derivative $\frac{d}{du_1} d_\gamma(g(\theta, u))$ is unattainable, and the integrals over $u_2, ..., u_m$ are irreducible.

In spite that $c$ is intractable, we are able to obtain the piecewise continuity of $c$ under an assumption. The assumption argues that the discriminator of the black-box generator, as a function of $u_1$ that is the first argument of $u$, is differentiable almost everywhere with piecewise continuous non-zero derivative. We limit ALFI on black-box models having the above assumption, see details in Supplementary Section B.

An explicit distribution assumption is imposed on the random variable $Y_\theta$ to follow the beta distribution, with the shape parameters being estimated by the beta-estimation network $b_\psi$. If we denote $f$ to be the beta probability density function, the likelihood becomes

$$p(x_{\text{obs}}|\theta) = \lim_{\epsilon \to 0} \frac{1}{2\epsilon} \int 1_{B_\epsilon(d_\gamma(x_{\text{obs}}))}(y) f(y; b_\psi^\alpha(\theta), b_\psi^\beta(\theta)) c(y|\theta) \, dy.$$

(9)

By taking the limit $\epsilon \to 0$, we have the final likelihood
estimation formula
\[ p(x_{\text{obs}}|\theta) = f(d_\phi(x_{\text{obs}}); b_\psi^a(\theta), b_\psi^b(\theta))c(d_\phi(x_{\text{obs}})|\theta). \]

Then, the likelihood ratio between the parameters \( \theta \) and \( \theta' \) is given as
\[ \frac{p(x_{\text{obs}}|\theta')}{p(x_{\text{obs}}|\theta)} = \frac{f(d_\phi(x_{\text{obs}}); b_\psi^a(\theta'), b_\psi^b(\theta'))c(d_\phi(x_{\text{obs}})|\theta')}{f(d_\phi(x_{\text{obs}}); b_\psi^a(\theta), b_\psi^b(\theta))c(d_\phi(x_{\text{obs}})|\theta)}. \]

This likelihood ratio is intractable, but the piecewise continuity of \( c \) assures that the ratio term \( \frac{c(d_\phi(x_{\text{obs}})|\theta')}{c(d_\phi(x_{\text{obs}})|\theta)} \) is close to 1, if the parameters \( \theta \) and \( \theta' \) are close. Therefore, if the proposal distribution \( q(\theta'|\theta) \) in the Metropolis-Hastings algorithm is the Gaussian distribution \( N(\theta, \sigma^2I) \), with sufficiently small \( \sigma \), then the next sample \( \theta' \) from the proposal of the Metropolis-Hastings algorithm will be close to \( \theta \) for approximating the likelihood ratio by
\[ \frac{p(x_{\text{obs}}|\theta')}{p(x_{\text{obs}}|\theta)} \approx \frac{f(d_\phi(x_{\text{obs}}); b_\psi^a(\theta'), b_\psi^b(\theta'))}{f(d_\phi(x_{\text{obs}}); b_\psi^a(\theta), b_\psi^b(\theta))} \quad (8) \]

Note that the posterior distribution \( p(\theta|x_{\text{obs}}) \) is the stationary distribution of the Metropolis-Hastings algorithm with ratio Eq. 8, if the function \( c \) is constant, and the random variable \( Y_\theta \) indeed follows the beta distribution, for all \( \theta \). See Supplementary Section B for details.

We argue that the beta distribution assumption on \( Y_\theta \) harmonizes with diverse black-box models by two reasons. First, the random variable is bounded on \([0,1]\), which requires the explicit distribution to be supported on \([0,1]\). Second, the reduced stochasticity by well-designed summary statistics leads the samples from \( Y_\theta \) to be concentrated on its mean, \( E_\theta(Y_\theta) \), as the variability of the summary statistics directly influence on the variance of the random variable \( Y_\theta(u) = d_\phi(g(\theta, u)) \). ALFI assumes that the black-box model embeds the well-designed summary statistics, and the model generates the output as the \( q \)-dimensional summary statistics of the \( p \)-dimensional raw output. The examples of such summary statistics are the average, the standard deviation, the quantiles, and the correlation coefficients. The concept of using the summary statistics in black-box generators is coherent with the Bayesian concept, where the Bayesian concept translates the distributions (Gaussian) into few summary statistics (mean and variance) to perform its inference.

### 4.4. Algorithm

ALFI utilizes the discriminator loss function of (Goodfellow et al., 2014).

\[ L_d(\phi) = -E_{x \sim p_c(x)}[\log d_\phi(x)] - E_{\theta,u}[\log (1 - d_\phi(g(\theta, u))]. \quad (9) \]

The expectation on \( \theta \) is calculated over the \( n \) samples selected by the Metropolis-Hastings algorithm after \( m \) iterations, where \( m \) is the exploration-exploitation hyperparameter that controls the speed of convergence to the stationary distribution of the Metropolis-Hastings algorithm, see Algorithm 1. As the neural network \( b_\psi \) converges to the static stochastic process \( d_{\psi^*}(g(\theta, u)) \), the sample expectation on the above \( n \) samples converges to the unbiased expectation of the \( n \) samples from the stationary distribution \( \pi_{\psi^*}(\theta) \), regardless of the choice of \( m \).

For each iteration, we have a training dataset \( D_\theta = \{\theta_1, d_\phi(g(\theta_1, u_1))\} \) to learn the beta-estimation network \( b_\psi \). The loss function is the negative log-likelihood of \( D_\theta \), with \( B \) referring to the beta function.

\[ L_b(\psi) = -E_{\theta,u}[(b_\psi^a(\theta) - 1) \log d_\phi(g(\theta, u)) \quad (10) \]

\[ + (b_\psi^b(\theta) - 1) \log (1 - d_\phi(g(\theta, u))) - \log B(b_\psi^a(\theta), b_\psi^b(\theta))]. \]

### 5. Experiments

#### 5.1. Illustrative Example

The illustrative black-box model elucidates the mechanism of ALFI. The below is a suggested illustrative model.

\[ g(\theta, u) = \sin(2\pi \theta) + u, \]

where \( u \sim N(0, 0.03^2) \). The true parameter \( \theta^* \) is 0.25, and consequently the observation \( x_{\text{obs}} \) is 1.

Figure 4-(a) argues two points. First, the samples of ALFI, blue dots, with mini-batch size 100, are concentrated around the true parameter \( \theta^* \). Second, the surrogate function, blue line, estimates the response curve \( d_{\psi^*}(g(\theta, u)) \).
Adversarial Likelihood-Free Inference on Black-Box Generator

(a) Beta Approximation
(b) Posterior Approximation
(c) Metropolis-Hastings samples

Figure 4. Illustrative model. (a) The estimated beta distribution fits to $\mathcal{N}_b(x_{obs})$ well around the true parameter. The Metropolis-Hastings samples are accumulated near the true parameter. (b) The kernel density estimation of the samples from ALFI matches with the rejection ABC density. (c) The samples are accumulated near the true parameter within 5000 simulation executions.

Table 1. Simulation models used in experiments

| SIMULATION MODELS | INPUT DIM.(d) | OUTPUT DIM.(s) | SIMULATION TYPE |
|-------------------|--------------|----------------|----------------|
| TUMOR             | 2            | 8              | ODE            |
| LORENZ            | 3            | 24             | ODE            |
| SUSCEPTIBLE       | 2            | 18             | ODE            |
| INFECTIOUS        | 2            | 18             | ODE            |
| RECOVERED         | 2            | 49             | PDE            |
| NEW PRODUCT       | 2            | 15             | SYSTEM DYNAMICS |
| ADOPTION          | 3            | 16             | DISCRETE TIME-SERIES |
| M/G/1             | 2            | 20             | AGENT BASED    |

Figure 5. Performance comparison with ALFI and five baselines

5.2. Simulations as Black-box Models

Considering simulations as black-box models, we compare the performance of the following baseline likelihood-free inference algorithms: Rejection-ABC (Tavaré et al., 1997), Monte-Carlo Markov-Chain (MCMC) ABC (Marjoram et al., 2003), BOLFI (Gutmann & Corander, 2016), and AVO (Louppe et al., 2019). Here, the performance of $-\log \|\hat{\theta} - \theta^*\|_2$ is the negative log euclidean distance between the true parameter $\theta^*$ and the posterior mode $\hat{\theta}$. A single observation $x_{obs}$ is the average of 100 simulation execution results with $\theta = \theta^*$, to minimize the stochasticity driven by the nuisance variable $u$.

Table 1 presents the simulation models used in the experiments, including continuous, discrete, and agent-based simulations, see Supplementary Section D for details on simulation models. We apply the Runge-Kutta 4-th order method (Leader, 2004) to solve the Ordinary Differential Equations (ODEs), and the finite difference method (Iserles, 2009) to solve the time-independent Partial Differential Equation (PDE).

Figure 5 summarizes the performances of algorithms. The performance is estimated by randomly chosen 30 different set of true parameters $\{\theta^*_j\}_{j=1}^{30}$. For each true parameter $\theta^*_j$, we estimate the posterior mode $\hat{\theta}_j$ with the same simulation time budgets. We choose $\epsilon$ adaptively to accept 0.1% of the simulation runs in Rejection ABC. We use the released code of AVO in AVO Gaussian, where the proposal distribution $p_\psi(\theta)$ is the Gaussian distribution $\mathcal{N}(\theta; \mu_{\psi}, \Sigma_{\psi})$. AVO NN alleviates the Gaussinity assump-
Adversarial Likelihood-Free Inference on Black-Box Generator

Figure 6. M/G/1 model (a) ALFI converges to the plateau within 100 iterations (b) ALFI finds the true parameter better than Rejection ABC with the same simulation budget.

Figure 7. Poisson second-order PDE model (a) Observations are equally distributed in the simulated region (b,c) ALFI regenerates the observation better than AVO.

Figure 8. ALFI and AVO regeneration results with masked observation. ALFI is robust on the magnitude of masking noise and masking density. In (b), the injected noises differ for every generating images, and these varying noises act as a nuisance variable for the DCGAN generator.

5.3. GAN Generator as a Black-box Model

We design an experiment by treating the DCGAN (Radford et al., 2015) generator as a black-box model. Specifically, we use a MNIST image generator $g(\theta, u | \omega)$, with the pre-trained fixed generator coefficient $\omega$, throughout the ALFI learning process. Our experiment is designed to regenerate a true image from its masked image by estimating the nearest latent embedding of the masked image. Figure 8 shows a true sample and its masked observation as a pair. Our question is whether ALFI or AVO NN can infer a vector input of the DCGAN generator to regenerate an image similar to the true image when its validation dataset is given as a masked and corrupted observation. Figure 8 concludes that ALFI is able to generate an image more similar to the true image than AVO NN can generate.

6. Conclusions

We are motivated to develop ALFI because of the problems that AVO has. In this aspect, ALFI estimates the true parameter more accurately than AVO. The contribution of ALFI lies at two different layers. The first layer of contribution is the introduction of a new Likelihood-Free Inference on a black-box generator in the adversarial framework. The second layer of contribution is the practical application of completing high fidelity simulation models by finding the best calibrated parameters given a validation dataset. Moreover, it should be noted that the utilization of the black-box generative model frequently occurs in the neural network communities where a few examples are given and where a pre-trained and fixed generator is used.
References

Beaumont, M. A., Cornuet, J.-M., Marin, J.-M., and Robert, C. P. Adaptive approximate bayesian computation. *Biometrika*, 96(4):983–990, 2009.

Fu, Y.-X. and Li, W.-H. Estimating the age of the common ancestor of a sample of dna sequences. *Molecular biology and evolution*, 14(2):195–199, 1997.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Gutmann, M. U. and Corander, J. Bayesian optimization for likelihood-free inference of simulator-based statistical models. *The Journal of Machine Learning Research*, 17(1):4256–4302, 2016.

Hartig, F., Calabrese, J. M., Reineking, B., Wiegand, T., and Huth, A. Statistical inference for stochastic simulation models—theory and application. *Ecology letters*, 14 (8):816–827, 2011.

Ikonomov, B. and Gutmann, M. U. Robust optimisation monte carlo. *arXiv preprint arXiv:1904.00670*, 2019.

Iserles, A. *A first course in the numerical analysis of differential equations*. Number 44. Cambridge university press, 2009.

Järvenpää, M., Gutmann, M., Vehtari, A., and Marttinen, P. Parallel gaussian process surrogate method to accelerate likelihood-free inference. *arXiv preprint arXiv:1905.01252*, 2019.

Leader, J. J. *Numerical Analysis and Scientific Computation*. Number Sirsi) i9780201734997. 2004.

Louppe, G., Hermans, J., and Cranmer, K. Adversarial variational optimization of non-differentiable simulators. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1438–1447, 2019.

Marjoram, P., Molitor, J., Plagnol, V., and Tavaré, S. Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328, 2003.

Meeds, T. and Welling, M. Optimization monte carlo: Efficient and embarrassingly parallel likelihood-free inference. In *Advances in Neural Information Processing Systems*, pp. 2080–2088, 2015.

Ong, V. M.-H., Nott, D. J., Tran, M.-N., Sisson, S. A., and Drovandi, C. C. Likelihood-free inference in high dimensions with synthetic likelihood. *Computational Statistics & Data Analysis*, 128:271–291, 2018.

Price, L. F., Drovandi, C. C., Lee, A., and Nott, D. J. Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*, 27(1):1–11, 2018.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Sisson, S. A., Fan, Y., and Tanaka, M. M. Sequential monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 104(6):1760–1765, 2007.

Stein, E. M. and Shakarchi, R. *Functional Analysis: Introduction to Further Topics in Analysis*, volume 4. Princeton University Press, 2011.

Tavaré, S., Balding, D. J., Griffiths, R. C., and Donnelly, P. Inferring coalescence times from dna sequence data. *Genetics*, 145(2):505–518, 1997.

Wood, S. N. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466(7310):1102, 2010.
Appendix for Adversarial Likelihood-Free Inference on Black-Box Generator

Anonymous

A. Likelihood Approximation

The likelihood is calculated as following equalities:

\[
p(x_{\text{obs}}|\theta) = \int p(x_{\text{obs}}|\theta, u)p_U(u) \, du
= \int p(g(\theta, u) = x_{\text{obs}})p_U(u) \, du
= \int \delta(g(\theta, u) = x_{\text{obs}})p_U(u) \, du
= \lim_{\epsilon \to 0} \int p(\phi(g(\theta, u)) - \phi(x_{\text{obs}})) < \epsilon p_U(u) \, du
= \lim_{\epsilon \to 0} \int p(\phi(g(\theta, u)) \in B(\phi(x_{\text{obs}}))p_U(u) \, du
= \lim_{\epsilon \to 0} \frac{1}{2\epsilon} \int 1_B(\phi(x_{\text{obs}})) (\phi(g(\theta, u))p_U(u) \, du
\]

The last equation is unattainable, since the probability distribution \( p_U(u) \) is not known. Therefore, we use change of variable with the transformation \( h : \mathbb{R}^m \rightarrow \mathbb{R}^m \) defined as \( h(u_1, ..., u_m) = (\phi(g(\theta, u_1), u_2, ..., u_m) \). With an assumption that the transformation is invertible, we have

\[
p(x_{\text{obs}}|\theta) = \lim_{\epsilon \to 0} \frac{1}{2\epsilon} \int 1_B(\phi(x_{\text{obs}})) (\phi(g(\theta, h^{-1}(y))) p_U(h^{-1}(y))|J(h^{-1})| dy
= \lim_{\epsilon \to 0} \frac{1}{2\epsilon} \int 1_B(\phi(x_{\text{obs}})) (\phi(g(\theta, h^{-1}(y))) p_U(h^{-1}(y))|J(h^{-1})| dy
\]

By the Inverse Function Theorem, we have

\[
\lim_{\epsilon \to 0} \frac{1}{2\epsilon} \int 1_B(\phi(x_{\text{obs}})) (y_1)p_U(h^{-1}(y))|J(h^{-1})| dy
= \lim_{\epsilon \to 0} \frac{1}{2\epsilon} \int 1_B(\phi(x_{\text{obs}})) (y_1)p_U(h^{-1}(y))|J(h^{-1})| dy
\]

Since the transformation is identity except at the first argument, the Jacobian is \( |Jh| = \left| \frac{d}{du_1} \phi(g(\theta, u)) \right| \), which leads

\[
p(x_{\text{obs}}|\theta) = \lim_{\epsilon \to 0} \frac{1}{2\epsilon} \int 1_B(\phi(x_{\text{obs}}))(y_1)p_U(h^{-1}(y))|J(h^{-1})| dy
= \lim_{\epsilon \to 0} \int 1_B(\phi(x_{\text{obs}}))(y_1) \int_{\mathbb{R}} \ldots \int_{\mathbb{R}} p_U(h^{-1}(y)) \left| \frac{d}{du_1} \phi(g(\theta, u)) \right|^{-1} dy_2 \ldots dy_m dy_1
\]

The integral of probability distribution \( p_U(h^{-1}(y)) \) over \( y_i \), \( i = 2, ..., m \), is one, thus we have the following equation.

\[
p(x_{\text{obs}}|\theta) = \lim_{\epsilon \to 0} \int 1_B(\phi(x_{\text{obs}}))(y_1) \int_{\mathbb{R}} \ldots \int_{\mathbb{R}} p_U(h^{-1}(y)) \left| \frac{d}{du_1} \phi(g(\theta, u)) \right|^{-1} dy_2 \ldots dy_m dy_1
= \lim_{\epsilon \to 0} \int 1_B(\phi(x_{\text{obs}}))(y_1) \int_{\mathbb{R}} \ldots \int_{\mathbb{R}} p_U(h^{-1}(y)) \left| \frac{d}{du_1} \phi(g(\theta, u)) \right|^{-1} dy_2 \ldots dy_m dy_1
= \lim_{\epsilon \to 0} \int 1_B(\phi(x_{\text{obs}}))(y_1) \int_{\mathbb{R}} \ldots \int_{\mathbb{R}} p_U(h^{-1}(y)) \left| \frac{d}{du_1} \phi(g(\theta, u)) \right|^{-1} dy_2 \ldots dy_m dy_1
= \lim_{\epsilon \to 0} \int 1_B(\phi(x_{\text{obs}}))(y_1) \int_{\mathbb{R}} \ldots \int_{\mathbb{R}} p_U(h^{-1}(y)) \left| \frac{d}{du_1} \phi(g(\theta, u)) \right|^{-1} dy_2 \ldots dy_m dy_1
= \lim_{\epsilon \to 0} \int 1_B(\phi(x_{\text{obs}}))(y_1) \int_{\mathbb{R}} \ldots \int_{\mathbb{R}} p_U(h^{-1}(y)) \left| \frac{d}{du_1} \phi(g(\theta, u)) \right|^{-1} dy_2 \ldots dy_m dy_1
\]

Here, the auxiliary intractable term \( c \) is defined as

\[
c(y_1|\theta) = \int_{\mathbb{R}} \ldots \int_{\mathbb{R}} \left| \frac{d}{du_1} \phi(g(\theta, u)) \right|^{-1} du_2 \ldots du_m.
\]

Therefore, we yield

\[
p(x_{\text{obs}}|\theta) = \lim_{\epsilon \to 0} \frac{1}{2\epsilon} \int 1_B(\phi(x_{\text{obs}}))(y_1)p_U(h^{-1}(y))c(y_1|\theta) dy
= p_{\phi}(\phi(x_{\text{obs}}))c(\phi(x_{\text{obs}})|\theta)
\]

If the random variable \( Y_0 \) follows the beta distribution with shape parameters \( \alpha \) and \( \beta \), the likelihood becomes

\[
p(x_{\text{obs}}|\theta) = Beta(\phi(x_{\text{obs}}); \alpha, \beta)c(\phi(x_{\text{obs}})|\theta)
\]

Note that above equalities, with little modifications, still hold when the discriminator output is multi-dimensional or unbounded.
Appendix: Adversarial Likelihood-Free Inference on Black-Box Generator

B. Properties of Auxiliary Function $c$

The intractable function $c$ is piecewise continuous, if the discriminator output, as a function of $u_1$, is differentiable almost everywhere with piecewise continuous non-zero derivative, which guarantees the existence of the piecewise continuous inverse $|d_{\pi_1}d_c(g(\theta, u))|$. The differentiability is a plausible assumption in the black-box generative model community by the following arguments. First, in case where the discriminator is defined on discrete $u_1$, the discriminator is not differentiable with respect to $u_1$. However, if not all the dimension of nuisance variable are discrete, i.e. if $u_k$ is continuously defined for $k \neq 1$, then we switch the variables $u_1$ and $u_k$ to make the discriminator be a function of a continuous variable $u_k$. There is virtually no black-box generative models with nuisance variables consisted solely of discrete variables. Furthermore, the discrete sampling could be represented as a set of continuous nuisance variable $u$, since the discrete sampling is equivalent with the consecutive operation of coin toss, where an the head is sampled if the randomly selected number on $[0,1]$ exceeds the bernoulli coefficient $p$, and the tail is sampled if the random generating number is less than $p$. Therefore, we consider the nuisance variable $p_U(u)$ is a continuous distribution.

Second, having said that the discriminator, as a function of $u_1$, is continuously defined, it is not a tight constraint for generator to be differentiable with respect to $u_1$, since most models developed in science and engineering are expected to have regularity with respect to the inputs, where regularity in science context (Gilbarg & Trudinger, 2015) refers for either continuity or differentiability. Without an appropriate level of regularity, the models are regarded as a highly chaotic system. Third, the discriminator is consisted of the matrix multiplications and non-linear activation functions, which is differentiable almost everywhere except at the measure zero set with respect to the Lebesgue measure. Therefore, considerable proportion of black-box models satisfy the assumption on $c$.

When the intractable function $c$ is constant and the random variable $Y_0$ indeed follows the beta distribution, then the following equation

$$p(x_{obs} | \theta') = \frac{f(d_c(x_{obs}); \beta_{\psi}^\gamma (\theta'), \beta_{p_x}^\beta (\theta'))}{f(d_c(x_{obs}); \beta_{\psi}^\alpha (\theta'), \beta_{p_x}^\beta (\theta'))}$$

holds, with $f$ denotes for the beta distribution pdf. Since the Metropolis-Hastings algorithm is a Markov process that satisfies the detailed balance equation with the ergodicity, the Metropolis-Hastings algorithm asymptotically reaches a unique stationary distribution $\pi(\theta)$ (Robert & Casella, 2013). The stationary distribution $\pi(\theta)$ is the posterior distribution $p(\theta | x_{obs})$, in the case where ALFI assumes prior to be uniform, since the acceptance rate is defined as the likelihood ratio.

C. Discussion on AVO with few observation

ALFI is not suffered from the problems in AVO. In AVO, with a single observation, the discriminator learns to classify the true observation with the generated samples too fast, and the proposal distribution is not guided by the loss function, since the loss function with the optimal discriminator is constant with respect to the proposal parameters $\psi$, if the generator distribution is a continuous distribution. Furthermore, ALFI infers the posterior distribution directly, without considering the proposal distribution. This enables ALFI to avoid the problems arisen by adopting proposal distribution.

Here, it is worth to note that the generator distribution inherits the proposal distribution type: if the proposal distribution $p_{\psi}(\theta)$ is continuous, then so is the generator distribution $p_{\psi}(x)$, and if $p_{\psi}(\theta)$ is singular, then so is $p_{\psi}(x)$. The next argument gives background on the above inheritance property. The generator distribution is the marginalized distribution of the proposal distribution, given by,

$$p_{\psi}(x) = \int p(x | \theta) p_{\psi}(\theta) d\theta.$$ 

The likelihood term then yields

$$p_{\psi}(x) = \int p(x | \theta) p_{\psi}(\theta) d\theta$$

$$= \int_{\theta} \int_{u} \delta(d(g(\theta, u), x)) p_{\psi}(u) d\psi(\theta) d\theta$$

$$= p(\{ (\theta,u) | d(g(\theta, u), x) = 0 \}),$$

where the probability distribution $p$, in Eq. 1, is the joint probability on $\theta$ and $u$, with mean-field probability density function. This leads that if $p_{\psi}(\theta)$ is continuous, then $p_{\psi}(x)$ is continuous (since we already assumed that $p_{\psi}(u)$ is continuous by considering discrete random selection as a consecutive continuous procedure), and if $p_{\psi}(\theta)$ is singular, $p_{\psi}(x)$ is singular.

The nature of the adversarial framework attempts to fit the generator distribution $p_{\psi}$ into the data distribution $p_r$. If a single instance is given as the observation, the data distribution is a singular measure, and the generator distribution pursue the singular distribution. This nature of adversarial framework, with the above inheritance of distribution type, indicates that AVO estimates the proposal distribution with a peaked, singular-like distribution, see Figure 1 of (Louppe et al., 2019).

In contrast, the basic characteristic of a stochastic simulation is that it has a continuous posterior distribution. For instance, the following simulation model (Ikonomov & Gutmann, 2019) is a toy model, where the posterior distribution
Appendix: Adversarial Likelihood-Free Inference on Black-Box Generator

has a plateau that have high probability near an optimal parameter \( \theta^* = 0 \), see Figure 1 of (Ikonomov & Gutmann, 2019).

\[
p(x|\theta) \sim \begin{cases} 
\theta^1 + u & \text{if } \theta \in [-0.5, 0.5] \\
\theta - c + u & \text{otherwise},
\end{cases}
\]

where the constant \( c = 0.5 - 0.5^4 \), the observation is \( x_{\text{obs}} = 0 \), and the nuisance variable \( u \) is sampled from the standard Gaussian distribution \( \mathcal{N}(0, 1) \). In such a model, the sharpened single-peaked proposal around 0 does not match with the flat posterior well. Therefore, the nature of adversarial framework with the proposal approach is not appropriate in the Bayesian inference task for generic black-box generators.

D. Simulation Models

- **Tumor Growth Model** Tumor growth model (Unni & Seshaiyer, 2019) is a system of ordinary differential equations, which combines the interactions between the tumor cells (\( T \)), and cells in the immune systems, including the natural killer cells (\( N \)), dendritic cells (\( D \)), and cytotoxic CD8\(^+ \) T cells (\( L \)).

\[
\begin{cases}
\frac{dN}{dt} = s_1 + \frac{q_1 NT^2}{n_1 + T^2} - (c_2 T - d_1 D + e) N \\
\frac{dT}{dt} = \frac{f_1 L + d_2 N - d_3 T + g)D}{D} - (1 - b) T - kL T \\
\frac{dL}{dt} = f_2 DT - h L T - u N L^2 + r_1 N T - i L
\end{cases}
\]

There are total 21 model parameters, but not all parameters affects to the simulation result in a macro scale. We set the initial tumor size, \( \theta_1 \), and the parameter \( c_1, \theta_2 \) to control parameters. Other parameter values follow the conventional values in (Unni & Seshaiyer, 2019).

For the temporal output of each cell, i.e. \( (T_1, ..., T_{50}) \) for the tumor cell, we obtain the mean through the simulation time and the autocorrelations to lag 5 as summary statistics. Total 24 summary statistics are used in the model. The search space range is set to be \( (\theta_1, \theta_2) \in [3, 7] \times [3e-6, 5e-6] \).

- **Lorenz Model** Lorenz model (Hilborn et al., 2000) is a well-known chaotic system that the next state is unpredictable, though it is deterministic, in the absence of perfect knowledge of initial state and parameter. Lorenz model is initially designed for atmospheric convection, where the coordinates \( x, y, z \) interacts with each other non-linearly.

\[
\begin{cases}
\frac{dx}{dt} = \theta_1 (y - x) \\
\frac{dy}{dt} = \theta_2 x - xz - y \\
\frac{dz}{dt} = xy - \theta_3 z
\end{cases}
\]

For each coordinate, the simulated output is the temporal positions \( (x(1), ..., x(T)) \), with \( T = 5000 \), and we use equally distributed 8 quantiles as summary statistics. Total 24 summary statistics are used. The search space range is set to be \( (\theta_1, \theta_2, \theta_3) \in [3, 12] \times [20, 30] \times [1, 5] \).

- **Susceptible-Infectious-Recovered Model** SIR model (Diekmann & Heesterbeek, 2000; Murray, 2007) presents an explosive chain reaction of infectious disease by observing the dynamics of the population \( (N) \) compartments devided by the Susceptible (S), the Infectious (I), and the Recovered (R).

\[
\begin{align*}
\frac{dS}{dt} &= -\frac{\theta_1 IS}{N} \\
\frac{dI}{dt} &= \frac{\theta_1 IS}{N} - \theta_2 I \\
\frac{dR}{dt} &= \theta_2 I.
\end{align*}
\]

For each simulated subpopulations, we use the same summary statistics used in the Tumor growth model. Total 18 summary statistics are used. The search space range is set to be \( (\theta_1, \theta_2) \in [0.5, 1.5]^2 \).

- **Poisson Model** Poisson model is the second-order elliptic partial differential equation. Laplace operator in Poisson equation, the left-hand-side of Eq. 2, models the random walk effect weighted by model parameters for each dimension. Laplace operator is one of the most basic differential equation in science. For example, Hamiltonian operator in Schrodinger equation contains the Laplace operator. There are a bunch of applications of Poisson model itself. For instance, the estimation of permeability in porous material (Jo & Kwak, 2017), or oil and gas reservoirs, is a crucial problem when it is inaccessible to obtain the inner pressure of given porous media.

\[
\begin{cases}
-\theta_1 u_{xx} - \theta_2 u_{yy} = \delta_0 & \text{in } \Omega \\
u = 0 & \text{on } \partial\Omega
\end{cases}
\]

Here, the domain \( \Omega \) is assumed to be \( [0, 1]^2 \). The observation is a set of solutions \( \{u_1^*, ..., u_{49}^*\} \) at equally spaced pre-designated positions \( x_1, ..., x_{49} \), respectively. The search space range is set to be \( (\theta_1, \theta_2) \in [0, 1]^2 \).

- **New Product Adoption Model** New Product Adoption model is a system dynamics model that expresses the adopters of a new product as the sum of the following sub-populations out of total populations (P). First sub-population is the imitators (I) who are influenced heavily by their surrounding product owners. Second sub-population (AE) adopts the product mainly driven
Appendix: Adversarial Likelihood-Free Inference on Black-Box Generator

by the advertising effect. Third sub-population is consis-
ted of the active adopters (AA). The feedback loop is
given by
\[
\begin{align*}
I(t) &= a(P - I(t) - AE(t) - AA(t)) \\
& \quad \times ((t) + AE(t) + AA(t)) \\
AE(t) &= b(P - I(t) - AE(t) - AA(t)) \\
AA(t) &= c(P - I(t) - AE(t) - AA(t))
\end{align*}
\]

We choose the parameters \( a \) and \( b \) to control parameters.

For each simulated sub-populations, we use equally
distributed 5 quantiles as summary statistics. Total 15
summary statistics are used. The search space range
is set to be \( \{\theta_1, \theta_2\} \in [0, 0.02] \times [0, 0.1] \).

- **Auto-Regressive Conditional Heteroskedasticity Model**
  Auto-Regressive Conditional Heteroskedasticity (ARCH) model (Gutmann et al., 2018) is a
time-series model that exhibit time-varying volatility
that is mainly utilized in financial time-series
modelling. The equation is given by
\[
x_t = \theta_1 x_{t-1} + \xi_t \sqrt{0.2 + \theta_2 x_{t-1}^2},
\]
where \( \xi_t \) and \( \epsilon_0 \) are independent standard Gaussian
random variables. The first parameter \( \theta_1 \) is the mean
process coefficient and the second parameter \( \theta_2 \) is the variance
process coefficient.

We use the autocorrelations to lag 5 as summary statistics
in ARCH model. The search space range is set to
be \( \{\theta_1, \theta_2\} \in [0, 1] \times [1, 2] \).

- **Queueing Model**
  M/G/1 model (Chen & Gutmann, 2019; Newell, 2013) is a queuing model with a single
server, where arrival \( (v) \) is modeled by Poisson pro-
cess, and service time \( (s) \) is modeled by a general distri-
bution. In our experiment, the sevice time follows the
uniform distribution.
\[
\begin{align*}
s(i) &\sim U(\theta_1, \theta_1 + \theta_2) \\
v(i) - v(i-1) &\sim Exp(\theta_3)
\end{align*}
\]
The observation is the sequence \( \{d(i) - d(i - 1)\}_{i} \),
where \( d(i) = d(i - 1) = s(i) + \max(0, v(i) - d(i - 1)) \).

We use equally distributed 16 quantiles as summary statistics.
The search space range is set to be
\( \{\theta_1, \theta_2, \theta_3\} \in [0, 10] \times [0, 5] \). The third parameter
is set to be \( \theta_3 = 0.2 \) because this parameter is insensitive
to the simulation result.

- **Wealth Distribution Model**
  A wealth distribution model (Wilensky, 1998) is an agent-based model that
describes the macroscopic wealth distribution via mi-
croscopic agent behaviors. In wealth model, a grid
provides its wealth to agents who lives in the grid. At
the end of each timestep, agents consume their wealth
to live and move toward the wealthiest grid among four
neighboring directions. In addition, grids recover their
wealth at the end of each timestep. \( i \)-th Agent has its
income \( (I) \) and outcome \( (O) \) as
\[
\begin{align*}
I(t) &= (\theta_1 + \theta_2) \epsilon_t \frac{W_i}{\sum_i W_i} \\
O(t) &= \frac{W_i}{\sum_i W_i}
\end{align*}
\]
where \( N_i \) is the number of agents in the grid where
agent \( i \) resides and \( W_i \) is the net wealth of the grid
where agent \( i \) resides.

Four types of simulation outputs are considers: first
three are low/middle/high class agents wealth average,
and the last output is the gini index. We use
equally distributed 5 quantiles as summary statistics
for each simulation output. Total 20 summary statistics
are used. The search space range is set to be
\( \{\theta_1, \theta_2\} \in [0, 0.3] \times [0.5, 0.1] \).

E. Alternative Loss Function for
Beta-Estimation Network

In the paper, we design the beta-estimation network to be
the log likelihood of the current training dataset \( D_b = \{\theta_i, d_\phi(g(\theta, u_i))\} \). The loss function is given by
\[
\mathcal{L}_b(\psi) = -E_{\theta, u}[(b_0^\phi(\theta) - 1) \log d_\phi(g(\theta, u)) + (b_0^\phi - 1) \log (1 - d_\phi(g(\theta, u))) - \log B(b_0^\phi(\theta), b_0^\phi(\theta))].
\] (3)
The optimum of the loss function Eq. 3 with training
data \( D_b = \{\theta_i, d_\phi(g(\theta, u_i))\} \) would yield \( b_0^\phi(\theta_i) = \frac{d_\phi(g(\theta_i, u_i))}{1-d_\phi(g(\theta, u_i))} \).

We suggest in this section an alternative loss function for
the beta-estimation network. The alternative loss is given by
\[
\mathcal{L}_b(\psi) = -E_{\theta, u}[E_{y \sim Beta(b_0^\phi(\theta), b_0^\phi(\theta))}[d_\phi(g(\theta, u)) \log y + (1 - d_\phi(g(\theta, u))) \log (1 - y)]]
\] (4)
The inner term in the alternative loss function is the
log likelihood of data \( d_\phi(g(\theta, u)) \), where the probabil-
ity for \( d_\phi(g(\theta, u)) = 1 \) is \( y \) and the probability for
\( d_\phi(g(\theta, u)) = 0 \) is \( 1 - y \). Since the term \( d_\phi(g(\theta, u)) \log y + (1 - d_\phi(g(\theta, u))) \log (1 - y) \) is minimized when \( y = d_\phi(g(\theta, u)) \), the optimum of the alternative loss function
with training data \( D_b = \{\theta_i, d_\phi(g(\theta_i, u_i))\} \) would satisfy \( \frac{b_0^\phi(\theta_i)}{b_0^\phi(\theta)} = \frac{d_\phi(g(\theta_i, u_i))}{1-d_\phi(g(\theta, u_i))} \). This leads the two loss functions,
Eq. 3 and 4, have the same property at the optimum point.
Appendix: Adversarial Likelihood-Free Inferece on Black-Box Generator

Figure 1. Performance comparison with ALFI and five baseline models

F. Experiments Details
Throughout the experiments, we use Titan XP GPU and Intel i7-9700K CPU. We use Adam optimizer with beta values (0.5, 0.999) for the discriminator and for the beta-estimation network, with learning rate 1e-4 for discriminator, and 1e-2 for beta-estimation network. Both learning rates are decayed by 0.99 exponentially for every iteration before 300 iterations, and decayed 0.98 exponentially for every iteration after 300 iterations. The mini-batch size is 100. In experiments, we use the discriminator loss function suggested in (Arjovsky et al., 2017).

F.1. Network Architectures and Hyperparameters for Simulations as Black-Box Generator
We use four fully-connected layers for the discriminator network. The first layer has 300 neurons, the second layer has 150 neurons, the third layer has 50 neurons, and the last layer has 10 neurons. We use the leaky ReLU activation functions with 0.2 as negative slope for the first, the third, and the fourth layers and use the ReLU activation function for the second layer. We use a single fully-connected layer with 30 neurons for the beta-estimation network, with the hyperbolic tangent activation function. We use hyperparameters \((m, k, l) = (10, 20, 50)\) for experiments in Figure 6-(b) and hyperparameters \((m, k, l) = (10, 3, 30)\) for experiments otherwise. We take 100,000 simulation budgets to obtain the performance, except for the Lorenz and the Wealth distribution models. In Lorenz model, we take 30,000 simulation budgets (timestep of Lorenz model is 5000), and in Wealth distribution model, we take 20,000 simulation budgets (running of agent-based model is expensive). In Rejection ABC, we take 10 times more simulations than other algorithms to obtain the lower bound performance.

F.2. Network Architectures and Hyperparameters for GAN Generator as Black-Box Generator
We use DCGAN discriminator for the discriminator network. We use three fully-connected layers for the beta-estimation network. The first layer has 30 neurons, the second layer has 20 neurons, and the last layer has 10 neurons. The first and the third layers use the leaky ReLU activation functions with 0.2 as negative slope and the second layer uses the ReLU activation function. The learning rate of the discriminator network is 1e-4, and the leaning rate of the beta-estimation network is 1e-3. We use hyperparameters \((m, k, l) = (10, 1, 30)\). The search space is \([-2, 2]^d\), where \(d\) is the dimension of the parameter \(\theta\). We take 20,000 evaluation budget for the given black-box model.

G. Additional Experiments

G.1. Alternative Loss Function
The alternative loss function introduced in Eq. 4 reports the similar performance with the original log-likelihood loss function as in Eq. 3, see Figure 1.

G.2. Metropolis-Hastings Samples for Various Simulation Budgets
Figure 2 and 3 illustrates the trajectory of the samples from Metropolis-Hastings algorithm in ALFI for New Product Adoption model and Tumor model, respectively. Figure 2 shows that ALFI finds a sample mode falling into the neighborhood of the true parameter \(\theta^*\) within 5000 simulation budgets, and the samples are concentrated in afterward simulations. Additional experiment on the Tumor Growth Model in Figure 3 illustrates the similar behavior in Figure 2.

References
Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein gan. arXiv preprint arXiv:1701.07875, 2017.
Chen, Y. and Gutmann, M. U. Adaptive gaussian copula abc. arXiv preprint arXiv:1902.10704, 2019.
Diekmann, O. and Heesterbeek, J. A. P. Mathematical epidemiology of infectious diseases: model building, analysis and interpretation, volume 5. John Wiley & Sons, 2000.
Appendix: Adversarial Likelihood-Free Inference on Black-Box Generator

(a) 5000 Simulations  
(b) 10000 Simulations  
(c) 15000 Simulations  
(d) 20000 Simulations

Figure 2. Metropolis-Hastings Samples and the Posterior Density Estimation for Various Simulation Budgets in New Product Adoption Model

Gilbarg, D. and Trudinger, N. S. *Elliptic partial differential equations of second order*. Springer, 2015.

Gutmann, M. U., Dutta, R., Kaski, S., and Corander, J. Likelihood-free inference via classification. *Statistics and Computing*, 28(2):411–425, 2018.

Hilborn, R. C. et al. *Chaos and Nonlinear Dynamics: an Introduction for Scientists and Engineers*. Oxford University Press on Demand, 2000.

Ikonomov, B. and Gutmann, M. U. Robust optimisation monte carlo. *arXiv preprint arXiv:1904.00670*, 2019.

Jo, G. and Kwak, D. Y. An impes scheme for a two-phase flow in heterogeneous porous media using a structured grid. *Computer Methods in Applied Mechanics and Engineering*, 317:684–701, 2017.

Louppe, G., Hermans, J., and Cranmer, K. Adversarial variational optimization of non-differentiable simulators. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1438–1447, 2019.

Murray, J. D. *Mathematical biology: I. An introduction*, volume 17. Springer Science & Business Media, 2007.

Newell, C. *Applications of Queueing Theory*, volume 4. Springer Science & Business Media, 2013.

Robert, C. and Casella, G. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

Figure 3. Metropolis-Hastings Samples and the Posterior Density Estimation for Various Simulation Budgets in Tumor Growth Model

Unni, P. and Seshaiyer, P. Mathematical modeling, analysis, and simulation of tumor dynamics with drug interventions. *Computational and mathematical methods in medicine*, 2019, 2019.

Wilensky, U. Netlogo wealth distribution model. *Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.* [En línea] Disponible en: http://ccl.northwestern.edu/NetLogo/models/WealthDistribution, 1998.