Fine Tuning a Data-Driven Estimator

Braghadeesh Lakshminarayanan and Cristian R. Rojas

***Abstract*— Industries often use high-fidelity simulators like digital twins to represent physical systems, but these simulators require calibrated parameters to match reality. Data-driven estimators address this by mapping synthetic observations—generated under various parameter settings—to those parameters using supervised learning. However, if the true parameters lie outside the sampled range, out-of-distribution (OOD) issues can arise. This paper introduces a fine-tuning method for the Two-Stage estimator, a particular type of data-driven estimator, to overcome OOD challenges and enhance its accuracy.**

I. INTRODUCTION

In this paper, we consider a data-driven approach to parameter estimation using digital twins (DTs), namely the two-stage (TS) estimator. With high-fidelity simulators such as DTs, one can generate synthetic input-output datasets paired with parameter values that correspond to different operating regimes of a DT. These datasets are used to train a machine learning model to map system observations to parameter estimates, forming a data-driven estimator. The TS estimator, in particular, compresses observations into representative features in its first stage and then uses these features to train a supervised model in its second stage. This approach avoids the complexity of explicit likelihood calculations and, as research shows [1], [2], can outperform traditional methods like Kalman filters, or the prediction error method. However, its effectiveness relies on the assumption that the system parameters lie within the range used during offline training. If this assumption fails—leading to an out-of-distribution (OOD) scenario—the estimates can be biased. To address this, we propose a perturbation-based tuning method for the TS estimator, where the second stage is a deep neural network. By comparing the compressed features from real observations with those from DT simulations (using the initially predicted parameters), the method constructs a perturbation set, based on which we generate a new synthetic dataset. This synthetic dataset is then used to recalibrate the neural network in the TS estimator, fine-tuning its weights and reducing the bias of the parameter estimates without prior knowledge of the true parameters.

II. PROBLEM STATEMENT

Consider a controlled system represented by a data generating mechanism $\mathcal{M}(\theta)$ with parameters $\theta \in \Theta \subseteq \mathbb{R}^d$. For example, a digital twin (DT) of a single-input-single-output (SISO) system produces a time series $\boldsymbol{z} = (u_1, y_1, u_2, y_2, \ldots, u_N, y_N)$, where u_i and y_i denote the inputs and outputs at time i. The true system is represented by $\mathcal{M}(\theta_0)$ for some unknown $\theta_0 \in \Theta$. In the simulation-driven estimation approach, synthetic observations are generated from $\mathcal{M}(\theta)$ at parameter values $\tilde{\theta}_i$ sampled from a subset $\Theta_p \subseteq \Theta$. This subset, based on the user's prior beliefs about operating scenarios, forms the training set $\mathcal{D}_{\text{tr}} = \{(\boldsymbol{z}_i, \tilde{\theta}_i)\}_{i=1}^m$. A supervised learning model is then trained to map observations to parameters by solving $\hat{\theta}_{\text{pre}}(\cdot) = \arg\min_{F\in\mathcal{F}} \frac{1}{m}\sum_{i=1}^m L\big(F(\mathbf{z}_i), \tilde{\theta}_i\big)$, where L is a loss function and $\mathcal{F}$ is a class of function approximators (e.g., deep neural networks). The pretrained estimator $\hat{\theta}_{\text{pre}}(\cdot)$ is then used to obtain an estimate from new observations $\mathbf{z}_0$ via $\hat{\theta}_0 = \hat{\theta}_{\text{pre}}(\mathbf{z}_0)$. If $\theta_0 \in \Theta_p$, the estimator is approximately unbiased. However, if $\theta_0 \notin \Theta_p$, an out-of-distribution (OOD) issue may arise, leading to bias. To address this, we propose fine-tuning the pretrained estimator—specifically, the Two-Stage (TS) estimator—to enhance its accuracy when $\theta_0 \notin \Theta_p$. The TS (Two-Stage) estimator is a data-driven method that leverages digital twins through offline pretraining on a synthetic dataset $\mathcal{D}_{\text{tr}}$. Its structure is as follows: (i) **Stage 1 (Data Compression):** High-dimensional inputs $\{\boldsymbol{z}_i\}_{i=1}^m$ are compressed to low-dimensional features using a function $h\colon \mathbb{R}^N \to \mathbb{R}^n \quad (n \ll N)$. For *i.i.d.* observations, h might compute quantiles; for time-series data, it could correspond to the coefficients of an estimated ARX/ARMAX model. This yields the compressed set $\mathcal{D}_{\text{tr}}^{\text{comp}} = \{(h(\boldsymbol{z}_i), \tilde{\theta}_i)\}_{i=1}^m$., (ii) **Stage 2 (Function Approximation):** A supervised learning model $g \in \mathcal{G}$ is then trained on $\mathcal{D}_{\text{tr}}^{\text{comp}}$ by minimizing the empirical risk

$$g_{\text{pre}}(\cdot) = \arg\min_{g\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^m L\big(g(h(\boldsymbol{z}_i)), \tilde{\theta}_i\big), \tag{1}$$

where $\mathcal{G}$ is a class of functions $g\colon \mathbb{R}^n \to \mathbb{R}^d$. The pretrained TS estimator is then given by $\hat{\theta}_{\text{pre}}(\cdot) = g_{\text{pre}} \circ h(\cdot)$. If observations $\boldsymbol{z}_0$ from a true system with $\theta_0 \notin \Theta_p$ are received, $\hat{\theta}_{\text{pre}}(\boldsymbol{z}_0)$ may yield poor estimates, highlighting the need for fine-tuning.

III. PERTURBATION APPROACH TO FINE-TUNE TS

We now describe a perturbation approach [3] to fine-tune the pretrained TS estimator. Recall that the estimator is originally trained on a synthetic dataset $\{(\boldsymbol{z}_i, \tilde{\theta}_i)\}_{i=1}^m$, with each $\tilde{\theta}_i \in \Theta_p \subseteq \mathbb{R}^d$, and is given by $\hat{\theta}_{\text{pre}}(\cdot) = g_{\text{pre}} \circ h(\cdot)$, where

$$g_{\text{pre}}(\cdot) = \arg\min_{g\in\mathcal{G}} \frac{1}{m}\sum_{i=1}^m L\big(g(h(\boldsymbol{z}_i)), \tilde{\theta}_i\big). \tag{2}$$

The authors are with the Division of Decision and Control Systems, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden (e-mails: blak@kth.se; crro@kth.se).

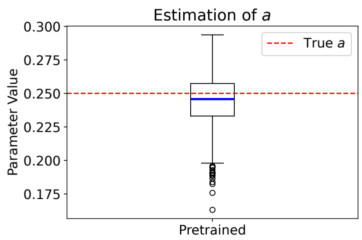

Fig. 1: Box plot of estimates of $a \in \Theta_p$. The red dashed line: true value of $a = 0.25$; thick blue line: mean of the pretrained TS estimates.

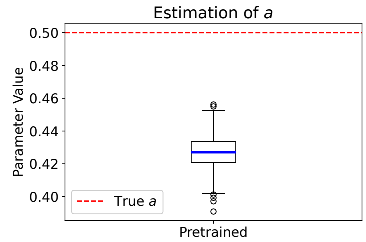

Fig. 2: Box plot of estimates of $a \notin \Theta_p$. The red dashed line: true value of $a = 0.5$; thick blue line: mean of the pretrained TS estimates.

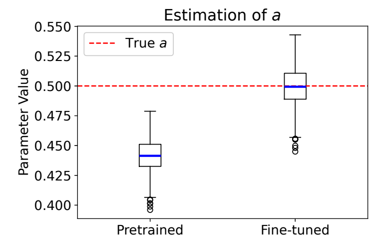

Fig. 3: Pretrained and fine-tuned TS estimates of $a \notin \Theta_p$. Red dashed line: true value of a; thick blue lines: respective mean estimates.

When $\mathcal{G}$ is a class of neural networks parameterized by weights $\boldsymbol{w} \in \mathbb{R}^{n_w}$, the optimal weights $\boldsymbol{w}_{\text{pre}}$ can be computed via stochastic gradient descent: $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \beta \nabla_{\boldsymbol{w}} f(\boldsymbol{w})\big|_{\boldsymbol{w}=\boldsymbol{w}_t}$, where $f(\boldsymbol{w}) := \frac{1}{m}\sum_{i=1}^{m} L\big(g_{\boldsymbol{w}}(h(\boldsymbol{z}_i)), \tilde{\theta}_i\big)$. After training, if new observations $\boldsymbol{z}_0$ are obtained from a system with $\theta_0 \notin \Theta_p$, the TS estimator can lead to poor estimates. Because the compression function h is fixed, only the second stage $g_{\text{pre}}(\cdot)$ is fine-tuned. Specifically, we 1) compute an initial estimate $\hat{\theta}_{\text{init}} = \hat{\theta}_{\text{pre}}(\boldsymbol{z}_0)$ and generate synthetic observations $\tilde{\boldsymbol{z}}_0 = \mathcal{M}(\hat{\theta}_{\text{init}})$; 2) compress both $\boldsymbol{z}_0$ and $\tilde{\boldsymbol{z}}_0$ with h and compute the discrepancy $\delta = \eta\|h(\boldsymbol{z}_0) - h(\tilde{\boldsymbol{z}}_0)\|_2$, where $\eta > 0$ is a hyperparameter; 3) construct a perturbation set $\Theta_\delta = \mathcal{B}(\hat{\theta}_{\text{init}}, \delta)$ and sample m' parameter values $\{\bar{\theta}_i\}_{i=1}^{m'}$ from it. For each $\bar{\theta}_i$, generate $\bar{\boldsymbol{z}}_i = \mathcal{M}(\bar{\theta}_i)$ to form the dataset $\mathcal{D}_{\text{fine}} = \{(\bar{\boldsymbol{z}}_i, \bar{\theta}_i)\}_{i=1}^{m'}$; 4) retrain the network on $\mathcal{D}_{\text{fine}}$ (starting from $\boldsymbol{w}_{\text{pre}}$) using $\boldsymbol{w}_{\text{fine}} = \arg\min_w \frac{1}{m'}\sum_{i=1}^{m'} L\big(g_w(h(\bar{\boldsymbol{z}}_i)), \bar{\theta}_i\big)$, yielding the fine-tuned network $g_{\text{fine}}(\cdot) = g_{\boldsymbol{w}_{\text{fine}}}(\cdot)$. The final fine-tuned TS estimator is $\hat{\theta}_{\text{fine}}(\cdot) = g_{\text{fine}} \circ h(\cdot)$, and the refined estimate is given by $\hat{\theta}_0 = \hat{\theta}_{\text{fine}}(\boldsymbol{z}_0)$.

IV. Simulation Study

In this section, we validate the effectiveness of our perturbation-based fine-tuning approach for a pretrained TS estimator. We consider a simple numerical example to demonstrate that the fine-tuned TS estimator, $\hat{\theta}_{\text{fine}}(\cdot)$, achieves improved performance. Consider an autonomous system described by the state-space model [1]: $x_{k+1}^{(1)} = ax_k^{(1)} + v_k^{(11)}$, $x_{k+1}^{(2)} = x_k^{(1)} + a^2 x_k^{(2)} + v_k^{(12)}$, $y_k = ax_k^{(1)} + x_k^{(2)} + v_k^{(2)}$, where $x_k^{(1)}$ and $x_k^{(2)}$ are hidden states, and y_k is the output at time k. Also, $v_k^{(11)} \sim \mathcal{N}(0,1)$ and $v_k^{(12)} \sim \mathcal{N}(0,1)$ are additive process noises, and $v_k^{(2)} \sim \mathcal{N}(0, 0.01)$ is the observation noise at time k, assumed to be mutually uncorrelated white noises. Here, a is the unknown parameter to be estimated and, for stability reasons, a is restricted to the interval $(-1, 1)$. To build the synthetic dataset, we first draw $m = 5000$ samples $\{\tilde{a}_i\}_{i=1}^m$ uniformly from $(-0.3, 0.3)$. For each $\tilde{a}_i$, observations of length $N = 1000$ are generated. In Stage 1, the TS estimator computes AR(5) coefficients $\boldsymbol{\alpha}_i \in \mathbb{R}^5$ and in Stage 2, a deep neural network (with linear layers and $\text{ReLU}(x) = \max(x, 0)$) is pretrained on the dataset $\{(\boldsymbol{\alpha}_i, \tilde{a}_i)\}_{i=1}^m$. Evaluating the pretrained TS estimator at $a = 0.25$ yields unbiased estimates with a mean squared error (MSE) of 0.0004 (see Figure 1). Now, consider the case where the true parameter is $a = 0.5$ (an OOD scenario, since the training samples lie in $(-0.3, 0.3)$). Figure 2 shows that the estimator becomes biased, and the MSE increases to 0.004 (about 10 times larger than when $a = 0.25$). To address this, we apply the perturbation method using the initial predictions from the pretrained TS estimator. In our experiment, we set $\eta = 0.7$ and $m' = 2000$. Figure 3 displays box plots of the parameter estimates for both the pretrained and fine-tuned TS estimators. The results clearly show that the fine-tuned TS estimator provides significantly improved estimates for $a = 0.5$, a value that lies outside the training range $\Theta_p = (-0.3, 0.3)$. Specifically, the pretrained TS estimator $\hat{\theta}_{\text{pre}}(\cdot)$ achieves an MSE of 0.0036, whereas the fine-tuned TS estimator $\hat{\theta}_{\text{fine}}(\cdot)$ obtains an MSE of 0.0003, which is ten times lower. This outcome validates that our perturbation approach effectively fine-tunes the TS estimator and enhances its performance in OOD scenario.

V. Conclusion

In this paper, we proposed a perturbation-based fine-tuning approach for the TS estimator to adapt to OOD observations from systems outside the original model set. By leveraging the variability in compressed features, we construct a perturbation set to generate a new dataset for retraining the estimator's second stage. Numerical example shows that the fine-tuned TS estimator yields improved parameter estimates. In future work, we shall provide theoretical justification for this approach.

References

[1] S. Garatti and S. Bittanti. Estimation of white-box model parameters via artificial data generation: a two-stage approach. *IFAC Proceedings Volumes*, 41(2):11409–11414, 2008.

[2] S. Garatti and S. Bittanti. A new paradigm for parameter estimation in system modeling. *International Journal of Adaptive Control and Signal Processing*, 27(8):667–687, 2013.

[3] B. Lakshminarayanan and C. R. Rojas. Fine tuning a data-driven estimator. In *Submitted to 64th IEEE Conference on Decision and Control*, 2025. `https://arxiv.org/abs/2504.04480`.