Reliable and Scalable Robot Policy Evaluation with Imperfect Simulators

Apurva Badithela[*]
Princeton University

David Snyder[†]
Princeton University

Lihan Zha[†]
Princeton University

Joseph Mikhail
UT Austin

Matthew O'Kelly[‡]

Anushri Dixit[‡]
UCLA

Anirudha Majumdar
Princeton University

Abstract: Rapid progress in robot manipulation driven by imitation learning, foundation models, and large-scale datasets has enabled generalization to a wide-range of tasks and environments. However, rigorous evaluation of these policies remains a challenge. Typically in practice, robot policies are often evaluated on a small number of hardware trials without any statistical assurances. We present SureSim, a framework to augment large-scale simulation with relatively small-scale real-world testing to provide reliable inferences on the real-world performance of a policy. Our key idea is to formalize the problem of combining real and simulation evaluations as a prediction-powered inference problem, in which a small number of paired real and simulation evaluations are used to rectify bias in large-scale simulation. We then leverage non-asymptotic mean estimation algorithms to give confidence intervals on policy performance. Using physics-based simulation, we evaluate both diffusion policy and multi-task fine-tuned π_0 on a joint distribution of objects and initial conditions, and find that on our approach saves over 20% of hardware evaluation effort to achieve similar bounds on policy performance.

Keywords: Evaluation, Finite-Sample Statistical Inferences, Real2Sim

1 Introduction

Advancing robot learning requires rigorous policy evaluation for reliably assessing how policies generalize to new tasks and environments [1]. Rapid progress in deep learning was driven by standardized metrics and evaluation benchmarks such as ImageNet [2] and COCO [3] in computer vision, and SquaD [4] and GLUE/SuperGLUE [5, 6] in natural language. Unlike the static benchmarks in these domains, robot policy evaluation in the real-world requires physical interaction of the robot and its environment which is resource intensive in time and human effort. For instance, consider the fundamental question of evaluating the success rate of a policy on a distribution of environments. Due to the expensive nature of real-world evaluation, most research studies report empirical success rates of policies evaluated on a small number (e.g., 20-40) of trials. At the same time, there is a growing consensus for rigorous statistical analysis and nuanced discussions of evaluation criteria and policy failure modes [7, 8, 9]. As a result, assessing whether a policy will perform reliably in a new environment distribution remains a core challenge [1].

[*]Correspondence: ab5832@princeton.edu
[†]Equal contribution.
[‡]Equal advising.

Workshop on Safe and Robust Robot Learning for Operation in the Real World at CoRL 2025.

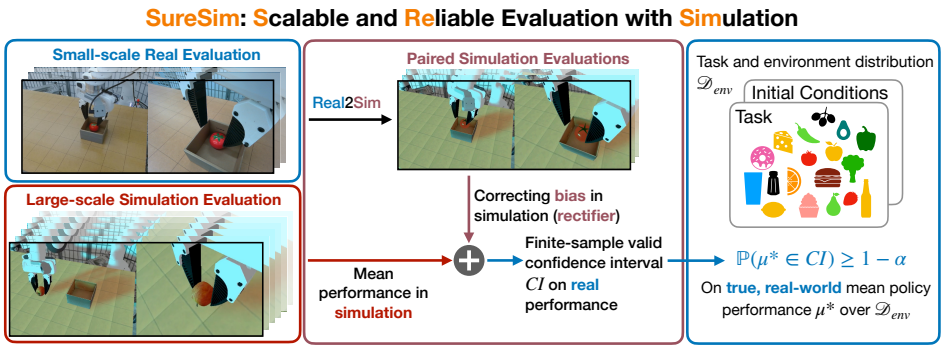


Figure 1: Our goal is to evaluate a policy by computing bounds on its mean real-world performance on a diverse environment distribution $\mathcal{D}_{env}$. We present a framework that augments real-world evaluations with simulation evaluations to provide stronger inferences on real-world policy performance that could otherwise only be obtained by scaling up real-world evaluations.

In robotic manipulation, recent advances in physics-based simulators [10] and action-conditioned video prediction models [11, 12] provide scalable alternatives to real-world policy evaluation. While growing evidence suggests that simulation performance correlates well with real-world performance in aggregate across a diversity of tasks and environments [13, 14], the simulation-to-real gap precludes rigorous statistical inferences about real-world outcomes from simulation results alone. This paper investigates this gap by augmenting a small number of real-world evaluations with large-scale simulations to achieve scalable policy evaluation with trustworthy statistical inferences about real-world performance. Crucially, our framework can achieve tighter statistical bounds on policy performance by scaling up the number of simulations in place of scaling the number of real-world evaluations. However, using large-scale simulations for policy evaluation for making trustworthy statistical inferences on real performance faces significant challenges due to the simulation-real gap, stemming from mismatches in visual features (e.g., lighting conditions, object textures) and inaccurate modeling of contact physics and real physical parameters (e.g., friction coefficients) [15, 16]. Current robot policies, including foundation models and imitation learning-based policies, can be sensitive to these mismatches. As a result, performance bounds solely relying on large-scale simulation predictions can be biased.

We tackle the aforementioned challenges to provide confidence intervals for mean estimation of the performance of robot manipulation policies. Our key idea is to connect the problem of combining simulated and real-world evaluations for trustworthy estimation of policy performance to prediction powered inference (PPI) [17, 18]. Prediction powered inference is a paradigm for computing statistically valid confidence intervals on the performance of learned models by leveraging a large set of learned predictions together with a comparatively small number of gold-standard labels. In our case, gold-standard labels constitute real-world evaluations of a robot manipulation policy and the predictions come from simulation evaluations. For a bounded performance metric, our confidence intervals are valid using the finite sample of data gathered from real and simulation evaluations. When simulation is sufficiently predictive, PPI yields tighter non-asymptotic confidence bounds than using real-world trials alone, enabling us to scale simulation evaluations rather than costly real-world evaluations.

Statement of Contributions. First, we present a rigorous policy evaluation framework that enables finite-sample valid inferences on real-world performance by combining large-scale simulation trials with a relatively small number of real trials. A key step is pairing each real trial with its corresponding simulation trial on the same task or environment instance to estimate and correct for simulation bias. To operationalize this, we introduce a real2sim pipeline that enables us to leverage prediction powered inference, and we additionally identify a set of best practices for using simulation alongside real-world evaluation to obtain tighter confidence intervals. Second, we demonstrate our evaluation paradigm on a single-task diffusion policy [19] trained from scratch as well as robot foundation models [20] finetuned on multiple tasks, yielding a 20% benefit in hardware trials saved.

2 Problem Statement

Let $\mathcal{D}_{\text{env}}$ denote a distribution over real-world environments $\mathcal{X}$ in which we wish to evaluate a robot policy $\pi \in \Pi$. In robotic manipulation, this distribution could be defined by the diversity of objects and tasks, environmental factors (e.g., lighting, background, table texture), and spatial variations in object and robot poses, among others. We assume a bounded evaluation metric $M : \mathcal{X} \times \Pi \to [0, 1]$, such as binary success/failure or continuous scores reflecting partial task completion.

We consider the mean estimation problem in policy evaluation, where the goal is to estimate the policy's average performance according to metric M over the environment distribution $\mathcal{D}_{\text{env}}$. Formally, we define mean policy performance μ^* as:

$$\mu^* = \mathbb{E}_{X \sim \mathcal{D}_{\text{env}}}[Y(X)], \tag{1}$$

where $Y(X) = M(X, \pi)$ is the outcome of evaluating policy π in environment X under metric M. For sampled environments $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \mathcal{D}_{\text{env}}$, the outcomes of real-world policy evaluation according to metric M are denoted as $Y_1, \ldots, Y_n$, respectively. We define the empirical evaluation sample as $S_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\} \subseteq \mathcal{D}_{\text{env}}^n$. Using the empirical data S_n we seek a confidence interval $CI = (l, u)$ that contains μ^* with high probability. Confidence intervals provide bounds on the true performance of a policy with high probability from sampled evaluations, and can be useful for decision-making and policy comparison. Mathematically stated, for any significance level $\alpha \in (0, 1)$ and any finite number n of real-world evaluations, we seek a confidence interval CI such that:

$$\mathbb{P}_{S_n \sim \mathcal{D}_{\text{env}}^n}(\mu^* \in CI) \geq 1 - \alpha, \tag{2}$$

where the probability measure is defined over the draw of the empirical evaluation sample S_n. Any method that satisfies Equation (2) is type-I error controlling at significance level α. While the interval $[0, 1]$ trivially satisfies this guarantee, it provides little insight; therefore, we seek a tight confidence interval satisfying Equation (2). Importantly, we make no assumptions on the distribution of Y_i beyond measurability and the boundedness induced by the metric M. We do not require *a priori* knowledge of a distribution family, the existence of a density, or other structural assumptions.

A nonasymptotic confidence interval for μ^* can be derived directly from the finite number of gold-standard evaluations $Y_1, \ldots, Y_n$ using standard non-asymptotic methods like Hoeffding [21] or Bernstein inequalities, or more recent state-of-the-art betting-based methods [22]. Ideally, we want a tight interval concentrated around μ^*, but collecting a large number of real-world evaluations is costly. In contrast, simulation evaluations are relatively cheap and scalable. This motivates the central question of our work: *Can we make valid inferences on the real performance of a policy by augmenting a small amount of real-world evaluations with a large number of simulation evaluations?*

3 Simulation to Augment Real Tests via Prediction Powered Inference

Suppose we have access to a simulator for policy evaluation, and let $\mathcal{X}_{\text{sim}}$ denote the simulation environments. We also assume a real2sim function $g : \mathcal{X} \to \mathcal{X}_{\text{sim}}$ that translates a real environment setup into simulation. For each real environment $X \in \mathcal{X}$, denote the corresponding simulation environment as $\tilde{X} = g(X)$. For example, as shown in Figure 1, if X is a robot manipulation environment-defined by the robot (type, dynamics, texture, and initial pose), the objects (3D models, textures, material properties, and initial pose), and background conditions (lighting and background textures) — then the corresponding simulation environment $\tilde{X}$ is constructed to closely match the real-world dynamics and visual features. Implementation details are presented in Section 4. Simulation evaluations are given by the function $f : \mathcal{X}_{\text{sim}} \to [0, 1]$ defined as $f(\tilde{X}) = M_{\text{sim}}(\tilde{X}, \pi)$, where $\tilde{X} \in \mathcal{X}_{\text{sim}}$ and M_{sim} simulation evaluation metric.

While we can run large-scale simulations and estimate the mean policy performance in simulation, it is not necessary that the true mean μ^* should be close to the simulation mean. Correcting for bias in simulation predictions and deriving valid confidence intervals for μ^* requires more than simply combining real and simulated evaluations. To tackle this challenge, we identify prediction

powered inference (PPI) [17] as a suitable mathematical framework for our problem. Prediction powered inference is a framework for valid statistical inference when experimental datasets are supplemented with machine-learning predictions. It has been applied to diverse problems such as protein structure analysis with AlphaFold, galaxy classification, and deforestation monitoring using computer vision [17]. For example, in galaxy classification, human annotators provide a limited set of ground-truth labels ("spiral" vs. "not spiral") from galaxy images, while computer vision models provide cheaper predictions on the input images at a much larger-scale. In our setting, each input corresponds to a robot manipulation environment X, with the ground-truth label given by real outcome $Y(X)$ of rolling out the policy. We choose simulation as a proxy for real-world evaluation but unlike the problems studied in [17], we cannot directly evaluate on X in simulation. Thus, we introduce the real2sim function which enables composing simulation predictions with the real2sim function: $f(\tilde{X}) = f(g(X))$. Therefore, we can use prediction powered inference rigorously combine real-world tests with large-scale simulation to reliably infer the real performance.

To apply PPI, we require a small number of paired evaluations in both real and simulation. For the set of $n + N$ real environments $X_1, \dots X_{n+N} \overset{\text{iid}}{\sim} \mathcal{D}_{\text{env}}$, we can apply the real2sim function to get a set of simulation environments $\tilde{X}_1, \dots, \tilde{X}_{n+N}$, where $\tilde{X}_i = g(X_i)$. The corresponding outcomes of evaluating the policy in simulation are denoted as $f(\tilde{X}_1), \dots, f(\tilde{X}_{n+N})$. Uniformly at random, we select n of those environments in which to conduct real trials. Thus, the paired evaluation data comprises of the real-world outcomes and associated simulation predictions: $D_{\text{paired}} = \{(Y_i, f(\tilde{X}_i))\}_{i=1}^{n}$. The remaining number of additional simulation evaluations N exceeds the number of real-world evaluations n, and these are denoted as $D_{\text{sim}} = \{f(\tilde{X}_i)\}_{i=n+1}^{n+N}$. The i^{th} data sample is defined as:

$$\Delta_i = \frac{n+N}{n}\big(Y_i - f(\tilde{X}_i)\big)\xi_i + f(\tilde{X}_i), \quad i \in \{1, \dots, n+N\}, \tag{3}$$

where ξ_i is an indicator of whether $(Y_i, f(\tilde{X}_i)) \in D_{\text{paired}}$. Taking the sample mean of Equation (3) results in the uniform PPI estimator [23]:

$$\mu_{\text{PPI}}^{\text{unif}} = \underbrace{\frac{1}{n}\sum_{i=N}^{n+N}(Y_i - f(\tilde{X}_i))}_{\text{Rectifier}} + \underbrace{\frac{1}{n+N}\sum_{i=1}^{n+N} f(\tilde{X}_i)}_{\text{Simulation evaluations}}, \tag{4}$$

where $\mu_{\text{PPI}}^{\text{unif}}$ is an unbiased estimate of the true mean. The first term is referred as the rectifier, since it adjusts the bias in simulation predictions. For some significance level α, a confidence interval for μ^* is computed from sampled evaluation data using non-asymptotic methods for mean estimation via the Waudby-Smith and Ramdas (WSR) algorithm [22], which just requires the bounds of the random variable to be specified a priori. That is, we pass the sampled evaluation data from Equation (3) as input to the WSR procedure with significance level α. This method is denoted as **SureSim** (**S**calable and **R**eliable Policy **E**valuation with **Sim**ulation). To hedge in instances when the real2sim correlation is low, we present a variant termed **SureSim-UB** which is a confidence interval resulting from a union bound of **SureSim** (computed at budget $\frac{3\alpha}{4}$) and **Classical** (at budget $\frac{\alpha}{4}$).

SureSim (2-Stage). Prediction-powered inference was originally introduced with a two-stage setup for data sampling [17]. This approach considers two sets of environments drawn i.i.d from $\mathcal{D}_{\text{env}}$: the first set consists of a small number n of environments for which we collect both real and paired simulation evaluations, and the second set consists of a large number N of additional simulations. The PPI estimator for mean estimation is defined as:

$$\mu_{\text{PPI}} = \underbrace{\frac{1}{n}\sum_{i=1}^{n}(Y_i - f(\tilde{X}_i))}_{\text{Rectifier}} + \underbrace{\frac{1}{N}\sum_{i=1}^{N} f(\tilde{X}_i)}_{\text{Additional simulation evaluations}}, \tag{5}$$

where μ_{PPI} is also an unbiased estimate of μ^*. If we assume real and simulation scores to lie in the range $[0, 1]$, the rectifier is bounded between $[-1, 1]$. For a significance level α, a confidence interval on μ^* is computed by separately deriving confidence intervals for the rectifier at some significance level $\delta < \alpha$ and for the additional simulation data at significance $\alpha - \delta$, and taking their Minkowski

sum [17].[4] For mean estimation, it can be proven that the true mean μ^* lies in the resulting confidence interval with probability $1-\alpha$ [17]. To obtain finite sample guarantees, the rectifier and prediction confidence intervals are computed using WSR [22]. In this two-stage approach, the bloating of the rectifier bounds coupled with the small number n of rectifier samples introduces inefficiencies in the resulting confidence interval. Similar to **SureSim-UB**, we also introduce a hedged version of this method termed **SureSim-UB (2-Stage)**.

Theorem 1. ***SureSim*** *and its variants give a finite-sample valid confidence interval CI that satisfies Equation (2).*

Proof. By construction of the real2sim function, the prediction rule is the functional composition $f \circ g : \mathcal{X} \to [0,1]$. Under the assumption that $\{X_i\}_{i=1}^{n+N}$ are drawn i.i.d from the task and distribution $\mathcal{D}_{\text{env}}$, the finite-sample validity of the resulting confidence interval follows directly from [17, 23]. □

Baseline. The problem of efficient mean estimation is ubiquitous in the natural sciences and engineering, and the associated theory has received significant attention in statistics and machine learning for a broad set of problem settings. The natural comparison in our context is with respect to methods designed to incorporate proxy variables in order to boost the effective sample size, especially when the acquisition of evaluation data is slow or expensive. Thus, our primary baseline, termed as **Classical**, is to compute finite-sample confidence intervals without augmenting simulation, and apply the non-asymptotic WSR procedure (see Algorithm 4) directly on real evaluations. These intervals represent the standard procedure for mean estimation and interval generation; they also act as an ablation with respect to the incorporation of proxy data in the subsequent methods.

Related Methods. While we primarily compare to **Classical** since it provides a finite-sample guarantee, we also implement and discuss the control variates procedure (denoted **Control Variate**) [24] in the sim2sim setting. We do not consider this as a baseline for the hardware experiments because it is not provably Type-I error controlling in finite samples. Therefore, the practitioner cannot know for their problem that the resulting confidence interval from [24] contains the true mean at a specified level of confidence. In particular, this procedure utilizes the empirical correlation of the simulation evaluations to make optimization-based reductions to the mean estimation, at the expense of looser dependence on the confidence level α. Specifically, the paired samples yield a variance estimate for the control variate estimator, and this variance estimate is subsequently employed in Chebyshev's inequality to derive a confidence interval for the mean [24]. However, in finite-sample settings, the variance estimate may be biased for small n, and even unbiased constructions (such as through data splitting) need not upper-bound the *true* variance, a requirement for Chebyshev's inequality.

4 Robot Experiments

We illustrate our method on pick-and-place tasks and evaluate policy generalization across diverse pick object types and initial conditions. Specifically, we seek to address the following questions: 1) How tight are our confidence intervals relative to the baselines? What benefit does this translate to in terms of real-world evaluation cost? 2) How does the confidence interval width decrease as we scale the number of simulation evaluations?

Real2Sim Pipeline. For evaluating object generalization, we gathered around 120 objects, most of which are toy kitchen items, shown in Figure 2. To carry out paired evaluations in simulations, 3D models for these objects were obtained from a single image of the object using an off-the-shelf tool Meshy. These 3D models were scaled to match real-world dimensions, and their pose was set according to real-world experiments. To construct the additional simulation dataset, we draw over 2100 objects from the RoboCASA repository [25], which includes assets from Objaverse [26] and assets generated using a text-to-3D model Luma AI. We filter out assets that are not semantically or

[4] The allocation of risk to δ and $\alpha - \delta$ can be approximately optimized. In practical settings, using $\delta \approx 0.9\alpha$ is a reliable heuristic.

geometrically equivalent—objects whose category or shape lacks a counterpart in the real-world set (e.g., plates).

In an ideal setting, we would have access to a large-scale repository of real-world objects paired with corresponding 3D models—akin to the YCB dataset [27], but expanded to include thousands of objects. This would allow us to uniformly sample a subset of objects for real-world and paired simulation evaluation, while using the remaining objects exclusively for additional simulation evaluations. However, these large datasets do not at present exist, and therefore, we take these 120 objects are taken to approximate the real-world distribution of objects that we wish to evaluate our policy on.

Figure 2: *Left:* Evaluation setup illustrating pick object initial conditions. *Right:* Objects used for real-world and paired simulation evaluations.

Experimental Setup. We evaluate policies on a Franka Panda robot equipped with a wrist-mounted RealSense D405 and a Logitech C920 third-person camera. For simulation, we replicate the setup in ManiSkill3 [10]. Further details on the simulator setup are given in the Appendix.

Policies. We evaluate two policies: i) a single-task diffusion policy [19] trained from scratch and ii) a generalist policy π_0 [20], fine-tuned on multiple objects. Our diffusion policy is trained on 200 demonstrations of a single task — to pick up a tomato and place it in a plate. The training distribution comprises of the tomato and the plate being placed randomly in a 30cm-by-40cm space. Though trained on a single object, we evaluate this diffusion policy on its generalization to multiple objects. We finetune π_0 for 7 different objects according to the language instruction "put <object> into the box" with 40 demonstrations for each object. In the fine-tuning demonstrations, the *pick* object is randomly placed in a 10cm-by-20cm grid, while the box is placed at roughly the same xy-position. After each inference step, the open-loop action horizon was set to full action chunk size of 30.

Evaluation Metrics. For each real object, we rollout diffusion policy and π_0 at five different initial conditions of the pick object as shown in Figure 2. Each rollout is assigned a partial evaluation score: 0 for no grasp, 0.25 for a failed grasp (object slips), 0.5 for a successful grasp, 0.75 for successful grasp but unsuccessful release over the place object, and 1 for complete task success. In simulation, we record a partial success score as follows: 0 for no grasp, 0.5 for successful grasp, and 1 for complete task success.

Real-Simulation Evaluation Gap. Additionally, we discuss the real-simulation evaluation gap for robot manipulation, and share a few insights to mitigate this. Crucially, for mean estimation, this gap manifests in the variance of the rectifier, which represents the difference in the real and simulation outcomes on the paired set D_{paired}. That is, a high rectifier variance corresponds to low correlation on D_{paired} and a high real-simulation gap, while a low variance corresponds to high correlation and a small gap. Depending on the evaluation criteria used for constructing D_{paired}, well-known sources of the real-simulation gap — such as visual and dynamics discrepancies — can reduce the reliability of simulation in predicting real outcomes and increase rectifier variance. For stochastic policies (e.g., diffusion policy which has randomness in the denoising process), this mismatch is further exacerbated by inconsistencies in policy seeding between real and simulated runs. For example, if we evaluate diffusion policy using a discrete evaluation metric over a set of initial conditions by pairing a single hardware trial with a simulation evaluation at the same initial condition, we are unlikely to see a high correlation on the paired set of evaluations. For the same real-simulation experimental setup, the rectifier variance can vary with the task and evaluation criteria, the policy under evaluation, and the axis of generalization considered in the distribution $\mathcal{D}_{\text{env}}$. Together, these factors can lead to low correlation in paired evaluations, thereby diminishing the predictive utility of simulation and undermining the advantage of large-scale simulation for trustworthy inference on real performance.

To address this issue, we implement the following measures: (1) we ensure that both real-world and simulation evaluations use the same random seed, and (2) in simulation, we sample 20 initial conditions from a 2cm-by-2cm box of the the real (x, y) initial condition, execute the policy for each, and average the results to obtain a more robust estimate of the simulation counterpart. These measures are designed to mitigate the real-simulation gap without requiring additional real evaluations.

4.1 Real2Sim Robot Experiments

Diffusion Policy. First, we evaluate a single-task diffusion policy on a distribution of various types of pick objects. For each real object X_i, we get the real label Y_i by taking the average of partial scores of trials conducted at 5 initial conditions shown in Figure 2. The paired evaluation score $f(\tilde{X}_i)$ is the average of simulation partial scores averaged over 100 simulation initial conditions corresponding to the 5 real-world initial conditions. On average, the correlation on the paired dataset is 0.72. For the additional simulation objects, we choose the Objaverse split of RoboCASA objects. For the following results, we use 100 random samples[5] of $n = 60$ paired evaluations and up to $N = 700$ additional evaluations. For the **SureSim (2-Stage)** family, the rectifier significance level is set to $\delta = 90\%$ of the the total significance level. All methods are given a significance level of $\alpha = 0.1$.

Figure 3 illustrates the size of confidence interval widths as we scale-up simulation. At just 100 additional simulations, **SureSim** tightens the confidence interval with respect to **Classical** as simulation is scaled up further. In cases where the confidence interval is not truncated at 0 or 1, the rectifier interval width corresponds to a lower bound on the confidence interval width as the number of additional simulations increase. Here, the rectifier interval width is computed from finite-sample confidence intervals for the rectifier at $\delta = 0.09$ level of significance, and is determined by the rectifier variance. **SureSim** and **SureSim-UB** in Figure 3 approach this lower bound relatively quickly, indicating efficient usage in incorporating simulation data up to the limit imposed by the real-simulation gap. At $N = 700$, the mean interval width of **SureSim** is 0.16 which is a decrease of 14.4% compared to the interval width of length 0.187 for the **Classical** method. The **SureSim (2-Stage)** family has a slower decrease in interval width with scaling simulations as compared to **SureSim** family due to: i) the two-stage procedure introducing inefficiencies in separately computing confidence intervals for the rectifier and additional simulations, and ii) the significance level allocated to the simulation confidence interval ($\alpha - \delta = 0.01$), requiring further simulation trials.

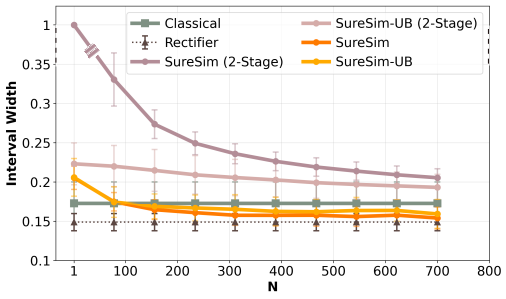


Figure 3: Evaluating Diffusion Policy with $n = 60$ paired trials and up to 700 additional simulations.

We study the advantage of our methods over hardware-only evaluations as follows. For each method, we compute a confidence interval at $n = 60$ samples, and iteratively search over the number of samples n given to **Classical** until the resulting confidence interval is tighter than the method's interval. Figure 4 illustrates the resulting savings, where the **SureSim** method family yields over 25% savings with respect to real-only evaluation.

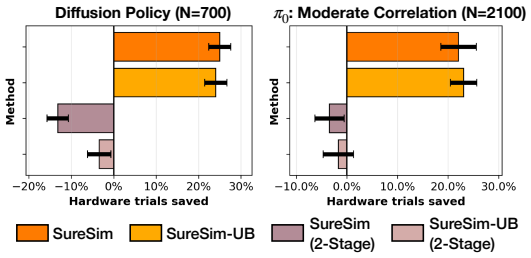


Figure 4: Average number of hardware trials saved compared to **Classical** over 100 random draws.

Finetuned π_0. We present two examples of evaluating π_0, where we consider a joint distribution

[5] In practice, this amounts to 100 random re-samplings of 60 objects (without replacement) from the bank of 120 real objects

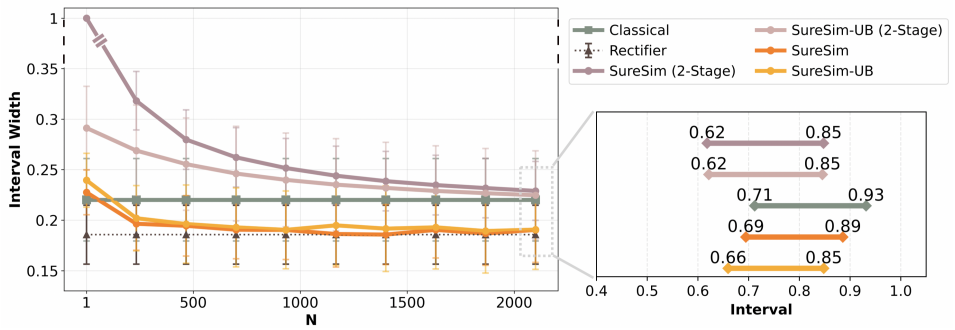


Figure 5: **How does interval width decrease with scaling simulations under moderate correlation?** This figure reports results for π_0 evaluated on initial conditions $\{1, 2, 3, 4\}$ for $n = 60$ paired trials and over 2100 additional simulations.

over objects and initial conditions. In the first case, an object is randomly selected and placed at an initial condition sampled from $\{1, 2, 3, 4\}$, as shown in Figure 2, which yields a moderate real-to-sim correlation. The real evaluation label for a specific object and initial condition is recorded according to the partial score metric, and the paired simulation records the average of simulation partial scores on the perturbed set of initial conditions corresponding to the real initial condition. In the second case, the initial condition is sampled from $\{1, 2, 3\}$, which produces a low correlation. We present both cases to evaluate our methods under contrasting real-simulation correlation regimes. The low correlation case is deferred to the Appendix.

Moderate Correlation. In Figure 5, we report average interval widths. Confidence intervals will vary in width and location for different draws of data from the same distribution; we illustrate one representative confidence interval in Figure 5. The real-simulation Pearson correlation ρ is 0.59 on average on the paired evaluation set. The number of additional simulations is sufficient for the **SureSim** family to approach the rectifier lower bound and result in an advantage over **Classical**. Further scaling up simulation would address the two-stage inefficiency in the **SureSim (2-Stage)** family. However, the **SureSim** family converges by $N = 500$ additional simulations, indicating efficiency with scaling simulations compared to the **SureSim (2-Stage)** family. As seen in Figure 4, this leads to over a 20% decrease in real trials on average. In future work, we study further improvements to these finite-sample results by fine-tuning simulation.

Across all Real2Sim experiments, we observe that **SureSim** yields the greatest savings in terms of hardware trials saved and reduction in interval width. **SureSim** converges relatively quickly in the number of additional simulations in comparison to the two-stage methods. Furthermore, as we scale the number of simulations, the confidence intervals from our methods do not shrink to arbitrarily small widths. This controlled behavior is desirable, as it prevents overconfidence and ensures robust estimation of the mean. The gain from large-scale simulation depends on the correlation between paired real and simulated evaluations, which determines the rectifier variance. As shown in the asymptotic setting [17], combining real and simulated data is effective only when the rectifier variance is smaller than the variance of real evaluations—a condition that remains necessary in the non-asymptotic regime before committing substantial effort to large-scale simulation.

Discussion. In summary, we present a finite-sample valid approach to augmenting real-world trials with large-scale simulation for robot policy evaluation. Our demonstrations focused on the utility of this method to robot manipulation tasks over a joint distribution of tasks and initial conditions. Our method results in over 20% savings compared to real-world only evaluation. A few directions for future work include: i) actively sampling real environments for evaluation as opposed to a batch evaluation approach, and ii) proposing efficient fine-tuning methods for simulation to address scenarios with low correlation.

References

[1] J. Gao, S. Belkhale, S. Dasari, A. Balakrishna, D. Shah, and D. Sadigh. A taxonomy for evaluating generalist robot policies. *arXiv preprint arXiv:2503.01238*, 2025.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[3] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[4] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

[5] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

[6] A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

[7] D. Snyder, A. J. Hancock, A. Badithela, E. Dixon, P. Miller, R. A. Ambrus, A. Majumdar, M. Itkina, and H. Nishimura. Is your imitation learning policy better than mine? policy comparison with near-optimal stopping. *arXiv preprint arXiv:2503.10966*, 2025.

[8] H. Kress-Gazit, K. Hashimoto, N. Kuppuswamy, P. Shah, P. Horgan, G. Richardson, S. Feng, and B. Burchfiel. Robot learning as an empirical science: Best practices for policy evaluation. *arXiv preprint arXiv:2409.09491*, 2024.

[9] J. Barreiros, A. Beaulieu, A. Bhat, R. Cory, E. Cousineau, H. Dai, C.-H. Fang, K. Hashimoto, M. Z. Irshad, M. Itkina, et al. A careful examination of large behavior models for multitask dexterous manipulation. *arXiv preprint arXiv:2507.05331*, 2025.

[10] S. Tao, F. Xiang, A. Shukla, Y. Qin, X. Hinrichsen, X. Yuan, C. Bao, X. Lin, Y. Liu, T. kai Chan, Y. Gao, X. Li, T. Mu, N. Xiao, A. Gurha, Z. Huang, R. Calandra, R. Chen, S. Luo, and H. Su. Maniskill3: Gpu parallelized robotics simulation and rendering for generalizable embodied ai. *arXiv preprint arXiv:2410.00425*, 2024.

[11] J. Jang, S. Ye, Z. Lin, J. Xiang, J. Bjorck, Y. Fang, F. Hu, S. Huang, K. Kundalia, Y.-C. Lin, et al. Dreamgen: Unlocking generalization in robot learning through neural trajectories. *arXiv e-prints*, pages arXiv–2505, 2025.

[12] J. Quevedo, P. Liang, and S. Yang. Evaluating robot policies in a world model. *arXiv preprint arXiv:2506.00613*, 2025.

[13] X. Li, K. Hsu, J. Gu, K. Pertsch, O. Mees, H. R. Walke, C. Fu, I. Lunawat, I. Sieh, S. Kirmani, et al. Evaluating real-world robot manipulation policies in simulation. *arXiv preprint arXiv:2405.05941*, 2024.

[14] X. W. M. Team. 1x world model: Evaluating bits, not atoms. Technical report, 1X, 2025.

[15] Z. Zhou, P. Atreya, Y. L. Tan, K. Pertsch, and S. Levine. Autoeval: Autonomous evaluation of generalist robot manipulation policies in the real world. *arXiv preprint arXiv:2503.24278*, 2025.

[16] N. Pfaff, E. Fu, J. Binagia, P. Isola, and R. Tedrake. Scalable real2sim: Physics-aware asset generation via robotic pick-and-place setups. *arXiv preprint arXiv:2503.00370*, 2025.

[17] A. N. Angelopoulos, S. Bates, C. Fannjiang, M. I. Jordan, and T. Zrnic. Prediction-powered inference. *Science*, 382(6671):669–674, 2023.

[18] A. N. Angelopoulos, J. C. Duchi, and T. Zrnic. PPI++: Efficient prediction-powered inference. *arXiv preprint arXiv:2311.01453*, 2023.

[19] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[20] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[21] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[22] I. Waudby-Smith and A. Ramdas. Estimating means of bounded random variables by betting. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 86(1):1–27, 2024.

[23] T. Zrnic and E. Candes. Active statistical inference. In *International Conference on Machine Learning*, pages 62993–63010. PMLR, 2024.

[24] R. Luo, H. Yang, M. Watson, A. Sharma, S. Veer, E. Schmerling, and M. Pavone. Leveraging correlation across test platforms for variance-reduced metric estimation. *arXiv preprint arXiv:2506.20553*, 2025.

[25] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. *arXiv preprint arXiv:2406.02523*, 2024.

[26] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153, 2023.

[27] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar. Yale-cmu-berkeley dataset for robotic manipulation research. *The International Journal of Robotics Research*, 36(3):261–268, 2017.

[28] M. Heo, Y. Lee, D. Lee, and J. J. Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. *The International Journal of Robotics Research*, page 02783649241304789, 2023.

[29] J. Luo, C. Xu, F. Liu, L. Tan, Z. Lin, J. Wu, P. Abbeel, and S. Levine. Fmb: a functional manipulation benchmark for generalizable robotic learning. *The International Journal of Robotics Research*, 44(4):592–606, 2025.

[30] B. Yang, D. Jayaraman, J. Zhang, and S. Levine. Replab: A reproducible low-cost arm benchmark for robotic learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8691–8697. IEEE, 2019.

[31] N. Khargonkar, S. H. Allu, Y. Lu, B. Prabhakaran, Y. Xiang, et al. Scenereplica: Benchmarking real-world robot manipulation by creating replicable scenes. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8258–8264. IEEE, 2024.

[32] J. Collins, M. Robson, J. Yamada, M. Sridharan, K. Janik, and I. Posner. Ramp: A benchmark for evaluating robotic assembly manipulation and planning. *IEEE Robotics and Automation Letters*, 9(1):9–16, 2023.

[33] D. Pickem, P. Glotfelter, L. Wang, M. Mote, A. Ames, E. Feron, and M. Egerstedt. The robotarium: A remotely accessible swarm robotics research testbed. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1699–1706. IEEE, 2017.

[34] G. Zhou, V. Dean, M. K. Srirama, A. Rajeswaran, J. Pari, K. Hatch, A. Jain, T. Yu, P. Abbeel, L. Pinto, et al. Train offline, test online: A real robot learning benchmark. *arXiv preprint arXiv:2306.00942*, 2023.

[35] Z. Liu, W. Liu, Y. Qin, F. Xiang, M. Gou, S. Xin, M. A. Roa, B. Calli, H. Su, Y. Sun, et al. Ocrtoc: A cloud-based competition and benchmark for robotic grasping and manipulation. *IEEE Robotics and Automation Letters*, 7(1):486–493, 2021.

[36] S. Bauer, M. Wüthrich, F. Widmaier, A. Buchholz, S. Stark, A. Goyal, T. Steinbrenner, J. Akpo, S. Joshi, V. Berenz, et al. Real robot challenge: A robotics competition in the cloud. In *NeurIPS 2021 Competitions and Demonstrations Track*, pages 190–204. PMLR, 2022.

[37] P. Atreya, K. Pertsch, T. Lee, M. J. Kim, A. Jain, A. Kuramshin, C. Eppner, C. Neary, E. Hu, F. Ramos, et al. Roboarena: Distributed real-world evaluation of generalist robot policies. *arXiv preprint arXiv:2506.18123*, 2025.

[38] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[39] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[40] Y. Tassa, Y. Doron, A. Muldal, T. Erez, Y. Li, D. d. L. Casas, D. Budden, A. Abdolmaleki, J. Merel, A. Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[41] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, et al. Isaac gym: High performance gpu-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[42] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv:2009.12293*, 2020.

[43] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. doi: 10.1109/LRA.2020.2974707.

[44] W. Pumacay, I. Singh, J. Duan, R. Krishna, J. Thomason, and D. Fox. The colosseum: A benchmark for evaluating generalization for robotic manipulation. *arXiv preprint arXiv:2402.08191*, 2024.

[45] K. Zheng, X. Chen, O. C. Jenkins, and X. Wang. Vlmbench: A compositional benchmark for vision-and-language manipulation. *Advances in Neural Information Processing Systems*, 35: 665–678, 2022.

[46] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

[47] M. Yang, Y. Du, K. Ghasemipour, J. Tompson, D. Schuurmans, and P. Abbeel. Learning interactive real-world simulators. *arXiv preprint arXiv:2310.06114*, 1(2):6, 2023.

[48] T. Brooks, B. Peebles, C. Holmes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1(8):1, 2024.

[49] N. Agarwal, A. Ali, M. Bala, Y. Balaji, E. Barker, T. Cai, P. Chattopadhyay, Y. Chen, Y. Cui, Y. Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[50] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, and D. Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics and Automation Letters*, 5(4):6670–6677, 2020.

[51] A. Majumdar, M. Sharma, D. Kalashnikov, S. Singh, P. Sermanet, and V. Sindhwani. Predictive red teaming: Breaking policies without breaking robots. *arXiv preprint arXiv:2502.06575*, 2025.

[52] M. O'Kelly, A. Sinha, H. Namkoong, R. Tedrake, and J. C. Duchi. Scalable end-to-end autonomous vehicle testing via rare-event simulation. *Advances in neural information processing systems*, 31, 2018.

[53] J. A. Vincent, H. Nishimura, M. Itkina, P. Shah, M. Schwager, and T. Kollar. How generalizable is my behavior cloning policy? a statistical approach to trustworthy performance evaluation. *IEEE Robotics and Automation Letters*, 2024.

5 Appendix

5.1 Related Work

Real-world Policy Evaluation. As robot foundation models become more capable at performing complex manipulation tasks, statistically rigorous evaluation of policies is essential for accurately understanding model capabilities while also guiding further progress in the field. A comprehensive list of best practices for evaluating robot policies is given in [8]. While real-world evaluation is expensive, it remains the gold-standard for assessing policy performance, and has driven the significant efforts to establish robotic benchmarks by defining standard tasks and environments, and building replicable robot setups to ensure standardized, reproducible policy evaluation [28, 29, 30, 31, 32]. Real-world evaluation is costly, largely because it requires substantial human effort to record outcomes and reset the environment between trials, and because such evaluations are difficult to parallelize. Furthermore, instead of having each lab investing in identical robot setups, cloud-based evaluation platforms have been established [33, 34, 35, 36, 15]. For example, AutoEval [15] is an open-access remote evaluation platform that autonomously evaluates generalist robot policies by automatically classifying the outcomes of trials and resetting the environment using finetuned robot foundation models. To alleviate this real-world evaluation cost, recent efforts include establishing a community-wide network of evaluators, allowing for a distributed and unbiased pairwise comparison of robot policies across a variety of scenes and tasks [37] on the DROID [38] robot platform. In summary, hardware evaluations alone are insufficient; large-scale simulation complements both small-scale real evaluations targeting a single axis of generalization and large-scale distributed evaluation.

Policy Evaluation in Simulation. Physics-based simulation benchmarks [39, 40, 41, 10, 42, 25, 43, 44, 45, 46] offer a reproducible and cheaper alternative to real-world robot policy evaluation. To mitigate visual and dynamics discrepancies, SIMPLER [13] uses system identification and real2sim image editing methods. Recent developments in action-conditioned video world models [11, 47, 48] promise a more photo-realistic alternative to simulate real-world interactions than physics-based counterparts, and offer faster scene initialization using a text, image, or video prompt [49]. The use of such video world models for policy evaluation is nascent but of growing interest to the research community [14, 12]. However, faithfully modeling real-world dynamics remains a challenge, and generative simulators are additionally susceptible to hallucinations. Despite these challenges, simulation-based evaluation remains valuable even when simulators do not perfectly reproduce real-world behavior. In visual navigation, Kadian et al. [50] test whether simulation-derived policy rankings predict real-world rankings using the sample Pearson correlation coefficient. Such metrics have been adopted in robotic manipulation, showing that simulation evaluation can correlate well

with real performance evaluations for policy rankings [14, 13] or evaluating policy performance in different environmental factors [44, 28]. Due to these challenges in accurately predicting real-world interactions, a new line of work on predictive red-teaming [51] has emerged. This approach predicts whether a policy will succeed in a new environment without actually rolling out the policy, and has shown a strong correlation between the rankings of real and predicted performance across various environmental factors. In contrast to these methods, our approach provides assurances on real-world mean policy performance.

Statistically Confident Policy Evaluation. In end-to-end self-driving applications, scalable simulation-based evaluation using importance sampling was used to provide statistical confidence on the safety of a self-driving policy [52]. However, real-world evaluation remains gold-standard since it is difficult to model the real distribution of environments in simulation, which can lead to a bias in the resulting guarantees. In manipulation, limited by real-world evaluation costs and the large diversity of environments to evaluate in, researchers typically compare policy performance using only 20-30 real-world trials. However, such small sample sizes are insufficient to draw statistically significant conclusions in policy comparisons [7]. Recognizing this need for reliable policy evaluation, a recent study compares generalist large behavior models to single-task policy counterpart using rigorous statistical evaluation methods, incorporating A/B real-world testing, and comprehensive real-world and simulation trials with robust statistical analysis [9]. Sequential testing frameworks for policy comparison [7] can save evaluators a considerable number of real-world evaluations (up to 200 in some instances) while maintaining statistical validity under anytime stopping. Additionally, for generalist policies, crowd-sourcing policy evaluations [37] can provide accurate policy comparisons by distributing large evaluation effort across multiple evaluators while also tackling a greater diversity of environments and tasks. In addition to policy comparison, it is also important to know the individual performance of each policy. For binary success criteria, [53] provide optimal confidence intervals from real evaluations.

Similarly, one can obtain confidence intervals from real evaluations using non-binary success criteria using concentration inequalities (e.g., Hoeffding [21]). However, we desire the resulting confidence intervals to be as tight as possible without imposing a large burden on real-world evaluation cost. To scalably achieve this, we seek to combine real and simulation evaluations. Finally, concurrent to our work is the application of control variates to combine simulation evaluation with real-world logged data for evaluation in self-driving applications [24]. Though similar at a conceptual level, our work differs in two important aspects. First, we present finite-sample confidence bounds on real-world performance of manipulation policies that are tighter than existing baselines. Secondly, we demonstrate the idea of combining real-world and simulation evaluations on robotic manipulation, which faces a unique set of challenges — robot policies can be sensitive to small perturbations in the environment, and the robot and environment state are more tightly interdependent.

5.2 Experiments

Simulator Setup. Our simulation setup replicates the real setup constructing a customized Franka Panda robot in which the default gripper is replaced with the 3D model used in our real-world experiments. The robot base pose in simulation is aligned with the real robot through manual calibration. Similarly, we transfer the camera calibration parameters from the real setup—covering both the wrist-mounted camera and the third-person camera—to their counterparts in simulation. We use the same control frequency as the robot in the real world. The workspace table is constructed by scripting a table-like mesh and overlaying it with the texture of the real table. For the background, we import a real-world mesh obtained via 3D scanning. Finally, we use the default shader with shadows enabled to strike a balance between simulation speed and visual quality, and tune lighting parameters until policy performance in simulation on randomly selected initial conditions is as high as possible.

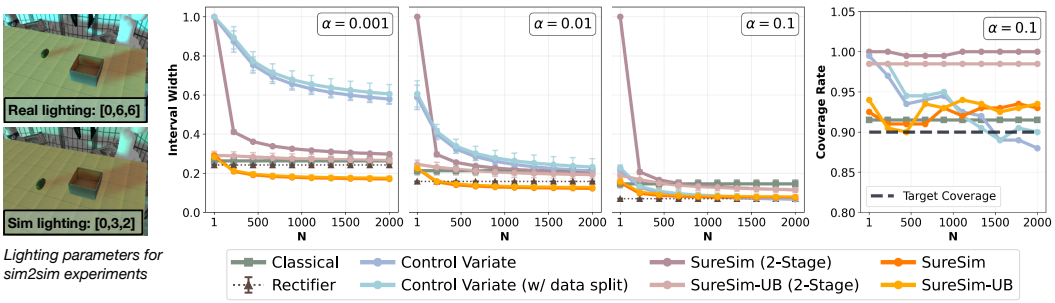


Figure 7: **Sim2Sim Experiments.** Confidence interval widths for 100 random draws of data at $n = 100$ paired trials with up to $N = 2000$ additional simulations. In each draw, we hold out 400 randomly sampled environments for validating coverage. Our methods always beat the classical baseline irrespective of increasing confidence levels.

Real2Sim experiments on finetuned π_0 *(Moderate Correlation).* Although our object repository is limited to approximately 2100 objects, we conduct a sanity check in Figure 6 by sampling additional simulations with replacement, up to $N = 50,000$. While the rectifier interval width is limited by the difference in the real and simulation outcomes $(Y_i - f(\tilde{X}_i))$ on a small number of evaluations, additional simulation evaluations can be scaled up in the two-stage methods to reduce interval width. We observe that **SureSim (2-Stage)** progressively approaches the rectifier lower bound as the number of simulations increases. **SureSim** and **SureSim-UB** more efficiently converge to the rectifier lower bound within 5000 additional simulations, illustrating an upper bound on the benefit that additional simulation can provide given the real–simulation gap.

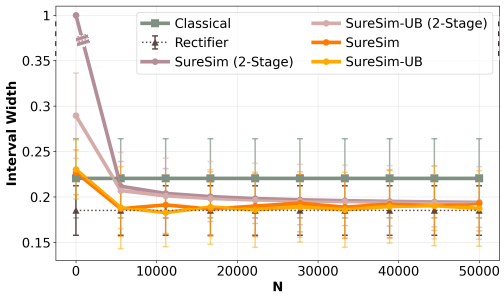


Figure 6: Evaluating π_0 at $n = 60$ and scaling simulations up to $N = 50000$.

Real2Sim experiments on finetuned π_0 *(Low Correlation).* In Figure 8, we present results for a low correlation case, where initial conditions are sampled from $\{1, 2, 3\}$ shown in Figure 2. Empirically, the finetuned π_0 demonstrates strong generalization to diverse object types despite being finetuned on only 7 objects. The initial conditions $\{1, 2, 3\}$ achieve higher success rates across object types compared to initial condition 4. While these initial conditions are correspondingly easy and difficult in simulation, the predictive signal from simulation is insufficient to capture subtle variations in real-world performance, resulting in a low correlation of around -0.05.

This results in qualitatively different behavior. As seen in Figure 8, none of our methods beat **Classical**. This is unsurprising because the there is nothing to infer about real policy performance from simulation. In particular, as listed in Table 1, the sample variance on real data is smaller than the sample rectifier variance. As a result, scaling up simulation does not help in reducing the variance in our estimates of the true mean (Equations (2) and (4)).

5.3 Sim2Sim Experiments

To illustrate coverage rate of confidence intervals, we run simulation-simulation experiments in which we can use a larger number of simulation evaluations as heldout samples to compute the "true" mean and validate coverage. As illustrated in Figure 7, we use one of the simulator settings as the "real" environment and use the other as "simulation. We evaluate finetuned π_0 on 3D object models scanned from real objects (see Figure 2) as well RoboCASA. Once again, we consider a joint distribution over objects and initial conditions $\{1, 2, 3, 4\}$, and evaluate each trial according to the simulation

partial score metric. The paired evaluation set has a very high correlation of around 0.97. We use 400 randomly drawn environments as heldout samples for validating coverage[6].

For these experiments, we also report intervals for the **Control Variate** method, including the standard implementation presented in [24] as well as a data split version for an unbiased variance estimate. For data splitting, we use 20% of the paired data for variance estimation and the remaining for inference. As shown in Figure 7, at $\alpha = 0.1$, all methods beat **Classical**, with the **SureSim** family efficiently converging to the rectifier interval width. As the significance level decreases, the **Control Variate** no longer beats **Classical** while **SureSim** always improves. Based on the prior discussion on the **Control**

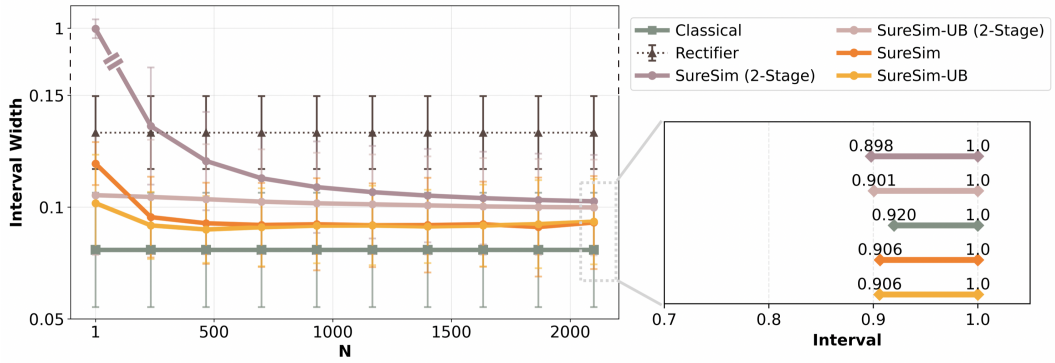


Figure 8: **How does interval width decrease with scaling simulations under low correlation?** π_0 evaluated on initial conditions $\{1, 2, 3\}$ at $n = 60$. *Left*: There is no decrease in interval width with scaling simulations, which is expected given the low correlation between paired real and simulation trials. Due to truncation, the interval widths are smaller than the rectifier interval width (which do not truncate here).

Variate method, we do not expect it to meet the required coverage rate, and as seen in Figure 7, our experiments suggest that the empirical coverage rate can degrade with additional simulation samples. This degradation is cause for caution when interpreting the tightness of interval widths at $\alpha = 0.1$ in Figure 7, which highlights the importance of controlling for Type-1 error.

5.4 Summary Statistics from Experiments

Experiment	n	N	ρ	$\frac{1}{n}\sum_{i=1}^{n} Y_i$	$\frac{1}{n}\sum_{i=1}^{n} f(\tilde{X}_i)$	$\frac{1}{N}\sum_{i=1}^{N} f(\tilde{X}_i)$	$\hat{\sigma}^2_Y$	$\hat{\sigma}^2_{Y-f}$
Diffusion Policy (Real2Sim)	60	700	0.702	0.246	0.188	0.174	0.104	0.054
π_0 (Real2Sim, moderate ρ)	60	2100	0.588	0.825	0.820	0.772	0.138	0.090
π_0 (Real2Sim, low ρ)	60	2100	-0.051	0.983	0.932	0.928	0.014	0.029
π_0 (Sim2Sim, high ρ)	100	2000	0.974	0.751	0.731	0.732	0.116	0.006

Table 1: Summary statistics indicating the number of paired trials n, additional simulation evaluations N, Pearson correlation coefficient ρ, sample real mean $\frac{1}{n}\sum_{i=1}^{n} Y_i$, sample paired simulation mean $\frac{1}{n}\sum_{i=1}^{n} f(\tilde{X}_i)$, sample additional simulation mean $\frac{1}{N}\sum_{i=1}^{N} f(\tilde{X}_i)$, the sample real variance $\hat{\sigma}^2_Y$, and the sample rectifier variance $\hat{\sigma}^2_{Y-f}$. The reported statistics are averaged over 100 draws of data.

[6]Note that at 400 samples, we can still expect some variance in the computed mean. For a very rigorous validation of coverage, we would need to use synthetic data, which we discuss below.

5.5 Algorithms

Algorithm 1: SureSim

Data: Real task and environment distribution $\mathcal{D}_{\text{env}}$, Real-to-sim function g, Policy π, Real metric M, Simulation metric M_{sim}, Significance levels $0 < \delta < \alpha < 1$.
Result: Confidence interval CI on true mean μ^*
Sample environments $X_1 \ldots, X_{n+N} \sim \mathcal{D}_{\text{env}}$ for evaluation
for $i \leftarrow 1$ **to** $n + N$ **do**
  $\tilde{X}_i \leftarrow g(X_i)$ `// Applying the real2sim function`
  $f(\tilde{X}_i) \leftarrow M_{\text{sim}}(\tilde{X}_i, \pi)$ `// Simulation evaluation outcomes for` $\tilde{X}_i$
end
for $i \leftarrow 1$ **to** n **do**
  $Y_i \leftarrow M(X_i, \pi)$ `// Real evaluation outcome for` X_i
end
$D_{\text{paired}} = \{(Y_i, f(\tilde{X}_i))\}_{i=1}^{n}$ `// Collect paired dataset`
$D_{\text{sim}} = \{f(\tilde{X}_i)\}_{i=n+1}^{n+N}$ `// Collect additional simulation evaluations dataset`
if using *uniform* prediction powered inference **then**
  $CI \leftarrow$ UNIFORMPPI$(D_{\text{paired}}, D_{\text{sim}}, f, n, N, \alpha)$
else if using *two-stage* prediction powered inference **then**
  $CI \leftarrow$ 2-STAGEPPI$(D_{\text{paired}}, D_{\text{sim}}, \alpha, \delta)$
return CI

Algorithm 2: Uniform Prediction Powered Inference (UNIFORMPPI)

Input: Paired dataset D_{paired}, Simulation dataset D_{sim}, Sim outcomes f, counts n, N, significance level α
Output: Confidence interval CI
for $i \leftarrow 1$ **to** $n + N$ **do**
  $\xi_i = 1$ if X_i has a real evaluation, else $\xi_i = 0$
  $\Delta_i = f(\tilde{X}_i) + \frac{n+N}{n}(Y_i - f(\tilde{X}_i)) \cdot \xi_i$
end
$D_{\text{unif}} = \{\Delta_i\}_{i=1}^{n+N}$ `// Problem Data`
$CI \leftarrow \text{WSR}(D_{\text{unif}},\ \alpha = \alpha, L = -\frac{n+N}{n}, U = 1 + \frac{n+N}{n})$ `// Single call to WSR`
return CI

Algorithm 3: Two-Stage Prediction Powered Inference (2-STAGEPPI)

Input: Paired dataset D_{paired}, Simulation dataset D_{sim}, levels α, δ
Output: Confidence interval CI
$(f_l, f_u) \leftarrow \text{WSR}(D_{\text{sim}},\ \alpha = \delta, L = 0, U = 1)$ `// Additional Simulation Confidence Interval`
for $i \leftarrow 1$ **to** n **do**
  $\Delta_i = Y_i - f(\tilde{X}_i)$
end
$(R_l, R_u) \leftarrow \text{WSR}(\{\Delta_i\}_{i=1}^{n},\ \alpha = \alpha - \delta, L = -1, U = 1)$ `// Rectifier Confidence Interval`
$CI \leftarrow (f_l - R_u, f_u - R_l)$ `// Union bound`
return CI

Algorithm 4: CI for Mean Estimation (`WSR`)

Data: Data points $\{Z_1, \ldots, Z_n\}$, error level $\alpha \in (0,1)$, range $[L, U]$ such that $Z_i \in [L, U]$.
Result: Confidence interval CI for the mean
for $i \leftarrow 1$ **to** n **do**
  $Z_i \leftarrow (Z_i - L)/(U - L)$ `// Normalize to` $[0,1]$
end
Construct fine grid M_{grid} over $[0,1]$
Initialize set of candidate means $\mathcal{A} \leftarrow M_{\text{grid}}$
for $t \leftarrow 1$ **to** n **do**
  $\hat{\mu}_t \leftarrow \dfrac{0.5 + \sum_{j=1}^{t} Z_j}{t+1}$
  $\hat{\sigma}_t^2 \leftarrow \dfrac{0.25 + \sum_{j=1}^{t} (Z_j - \hat{\mu}_t)^2}{t+1}$
  $\lambda_t \leftarrow \sqrt{\dfrac{2\log(2/\alpha)}{n\hat{\sigma}_{t-1}^2}}$
  for $m \in M_{\text{grid}}$ **do**
    In computing the martingales, we choose the hyperparameter $c = 0.99$ due to its empirical performance
    $M_t^+(m) \leftarrow \Big(1 + \min\big(\lambda_t, \frac{c}{m}\big)(Z_t - m)\Big) M_{t-1}^+(m)$
    $M_t^-(m) \leftarrow \Big(1 - \min\big(\lambda_t, \frac{c}{1-m}\big)(Z_t - m)\Big) M_{t-1}^-(m)$
    $M_t(m) \leftarrow \frac{1}{2}\max\{M_t^+(m), M_t^-(m)\}$ `// Martingale`
    if $M_t(m) \geq 1/\alpha$ **then**
      $\mathcal{A} \leftarrow \mathcal{A} \setminus \{m\}$ `// Remove` m `from set of candidate means`
    end
  end
end
$C_\alpha = \{m(U-L) + L : m \in \mathcal{A}\}$ `// True mean lies in this set with high probability`
$CI = [\max\{0, \min C_\alpha\}, \min\{1, \max C_\alpha\}]$
return *CI*

5.6 Evaluation on Artificial Data

In order to investigate counterfactual properties of the evaluation methods, we test all method using artificial (simulated) data with known statistical properties. This is **not** data generated by a physics-based *robot simulator*, but is rather simulated i.i.d. draws of *scalar random variables* with known statistical properties. We term this data "artificial" in order to avoid any confusion with the the simulator predictions in Section 4.

5.6.1 Value of Artificial Data and Research Questions

Practical estimation problems arise precisely because the investigator does not have access to the true statistical parameter in question (in this case, the mean performance). Thus, when evaluating on real data as in Section 4, we cannot verify whether the confidence intervals we generate – or those generated by any baseline procedure – actually contain the true mean. As such, using artificial data allows for the important step of verifying the theoretical claims of each method in practice, so that they may profitably be used on such problems as may be encountered in, for example, the sciences and engineering. Furthermore, access to the "true parameter labels" for artificial data allow us to efficiently pose hundreds or even thousands of estimation problems reflective of varying contexts, which inform the reader as to the best method for their particular application.

The core additional technical objection this must raise is the degree to which the simulated data fails to represent some data that may be observed by the practitioner; necessarily, it is impossible to

sample from, and validate against, *all distributions* – certainly in finite time. Addressing this problem will be crucial to effective characterization and evaluation.

To the aforementioned ends, we provide the following core analyses via the evaluations on artificial data:

- A justification of the generality of our data generation process with respect to key parameters;
- A discussion of the most informative metrics in evaluating estimation procedures;
- An investigation of the effectiveness of all methods subject to variation in the key parameters;
- A brief discussion and usage guide for the strengths of each method, and interpretable scenarios in which one should likely be preferred to the others.

5.6.2 The Key Parameters and Data Generation

As introduced in Section 2, the robot evaluation problem tackled here is a special case of a more general mean estimation problem in statistics. The canonical Neyman-Pearson framework for understanding these estimation problems relies on several key parameters: the batch size (n) and the significance level (α). As introduced in Section 3, we are interested in using the information contained in proxy signals (e.g., simulators) to effectively increase the sample size of real evaluations. Thus, this procedure *also depends* on the amount of proxy data (N) and the degree to which the simulator is "useful" – informally, the amount of additional information contained in the proxy variables.

This last piece of information is of course key to the investigation. We argue that, consistent with the analysis of control variates methods, the critical measure by which the proxy variable improves nonasymptotic (finite-sample) confidence interval generation is in the variance reduction of the (unbiased) mean estimator. Intuitively, such a reduction tightens the confidence intervals while ensuring Type-1 error control at all data scales. With this in mind, we use as the core "effectiveness measure" the Pearson correlation coefficient, ρ, which is a direct ratio of the real-to-proxy covariance to their geometric mean variance. This intuition is reflected directly in the control variates analysis of [24], particularly with respect to their Theorem 1.

Given the preceding discussion, we intend to investigate the relative advantages of each estimation procedure as a function of the four stated parameters: α, n, N, and ρ. To generate artificial data, we construct artificial real data of size $k(n+N)$ drawn uniformly in the interval $[\max\{0, 2\mu - 1\}, \min\{2\mu, 1\}]$. This enforces a tunable true population mean μ while ensuring that the data is always bounded in $[0, 1]$.[7] The proxy data requires a desired value ρ^*. To generate the artificial proxy data, the real data is copied, shifted to mean μ_{sim}, and then iteratively perturbed by small amounts of random noise or small perfect-signal gradients in order to push the empirical correlation to ρ. Matched and unmatched datasets of respective size n and N are drawn from partitions of the large dataset; for the unmatched data, the real labels are discarded for the purposes of running the algorithms. To save time, this single large dataset can be sampled from repeatedly (i.e., bootstrapped), or new datasets can be generated for each experiment. We opt for the latter, though it is more time-consuming in practice for empirically negligible effects.

5.6.3 A Brief Discussion of Estimation Metrics

The "proper" metrics to report for the problem of mean estimation admits a wide array of context-relevant options. We argue for the metrics herein, and attempt to briefly justify the preference.

5.6.4 The Natural Option: Interval Widths

The ultimate purpose of estimation in our context is to minimize the region of uncertainty in which the true parameter lies (subject to a tunable risk of error); this is a dual result to many "operationalizable"

[7] To avoid unnecessary subtleties around the effects of interval truncation at 0 and 1 on the aggregate interval width metrics, we will in general set the means to be equal to 0.5.

uses, including tests of maximal efficiency and power, certification in the least number of trials, etc. As such, it is unsurprising that reporting interval width directly is the most natural metric, and is our primary metric of choice in this work.

5.6.5 A Caution About Variance Minimization

Another natural metric, albeit one slightly upstream of the intended methodological usage, is estimator variance. This analysis is interchangeable with the interval width (i.e., equivalent under monotonic transformations), but *only when the space of estimators is constrained to be unbiased*. Unbiased estimators overwhelmingly dominate among methods used in practice, but analysis can be misleading when the constraint is not satisfied. A minimum-variance estimator is *essentially meaningless* (for example, the estimator '5' has no variance over the draw of the data); a minimum-variance *unbiased* estimator, on the other hand, can be exceedingly novel and useful.

5.6.6 Dependence on Significance Level

Bounded random variables are a special case of random variables with bounded moments. These random variables are sub-Gaussian, and therefore any optimal interval widths (across data scales) should be able to attain poly-logarithmic dependence on the significance level.

5.7 Results on Artificial Data

We illustrate the three aforementioned themes in evaluation over artificial data. All results will report interval widths as the primary metric (Theme 1), and will discuss the downside of additional metrics via the example cases. Second, the limitations of variance minimization will be illustrated (Theme 2), which will also inform our investigation of each method's coverage (i.e., enforcing the Type-I error control constraint in Equation (2)). Finally, we will sketch the gap in efficiency with respect to confidence or significance level, which has implications for different types of validation settings encountered in practice. To be specific, we will discuss in particular the implications of statistical guarantees in safety-critical certification and evaluation paradigms.

In order to avoid certain distracting or confounding phenomena, we present intervals for data with characteristics designed to highlight the fundamental behavior of the algorithms. Specifically: the real data and simulator data means are set arbitrarily to 0.5 each, in order to minimize instances of truncation of the intervals at 0 or 1. Truncation does not in general benefit any particular method, but it does increase variation in interval widths that can make the results noisier. As these results are purely designed to validate existential and not universal quantifications of algorithm behavior, this choice does not bias the result and discussion.

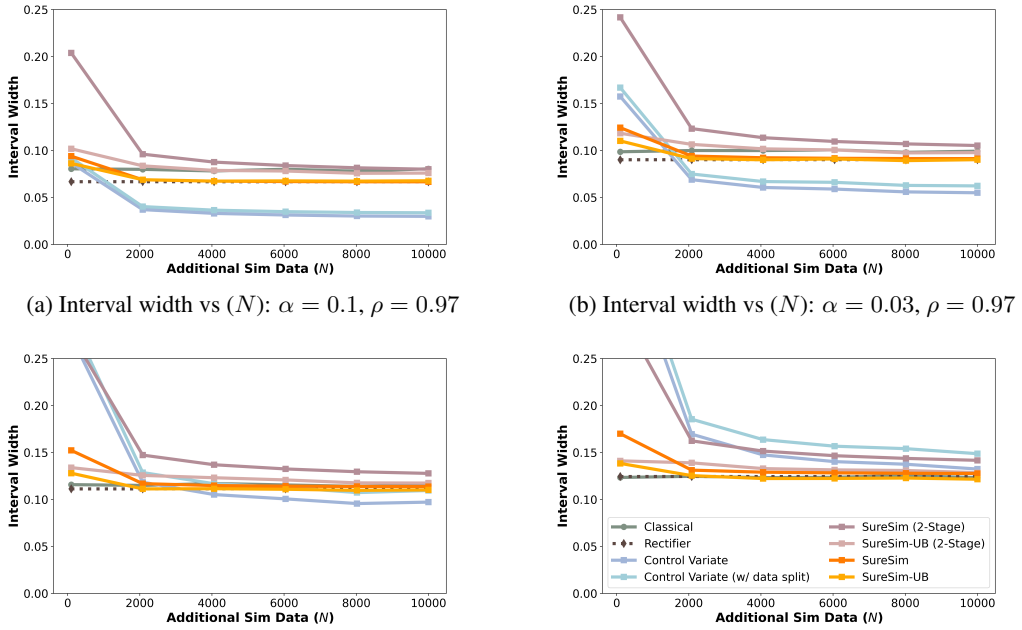
5.7.1 Interval Width as Simulator Data or Correlation Grows

We begin by validating the behavior of each algorithm seen in Section 4. The artificial data is iteratively redrawn for $n = 100$ and varying levels of N up to 10k additional simulation runs.

We first consider a case of relative strength for **Control Variate**, taking a large correlation $\rho = 0.97$ and varying α across approximately two orders of magnitude. As shown in Figure 9, every method has near-monotonic improvement (in expectation) as the amount of additional sim data grows, reflecting the intuition that there must be more 'information' being given to the evaluator. However, the intervals do not asymptote to zero, indicating that, even so, there remains fundamental uncertainty in linking the sim data to the real data (the rectifier uncertainty) that is *irreducible* given a fixed amount of real data. In other words, we cannot trust the sim data to an arbitrary degree, but can still use the data productively to tighten the intervals.

In Figure 11, we generalize these results to variations across the true correlation between real data and simulation. Naturally, higher correlation implies greater signal in the proxy (simulation) data, and therefore more achievable tightening. This also validates the analysis of monotonic and quadratic

Control Variate interval width scaling given in [24]. Note that the numbers in Figure 9 correspond to nearly the right-most points of the curves in Figure 11.[8]



(a) Interval width vs (N): $\alpha = 0.1$, $\rho = 0.97$

(b) Interval width vs (N): $\alpha = 0.03$, $\rho = 0.97$

(c) Interval width vs (N): $\alpha = 0.01$, $\rho = 0.97$

(d) Interval width vs (N): $\alpha = 0.004$, $\rho = 0.97$

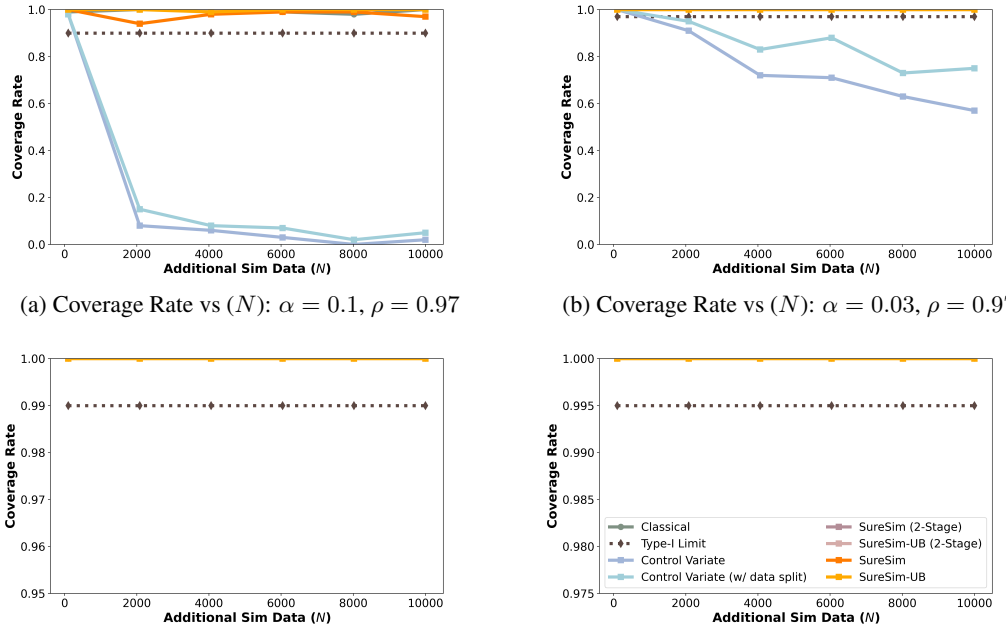
Figure 9: Interval widths on all methods for artificial data. Each plot shows the width against varying N ($n = 100$). Results averaged over 100 independent redraws of data. The desired confidence level increases (α decreases) left-to-right, top-to-bottom. Note that **Control Variate** constructs tighter intervals at large α, but that the interval widths are much more sensitive as α changes. As will be shown in Figure 10, the biased nature of the CV estimator leads to miscoverage in regimes for which its intervals appear to be narrower.

5.7.2 Coverage as Simulator Data or Correlation Grows

We now investigate the second thematic point, on the limitations of variance minimization as a certification for estimation efficiency. The first key comment pertains to the resulting interval coverage, to which we have access by virtue of constructing the artificial data and knowing its key features (including the mean).

Variance minimization is generally synonymous with improving the estimator efficiency – i.e., shrinking the interval width – *but only so long as the intervals enforce validity*. As shown in Figure 10, this is a challenge for **Control Variate**, because the technique is *not unbiased*. Thus, it is susceptible to excessive optimism ("trusting the simulator too much"), which leads to miscoverage when the amount of simulator data grows. As shown, this effect is most pronounced precisely when **Control Variate** is relatively strongest (at larger α). Importantly, for practical problems, the evaluator cannot know whether they are in an excessively optimistic regime; this is precisely the reason for enforcing Equation (2) as a property of the evaluation procedure. As shown, such methods cover uniformly, while being generally efficient (recovering the rectifier variance) across different levels of α and ρ.

[8] This statement is modulo the small differences in α for two of the plots, which differed in order to allow us to illustrate qualitatively different coverage behavior for **Control Variate** in Figure 10.



(a) Coverage Rate vs (N): $\alpha = 0.1$, $\rho = 0.97$

(b) Coverage Rate vs (N): $\alpha = 0.03$, $\rho = 0.97$

(c) Coverage Rate vs (N): $\alpha = 0.01$, $\rho = 0.97$

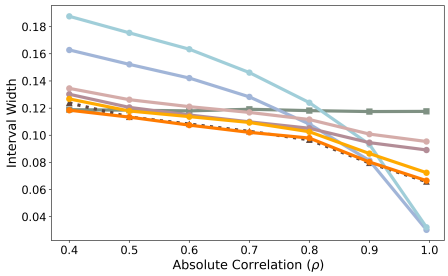
(d) Coverage Rate vs (N): $\alpha = 0.005$, $\rho = 0.97$

Figure 10: Interval coverage rate on all methods for artificial data. Each plot shows the coverage rate against varying N ($n = 100$). Results averaged over 100 independent redraws of data. The desired confidence level increases (α decreases) left-to-right, top-to-bottom. First, every provably nonasymptotically valid method covers in every regime, as expected. By contrast, note that **Control Variate** fails to cover in the top row as the amount of simulation data grows; this is a result of bias in the estimator causing inconsistency. Importantly: it is *precisely when CV intervals become narrower than our methods that they lose validity*. Thus, the only valid instance of empirical improvement of **Control Variate** over our procedure on this data is in the case $\alpha = 0.01$; crucially, however, the practitioner cannot know for their problem whether they are in this regime (as for different problem instances, the critical value of α will differ in general from 0.01). Thus, the practitioner may be in the (narrow) valid and efficient regime, or may be in the invalid (too-optimistic) regime, or in the inefficient regime – and will not be able to distinguish which one.

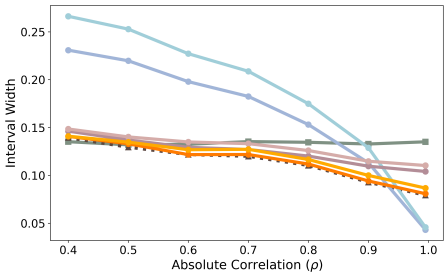
5.7.3 Interval Width versus Significance Level

Finally, we evaluate the effect of changes in significance level on the interval scaling. Because WSR [22] procedures are constrained to bounded (and therefore, sub-Gaussian) random variables, they can recover $\mathcal{O}(\log \frac{1}{\alpha})$ scaling as $\alpha \to 0^+$. This is broadly indicative of Hoeffding- or Bernstein-type concentration bounds. By contrast, **Control Variate** is slightly more general (not requiring boundedness), but loses this scaling rate as a consequence. This explains the significantly increased sensitivity to α (specifically, $\mathcal{O}(\frac{1}{\sqrt{\alpha}})$ scaling) obtained via Chebyshev's inequality.

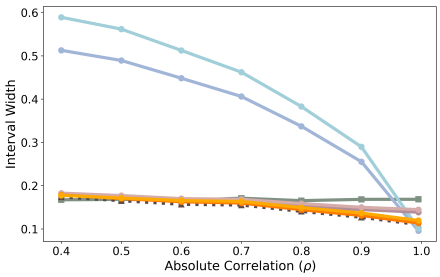
The tradeoffs of this design choice can be seen in several intuitive ways. First, naturally-unbounded metrics (e.g., log likelihoods) are more suited to **Control Variate**. However, from a practical standpoint, guarantees requiring very high confidence (often the best that statistical assurances can achieve for safety-critical applications) will scale much more efficiently with our methods; that is, **Control Variate** will very often yield vacuous intervals for, e.g., $\alpha < 0.0001$, especially when n is constrained. The method can tighten this by instead using a Hoeffding-type bound, but then loses the generality that sets it apart from WSR-based procedures, as it must also enforce a boundedness constraint.



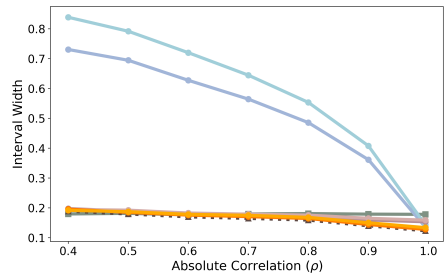
(a) Interval width vs (ρ); $n = 100$, $N = 5000$, $\alpha = 0.1$



(b) Interval width vs (ρ); $n = 100$, $N = 5000$, $\alpha = 0.05$



(c) Interval width vs (ρ); $n = 100$, $N = 5000$, $\alpha = 0.01$



(d) Interval width vs (ρ); $n = 100$, $N = 5000$, $\alpha = 0.005$

Figure 11: Interval widths versus true data correlation between real and paired data. As correlation increases, intervals narrow due to the greater amount of signal present. Results averaged over 100 independent redraws of data. As can be seen, **Control Variate** has greater sensitivity to the true correlation because of the direct correspondence of ρ to the attainable rectifier variance. However, the construction comes at a significant cost in lower-correlation regimes, and again illustrates strong sensitivity to α. Furthermore, as noted in Figure 10, cases of very high correlation often result in miscoverage using the standard control variates implementation with Chebyshev's Inequality [24].

5.7.4 Key Takeaways

A recurring theme of the methodological comparison given in the preceding sections is the relative strength of CV procedures for (a) less stringent significance requirements (higher α), and (b) higher correlation ρ. Intuitively, it is better able to exploit 'easier' settings to tighten intervals more aggressively, at the cost of underperforming (in terms of downstream interval width) in 'harder' ones, where greater confidence is required and the amount of signal in the proxy data is limited.