Reliable and Scalable Robot Policy Evaluation with Imperfect Simulators

Apurva Badithela[1], David Snyder[1,*], Lihan Zha[1,*], Joseph Mikhail[2],
Matthew O'Kelly[3,†], Anushri Dixit[4,†], Anirudha Majumdar[1]

***Abstract*— Rapid progress in imitation learning, foundation models, and large-scale datasets has led to robot manipulation policies that generalize 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 provide confidence intervals on mean 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 our approach saves over $20-25\%$ of hardware evaluation effort to achieve similar bounds on policy performance. Supplementary notes and videos can be found at https://suresim-robot-eval.github.io.**

I. Introduction

Advancing robot learning requires statistically rigorous policy evaluation for reliably assessing how policies generalize to new tasks and environments [1]–[3]. Rapid progress in deep learning was driven by standardized metrics and evaluation benchmarks such as ImageNet [4] in computer vision, and SquaD [5] and GLUE/SuperGLUE [6, 7] 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 [1, 3, 8]. As a result, assessing whether a policy will perform reliably in a new environment distribution remains a core challenge [2].

In robotic manipulation, recent advances in physics-based simulators [9] and action-conditioned video prediction models [10, 11] 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 [12, 13], the simulation-to-real gap precludes rigorous statistical inferences about real-world outcomes from simulation results alone. This paper presents a framework to augment 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 with trustworthy statistical inferences on real performance faces significant challenges due to the simulation-to-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) [14, 15]. Current robot policies, including foundation models and imitation learning-based policies, can be sensitive to such discrepancies. As a result, performance bounds solely relying on large-scale simulation predictions can be biased.

We tackle the aforementioned challenges in order to provide confidence intervals on mean performance of robot manipulation policies. Our key idea is to connect the problem of combining simulated and real-world evaluations to that of prediction powered inference (PPI) [16]. PPI is a paradigm for valid statistical inference that can leverage a large set of learned model predictions together with a comparatively small number of gold-standard data. In our setting, gold-standard labels are real-world evaluations of a policy, while the predictions are obtained from simulation evaluations. For a bounded performance metric, the resulting confidence intervals on mean policy performance are valid with the finite samples of real and simulated data. When simulation is sufficiently predictive, PPI yields tighter non-asymptotic confidence bounds than using real-world trials alone, allowing us to scale with simulation rather than costly hardware evaluations. Our experiments show that it is possible to save $20-25\%$ of hardware trials using state-of-the-art physics-based simulators [9].

Statement of Contributions. First, we present a rigorous policy evaluation framework for finite-sample valid

*Equal contribution, †Equal advising.

A. Majumdar is partially supported by NSF CAREER Award 2044149 and the Sloan Fellowship. A. Badithela is supported by the Presidential Postdoctoral Fellowship.

[1]Department of Mechanical and Aerospace Engineering, Princeton University. {ab5832, dasnyder, lihanzha,ani.majumdar}@princeton.edu

[2]Department of Aerospace Engineering, University of Texas, Austin. josephmikhail@utexas.edu

[3]Waymo. mokelly@waymo.com

[4]Department of Mechanical Engineering, University of California, Los Angeles. anushridixit@ucla.edu

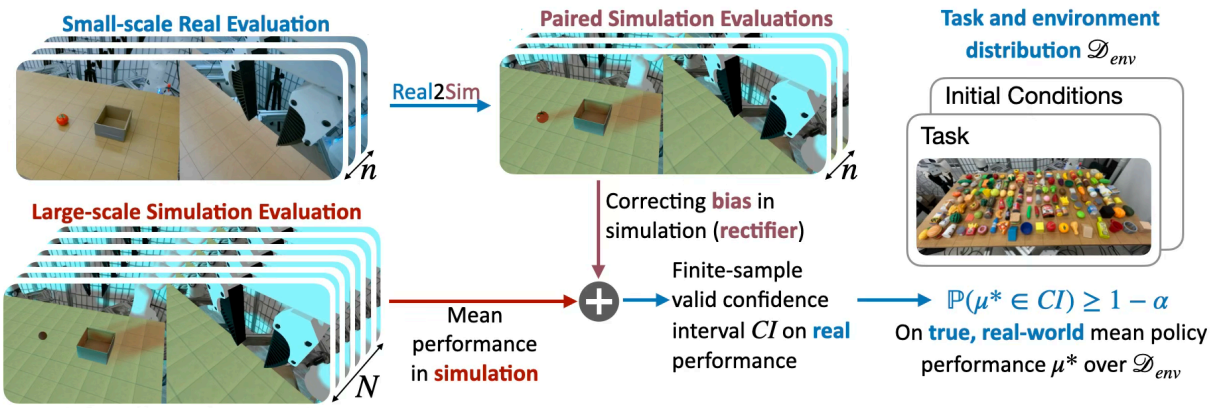

Fig. 1: Our goal is to evaluate a policy by computing bounds on its mean real-world performance on a diverse environment distribution $\mathcal{D}_{\text{env}}$. We present a framework that augments real-world evaluations with simulation evaluations to provide stronger inferences on real-world policy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

inferences on real-world performance by combining large-scale simulation trials with a relatively small number of real trials. A key element 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 to leverage prediction powered inference, and we identify best practices for integrating simulation with real-world evaluation to obtain tighter confidence intervals. Second, we demonstrate our evaluation paradigm on a single-task diffusion policy [17] trained from scratch as well as the robot foundation model π_0 [18] finetuned on multiple tasks. Finally, we discuss the sensitivity of our method to different types of real-simulation gap, including an example of when the gap is too large for simulation to provide benefits over real-only trials.

II. Related Work

Real-world Policy Evaluation. Real-world evaluation is expensive because it offers limited parallelism while often requiring manual logging of outcomes and careful resetting of environments. Yet, it remains the gold-standard for assessing policy performance, driving significant efforts to establish standardized robotic benchmarks with carefully defined tasks, environments, and robot setups for reproducible policy evaluation [19]–[23]. A comprehensive list of best practices for rigorously evaluating robot policies is detailed in [1]. To address these challenges, the community is building cloud-based evaluation platforms [14, 24]–[27] and distributed evaluator networks for unbiased, pairwise policy comparisons [28]. However, rapidly evaluating policies by scaling hardware evaluations with sufficient coverage for statistical assurances remains challenging.

Policy Evaluation in Simulation. Physics-based simulation benchmarks [9, 29]–[35] offer a reproducible and cheaper alternative to real-world robot policy evaluation, though they can be time-consuming to set up. Action-conditioned video world models [10, 36, 37] promise faster scene initialization via text, image, or video prompts [38]. While their use in policy evaluation is nascent, it is attracting growing interest due to advantages over physics-based simulation [11, 13]. These models, however, remain susceptible to hallucinations, and accurately capturing real-world dynamics is still a major challenge. Yet, simulation remains a valuable proxy. For example, simulation-based rankings—whether across different policies or for a given policy under diverse environmental factors—have been shown to correlate well with real-world performance [12, 13, 19, 34, 39]. Predictive red-teaming algorithms [40] offer an alternative to simulation by predicting whether a policy will succeed in a new environment without policy rollouts, showing strong correlation between real and predicted performance rankings across various environmental factors. In contrast to these methods, our approach leverages simulation, even when imperfect, to provide assurances on *real* policy performance over a distribution of tasks and environments.

Statistically Confident Policy Evaluation. A recent study presents statistically rigorous protocols for comparing the capabilities of generalist robot policies in real-world and simulation tests [3] Sequential policy comparison frameworks further reduce real-world evaluation cost while maintaining statistical validity under anytime stopping [8]. Beyond comparing policies, it is also important to assess the individual policy performance. For binary success criteria, Vincent *et al.* [41] provide optimal confidence intervals from real evaluations. For non-binary metrics, confidence intervals may be obtained from real-world evaluation samples via concentration inequalities (e.g., Hoeffding [42]), though this would require a large real evaluation budget. Instead, we scale simulation while requiring a small number of real evaluations to ensure reliability.

Finally, concurrent to our work are efforts that apply statistical inference techniques to off-policy evaluation [43], and the use of control variates to combine simulation evaluation with real-world logged data, demonstrating applications in self-driving and robot navigation [44]. While conceptually related, our work differs in two key respects. First, we present finite-sample valid 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.

III. 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 a success/failure metric or a continuous metric for 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:

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

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 $S_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$. 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, 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:

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

where the probability measure is defined over the draw of the empirical evaluation sample S_n. Any method that satisfies Equation (1) 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 (1). 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 [42] or Bernstein inequalities, or more recent state-of-the-art betting-based methods [45]. 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?*

IV. Approach: Simulation to Augment Real Tests via Prediction Powered Inference

Suppose we have access to a simulator for policy evaluation. Let $\mathcal{X}_{\text{sim}}$ denote the set of simulation environments. We can define a real2sim function $g : \mathcal{X} \to \mathcal{X}_{\text{sim}}$ that translates a real environment setup into simulation. For each real environment $X \in \mathcal{X}$, the corresponding simulation environment is defined 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. 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.

Correcting for bias in large-scale simulation predictions and deriving valid confidence intervals for μ^* requires more than simply combining real and simulated evaluations through imputation. To tackle this challenge, we identify prediction powered inference (PPI) [16] as a suitable mathematical framework for our problem. Prediction powered inference enables 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 [16]. 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.

SureSim. 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 [16], we cannot directly evaluate on X in simulation. Composing the real2sim function with the simulator yields simulation predictions for real environments: $f(\tilde{X}) = f(g(X))$. This formulation enables us to apply prediction-powered inference to rigorously combine real-world trials with large-scale simulation for reliable estimates of 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 $X_1, \ldots X_{n+N} \overset{\text{iid}}{\sim} \mathcal{D}_{\text{env}}$, we apply the real2sim function to get simulation environments $\tilde{X}_1, \ldots, \tilde{X}_{n+N}$, where $\tilde{X}_i = g(X_i)$. The corresponding outcomes of evaluating the policy in simulation are denoted as $f(\tilde{X}_1), \ldots, f(\tilde{X}_{n+N})$. 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 ith data sample is defined as:

$$\Delta_i = \frac{n+N}{n}\big(Y_i - f(\tilde{X}_i)\big)\xi_i + f(\tilde{X}_i),\ 1 \le i \le n+N, \quad (2)$$

where ξ_i is an indicator of whether $(Y_i, f(\tilde{X}_i)) \in D_{\text{paired}}$. The sample mean of Equation (2) yields the uniform PPI estimator for μ^* [46]:

$$\mu_{\text{PPI}}^{\text{unif}} = \underbrace{\frac{1}{n}\sum_{i=N}^{n+N}(Y_i - f(\tilde{X}_i))}_{\text{Rectifier}} + \underbrace{\frac{1}{n+N}\sum_{i=1}^{n+N} f(\tilde{X}_i)}_{\text{Simulation evaluations}}. \quad (3)$$

The first term is referred to as the rectifier, since it corrects for the evaluation bias in large-scale simulation predictions. Intuitively, the rectifier encodes the real2sim gap in policy evaluations: a lower rectifier variance relative to real evaluation scores indicates that the simulator is sufficiently predictive of real outcomes.

For some significance level α, a confidence interval for μ^* is computed from the sample evaluation data (Equation (2)) using non-asymptotic methods for mean estimation via the Waudby-Smith and Ramdas (WSR) algorithm [45], which just requires the bounds of the random variable to be specified a priori. This method is denoted as **SureSim** (**S**calable and **R**eliable Policy **E**valuation with **Sim**ulation). We also present a hedged variant termed **SureSim-UB** which returns 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: one for paired data and the other for proxy predictions [16]. The resulting estimator for μ^* is:

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

where μ_{PPI} is also an unbiased estimate of μ^*. The method is two-stage because it splits the confidence interval construction: one interval for the rectifier at level $\delta < \alpha$, another for the simulation data at level $\alpha - \delta$, and combines them with a Minkowski sum via union bound [16][1]. If real and simulation scores lie in the range $[0, 1]$, the rectifier is bounded between $[-1, 1]$. The bloated rectifier bounds coupled with the small number of paired samples for the rectifier confidence interval leads to wider confidence intervals for μ^* under the two-stage approach. Similar to **SureSim-UB**, we also introduce a hedged version of this method termed **SureSim-UB (2-Stage)**.

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

[1]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.

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 [16, 46].

Baseline. Termed as **Classical**, our primary baseline computes finite-sample confidence intervals — without augmenting simulation — by applying the non-asymptotic WSR procedure [16, 45] directly on real evaluations. These intervals represent the standard procedure for obtaining confidence intervals on the mean, and serve as an ablation with respect to the incorporation of proxy data.

Related Methods. While we primarily compare our methods to **Classical** since it provides a finite-sample guarantee, we also implement and discuss the control variates procedure (denoted **Control Variate**) [44] in the sim2sim setting. We do not consider it 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 [44] 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. These paired samples yield a variance estimate for the control variate estimator, which is subsequently used in Chebyshev's inequality to derive a confidence interval for the mean [44]. 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.

V. Robot Experiments

We evaluate policy generalization on the pick-and-place family of tasks for diverse 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? 3) How well does our method perform under high and low real-simulation correlation?

Experimental Setup. We evaluate policies on a Franka Panda robot with a wrist-mounted RealSense D405 and a Logitech C920 third-person camera. This setup is replicated in simulation using ManiSkill3 [9] by constructing a customized Franka Panda robot in which the default gripper is replaced with a 3D model of the real gripper. 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 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.

Real2Sim Pipeline. We collected 120 real objects, primarily toy kitchen items (Figure 2), to evaluate object generalization. For paired simulations, each object's texture and 3D mesh were reconstructed from a single image using Meshy. These 3D models were scaled to match real-world dimensions, and their pose was set according to real experiments. The additional simulation set consists of 2100 objects from the RoboCASA repository [33], excluding those without a semantic or geometric equivalent in the real-world set (e.g., plates). Ideally, we would have access to a large-scale repository of real-world objects with corresponding 3D model pairs—akin to the YCB dataset [47], 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, such large datasets do not at present exist, and therefore, we take these 120 objects are taken to approximate the real-world object distribution that we wish to evaluate our policy on.

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

Policies. We evaluate two policies: i) a single-task diffusion policy [17] trained from scratch and ii) a generalist policy π_0 [18], 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. The open-loop action horizon for each inference step was set to full action chunk size of 30.

Evaluation Metrics. For each real object, we rollout diffusion policy and π_0 at the different initial conditions of the pick object as shown in Figure 2, respectively. 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.

Rectifier and the Real-Simulation Evaluation Gap. Additionally, we share a few insights on mitigating the real-simulation gap in paired evaluations. Crucially, for mean estimation, this gap manifests in the variance of the rectifier, which represents the difference between paired real and simulation outcomes. A high rectifier variance corresponds to low correlation on D_{paired} and a high real-simulation gap. Low correlation in paired evaluations limits the predictive utility of simulation, thereby undermining the advantage of large-scale simulation for trustworthy inference on real performance. To address this, we implement the following: (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 mitigate the real-simulation gap without requiring additional real evaluations.

A. *Real2Sim Robot Experiments*

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 $\alpha = 0.1$.

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 real trials at 5 initial conditions shown in Figure 2. The paired evaluation score $f(\tilde{X}_i)$ is the average of simulation partial scores from 100 evaluations corresponding to the 5 real 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[2] of $n = 60$ paired evaluations and up to $N = 700$ additional evaluations.

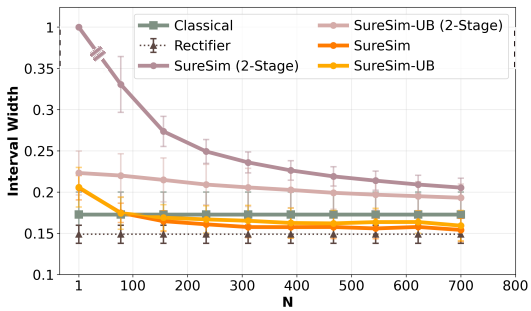

Fig. 3: Evaluating Diffusion Policy with $n = 60$ paired trials and up to 700 additional simulations.

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

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

interval width serves as a lower bound on the confidence interval width as the number of additional simulations increase. 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.

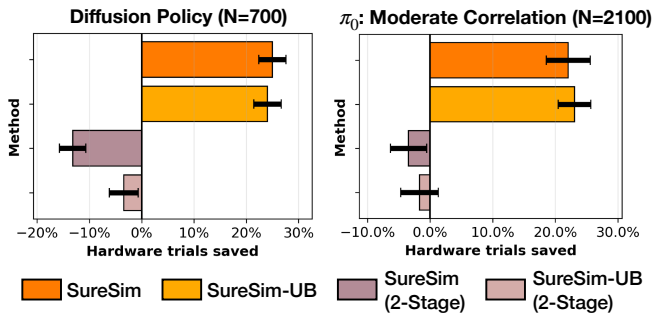

Fig. 4: Average number of hardware trials saved compared to **Classical**, computed over 100 random draws of data. Error bars indicate standard error of the mean savings.

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 increase 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** and **SureSim-UB** yield around 25% savings with respect to real-only evaluation for diffusion policy.

Finetuned π_0. We present two examples of evaluating π_0, where we consider a joint distribution 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. In the second case, the initial condition is sampled from $\{1, 2, 3\}$. The former setting yields a moderate real-to-sim correlation, whereas the latter produces a low correlation. We present both cases to evaluate our methods under contrasting real-simulation correlation regimes. 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. Figure 5 illustrates average confidence interval widths with increasing number of additional simulations. Confidence intervals will vary in width and location for different draws of data from the same distribution; we illustrate one representative confidence interval for each method in Figure 5.

Moderate Correlation. The real-simulation Pearson correlation ρ is 0.59 on average on the paired evaluation set. Despite the modest correlation, at just $N = 1000$ additional simulations, **SureSim** and **SureSim-UB** approach the rectifier lower bound, resulting in an advantage over **Classical**. At $N = 2100$ additional simulations, **SureSim (2-Stage)** and **SureSim-UB (2-Stage)** approach the average **Classical** interval width, and are projected to decrease further with additional scaling. Compared to the 2-stage methods, **SureSim** and **SureSim-UB** efficiently efficiently tighten confidence intervals up to the limit afforded by the real-simulation gap. Furthermore, as seen in Figure 4, **SureSim** and **SureSim-UB** require 20% fewer real trials compared to hardware only evaluation. In future work, we will study further improvements to these finite-sample results by fine-tuning simulation.

Low Correlation. Empirically, finetuned π_0 demonstrates strong generalization to diverse object types, with initial conditions $\{1, 2, 3\}$ consistently achieving high success rates compared to initial condition 4. While simulation evaluations reflect this trend, they do not reflect subtle variations in real performance on the subset $\{1, 2, 3\}$, resulting in a low correlation of around -0.05 on D_{paired}. This results in qualitatively different behavior. As shown in Figure 5, none of our methods beat **Classical**. This is unsurprising because the there is nothing to infer about real policy performance from simulation. While scaling simulation does not help in this instance, our methods still return statistically valid confidence intervals, ensuring that interval width is not spuriously reduced when simulation is not a good proxy.

Across all Real2Sim experiments with moderate correlation, 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 discussed in asymptotic settings for mean estimation [16], 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.

B. Sim2Sim Experiments

We run simulation-simulation experiments to allow for a large holdout set to compute the "true" mean for validating coverage. As illustrated in Figure 6, we use one simulator setting as the "real" environment and the other as "simulation". We evaluate finetuned π_0 on 3D object models of real objects (Figure 2) as well RoboCASA. The paired evaluation set

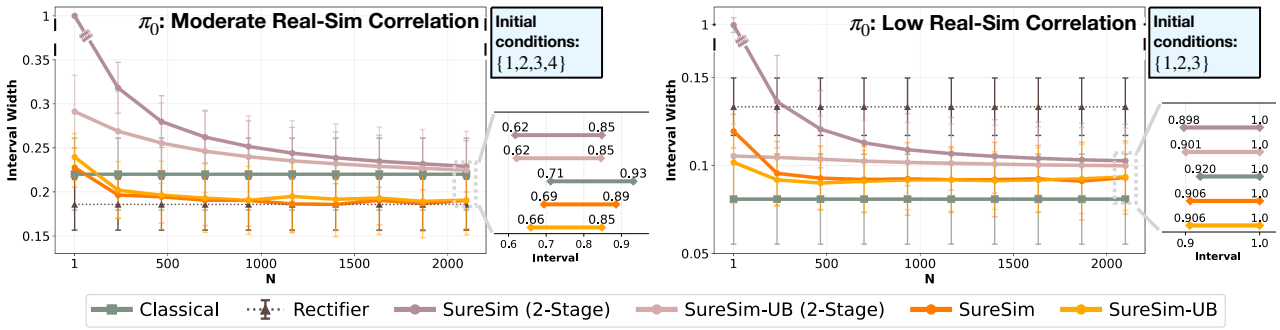

Fig. 5: **How does interval width decrease with scaling simulations?** Confidence interval widths computed for $n = 60$ paired trials and up to $N = 2100$ additional simulations. *Left*: Under moderate correlation, **SureSim** and **SureSim-UB** achieve a 20% decrease in average interval width at $N = 2100$ compared to **Classical**. *Right*: As expected, there is no decrease in interval width with scaling simulations under low correlation. Due to truncation at 1, the interval widths are smaller than the rectifier interval width (which do not truncate here).

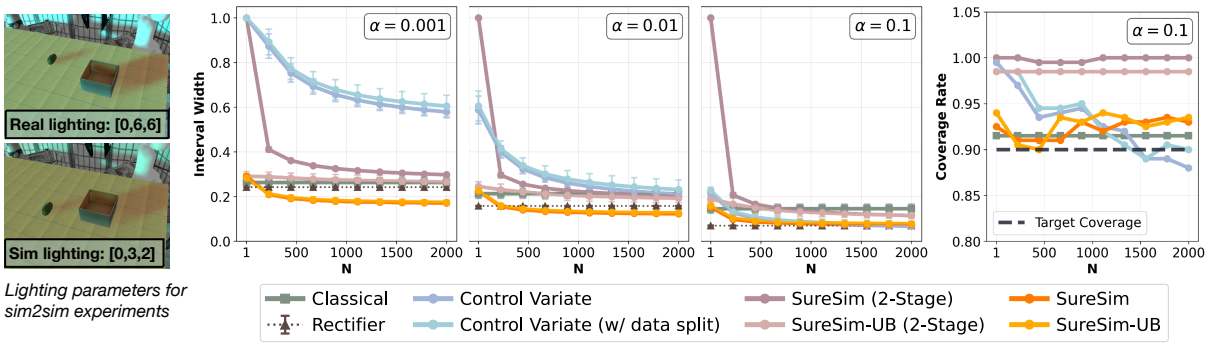

Fig. 6: **Sim2Sim Experiments.** Confidence interval widths for 100 random draws of data at $n = 100$ paired trials with up to $N = 2000$ additional simulations. In each draw, we hold out 400 randomly sampled environments for validating coverage. Our methods always beat **Classical** irrespective of confidence level α.

has a very high correlation of 0.97. For these experiments, we also report intervals for the **Control Variate** method, including the standard implementation presented in [44] 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 6, at $\alpha = 0.1$, all methods beat **Classical**, with the **SureSim** family efficiently converging to the rectifier interval width. As the significance level decreases, **Control Variate** no longer beats **Classical** while **SureSim** always improves. Based on the prior discussion, we do not expect **Control Variate** to meet the required coverage rate, and as seen in Figure 6, 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 $\alpha = 0.1$, which highlights the importance of controlling for Type-1 error.

VI. Conclusions

We introduce **SureSim** for augmenting large-scale simulation with a relatively smaller number of real-world evaluations to provide non-asymptotically valid inferences on real-world policy performance. With a real2sim formalism, we can characterize the problem of combining real and simulation evaluations as a prediction-powered inference problem, and leverage finite-sample valid mean estimation algorithms. This pipeline allows us to evaluate the generalization capabilities of robot foundation models such as diffusion policy and π_0. Compared to hardware-only evaluation, our method saves over $20-25\%$ of hardware evaluations on average. While this work is a foundational step toward scalable and reliable robot policy evaluation, future research directions include data-efficient fine-tuning to improve real-simulation correlation, replacing physics-based simulation with action-conditioned world models, and active sampling algorithms that prioritize real evaluations with large real-simulation gap.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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*. IEEE, 2009, pp. 248–255.

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

[6] 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.

[7] 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*, vol. 32, 2019.

[8] 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.

[9] 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.

[10] 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 video world models," *arXiv preprint arXiv:2505.12705v2*, 2025.

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

[12] 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:2405.05941*, 2024.

[13] X. W. M. Team, "1x world model: Evaluating bits, not atoms," 1X, Tech. Rep., 2025.

[14] 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.

[15] 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.

[16] A. N. Angelopoulos, S. Bates, C. Fannjiang, M. I. Jordan, and T. Zrnic, "Prediction-powered inference," *Science*, vol. 382, no. 6671, pp. 669–674, 2023.

[17] 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*, 2023.

[18] 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," in *Robotics: Science and Systems*, 2025.

[19] M. Heo, Y. Lee, D. Lee, and J. J. Lim, "Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation," in *Robotics: Science and Systems*, 2023.

[20] 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*, vol. 44, no. 4, pp. 592–606, 2025.

[21] 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)*. IEEE, 2019, pp. 8691–8697.

[22] 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)*. IEEE, 2024, pp. 8258–8264.

[23] 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*, vol. 9, no. 1, pp. 9–16, 2023.

[24] 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)*. IEEE, 2017, pp. 1699–1706.

[25] 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.

[26] 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*, vol. 7, no. 1, pp. 486–493, 2021.

[27] 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*. PMLR, 2022, pp. 190–204.

[28] 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.

[29] 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*. IEEE, 2012, pp. 5026–5033.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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*, vol. 7, no. 3, pp. 7327–7334, 2022.

[36] M. Yang, Y. Du, K. Ghasemipour, J. Tompson, D. Schuurmans, and P. Abbeel, "Learning interactive real-world simulators," *arXiv preprint arXiv:2310.06114*, vol. 1, no. 2, p. 6, 2023.

[37] 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*, vol. 1, no. 8, p. 1, 2024.

[38] 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.

[39] 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*, vol. 5, no. 4, pp. 6670–6677, 2020.

[40] 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.

[41] 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.

[42] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American statistical association*, vol. 58, no. 301, pp. 13–30, 1963.

[43] A. Mandyam, J. Meng, G. Gao, J. Sun, M. Schwager, B. E. Engelhardt, and E. Brunskill, "Perry: Policy evaluation with confidence intervals using auxiliary data," *arXiv preprint arXiv:2507.20068*, 2025.

[44] 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.

[45] I. Waudby-Smith and A. Ramdas, "Estimating means of bounded random variables by betting," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 86, no. 1, pp. 1–27, 2024.

[46] T. Zrnic and E. Candes, "Active statistical inference," in *International Conference on Machine Learning*. PMLR, 2024, pp. 62 993–63 010.

[47] 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*, vol. 36, no. 3, pp. 261–268, 2017.