Sim2Val: Leveraging Correlation Across Test Platforms for Variance-Reduced Metric Estimation

Rachel Luo[1], **Heng Yang**[1,2], **Michael Watson**[1], **Apoorva Sharma**[1], **Sushant Veer**[1], **Edward Schmerling**[1], **Marco Pavone**[1,3]
[1]NVIDIA [2]Harvard University [3]Stanford University
{raluo, hengy, mwatson, apoorvas, sveer, eschmerling, mpavone}@nvidia.com

Abstract: Learning-based robotic systems demand rigorous validation to assure reliable performance, but extensive real-world testing is often prohibitively expensive, and if conducted may still yield insufficient data for high-confidence guarantees. In this work, we introduce a general estimation framework that leverages *paired* data across test platforms, e.g., paired simulation and real-world observations, to achieve better estimates of real-world metrics via the method of control variates. By incorporating cheap and abundant auxiliary measurements (for example, simulator outputs) as control variates for costly real-world samples, our method provably reduces the variance of Monte Carlo estimates and thus requires significantly fewer real-world samples to attain a specified confidence bound on the mean performance. We provide theoretical analysis characterizing the variance and sample-efficiency improvement, and demonstrate empirically in autonomous driving and quadruped robotics settings that our approach achieves high-probability bounds with markedly improved sample efficiency. Our technique can lower the real-world testing burden for validating the performance of the stack, thereby enabling more efficient and cost-effective experimental evaluation of robotic systems.

Keywords: metric estimation, sample efficiency, control variates

1 Introduction

Rigorous evaluation of autonomous systems is essential for both research and real-world deployment. However, conducting comprehensive real-world tests is often prohibitively expensive and thus, acquiring a sufficient number of real-world samples to ensure high-confidence estimates of performance metrics (as is necessary for validation) is often infeasible [1]. To mitigate these challenges, system designers frequently rely on alternative test platforms — such as simulators or historical data from previous versions of a robot's policy — to obtain cost-effective, scalable data. However, these auxiliary domains may not accurately reflect performance of the current system in the target deployment environment, and treating outputs from these sources as equivalent to real data can therefore lead to misleading performance estimates. Instead, our goal in this work is to leverage the abundant auxiliary-domain data to more accurately estimate the real-world performance metrics of interest.

We introduce a general framework, Sim2Val, that leverages correlation across heterogeneous test platforms to improve metric estimation in a target domain. For instance, we may wish to estimate a safety-related real-world autonomous driving metric for validation purposes, but want to leverage our extensive simulation data to do so. Our key insight is to exploit *paired* observations — instances for which we collect simultaneous measurements on both an auxiliary platform (e.g. a simulator or offline policy logs) and the target platform (e.g. real-world tests). From these paired samples, we can then learn a *metric correlator function* that maps surrogate metrics to real-world metrics, thereby improving the correlation between the two. By incorporating the auxiliary outputs as a control variate in a Monte Carlo estimator, we provably reduce variance whenever the two platforms exhibit nontrivial correlation, yielding unbiased performance estimates with substantially fewer target-domain (e.g., real-world) samples. Notably, this approach works well for auxiliary data from a simulator or from offline logs of an earlier policy: in either case, a small set of paired examples suffices to "transfer" cheap measurements into tighter confidence bounds on the metric of interest. We empirically demonstrate our metric estimation framework in autonomous driving and quadruped robotics settings, showing

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

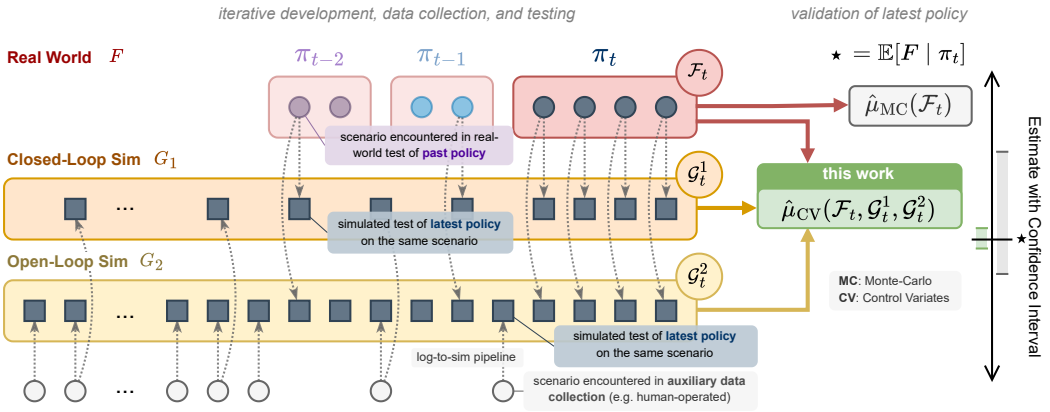


Figure 1: When validating an autonomous system, we can measure performance across multiple test platforms, but fundamentally seek to measure performance in the real-world. While real-world measurements (circles) are expensive, log-to-sim pipelines enable measuring simulated metrics (squares) for the latest policy on scenarios encountered in past tests and data-collection campaigns. In this work, we propose a control variates approach which leverages these additional simulated tests to estimate the expected real-world performance. This method achieves tighter confidence intervals than using real-data alone so long as the simulated metrics are predictive of real-world performance.

that our approach achieves high-probability bounds with markedly improved sample efficiency.

Thus, our contributions are four-fold: (1) a method for leveraging *paired* data to reduce the variance of Monte Carlo estimates of real-world performance metrics; (2) the adaptation of classical control-variates techniques for robotics applications, enhanced by a metric correlator function to improve performance when the baseline correlation is low; (3) theoretical bounds on variance reduction and sample-complexity when using a correlated auxiliary signal; and (4) experimental validation of our framework in autonomous driving and quadruped robotics settings.

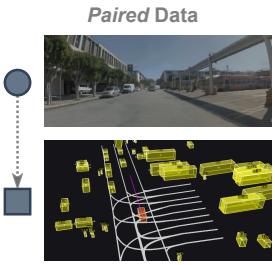


Figure 2: Paired real world and simulation data.

By leveraging correlation across different test platforms, our approach enables more scalable and stronger validation of robotic systems.

2 Background and Related Work

This paper considers the problem of validating an autonomous system in a data-efficient way. In particular, our focus is on estimating the expected value of a metric of interest on the distribution of environmental conditions expected at deployment time, potentially conditioned on a particular subdomain of interest.[1] Testing autonomous systems in the real world can be expensive; real-world mistakes can be costly, requiring tests to be carefully supervised. Given the costs of real-world testing, one key application of our work is how to use surrogate metrics obtained through log-based simulation to improve our real-world sample efficiency in estimating performance.

Log-Replay Simulation Physics-based simulators have provided an accurate test-platform for some robotics tasks, but closing the sim-to-real gap with purely synthetic simulation is challenging, particularly for open-world robotics applications with high-dimensional input spaces. To address this, an increasingly common paradigm is the concept of "log-replay simulation," which decouples real-world data collection (i.e. obtaining logs) from system evaluation (replaying those logs). Log-replay simulation can be as simple as open-loop evaluation of a policy on scenarios sampled in human-operated data collection [2, 3, 4], or as complex as closed-loop simulation with reactive agents [5, 3] and re-simulated sensor data [6, 7] seeded from sensor logs of real-world scenarios. Notably, log-replay based simulation enables *paired* measurements between real-world and simulated domains. In this work, we leverage the availability of such paired data to incorporate surrogate-only measurements at scale into statistical estimators of real-world metrics.

[1] We stress that this form of statistical validation is only one element of full system validation; measuring the tail metrics and targeting known failure modes are also critical, but not the focus of this work.

Offline (Batch) Reinforcement Learning & Off-Policy Evaluation Motivated by the costs of on-policy data collection, the field of *offline* (or *batch*) reinforcement learning aims to study policies with limited ability to test in closed-loop [8]. Most relevant to this work is the topic of *off-policy evaluation*, where the goal is to estimate the performance of the current policy π_t using data collected under a different policy. Methods for off-policy evaluation typically leverage importance sampling to re-weight samples in the dataset to account for the difference between π_t and the data-collection policy [9], or rely on using observed data fitting a model of the environment to obtain surrogate measurements of the performance of π_t. Of particular relevance to our work is the concept of the *doubly robust estimator* [10, 11], which proposes augmenting the importance sampling approach by leveraging an approximate Q function as a control variate to reduce variance. In this work, we consider the system under test and the behavior policy to be a black box, so we do not use action probabilities for importance sampling, but we use surrogate metrics from a test platform to serve the role of a control variate.

Monte-Carlo Methods for Variance Reduction Advanced Monte-Carlo methods, such as importance sampling, stratified sampling, and control variates, provide practitioners with tools to leverage domain knowledge to obtain tighter confidence intervals than naïve Monte-Carlo [12]. In this work, we use the technique of control variates, which reduces variance by leveraging an auxiliary signal that is correlated with the metric of interest, but whose expectation is known. Prediction-powered inference [13, 14] applies similar ideas to improve the estimation of statistical quantities by leveraging synthetic labels as control variates, and has been applied to reduce human labeling costs for LLM evaluation [15, 16]. In this work, we adapt these ideas for autonomous system validation where results on surrogate test platforms serve the role of the control variate or synthetic labels, and highlight how learning a mapping from a surrogate platform metric to the target metric can further tighten confidence bounds.

3 Sim2Val Method

3.1 Problem Setup

Let $X \in \mathcal{X}$ denote a random variable with probability distribution P_X over some environmental conditions. Let $F \in \mathbb{R}$ be a random variable corresponding to the value of a metric of interest, with conditional distribution P_X. For example, X may represent the environmental conditions for a driving scenario (road geometry, location of road agents, road agent intents), and F could represent the minimum time-to-collision experienced by an AV planner when rolled out in the scenario. Our goal is to estimate the expected value of F over the joint distribution of (X, F),

$$\mu \equiv \mathop{\mathbb{E}}_{X \sim P_X} \left[\mathop{\mathbb{E}}_{F \sim P_{F|X}} [F \mid X] \right]. \tag{1}$$

We assume that we have a measurement of F on n samples from P_X, $\{(X_i, F_i)\}_{i=1}^n$. A naïve Monte Carlo estimator of μ corresponds to the empirical mean,

$$\hat{\mu}_{\mathrm{MC}} := \frac{1}{n} \sum_{i=1}^{n} F_i, \qquad \mathrm{Var}[\hat{\mu}_{\mathrm{MC}}] = \frac{1}{n} \mathrm{Var}[F] \approx \frac{1}{n^2} \sum_{i=1}^{N} (F_i - \hat{\mu}_{\mathrm{MC}})^2. \tag{2}$$

Cheaper surrogate test platforms, e.g. log-replay simulation, allow us to measure surrogate metrics $\boldsymbol{G} \in \mathbb{R}^d$ conditioned on a scenario X. These surrogate measurements may be stochastic themselves, so we model them as a random vector with conditional distribution $P_{\boldsymbol{G}|X}$. Importantly, we assume that we can sample $\boldsymbol{G}$ both for the n scenarios for which we have a measurement of F, as well as for additional logged scenarios, e.g., those sampled during tests of other systems or in human-operated data collection. Ultimately, we assume that we have n paired samples $\mathcal{D}_{\mathrm{paired}} = \{(X_i, F_i, \boldsymbol{G}_i)\}_{i=1}^n$, and k surrogate-only samples, $\mathcal{D}_{\mathrm{surrogate}} = \{(X'_j, \boldsymbol{G}'_j)\}_{j=1}^k$, where

$$\begin{aligned} X_i &\sim P_X \quad \forall\, i = 1, \ldots, n & X'_j &\sim P_X \quad \forall\, j = 1, \ldots, k \\ F_i &\sim P_{F|X}(\cdot \mid X_i) \quad \forall\, i = 1, \ldots, n \\ \boldsymbol{G}_i &\sim P_{\boldsymbol{G}|X}(\cdot \mid X_i) \quad \forall\, i = 1, \ldots, n & \boldsymbol{G}'_j &\sim P_{\boldsymbol{G}|X}(\cdot \mid X'_j) \quad \forall\, j = 1, \ldots, k \end{aligned}$$

The nature of $(X_i, F_i, \boldsymbol{G}_i)$ depends on the experimental setting. For example: (i) X_i may be a specific log, with F_i denoting the real-world measurement and $\boldsymbol{G}_i$ the corresponding simulation replay measurement; (ii) X_i may again be a specific log with F_i denoting the real-world measurement, but

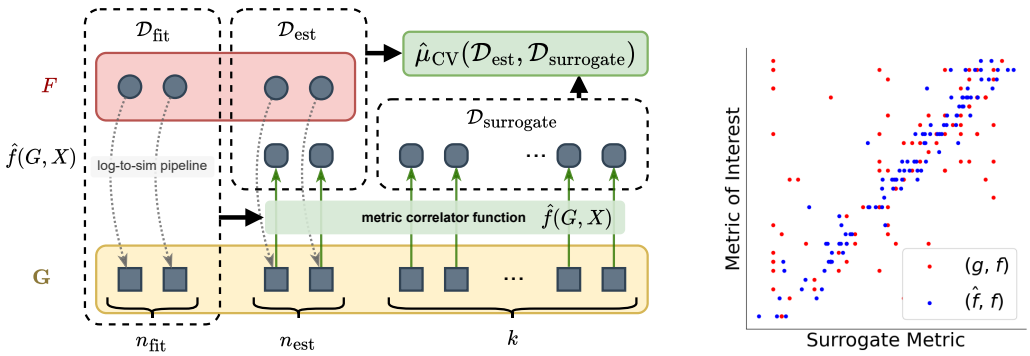


Figure 3: (Left) Given some paired measurements of our real (F) and surrogate ($\boldsymbol{G}$) metrics, our approach uses some of that data $\mathcal{D}_{\text{fit}}$ to learn a *metric correlator function* (MCF, $\hat{f}$), mapping the surrogate metric and context features to a prediction of the corresponding real-world measurement. Then, we can apply the MCF to the remaining surrogate metric values to construct the paired $\mathcal{D}_{\text{est}}$ and surrogate only $\mathcal{D}_{\text{surrogate}}$ datasets used produce the control-variates estimator of $\mathbb{E}[F]$. (Right) Empirical results on the NuPlan dataset showing that fitting the MCF can extract a more predictive signal for F from the surrogate metrics $\boldsymbol{G}$, improving correlation.

$\boldsymbol{G}_i$ may be obtained by running simulation on a scenario X_i' retrieved from a simulation database of scenarios similar to X_i; or (iii) X_i may represent a semantic category (e.g., urban night driving), with F_i denoting the real-world performance and $\boldsymbol{G}_i$ the average simulation performance on scenarios from the same category. The key requirement is that F and $\boldsymbol{G}$ exhibit nontrivial correlation.

Our goal is to strategically leverage the surrogate metric observations to achieve a tighter estimate for μ. Critically, we make no a priori assumptions on the relationship between $P_{F|X}$ and $P_{\boldsymbol{G}|X}$; we treat the surrogate test platforms as a black box. Nevertheless, as we will demonstrate, a surrogate platform where $\boldsymbol{G}$ is highly correlated with F can enable more efficient, lower variance statistical estimators for μ.

3.2 Control Variate Framework with Surrogate Data

We adapt the method of control variates to improve our estimation of μ using the inexpensive surrogate $\boldsymbol{G}$ as a control variate. Typically, the method of control variates requires perfect knowledge of $\theta := \mathbb{E}[\mathbb{E}[\boldsymbol{G} \mid X]]$, i.e., the expected value of our surrogate metric; however, in our application the *exact* value of θ is not accessible. Nevertheless, given that the number of surrogate samples (k) is much higher than the number of true metric samples we can collect (n), applying this strategy with an estimated value of θ can produce a tighter variance. Specifically, we define the estimator

$$\hat{\mu}_{CV}(\mathcal{D}_{\text{paired}}, \mathcal{D}_{\text{surrogate}}; \boldsymbol{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\left(F_i - \boldsymbol{\beta}^T \boldsymbol{G}_i\right) + \frac{1}{k}\sum_{j=1}^{k}\boldsymbol{\beta}^T \boldsymbol{G}_j', \tag{3}$$

where $\boldsymbol{\beta} \in \mathbb{R}^d$ is a hyperparameter. This estimator has lower variance than the simple Monte-Carlo estimator when $\boldsymbol{G}$ is correlated with F, and the degree of variance reduction depends on the degree of correlation and the number of additional surrogate only samples k.

Theorem 1 (Optimal Variance of $\hat{\mu}_{\text{CV}}$)**.** *Let F and $\boldsymbol{G}$ be as defined in Section 3.1. Let $\hat{\mu}_{\text{CV}}$ be as defined in (3). Then, if $\boldsymbol{\beta}$ is chosen as $\boldsymbol{\beta}_{\text{opt}}$, defined as:*

$$\boldsymbol{\beta}_{\text{opt}} := \left(\frac{k}{k+n}\right)\text{Var}(\boldsymbol{G})^{-1}\text{Cov}(\boldsymbol{G}, F), \tag{4}$$

the variance of $\hat{\mu}_{\text{CV}}$ takes the following form:

$$\text{Var}(\hat{\mu}_{\text{CV}}(\boldsymbol{\beta}_{\text{opt}})) = \frac{1}{n}\left(1 - \frac{k}{k+n}\rho^2(\boldsymbol{G}, F)\right)\text{Var}(F), \tag{5}$$

where $\rho^2(\boldsymbol{G}, F) := \|\text{Var}(\boldsymbol{G})^{-\frac{1}{2}}\text{Cov}(\boldsymbol{G}, F)\text{Var}(F)^{-\frac{1}{2}}\|_{\text{Fr}}^2$ and captures how correlated the surrogate metric is to the metric of interest[2]*.*

[2]Note that in the case of scalar $\boldsymbol{G}$, ρ corresponds to the Pearson correlation coefficient between F and $\boldsymbol{G}$.

Algorithm 1 Control Variate Estimator

Require: Metric of interest F, surrogate metrics $\boldsymbol{G}$, paired samples $\mathcal{D}_{\text{paired}} = \{(X_i, F_i, \boldsymbol{G}_i\}_{i=1}^n$, surrogate-only samples $\mathcal{D}_{\text{surrogate}} = \{(X'_j, \boldsymbol{G}'_j)\}_{j=1}^k$, desired confidence interval size α

1: Estimate the optimal coefficient $\hat{\boldsymbol{\beta}}_{\text{opt}}$ Eq. (4), using sample estimates of $\text{Var}(\boldsymbol{G})$, $\text{Cov}(\boldsymbol{G}, F)$. [a]

2: Compute the estimate of mean performance $\hat{\mu}_{CV}(\mathcal{D}_{\text{paired}}, \mathcal{D}_{\text{surrogate}}; \hat{\boldsymbol{\beta}}_{\text{opt}})$, Eq. (3).

3: Compute the estimator variance $\widehat{\text{Var}}(\hat{\mu}_{\text{CV}}(\hat{\boldsymbol{\beta}}_{\text{opt}}))$, Eq. (5), using sample estimates of $\text{Var}(F)$, $\text{Var}(\boldsymbol{G})$, $\text{Cov}(\boldsymbol{G}, F)$.

4: Compute the confidence interval $\mathbb{P}\left(\left|\hat{\mu}_{CV}(\mathcal{D}_{\text{paired}}, \mathcal{D}_{\text{surrogate}}; \hat{\boldsymbol{\beta}}_{\text{opt}}) - \mu\right| \geq \alpha\right) \leq \widehat{\text{Var}}(\hat{\mu}_{\text{CV}}(\hat{\boldsymbol{\beta}}_{\text{opt}}))/\alpha^2$, Chebyshev or CLT.

[a] Note that using the same samples to estimate $\hat{\boldsymbol{\beta}}$ and to subsequently compute the mean introduces a very small bias; separate data may be used to estimate $\hat{\boldsymbol{\beta}}$ to eliminate this bias if necessary.

We see that the optimal control variate estimator can do no worse in terms of variance than the simple Monte Carlo estimator, and that as the amount of the surrogate data increases, the variance of the estimator approaches a limit governed by the correlation of $\boldsymbol{G}$ and F, $(1-\rho^2)\text{Var}(F)$. Algorithm 1 summarizes our approach.

A direct consequence of the lower variance from the control variate estimator is a reduction in the amount of real-world data needed to estimate the same confidence interval around μ. We define a confidence interval (CI) $\gamma := (\alpha, p)$ by the tuple that includes a measure of the "size" $\alpha \in [0, \infty)$ of the interval in which we expect the true mean to lie and the probability $p \in [0, 1]$ of the interval *not* covering the true mean. A CI $\gamma_1 := (\alpha_1, p_1)$ is considered to be tighter than $\gamma_2 := (\alpha_2, p_2)$ if $\alpha_1 \leq \alpha_2$ while $p_1 \leq p_2$. With this notion of CI tightness, we present the following theorem.

Theorem 2 (Sample Efficiency for CI Estimation with Chebyschev Inequality)**.** *Let $n_r \in \mathbb{Z}_{>0}$ be the number of real-world samples of F for direct estimation of μ (as in Eq. 2). Let $n_p \in \mathbb{Z}_{>0}$ be the number of real-world and surrogate data pairs $(F, \boldsymbol{G})$ drawn for estimating μ using control variates (as in Eq. 3 and 4). Let k be the number of surrogate-only samples. If we use the Chebyschev inequality to estimate μ, then (i) if $n_r = n_p$, the CI from control variates is tighter than the CI from direct estimation; and (ii) to obtain a CI of equivalent width and confidence, we require n_p to be at least $n_{\min}$ where*

$$n_{\min} := \frac{-(k - n_r) + \sqrt{(k - n_r)^2 + 4n_r k(1 - \rho^2)}}{2} \leq n_r. \tag{6}$$

The proofs of Theorem 1 and Theorem 2 are provided in Appendix A. [3]

3.3 Improving Performance by Learning a Metric Correlator Function

Algorithm 2 Improved Control Variate Estimation with a Metric Correlator Function

Require: Metric of interest F, surrogate metrics $\boldsymbol{G}$, paired samples $\mathcal{D}_{\text{paired}} = \{(X_i, F_i, \boldsymbol{G}_i\}_{i=1}^n$, surrogate-only samples $\mathcal{D}_{\text{surrogate}} = \{(X'_j, \boldsymbol{G}'_j)\}_{j=1}^k$, scenario features $\phi(X)$, desired confidence interval size α

1: Split the n paired samples into $\mathcal{D}_{\text{fit}}$ of size n_{fit} and $\mathcal{D}_{\text{est}}$ of size n_{est}, such that $n = n_{\text{fit}} + n_{\text{est}}$.

2: Train a metric correlator function $\hat{f} : (\boldsymbol{G}, \phi(X)) \longrightarrow F$ using the samples in $\mathcal{D}_{\text{fit}}$.

3: Apply Algorithm 1, with $\boldsymbol{G} := \hat{f}(\boldsymbol{G}, \phi(X))$ and $n := n_{\text{est}}$.

The efficacy of the control variate method described in Algorithm 1 relies on the correlation of the surrogate metrics $\boldsymbol{G}$ to the true metric of interest F conditioned on a scenario X. Although we would hope to ensure that these are strongly correlated when we design our surrogate test platforms (e.g., by designing predictive open-loop metrics [17], or improving the fidelity of our simulator [6]), there may still be a benefit in considering additional features of X. To further improve the correlation, we propose learning a *metric correlator function* (MCF) $\hat{f}$ to map the surrogate measurements $\boldsymbol{G}$ as well as features of the scenario $\phi(X)$ to a prediction of the metric of interest F. As shown in Figure 3, we partition the paired data $\mathcal{D}_{\text{paired}}$ into $\mathcal{D}_{\text{fit}}$ of size n_{fit}, and $\mathcal{D}_{\text{est}}$ of size $n_{\text{est}} := n - n_{\text{fit}}$, and train the MCF on n_{fit} with an MSE loss.

We predict the metric of interest $\hat{F}$ for all remaining data. Then, we can apply the same control variate estimator algorithm as above, replacing $\boldsymbol{G}$ with $\hat{f}(\boldsymbol{G}, \phi(X))$ (and n with n_{est}). In our experiments, we find that training a metric correlator function in this manner can dramatically increase

[3] While Theorem 2 is stated using Chebyshev's inequality, similar conclusions hold for any confidence interval construction whose width scales with the variance. In practice, confidence intervals can be constructed from the control variate samples using any standard approach (such as Chebyshev, Central Limit Theorem, or Hoeffding bounds). CLT-based intervals are typically the tightest, as they exploit asymptotic normality.

correlation. However, this comes at the cost of losing some paired data; the samples that are used in training the metric correlator function cannot be used in the computation of the estimate itself. The MCF-enhanced estimator provides a net variance reduction so long as

$$\rho^2_{\mathrm{MCF}} / \left(1 + \frac{n_{\mathrm{est}}}{k}\right) > \rho^2 / \left(1 + \frac{n}{k}\right). \tag{7}$$

Note, however, that the MCF can be trained on any data where both the true metric and surrogate features are available; the training procedure is not limited to samples from $\mathcal{D}_{\mathrm{paired}}$. For example, data from related but non-target domains may be leveraged for training if the learned correlations generalize well to the target domain. Importantly, leveraging data from outside $\mathcal{D}_{\mathrm{paired}}$ does not consume any of the paired samples from $\mathcal{D}_{\mathrm{paired}}$ that would otherwise be used in the control variate estimation (i.e., it does not affect n_{est}). Algorithm 2 outlines our improved estimation procedure incorporating the learned MCF.

4 Experimental Results

We evaluate our variance-reduction framework on three distinct settings: (1) closed-loop versus open-loop metrics in the nuPlan autonomous driving dataset, (2) autonomous vehicle (AV) simulator outputs for predicting real-world driving performance, and (3) simulator outputs for predicting real-world measurements in a quadruped locomotion task. In each case, we compare the sample variance of our control-variates estimator against the standard Monte Carlo baseline, demonstrating consistent reductions in uncertainty around the estimated mean. Furthermore, by learning an MCF, we observe additional variance reduction, particularly in domains where the raw correlation is low.

4.1 nuPlan Simulation Experiments

We first consider the nuPlan dataset [18], which provides large-scale open-loop and closed-loop simulation metrics (e.g. time to collision (TTC), average displacement error (ADE), drivable area compliance, etc.), for various high-level scenario classes (e.g. "on_intersection," "near_multiple_vehicles"). Here, open-loop simulation employs non-reactive agents following a fixed trajectory (and is thus computationally inexpensive), whereas closed-loop simulation uses fully reactive agents that respond to their environment (and is thus computationally expensive). We demonstrate how our approach can leverage results on the cheap open-loop simulation to improve estimates of performance in closed-loop.

Within each scenario class, we draw a limited set of n paired samples $\{(X_i, \boldsymbol{G}_i, F_i)\}_{i=1}^n$ where $\boldsymbol{G}_i$ and F_i denote the open-loop and closed-loop metrics respectively sampled for X_i. Furthermore, we draw k unpaired open-loop observations $\{(X'_j, \boldsymbol{G}'_j\}_{j=1}^k$. We compare three estimators for the population mean $\mu = \mathbb{E}[f(X)]$: (i) **Monte Carlo baseline (MC):** $\hat{\mu}_{\mathrm{MC}} = \frac{1}{n}\sum_{i=1}^n F_i$. (ii) **Control variates (CV):** $\hat{\mu}_{CV} = \frac{1}{n}\sum_{i=1}^n (F_i - \hat{\boldsymbol{\beta}}^T \boldsymbol{G}_i) + \frac{1}{k}\sum_{j=1}^k \hat{\boldsymbol{\beta}}^T \boldsymbol{G}'_j$, with $\hat{\boldsymbol{\beta}}$ chosen as described in Section 3. (iii) **Control variates with a metric correlator function (CV-MCF):** We replace $\boldsymbol{G}$ by $\hat{f}(\boldsymbol{G}, \phi(X))$, where n_{fit} of the paired samples are used to fit $\hat{f}$, and the remaining $n_{\mathrm{est}} = n - n_{\mathrm{fit}}$ paired samples are used in the control variate estimator.

Variance vs. number of unpaired samples. We first fix the number of paired samples n and sweep the number of open-loop only samples k. Figure 4 plots the sample variance of each estimator for two representative metrics (ADE and TTC $<$ 1s rate) in the "on_intersection" class, with $n = 565$ and k from 0 to 1320. As k increases, the control variate estimators rapidly reduce the variance relative to the Monte Carlo baseline – achieving up to 22% reduction for the direct approach and 35% reduction for the learned approach when $k = 1320$. Similar results for additional metrics and scenario classes are shown in the Appendix.

Impact of the learned metric correlator function. In settings where $\rho(\boldsymbol{G}, F)$ is relatively low (e.g. $\rho(\boldsymbol{G}, F) \approx 0.43$ for the time-to-collision $<$ 1s metric in the "on_intersection" scenario class), directly using $\boldsymbol{G}$ as the control variate yields limited gains. We therefore train a lightweight feed-forward network $\hat{f}$ on a subset of the n paired samples, predicting F from the set of all available open-loop metrics $\boldsymbol{G}$ and additional scene features $\phi(X)$; we augment the target domain samples with data from non-target domains (i.e. other scenario classes). While training the MCF may "consume" paired samples that could otherwise be used directly in the control variate computation, it can be worthwhile if the resulting correlation improvement compensates for the reduced sample size. See Appendix for additional training details.

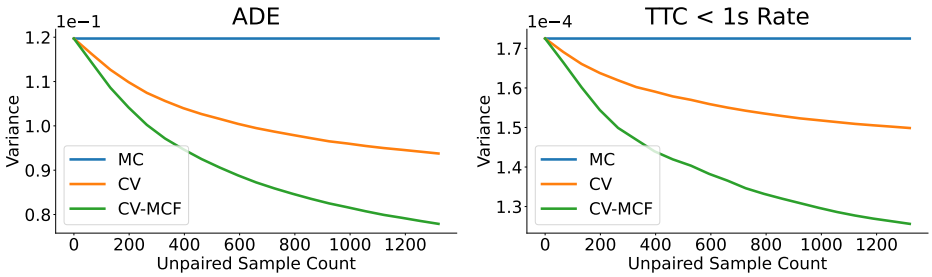


Figure 4: Variance estimates for two representative metrics as a function of the number of unpaired open-loop samples k in the "on_intersection" scenario class. Plots averaged over 10 random trials.

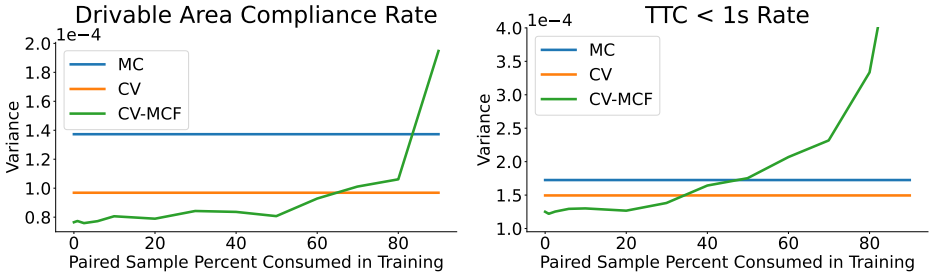


Figure 5: Variance estimates for two representative metrics as a function of the fraction of samples used to train the MCF in the "on_intersection" scenario class. Using data to train an MCF to improve correlation can reduce variance, provided enough data remains for the control variates estimator. Plots averaged over 10 random trials.

Figure 5 plots the variance as a function of the fraction of in-domain paired samples used for training the MCF (with fixed k) for the "on_intersection" scenario class, averaged over 10 trials. Results for additional metrics and scenario classes are shown in the Appendix. We find that the improvement from the metric correlator function is more pronounced in domains with weaker baseline correlation. Notably, even when the model is trained exclusively on out-of-domain data (i.e. zero in-domain samples are used), it still improves the in-domain correlation, indicating strong generalization from related scenarios. Crucially, this comes *without* sacrificing any of the n samples used for MC or CV estimation, yielding consistent variance improvements across subdomains.

Our experiments demonstrate that using open-loop metrics as control variates significantly reduces the variance in estimating closed-loop performance across various metrics and scenario classes. This variance reduction enables stronger statistical guarantees using fewer expensive closed-loop evaluations. As a representative example, by applying Theorem 2 in the "near_long_vehicle" domain for the ADE metric with $n_r = 715$, $k = 1669$, $\rho_{CV} = 0.79$, and $\rho_{CV-MCF} = 0.83$, we find that the same confidence interval can be achieved with just $n_{p,\text{CV}} = 350$ (a 51% reduction) or $n_{p,\text{CV-MCF}} = 303$ (a 58% reduction) paired samples.

4.2 Real-World Autonomous Driving Performance

We next evaluate the effectiveness of our control variate estimator in assessing real-world autonomous driving performance in different types of scenarios (e.g. at road curves or at intersections). To this end, we use paired data comprising outputs from an internal, state-of-the-art neural reconstruction simulator and corresponding real-world driving logs, all collected under the same driving policy. The simulator is capable of reconstructing logged driving scenes (and can render novel camera views when the ego-vehicle deviates from its original trajectory) [7], enabling the construction

Class/Metric	$\hat{\mu}_{\text{MC}}$	$\hat{\mu}_{\text{CV}}$	$\widehat{\text{Var}}(\hat{\mu}_{\text{MC}})$	$\widehat{\text{Var}}(\hat{\mu}_{\text{CV}})$	Var Reduction
intersections/closest dist	10.5636	9.5850	1.0374	0.1776	82.9 %
curvy roads/closest dist	10.7699	11.9377	1.4205	0.4335	69.5 %
intersections/lane centering dist	0.7065	0.6917	0.0016	0.0003	81.7 %
curvy roads/lane centering dist	0.6376	0.6647	0.0038	0.0010	74.3 %

Table 1: Estimates of the mean and the variance of the mean for two different metrics, where ($n = 138$, $k = 781$) for intersection events and ($n = 75$, $k = 423$) for curvature events.

of paired (simulated, real-world) scenes. For each paired episode ($n = 138$ for intersection events and $n = 75$ for curvy road events), we extract two real-world metrics F — nearest distance to another vehicle and average centerline deviation. We also obtain their simulated counterparts $\boldsymbol{G}$, and generate additional simulated samples ($k = 781$ for intersection events and $k = 423$ for curvy road events). Our objective is to leverage the relatively abundant and inexpensive simulator data to improve inference of real-world driving performance. Since the raw correlation $\rho(\boldsymbol{G}, F)$ exceeds 0.9 for both metrics, we apply the control variate estimator directly, without learning a metric correlator function. Table 1 reports the sample variance of each estimator for the two metrics. Relative to the Monte Carlo baseline, the control variate estimator achieves variance reductions of up to 82.9%.

These results confirm that our Sim2Val framework substantially tightens confidence intervals on real-world performance estimates. For example, for intersection events with the closest distance metric, the Monte Carlo estimator requires $n_r = 807$ real-world samples to match the confidence interval of the control variate estimator with $n_p = 138$ real-world samples (where $k = 781$ and $\rho_{cv} = 0.995$), a nearly 6x increase in real-world data requirements.

4.3 Quadruped Velocity Tracking

We then consider the problem of evaluating the velocity tracking performance of a policy trained via reinforcement learning in simulation, and then deployed on a quadruped robot in the real world. Here, the scenario is parameterized by $X \in \mathbb{R}^3$, the three-dimensional command velocity vector, representing desired velocities in the forward, sideways, and yaw directions, respectively. The metric of interest F here is the average velocity tracking error seen over a 2 second deployment of the reinforcement learning policy in the real world, while the performance of the policy in a rollout in the simulator serves as the surrogate metric $\boldsymbol{G}$. We collect $n = 200$ paired samples from real-world tests and paired simulation results, and $k = 400$ additional simulator-only samples.

Method	MC	CV	CV-MCF ($n_{\text{fit}} = 50$)	CV-MCF ($n_{\text{fit}} = 40$)
Mean	4.815×10^{-1}	4.811×10^{-1}	4.748×10^{-1}	4.749×10^{-1}
Variance	2.048×10^{-5}	2.041×10^{-5}	1.931×10^{-5}	1.926×10^{-5}

Table 2: Estimated means and variances using different methods.

We first apply the direct control variates approach (CV). As shown in Table 2, the direct CV method reduces the variance from 2.048×10^{-5} (using only the n real-world evaluations) to 2.041×10^{-5}. The Pearson correlation coefficient ρ between the n paired simulation and real-world metrics is 0.0728, indicating a weak correlation between simulated and real-world performance; consequently, the reduction in variance achieved is limited. By applying Eq. 6 from Theorem 2, we compute $n_{\min} = 200$, indicating that when ρ is close to zero, the reduction in sample complexity is negligible.

Given the weak correlation between the simulated and real-world metrics, we learn a mapping, conditioned on the command velocity X, that predicts the real-world metric from the simulation metric using $n_{\text{fit}} < n$ paired evaluations. We then apply the control variates method with this learned mapping (CV-MCF). As shown in Table 2, when using $n_{\text{fit}} = 50$ and $n_{\text{fit}} = 40$ paired samples, respectively, the CV-MCF method further reduces the variance to 1.931×10^{-5} and 1.926×10^{-5}. When $n_{\text{fit}} = 50$, the Pearson correlation coefficient after learning the metric correlation function increases to 0.6158, in which case $n_{\min} = 145 < n$ from (6), showing improved sample efficiency. Note that even this relatively modest reduction in variance from 2.048×10^{-5} to 1.926×10^{-5}, would have required 38% more real-world tests to achieve without leveraging simulation.

5 Conclusion

In this work, we present Sim2Val, a general, theoretically-grounded framework for improving metric estimation in robotics systems by leveraging *paired* observations across heterogeneous test platforms. By treating auxiliary measurements – such as simulator outputs or offline policy logs – as control variates in a Monte Carlo estimator, our method provably decreases estimator variance whenever the auxiliary and target domains exhibit nontrivial correlation. We provide theoretical analysis on variance reduction and sample efficiency improvements, and empirically validate our approach in autonomous driving and quadruped robotics settings. This framework suggests several interesting avenues for future work, including developing policy-conditional extensions that adapt to evolving control policies, advancing distributional validation to encompass quantile estimation beyond the mean, and optimally allocating repeated simulator calls under a fixed compute budget to further reduce estimator variance.

6 Limitations

Our theoretical analysis relies on the assumption that all test samples (real and surrogate) are drawn i.i.d. from the target distribution. In practice, data collection is often biased towards "interesting" or safety-critical scenarios. Addressing this mismatch may require integrating selection biases into our framework, for example by viewing targeted sampling as a form of importance sampling to further reduce variance. Additionally, by focusing exclusively on mean performance, our approach may not fully capture safety-critical concerns, where uniform reliability across diverse operating conditions is important. Although conditioning on specific operational design domains can reveal deficiencies masked by the population mean, extending our methodology to tail-risk measures (such as conditional value at risk or other quantile-based metrics) would be a useful direction for future work. Finally, we have assumed that only a single auxiliary sample $\boldsymbol{G} \sim P_{\boldsymbol{G}|X}$ is available per scenario X, despite the fact that simulators can generate multiple roll-outs of $\boldsymbol{G}$ from the same X. Optimally allocating repeated simulator calls across scenarios could yield additional variance reduction in the control variate estimator for μ, and would be an interesting avenue for future work.

Acknowledgment

We thank Rajiv Swamy and Alex Tong for their efforts in collecting the quadruped dataset in the Computational Robotics Lab at Harvard University.

References

[1] N. Kalra and S. M. Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016. ISSN 0965-8564. doi:https://doi.org/10.1016/j.tra.2016.09.010. URL https://www.sciencedirect.com/science/article/pii/S0965856416302129.

[2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[3] N. Karnchanachari, D. Geromichalos, K. S. Tan, N. Li, C. Eriksen, S. Yaghoubi, N. Mehdipour, G. Bernasconi, W. K. Fong, Y. Guo, et al. Towards learning-based planning: The nuplan benchmark for real-world autonomous driving. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 629–636. IEEE, 2024.

[4] O. X.-E. Collaboration, A. O'Neill, A. Rehman, A. Gupta, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, A. Tung, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Gupta, A. Wang, A. Kolobov, A. Singh, A. Garg, A. Kembhavi, A. Xie, A. Brohan, A. Raffin, A. Sharma, A. Yavary, A. Jain, A. Balakrishna, A. Wahid, B. Burgess-Limerick, B. Kim, B. Schölkopf, B. Wulfe, B. Ichter, C. Lu, C. Xu, C. Le, C. Finn, C. Wang, C. Xu, C. Chi, C. Huang, C. Chan, C. Agia, C. Pan, C. Fu, C. Devin, D. Xu, D. Morton, D. Driess, D. Chen, D. Pathak, D. Shah, D. Büchler, D. Jayaraman, D. Kalashnikov, D. Sadigh, E. Johns, E. Foster, F. Liu, F. Ceola, F. Xia, F. Zhao, F. V. Frujeri, F. Stulp, G. Zhou, G. S. Sukhatme, G. Salhotra, G. Yan, G. Feng, G. Schiavi, G. Berseth, G. Kahn, G. Yang, G. Wang, H. Su, H.-S. Fang, H. Shi, H. Bao, H. B. Amor, H. I. Christensen, H. Furuta, H. Bharadhwaj, H. Walke, H. Fang, H. Ha, I. Mordatch, I. Radosavovic, I. Leal, J. Liang, J. Abou-Chakra, J. Kim, J. Drake, J. Peters, J. Schneider, J. Hsu, J. Vakil, J. Bohg, J. Bingham, J. Wu, J. Gao, J. Hu, J. Wu, J. Wu, J. Sun, J. Luo, J. Gu, J. Tan, J. Oh, J. Wu, J. Lu, J. Yang, J. Malik, J. Silvério, J. Hejna, J. Booher, J. Tompson, J. Yang, J. Salvador, J. J. Lim, J. Han, K. Wang, K. Rao, K. Pertsch, K. Hausman, K. Go, K. Gopalakrishnan, K. Goldberg, K. Byrne, K. Oslund, K. Kawaharazuka, K. Black, K. Lin, K. Zhang, K. Ehsani, K. Lekkala, K. Ellis, K. Rana, K. Srinivasan, K. Fang, K. P. Singh, K.-H. Zeng, K. Hatch, K. Hsu, L. Itti, L. Y. Chen, L. Pinto, L. Fei-Fei, L. Tan, L. J. Fan, L. Ott, L. Lee, L. Weihs, M. Chen, M. Lepert, M. Memmel, M. Tomizuka, M. Itkina, M. G. Castro, M. Spero, M. Du, M. Ahn, M. C. Yip, M. Zhang, M. Ding, M. Heo, M. K. Srirama, M. Sharma, M. J. Kim, N. Kanazawa, N. Hansen, N. Heess, N. J. Joshi, N. Suenderhauf, N. Liu, N. D. Palo, N. M. M. Shafiullah, O. Mees, O. Kroemer, O. Bastani, P. R. Sanketi, P. T. Miller, P. Yin, P. Wohlhart, P. Xu, P. D. Fagan, P. Mitrano, P. Sermanet, P. Abbeel, P. Sundaresan, Q. Chen, Q. Vuong, R. Rafailov, R. Tian, R. Doshi, R. Martin-Martin, R. Baijal, R. Scalise, R. Hendrix, R. Lin, R. Qian, R. Zhang, R. Mendonca, R. Shah, R. Hoque, R. Julian, S. Bustamante, S. Kirmani, S. Levine, S. Lin,

S. Moore, S. Bahl, S. Dass, S. Sonawani, S. Tulsiani, S. Song, S. Xu, S. Haldar, S. Karamcheti, S. Adebola, S. Guist, S. Nasiriany, S. Schaal, S. Welker, S. Tian, S. Ramamoorthy, S. Dasari, S. Belkhale, S. Park, S. Nair, S. Mirchandani, T. Osa, T. Gupta, T. Harada, T. Matsushima, T. Xiao, T. Kollar, T. Yu, T. Ding, T. Davchev, T. Z. Zhao, T. Armstrong, T. Darrell, T. Chung, V. Jain, V. Kumar, V. Vanhoucke, W. Zhan, W. Zhou, W. Burgard, X. Chen, X. Chen, X. Wang, X. Zhu, X. Geng, X. Liu, X. Liangwei, X. Li, Y. Pang, Y. Lu, Y. J. Ma, Y. Kim, Y. Chebotar, Y. Zhou, Y. Zhu, Y. Wu, Y. Xu, Y. Wang, Y. Bisk, Y. Dou, Y. Cho, Y. Lee, Y. Cui, Y. Cao, Y.-H. Wu, Y. Tang, Y. Zhu, Y. Zhang, Y. Jiang, Y. Li, Y. Li, Y. Iwasawa, Y. Matsuo, Z. Ma, Z. Xu, Z. J. Cui, Z. Zhang, Z. Fu, and Z. Lin. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

[5] C. Gulino, J. Fu, W. Luo, G. Tucker, E. Bronstein, Y. Lu, J. Harb, X. Pan, Y. Wang, X. Chen, et al. Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. *Advances in Neural Information Processing Systems*, 36:7730–7742, 2023.

[6] R. Urtasan. Simulator realism: The new safety standard for the av industry, 2025. URL https://waabi.ai/simulator-realism-the-new-safety-standard-for-the-av-industry/.

[7] Reference for Simulator, author and organization redacted for review.

[8] S. Levine, A. Kumar, G. Tucker, and J. Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

[9] D. Precup, R. S. Sutton, and S. Singh. Eligibility traces for off-policy policy evaluation. In *ICML*, volume 2000, pages 759–766. Citeseer, 2000.

[10] N. Jiang and L. Li. Doubly robust off-policy value evaluation for reinforcement learning. In *International conference on machine learning*, pages 652–661. PMLR, 2016.

[11] P. Thomas and E. Brunskill. Data-efficient off-policy policy evaluation for reinforcement learning. In *International conference on machine learning*, pages 2139–2148. PMLR, 2016.

[12] A. B. Owen. *Monte Carlo theory, methods and examples*. https://artowen.su.domains/mc/, 2013.

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

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

[15] Z. Zhou, Y. Song, and A. Zanette. Accelerating unbiased llm evaluation via synthetic feedback. *arXiv preprint arXiv:2502.10563*, 2025.

[16] P. Boyeau, A. N. Angelopoulos, N. Yosef, J. Malik, and M. I. Jordan. Autoeval done right: Using synthetic data for model evaluation. *arXiv preprint arXiv:2403.07008*, 2024.

[17] D. Dauner, M. Hallgarten, T. Li, X. Weng, Z. Huang, Z. Yang, H. Li, I. Gilitschenski, B. Ivanovic, M. Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.

[18] K. T. e. a. H. Caesar, J. Kabzan. Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. In *CVPR ADP3 workshop*, 2021.

[19] M. Treiber, A. Hennecke, and D. Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review. E, Statistical physics, plasmas, fluids, and related interdisciplinary topics*, 62 2 Pt A:1805–24, 2000. URL https://api.semanticscholar.org/CorpusID:1100293.

[20] W. Ding, S. Veer, K. Leung, Y. Cao, and M. Pavone. Surprise potential as a measure of interactivity in driving scenarios. *arXiv preprint arXiv:2502.05677*, 2025.

A Proofs

Proof of Theorem 1. We start by deriving the variance of the control variate estimator for fixed $\boldsymbol{\beta}$.

$$\begin{aligned}
\mathrm{Var}(\hat{\mu}_{\mathrm{CV}}) &= \frac{1}{n}\mathrm{Var}(F - \boldsymbol{\beta}^T\boldsymbol{G}) + \frac{1}{k}\mathrm{Var}(\boldsymbol{\beta}^T\boldsymbol{G}) \\
&= \frac{1}{n}\left(\mathrm{Var}(F) - 2\boldsymbol{\beta}^T\mathrm{Cov}(\boldsymbol{G}, F) + \boldsymbol{\beta}^T\mathrm{Var}(\boldsymbol{G})\boldsymbol{\beta}\right) + \frac{1}{k}\boldsymbol{\beta}^T\mathrm{Var}(\boldsymbol{G})\boldsymbol{\beta} \\
&= \frac{1}{n}\mathrm{Var}(F) - \frac{2}{n}\boldsymbol{\beta}^T\mathrm{Cov}(\boldsymbol{G}, F) + \frac{n+k}{nk}\boldsymbol{\beta}^T\mathrm{Var}(\boldsymbol{G})\boldsymbol{\beta} \qquad (8)
\end{aligned}$$

Noting that $\mathrm{Var}(\hat{\mu}_{\mathrm{CV}})$ is a convex quadratic in $\boldsymbol{\beta}$, we determine the optimal $\boldsymbol{\beta}$ that minimizes this variance by differentiating (8) with respect to $\boldsymbol{\beta}$ and setting the result equal to zero, which gives:

$$\begin{aligned}
\mathbf{0} &= -\frac{2}{n}\mathrm{Cov}(\boldsymbol{G}, F) - 2\frac{n+k}{nk}\mathrm{Var}(\boldsymbol{G})\boldsymbol{\beta}_{\mathrm{opt}} \\
\boldsymbol{\beta}_{\mathrm{opt}} &= \frac{k}{n+k}\mathrm{Var}(\boldsymbol{G})^{-1}\mathrm{Cov}(\boldsymbol{G}, F).
\end{aligned}$$

To obtain the optimal variance, we plug this result back into (8):

$$\begin{aligned}
\mathrm{Var}(\hat{\mu}_{\mathrm{CV}}) &= \frac{1}{n}\mathrm{Var}(F) \\
&\quad - \frac{2}{n}\frac{k}{n+k}\mathrm{Cov}(F, \boldsymbol{G})\mathrm{Var}(\boldsymbol{G})^{-1}\mathrm{Cov}(\boldsymbol{G}, F) \\
&\quad + \frac{n+k}{nk}\left(\frac{k}{n+k}\right)^2\mathrm{Cov}(F, \boldsymbol{G})\mathrm{Var}(\boldsymbol{G})^{-1}\mathrm{Var}(\boldsymbol{G})\mathrm{Var}(\boldsymbol{G})^{-1}\mathrm{Cov}(\boldsymbol{G}, F) \\
&= \frac{1}{n}\left(\mathrm{Var}(F) - \frac{k}{n+k}\mathrm{Cov}(F, \boldsymbol{G})\mathrm{Var}(\boldsymbol{G})^{-1}\mathrm{Cov}(\boldsymbol{G}, F)\right) \\
&= \frac{1}{n}\left(1 - \frac{k}{n+k}\underbrace{\mathrm{Cov}(F, \boldsymbol{G})\mathrm{Var}(\boldsymbol{G})^{-1}\mathrm{Cov}(\boldsymbol{G}, F)\mathrm{Var}(F)^{-1}}_{:=\rho^2(\boldsymbol{G}, F)}\right)\mathrm{Var}(F).
\end{aligned}$$

We can simplify $\rho^2(\boldsymbol{G}, F)$ as

$$\begin{aligned}
\rho^2(\boldsymbol{G}, F) &= \mathrm{Cov}(F, \boldsymbol{G})\mathrm{Var}(\boldsymbol{G})^{-1}\mathrm{Cov}(\boldsymbol{G}, F)\mathrm{Var}(F)^{-1} \\
&= \mathrm{Tr}\left(\mathrm{Cov}(F, \boldsymbol{G})\mathrm{Var}(\boldsymbol{G})^{-1}\mathrm{Cov}(\boldsymbol{G}, F)\mathrm{Var}(F)^{-1}\right) \\
&= \mathrm{Tr}\left(\mathrm{Var}(F)^{-\frac{1}{2}}\mathrm{Cov}(F, \boldsymbol{G})\mathrm{Var}(\boldsymbol{G})^{-\frac{1}{2}}\mathrm{Var}(\boldsymbol{G})^{-\frac{1}{2}}\mathrm{Cov}(\boldsymbol{G}, F)\mathrm{Var}(F)^{-\frac{1}{2}}\right) \\
&= \left\|\mathrm{Var}(\boldsymbol{G})^{-\frac{1}{2}}\mathrm{Cov}(\boldsymbol{G}, F)\mathrm{Var}(F)^{-\frac{1}{2}}\right\|_{\mathrm{Fr}}^2,
\end{aligned}$$

where the first step is possible since $\rho^2(\boldsymbol{G}, F)$ is a scalar (since F is a scalar). □

Proof of Theorem 2. For any unbiased estimator $\hat{\mu}(\mathcal{D})$ of $\mu := \mathbb{E}[F]$ with variance $\mathrm{Var}(\hat{\mu})$, Chebyschev's inequality states that

$$\mathbb{P}\left(|\hat{\mu} - \mu| \geq \alpha\right) \leq \frac{\mathrm{Var}(\hat{\mu})}{\alpha^2}. \qquad (9)$$

Fixing a desired confidence level $1 - \delta$, the confidence interval over μ is,

$$\hat{\mu} \pm \sqrt{\frac{\mathrm{Var}(\hat{\mu})}{\delta}}, \qquad (10)$$

i.e., the confidence interval has a radius of $\sqrt{\frac{\mathrm{Var}(\hat{\mu})}{\delta}}$.

Let $n_r \in \mathbb{Z}_{>0}$ be the number of real-world samples of F for computing $\hat{\mu}_{\mathrm{MC}}$ (as in Eq. 2). Let $n_p \in \mathbb{Z}_{>0}$ be the number of real-world and surrogate data pairs $(F, \boldsymbol{G})$ drawn for computing $\hat{\mu}_{\mathrm{CV}}$ using control variates (as in Eq. 3 and 4).

First, we consider the case where $n_r = n_p = n$. Here, we have

$$\begin{aligned}\mathrm{Var}(\hat{\mu}_{\mathrm{CV}}) &= \frac{\mathrm{Var}(F)}{n}\left(1 - \frac{k}{k+n}\rho^2(\boldsymbol{G}, F)\right) \\ &\leq \frac{\mathrm{Var}(F)}{n} = \mathrm{Var}(\hat{\mu}_{\mathrm{MC}})\end{aligned}$$

since $\rho^2(\boldsymbol{G}, F) \in [0, 1]$. Plugging these into Eq. 10, we see that the consequence of this is that the confidence interval of the control variate estimator will be tighter than that of the simple Monte Carlo estimator.

Next, we fix a desired interval size α. To achieve a desired confidence in this interval of $1 - \delta$, we need

$$\delta = \frac{\mathrm{Var}(\hat{\mu})}{\alpha^2}.$$

For the MC estimator, we have

$$\delta = \frac{1}{\alpha^2}\frac{\mathrm{Var}(F)}{n_r} \tag{11}$$

$$n_r = \frac{\mathrm{Var}(F)}{\alpha^2 \delta}. \tag{12}$$

For the control variate estimator, we have

$$\delta = \frac{1}{\alpha^2}\frac{\mathrm{Var}(F)}{n_p}\left(1 - \frac{k}{k+n_p}\rho^2(\boldsymbol{G}, F)\right) \tag{13}$$

$$n_p \delta = \frac{\mathrm{Var}(F)}{\alpha^2} - \frac{k\mathrm{Var}(F)}{(k+n_p)\alpha^2}\rho^2(\boldsymbol{G}, F) \tag{14}$$

$$n_p(k+n_p) = (k+n_p)\frac{\mathrm{Var}(F)}{\alpha^2\delta} + \frac{k\mathrm{Var}(F)}{\alpha^2\delta}\rho^2(\boldsymbol{G}, F). \tag{15}$$

For a given confidence interval specified by α and δ, Eq. 12 relates $\mathrm{Var}(F)$, α, and δ to the number of samples required using simple Monte Carlo. To compute the minimum number of samples required to achieve the same interval with the control variate estimator, we substitute Eq. 12 into Eq. 15 above, and replace n_p with $n_{\min}$

$$n_{\min}(k + n_{\min}) = (k + n_{\min})n_r + kn_r\rho^2(\boldsymbol{G}, F) \tag{16}$$

$$n_{\min}k + n_{\min}^2 = n_r k + n_{\min}n_r + kn_r\rho^2(\boldsymbol{G}, F). \tag{17}$$

Solving this quadratic for $n_{\min}$, we have

$$n_{\min} = \frac{1}{2}\left(-(k - n_r) + \sqrt{(k-n_r)^2 + 4n_r k(1 - \rho^2(\boldsymbol{G}, F))}\right), \tag{18}$$

which gives us the required number of paired samples such that $\hat{\mu}_{\mathrm{CV}}$ achieves the same confidence interval as $\hat{\mu}_{\mathrm{MC}}$.

Finally, we show that $n_p \in [0, n_r]$. We begin by noting that $n_{\min}$ is monotonically decreasing in ρ^2. Hence, setting $\rho^2 = 0$, we attain the maximum value of $n_{\min} = n_r$, while setting $\rho^2 = 1$, we attain the minimum value $n_{\min} = 0$. As n_p is a continuous monotonic function in ρ^2, it will attain all values between its extrema of 0 and n_r. □

B Computing Sample Means and Variances

We computed $\hat{\mu}$ and $\widehat{\mathrm{Var}}$ in our experiments as follows.

The plug-in estimator for β_{opt} (a scalar in our experiments) is

$$\hat{\beta}_{opt} = \left(\frac{k}{k+n}\right) \frac{\sum_{i=1}^{n} \left(F_i - \bar{F}\right)\left(G_i - \bar{G}\right)}{\sum_{i=1}^{n} \left(G_i - \bar{G}\right)^2},$$

where

$$\bar{F} = \frac{1}{n}\sum_{i=1}^{n} F_i, \quad \bar{G} = \frac{1}{n}\sum_{i=1}^{n} G_i.$$

The plug-in estimator for $\hat{\mu}_\beta$ is

$$\hat{\mu}_{\hat{\beta}} = \frac{1}{n}\sum_{i=1}^{n}\left(F_i - \hat{\beta}\, G_i\right) + \hat{\beta}\,\hat{\theta},$$

where

$$\hat{\theta} = \frac{1}{k}\sum_{j=1}^{k} G'_j.$$

The sample variance of $\hat{\mu}_{\hat{\beta}}$ is then

$$\widehat{\mathrm{Var}}(\hat{\mu}_{\hat{\beta}}) = \frac{1}{n(n-1)}\sum_{i=1}^{n}\left(F_i - \hat{\beta}\, G_i - \bar{F} + \hat{\beta}\,\bar{G}\right)^2 + \frac{\hat{\beta}^2}{k(k-1)}\sum_{j=1}^{k}\left(G'_j - \hat{\theta}\right)^2.$$

Note that using the same data to estimate β_{opt} and to compute the control variate estimator introduces some bias; this is standard with control variates when β_{opt} is estimated from the same samples. However, this bias is typically quite small in practice – it is $O(1/n)$ and dominated by the $O(1/\sqrt{n})$ error between the estimator and the true mean [12]. If an unbiased estimate is needed, a simple approach is to estimate β_{opt} from separate data, such as held-out samples or data from prior policy evaluations (ensuring that none of the current policy's samples are used). This allows β_{opt} to be selected independently of the control variate estimate when n is very small and bias is a concern.

C Additional Experimental Details and Results

C.1 nuPlan Simulation Experiments

We generated paired open-loop and closed-loop driving data using the nuPlan simulator, instantiated with the Intelligent Driver Model (IDM) planner [19]. For each scene, an open-loop rollout records surrogate metrics under non-reactive agent behavior, while a corresponding closed-loop rollout – with the same planner and policy – captures metrics under reactive agent behavior.

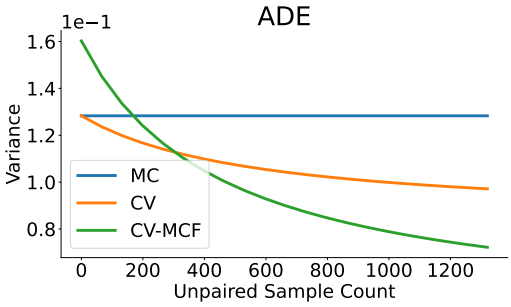
The metric correlator function (MCF) is implemented as a fully connected neural network with three hidden layers. Each hidden layer applies a ReLU activation. The final output layer uses a sigmoid activation to produce a prediction bounded in $[0, 1]$ for binary metrics. The network input comprises

$$\boldsymbol{G}_{i,train} = (\mathrm{ADE},\ \mathrm{TTC{<}1s},\ \mathrm{drivable_area_compliance},\ \mathrm{no_ego_at_fault_collisions}),$$

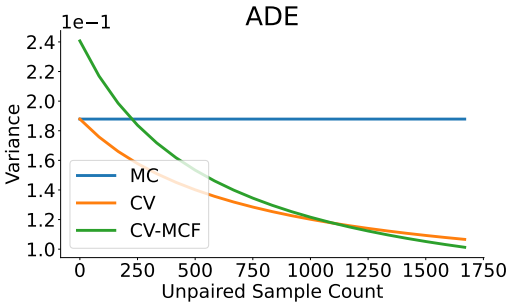
augmented by a feature vector $\phi(\cdot)$ extracted from a trained trajectory prediction model [20]. The MCF maps $(\boldsymbol{G}_{i,train},\ \phi)$ to a scalar estimate of the target closed-loop simulation metric for the corresponding sample.

The MCF was trained on n_{fit} paired samples from our target domain (for instance the "on_intersection," "near_multiple_vehicles," or "near_high_speed_vehicle" scenario class), with a mean squared error loss for continuous metrics, and a binary cross-entropy loss for binary metrics (with early stopping based on validation loss to prevent overfitting). In some experiments, the target domain data was supplemented with paired samples from auxiliary domains (data from non-target scenario classes), and in others, the MCF was trained exclusively on target domain pairs.

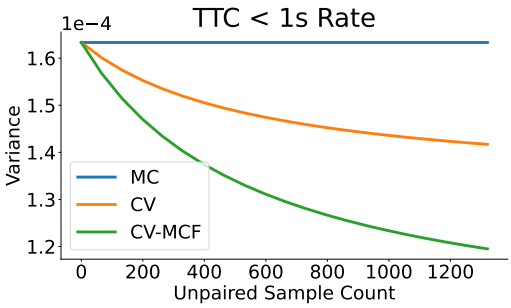
Fig. 6 compares the Monte Carlo (MC), control variates (CV), and control variates with MCF (CV–MCF) estimators across multiple metrics and domains as a function of the number of unpaired surrogate samples k. In all cases, estimator variance decreases as k increases. Panels (a) and



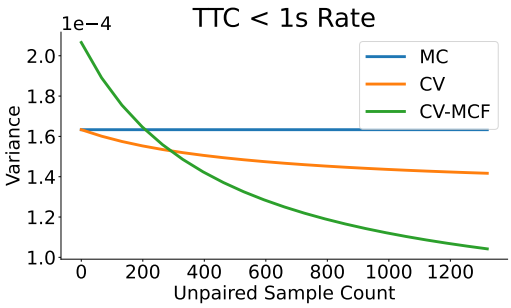
(a) ADE metric, "on_intersection" scenario class. MCF trained on in-domain pairs only. $n = 565$, $n_{\text{fit}} = 113$.



(b) ADE metric, "near_long_vehicle" scenario class. MCF trained on in-domain pairs only. $n = 715$, $n_{\text{fit}} = 143$.



(c) TTC $<$ 1s metric, "near_multiple_vehicles" scenario class. MCF trained on out-of-domain data only. $n = 432$, $n_{\text{fit}} = 0$ (+ out-of-domain data)



(d) TTC $<$ 1s metric, "on_intersection" scenario class. MCF trained on data from all domains. $n = 565$, $n_{\text{fit}} = 113$ (+ out-of-domain data)

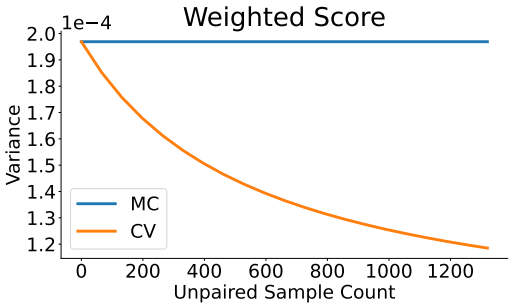
Figure 6: Variance vs. unpaired sample count for various metrics and domains. All plots averaged over 10 random trials.

(b) show results for models trained solely on target-domain pairs. In (a), the improved correlation from CV-MCF is high enough to outpace the CV approach, but in (b) it is not (until the unpaired sample count is much higher). Panels (c) and (d) correspond to models trained with auxiliary out-of-domain data pairs. Models trained on the out-of-domain data can generalize well to the in-domain data, resulting in lower variance estimates. Fig. 7 also plots the variance vs. the number of unpaired surrogate samples k, but uses the control variate vector (as in Eq. 5) rather than a scalar, with $\boldsymbol{G} = (\text{ADE}, \text{TTC}, \text{drivable_area_compliance}, \text{no_ego_at_fault_collisions})$ and $F = \text{aggregate_score}$. (The aggregate score is a weighted average of several individual metrics, and serves as an overall measure of driving performance.)

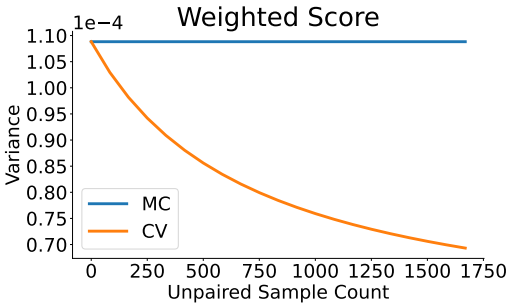
Fig. 8 plots the estimator variance as a function of the fraction of paired samples used in training the MCF (holding k constant). These plots confirm that increasing correlation strength (e.g. through using more samples for training) yields lower variance only when the increased correlation outweighs the reduction in available control variates samples. Note that in panels (a) and (b), the initial correlation is low, but even a small training set yields a substantial increase in correlation. Panels (c) and (d) show initial variance reduction from improved correlation, but this benefit diminishes as the number of paired samples available for the control variate computation decreases further, and becomes insufficient to offset further correlation gains.

C.2 Real-World Autonomous Driving Performance

Data was collected from on-road testing of an end-to-end autonomous vehicle policy. Non-overlapping events were selected according to domain-specific definitions, including intersections, curvy roads, and stop events. Each selected event was converted into a simulation scenario by replaying 2 seconds of recorded log data, followed by 8 seconds of closed-loop simulation. Metrics such as average lane-centering error and distance to the nearest other agent at 10 seconds were computed from both the in-vehicle (real-world) and simulated logs. Due to the high correlation observed

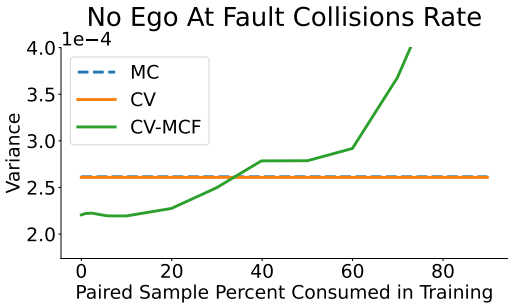


(a) Aggregate score, "on_intersection" class, $n = 565$, $\rho^2(\boldsymbol{G}, F) = 0.568$.

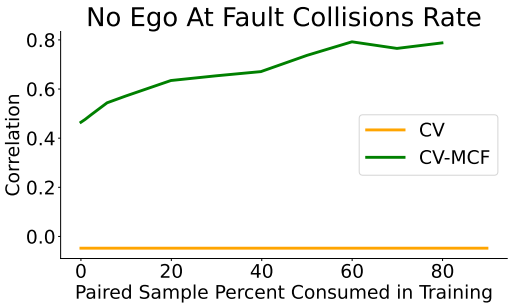


(b) Aggregate score, "near_long_vehicle" class. $n = 715$, $\rho^2(\boldsymbol{G}, F) = 0.515$.

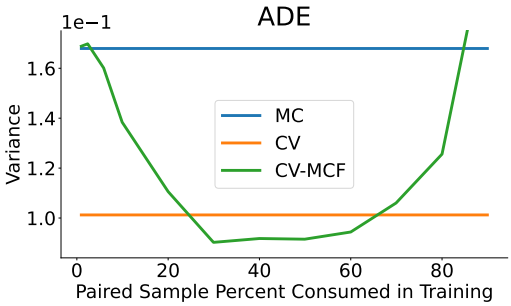
Figure 7: Variance vs. unpaired sample count for NuPlan aggregate score, using ADE, TTC, drivable area compliance, and no ego at fault collisions as a control variate vector. Plots averaged over 10 random trials.



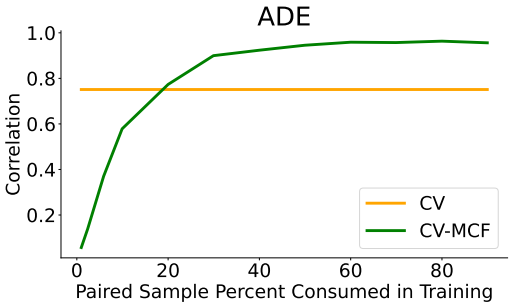
(a) No ego at fault collisions metric, "near_multiple_vehicles" scenario class. MCF trained on out-of-domain data plus increasing in-domain data; $n = 432$, $k = 1008$.



(b) Corresponding correlation for the no ego at fault collisions metric, "near_multiple_vehicles" scenario class; $n = 432$, $k = 1008$.



(c) ADE metric, "near_long_vehicle" scenario class. MCF trained on in-domain samples only; $n = 715$, $k = 1669$.



(d) Corresponding correlations for the ADE metric, "near_long_vehicle" scenario class; $n = 715$, $k = 1669$.

Figure 8: Variance and corresponding correlation vs. fraction of samples used to train the MCF for different metrics, scenario classes, and training approaches. All plots averaged over 10 random trials.

between simulated and real-world metrics, employing a metric correlator function did not provide additional benefit compared to the direct use of the control variate estimator.

C.3 Quadruped Velocity Tracking

Data was collected for the quadruped as follows. Define $F : \mathcal{X} \to \mathbb{R}$ as the real-world evaluation metric, computed through these steps:

1. Initialize the quadruped at zero velocity.
2. Deploy the reinforcement learning policy with the command velocity X.
3. Allow the policy to run on the quadruped for two seconds, recording the actual velocity at each timestep as $X^{(t)}$, for $t = 1, \ldots, T$, where $T = 100$ given a 50 Hz sampling rate.
4. Compute the average relative velocity tracking error:

$$\eta = \frac{1}{T} \sum_{t=1}^{T} \frac{\|X^{(t)} - X\|}{\|X\|}. \tag{19}$$

We conducted $n = 200$ paired simulation-real-world evaluations of η, supplemented by $k = 400$ simulation-only evaluations.

To train the MCF, we used a simple multilayer perceptron (MLP) that takes as input the concatenation of X and the simulation metric, and outputs a prediction of the real-world metric. The MLP consists of two hidden layers, each with 4 units. To ensure that the predicted outputs are bounded between 0 and 1 (consistent with the empirical range of the real-world metrics), we apply a sigmoid activation function at the output layer.

D Cost-Constrained Budget Allocation

In practice, collecting different types of measurements may incur different costs. For example, real-world measurements F_i are often expensive, while surrogate measurements $\boldsymbol{G}_i$ (e.g. from simulation) are comparatively inexpensive. Given a total cost budget C and per-sample costs:

- c_f: the cost of generating metric F_i
- c_g: the cost of generating surrogate $\boldsymbol{G}_i$

we can allocate the budget between n paired samples and k additional unpaired surrogate samples to minimize the variance of the control variate estimator:

$$\begin{aligned} &\min_{n,k \in \mathbb{Z}_{\geq 0}} \operatorname{Var}(\hat{\mu}_{\mathrm{CV}}) = \frac{\operatorname{Var}(F)}{n} \left(1 - \frac{k}{n+k} \rho^2\right) \\ &\text{subject to:} \quad nc_f + (n+k)c_g \leq C. \end{aligned} \tag{20}$$

Figure 9 illustrates the optimal (n, k) for various cost limits C. For large C, the optimal n–k relationship is approximately linear.

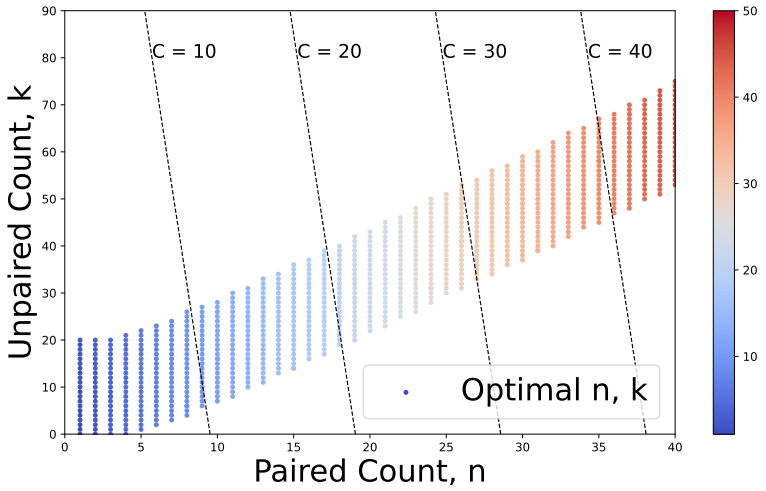
Closed-Form Solution (Continuous Relaxation). If we relax n and k to be positive real numbers, the optimal allocation can be derived in closed form. Using the cost constraint

$$nc_f + (n+k)c_g = C$$

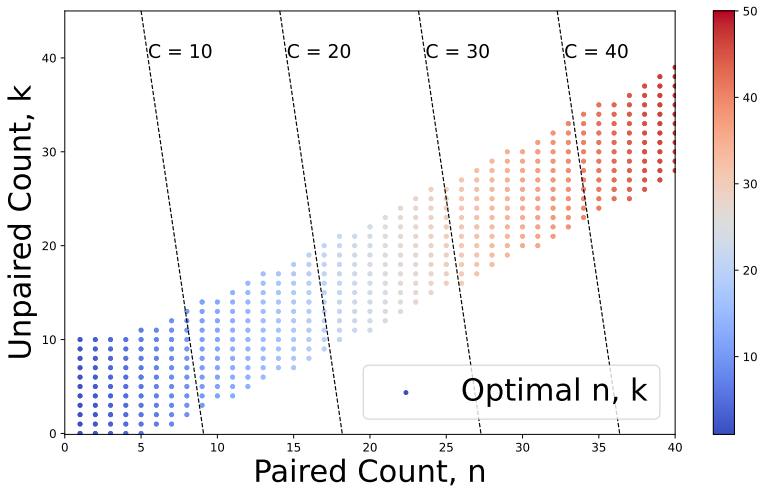
and substituting $k = C/c_g - (c_f + c_g)n/c_g$, minimizing Eq. 20 yields:

$$\begin{aligned} n^\star &= \frac{C\sqrt{1-\rho^2}}{c_f\sqrt{1-\rho^2} + \sqrt{c_f c_g}\,\rho}, \\ k^\star &= \frac{C\left(\sqrt{c_f c_g}\,\rho - c_g\sqrt{1-\rho^2}\right)}{c_g\left(\sqrt{c_f c_g}\,\rho + c_f\sqrt{1-\rho^2}\right)}. \end{aligned}$$

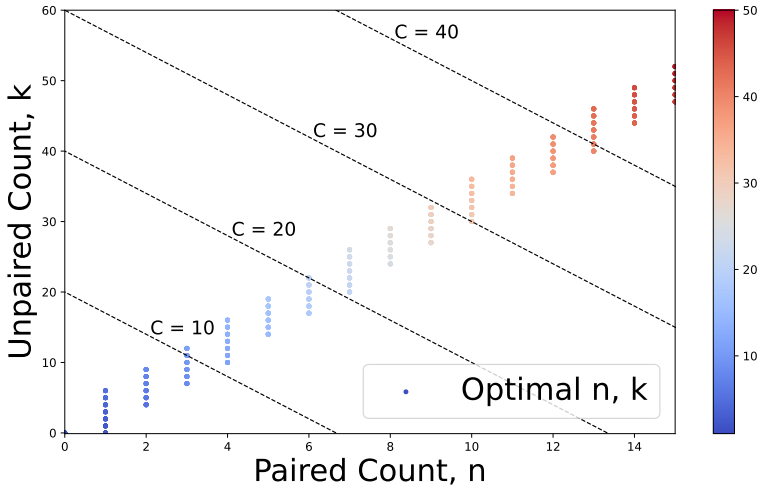
These values can be rounded to integers for practical use. These optimal values highlight that a higher correlation ρ favors collecting more inexpensive unpaired surrogate samples, while a lower ρ shifts the budget towards paired samples.



(a) $c_f = 1.0, c_g = 0.05, \rho = 0.5$.



(b) $c_f = 1.0, c_g = 0.1, \rho = 0.5$.



(c) $c_f = 1.0, c_g = 0.5, \rho = 0.95$.

Figure 9: Optimal solutions (n, k) to Eq. 20 for selected cost limits C. Lines of constant cost are shown at lower granularity.