Journal of Machine Learning Research 26 (2025) 1-52 Submitted 3/23; Revised 10/24; Published 3/25

Optimizing Data Collection for Machine Learning

Rafid Mahmood[1,2] RMAHMOOD@NVIDIA.COM
James Lucas[1] JALUCAS@NVIDIA.COM
Jose M. Alvarez[1] JOSEA@NVIDIA.COM
Sanja Fidler[1,3,4] SFIDLER@NVIDIA.COM
Marc T. Law[1] MARCL@NVIDIA.COM

[1] *NVIDIA*
[2] *University of Ottawa, Ottawa, Canada*
[3] *University of Toronto, Toronto, Canada*
[4] *Vector Institute, Toronto, Canada*

Editor: Isabelle Guyon

Abstract

Modern deep learning systems require huge data sets to achieve impressive performance, but there is little guidance on how much or what kind of data to collect. Over-collecting data incurs unnecessary present costs, while under-collecting may incur future costs and delay workflows. We propose a new paradigm to model the data collection workflow as a formal *optimal data collection problem* that allows designers to specify performance targets, collection costs, a time horizon, and penalties for failing to meet the targets. This formulation generalizes to tasks with multiple data sources, such as labeled and unlabeled data used in semi-supervised learning, and can be easily modified to customized analyses such as how to introduce data from new classes to an existing model. To solve our problem, we develop Learn-Optimize-Collect (LOC), which minimizes expected future collection costs. Finally, we numerically compare our framework to the conventional baseline of estimating data requirements by extrapolating from neural scaling laws. We significantly reduce the risks of failing to meet desired performance targets on several classification, segmentation, and detection tasks, while maintaining low total collection costs.

Keywords: data collection, neural scaling laws, active learning

1. Introduction

Before deploying a machine learning model in production, stakeholders may mandate that the model meets a pre-determined baseline performance, such as a target score over a validation data set. One of the most reliable ways to achieve a desired performance is by augmenting current training sets with more data. Consequently, engineers must regularly determine how much and what kind of data they need.

Managing data collection campaigns can impact costs and delays in model development. Overestimating how much data the model needs to reach a target performance will incur excess costs from collection, cleaning, and annotation. For example, annotating segmentation masks on driving data requires 15 to 40 seconds and between \$0.02 to \$0.08 per object (Acuna

et al., 2018; AWS, 2023). Processing 100,000 images with an average of 10 cars per image can take between 170 to 460 days-equivalent of time and cost between $20,000 to $80,000. Meanwhile, underestimating how much data is needed will incur delays from having to collect more data later. For instance, a 2019 survey of machine learning practitioners revealed 51% of engineering teams faced delays due to failing to collect enough data (Dimensional Research, 2019). This problem grows more challenging when given multiple vendors who can provide different levels of quality at different prices. Consider the following examples:

• **Medical Imaging:** A medical imaging company is planning to deploy automatic segmentation software on their devices within the next three years. They need to achieve an 80% Intersection-over-Union (IoU) to meet clinical standards. The company will partner with hospitals and hire clinicians to collect and annotate patient data, which can be expensive. If the company overestimates how much data to collect, they will spend more than necessary, but if they underestimate, they may have to restart the collection process next year.

• **Autonomous Vehicles:** A startup working on autonomous vehicles needs to build an object detector in the next five years. This model must achieve a minimum mean Average Precision of 95% on their validation set, or else it will not be deployable and the company will lose $1,000,000 in revenue. Collecting high-quality training data requires employing drivers to record video and annotators to label the data, where the marginal cost of each image is approximately $5. Alternatively, the startup could use lower quality synthetic data at $1 per image. To manage their resources, the startup must plan how much real versus synthetic data they need at the start of each year.

The extant literature on learning curves and neural scaling laws suggests that the relationship between the performance of a deep learning model and training data set size follows a power law (Frey and Fisher, 1999; Gu et al., 2001; Hestness et al., 2017; Rosenfeld et al., 2020; Kaplan et al., 2020; Hoiem et al., 2021; Bahri et al., 2024; Bisla et al., 2021). This motivates an intuitive approach of fitting a power law learning curve with the performance statistics of the current data set, extrapolating this learning curve to estimate how the model will perform with more data, and then forecasting how much data is needed to reach the desired performance (Rosenfeld et al., 2020). However, the decay rate of power laws implies that even small errors in estimating a learning curve can lead to massively over- or underestimating how much data is actually needed (Mahmood et al., 2022b). Moreover, estimating these learning curves becomes difficult with multiple data vendors, since different types of data have different costs and scale differently with performance (Mikami et al., 2022; Acuna et al., 2021; Prakash et al., 2021; Acuna et al., 2022; Prabhu et al., 2023). For example, in a semi-supervised learning task, unlabeled data may be easier to collect than labeled data, but we may require an order of magnitude more unlabeled data to match the performance of a small labeled set (Viering and Loog, 2022). Thus, collecting more data based only on scaling law estimates will fail to capture uncertainty and collection costs.

In this paper, we propose a new paradigm to model the data collection workflow as an *optimal data collection problem*, where a firm must minimize the cost of collecting enough data to obtain a model capable of achieving a desired performance score. They have multiple collection rounds, and after each round, they re-evaluate their model and decide how much more data to order. There are per-sample collection costs, and the firm pays a penalty if they fail to meet the target score within a finite horizon. Using this framework, we develop

an optimization approach to minimize expected future costs and show that this problem can be optimized in each collection round via gradient descent.

Our optimization problem generalizes to decisions over multiple data sources (e.g., unlabeled, long-tail, cross-domain, synthetic) that have different costs and impacts on performance. Most importantly, our framework presents natural tools for custom economic analyses such as comparing different collection strategies or introducing new classes to expand an existing model. Finally, we demonstrate the value of optimization over naïvely estimating data set requirements for several machine learning tasks and data sets. Our contributions are as follows:

1. We propose the *optimal data collection* problem in machine learning, which extends the estimation of learning curves to a formal dynamic optimization problem to determine how much and what kind of data to collect over the model development life cycle.

2. We introduce Learn-Optimize-Collect (LOC), a learning-and-optimizing framework that minimizes expected collection costs and can be solved via gradient descent. This is the first exploration of optimizing data collection with multiple arbitrary data sources in machine learning, covering, for example, semi-supervised and long-tail learning.

3. We analyze the one-round problem of deciding how much data to collect. We show that this problem is equivalent to estimating a $(1-\epsilon)$-quantile of the distribution of the minimum data needed to meet the target. Under Gaussian assumptions, LOC guarantees lower regret than estimation-only baselines.

4. We evaluate LOC over classification, segmentation, and detection tasks to show, on average, approximately a $2\times$ reduction in the chances of failing to meet performance targets, versus estimation baselines. We also show the flexibility of our framework to solve customized managerial questions faced by engineering firms.

The overall goal of this work is to provide a high-level framework with which machine learning model developers can obtain policy insights on how much data to collect. In practice, these decisions are strategic in nature and are determined over long-term horizons (see our motivating examples above). Moreover, the final policy decisions are often informed via a combination of our data-driven approach and human expert judgment. Consequently, we validate our methodology via a breadth of experiments on different data sets, tasks, and different managerial scenarios.

A preliminary version of this article was published in Mahmood et al. (2022c). Our complete paper introduces theoretical analysis of the optimal data collection problem. First, we characterize the optimal solution space of the data collection problem (Theorem 1) and show that LOC yields the optimal solution to the problem (Lemma 2). Further, we extend the theoretical analysis of one-round data collection by deriving an analytic formula for the optimal solution as well as a regret bound under Gaussian assumptions (Proposition 5, Corollary 6, Lemma 7, Proposition 8). Finally, we present several new experiments, including comparisons with different learning curve estimators (Section C.2), analysis under active learning strategies (Appendix C.3), and two new empirical case studies (Appendix 6.4) to show how our high-level modeling approach can answer customized data collection challenges. In these experiments, we adapt LOC to estimate the potential costs and yield decisions when choosing between different data collection strategies.

2. Related work

This paper employs neural scaling laws to solve operational design problems. The problem of data-efficient learning is closely tied to active learning and statistical sample complexity. Below, we summarize the most relevant literature.

2.1 Learning Curves and Neural Scaling Laws.

According to the learning curve literature, the performance of a machine learning model on a validation set scales with the size of the training data set with diminishing marginal value in a way that can usually be modeled via concave functions (Cortes et al., 1993; Frey and Fisher, 1999; Provost et al., 1999; Meek et al., 2002; Tomanek and Hahn, 2008; Figueroa et al., 2012; Kolachina et al., 2012). Consequently, performance at large data scales can be extrapolated by estimating the learning curve at smaller scales, both from a point perspective and by capturing uncertainty (Domhan et al., 2015). Although the literature primarily considers a single-round estimate of data collection requirements, John and Langley (1996) propose dynamic sampling over multiple data collection rounds, similar to our view. Furthermore while the literature primarily explores estimating dataset sizes, Last (2007) propose an optimization approach for data collection. Their framework minimizes the data collection cost plus a penalty associated with the amount of generalization error of the downstream model. In contrast to these two works, we propose a multi-round data collection optimization problem that minimizes the collection cost plus a penalty associated with failure to achieve a performance requirement. We refer to Viering and Loog (2022) for a detailed review on learning curves.

The neural scaling law literature focuses on deep learning to empirically demonstrate a power law relationship between model performance and data set size (Hoiem et al., 2021; Bisla et al., 2021; Zhai et al., 2022; Caballero et al., 2023). For instance, Hestness et al. (2017) observe this property over vision, language, and audio tasks, Bahri et al. (2024) develop a theoretical relationship under assumptions on over-parametrization and the Lipschitz continuity of the loss, model, and data, and Rosenfeld et al. (2020) estimate power laws using smaller data sets and models to extrapolate future performance. Multi-variate scaling laws have also been considered for some specific tasks, for example in transfer learning from synthetic to real data sets (Mikami et al., 2022). Finally, Mahmood et al. (2022b) explore data collection by estimating the minimum amount of data needed to meet a given target performance over multiple rounds. Our paper extends these studies by introducing a sequential optimization problem where the amount of data to collect is optimized, rather than estimated, over multiple collection rounds and from multiple data sources that may feature different costs.

2.2 Active Learning

In active learning, a model sequentially collects data by selecting new subsets of an unlabeled data pool to label under a predetermined labeling budget that replenishes after each round (Settles, 2009; Sener and Savarese, 2018; Yoo and Kweon, 2019; Sinha et al., 2019; Mahmood et al., 2022a). In contrast, we focus on systematically determining an optimal

collection budget. Thus, our work complements active learning which can be used to collect data up to the budget determined by our optimization policy.

2.3 Statistical Learning Theory

The rich theoretical analysis on the sample complexity of machine learning explores worst-case bounds relating data set size and model performance, but these bounds are typically only tight asymptotically; we refer the reader to Section 7.1 of Viering and Loog (2022) for details on this challenge. Recent work have empirically analyzed these relationships (Jiang et al., 2020, 2021) For instance, Bisla et al. (2021) study generalization bounds for deep neural networks, provide empirical validation, and suggest using them to estimate data requirements. Ultimately, making data collection decisions based on worst-case bounds may be pessimistic and have consequences on collection costs.

2.4 Optimal Experiment Design

The topic of how to collect data, select samples, and design statistical experiments is well-studied in econometrics (Smith, 1918; Cohn, 1993; Emery and Nenarokomov, 1998). For instance, pseudo-random strategies such as Latin Hypercube Sampling may be more efficient than i.i.d. sampling in data collection for statistical tests (Viana, 2016). Similarly, Bertsimas et al. (2015) optimize the assignment of samples into control and trial groups to minimize inter-group variances. Most recently, Carneiro et al. (2020) optimize how many samples and covariates to collect in a statistical experiment by minimizing a treatment effect estimation error or maximizing t-test power. However, our focus on industrial machine learning applications differs from experiment design by having target performance metrics and continual rounds of collection and modeling.

2.5 Sequential Decision-Making

Our problem is a sequential decision-making problem with an unobservable state, i.e., we determine how much to collect in each round without knowing how much we will need. Although such problems can be formed as Partially Observable Markov Decision Processes (POMDPs) (Smallwood and Sondik, 1973; Puterman, 2014), the dimension of the state and action space combined with sampling limitations make such approaches untenable. We provide a detailed reformulation and discussion in Appendix A.

3. Main Problem

We first introduce a formal model of data collection in machine learning as a dynamic decision-making game played over multiple collection rounds. We then discuss challenges with current intuitive, but naïve approaches for determining how much data to collect.

3.1 Optimal Data Collection

Consider $K \in \mathbb{N}$ different data sources, where for each $k \in \{1, \ldots, K\}$, let z_k be a data point and let $\mathcal{D}^k$ be a data set of points generated from a fixed algorithm, such as i.i.d. sampling or an active learning strategy (Settles, 2009). We train a learning model with data

sets $\mathcal{D}^1, \ldots, \mathcal{D}^K$ and evaluate a score function $V(\mathcal{D}^1, \ldots, \mathcal{D}^K)$. For example, if the learning problem is binary image classification, let $K = 1$ where $z_1 := (x, y)$ corresponds to images $x \in \mathcal{X}$ and labels $y \in \{0, 1\}$, and $V(\mathcal{D}^1)$ is the validation set accuracy of a model trained on $\mathcal{D}^1$. Alternatively in semi-supervised learning, let $K = 2$ where $z_1 := (x_1, y_1)$ and $z_2 := x_2$, while $V(\mathcal{D}^1, \mathcal{D}^2)$ is the validation accuracy of a model trained with both data sets. We omit superscripts and subscripts unless necessary.

In general, we have training sets $\mathcal{D}^1_{q_{0,1}}, \ldots, \mathcal{D}^K_{q_{0,K}}$ of $q_{0,1}, \ldots, q_{0,K}$ points, respectively, a target score $V^* > V(\mathcal{D}^1_{q_{0,1}}, \ldots, \mathcal{D}^K_{q_{0,K}})$, and a horizon of T rounds. Let $\mathbf{q}_0 := (q_{0,1}, \ldots, q_{0,K})^\mathsf{T}$ be a vector of data set sizes and let $V_{\mathbf{q}_0} := V(\mathcal{D}^1_{q_{0,1}}, \ldots, \mathcal{D}^K_{q_{0,K}})$. For each $t \in \{1, \ldots, T\}$, we

(i) Determine how much data to have at the end of the round $\mathbf{q}_t := (q_{t,1}, \ldots, q_{t,K})^\mathsf{T}$.

(ii) Generate data until each $\mathcal{D}^k$ has $q_{t,k}$ points.

(iii) Re-train our learning model. If $V_{\mathbf{q}_t} \geq V^*$ or if $t = T$, we terminate.

In each round, we pay a cost $c_k > 0$ for each additional point generated for the k-th data set. Further, if we do not reach V^* after T rounds, we pay a penalty P. Let $\mathbf{c} := (c_1, \ldots, c_K)^\mathsf{T}$ be the cost vector. Then, the ***optimal data collection problem*** is

$$\begin{aligned}
\min_{\mathbf{q}_1 \leq \cdots \leq \mathbf{q}_T} \quad & \mathbf{c}^\mathsf{T}(\mathbf{q}_1 - \mathbf{q}_0) + \mathbb{1}\{V_{\mathbf{q}_1} < V^*\} \Big(\mathbf{c}^\mathsf{T}(\mathbf{q}_2 - \mathbf{q}_1) \\
& \quad + \mathbb{1}\{V_{\mathbf{q}_2} < V^*\} \Big(\mathbf{c}^\mathsf{T}(\mathbf{q}_3 - \mathbf{q}_2) \\
& \qquad \vdots \\
& \quad + \mathbb{1}\{V_{\mathbf{q}_{T-1}} < V^*\} \Big(\mathbf{c}^\mathsf{T}(\mathbf{q}_T - \mathbf{q}_{T-1}) \\
& \quad + P\mathbb{1}\{V_{\mathbf{q}_T} < V^*\} \Big) \cdots \Big) \Big) \qquad (1) \\
= \min_{\mathbf{q}_1 \leq \cdots \leq \mathbf{q}_T} \quad & \sum_{t=1}^{T} \mathbf{c}^\mathsf{T}(\mathbf{q}_t - \mathbf{q}_{t-1}) \prod_{s=1}^{t-1} \mathbb{1}\{V_{\mathbf{q}_s} < V^*\} + P \prod_{t=1}^{T} \mathbb{1}\{V_{\mathbf{q}_t} < V^*\} \qquad (2)
\end{aligned}$$

Problem (1) is defined recursively where the objective includes the cost of collecting additional data in each round t and then conditioned on not collecting enough data in that round, the problem continues to the next round. Problem (2) refactors the objective by extracting the action in each round t to be depedendent on $\prod_{s=1}^{t-1} \mathbb{1}\{V_{\mathbf{q}_s} < V^*\}$.

If we use randomized algorithms to train a learning model and to sample data, the score function must be a random variable. Moreover, a general observation is that the score function typically increases monotonically with data set size (Frey and Fisher, 1999; Sun et al., 2017; Rosenfeld et al., 2020). We combine these two into the following assumption.

Assumption 1 *The score function is a realization of a stochastic process $V_{\mathbf{q}} := V(\mathcal{D}^1, \ldots, \mathcal{D}^K)$ as a function of the data set size. Furthermore, $V_{\mathbf{q}}$ increases monotonically with $\mathbf{q}$.*

The score function may not always increase monotonically (Mohr et al., 2022), due to factors such as model mis-specification (Viering and Loog, 2022), active learning (Tomanek and Hahn, 2008), or mixing labeled and unlabeled data in semi-supervised learning (Cozman et al., 2002)). However, monotonic non-decreasing trends such as power laws has been consistently observed in large-scale deep learning systems (Hestness et al., 2017; Rosenfeld et al., 2020; Hernandez et al., 2021; Hoffmann et al., 2022). Furthermore, this assumption ensures $V_{\mathbf{q}_1} \leq \cdots \leq V_{\mathbf{q}_T}$, meaning $\prod_{s=1}^{t-1} \mathbb{1}\left\{V_{\mathbf{q}_s} < V^*\right\} = \mathbb{1}\left\{V_{\mathbf{q}_{t-1}} < V^*\right\}$, which allows us to simplify problem (2) to

$$\text{RHS } (2) = \min_{\mathbf{q}_1 \leq \cdots \leq \mathbf{q}_T} \sum_{t=1}^{T} \mathbf{c}^\mathsf{T} \left(\mathbf{q}_t - \mathbf{q}_{t-1}\right) \mathbb{1}\left\{V_{\mathbf{q}_{t-1}} < V^*\right\} + P\mathbb{1}\left\{V_{\mathbf{q}_T} < V^*\right\}. \tag{3}$$

As $V_{\mathbf{q}}$ is monotonically increasing, an intuitive strategy may be to collect the minimum amount of data such that $V_{\mathbf{q}} = V^*$. We refer to this amount as the *minimum data requirement*

$$\mathbf{D}^* := \arg\min_{\mathbf{q}} \left\{\mathbf{c}^\mathsf{T}\mathbf{q} \mid V_{\mathbf{q}} \geq V^*\right\}. \tag{4}$$

The minimum data requirement is also the stopping time of the stochastic process, i.e., a random variable that gives the lowest-cost index that passes V^*. We assume that problem (4) always has a solution, i.e., we can achieve V^* performance with a finite amount of data. Furthermore, we randomly pick a unique solution to break ties in (4). Below, we show that unless our penalty for failing to reach the target is too small, the optimal solution is to collect this minimum data requirement.

Theorem 1 *Suppose Assumption 1 holds and that* $\mathbf{q}_0 < \mathbf{D}^*$. *If* $P < \mathbf{c}^\mathsf{T}(\mathbf{D}^* - \mathbf{q}_0)$, *then an optimal solution to problem* (3) *is* $\mathbf{q}_1^* = \cdots = \mathbf{q}_T^* = \mathbf{q}_0$. *Otherwise, an optimal solution is* $\mathbf{q}_1^* = \cdots = \mathbf{q}_T^* = \mathbf{D}^*$.

Proof We prove for $P < \mathbf{c}^\mathsf{T}(\mathbf{D}^* - \mathbf{q}_0)$ by breaking into two cases. First, consider any solution $\mathbf{q}_1, \ldots, \mathbf{q}_T$ where $\mathbf{q}_T > \mathbf{q}_0$ but $V_{\hat{\mathbf{q}}_T} < V^*$. Inputting this solution to problem (3) incurs an objective function value of $\mathbf{c}^\mathsf{T}(\hat{\mathbf{q}}_T - \mathbf{q}_0) + P \geq P$, where the right-hand-side is the objective function value incurred by $\mathbf{q}_1^* = \cdots = \mathbf{q}_T^* = \mathbf{q}_0$.

Next, consider the case where $V_{\mathbf{q}_T} \geq V^*$. Let $\hat{t}$ indicate the smallest $\mathbf{q}_t$ for which $V_{\mathbf{q}_t} \geq V^*$. The objective function value is at least $\mathbf{c}^\mathsf{T}(\mathbf{q}_{\hat{t}} - \mathbf{q}_0) \geq \mathbf{c}^\mathsf{T}(\mathbf{D}^* - \mathbf{q}_0) > P$. Here, the first inequality follows from the definition of $\mathbf{D}^*$ in (4), and the second follows from our assumption. Therefore, $\mathbf{q}_T^* = \mathbf{q}_0$, which implies our unique optimal solution.

We now prove for $P \geq \mathbf{c}^\mathsf{T}(\mathbf{D}^* - \mathbf{q}_0)$. First, consider any solution $\mathbf{q}_1, \ldots, \mathbf{q}_T$ for which $V_{\mathbf{q}_T} < V^*$. This means that the objective function value is greater than $P \geq \mathbf{c}^\mathsf{T}(\mathbf{D}^* - \mathbf{q}_0)$, where the right-hand-side is the objective function value incurred by $\mathbf{q}_1^* = \cdots = \mathbf{q}_T^*$.

Second, consider any solution where $V_{\mathbf{q}_T} \geq V^*$. Let $\hat{t}$ indicate the smallest $\mathbf{q}_t$ for which $V_{\mathbf{q}_t} \geq V^*$. The objective function value is $\mathbf{c}^\mathsf{T}(\mathbf{q}_{\hat{t}} - \mathbf{q}_0) \geq \mathbf{c}^\mathsf{T}(\mathbf{D}^* - \mathbf{q}_0)$, where the inequality follows from the optimality of $\mathbf{D}^*$. ■

Theorem 1 shows that the penalty for failing to meet a target performance must be sufficiently high for the optimal data collection problem to be non-trivial. In practice, $\mathbf{c}$ and T are determined by the real costs and constraints of the machine learning project, but P simply

Scaling Law Estimator	$v(q;\boldsymbol{\theta})$
Power Law	$\theta_1 q^{\theta_2} + \theta_3$
Arctan	$\frac{200}{\pi} \arctan\left(\theta_1 \frac{\pi}{2} q + \theta_2\right) + \theta_3$
Logarithmic	$\theta_1 \log(q + \theta_2) + \theta_3$
Algebraic Root	$\frac{100q}{(1 + \lvert\theta_1 q\rvert^{\theta_2})^{1/\theta_2}} + \theta_3$

Table 1: Four common scaling law functions with learnable parameters $\boldsymbol{\theta} = \{\theta_1, \theta_2, \theta_3\}$ when $K = 1$. See Viering and Loog (2022) for an extensive list. For $K > 1$, we can add the scaling law for each data source according to (5).

Algorithm 1 Naïve Estimation of the Data Requirement

Input: Initial data set $\mathcal{D}^1, \ldots, \mathcal{D}^K$ of $\mathbf{q}$ points, Regression model $\hat{v}(q;\boldsymbol{\theta})$, Regression set size R.
Collect Performance Statistics($\mathbf{q}$)
  Initialize $\mathcal{R} = \emptyset$, $\hat{\mathcal{D}}^1 = \cdots = \hat{\mathcal{D}}^K = \emptyset$
  for $r \in \{1, \ldots, R\}$ **do**
    Sub-sample $\lfloor q_k / R \rfloor$ additional points from each $\mathcal{D}^k$ without replacement to augment $\hat{\mathcal{D}}^k$.
    Evaluate $V(\hat{\mathcal{D}})$ and update $\mathcal{R} \leftarrow \mathcal{R} \cup \left\{ \left(\lfloor \mathbf{q} r / R \rfloor, V(\hat{\mathcal{D}}) \right) \right\}$.
  end for
end
Estimate $\mathbf{D}^*$
  Fit regression model $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{r=1}^{|\mathcal{R}|} \left(V_{\mathbf{q}_r} - v(\mathbf{q}_r;\boldsymbol{\theta}) \right)^2$.
  Estimate the data requirement by solving $\hat{\mathbf{D}} = \arg\min_{\mathbf{q}} \left\{ \mathbf{c}^\mathsf{T} \mathbf{q} \;\middle|\; v(\mathbf{q};\boldsymbol{\theta}^*) \geq V^* \right\}$.
end
Output: Estimated $\hat{\mathbf{D}}$.

reflects how much a firm stands to lose from not meeting performance targets. As such, practitioners have freedom to tune this parameter when modeling data collection, where setting a high P suggests that a strong need to meet the target within the time horizon. We expand on this observation in our theoretical analysis in Section 5.

We assume in the rest of this paper that $P > \mathbf{c}^\mathsf{T}(\mathbf{D}^* - \mathbf{q}_0)$, which implies that the optimal amount of data to have is $\mathbf{D}^*$. However, $\mathbf{D}^*$ is unknown to us at the time of decision-making.

3.2 An Intuitive but Naïve Approach to Estimating the Data Requirement

The recent neural scaling law literature suggests that if we can model the score function $V_{\mathbf{q}}$, then we can estimate $\mathbf{D}^*$ directly and use this to determine how much data to collect. For instance, Rosenfeld et al. (2020); Hoiem et al. (2021); Caballero et al. (2023) fit parameters $\boldsymbol{\theta}$ to an estimator $v(\mathbf{q};\boldsymbol{\theta}) \approx \mathbf{V}_{\mathbf{q}}$ of the score. They subsample from the current data sets $\mathcal{D}^1_{q_{t,1}}, \ldots, \mathcal{D}^K_{q_{t,K}}$ to simulate small data set sizes, retrain the model, and evaluate the score. Repeating this process with R different training subsets yields a data set of training statistics $\mathcal{R} := \{\mathbf{q}_r, V_{\mathbf{q}_r}\}_{r=1}^R$, which can be used to solve a Least Squares minimization problem. Once fitted, $v(\mathbf{q};\boldsymbol{\theta}^*)$ can replace $V_{\mathbf{q}}$ in problem (4) to estimate the stopping time. Algorithm 1 summarizes the general steps.

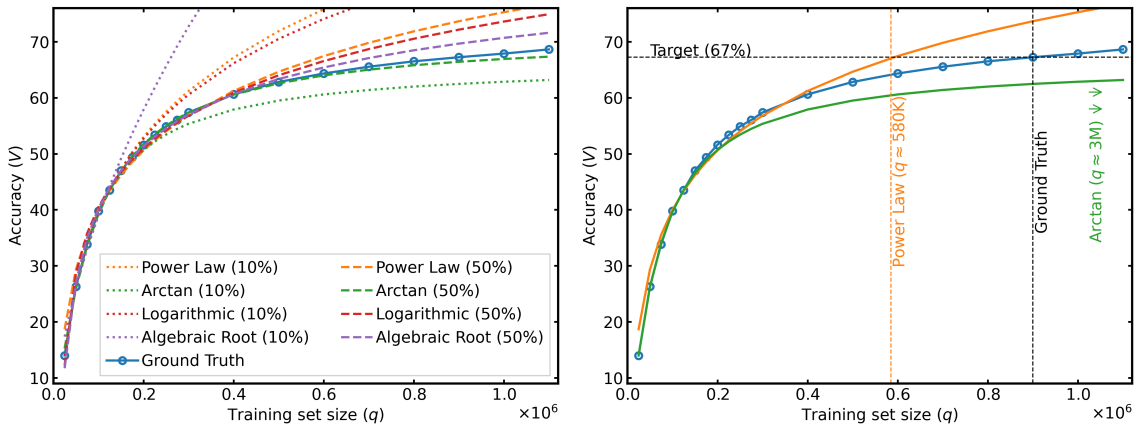

Figure 1: Extrapolating scaling laws to estimate D^* on ImageNet (Deng et al., 2009). The solid blue line is the ground truth test accuracy as a function of data set size. *Left:* Fitting four different scaling law functions from Table 1 when initializing with 10% ($q_0 = 125,000$, dotted) and 50% ($q_0 = 600,000$, dashed) of the data set. All functions struggle to accurately extrapolate accuracy when q_0 is small, but are accurate when q_0 is large. *Right:* To hit a target $V^* = 67\%$ accuracy, we need $900,000$ images. If the scaling laws over- or underestimate by only a small amount ($\leq 6\%$ error at $q = 900,000$), they massively under- (e.g., $\hat{D} = 580,000$) or overestimate (e.g., $\hat{D} = 3,000,000$) how much data is needed.

The scaling law literature almost exclusively focuses on $K = 1$, wherein they propose different parametric functions for $v(q;\boldsymbol{\theta})$(Rosenfeld et al., 2020; Viering and Loog, 2022; Hoiem et al., 2021; Caballero et al., 2023); we give examples in Table 1. The most common choice is a power law $v(q;\boldsymbol{\theta}) = \theta_1 q^{\theta_2} + \theta_3$. We remark that some recent research has explored $K > 1$; for instance, Mikami et al. (2022) explore a $K = 2$ power law for transfer learning from synthetic to real data. To explore arbitrary K, we propose an easy-to-implement estimator that adds the contributions of each data source

$$v(\mathbf{q};\boldsymbol{\theta}) := \theta_0 + \sum_{k=1}^{K} v_k(q_k;\boldsymbol{\theta}_k), \tag{5}$$

where $v_k(q_k;\boldsymbol{\theta}_k)$ can be any $K = 1$ estimator of the learning curve. Moreover, this additive model can be easily fit with a Least Squares algorithm and offers interpetable explanations of the contributions of different data types by assumping each data set has an independent contribution (Ghorbani and Zou, 2019).

Remark 1 *The above estimator extends the existing literature to arbitrary $K \in \mathbb{N}$ and can also be compounded on future $K = 1$ estimation algorithms. Further, different estimation functions $v(q;\boldsymbol{\theta})$ demonstrate specific biases towards either over- or underestimating performance (Mahmood et al., 2022b). Designing a specific estimator is not the focus of this paper, as we will next show that all estimators have the same weaknesses with respect to data collection, i.e., disproportionate over- or undercollection due to the concavity of the learning curve. For the remainder of the main paper, we focus on the specific case of $v(q;\boldsymbol{\theta})$*

being a linear combination of power laws, since power laws are the most common choice for learning curves. In Appendix C, we include ablations with different estimators (see Table 7).

Due to the fact that estimation will have some degree of inaccuracy, this intuitive strategy naïvely suffers from two major consequences when used in data collection. These properties were initially observed by Mahmood et al. (2022b), but we expand on their analysis below. We highlight them using the ImageNet data set in Figure 1:

1. **Extrapolation performance is tied to current amount of data:** Figure 1 (Left) plots four different scaling law functions that were fit using an initial $q_0 = 125,000$ and $q_0 = 600,000$ images, i.e., 10% and 50% of the entire data set respectively. When q_0 is small, every function fails to accurately extrapolate future performance and ultimately diverges from the ground truth learning curve. Alternatively when q_0 is large, every function reasonably accurately follows the learning curve. Thus, an initial estimate of D^* from a small q_0 is likely to be inaccurate. This property was also observed by Hestness et al. (2017), who referred to this as the the small-data regime.

2. **Small estimation errors lead to large over- or undercollection:** Figure 1 (Right) plots two different scaling law functions. To collect a target $V^* = 67\%$, the ground truth D^* is 900,000 images. Both estimated $v(q; \boldsymbol{\theta})$ functions are reasonably accurate and have less than 6% error at $q = 900,000$. However, the function that overestimates V_q (orange) suggests collecting 580, 000 images, which is about 60% of the true amount, whereas the function underestimating V_q (green) suggests collecting 3 million images, which is over three times the true amount. Finally, note that the magnitude of error is much larger in this example when we underestimate $v(q; \boldsymbol{\theta})$ (i.e., the green curve). As q increases, both $v(q; \boldsymbol{\theta})$ and the estimated curve flatten. If the rate of change for both curves approach zero, the distance between the points at which the two curves respectively reach V^* can increase arbitrarily.

We conclude that data collection policies that simply estimate how much data is needed for a given task can lead to paying arbitrarily large collection costs, even if the estimated learning curves are close to the true curves. Moreover, estimators diverge drastically from the true learning curves when given a limited amount of data. As a result, robust data collection policies must also capture the uncertainty of estimation.

4. Learn-Optimize-Collect (LOC)

Our solution approach, which we refer to as Learn-Optimize-Collect (LOC), minimizes the total collection cost while incorporating the uncertainty of estimation. To facilitate this problem, we assume that $\mathbf{D}^*$ is a continuous random variable that has a well-behaved, differentiable cumulative density function.

Assumption 2 *The random variable $\mathbf{D}^*$ is absolutely continuous and has a differentiable cumulative density function (CDF) $F(\mathbf{q}) := \Pr\{\mathbf{D}^* \leq \mathbf{q}\}$ and probability density function (PDF) $f(\mathbf{q}) := dF(\mathbf{q})/d\mathbf{q}$.*

Under this assumption, we will optimize over a continuous decision space for $\mathbf{q}$ and round any non-integer values. Although in practice, $\mathbf{D}^*$ is discrete, it is often realized on the order

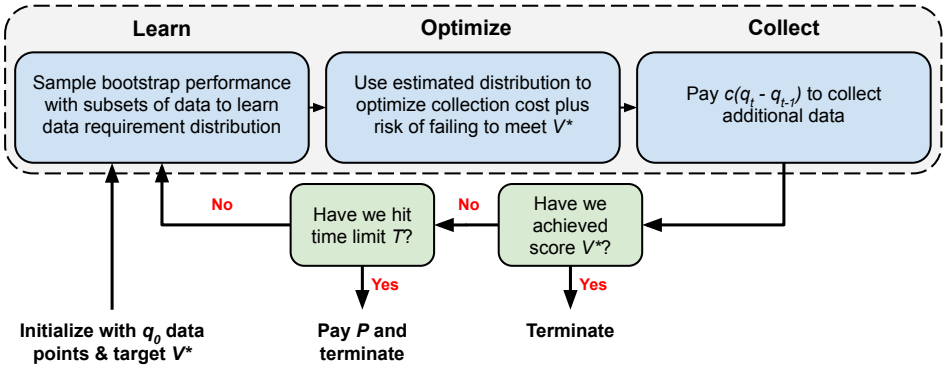

Figure 2: In the optimal data collection problem, we iteratively determine the amount of data that we should have, pay to collect the additional data, and then re-evaluate our model. Learn-Optimize-Collect (LOC) optimizes for the minimum amount of data q_t^* to collect.

of thousands or greater, which makes the assumption of continuity mild and rounding errors generally minor.

In this section, we first propose a stochastic optimization alternative to the optimal data collection problem (3). We then develop our framework, where we estimate the probability distribution of $\mathbf{D}^*$ and optimize over the estimated distribution.

4.1 A Stochastic Reformulation of Optimal Data Collection

Solving problem (3) directly is difficult because evaluating whether or not a given amount of data $\mathbf{q}$ is sufficient to reach V^* requires collecting the data itself and training the learning model. However, note that for any solution $\mathbf{q}$ to problem (3), if $\mathbf{q} \geq \mathbf{D}^*$, then by definition $V_{\mathbf{q}} \geq V^*$. As a result, consider the following approximation of the original problem:

$$\min_{\mathbf{q}_1 \leq \cdots \leq \mathbf{q}_T} \sum_{t=1}^{T} \mathbf{c}^\mathsf{T} (\mathbf{q}_t - \mathbf{q}_{t-1}) \mathbb{1}\{\mathbf{q}_{t-1} \ngeq \mathbf{D}^*\} + P\mathbb{1}\{\mathbf{q}_T \ngeq \mathbf{D}^*\}. \tag{6}$$

Problem (6) replaces the condition of achieving V^* from problem (3) with the condition of collecting at least $\mathbf{D}^*$ points over all of the data sources. We show below that when $K = 1$, these two problems are exactly equivalent. For general K, the two problems share the same optimal solution.

Lemma 2 *The following statements are true:*

1. *If $P < \mathbf{c}^\mathsf{T}(\mathbf{D}^* - \mathbf{q}_0)$, then an optimal solution to Problem (6) is $\mathbf{q}_1^* = \cdots = \mathbf{q}_T^* = \mathbf{q}_0$. Otherwise, an optimal solution is $\mathbf{q}_1^* = \cdots = \mathbf{q}_T^* = \mathbf{D}^*$.*

2. *If $K = 1$, then problem (6) is equivalent to problem (3).*

Proof The proof for the first statement is identical to the proof of Theorem 1. To prove the second statement note that from Assumption 1 and the fact that q is a scalar, $D^* = \arg\min_q\{q \mid V_q \geq V^*\}$. Thus, we have the following equivalence $q \geq D^* \Leftrightarrow V_q \geq V^*$.

Substituting this into the indicators completes the proof. ■

Lemma 2 states that $\mathbf{D}^*$ is an optimal solution to the approximation (6). Because this is also an optimal solution to the original problem, solving either problem to optimality can achieve the same decision. Furthermore, the total cost is the same for both problems when collecting up to $\mathbf{D}^*$ data points.

The approximation (6) relies on $\mathbf{D}^*$, which is not observable a priori. However, since $\mathbf{D}^*$ is a random variable, we can take the expectation over the objective in problem (6) to obtain

$$\begin{aligned}
&\min_{\mathbf{q}_1 \leq \cdots \leq \mathbf{q}_T} \mathbb{E}_{\mathbf{D}^* \sim f(\mathbf{q})} \left[\sum_{t=1}^{T} \mathbf{c}^\mathsf{T} (\mathbf{q}_t - \mathbf{q}_{t-1}) \mathbb{1} \{ \mathbf{q}_{t-1} \ngeq \mathbf{D}^* \} + P \mathbb{1} \{ \mathbf{q}_T \ngeq \mathbf{D}^* \} \right] \\
&= \min_{\mathbf{q}_1 \leq \cdots \leq \mathbf{q}_T} \sum_{t=1}^{T} \mathbf{c}^\mathsf{T} (\mathbf{q}_t - \mathbf{q}_{t-1}) (1 - F(\mathbf{q}_{t-1})) + P (1 - F(\mathbf{q}_T)), \qquad (7)
\end{aligned}$$

where $F(\mathbf{q})$ is the CDF of $\mathbf{D}^*$. Due to Assumption 2, the reformulated stochastic objective is differentiable in $\mathbf{q}$. We treat problem (7) to be continuous over $\mathbf{q}_1, \cdots, \mathbf{q}_T$ and round any non-integer values to determine the amount of data to collect. Therefore, this problem can be solved via gradient descent-based algorithms.

Finally, we remark on the interpretation of problem (7). Recall that problem (6) is equivalent to the original data collection problem (3) insofar as they both share an optimal solution $\mathbf{D}^*$. Furthermore, for any data collection decision $\mathbf{q}_t$, the CDF $F(\mathbf{q}_t)$ gives the probability that $\mathbf{q}_t \geq \mathbf{D}^*$. In the optimization problem (7), we determine how much data to collect in each round $\mathbf{q}_t$ by minimizing the probability of $\mathbf{q}_t \ngeq \mathbf{D}^*$, i.e., $1 - F(\mathbf{q}_t)$, multiplied by the cost of collecting this additional data and the penalty of failing to achieve the model within T rounds. This approach contrasts with the previous naïve estimator which directly used a point estimate of $\mathbf{D}^*$, by now incorporating the risk and potential cost of over- or under-collecting data due to the stochasticity in this random variable.

4.2 Estimating-then-Optimizing How Much Data to Collect

Solving problem (7) requires access to the distribution $F(\mathbf{q})$. However, just as we can estimate $\mathbf{D}^*$, we can now estimate the distribution and use this estimated distribution in the optimization problem.

We propose a simple strategy of estimating this distribution by bootstrapping the point estimates of $\mathbf{D}^*$ obtained via Algorithm 1. We first use the same steps as before to create a regression set of training statistics $\mathcal{R}$. Then, let $B > 1$ be the number of bootstrap estimates. For each $b \in \{1, \ldots, B\}$, we create a bootstrap resampled set of $\mathcal{R}$ and solve a corresponding Least Squares minimization problem to fit a scaling law estimator $v_b(q; \boldsymbol{\theta}_b)$ with parameters $\boldsymbol{\theta}_b$. We use this estimator in place of $V_\mathbf{q}$ in problem (4) to estimate the minimum data requirement. After repeating this process, we obtain a bootstrap set of estimates $\{\hat{\mathbf{q}}_b\}_{b=1}^B$, which we then use to fit a Kernel Density Estimator (KDE) $\hat{f}(\mathbf{q})$ of the PDF of the data requirement. Numerical integration of this KDE model yields the estimated CDF $\hat{F}(\mathbf{q}) := \int_0^\mathbf{q} \hat{f}(\mathbf{q}) d\mathbf{q}$. Algorithm 2 summarizes the steps.

Although there may exist several strategies for estimating this distribution, our bootstrapping procedure remains easy-to-perform and requires computation only to call Algorithm 1 for B rounds. In numerical experiments, we set $B = 500$ and find that this can generate

Algorithm 2 Estimating the Data Requirement Distribution $F(\mathbf{q})$

Input: Initial data set $\mathcal{D}_{\mathbf{q}}$, Regression model $\hat{v}(\mathbf{q};\boldsymbol{\theta})$, Regression set size R, Number of bootstrap samples B, Kernel Density Estimation (KDE) model $\hat{f}(\mathbf{q})$.
Initialize $\mathcal{R} = \emptyset$, $\hat{\mathcal{D}} = \emptyset$
Update $\mathcal{R}$ using the Collect Performance Statistics($\mathbf{q}$) sub-routine of Algorithm 1
Initialize $\mathcal{Q} = \emptyset$
Bootstrapped CDF($\mathcal{R}$)
  for $b \in \{1, \ldots, B\}$ **do**
    Create bootstrap $\mathcal{R}_b$ by sub-sampling R points with replacement from $\mathcal{R}$
    Fit regression model $\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_{(\mathbf{q},V_{\mathbf{q}})\in\mathcal{R}_b} \left(V_{\mathbf{q}} - v(\mathbf{q};\boldsymbol{\theta})\right)^2$
    Estimate the data requirement $\hat{\mathbf{q}}_b = \arg\min_{\mathbf{q}} \left\{\mathbf{c}^\mathsf{T}\mathbf{q} \mid v(\mathbf{q};\boldsymbol{\theta}^*) \geq V^*\right\}$
    Update $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\hat{\mathbf{q}}_b\}$
  end for
  Fit KDE model $\hat{f}(\mathbf{q})$ using the empirical distribution $\mathcal{Q}$ and let $\hat{F}(\mathbf{q}) := \int_0^{\mathbf{q}} \hat{f}(\mathbf{q}) d\mathbf{q}$
end
Output: Estimate of the requirement distribution $\hat{F}(\mathbf{q})$

high-quality distribution estimates (see Appendix C.1 for details). Most importantly, the first derivative of this CDF $\hat{F}(\mathbf{q})$ is immediately recoverable as the original KDE model $\hat{f}(\mathbf{q})$. This derivative is necessary for gradient descent optimization of problem (7).

4.3 Putting It All Together

We now describe the complete framework for optimizing the amount of data to collect in each round of the data collection problem. This framework uses a model-predictive-control approach of re-estimating the data requirement distribution, solving the stochastic optimization problem (8), and taking only the recommendation of how much data to collect for the immediate round. Figure 2 summarizes the steps of our algorithm, Learn-Optimize-Collect (LOC), which is also described in detail in Algorithm 3.

Given a target score V^* and an initial amount of data $\mathbf{q}_0$, we must collect data until we have met the target or until T rounds have passed. In the t-th round, we initialize with $\mathbf{q}_{t-1}$ data points over the K data sources. We first collect performance statistics by measuring the training dynamics of the current datasets $\mathcal{D}^1, \ldots, \mathcal{D}^K$. In each round t, rather than re-collecting the full performance statistic $\mathcal{R}$ in each round, we alternatively update $\mathcal{R}$ with the latest result $(\mathbf{q}, V_{\mathbf{q}_t})$; this significantly reduces the computational burden of collecting training statistics in each round. We then bootstrap these statistics to estimate the data requirement distribution for the t-th round, which we refer to as $\hat{F}_t(\mathbf{q})$.

Given an estimated distribution of $\mathbf{D}^*$, we first define the variables $\mathbf{d}_t \in \mathbb{R}^K$ where $\mathbf{d}_t := \mathbf{q}_t - \mathbf{q}_{t-1}$ and solve a reformulated version of problem (7), below

$$\min_{\mathbf{d}_1,\ldots,\mathbf{d}_T \geq \mathbf{0}} \sum_{t=1}^{T} \mathbf{c}^\mathsf{T}\mathbf{d}_t \left(1 - \hat{F}_t\left(\mathbf{q}_0 + \sum_{s=1}^{t-1} \mathbf{d}_s\right)\right) + P\left(1 - \hat{F}_t\left(\mathbf{q}_0 + \sum_{t=1}^{T} \mathbf{d}_t\right)\right). \tag{8}$$

The above problem (8) is differentiable on its domain, and the variables are only constrained to non-negativity. Thus, this problem can be treated as a continuous optimization problem with only non-negativity constraints, and can be optimized via projected gradient descent

Algorithm 3 Optimal Data Collection via LOC

Input: Initial data sets $\mathcal{D}^1, \ldots, \mathcal{D}^K$ of $\mathbf{q}_0$ points, Regression model $\hat{v}(\mathbf{q}; \boldsymbol{\theta})$, Regression set size R, Number of bootstrap samples B, Kernel Density Estimation (KDE) model $f(\mathbf{q})$, Target V^*, Maximum number of rounds T, Cost c, Penalty P.
Initialize round $t = 0$, Total cost $L = 0$
Initialize statistics $\mathcal{R} = \emptyset$, datasets $\hat{\mathcal{D}}^1 = \cdots = \hat{\mathcal{D}}^K = \emptyset$
repeat
    Update $\mathcal{R}$ using the Collect Performance Statistics($\mathbf{q}_t$) sub-routine of Algorithm 1
    Learn-Optimize-Collect
        Initialize $\mathcal{Q} = \emptyset$
        Update KDE model $\hat{F}_t(\mathbf{q})$ using the Bootstrapped CDF($\mathcal{R}$) sub-routine of Algorithm 2
        Freeze variables $\mathbf{d}_s$ for $s < t$ and solve problem (8) using $\hat{F}_t(\mathbf{q})$ to obtain $\mathbf{d}_t^*, \ldots, \mathbf{d}_T^*$.
    end
    Collect Data($\mathbf{d}_t$)
        for $k \in \{1, \ldots, K\}$ **do**
            Collect data z_k until $|\mathcal{D}^k| = q_{t,k} + d_{t,k}$
            Update cost $L \leftarrow L + c_k d_{t,k}$
        end for
        Re-train model and update performance $V_{\mathbf{q}_t}$
    end
    $t \leftarrow t + 1$
until $V_{\mathbf{q}_t} \geq V^*$ or $t = T$
if $V_{\mathbf{q}_T} < V^*$ **then**
    Update loss $L \leftarrow L + P$
end if
Output: Final collected data sets $\mathcal{D}^1, \ldots, \mathcal{D}^K$, Total cost L, Final model performance $V_{\mathbf{q}_t}$

algorithms. Finally, in each round t, we fix the previous decision variables $\mathbf{d}_1, \ldots, \mathbf{d}_{t-1}$ constant to their previously optimized values since we have already collected this data. Let $\mathbf{d}_t^*, \ldots, \mathbf{d}_T^*$ be the solution obtained from using a gradient method to optimize problem (8). We then collect data from each data source as determined by $\mathbf{d}_t^*$, i.e., the recommendation of how much data to collect immediately in round t. Once this data is collected, we re-train the machine learning model to evaluate the current performance $V_{\mathbf{q}_t}$ and proceed to the next round of the LOC pipeline.

5. Theoretical Insights in One-Round Data Collection

Although LOC can be used to generate long-term strategic data collection-term decisions over multiple rounds, a common use-case is to obtain a one-round $T = 1$ estimate of how much data is needed to meet the target V^*. For example, consider a firm that is deciding whether or not they should pursue a machine learning-based solution for a specific problem; to make an informed decision, the firm may want a single estimate of whether the amount of data needed to build the desired model is financially feasible. Such use-cases typically feature a single data type $K = 1$, a limited or potentially zero initial data q_0, and a noisy estimator $\hat{F}(q)$ of the data requirement distribution $F(q)$ (for example, see Section 3.2). This setting permits theoretical analysis wherein we can derive exact globally optimal solutions

for how much data to collect as well as structural insights to the relationships between costs, penalties, and the estimated data distribution.

In this section, we focus on the one-round, single-data source problem:

$$\min_{d_1} \; cd_1 + P\left(1 - \hat{F}(q_0 + d_1)\right). \tag{9}$$

In Section 5.1, we first develop an analytic solution to problem (9). This solution is dependent on the ratio c/P of the per-sample cost and how much we stand to pay in penalty for failing to collect. Then in Section 5.2, under the assumption that $\hat{F}(q)$ follows a Gaussian distribution, we fully characterize the analytic solution to obtain the globally optimal amount of data to collect. Finally in Section 5.3, we consider the case where $\hat{F}(q)$ is a noisy estimate of the *true* data requirement distribution $F(q)$. Here, we develop a regret bound to show that our optimization strategy outperforms the estimation-based approach (i.e., Algorithm 2) to data collection summarized in Section 3.2.

5.1 An Analytic Solution to One-Round Data Collection

The cost c parameter reflects real data collection costs whereas the penalty P reflects how much a firm stands to pay if they cannot obtain a model with performance V^*. Since this term may not have an exact real value, it may be difficult to determine the appropriate P in practice. Instead, consider an alternate, more intuitive parameter $\epsilon \geq 0$ to be a measure of the maximum tolerable probability of not meeting V^*. Since the data requirement D^* is stochastic, ϵ represents how much a firm is willing to tolerate the chance of undercollecting. That is, we should collect enough data d_1 such that $\hat{F}(q_0 + d_1) \geq 1 - \epsilon$. Our main theorem below states an optimal solution d_1^* to problem (9) in terms of this acceptable risk ϵ.

Theorem 3 *Assume $\hat{F}(q)$ is strictly increasing and continuous. If there exists $d_1 \geq 0$ where*

$$\frac{c}{P} \leq \frac{\hat{F}(q_0 + d_1) - \hat{F}(q_0)}{d_1} \tag{10}$$

then there exists an $\epsilon \leq 1 - \hat{F}(q_0)$ that satisfies $P = c/\hat{f}(\hat{F}^{-1}(1-\epsilon))$ and an optimal solution to problem (9) is $d_1^* := \hat{F}^{-1}(1-\epsilon) - q_0$. Otherwise, $d_1^* := 0$.

Before proving this result, we first summarize the intuition and consequences. Theorem 3 states that for the one-round $T = 1$ problem, when there is only a single data type $K = 1$, the optimal one-round estimate of the data requirement is determined by taking a $1 - \epsilon$ quantile of the distribution of D^*. Rather than solving the optimization problem (9), we can instead just use the estimated data requirement distribution $\hat{F}(q)$ and prescribe a maximum acceptable risk of failing to collect enough data $\epsilon := \Pr\{q_0 + d_1 < D^*\}$. We then collect exactly $d_1^* = \hat{F}^{-1}(1-\epsilon) - q_0$ additional points. The equivalence between simply prescribing a maximum risk ϵ and solving problem (9) is determined via choice of the cost and penalty parameters to satisfy $c/P = \hat{f}(\hat{F}^{-1}(1-\epsilon))$. Nonetheless, we remark that this equivalence only holds if the cost-to-penalty ratio c/P is sufficiently small and also only for $T = 1, K = 1$. Thus, problem (9) can also be seen as a generalization of Bayesian approaches to determine how much data to collect with respect to estimating the distribution of the learning curves and selecting a minimum pre-determined quantile (Domhan et al., 2015).

To prove Theorem 3, we will equate the original problem (9) to the following alternative *constrained optimal data collection problem*

$$\min_{d_1} \; c\, d_1 \qquad \text{s.t. } \hat{F}(q_0 + d_1) \geq 1 - \epsilon \,, \;\; d_1 \geq 0. \tag{11}$$

The above problem incorporates the maximum acceptable risk parameter ϵ as a constraint to replace the previous penalty-based formulation. We first prove that this problem is is convex and has an intuitive analytic solution.

Lemma 4 *Assume that $\hat{F}(q)$ is strictly increasing and continuous. Then,*

1. *Problem* (11) *is a convex optimization problem.*
2. *The unique optimal solution to problem* (11) *is $d_1^* = \max\{\hat{F}^{-1}(1-\epsilon) - q_0, 0\}$.*

Proof To prove the first statement, note that the objective and the second constraint are convex. Thus, we only need to prove that the set $\{d_1 \mid \hat{F}(q_0 + d_1) \geq 1 - \epsilon\}$ is a convex set. Since $\hat{F}(q)$ is strictly increasing in q, for any $\theta \in [0, 1]$ and $\hat{d} \geq d \geq 0$, we have $\hat{F}(q_0 + \theta\hat{d} + (1-\theta)d) \geq \hat{F}(q_0 + d) \geq 1 - \epsilon$. Because the convex combination of any two points is in the set, the set must be convex, which makes the optimization problem convex.

To prove the second statement, we consider two cases. First, if $\hat{F}^{-1}(1-\epsilon) \geq q_0$, let d_1 be the value that satisfies

$$q_0 + d_1 = \hat{F}^{-1}(1-\epsilon) := \inf\left\{q \mid \hat{F}(q) \geq 1 - \epsilon\right\},$$

which makes d_1 the smallest value to satisfy $\hat{F}(q_0+d_1) \geq 1-\epsilon$. Therefore, $d_1^* = \hat{F}^{-1}(1-\epsilon)-q_0$ is a minimizer in this case. In the second case where $\hat{F}^{-1}(1-\epsilon) < q_0$, we have $\hat{F}(q_0) > 1-\epsilon$. Since $d_1 \geq 0$ is a constraint, the minimizer is $d_1^* = 0$. ■

We prove Theorem 3 by showing that under (10), problems (9) and (11) are equivalent.

Proof of Theorem 3 First, note that when $T = 1$ and $K = 1$, problem (9) is a minimization of a continuous function over a single variable with a single non-negative constraint. Then, there must exist an optimal solution d_1^* that satisfies either $d_1^* = 0$ via the boundary condition, or $\hat{f}(q_0 + d_1^*) = c/P$ as a zero-gradient solution. Moreover, for the zero gradient solution to be optimal, there must exist some decision $d_1 > 0$ that achieves a lower objective than the zero solution. We rewrite this condition to

$$cd_1 + P(1 - \hat{F}(q_0 + d_1)) \leq P(1 - \hat{F}(q_0)) \Longrightarrow \frac{c}{P} \leq \frac{\hat{F}(q_0 + d_1) - \hat{F}(q_0)}{d_1}. \tag{12}$$

To derive the structure of the optimal solution, we rewrite the original problem (9) to

$$\min_{\epsilon} \min_{d_1} \; cd_1 + P\epsilon \qquad\qquad \text{s.t.} \quad \epsilon \geq 1 - \hat{F}(q_0 + d_1), \quad d_1 \geq 0, \tag{13}$$

where ϵ is a slack variable. However, for any fixed value of ϵ, the inner problem (13) is equivalent to the constrained alternative problem (11), meaning that it has an analytic

solution as determined by Lemma 4. Specifically, for any $\epsilon \geq 1 - \hat{F}(q_0)$, the optimal solution is $d_1^* = 0$, but for any $\epsilon \leq 1 - \hat{F}(q_0)$, the optimal solution is $d_1^* = \hat{F}^{-1}(1-\epsilon) - q_0$. Consequently, problem (13) can be further rewritten as

$$\min \left(\min_{\epsilon} \left\{ c\left(\hat{F}^{-1}(1-\epsilon) - q_0\right) + P\epsilon \mid \epsilon \leq 1 - \hat{F}(q_0) \right\},\ P(1 - \hat{F}(q_0)) \right). \tag{14}$$

Note that problem (14) has an equivalent optimality condition to (12). That is, for the left term to be the minimum, there must exist some $\epsilon \leq 1 - \hat{F}(q_0)$ that satisfies

$$c\left(\hat{F}^{-1}(1-\epsilon) - q_0\right) + P\epsilon \leq P(1 - \hat{F}(q_0)) \Longrightarrow \frac{c}{P} \leq \frac{1 - \hat{F}(q_0) - \epsilon}{\hat{F}^{-1}(1-\epsilon) - q_0} \tag{15}$$

Setting $d_1 = \hat{F}^{-1}(1-\epsilon) - q_0$ make conditions (12) and (15) equivalent. Most importantly, this must also hold for the minimizer $d_1^* = \hat{F}^{-1}(1-\epsilon^*) - q_0$. We can substitute this condition back into the zero-gradient solution of problem (9) to obtain

$$f\left(\hat{F}^{-1}(1-\epsilon)\right) = \frac{c}{P}, \qquad \epsilon \leq 1 - \hat{F}(q_0).$$

If such an ϵ does not exist, then the optimal solution must be $d_1^* = 0$ and $\epsilon^* = 1 - \hat{F}(q_0)$. ■

Finally, we remark on the assumption that $\hat{F}(q)$ be strictly increasing and continuous. Since in practice, $\hat{F}(q)$ is the integral of a KDE $\hat{f}(q)$, our assumption is always satisfied given an appropriate kernel (e.g., Gaussian). Furthermore as we show next, we can derive an exact formula under the case where the estimated data requirement distribution is Gaussian.

5.2 An Analytic Solution for Gaussian Distributions

Theorem 3 demonstrates an equivalence between optimizing how much data to collect given certain costs and penalties versus collecting data according to a minimum pre-specified risk tolerance ϵ. However, to use Theorem 3, we must compute both $\hat{F}(q)$ and $\hat{F}^{-1}(1-\epsilon)$. With a structural form of this CDF, we may further simplify Theorem 3 to obtain an exact analytic formula on how much data to collect.

We focus on the case where the estimated data requirement follows a Gaussian distribution $D^* \sim \mathcal{N}(\hat{\mu}, \hat{\sigma})$ and where $F(q) := 1/2 + \text{erf}((q - \hat{\mu})/(\sqrt{2}\hat{\sigma}))/2$ is the corresponding Gaussian CDF. The motivation for a Gaussian distribution stems from an observation that even when we estimate $\hat{F}(q)$ using KDE, the estimated distribution is often unimodular (see Appendix C.1 for details on this observation). As such, an intuitive strategy may be to simply estimate $\hat{F}(q)$ using a single Gaussian distribution rather than with many Gaussian kernels.

We first consider the case where our estimated CDF is exactly equal to the true CDF $\hat{F}(q) = F(q)$. Here, problem (9) simplifies to

$$\min_{d_1}\ cd_1 + \frac{P}{2}\left(1 - \text{erf}\left(\frac{q_0 + d_1 - \hat{\mu}}{\sqrt{2}\hat{\sigma}}\right)\right). \tag{16}$$

We can obtain an exact solution for any cost and penalty values. We first show that if the penalty P is too small, then the optimal solution to problem (16) is to collect no data,

i.e., the penalty for failing to achieve V^* is insufficient to outweigh the cost of collecting additional data. We then show that given a sufficiently large P, the optimal amount of data to collect can only take one of two possible values.

Proposition 5 *Suppose that $\hat{F}(q)$ models a Gaussian distribution $\mathcal{N}(\hat{\mu}, \hat{\sigma})$. If $P < c\hat{\sigma}\sqrt{2\pi}$, then the optimal solution to problem* (16) *is $d_1^* = 0$.*

Proof The gradient of the objective function of problem (16) is

$$c - \frac{P}{\sqrt{2\pi}\hat{\sigma}} \exp\left(-\frac{(q_0 + d_1 - \hat{\mu})^2}{2\hat{\sigma}^2}\right).$$

Setting this gradient to zero yields

$$d_1^* = \hat{\mu} \pm \sqrt{2}\hat{\sigma}\sqrt{\log \frac{P}{c\hat{\sigma}\sqrt{2\pi}}} - q_0.$$

However, this only exists if $P > c\hat{\sigma}\sqrt{2\pi}$; otherwise if $P < c\hat{\sigma}\sqrt{2\pi}$, then there is no zero-gradient solution to problem (16). Furthermore, by inspection, the objective function is monotone non-decreasing, meaning that the minimizer is the boundary point $d_1^* = 0$. ■

Proposition 5 states that if the penalty is less than $c\hat{\sigma}\sqrt{2\pi}$, then it is strategically better to not collect any data at all. Recall that P represents a potential loss from failing to develop a machine learning model that achieves V^* performance. Thus, not collecting additional data is equivalent to deciding not to build this model and simply accepting the penalty. This represents real-world scenarios where the costs of developing a machine learning model outweigh the benefits that this machine learning model can yield to the developer. If the penalty is sufficiently large such that $d_1^* > 0$, then building the machine learning model can be deemed useful.

The minimum value on the penalty parameter is dependent only on the cost and the standard deviation $\hat{\sigma}$ of the estimated data requirement distribution. When using LOC, if the variance of the estimated distribution is too large, then there is an increasingly larger chance that the minimum amount of data required D^* is low. If the penalty for undercollecting is low compared to this probability, then the optimal strategy would be to refrain from collecting data and correspondingly incur the penalty, rather than overcollect data. Figure 3 (Left) visualizes this trend by sweeping different values for P and plotting the objective function of problem (16). Here, if $P < \hat{\sigma}\sqrt{2\pi}$ (blue and orange curves), then the optimal $d_1^* = 0$.

Corollary 6 *Suppose that $\hat{F}(q)$ models a Gaussian distribution $\mathcal{N}(\hat{\mu}, \hat{\sigma})$ and $P > c\hat{\sigma}\sqrt{2\pi}$. Let $\zeta := \sqrt{\log P - \log(c\hat{\sigma}\sqrt{2\pi})}$. The optimal solution to problem* (16) *satisfies:*

1. *If $q_0 \leq \hat{\mu} - \sqrt{2}\hat{\sigma}\zeta$, then*

$$d_1^* = \begin{cases} \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta - q_0 & \text{if } \dfrac{c}{P} < \dfrac{\operatorname{erf}\zeta - \operatorname{erf}\left(\frac{q_0 - \hat{\mu}}{\sqrt{2}\hat{\sigma}}\right)}{2\left(\hat{\mu} + \sqrt{2}\hat{\sigma} - q_0\right)} \\ 0 & \text{otherwise} \end{cases}$$

2. *If* $\hat{\mu} - \sqrt{2}\hat{\sigma}\zeta < q_0 \leq \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta$, *then* $d_1^* = \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta - q_0$.

3. *If* $q_0 \geq \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta$, *then* $d_1^* = 0$.

Proof Following the same steps from the proof to Proposition 5, we find that the optimal solution must be a zero-gradient point or the boundary point

$$d_1^* \in \left\{0, \hat{\mu} \pm \sqrt{2}\hat{\sigma}\zeta - q_0\right\},$$

conditioned on whether the zero-gradient solution is feasible. Furthermore, we can show below that $\hat{\mu} - \sqrt{2}\hat{\sigma}\zeta - q_0$ is a local maximizer whereas $\hat{\mu} + \sqrt{2}\hat{\sigma}\zeta - q_0$ is a local minimizer via the second-order condition. Specifically, note that the second derivative of the objective function is

$$\frac{P\left(q_0 + d_1 - \hat{\mu}\right)}{\sqrt{2\pi}\hat{\sigma}} \exp\left(-\frac{\left(q_0 + d_1 - \hat{\mu}\right)^2}{2\hat{\sigma}^2}\right).$$

Substituting $d_1 = \hat{\mu} - \sqrt{2}\hat{\sigma}\zeta - q_0$ into the above equation admits a negative second derivative, whereas $d_1 = \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta - q_0$ admits a positive second derivative. Therefore, the optimal amount of data to collect can only be in

$$d_1^* \in \left\{0, \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta - q_0\right\}.$$

We now break the problem down into the three cases.

First, if $q_0 \leq \hat{\mu} - \sqrt{2}\hat{\sigma}\zeta$, then both the local minimizer and the boundary point are feasible. For $d_1^* = \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta$ to be the global minimizer, the condition on the c/P ratio from Theorem 3 must hold. That is,

$$\frac{c}{P} \leq \frac{\frac{1}{2}\left(1 + \operatorname{erf}\frac{q_0 + \sqrt{2}\hat{\sigma} - \hat{\mu}}{\sqrt{2}\hat{\sigma}}\right) - \frac{1}{2}\left(1 + \operatorname{erf}\left(\frac{q_0 - \hat{\mu}}{\sqrt{2}\hat{\sigma}}\right)\right)}{\hat{\mu} + \sqrt{2}\hat{\sigma}\zeta - q_0} = \frac{\operatorname{erf}\zeta - \operatorname{erf}\left(\frac{q_0 - \hat{\mu}}{\sqrt{2}\hat{\sigma}}\right)}{2\left(\hat{\mu} + \sqrt{2}\hat{\sigma}\zeta - q_0\right)}.$$

Otherwise, the optimal solution in this regime is $d_1^* = 0$.

Second, if $\hat{\mu} - \sqrt{2}\hat{\sigma}\zeta < q_0 \leq \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta$, then note that the objective function within the feasible region for d_1 consists of a curve with a single local minimum and no other zero-gradient points. Therefore, this local minimum $d_1^* = \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta$ must be the global minimizer.

Finally, if $q_0 \geq \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta$, then none of the zero-gradient points are feasible solutions. Therefore, the optimal solution must lie on the boundary point $d_1^* = 0$. ■

Corollary 6 gives an exact characterization of the optimal data collection problem when $\hat{F}(q)$ is modeled via a single Gaussian distribution. We first confirm that the penalty is sufficiently high for the problem to be meaningful; if the penalty for under-collection is too small, then it would incur lower costs to simply not collect any data. Managerially, this implies that the machine learning product is a poor investment as the costs of data collection are higher than simply not building the machine learning model.

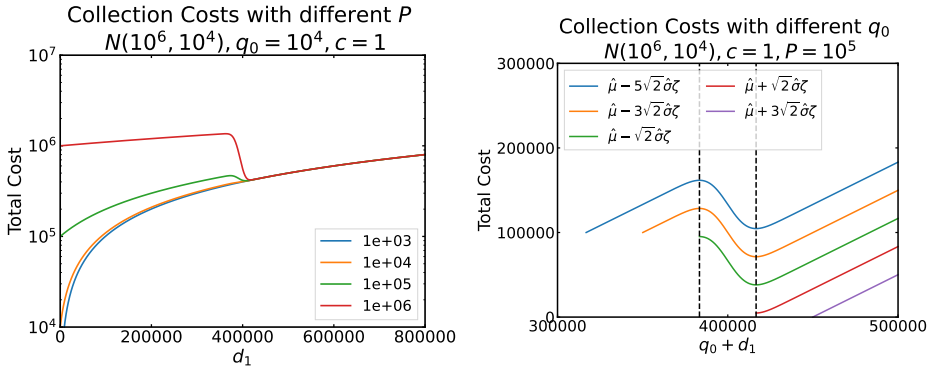

Figure 3: Evaluating problem (16). *Left:* We set $c = 1, q_0 = 10^4$ and sweep different values for P. If P is sufficiently small (i.e., $P < c\hat{\sigma}\sqrt{2\pi}$), then the total expected cost is minimized by setting $d_1 = 0$ (i.e., orange, blue curves). *Right:* We set $c = 1, P = 10^5$ and sweep different values for q_0. The dashed lines point to the local maxima and minima at $\hat{\mu} \pm \sqrt{2}\hat{\sigma}\zeta$. When $q_0 \geq \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta$, the optimal $d_1^* = 0$ (i.e., red, purple). When $q_0 \in [\hat{\mu} \pm \sqrt{2}\hat{\sigma}\zeta]$, the optimal d_1^* is at the zero-gradient minima (i.e., green, red). When $q_0 < \hat{\mu} - \sqrt{2}\hat{\sigma}\zeta$, the optimal d_1^* depends on the relationship between q_0, c, P. That is, $d_1^* = 0$ for the blue curve, but d_1^* is equal to the zero-gradient minima for the orange curve.

Assuming the penalty is sufficiently high, we can compare the value of the initial amount of data q_0 to the critical points $\hat{\mu} \pm \sqrt{2}\hat{\sigma}\zeta$. Specifically, if q_0 is greater than $\hat{\mu} + \sqrt{2}\hat{\sigma}\zeta$, then the optimal solution to the data collection problem is to not collect any data. This scenario is equivalent to setting a small value of P (c.f. Proposition 5), since ζ is a function of P, c, and $\hat{\sigma}$. On the other hand, if q_0 is between $\hat{\mu} - \sqrt{2}\hat{\sigma}\zeta$ and $\hat{\mu} + \sqrt{2}\hat{\sigma}\zeta$, then the optimal solution is always to collect data up to having $\hat{\mu} + \sqrt{2}\hat{\sigma}\zeta$ points. Finally, if q_0 is very small, i.e., less than $\hat{\mu} - \sqrt{2}\hat{\sigma}\zeta$, then we must again check if the ratio of cost over penalty is sufficiently small before collecting data. If this ratio is too high, then the penalty is again too low, and the optimal strategy would be to not collect any data and incur the penalty.

Figure 3 (Right) visualizes the implications of Corollary 6 by evaluating the objective function of problem (16) for different values of q_0; we plot with respect to $q = q_0 + d_1$, since the zero-gradient points for this formulation are always $\hat{\mu} \pm \sqrt{2}\hat{\sigma}\zeta$ for any q_0. Here, we observe that if $q_0 \geq \hat{\mu} - \sqrt{2}\hat{\sigma}\zeta$, the optimal total amount of data is $q^* = \max\left(\hat{\mu} + \sqrt{2}\hat{\sigma}\zeta, q_0\right)$ (i.e., the green, red, and purple curves). On the other hand if $q_0 < \hat{\mu} - \sqrt{2}\hat{\sigma}\zeta$, then the optimal amount of data to collect is equal to the zero-gradient minimizer only if q_0 is sufficiently large with respect to ζ, and therefore c and P (i.e., the orange curve). Otherwise $d_1^* = 0$ and $q^* = q_0$ (i.e., the blue curve). This condition for 'how large q_0 must be' can be most easily determined by checking whether the cost-to-penalty ratio c/P satisfies the conditions from Theorem 3.

5.3 Regret Bounds under Noisy Estimates of the Gaussian CDF

We have shown so far that in the single-round setting, the optimal decision d_1^* has an analytic solution, especially when the estimated $\hat{F}(q)$ of D^* follows a Gaussian distribution $\mathcal{N}(\hat{\mu}, \hat{\sigma})$.

In practice however, the estimated $\hat{F}(q) \neq F(q)$ will differ from the true CDF of D^*. In this sub-section, we assume that $D^* \sim \mathcal{N}(\mu, \sigma)$ follows a Gaussian distribution and $\hat{F}(q)$ is a noisy estimate where $\hat{\mu} \neq \mu$ and $\hat{\sigma} \neq \sigma$. Our goal is to study the expected regret $R(d_1)$ from a data collection decision d_1 made via the noisy estimate $\hat{F}(q)$ versus the true optimal solution D^* to problem (9):

$$\begin{aligned} R(d_1) :=& \mathbb{E}_{D^* \sim \mathcal{N}(\sigma,\mu)} \Big[c d_1 + P \mathbb{1} \left\{ q_0 + d_1 < D^* \right\} - c \left(D^* - q_0 \right) \Big] \\ =& c \left(d_1 - \mu + q_0 \right) + P \left(1 - F(q_0 + d_1) \right). \end{aligned}$$

This regret is measured in expectation over the true distribution of D^*.

We compare two data collection strategies on the amount of regret incurred. First, we consider LOC, which determines the amount of data to collect d_1^* from Corollary 6. Specifically, we focus on the case where $d_1^* = \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta - q_0$, rather than equal to 0. To compare against LOC, we consider the naïve estimation-only baseline from Section 3.2. This baseline sets $d_1 = \hat{\mu} - q_0$, which means it collects enough data to reach the estimated expected value of the minimum data requirement. We show that when the data collection decision is made using the LOC strategy, the expected regret is lower than if the d_1 is determined using the estimation-only approach.

We consider two scenarios, where the estimated distribution is noise-free (i.e., $\hat{\mu} = \mu, \hat{\sigma} = \hat{\sigma}$) and one where the estimated mean and variance differ from their true values (i.e., $\hat{\mu} \neq \mu, \hat{\sigma} \neq \sigma$). Below, we first consider the noise-free setting to demonstrate that LOC incurs a lower expected regret over the distribution of D^* versus the naïve estimation-only approach.

Lemma 7 *Suppose we perfectly estimate $F(q)$, i.e., $\hat{\mu} = \mu, \hat{\sigma} = \sigma$. Then, $R(d_1^*) \leq R(\hat{\mu} - q_0) = \frac{P}{2}$.*

Proof The inequality follows from the fact that d_1^* minimizes the objective $cd_1 + P(1 - F(q_0 + d_1))$. The equality follows from substituting $\hat{\mu} - q_0$ into the regret equation. ■

Lemma 7 states that even when we know the exact distribution D^*, it is disadvantageous to make data collection decisions from using only the estimate. Due to the inherent variability in the distribution, the expected regret is $P/2$ when using an estimation-only strategy. Specifically, using the expected value means that with 50% probability, the naïve strategy will under-estimate how much data to collect, and consequently incur the penalty. Instead, setting d_1^* according to Corollary 6 will incur a lower expected regret. Note that Lemma 7 holds for any symmetric probability distribution on D^* and not just the Gaussian setting.

We now present our main theoretical result, which explores the case where our estimated distribution $\mathcal{N}(\hat{\mu}, \hat{\sigma})$ of the minimum data requirement D^* differs from the true distribution $\mathcal{N}(\mu, \sigma)$. If the estimated distribution is sufficiently close to the true distribution, then the data collection decision generated by LOC will still incur a lower expected regret than a data collection decision generated by estimation alone. This sufficiency is characterized as an upper bound on the Total Variation distance between the two distributions, with respect to the cost, penalty, initial data, and estimated mean.

Proposition 8 *Let* $d_{TV} := \sup_A |\Pr_{D^* \sim \mathcal{N}(\mu,\sigma)}\{A\} - \Pr_{D^* \sim \mathcal{N}(\hat{\mu},\hat{\sigma})}\{A\}|$ *be the Total Variation distance between* $\mathcal{N}(\mu,\sigma)$ *and* $\mathcal{N}(\hat{\mu},\hat{\sigma})$*. If*

$$d_{TV} \leq \frac{c}{2P}(\hat{\mu} - q_0) - \frac{1}{2} \tag{17}$$

then, $R(d_1^*) \leq R(\hat{\mu} - q_0)$*.*

Proof We want to show

$$\begin{aligned}
0 &\geq R(d_1^*) - R(\hat{\mu} - q_0) && (18)\\
&= c\left(\hat{\mu} - \mu + \sqrt{2}\hat{\sigma}\zeta\right) + P\left(1 - F(\hat{\mu} + \sqrt{2}\hat{\sigma}\zeta)\right) - c(\hat{\mu} - \mu) - P(1 - F(\hat{\mu})) && (19)\\
&= c\sqrt{2}\hat{\sigma}\zeta + P\left(\Pr_{D^* \sim \mathcal{N}(\mu,\sigma)}\{\hat{\mu} + \sqrt{2}\hat{\sigma}\zeta < D^*\} - \Pr_{D^* \sim \mathcal{N}(\mu,\sigma)}\{\hat{\mu} < D^*\}\right) && (20)\\
&= c\sqrt{2}\hat{\sigma}\zeta - P\left(\Pr_{D^* \sim \mathcal{N}(\mu,\sigma)}\{\hat{\mu} \leq D^* < \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta\}\right). && (21)
\end{aligned}$$

Above, (19) applies the definition of $R(d_1^*)$ and $R(\hat{\mu} - q_0)$, (20) rewrites the CDFs and (21) collects the corresponding probabilities.

Note from Corollary 6 that $d_1^* = \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta$ implies it incurs a lower objective function value that $d_1^* = 0$ with respect to problem (9). That is,

$$c\left(\hat{\mu} + \sqrt{2}\hat{\sigma}\zeta - q_0\right) + P\left(\Pr_{D^* \sim \mathcal{N}(\hat{\mu},\hat{\sigma})}\{\hat{\mu} + \sqrt{2}\hat{\sigma}\zeta < D^*\}\right) \leq P\left(\Pr_{D^* \sim \mathcal{N}(\hat{\mu},\hat{\sigma})}\{q_0 < D^*\}\right). \tag{22}$$

By adding and subtracting $P\Pr_{D^* \sim \mathcal{N}(\mu,\sigma)}\{\hat{\mu} + \sqrt{2}\hat{\sigma}\zeta \leq D^*\}$ and $P\Pr_{D^* \sim \mathcal{N}(\mu,\sigma)}\{q_0 \leq D^*\}$, respectively, on both sides of the above equation and re-arranging the terms, we obtain

$$\begin{aligned}
&c\left(\hat{\mu} + \sqrt{2}\hat{\sigma}\zeta - q_0\right) - P\left(\Pr_{D^* \sim \mathcal{N}(\mu,\sigma)}\{q_0 < D^*\} - \Pr_{D^* \sim \mathcal{N}(\mu,\sigma)}\{\hat{\mu} + \sqrt{2}\hat{\sigma}\zeta < D^*\}\right)\\
&\quad \leq P\left(\Pr_{D^* \sim \mathcal{N}(\hat{\mu},\hat{\sigma})}\{q_0 < D^*\} - \Pr_{D^* \sim \mathcal{N}(\mu,\sigma)}\{q_0 < D^*\}\right)\\
&\qquad + P\left(\Pr_{D^* \sim \mathcal{N}(\mu,\sigma)}\{\hat{\mu} + \sqrt{2}\hat{\sigma}\zeta < D^*\} - \Pr_{D^* \sim \mathcal{N}(\hat{\mu},\hat{\sigma})}\{\hat{\mu} + \sqrt{2}\hat{\sigma}\zeta < D^*\}\right) \qquad (23)
\end{aligned}$$

Collecting the probability terms in the left-hand-side of (23) and bounding the right-hand-side by the total variation distance yields

$$\begin{aligned}
(23) \Rightarrow \quad & c\sqrt{2}\hat{\sigma}\zeta + c(\hat{\mu} - q_0) - P\left(\Pr_{D^* \sim \mathcal{N}(\mu,\sigma)}\{q_0 < D^* < \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta\}\right) \leq 2Pd_{TV} && (24)\\
\Rightarrow \quad & c\sqrt{2}\hat{\sigma}\zeta + c(\hat{\mu} - q_0) - P\left(\Pr_{D^* \sim \mathcal{N}(\mu,\sigma)}\{q_0 < D^* < \hat{\mu}\}\right)\\
& \qquad - P\left(\Pr_{D^* \sim \mathcal{N}(\mu,\sigma)}\{\hat{\mu} \leq D^* < \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta\}\right) \leq 2Pd_{TV} && (25)
\end{aligned}$$

Above (25) breaks the probability into two independent components between $[q_0, \hat{\mu})$ and $[\hat{\mu}, \hat{\mu} + \sqrt{2}\hat{\sigma}\zeta)$.

From (25), it remains only to prove that $2Pd_{TV} + P\left(\Pr_{D^* \sim \mathcal{N}(\mu,\sigma)}\{q_0 \leq D^* < \hat{\mu}\}\right) - c(\hat{\mu} - q_0) \leq 0$. Here, we rearrange the Total Variation distance bound (17) to

$$c(\hat{\mu} - q_0) \geq P(2d_{TV} + 1) \geq P\left(2d_{TV} + \Pr_{D^* \sim \mathcal{N}(\mu,\sigma)}\{q_0 \leq D^* < \hat{\mu}\}\right),$$

showing that the inequality is satisfied and proving inequality (21). ■

Proposition 8 guarantees that even when LOC uses a noisy estimate of $\hat{F}(q)$ of the true CDF $F(q)$ of the minimum data requirement, the optimization problem solved by LOC will yield, on average, lower cost and regret when compared to a strategy that only estimates D^* from the noisy $\hat{F}(q)$. This guarantee holds under condition (17), which states that the estimated distribution is sufficiently close to the true distribution. We first note that this upper bound can be computed from known values and furthermore, can be controlled by appropriate selection of the penalty parameter. Specifically, if we suspect the estimated $\hat{F}(q)$ is far from $F(q)$, then d_{TV} is large, and we can select a smaller P to ensure that LOC will generate a high-quality data collection decision. If we suspect that $\hat{F}(q)$ is close to $F(q)$ or if we observe that $\hat{\mu} - q_0$ is large, then we are safe to select a large P and still ensure that LOC will generate a high-quality decision. Finally, we note that the bound in Proposition 8 is a sufficiency condition only and not necessary for guaranteeing the quality of LOC-generated data collection decisions.

6. Empirical Results

In this section, we numerically evaluate LOC on data collection for six computer vision data sets spread over three tasks: image classification, object detection, and segmentation. We evaluate on $K = 1$ and $K = 2$, which are the two most common use-cases for obtaining high-level collection guidelines on data collection. In the first setting, a firm can only determine the data set size without focusing on the type of data. The second setting reflects a common problem class where data can be categorized into expensive and inexpensive types, e.g., long-tail versus common data or acquiring real versus auto-labeled data via semi-supervised learning. Our experiments reveal:

- **Section 6.2:** For every data set and task, LOC significantly reduces the number of instances where we fail to meet the data requirement V^*, while incurring only marginally more expensive costs compared to an oracle policy that determines exactly the minimum amount of data. In contrast, data collection decisions via estimation alone almost always leads to undercollecting and failing to meet V^*.

- **Section 6.3:** LOC is robust to both the cost and penalty parameters. For nearly all settings, modifying either parameter by orders of magnitude does not significantly affect the quality of decisions. Therefore in practical use, it is sufficient to roughly estimate the costs and penalties rather than obtain precise values.

- **Section 6.4:** LOC can be easily adapted to solve custom economic analyses and machine learning scenarios. We demonstrate two unique examples where a firm would like to (i) modify an existing machine learning model to now accommodate a new class;

and (ii) contrast two entirely different data collection policies. For both cases, LOC yields appropriate data collection decisions.

We expand on our experiment setup in Appendix B. We expand on the main results in Appendix C and include further ablations and comparisons against different baselines.

6.1 Experiment Setup

We explore three tasks for $K = 1$. First, we consider classification on CIFAR-10 (Krizhevsky, 2009), CIFAR-100 (Krizhevsky, 2009), and ImageNet (Deng et al., 2009), where we train ResNets (He et al., 2016) to meet a target validation accuracy. We explore semantic segmentation using Deeplabv3 (Chen et al., 2018) on BDD100K (Yu et al., 2020), which is a large-scale driving data set, as well as Bird's-Eye-View (BEV) segmentation on nuScenes (Caesar et al., 2020) using the 'Lift Splat' architecture (Philion and Fidler, 2020); both tasks require a target mean intersection-over-union (IoU). Finally, we explore 2-D object detection on PASCAL VOC (Everingham et al., 2007, 2012) using SSD300 (Liu et al., 2016), where we evaluate mean average precision (mAP).

We evaluate two tasks for $K = 2$. First, we mimic the scenario of long-tail or imbalanced learning where data for some classes (e.g., long-tail) is much more expensive to collect than others. Here, we divide CIFAR-100 into two subsets containing data from the first and last 50 classes, respectively, and assign a higher cost to the first 50 classes versus a lower cost to the last 50. Our second experiment models the scenario where one can acquire real labeled data versus use a prior model to cheaply autolabel data. Labeled data incurs a higher cost from collection plus annotation whereas autolabeled data incurs only collection costs, since autolabeled annotations are effectively free. We design this experiment using the labeled and unlabeled data splits of BDD100K.

We simulate the deep learning workflow following the procedure of Mahmood et al. (2022b), to approximate the true problem while simplifying the experiments (see Appendix B for details). To avoid repeatedly sampling data, re-training a model, and evaluating the score, this simulation uses a piecewise-linear approximation of a 'ground truth' learning curve that returns model performance as a function of data set size. We initialize with $\mathbf{q}_0 = 10\%$ of the full data set (we use 20% for VOC). In each round, we solve for the amount of data to collect and call the piecewise-linear learning curve to obtain the current score.

For each data set and task, we consider $T = 1, 3, 5$ rounds. For each T, we sweep $V^* \in [V(\mathcal{D}_{\mathbf{q}_0}) + 1, V(\mathcal{D})]$ where $\mathcal{D}$ is the entire labeled data set; we then aggregate the LOC performance over the full range of potential target settings. For example, in our experiments of optimizing the number of training examples of CIFAR-100 to collect, we find that the initial dataset $\mathcal{D}_{\mathbf{q}_0}$ achieves 42% accuracy and $\mathcal{D}$ achieves 75% accuracy; consequently, we sweep over all values of $V^* \in [42, 75]$ and evaluate the average performance of the data collection policies. We first evaluate policies on the *failure rate*, which is the fraction of V^* settings in which our data collection policy fails to achieve the target performance within T rounds. Second, we evaluate the *cost ratio*

$$\frac{\mathbf{c}^\mathsf{T}(\mathbf{q}_T^* - \mathbf{q}_0)}{\mathbf{c}^\mathsf{T}(\mathbf{D}^* - \mathbf{q}_0)} - 1,$$

which is the relative sub-optimality of the policy compared to an oracle policy that knows exactly $\mathbf{D}^*$ a priori. To ensure that this metric is non-negative and is scaled reasonably, we average the cost ratio over all settings of V^* in which the policy yields a model that exceeds V^*; i.e., this metric ignores instances that fail to collect enough data. For $K = 1$, we also measure the ratio of points collected q_T^*/D^* following Mahmood et al. (2022b). Although there is a trade-off between low cost ratio (undercollecting) and failure rate (overcollecting), we emphasize that our goal is to have low cost but with zero chance of failure.

Our primary baseline is the conventional estimation detailed in Section 3.2, which fits a regression model to the learning curve statistics, extrapolates the learning curve for more data, and then solves for the minimum data requirement under this extrapolation. We refer to this as the Power Law Regression approach. In addition, we also ablate the effect of optimizing the minimum data requirement by comparing against an intermediate estimation baseline which estimates the distribution of the minimum data requirement and selects the average estimated value. This baseline, referred to as Ensemble Regression, leverages the bootstrapping and density estimation procedure to learn the distribution.

There are many different regression models that can be used to fit learning curves (Jones et al., 2003; Figueroa et al., 2012; Hestness et al., 2017; Hoiem et al., 2021; Viering and Loog, 2022). Since power laws are the most commonly studied approach in the neural scaling law literature, we focus on power laws here, but compare against the other functions from Table 1 in Appendix C.2.

6.2 Main Results: The Value of Optimization over Estimation

Below, we discuss results for $K = 1$ and $K = 2$.

6.2.1 Case with $K = 1$

We first discuss experiments on $K = 1$. Figure 4 compares LOC versus the corresponding power law regression baseline when $c = 1$. We fix $P = 10^7$ as the default penalty for CIFAR-10, CIFAR-100, BDD100K, and nuScenes, but set $P = 10^8$ for ImageNet and $P = 10^6$ for VOC[1]. If a curve is below the black line, then it failed to collect enough data to meet the target. LOC consistently remains above this black line for most settings, whereas even with $T = 5$ rounds, collecting data based only on regression estimates leads to failure.

Table 2 aggregates failure rates and cost ratios for each setting. Here, LOC fails at less than 10% of instances for 12/18 settings, whereas regression fails over 30% for 15/18 settings. In particular, regression nearly always under-collects data when given a single $T = 1$ round. Here, LOC reduces the risk of under-collecting by 40% to 90% over the baseline. Moreover, our cost ratios are consistently less than 0.5 for 12/18 settings, meaning that we spend at most 50% more than the true minimum cost.

Table 2 also ablates the effect of optimization versus only estimating the distribution. The ensemble uses the same KDE distribution as in LOC but simply outputs the mean of the distribution. First, LOC always outperforms the Ensemble Regression baseline, showing that

1. This tuning is due to the fact that ImageNet naturally has an order of magnitude more data within it, meaning that the scale of data required to reach target performances is naturally higher; nonetheless in Section 6.3, we demonstrate that our LOC results are generally robust to order-of-magnitude shifts in the penalty parameter for most settings.

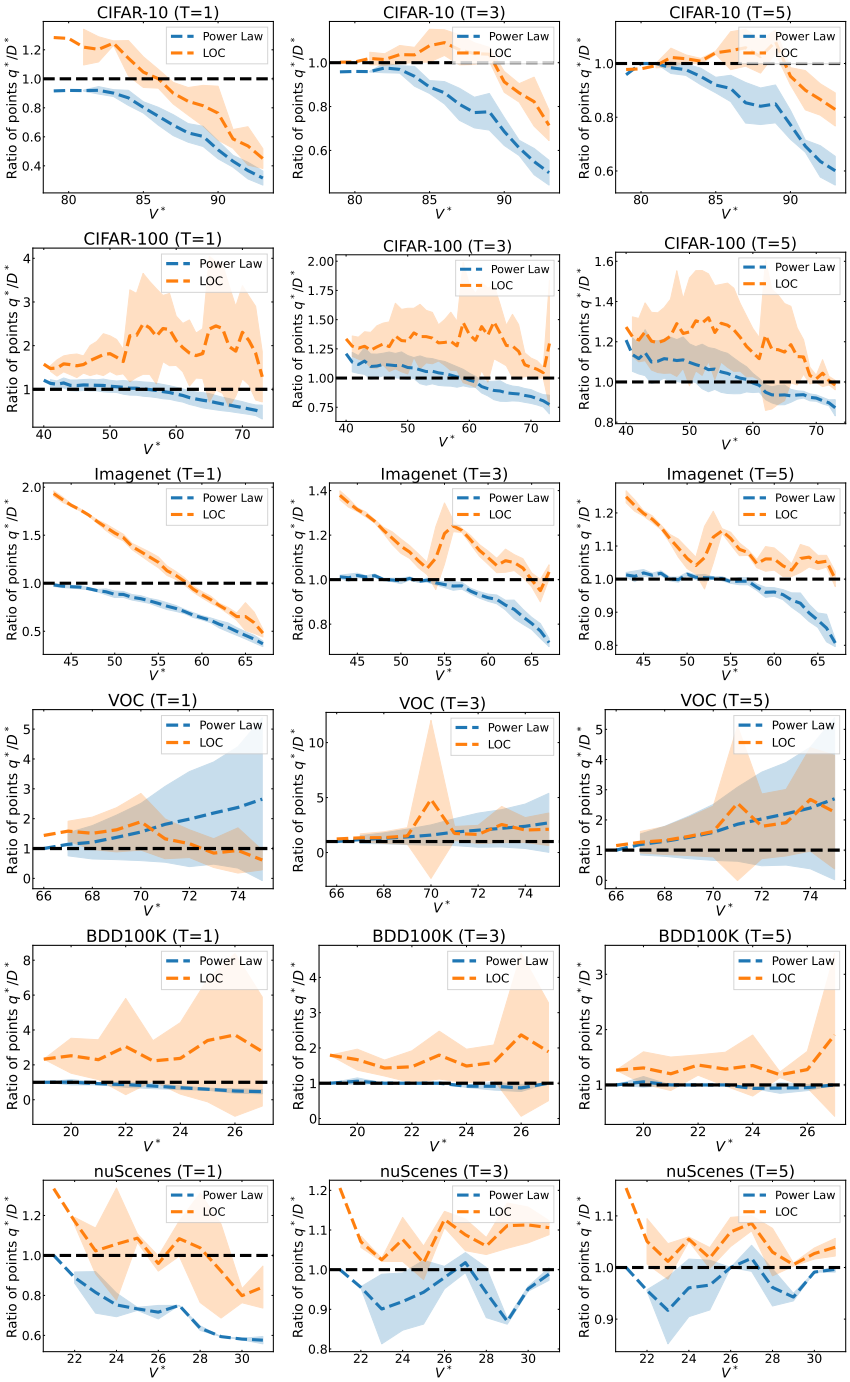

Figure 4: Mean ± standard deviation of 5 seeds of the ratio of data collected q_T^*/D^* for different V^*. The rows correspond to $T = 1, 3, 5$ and the columns to different data sets. The black line corresponds to collecting exactly the minimum data requirement. LOC almost always remains slightly above the black line, meaning we rarely fail to meet the target.

	Data set	T	Power Law Regression		Ensemble Regression		LOC (Ours)	
			Failure rate	Cost ratio	Failure rate	Cost ratio	Failure rate	Cost ratio
Class.	CIFAR-10	1	100%	–	100%	–	**60**%	0.19 ± 0.00
		3	95%	0.00 ± 0.00	100%	–	**32**%	0.05 ± 0.00
		5	86%	0.00 ± 0.00	92%	0.00 ± 0.00	**29**%	0.03 ± 0.00
	CIFAR-100	1	56%	0.12 ± 0.00	57%	0.11 ± 0.00	**4**%	0.99 ± 0.01
		3	48%	0.10 ± 0.00	51%	0.10 ± 0.00	**3**%	0.31 ± 0.00
		5	48%	0.10 ± 0.00	46%	0.09 ± 0.00	**2**%	0.19 ± 0.00
	Imagenet	1	99%	0.00 ± 0.00	100%	–	**37**%	0.49 ± 0.00
		3	75%	0.01 ± 0.00	96%	0.00 ± 0.00	**5**%	0.16 ± 0.00
		5	56%	0.01 ± 0.00	82%	0.00 ± 0.00	**2**%	0.10 ± 0.00
Seg.	BDD100K	1	77%	0.03 ± 0.01	69%	0.07 ± 0.01	**12**%	2.03 ± 0.10
		3	31%	0.00 ± 0.00	23%	0.03 ± 0.00	**0**%	0.72 ± 0.03
		5	23%	0.01 ± 0.00	15%	0.03 ± 0.00	**0**%	0.35 ± 0.02
	nuScenes	1	95%	0.00 ± 0.00	95%	0.00 ± 0.00	**52**%	0.16 ± 0.00
		3	71%	0.01 ± 0.00	90%	0.00 ± 0.00	**0**%	0.09 ± 0.00
		5	62%	0.00 ± 0.00	76%	0.00 ± 0.00	**0**%	0.04 ± 0.00
Det.	VOC	1	36%	1.24 ± 0.06	33%	1.12 ± 0.06	**25**%	0.56 ± 0.02
		3	8%	0.88 ± 0.04	6%	0.80 ± 0.04	**0**%	1.10 ± 0.07
		5	6%	0.86 ± 0.04	6%	0.81 ± 0.04	**0**%	0.84 ± 0.04

Table 2: Average cost ratio $\pm$ standard error and failure rate measured over a range of V^* for each T and data set. We fix $c = 1$ and $P = 10^7$ ($P = 10^6$ for VOC and $P = 10^8$ for ImageNet). The best performing failure rate for each setting is bolded. LOC consistently reduces the average failure rate, often down to 0%, while keeping the average cost ratio almost always below 1 (i.e., spending at most $2\times$ the optimal amount).

optimization is indeed necessary when determining how much data to collect. Furthermore, the Ensemble Regression only outperforms the naïve Power Law Regression baseline for BDD100K and VOC. For the other four data sets, the two baselines are either equivalent or Power Law Regression outperforms Ensemble Regression. This shows that estimating the distribution of the minimum data requirement is insufficient for determining how much data to collect. Specifically, estimating this distribution can capture the uncertainty in the naïve Pow Law Regression estimator, but knowing this stochasticity does not yield accurate data collection decisions. This result validates the two-step approach of LOC.

Finally, we remark that our results here for $T = 1$ validate our theoretical analysis in Section 5, which showed that when our estimator of the data requirement distribution produces underestimates, it incurs significantly larger regret than LOC. This can be observed particularly in the failure rates, as every failure will incur a penalty of P. Note that the cost ratios listed in Table 2 do not include P, since the penalty is always significantly greater than the collection cost and including it would obfuscate cost comparisons for instances where V^* was reached. In other words, for CIFAR-10, ImageNet, or nuScenes, the baseline incurs a *true* unfiltered cost ratio (i.e., factoring in the penalty) at approximately P divided by the oracle cost. Due to the high failure rate of the baseline, this unfiltered cost ratio would be several orders of magnitude higher than the values reported in Table 2. Since,

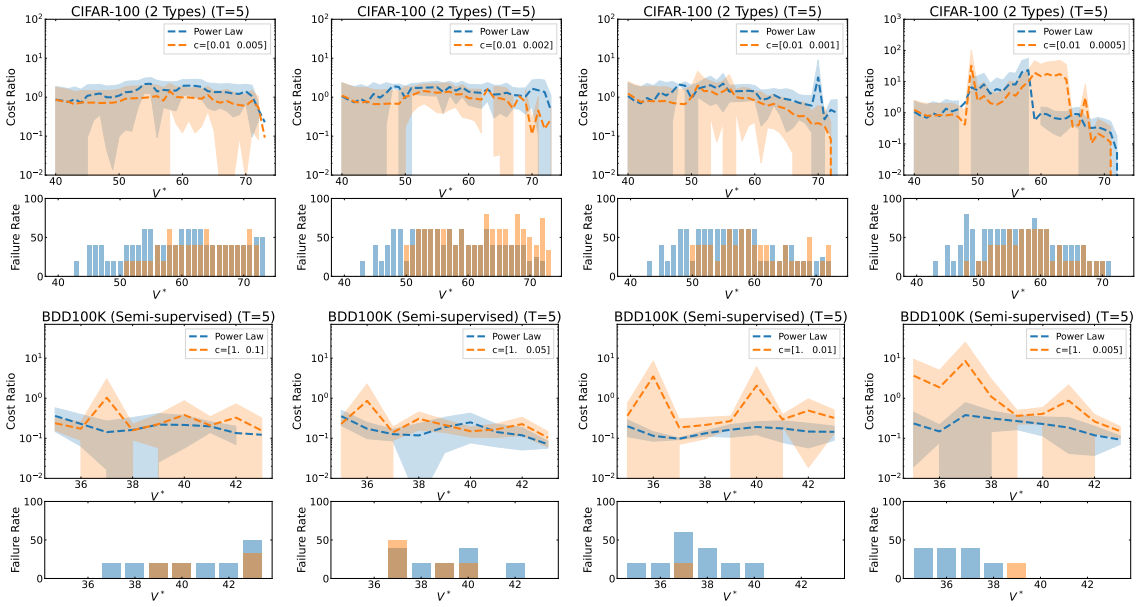

Figure 5: Mean ± standard deviation over 5 seeds of the cost ratio and failure rate for different V^*, after removing 99-th percentile outliers. The columns correspond to scenarios where the first set c_1 costs increasingly more than the second c_2. See Appendix C for the results with $T = 1, 3$.

LOC incurs less than 50% failure rate for all three of these instances, our reported cost ratios more accurately reflect the true values.

6.2.2 Case with $K = 2$

We now discuss experiments on $K = 2$. Figure 5 compares LOC versus regression at $T = 5$ with different costs. LOC decreases the failure rates while keeping a similar cost ratio to the baseline. We include similar plots for $T = 1, 3$ in Appendix C.4. Table 3 aggregates failure rates and cost ratios for all settings, showing LOC consistently achieves lower failure rates for nearly all settings of T. When $T = 5$, LOC also achieves lower cost ratios versus regression on CIFAR-100, meaning that with multiple rounds of collection, we can ensure meeting performance requirements while paying nearly the optimal amount of data. However, the optimization problem is generally more difficult as K increases and we sometimes over-collect data by several orders of magnitude margins. Because these outliers drastically skew the summary average statistics that we measure, we remove the 99-th percentile with respect to total cost for both LOC and the baseline regression estimator. In practice, outlier data collection decisions would naturally be pruned by decision-makers via common-sense reasoning. For instance, if an algorithm suggests to collect 10 million unlabeled images when a human expert would guess at 10 thousand images, the practical decision would not be to blindly use the algorithmic solution.

Although the results in this section compare specifically against power law-based regression, Mahmood et al. (2022b) show that we can use other regression functions (e.g., see Table 1) as well as modifications to reduce the undercollection of these estimation-based

Data set	T	Cost	Power Law Regression		LOC	
			Failure rate	Cost ratio	Failure rate	Cost ratio
CIFAR-100 (2 Types)	1	$(0.01, 0.0005)$	62%	0.84 ± 0.02	**40**%	41.80 ± 1.80
		$(0.01, 0.001)$	58%	1.19 ± 0.01	**46**%	9.85 ± 0.27
		$(0.01, 0.002)$	56%	1.55 ± 0.01	**54**%	6.98 ± 0.18
		$(0.01, 0.005)$	54%	1.65 ± 0.01	**33**%	4.43 ± 0.07
	3	$(0.01, 0.0005)$	43%	3.47 ± 0.16	**30**%	4.88 ± 0.26
		$(0.01, 0.001)$	45%	1.22 ± 0.02	**43**%	1.31 ± 0.03
		$(0.01, 0.002)$	45%	1.47 ± 0.02	**44**%	1.21 ± 0.02
		$(0.01, 0.005)$	38%	1.31 ± 0.01	**36**%	1.17 ± 0.01
	5	$(0.01, 0.0005)$	38%	3.31 ± 0.16	**24**%	5.19 ± 0.22
		$(0.01, 0.001)$	35%	1.22 ± 0.02	**24**%	0.79 ± 0.01
		$(0.01, 0.002)$	**37**%	1.33 ± 0.01	38%	0.90 ± 0.01
		$(0.01, 0.005)$	36%	1.30 ± 0.01	**24**%	0.82 ± 0.00
BDD100K (Semi-supervised)	1	$(1, 0.005)$	86%	0.11 ± 0.01	**44**%	7.02 ± 1.11
		$(1, 0.01)$	79%	0.15 ± 0.01	**30**%	13.47 ± 1.50
		$(1, 0.05)$	72%	0.19 ± 0.01	**49**%	1.02 ± 0.19
		$(1, 0.1)$	70%	0.19 ± 0.01	**65**%	0.40 ± 0.05
	3	$(1, 0.005)$	23%	0.18 ± 0.01	**7**%	1.20 ± 0.12
		$(1, 0.01)$	21%	0.15 ± 0.00	**7**%	2.57 ± 0.58
		$(1, 0.05)$	26%	0.18 ± 0.01	**23**%	0.50 ± 0.06
		$(1, 0.1)$	26%	0.21 ± 0.01	**30**%	0.15 ± 0.01
	5	$(1, 0.005)$	16%	0.22 ± 0.01	**2**%	1.91 ± 0.30
		$(1, 0.01)$	21%	0.15 ± 0.00	**2**%	0.86 ± 0.13
		$(1, 0.05)$	16%	0.17 ± 0.01	**9**%	0.27 ± 0.03
		$(1, 0.1)$	16%	0.20 ± 0.01	**7**%	0.32 ± 0.03

Table 3: Average cost ratio $\pm$ standard error and failure rate over different V^* for each T and $\mathbf{c}$, after removing 99-th percentile outliers. We fix $P = 10^{13}$ for CIFAR-100 and $P = 10^8$ for BDD100K. The best performing failure rate for each setting is bolded. LOC reduces the average failure rate, is more robust to uneven costs than regression, and for $T > 1$, preserves the cost ratio.

baselines. In Appendix C.2, we include further results comparing against these alternate functions. In every setting, we show that while baseline estimators consistently either under- or overestimate how much data to collect, LOC consistently reduces either the cost or the failure rates significantly.

6.3 Robustness to Cost and Penalty Parameters

Figure 6 evaluates the ratio of points collected for $T = 5$ when the cost and the penalty of the optimization problem are varied. We include similar results for $T = 1, 3$ in Appendix C.5. LOC is robust to variations in these parameters, as LOC retains the same shape and scale for almost every parameter setting and data set. Further, LOC consistently remains above the horizontal 1 line, showing that even after varying c and P, we do not fail as frequently as the baseline. Finally, validating Theorem 3, the penalty parameter P provides natural

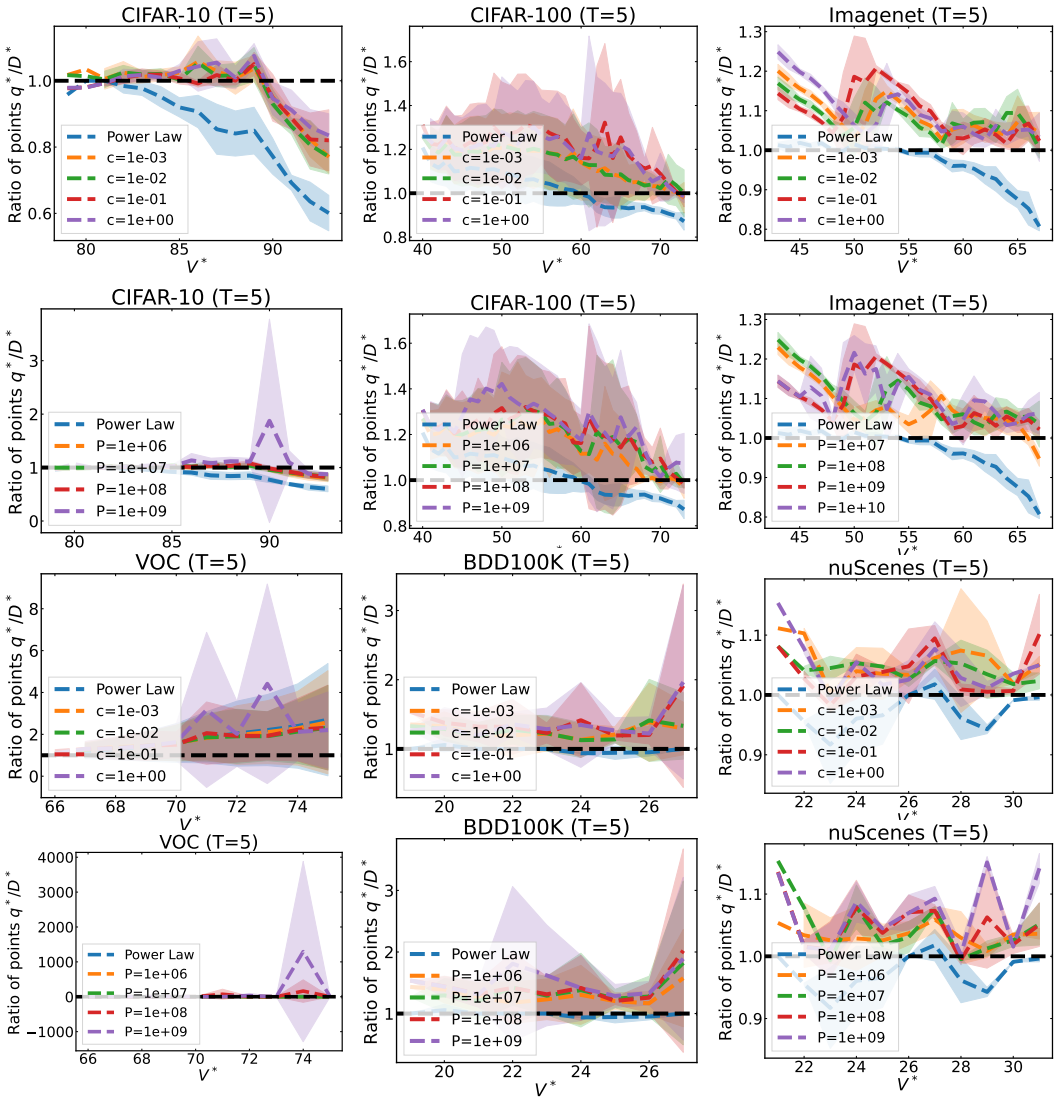

Figure 6: Mean ± standard deviation of 5 seeds of the ratio of data collected q_T^*/D^* for different V^* and fixed $T = 5$. *Rows 1 & 3:* We sweep the cost parameter from 0.001 to 1 and fix $P = 10^7$. *Rows 2 & 4:* We sweep the penalty parameter from 10^6 to 10^9 and fix $c = 1$. The dashed black line corresponds to collecting exactly the minimum data requirement. See Appendix C for all T.

control over the amount of data collected. As we increase P, the ratio of data collected increases consistently.

6.4 Adapting LOC to Custom Modeling Problems

Our optimal data collection framework and LOC can also be adapted to solve custom questions faced by machine learning developers. Here, we present two case studies where we demonstrate how LOC yields high-quality data collection decisions.

T	Power Law Regression		LOC	
	Failure rate	Cost ratio	Failure rate	Cost ratio
1	89%	0.19	**8**%	1.35
3	38%	0.78	**1**%	0.63
5	24%	0.64	**0**%	0.26

Table 4: On CIFAR-100, average cost ratio and failure rate for 'beaver' (new class) measured over a range of V^* for each T, when the model is initialized with only 99 classes and zero training examples for the new class. We fix $c = 1$ and $P = 10^5$. The best performing failure rate for each setting is bolded. LOC consistently achieves less than 10% failure rate, while keeping the average cost ratio even lower than the estimation baseline when $T \geq 3$.

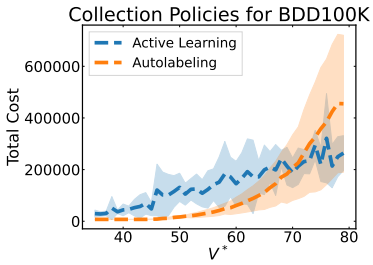

Figure 7: Mean ± standard deviation of 5 seeds when comparing the total collection cost to reach different V^* on BDD100K depending on active learning versus autolabeling. We fix $c_L = 0.995$ and $c_U = 0.005$. Autolabeling is more cost-effective until $V^* \geq 70$ mIoU, at which point the diminishing power law of the unlabeled data make active learning more effective.

6.4.1 Adding New Classes to an Existing Model

Suppose that we have an existing M-class classifier achieving a desired accuracy V^* for each of the current classes individually. We now want to update this model with a new $(M+1)$-th class. We require the classifier to also achieve V^* validation set accuracy for this class as well or else we will pay P. If we can collect training examples for this specific class at a per-sample cost c, how many should we obtain within the time horizon T?

To address this problem, we use LOC where q_t represents the number of training examples of the $(M+1)$-th class. However, $q_0 = 0$ since this is a new class; this means that we cannot fit training statistics to estimate $F(q)$ at time $t = 1$. Instead in the first data collection round, we select one of the existing M classes, model the data requirement distribution for this prior class, and use this as a proxy for $F(q)$. If $T > 1$, then after the first round, we will have an initial amount of data for the $(M+1)$-th class, which we can use to learn $F(q)$ using the standard technique.

We simulate this problem using CIFAR-100, where we set $M = 99$ and first train the model using the training set data of all classes except for 'beaver' (the 100-th class). For the first round estimate, we design $F(q)$ using the statistics of 'dolphin', which is a similar class in the same hierarchy. Table 4 provides the failure rate and cost ratio for both LOC and the power law regression baseline. Here, LOC consistently achieves less than 10% failure rate and moreover, outperforms the baseline on both the failure rate and cost ratio for both $T = 3, 5$. Note that since we start with $q_0 = 0$ initial data, our first estimation of $F(q)$ will be poor for both our method and the baseline; this leads the baseline to high failure rates for $T = 1$ and surprisingly high costs for $T = 3, 5$. Instead, LOC, by virtue of optimization, yields low failure rates for $T = 1$ and strong performance at larger T.

6.4.2 Deciding Between Two Data Annotation Campaigns

Suppose that we have collected a large pool of unlabeled data. To annotate each data point, there is a small cost c_C for collecting and pre-processing an unlabeled image and a labeling cost $c_L > c_C$ for manually labeling the image. Using an initial labeled set of q_0 points, we must decide between two potential annotation strategies via $T = 1$ round forecasts:

1. **Autolabeling:** Rather than manually labeling every data point, we can use our current model to synthetically generate labels for the unlabeled data. These autolabeled points may not as effective as the manually labeled points, but autolabeling incurs zero additional labeling costs. Consider a strategy where we manually label a fraction of the data that we have collected (i.e., incurring the cost of collecting and labeling the data $c_C + c_L$) and autolabel the rest of the data (i.e., incurring only the collection cost c_C per data point).

   Let $\mathbf{q} = (q_L, q_C)$ where q_L is the amount of manually labeled data and q_U is the amount of unlabeled data that we autolabel. Then, our total cost under this strategy is

   $$(c_L + c_C)(q_L - q_0) + c_C q_C + P\mathbb{1}\{V_{\mathbf{q}} \geq V^*\}.$$

2. **Active learning:** Since the quality of autolabeled data may not be as high as that of manually labeled data, we may instead opt to only manually label data. Moreover, we can re-direct the engineering effort of building an autolabeler towards designing an active learning framework, which mines the collected (unlabeled) data to find the most useful training examples.

   With active learning, $\mathbf{q} = q_L$ is the amount of data that is manually labeled. Furthermore, the cost of labeling each data point is equal to the cost of collection plus labeling $c_L + c_C$. The optimization objective under this strategy is

   $$(c_L + c_C)(q_L - q_0) + P\mathbb{1}\{V_{q_L} \geq V^*\}$$

Given a target performance V^*, we can solve both optimization problems to determine which of the two policies is more cost-efficient.

We evaluate this experiment using the labeled and unlabeled splits of BDD100K. The autolabeling setting is equivalent to the experiments performed in Section 6 for $K = 2$. The active learning setting is a special case of the $K = 1$ experiment where the data collected in each round and subsampled for fitting scaling laws, are collected according to an active learning sampler rather than random sampling. We follow the approach in Mahmood et al. (2022b) where the steps of Algorithm 3 that involve estimating the minimum data requirement distribution are unchanged other than the fact that the data is collected via active learning (see Appendix C.3 for details and additional active learning experiments). For data collection, we use a simple strategy of computing the average Confidence scores of estimates per scene and selecting the Least Confident scenes; we leave further details in Appendix B.2 (Settles, 2009). Figure 7 plots the total cost as determined from an initial $q_0 = 7,000$ labeled points and costs $c_L = 0.995$ and $c_C = 0.005$; these costs replicate the settings explored in Table 3. We extrapolate the performance and costs up to $V^* = 80$. Since in this scenario, autolabeling data is $20\times$ cheaper than manually labeling it, our

analysis suggests that for moderate target scores (e.g., $V^* = 58$ mIoU), we can save up to $100,000$ in total costs. However, in general, active learning produces more high-quality data than autolabeling, so we expect the 2-dimensional power law for autolabeling to flatten sooner than the active learning estimator. For $V^* \geq 70$, the minimum data requirement and consequently, the total cost of autolabeling grows exponentially and exceeds the total cost of active learning. This leads us to an intuitive conclusion: in applications requiring high performance models, at some point it becomes more important to use high-quality data rather than simply lots of data.

7. Discussion

Decisions on how much data to collect to improve a given machine learning model are fundamental in all machine learning applications, but such methods are typically handled by estimating a scaling law of training data set size to model performance and then extrapolating future behavior. This naïve extrapolation tends to lead to costly decisions from overcollecting data or delays and future costs from undercollecting data. In our paper, we develop a rigorous framework for optimizing data collection workflows by introducing an optimal data collection problem that captures the uncertainty in estimating data requirements. Our general framework can model a variety of settings such as where multiple data sources incur different collection costs, where an existing model must be upgraded with data for a new class, or where we must assess the benefits of different sampling and labeling practices. We numerically validate our solution algorithm, Learn-Optimize-Collect, on six computer vision data sets covering classification, segmentation, and detection tasks to show that we consistently meet pre-determined performance metrics regardless of costs and time horizons.

Our optimization model minimizes the expected total future collection cost, which is modeled by two parameters: (i) a per-sample cost $\mathbf{c}$ that models the cost of collecting, cleaning, and annotating data; and (ii) a penalty P that models the opportunity cost of the model failing to meet our desired performance metric. The second parameter may not be readily available in practice. However, we show empirically that LOC is typically robust to parameter variations on one to three orders of magnitude. That is, as long as we can roughly estimate these parameters for our setting, we will still be able to make good data collection decisions. Moreover, we theoretically analyze the one-round $T = 1$ setting to draw two high-level insights. First, our problem is equivalent to specifying a minimum tolerance $\epsilon \leq \Pr\{V_q < V^*\}$ on the probability that we do not collect enough data and simply collecting the quantile $\hat{F}^{-1}(1-\epsilon)$ of the distribution of the minimum amount of data needed; consequently, practitioners can instantly obtain one-round estimates for how much data to collect. This analysis can be further specialized when the distribution has a given structure (e.g., Gaussian). Second, we prove that as long as the estimated distribution $\hat{F}$ is sufficiently close to the true distribution of the minimum data requirement, our optimization model provably improves upon estimation-only strategies in terms of minimizing the total cost.

LOC combines neural scaling law estimators with a stochastic optimization problem that can be solved via gradient descent. Our fundamental step is to treat the minimum amount of data we would need as a random variable and bootstrap a neural scaling law estimator to estimate its probability distribution. As future advances in neural scaling laws arrive, our bootstrapping can be deployed on top of new scaling law estimation algorithms to

optimize data collection decisions better than any estimation-only techniques. In particular, estimating neural scaling laws for arbitrary multi-dimensional settings (e.g., when combining different data sources) remains a difficult problem, but a side-product of this paper presents a simple multi-dimensional additive power law estimator that works reasonably well in our experiments.

Our framework is designed specifically for settings where the machine learning model and training algorithm are fixed, that the data is generally of high-quality, and that there is no misspecification between model and data. In practice, these assumptions may not always be satisfied. For example, designers may modify the model or the data source in between collection rounds, change the evaluation metric, or seek to address alternative targets. Mathematically, our framework assumes that the desired performance target is achievable with a finite amount of data and that the true neural scaling law of a given problem instance is monotonically non-decreasing with the dataset size (Viering and Loog, 2022). The first assumption is necessary for this problem to be solvable. Although the second assumption is not always satisfied in machine learning tasks, it has been empirically observed to hold in most practical deep learning applications and is a standard assumption when scaling deep learning models (Hestness et al., 2017; Hoffmann et al., 2022). Finally, our numerical experiments rely on simulations with pre-constructed ground truth learning curves $v(n)$. An alternative experimental setup may be to explicitly sample points based on the decisions generated by LOC, train the neural network model, and evaluate its score. However, such repeated retraining is computationally too expensive to perform for the range of experiments explored in this paper. Furthermore, the quality of our simulation is proportional to the number of data points that we sample, meaning that we can maintain accurate experimental analysis.

Improving data collection practices yields potentially positive and negative societal impacts. By reducing the collection of extraneous data, we implicitly reduce the environmental costs of training models. On the other hand, equitable data collection should also be considered in real-world data collection practices that involve humans. We envision a potential future work to incorporate privacy and fairness constraints to prevent over- or under-sampling of protected groups. Finally, our method is guided by a score function on a held-out validation set. Biases in this set may be exacerbated when optimizing data collection to meet target performance.

Finally, we emphasize that this work addresses a longstanding problem in machine learning practice. There is a folklore observation that over 80% of industry machine learning projects fail to reach production, often due to insufficient, noisy, or inappropriate data (VentureBeat, 2019). As mentioned previously, industry surveys have reported that 51% of practitioners face delays from under-collection (Dimensional Research, 2019). Our numerical experiments verify this observation by showing that naïvely estimating power laws typically leads to undercollection. We believe that robust data collection policies obtained via LOC can reduce failures while further guiding practitioners on how to manage both costs and time.

Acknowledgments

The authors thank the Action Editor and all referees for their valuable comments, which have significantly improved this work. The authors also thank Daiqing Li, Jonah Philion, and Zhiding Yu for valuable feedback and help on earlier versions of this work.

Appendices

Appendix A. An Alternative Partially Observable Markov Decision Process Formulation of the Optimal Data Collection Problem

Our problem defined in Section 3 can be written as a Partially Observable Markov Decision Process (POMDP) (Puterman, 2014; Bertsekas, 2012), modeled by the tuple $(\Theta, \mathcal{A}, \mathcal{S}, p, r_t)$. Here, the state space characterizes the data requirement $\mathbf{D}^* \in \Theta := \mathbb{R}_+^K$, the action space characterizes the additional data collected $\mathbf{d}_t := (\mathbf{q}_t - \mathbf{q}_{t-1}) \in \mathcal{A} := \mathbb{R}_+^K$, and the observation set $\mathcal{S} := \{0, 1\}$ characterizes a binary variable $\mathbb{1}\{V(\mathcal{D}_{\mathbf{q}_t}) \geq V^*\}$. Furthermore, $p(\cdot|\mathbf{D}^*, \mathbf{d}_t)$ is the observation transition probability and $r_t(\cdot)$ is the reward function where $r_t(\mathbf{q}_t, \mathbf{q}_{t-1}, \mathbf{D}^*) := -c(\mathbf{q}_t - \mathbf{q}_{t-1})$ for $t \leq T$ and $r_{T+1}(\mathbf{q}_t, \mathbf{q}_{t-1}, \mathbf{D}^*) := -P\mathbb{1}\{\mathbf{q}_T < \mathbf{D}^*\}$. Finally, note that the state variable is constant throughout the MDP, meaning this problem can be written as an EK 'Learning-and-Doing' model (Easley and Kiefer, 1988).

POMDPs are typically solved by using a belief distribution of the state variable to average the reward in the value function. In general, these methods are susceptible to a curse of dimensionality and can sometimes be only tackled via approximations (Zhao et al., 2021). Alternatively, we may consider applying reinforcement learning. However, note that real-world data collection tasks do not contain the requisite sizes of learning data or generalizable simulation mechanisms that are staples in reinforcement learning techniques. All of these challenges motivate our approach, which has the benefit of being an easy-to-solve optimization problem on top of existing neural scaling law methods.

Appendix B. Simulation Experiment Setup

The most intuitive approach of validating our data collection problem is by repeatedly sampling from a data set, training a model, and solving the optimization problem. However, since performing a large set of such experiments over many data sets becomes computationally intractable, we follow the approach of Mahmood et al. (2022b), who propose a simulation model of the data collection problem. Below, we summarize the simulation setup.

The simulation replicates the steps in Algorithm 3 except with one key difference. In the simulation, we replace the score function $V(\mathcal{D})$ with a *ground truth* function $v_{\text{gt}}(\mathbf{q})$ that serves as an oracle which reports the expected score of the model trained with q data points. Thus, rather than having to collect data and train a model in each round, we evaluate $v_{\text{gt}}(\mathbf{q}_t)$ and treat this as the current model score. The optimization and regression models do not have access to $v_{\text{gt}}(\mathbf{q})$.

B.1 A Piecewise-Linear Ground Truth Approximation

In order to build a ground truth function, we first use the sub-sampling procedure in Algorithm 3 to collect performance statistics over subsets of the entire training data set.

Data set	Task	Score	Full data set size	
CIFAR-10 (Krizhevsky, 2009)	Classification	Accuracy	50,000	
CIFAR-100 (Krizhevsky, 2009)	Classification	Accuracy	50,000	
ImageNet (Deng et al., 2009)	Classification	Accuracy	1,281,167	
BDD100K (Yu et al., 2020)	Semantic Segmentation	Mean IoU	7,000	
nuScenes (Caesar et al., 2020)	BEV Segmentation	Mean IoU	28,130	
VOC (Everingham et al., 2007, 2012)	2-D Object Detection	Mean AP	16,551	
CIFAR-100 (Krizhevsky, 2009)	Classification	Accuracy	25,000 (Classes 0-49)	25,000 (Classes 50-99)
BDD100K (Yu et al., 2020)	Semantic Segmentation	Mean IoU	7,000 (Labeled)	70,000 (Unlabeled)

Table 5: Data sets, tasks, and score functions considered.

Using these observed statistics, we then build a piecewise-linear model of the ground truth. Below, we first highlight how to construct a piecewise-linear model when given a set of data set sizes and their corresponding scores. In the next subsection, we will detail the exact data collection process.

The Single-variate ($K = 1$) Case. Mahmood et al. (2022b) develop a ground truth function by collecting training statistics over subsets of the entire data set and performing a linear interpolation. Let $q_0 \leq q_1 \leq q_2 \leq \cdots$ be a series of data set sizes and let $\mathcal{D}_{q_0} \subset \mathcal{D}_{q_1} \subset \mathcal{D}_{q_2} \subset \cdots$ be their corresponding sets. Then, the piecewise-linear function:

$$v_{\text{gt}}(q) := \begin{cases} \dfrac{V(\mathcal{D}_{q_0})}{q_0} n, & q \leq q_0 \\ \dfrac{V(\mathcal{D}_{q_t}) - V(\mathcal{D}_{q_{t-1}})}{q_t - q_{t-1}} (q - q_t) + V(\mathcal{D}_{q_{t-1}}), & q_{t-1} \leq q \leq q_t \end{cases}$$

is concave and monotonically increasing, which follows the general trend of real learning curves Hestness et al. (2017). Furthermore Mahmood et al. (2022b) show that given sufficient resolution, i.e., enough data subsets, this piecewise linear function is an accurate approximation of the true learning curve $V(\mathcal{D})$.

The Multi-variate ($K = 2$) Case. In the previous $K = 1$ case, the ground truth was formed by taking linear interpolations between different subset sizes. When $K > 1$, we have multiple subsets that are used to evaluate the score $V(\mathcal{D}^1, \ldots, \mathcal{D}^K)$.

In our numerical experiments, we focus on $K = 2$. Here, we can generalize the linear interpolation process to a bilinear interpolation (Wang and Yang, 2008). Let $q_0^1 \leq q_1^1 \leq q_2^1 \leq \cdots$ and $q_0^2 \leq q_1^2 \leq q_2^2 \leq \cdots$ be two series of data set sizes, and consider the grid

$$\begin{matrix} (q_0^1, q_0^2) & (q_1^1, q_0^2) & (q_2^1, q_0^2) & \cdots \\ (q_0^1, q_1^2) & (q_1^1, q_1^2) & (q_2^1, q_1^2) & \cdots \\ (q_0^1, q_2^2) & (q_1^1, q_2^2) & (q_2^1, q_2^2) & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{matrix}$$

We estimate $v(q^1, q^2)$ for any $q^1, q^2 \in [q_{s-1}^1, q_s^1] \times [q_{t-1}^2, q_t^2]$ via bilinear interpolation over the square defined by these borders (Wang and Yang, 2008).

For $K > 2$. The piecewise linear approximations grow increasingly complex as the dimension K increases. Furthermore, the number of subsets of data set sizes required to create a piecewise linear approximation increases exponentially with K. Specifically for $k \in \{1, \ldots, K\}$, let M_k denote the number of subsets (i.e., $|\{q_0^k, q_1^k, \ldots, q_{M_k}^k\}|$) of a data set that

we consider when creating subsets. For each combination of K subsets, we must then train a model and evaluate it's performance to record $V(\mathcal{D}^1, \ldots, \mathcal{D}^K)$. Thus, we must subsample and train our model for $O(\prod_k M_k)$ combinations, which quickly becomes computationally prohibitive.

B.2 Data Collection

We now summarize the data collection and training process used to create the above piecewise-linear functions for each data set and task. All models were implemented using PyTorch and trained on machines with up to eight NVIDIA V100 GPU cards. Table 5 details each task and data set size.

Image Classification Tasks. For all experiments with CIFAR-10 and CIFAR-100, we use a ResNet18 He et al. (2016) following the same procedure as in Coleman et al. (2020). For ImageNet, we use a ResNet34 He et al. (2016) using the procedure in Coleman et al. (2020). All models are trained with cross entropy loss using SGD with momentum. We evaluate all models on Top-1 Accuracy.

For all experiments, we set the initial data set at $q_0 = 10\%$ of the data. In data collection, we create five subsets containing $2\%, 4\%, \cdots, 10\%$ of the training data, five subsets containing $12\%, 14\%, \cdots, 20\%$ of the training data, and eight subsets containing $30\%, 40\%, \cdots, 100\%$ of the data. Each subset is contained in the following subsets. Note that we use higher granularity in the early stage as this is where the dynamics of the learning curve vary the most. With more data, the learning curve eventually has a nearly zero slope. For each subset, we train our respective model and evaluate performance.

VOC. We use the Single-Shot Detector 300 (SSD300) Liu et al. (2016) based on a VGG16 backbone Simonyan and Zisserman (2015), following the same procedure as in Elezi et al. (2022). We evaluate all models on mean AP.

For all experiments, we set the initial data set at $q_0 = 10\%$ of the data. In data collection, we sample twenty subsets at 5% intervals, i.e., $5\%, 10\%, 15\%, \cdots, 100\%$ of the training data.

BDD100K. We use Deeplabv3 Chen et al. (2018) with ResNet50 backbone. We use random initialization for the backbone. We use the original data set split from Yu et al. (2020) with $7,000$ and $1,000$ data points in the train and validation sets respectively. The evaluation metrics is mean Intersection over Union (IoU). We follow the same protocol used in the Image classification tasks to create our subsets of data.

Active Learning. We perform experiments involving active learning on CIFAR-100 and BDD100K. For these experiments, rather than collecting data by i.i.d. sampling of subsets at the above respective intervals, we use an active learning algorithm to select the subsets at each interval; for example, with CIFAR-100, we start with a random i.i.d. sample of 2% of the data and for each subsequent $4\%, 6\%, 8\%, \cdots, 20\%, 30\%, 40\%, \cdots, 100\%$, we select this data with an active learning algorithm. Each subset is contained in the following subsets. For CIFAR-100, we explore active learning via Maximum Entropy (Settles, 2009), Least Confidence (Settles, 2009), and Greedy k-Centers (Sener and Savarese, 2018). For BDD100K, we use only Least Confidence (Settles, 2009). We use the same models and training practice as described previously.

nuScenes. We use the "Lift Splat" architecture Philion and Fidler (2020), which is used for BEV segmentation from driving scenes, following the steps from the original paper to

Parameter	Setting
Optimizer	GD with Momentum ($\beta = 0.9$), Adam ($\beta_0, \beta_1 = 0.9, 0.999$)
Learning rate	$0.005, \ldots, 500$
Number of bootstrap samples B	500
Number of regression subsets R	See Appendix B.2
Density Estimation Model	KDE for $K = 1$, GMM for $K = 2$
KDE Bandwidth	$20000, \ldots, 20000000$ for ImageNet $200, \ldots, 4000$ for all others
GMM number of clusters	$4, \ldots, 10$

Table 6: Summary of hyperparameters used in our experiments.

train this model. We evaluate on mean IoU. Our data collection procedure follows the same steps as used for BDD100K and the Image classification tasks.

CIFAR-100 (2 Types). We partition this data set into two subsets $\mathcal{D}^1$ and $\mathcal{D}^2$ of $25,000$ images each containing the first 50 and last 50 classes, respectively. We then train a ResNet18 (He et al., 2016) using different fractions of the two subsets. We follow the same training procedure as in the single-variate case except with one difference. Since some of the data sets will naturally be imbalanced (e.g., if we train with half of the first subset and all of the second subset), we employ a class-balanced cross entropy loss using the inverse frequency of samples per class.

For each $\mathcal{D}^k$ subsets, respectively, we follow the same subsampling procedure used in the single-variate case. That is, we let $q_0^1 = 10\%$ of the first data subset and $q_0^2 = 10\%$ of the second data subset. For each subset, we create 10 subsampled sets at intervals of $2\%, 4\%, 6\%, \cdots, 20\%$ of the respective data subset. We then create eight further subsampled sets at $30\%, 40\%, \cdots, 100\%$ of the respective data subset. Finally, we train our model and evaluate the score on every combination of the subsampled subsets of $\mathcal{D}^1 \times \mathcal{D}^2$.

BDD100K (Semi-supervised). For this task, we consider semi-supervised segmentation via pseudo-labeling the unlabeled data set in BDD100K. The data is partitioned into two subsets $\mathcal{D}^1$ and $\mathcal{D}^2$ containing $7,000$ labeled and $70,000$ unlabeled scenes. We use Deeplabv3 (Chen et al., 2018) with a ResNet50 backbone. Here however, we:

1. First train with a labeled subset of $\mathcal{D}^1$ via supervised learning.
2. Pseudo-label an unlabeled subset of $\mathcal{D}^2$ using the trained model.
3. Re-train the segmentation model with the labeled subset and the pseudo-labeled subset.

We follow the same procedure as in the single-variate case for both training steps, except we weigh the unlabeled data by 0.2 to reduce its contribution to the loss.

Training via semi-supervised learning on BDD100K requires long compute times, so we reduce the number of subsets used in this experiment. For the labeled set $\mathcal{D}^1$, we create subsets with $5\%, 10\%, 15\%, 20\%, 40\%, 60\%, 80\%, 100\%$ of the data. For the unlabeled set $\mathcal{D}^2$, we create subsets with $0\%, 10\%, 25\%, 50\%, 100\%$ of the data. Note that we have five settings of unlabeled data since we include the case of training with no unlabeled data as well.

B.3 LOC Implementation

For all experiments, we initialize with 10% of the training data set. We consider $T = 1, 3, 5$ rounds and sweep a range of V^*. We provide a summary of parameters in Table 6.

For the experiments with $K = 1$, we model the data requirement PDF $f(q)$ in each round of the problem as follows. We first draw $B = 500$ bootstrap resamples of the current training statistics $\mathcal{R}$, where $\mathcal{R} = \{(rq_0/R, V(\mathcal{D}_{rq_0/R}))\}_{r=1}^{R} \cup \{(q_s, V(\mathcal{D}_{q_s}))\}_{s=1}^{t}$ contains all of the measured statistics up to the initial data set (e.g., for CIFAR-10, this includes performance with $2\%, 4\%, \cdots, 10\%$ of the data), and the previous collected data. The latter is obtained by calling our piecewise-linear ground truth approximation. For each bootstrap resample, we fit a power regression model $\hat{v}(q; \boldsymbol{\theta}) = \theta_0 q^{\theta_1} + \theta_2$ and solve for the estimated minimum data requirement by minimizing a least squares problem using the Trust Region Reflective algorithm, with initial values set to $1, 0, 0$ and bounds set to $[0, 100], [0, 0.999], [-10, 10]$ for $\theta_0, \theta_1, \theta_2$, respectively. After fitting $\hat{v}(q; \boldsymbol{\theta})$ for each resample, we then compute the corresponding estimate of D^*. We then use our set of estimates to fit a Kernel Density Estimation (KDE) model after gridsearching for the best bandwidth parameter. For the ImageNet data set, we grid search for the best bandwidth parameter between $[20000, 40000, 100000, 200000, 400000, 1000000, 2000000, 4000000, 10000000, 20000000]$, whereas for all other data sets, we grid search for the best bandwidth parameter between $[200, 400, 1000, 2000, 4000]$. We use these different settings because the scale of data on ImageNet is between two to three orders of magnitude greater than the scale of data for the other data sets that we consider. Thus, to ensure that the estimated distribution can reasonably capture the variation on larger scales, we find that larger bandwidths are necessary. We note that in practice, a reasonable choice of bandwidth ranges for gridsearching can be visually determined by plotting the histogram of the estimated data requirement distribution (e.g., see Figure 9 in Appendix C.1) for different choices of bandwidth and inspecting the fitted distributions.

For the experiments with $K = 2$, we use the same above procedure but fit Gaussian Mixture Models (GMM) due to their having an easily computable CDF via the Gaussian $\text{erf}(\cdot)$, rather than numerically integrating the PDF. We grid-search over the number of mixture components for the GMM model.

We optimize over problems (8) using gradient descent. Depending on the current state and data set, different hyperparameters perform better. As a result, we perform extensive hyperparameter tuning every time we need to solve the optimization problem. Here, we sweep all combinations of gradient descent with momentum ($\beta = 0.9$), and Adam ($\beta_0 = 0.9, \beta_1 = 0.999$), and learning rates between $[0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10, 50, 100, 500]$.

We initialize each problem with $\mathbf{q}_t$ equal to the baseline regression solution and $\mathbf{q}_{t+s} = \mathbf{q}_t/(s+1)$ for all $1 \leq s \leq T - t$. That is, we set the initial value for future collection amounts to be fractions of the initial value of the immediate amount of data to collect. We identified this initialization by manually inspecting the solutions found by LOC; it improves the conditioning of the loss landscape relative to other random initialization schemes.

Appendix C. Additional Numerical Results

This section contains expanded results of our numerical experiments and further ablations. Our key results include:

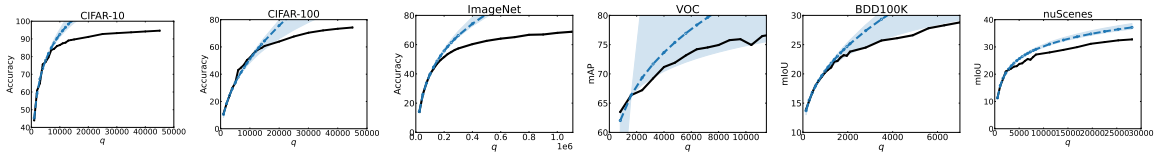

Figure 8: For a fixed seed, ground truth learning curves (black) and the estimated power law learning curves (blue) obtained via bootstrapping and ensembling. The shaded region represents the 95 percentile of the ensemble and the dashed blue line represents the mean of the regression functions. The mean is consistently higher than the unknown ground truth, whereas the shaded region can at times cover it.

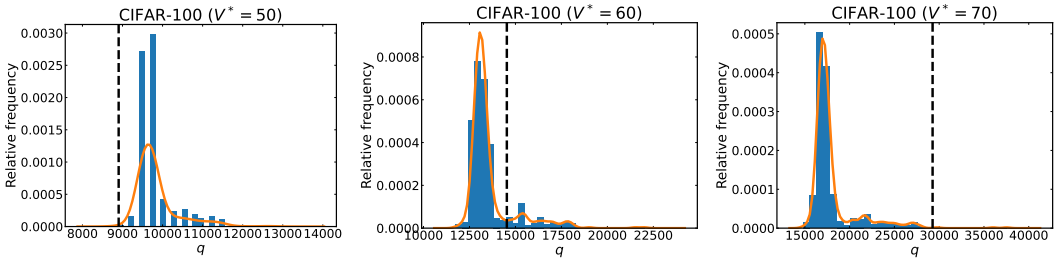

Figure 9: For a fixed seed, the histogram of estimates of D^* from different bootstrapped models (blue bars), the estimated $F(q)$ (orange curve), and the ground truth D^* (black dashed line). Each plot corresponds to a different V^* for CIFAR-100 (see Figure 8 for the learning curve). With higher targets, regression (i.e., collecting the mean of the distribution) will lead to larger under-estimations.

- In Appendix C.1, we evaluate the effectiveness of estimating $F(q)$ by plotting the estimated learning curves as well as the empirical histograms used to model the data requirement distribution.
- In Appendix C.2, we consider variants of LOC where we use different regression functions to estimate the data requirement distribution. Our optimization framework can be deployed on top of any regression function to reduce the failure rate.
- In Appendix C.3, we consider LOC for scenarios where data is not collected via random sampling, but instead by active learning. Our optimization framework consistently outperforms baselines even under such non-i.i.d. settings.
- In Appendix C.4, we explore the multi-variate LOC (i.e., $K = 2$) for problems where we have a small number of $T = 1, 3$ rounds. The baseline fails for almost all instances of $T = 1$, whereas LOC maintains a low failure rate.
- In Appendix C.5, we explore the sensitivity of our optimization algorithm to variations in the cost and penalty parameters. In all except one instance, LOC consistently maintains a low total cost and failure rate.

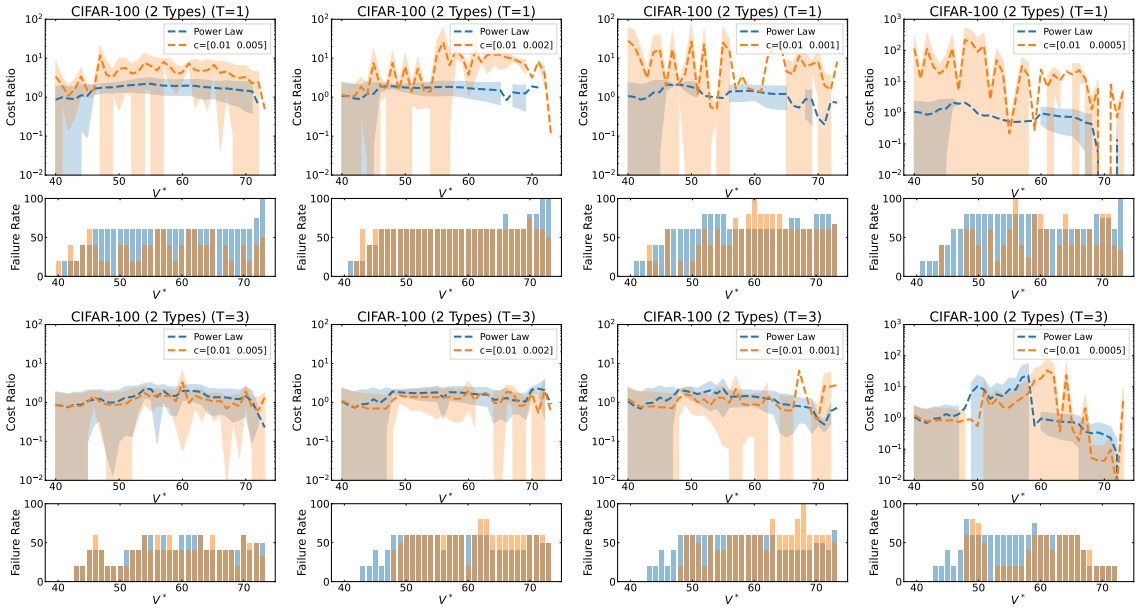

Figure 10: For experiments on CIFAR-100 with two data types, mean ± standard deviation over 5 seeds of the cost ratio $\mathbf{c}^\mathsf{T}(\mathbf{q}_T^* - \mathbf{q}_0)/\mathbf{c}^\mathsf{T}(\mathbf{D}^* - \mathbf{q}_0) - 1$ and failure rate for different V after removing 99-th percentile outliers. We fix $c_0 = 1$ and $P = 10^{13}$. The rows correspond to $T = 1, 3$ (see the main paper for $T = 5$) and the columns correspond to $c_1 = c_0/2, c_0/5, c_0/10, c_0/20$.

C.1 Estimating the Data Requirement Distribution $F(q)$

To estimate $F(q)$, we first create an ensemble of estimated learning curves, which we then invert to obtain an empirical distribution of estimated values for D^*. Figure 8 plots our bootstrap resampled estimated learning curves versus the ground truth performance for the first round of data collection when we have access to an initial $\mathcal{D}_{q_0}$ containing 10% of the full data set. As noted in Mahmood et al. (2022b), the mean estimated learning curve diverges from the ground truth. However, by bootstrap resampling an ensemble of learning curves, we can cover the ground truth with some probability.

Figure 9 plots the empirical histograms of estimated D^* as well as the estimated $F(q)$ obtained via KDE on CIFAR-10 with three different values for V^*. Although the mode of the estimated distribution is far from the ground truth D^*, the estimated distribution assigns some probability to the ground truth region. LOC optimizes over this estimated $F(q)$, which allows us to conservatively collect data and reduce the chances of failure.

C.2 LOC with Alternative Regression Functions

Mahmood et al. (2022b) show that we can use other regression functions instead of the power law to estimate the data requirement. Moreover, some functions tend to consistently over- or under-estimate the requirement. LOC can be deployed on top of any such regression function, since the regression function is only used to generate bootstrap samples. In this

	Data set	T	Regression With Logarithmic $\hat{v}(q)$		LOC		Regression With Arctan $\hat{v}(q)$		LOC		Regression With Algebraic Root $\hat{v}(q)$		LOC	
			Failure rate (FR)	Cost ratio (CR)	FR	CR	FR	CR	FR	CR	FR	CR	FR	CR
Class.	CIFAR-10	1	3%	0.59	**0**%	2.18	88%	0.01	**6**%	1.93	100%	–	**12**%	1.74
		3	3%	0.59	**0**%	1.16	57%	0.01	**0**%	0.91	97%	0	**0**%	0.87
		5	3%	0.59	**0**%	0.9	43%	0.01	**0**%	0.62	88%	0	**0**%	0.70
	CIFAR-100	1	43%	0.19	**2**%	1.17	23%	3.31	**0**%	5.56	52%	0.11	**23**%	0.81
		3	37%	0.17	**2**%	0.54	15%	3.01	**0**%	3.92	44%	0.09	**2**%	0.87
		5	34%	0.16	**2**%	0.39	12%	2.90	**0**%	3.6	44%	0.09	**2**%	0.54
	Imagenet	1	**0**%	24.5	34%	0.48	100%	–	**0**%	2.65	100%	–	**40**%	0.48
		3	**0**%	24.5	2%	0.17	98%	0.01	**0**%	0.81	94%	0.01	**0**%	0.19
		5	**0**%	24.5	**0**%	0.1	89%	0.01	**0**%	0.93	79%	0.01	**0**%	0.10
Seg.	BDD100K	1	**0**%	0.55	12%	4.78	85%	0.04	**12**%	9.51	73%	0.08	**12**%	4.82
		3	**0**%	0.55	**0**%	1.78	62%	0.02	**0**%	5.33	27%	0.04	**0**%	1.48
		5	**0**%	0.55	**0**%	0.72	54%	0.01	**0**%	2.77	15%	0.03	**0**%	0.79
	nuScenes	1	**0**%	0.52	**0**%	2.78	95%	0	**0**%	8.96	95%	0	**0**%	3.42
		3	**0**%	0.52	**0**%	0.78	95%	0	**0**%	6.63	81%	0.02	**0**%	1.73
		5	**0**%	0.52	**0**%	0.5	95%	0	**0**%	7.55	67%	0.01	**0**%	2.25
Det.	VOC	1	100%	–	**6**%	1.85	83%	0.11	**14**%	0.49	**14**%	11.2	**14**%	13.5
		3	100%	–	**0**%	1.60	75%	0.07	**0**%	1.00	**0**%	9.61	**0**%	12.1
		5	94%	0	**0**%	1.37	67%	0.06	**0**%	1.32	**0**%	9.61	**0**%	18.4

Table 7: Comparing against the following alternate estimators of Mahmood et al. (2022b) with the same setup as in Table 2: Logarithmic $\hat{v}(q;\boldsymbol{\theta}) = \theta_0 \log(q + \theta_1) + \theta_2$, Arctan $\hat{v}(q;\boldsymbol{\theta}) = \frac{200}{\pi} \arctan(\theta_0 \frac{\pi}{2} q + \theta_1) + \theta_2$. Algebraic Root $\hat{v}(q;\boldsymbol{\theta}) = \frac{100q}{1+|\theta_0 q|^{\theta_1})^{1/\theta_1}} + \theta_2$. The best performing cost ratio is underlined and the best performing failure rate for each setting is bolded.

	Data set	T	Regression With Correction (Mahmood et al., 2022b)		LOC	
			Failure rate	Cost ratio	Failure rate	Cost ratio
Class.	CIFAR-100	1	14%	0.94	**4**%	0.99
		3	**1**%	0.23	3%	0.31
		5	**0**%	0.17	2%	0.19
	Imagenet	1	**7**%	1.03	37%	0.49
		3	**0**%	0.21	5%	0.16
		5	**0**%	0.14	2%	0.10
Seg.	BDD100K	1	**4**%	4.03	12%	2.03
		3	**0**%	1.02	**0**%	0.72
		5	**0**%	0.62	**0**%	0.35
	nuScenes	1	**0**%	27.2	52%	0.16
		3	**0**%	0.75	**0**%	0.09
		5	**0**%	0.30	**0**%	0.04
Det.	VOC	1	**0**%	44.6	25%	0.56
		3	**0**%	7.02	**0**%	1.10
		5	**0**%	3.98	**0**%	0.84

Table 8: Comparing against the correction factor-based Power Law Regression of Mahmood et al. (2022b) with the same setup as in Table 2. The best performing cost ratio is underlined and the best performing failure rate for each setting is bolded. Although the baseline achieves low failure rates, LOC often can achieve competitive failure rates while reducing the cost ratios by an order of magnitude.

Strategy	T	Regression		LOC	
		Failure rate	Cost ratio	Failure rate	Cost ratio
Entropy	1	23%	0.53	**1**%	1.79
	3	18%	0.50	**3**%	1.02
	5	18%	0.50	**2**%	0.83
k-Centers	1	51%	0.16	**26**%	0.38
	3	44%	0.14	**13**%	0.17
	5	44%	0.14	**8**%	0.13
Least Confidence	1	24%	0.70	**14**%	3.36
	3	23%	0.68	**5**%	1.71
	5	21%	0.67	**6**%	1.44

Table 9: Different active learning policies for experiments on CIFAR-100, measuring the average cost ratio and failure rate measured over a range of V^* and T. We fix $c = 1$ and $P = 10^7$. The best performing failure rate for each setting is bolded. The cost ratio is measured only for instances that achieve V^*.

section, we show that these baseline estimators consistently either under- or overestimate how much data to collect, in contrast to LOC.

Table 7 highlights experiments on all six data sets with three alternative regression functions that were used by Mahmood et al. (2022b). For each function and almost on each

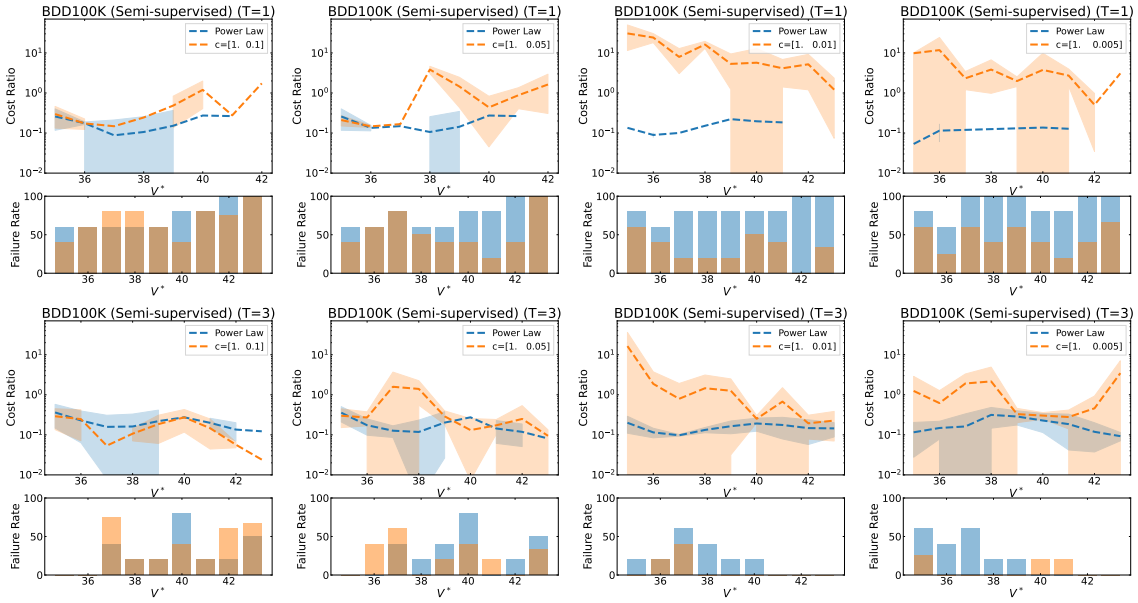

Figure 11: For experiments on BDD100K with two data types, mean ± standard deviation over 5 seeds of the cost ratio $\mathbf{c}^\mathsf{T}(\mathbf{q}_T^* - \mathbf{q}_0)/\mathbf{c}^\mathsf{T}(\mathbf{D}^* - \mathbf{q}_0) - 1$ and failure rate for different V after removing 99-th percentile outliers. We fix $c_0 = 1$ and $P = 10^{13}$. The rows correspond to $T = 1, 3$ (see the main paper for $T = 5$) and the columns correspond to $c_1 = c_0/2, c_0/5, c_0/10, c_0/20$.

dataset, we observe the same trends seen in the original Table 2. That is, LOC reduces the failure rate down to approximately zero, at a marginal relative increase in cost.

Noting that Power Law Regression often leads to failure, Mahmood et al. (2022b) also propose a correction factor heuristic wherein they learn a parameter τ such that if the data collection problem requires a target performance V^*, we should instead aim to collect enough data to meet $V^* + \tau$. In order to learn this correction factor, we require a pre-existing data set upon which we can simulate a data collection policy. Mahmood et al. (2022b) set τ such that we can achieve the data requirement V^* for any V^* on the pre-existing data set, and then fixing this parameter for new data sets.

Table 8 compares LOC (i.e., repeating Table 2) with the Correction factor-based Power Law regression baseline of Mahmood et al. (2022b). Following the original paper, we tune τ using CIFAR-10 and apply it on all other data sets. The correction factor is designed to minimize the failure rate and thus, achieves nearly 0% failure rate for all settings, but often at high cost ratios. On the other hand, LOC achieves generally low failure rates and low cost ratios. Specifically, for $T = 3, 5$, we are competitive with the baseline on failure rates for most tasks while obtaining up to an order of magnitude decrease in costs. For $T = 1$, we typically admit higher failure rates; however for the segmentation and detection tasks, we obtain up multiple orders of magnitude lower costs. Finally, note that this baseline requires a similar prior task to be effective. For example, the baseline outperforms us on cost and failure rate both only on CIFAR-100, since it is tuned on CIFAR-10. On the other hand,

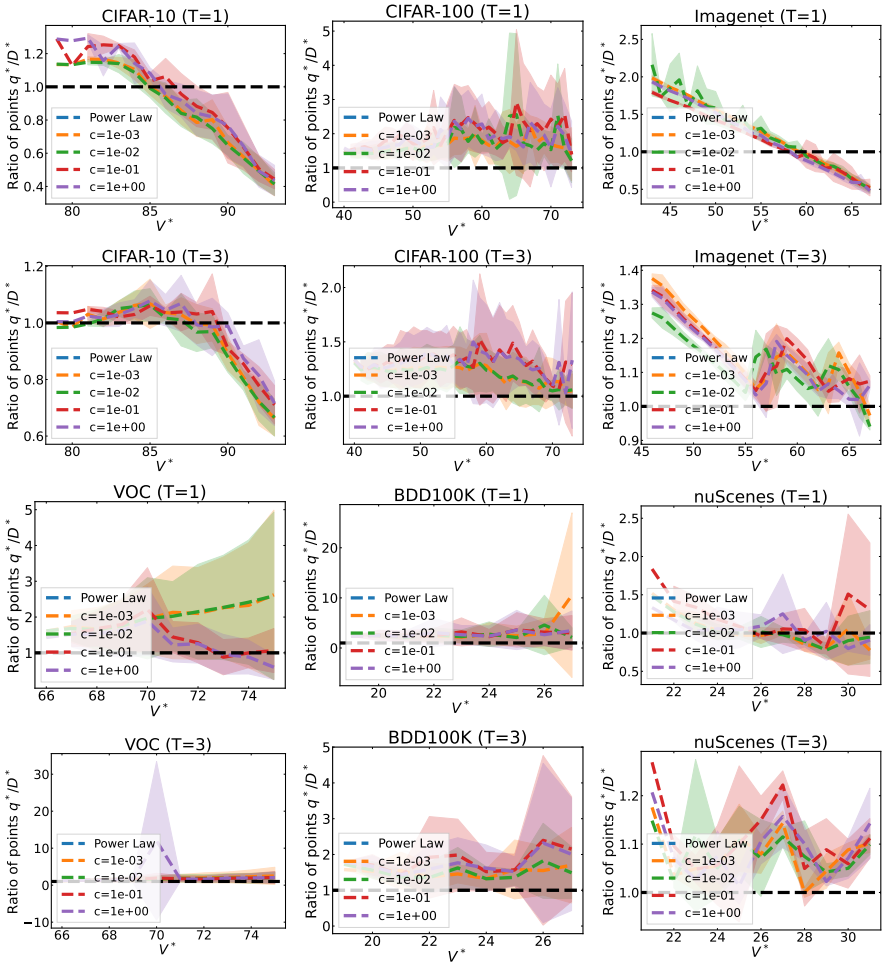

Figure 12: Mean ± standard deviation of the ratio of data collected q_T^*/D^* for different V^* when we sweep the cost parameter from 0.001 to 1 and fix $P = 10^7$. We show $T = 1, 3$ and refer to the main paper for $T = 5$. The dashed black line corresponds to collecting exactly the minimum data requirement.

LOC does not require this prior data set to be effective as evidence by its performance on non-classification tasks.

C.3 LOC with Active Learning

Although most of our experiments assume that data is collected via i.i.d. random sampling, we may also consider alternative approaches to data collection such as active learning. Here, the optimal data collection problem amounts to determining the optimal budget to set when running an active learning query algorithm. Given the budget, we then sample data according to the active learning strategy. This relies on the assumption that the learning curve under active learning is monotone non-decreasing; this may not always be the case

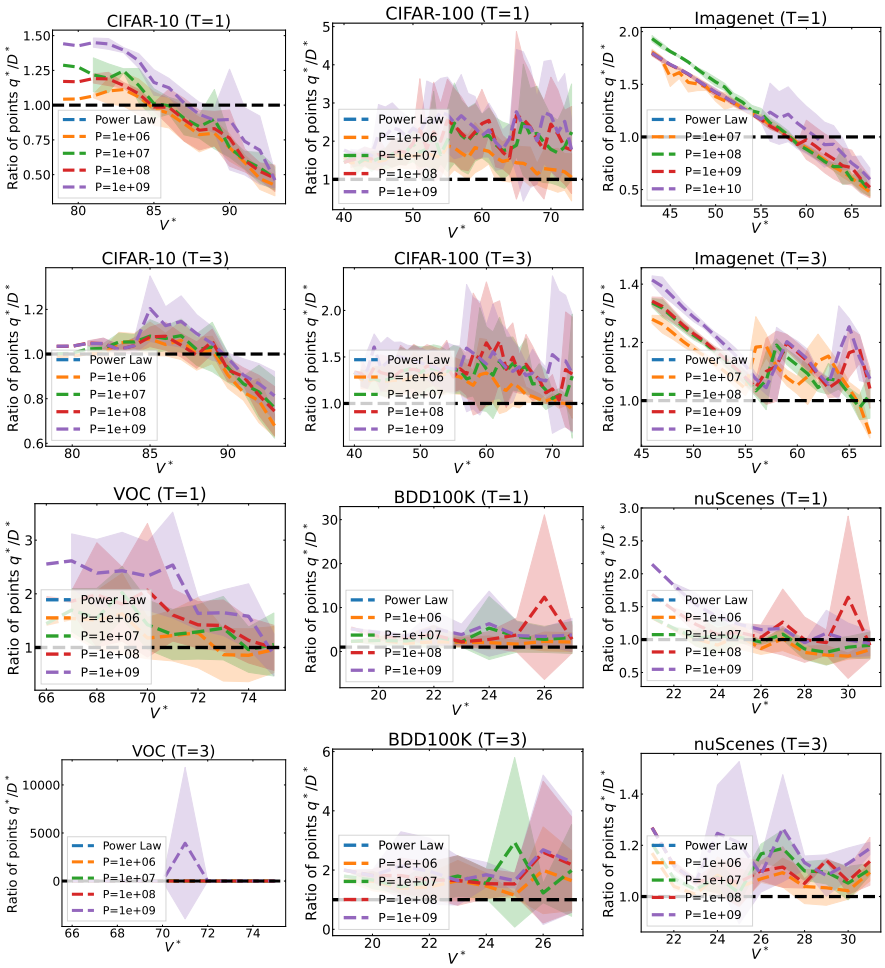

Figure 13: Mean ± std of the ratio of data collected q_T^*/D^* for different V^* when we sweep the penalty parameter from 10^6 to 10^9 and fix $c = 1$. We show $T = 1, 3$ and refer to the main paper for $T = 5$. The dashed black line corresponds to collecting exactly the minimum data requirement.

(e.g., see Viering and Loog (2022)), but has been empirically demonstrated for many common batch-mode active learning strategies for computer vision (e.g., see Sener and Savarese (2018); Mahmood et al. (2022b)). Overall, the LOC framework does not change from Algorithm 3, except in how the training set statistics are collected in Algorithm 1.

Given a small amount of data, we can estimate the neural scaling law from the active learning algorithm and apply estimation-only approaches and LOC. Table 9 highlights the effect of using LOC when the data collection policy uses active learning instead of random sampling. We focus on CIFAR-100 and test three different active learning strategies: Maximum Entropy (Settles, 2009), k-Centers (Sener and Savarese, 2018), and Least Confidence (Settles, 2009). This table demonstrates the same trends as before, namely that our policy

can better make decisions on the data collection budget compared to an estimation-only policy regardless of the specific collection algorithm.

C.4 The Value of Optimization over Estimation when $K = 2$

Figure 10 and Figure 11 expand Figure 5 to $T = 1, 3$ rounds. The results validate the summary observations from Table 3 in that the baseline has considerably higher failure rates versus LOC. In particular for BDD100K at $T = 1$, the baseline fails consistently for four out of five random seeds. On the other hand, recall that LOC admits a higher cost ratio compared to the baseline when $T = 1$. We can observe now that this high cost ratio is due to the method incurring high cost for a few target V^* values. This behavior is similar to the observation above on VOC with high penalties at $T = 3$.

C.5 Robustness to the Cost and Penalty Parameters

Figure 12 expands the cost parameter sweep from Figure 6 (Top row) to the settings of $T = 1, 3$. For nearly all settings, LOC remains stable to variations in the cost parameter. Nonetheless, careful parameter selection becomes important as T decreases. This is due to the fact that for low costs, the total amount of data collected increases as T decreases (e.g., $c = 0.001$ for BDD100K). Furthermore, Figure 13 expands the penalty parameter sweep from Figure 6 (Bottom row). Here, we observe similar properties to the cost parameter sweep.

Although LOC is relatively stable on all other data sets, our results demonstrate some extreme results for VOC, potentially due to noise in the simulation. For example in Figure 13, setting $P = 10^9$, $V^* = 71$, and $T = 3$ led to collecting $10,000$ times the minimum data requirement. Such a situation is unrealistic in a production-level implementation, since in a real implementation, we could impose further constraints onto problem (8), such as upper bounds on the total amount of data permissible.

References

Amazon sagemaker data labeling pricing, 2023. URL `https://aws.amazon.com/sagemaker/data-labeling/pricing/`.

David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

David Acuna, Guojun Zhang, Marc T Law, and Sanja Fidler. f-domain adversarial learning: Theory and algorithms. In *International Conference on Machine Learning*, pages 66–75. PMLR, 2021.

David Acuna, Marc T Law, Guojun Zhang, and Sanja Fidler. Domain adversarial training: A game perspective. In *International Conference on Learning Representations*, 2022.

Yasaman Bahri, Ethan Dyer, Jared Kaplan, Jaehoon Lee, and Utkarsh Sharma. Explaining neural scaling laws. *Proceedings of the National Academy of Sciences*, 121(27):e2311878121, 2024.

Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 1. Athena scientific, 2012.

Dimitris Bertsimas, Mac Johnson, and Nathan Kallus. The power of optimization over randomization in designing experiments involving small samples. *Operations Research*, 63 (4):868–876, 2015.

Devansh Bisla, Apoorva Nandini Saridena, and Anna Choromanska. A theoretical-empirical approach to estimating sample complexity of dnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3270–3280, 2021.

Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. Broken neural scaling laws. In *The Eleventh International Conference on Learning Representations*, 2023.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar 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.

Pedro Carneiro, Sokbae Lee, and Daniel Wilhelm. Optimal data collection for randomized control trials. *The Econometrics Journal*, 23(1):1–31, 2020.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

David Cohn. Neural network exploration using optimal experiment design. In *Advances in Neural Information Processing Systems*, volume 6, 1993.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations*, 2020.

Corinna Cortes, Lawrence D Jackel, Sara Solla, Vladimir Vapnik, and John Denker. Learning curves: Asymptotic values and rate of convergence. In *Advances in Neural Information Processing Systems*, volume 6, 1993.

Fabio Gagliardi Cozman, Ira Cohen, and M Cirelo. Unlabeled data can degrade classification performance of generative classifiers. In *FLAIRS*, pages 327–331, 2002.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

Dimensional Research. Artificial intelligence and machine learning projects are obstructed by data issues: Global survey of data scientists, ai experts and stakeholders. Technical report, 05 2019.

Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *Twenty-fourth International Joint Conference on Artificial Intelligence*, 2015.

David Easley and Nicholas M Kiefer. Controlling a stochastic process with unknown parameters. *Econometrica: Journal of the Econometric Society*, pages 1045–1064, 1988.

Ismail Elezi, Zhiding Yu, Anima Anandkumar, Laura Leal-Taixe, and Jose M Alvarez. Not all labels are equal: Rationalizing the labeling costs for training object detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Ashley F Emery and Aleksey V Nenarokomov. Optimal experiment design. *Measurement Science and Technology*, 9(6):864, 1998.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007.

Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012.

Rosa L Figueroa, Qing Zeng-Treitler, Sasikiran Kandula, and Long H Ngo. Predicting sample size required for classification performance. *BMC Medical Informatics and Decision Making*, 12(1):1–10, 2012.

Lewis J Frey and Douglas H Fisher. Modeling decision tree performance with the power law. In *Seventh International Workshop on Artificial Intelligence and Statistics*. PMLR, 1999.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251. PMLR, 2019.

Baohua Gu, Feifang Hu, and Huan Liu. Modelling classification performance for large data sets. In *International Conference on Web-Age Information Management*, pages 317–328. Springer, 2001.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021.

Joel Hestness, Sharan Narang, Newsha Ardalani, Gregory Diamos, Heewoo Jun, Hassan Kianinejad, Md Patwary, Mostofa Ali, Yang Yang, and Yanqi Zhou. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2022.

Derek Hoiem, Tanmay Gupta, Zhizhong Li, and Michal Shlapentokh-Rothman. Learning curves for analysis of deep networks. In *International Conference on Machine Learning*, pages 4287–4296. PMLR, 2021.

Yiding Jiang, Pierre Foret, Scott Yak, Daniel M Roy, Hossein Mobahi, Gintare Karolina Dziugaite, Samy Bengio, Suriya Gunasekar, Isabelle Guyon, and Behnam Neyshabur. Neurips 2020 competition: Predicting generalization in deep learning. *arXiv preprint arXiv:2012.07976*, 2020.

Yiding Jiang, Parth Natekar, Manik Sharma, Sumukh K Aithal, Dhruva Kashyap, Natarajan Subramanyam, Carlos Lassance, Daniel M Roy, Gintare Karolina Dziugaite, Suriya Gunasekar, et al. Methods and analysis of the first competition in predicting generalization of deep learning. In *NeurIPS 2020 Competition and Demonstration Track*, pages 170–190. PMLR, 2021.

George H. John and Pat Langley. Static versus dynamic sampling for data mining. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 367–370. AAAI Press, 1996.

S Jones, S Carley, and M Harrison. An introduction to power and sample size estimation. *Emergency Medicine Journal: EMJ*, 20(5):453, 2003.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Prasanth Kolachina, Nicola Cancedda, Marc Dymetman, and Sriram Venkatapathy. Prediction of learning curves in machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22–30, 2012.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Mark Last. Predicting and optimizing classifier utility with the power law. In *Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007)*, pages 219–224. IEEE, 2007.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proceedings of the European Conference on Computer Vision*, pages 21–37. Springer, 2016.

Rafid Mahmood, Sanja Fidler, and Marc T. Law. Low-budget active learning via wasserstein distance: An integer programming approach. In *International Conference on Learning Representations*, 2022a.

Rafid Mahmood, James Lucas, David Acuna, Daiqing Li, Jonah Philion, Jose M. Alvarez, Zhiding Yu, Sanja Fidler, and Marc T. Law. How much more data do we need? estimating requirements for downstream tasks. In *2022 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2022b.

Rafid Mahmood, James Lucas, Jose M Alvarez, Sanja Fidler, and Marc T Law. Optimizing data collection for machine learning. In *Advances in Neural Information Processing Systems*, volume 36, 2022c.

Christopher Meek, Bo Thiesson, and David Heckerman. The learning-curve sampling method applied to model-based clustering. *Journal of Machine Learning Research*, 2(Feb):397–418, 2002.

Hiroaki Mikami, Kenji Fukumizu, Shogo Murai, Shuji Suzuki, Yuta Kikuchi, Taiji Suzuki, Shin-ichi Maeda, and Kohei Hayashi. A scaling law for syn2real transfer: How much is your pre-training effective? In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2022, Grenoble, France, September 19–23, 2022, Proceedings, Part III*, page 477–492, 2022.

Felix Mohr, Tom J Viering, Marco Loog, and Jan N van Rijn. Lcdb 1.0: An extensive learning curves database for classification tasks. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 3–19. Springer, 2022.

Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Proceedings of the European Conference on Computer Vision*, 2020.

Viraj Uday Prabhu, David Acuna, Rafid Mahmood, Marc T Law, Yuan-Hong Liao, Judy Hoffman, Sanja Fidler, and James Lucas. Bridging the sim2real gap with CARE: Supervised detection adaptation with conditional alignment and reweighting. *Transactions on Machine Learning Research*, 2023.

Aayush Prakash, Shoubhik Debnath, Jean-Francois Lafleche, Eric Cameracci, Gavriel State, Stan Birchfield, and Marc T. Law. Self-supervised real-to-sim scene generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16044–16054, October 2021.

Foster Provost, David Jensen, and Tim Oates. Efficient progressive sampling. In *Proceedings of the fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 23–32, 1999.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Jonathan S Rosenfeld, Amir Rosenfeld, Yonatan Belinkov, and Nir Shavit. A constructive prediction of the generalization error across scales. In *International Conference on Learning Representations*, 2020.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

Burr Settles. Active learning literature survey. 2009.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019.

Richard D Smallwood and Edward J Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations Research*, 21(5):1071–1088, 1973.

Kirstine Smith. On the standard deviations of adjusted and interpolated values of an observed polynomial function and its constants and the guidance they give towards a proper choice of the distribution of observations. *Biometrika*, 12(1/2):1–85, 1918.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 843–852, 2017.

Katrin Tomanek and Udo Hahn. Approximating learning curves for active-learning-driven annotation. In *LREC*, volume 8, pages 1319–1324, 2008.

VentureBeat. Why do 87% of data science projects never make it into production?, Jul 2019. URL `https://venturebeat.com/2019/07/19/why-do-87-of-data-science-projects-never-make-it-into-production/`.

Felipe AC Viana. A tutorial on latin hypercube design of experiments. *Quality and reliability engineering international*, 32(5):1975–1985, 2016.

Tom Viering and Marco Loog. The shape of learning curves: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Sen Wang and KJ Yang. An image scaling algorithm based on bilinear interpolation with vc++. *Techniques of Automation and Applications*, 27(7):44–45, 2008.

Donggeun Yoo and In So Kweon. Learning loss for active learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 93–102, 2019.

Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. BDD100K: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2020.

Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Eric Zhao, Anqi Liu, Animashree Anandkumar, and Yisong Yue. Active learning under label shift. In *International Conference on Artificial Intelligence and Statistics*, pages 3412–3420. PMLR, 2021.