

Spend Less, Fit Better: Budget-Efficient Scaling Law Fitting via Active Experiment Selection

Sijie Li^{1*}, Shanda Li^{2*}, Haowei Lin¹, Weiwei Sun², Ameet Talwalkar^{2,3}, Yiming Yang²

¹Peking University ²Carnegie Mellon University ³Datadog

Abstract

Scaling laws are used to plan multi-million-dollar training runs, but fitting those laws can itself cost millions. In modern large-scale workflows, assembling a sufficiently informative set of pilot experiments is already a major budget-allocation problem rather than a routine preprocessing step. We formulate scaling-law fitting as budget-aware sequential experimental design: given a finite pool of runnable experiments with heterogeneous costs, choose which runs to execute so as to maximize extrapolation accuracy in a high-cost target region. We propose SL² (Scaling Laws, Spend Less), an uncertainty-aware method for sequentially allocating experimental budget toward the runs most useful for target-region extrapolation. Across a diverse benchmark of scaling-law tasks, SL² outperforms classical design-based baselines, and often approaches the performance of fitting on the full experimental set while using only about 10% of the total training budget. Our code is available at <https://github.com/PlanarG/active-sl>.

1 Introduction

Scaling laws have become a central tool for analyzing and planning large-scale language model training (Hestness et al., 2017; Kaplan et al., 2020; Gao et al., 2023; Li et al., 2025b). By fitting a parametric relationship between performance and variables such as model size, data size, and compute, practitioners can use a limited set of pilot runs to predict behavior at much larger scales and allocate future training budget accordingly (Yang et al., 2021; Li et al., 2025a; Xu et al., 2026). This paradigm has already shaped influential decisions in practice: For example, Chinchilla-style compute-optimal training was derived from an extensive empirical study spanning more than 400 training runs across a wide range of scales (Hoffmann et al., 2022).

Despite their practical importance, scaling laws remain expensive to fit and heavily reliant on manual experiment design. In typical workflows, researchers hand-select experimental configurations, run many pilot trainings, and then fit a parametric law to the resulting observations. At industrial scale, the pilot runs needed just to fit a scaling law can themselves consume a massive budget (Porian et al., 2024; Hägele et al., 2024), with the full fit-and-verify pipeline already reaching the **million-dollar scale** before any deployment-scale training is committed. Accurate scaling-law fitting is therefore not only a modeling problem, but also a problem of budget allocation.

This challenge is becoming more acute as scaling-law analysis expands beyond classical dense pretraining. Recent work has extended scaling laws to a much broader range of settings, including vocabulary design, data mixing, sparsity, mixture-of-experts architectures, and inference-time scaling (Muennighoff et al., 2023; Caballero et al., 2023; Ludziejewski et al., 2024). As these settings become more diverse and expensive, manually selecting pilot runs becomes increasingly inefficient. This raises a natural question:

Given a pool of runnable experiments and a limited budget, how should we select experiments to ensure that the fitted scaling law extrapolates accurately in the target region?

*Equal contribution.

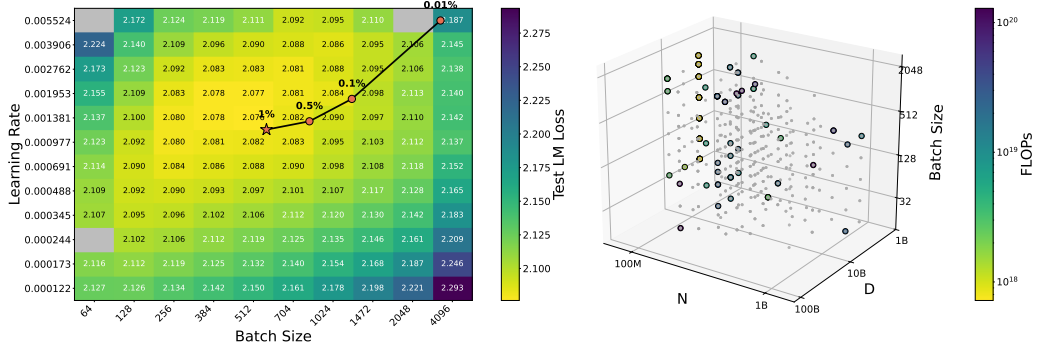


Figure 1: **Budget-efficient extrapolation** on 1r&bsz. SL^2 identifies the extrapolation optimum using only a small fraction of the original scaling-law fitting budget. On 1r&bsz, the predicted optimum over learning rate and batch size for a 1B-parameter LLM trained on 100B tokens reaches the low-loss region within 1% of the budget, as shown by the trajectory on the test-loss heatmap (left). The selected training configurations used for fitting are shown on the right in a 3D view of the design space, colored by training cost ($6ND$), illustrating that accurate extrapolation can be achieved from a sparse, low-cost subset of the full configuration space.

In this paper, we formulate scaling-law fitting as a problem of *budget-aware sequential experimental design*. Rather than assuming that fitting data are given in advance, we treat each candidate experiment as a costly query and ask how to allocate a limited budget over a finite pool of runnable configurations. The objective is not merely to fit the observed points well, but to maximize predictive accuracy in a held-out target region lying in the large-scale, high-cost regime. This formulation more directly reflects the practical use of scaling laws, where the ultimate goal is to choose a few expensive final configurations rather than to uniformly fit all pilot runs. To address this objective, we propose SL^2 (Scaling Laws, Spend Less), a sequential, uncertainty-aware design method for budget-efficient scaling-law fitting.

Our contributions are twofold.

- We formulate the problem and present the first systematic study of scaling-law fitting in the low-budget regime, where accurate extrapolation must be recovered from only a small fraction of the training runs used in standard scaling-law analyses. To support this study, we conduct a comprehensive evaluation spanning diverse scaling scenarios, heterogeneous law families, and task-specific cost structures, enabling controlled comparison of point-selection strategies under budget constraints.
- SL^2 explicitly models ambiguity over scaling-law parameters and selects new experiments according to their expected value for reducing target-region prediction error. This positions scaling-law fitting as a target-extrapolation compute-allocation problem rather than a standard parameter-estimation design problem. Empirically, our method is highly effective: as shown in Figure 1, it approaches the extrapolation ground-truth optimum using only a small fraction of the original fitting FLOPs, reaching the low-loss region within 1% budget on *Step Law* (Learning Rate and Batch Size Scaling Law) (Li et al., 2025a).

2 Related Work

Scaling Laws. Scaling laws have transformed the design and optimization of large-scale AI systems by revealing predictable relationships among model size, data volume, and compute budget (Cortes et al., 1993; Kaplan et al., 2020; Hoffmann et al., 2022; Hu et al., 2024). Recent work has extended this paradigm far beyond its original setting, covering model architectures (Fedus et al., 2022), data scaling (Ye et al., 2025; Que et al., 2024), post-training behavior (Hernandez et al., 2021; Lin et al., 2024), multimodal regimes (Radford et al., 2021;

Zhai et al., 2022), and inference-time settings (Chen et al., 2024; 2025; Wu et al., 2025; Huang et al., 2026). Yet fitting scaling laws in practice remains costly and heavily manual: the final fit can be highly sensitive to the choice of law family, initialization, and, crucially, the strategy used to collect training runs (Li et al., 2025c). SL² automatically selects new runs based on the current fitting state and the target region where accurate prediction matters most, without relying on manually designed experimental points.

Optimal experiment design for nonlinear models. Classical optimal experiment design (OED) studies how to place experiments to estimate model parameters or derived quantities efficiently, with criteria such as D-optimality and A-optimality defined through the Fisher information matrix (Silvey, 2013; Kiefer, 1959; Gilmour & Trinca, 2013). For nonlinear models, these criteria typically depend on unknown parameters, leading to a large literature on locally optimal design for nonlinear and generalized linear models (White, 1973; Khuri et al., 2006; Atkinson & Woods, 2015; Yang et al., 2013). However, this line of work is primarily local and parameter-estimation-oriented, and typically does not consider heterogeneous experiment costs. Bayesian optimal experimental design addresses parameter uncertainty by optimizing expected utility under a posterior distribution (Huan & Marzouk, 2013; Zhong et al., 2026), which is appealing for scaling-law fitting where the objective is highly nonlinear and limited observations may support multiple plausible fits. However, existing Bayesian OED methods do not directly address our setting, in which candidate experiments are discrete training runs with heterogeneous compute costs and must be selected sequentially under a strict budget. We therefore study a cost-aware sequential design problem tailored to scaling-law fitting, where experiments are prioritized by their expected predictive benefit relative to their cost.

3 Problem Setup

We study the scaling-law fitting problem in a budget-constrained sequential setting. We assume that the underlying performance trend is described by a parametric scaling law $y = f(x; \theta)$, where $x \in \mathcal{X}$ denotes the modeling configuration, e.g., model size, token count, or other training- or inference-related hyperparameters; $y \in \mathbb{R}$ denotes the prediction target of the scaling law, e.g., training loss; and $\theta \in \mathbb{R}^p$ denotes the parameter to be estimated from the experiments. Running an experiment under configuration x incurs a nonnegative cost $c(x)$ and reveals an outcome y . At a high level, we aim to run a set of experiments under a cost constraint so as to obtain data for fitting an accurate scaling law.

In practice, experiment selection is performed over a set of predefined runnable configurations. We therefore consider a candidate pool $\mathcal{X}_{\text{cand}} = \{x_1, \dots, x_N\}$, where each candidate x_i has an associated cost $c_i = c(x_i)$. At each round, the learner selects one previously unobserved candidate, pays its cost, observes its outcome, and adds the resulting pair to the current dataset. After t rounds, the accumulated observations form a dataset $\mathcal{D}_t$, and the total cost of all selected experiments must remain within a budget C .

The ultimate goal is not merely to fit the observed points well, but to learn a scaling law that extrapolates accurately in a target region $\mathcal{X}_{\text{tar}}$. This target region typically contains the larger-scale configurations that matter most for downstream planning, but are too expensive to explore exhaustively, whereas $\mathcal{X}_{\text{cand}}$ typically contains more affordable small-scale runs. Our objective is therefore to design a sequential experiment-selection strategy that uses the available budget as efficiently as possible, so that the fitted scaling law is accurate where it ultimately matters.

4 SL²: Budget-Aware Sequential Scaling-Law Design

We now describe SL², our sequential design strategy for scaling-law fitting under a budget constraint. At round t , given the current dataset $\mathcal{D}_t$, our goal is to select the next experiment $x \in \mathcal{X}_{\text{cand}}$ that most improves predictive accuracy on the target region $\mathcal{X}_{\text{tar}}$.

To make the design objective concrete, we assume the standard observation model $y = f(x; \theta^*) + \varepsilon$ with some unknown ground-truth parameter θ^* and noise $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.

4.1 A Target-Aware Uncertainty Objective

We would like the utility of an experiment to reflect our downstream goal: improving prediction accuracy on the target region $\mathcal{X}_{\text{tar}}$. In our setting, uncertainty comes from two sources: local uncertainty within a plausible fit, and disagreement across multiple plausible fits that extrapolate differently on $\mathcal{X}_{\text{tar}}$. We call each locally optimal fit of the scaling law a *basin*. Given the observations $\mathcal{D}_t$ collected so far, we approximate the posterior:

$$p(\theta \mid \mathcal{D}_t) \approx \sum_{k=1}^K w_k q_k(\theta), \quad q_k(\theta) = \mathcal{N}(\theta_k, \Sigma_k),$$

where each component represents one plausible local basin, with representative parameter θ_k , local covariance Σ_k , and mixture weight $w_k \geq 0$ satisfying $\sum_{k=1}^K w_k = 1$. In our implementation, θ_k , Σ_k , and w_k are estimated from local refits and local posterior approximations; details are given in Appendix C.

To measure uncertainty where extrapolation matters, define the target-region prediction map

$$F(\theta) = (f(x; \theta))_{x \in \mathcal{X}_{\text{tar}}} \in \mathbb{R}^{|\mathcal{X}_{\text{tar}}|}.$$

Let

$$\hat{f}_k = F(\theta_k), \quad \bar{f} = \mathbb{E}_{\theta \sim p(\theta \mid \mathcal{D}_t)}[F(\theta)] \approx \sum_{k=1}^K w_k \hat{f}_k.$$

We use the target-region mean squared prediction error

$$\text{MSPE}_{\text{tar}} = \frac{1}{|\mathcal{X}_{\text{tar}}|} \mathbb{E}_{\theta \sim p(\theta \mid \mathcal{D}_t)} [\|F(\theta) - \bar{f}\|_2^2]$$

as our uncertainty objective. Under a local Gaussian approximation within each basin, this quantity decomposes into

$$\text{MSPE}_{\text{tar}} = V_{\text{intra}} + V_{\text{inter}},$$

where

$$V_{\text{intra}} = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K w_k \text{tr}(J_k \Sigma_k J_k^\top), \quad V_{\text{inter}} = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K w_k \|\hat{f}_k - \bar{f}\|_2^2.$$

Here $J_k \in \mathbb{R}^{|\mathcal{X}_{\text{tar}}| \times p}$ is the Jacobian of $F(\theta)$ evaluated at θ_k . Intuitively, the first term quantifies local predictive uncertainty within each basin, while the second quantifies disagreement across basins in the target region.

4.2 Scoring Candidate Experiments

For a candidate experiment $x \in \mathcal{X}_{\text{cand}}$, we define its utility as the expected reduction in target-region uncertainty after observing its outcome:

$$\Delta \text{MSPE}_{\text{tar}}(x) = \text{MSPE}_{\text{tar}} - \mathbb{E}_{y \mid x, \mathcal{D}_t} [\text{MSPE}_{\text{tar}}^+(x, y)],$$

where $\text{MSPE}_{\text{tar}}^+(x, y)$ denotes the updated target-region MSPE after augmenting $\mathcal{D}_t$ with (x, y) . Using the decomposition above, we write

$$\Delta \text{MSPE}_{\text{tar}}(x) = \Delta V_{\text{intra}}(x) + \Delta V_{\text{inter}}(x),$$

with

$$\Delta V_{\text{intra}}(x) = V_{\text{intra}} - \mathbb{E}_{y \mid x, \mathcal{D}_t} [V_{\text{intra}}^+(x, y)], \quad \Delta V_{\text{inter}}(x) = V_{\text{inter}} - \mathbb{E}_{y \mid x, \mathcal{D}_t} [V_{\text{inter}}^+(x, y)].$$

The first term favors candidates that reduce within-basin predictive variance on $\mathcal{X}_{\text{tar}}$, while the second favors candidates that disambiguate basins with different extrapolations. To account for heterogeneous experiment costs, we rank candidates by the cost-aware score

$$S(x) = \frac{\Delta V_{\text{intra}}(x) + \Delta V_{\text{inter}}(x)}{c(x)^\alpha},$$

where $\alpha \geq 0$ controls the strength of cost penalization.

Algorithm 1 SL²: Budget-aware sequential design for scaling-law fitting.

Require: Initial dataset $\mathcal{D}_0$, candidate pool $\mathcal{X}_{\text{cand}}$, target region $\mathcal{X}_{\text{tar}}$, cost function $c(\cdot)$, cost exponent α

```

1: for  $t = 0, 1, 2, \dots$  until budget is exhausted do
2:    $\{(\theta_k, \Sigma_k, w_k)\}_{k=1}^K \leftarrow \text{ESTIMATEBASINS}(\mathcal{D}_t)$ 
3:   for each  $x \in \mathcal{X}_{\text{cand}}$  do
4:      $\Delta V_{\text{intra}}(x) \leftarrow \text{INTRAUILITY}(x, \{(\theta_k, \Sigma_k, w_k)\}_{k=1}^K, \mathcal{X}_{\text{tar}})$ 
5:      $\Delta V_{\text{inter}}(x) \leftarrow \text{INTERUTILITY}(x, \{(\theta_k, \Sigma_k, w_k)\}_{k=1}^K, \mathcal{X}_{\text{tar}})$ 
6:      $S(x) \leftarrow (\Delta V_{\text{intra}}(x) + \Delta V_{\text{inter}}(x)) / c(x)^\alpha$ 
7:   end for
8:    $x_{t+1} \leftarrow \arg \max_{x \in \mathcal{X}_{\text{cand}}} S(x)$ 
9:    $y_{t+1} \leftarrow \text{RUNEXPERIMENT}(x_{t+1})$ 
10:   $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(x_{t+1}, y_{t+1})\}$ 
11:   $\mathcal{X}_{\text{cand}} \leftarrow \mathcal{X}_{\text{cand}} \setminus \{x_{t+1}\}$ 
12: end for
13: return  $\mathcal{D}_t$ 

```

4.3 Computing Intra- and Inter-Basin Utility

We approximate both utility terms by locally linearizing the predictor within each basin while preserving the multimodal structure across basins. Full derivations are deferred to Appendix D.

Intra-basin utility. For a candidate x , let

$$J_x(\theta_k) = \frac{\partial f(x; \theta)}{\partial \theta} \Big|_{\theta=\theta_k} \in \mathbb{R}^{1 \times p}$$

denote the parameter Jacobian at basin k . Under the local linear approximation, the reduction in within-basin target uncertainty is

$$\Delta V_{\text{intra}}(x) = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K w_k \frac{\|J_k \Sigma_k J_x(\theta_k)^\top\|_2^2}{\sigma^2 + J_x(\theta_k) \Sigma_k J_x(\theta_k)^\top}.$$

This term is large when observing y at x is expected to substantially reduce predictive variance over the target region.

Inter-basin utility. The inter-basin gain is

$$\Delta V_{\text{inter}}(x) = V_{\text{inter}} - \int V_{\text{inter}}^+(x, y) p(y | x, \mathcal{D}_t) dy,$$

where $V_{\text{inter}}^+(x, y)$ is the updated between-basin uncertainty after observing outcome y at candidate x . The predictive distribution $p(y | x, \mathcal{D}_t)$ is the scalar mixture induced by the current basin approximation. Because the expectation is one-dimensional, we evaluate it efficiently using numerical quadrature.

4.4 The Sequential Design Procedure

Algorithm 1 summarizes the full procedure. At each round, we first update the basin approximation from the current dataset, then score all remaining candidates using the target-aware acquisition function above, and finally select the highest-scoring affordable experiment.

Task	# Laws	Avg. Params	# Train	# Test	Target	Cost
parallel	10	4.2	36	12	$L(N, P)$	N
vocab	10	6.8	1080	120	$L(N, V, D)$	$6ND$
domain	10	29.0	504	42	$\{L_i(\mathbf{r})\}_{i=1}^5$	1
moe	10	5.3	193	28	$L(N, E)$	NE
data_con	10	7.0	161	21	$L(N, D, U)$	$6ND$
lr&bsz	10	20.2	2702	117	$L(l, b, N, D)$	$6ND$
sparsity	4	5.0	70	18	$L(P, N_2)$	$6N_1D_1 + 6N_2D_2$
farseeer	1	9.0	404	7	$L(N, D)$	$6ND$

Table 1: **Benchmark task statistics.** Each task contains a collection of scaling-law instances for evaluating budget-aware sequential design on target-region extrapolation.

5 Experiments

5.1 Experimental Setup

Benchmark overview. We evaluate SL^2 on a benchmark for budget-aware sequential design in scaling-law fitting, comprising 8 tasks and 65 scaling-law instances. Each instance specifies a parametric law family, a finite pool of runnable candidate experiments with associated costs, and a held-out target region for evaluation. The tasks cover diverse LLM scaling scenarios, including pre-training hyperparameter tuning, data allocation, architecture design, sparsity, and inference-time scaling. Table 1 summarizes the task-level statistics; detailed task descriptions and data sources are deferred to Appendix B.1.

Law families and cost models. The benchmark spans a heterogeneous collection of scaling-law families, including classical power laws, log-space interaction models, compositional mixture laws, hyperparameter response surfaces, and several more expressive nonlinear forms; full parameterizations are deferred to Appendix B.2. We assign each task a simple cost proxy aligned with its dominant resource. For dense-training settings (data_con, farseeer, lr&bsz, and vocab), we use $6ND$. For the remaining tasks, we use task-specific proxies: NE for moe, N for parallel, $6N_1D_1 + 6N_2D_2$ for sparsity, and unit cost for domain, which varies mixture proportions rather than overall training scale.

Evaluation protocol and metrics. At the start of each episode, a method receives the candidate pool, target region, and candidate costs, but not the outcomes. It sequentially selects feasible unobserved candidates and refits the law after every observation using L-BFGS-B from 64 initializations. We repeat each method for 10 runs. We report target-region R^2 at 1%, 5%, and 10% of total training cost for most tasks, and at 20%, 35%, and 50% for domain and sparsity, whose costs are more uniform. Scores are aggregated over runs and instances within each task, clipped to $[-1, 1]$, and reported as mean $\pm$ standard deviation. All methods use the same fitting procedure and differ only in experiment selection.

Baselines. We compare against five baselines. (1) **Random** uniformly samples the next experiment from the feasible unobserved candidates. (2) **Cheapest** always selects a minimum-cost feasible candidate, breaking ties uniformly at random. (3) **Cost Rand** samples each feasible unobserved candidate with probability proportional to $1/c(x)$. These three heuristics respectively capture uninformed exploration, an aggressive preference for cheap experiments, and a simple stochastic bias toward lower-cost candidates.

We further compare against two classical design criteria adapted to our nonlinear, cost-constrained setting. (4) **D-opt** selects the candidate that maximizes the increase in a D-optimality objective, which favors experiments that most reduce the overall volume of parameter uncertainty. In the locally linearized model, this corresponds to preferring points that most increase the log-determinant of the Fisher information matrix. (5) **V-opt** selects the candidate that maximizes a V-optimality objective, which favors experiments expected to most reduce predictive variance over the target region. In our implementation, both

Component	Refitting	Basin construction	Acquisition scoring	Total decision time
Mean $\pm$ Std. (s)	1.136 $\pm$ 0.575	0.065 $\pm$ 0.055	0.059 $\pm$ 0.188	1.260 $\pm$ 0.582

Table 2: **Decision overhead** on lr&bsz. Wall-clock decision time is reported separately from the executed training-run budget.

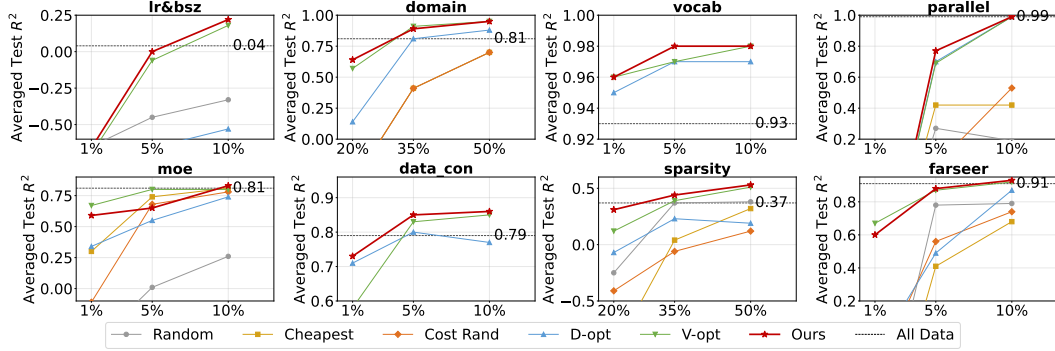


Figure 2: **Budget-accuracy curves** for target-region extrapolation. Mean target-region R^2 is shown as a function of consumed budget on the benchmark. SL^2 reaches the best aggregate budget-accuracy trade-off and approaches the full-data reference using only a small fraction of the total experimental cost.

D-opt and **V-opt** locally linearize the nonlinear scaling law around the parameter estimate with the lowest MSE among the fitted solutions from the previous step, and differ only in the acquisition score computed from this linearization. To account for heterogeneous costs, each candidate score is normalized by $c(x)^\alpha$, with a single global $\alpha = 0.4$ for all tasks and instances; Appendix E.1 reports a sensitivity analysis.

For **D-opt**, **V-opt**, and SL^2 , we use a short warm-start phase before the first criterion-based acquisition step: each method first selects the $2.5p$ lowest-cost candidates, where p is the number of law parameters. The cost of these initialization points is counted toward the total budget, and Appendix E.2 reports both the warm-start cost fraction and an ablation over warm-start sizes. As a full-information reference, we also report **All Data**, obtained by fitting the same scaling law on the entire training set and evaluating the resulting target-region R^2 .

Computational overhead. To quantitatively characterize the computational overhead of using SL^2 in the sequential design loop, we profile the wall-clock time of each selection decision on lr&bsz. This task has the largest candidate pool and relatively high-dimensional laws, making it one of the most computationally demanding settings in our benchmark. Table 2 decomposes the total decision time into refitting, basin construction, and acquisition scoring. The results indicate that a decision takes 1.260 seconds on average, of which 1.136 seconds (about 90%) is spent on refitting, while basin construction and acquisition scoring together account for only about 10%. Thus, the additional overhead specific to our acquisition procedure is small relative to the nonlinear refitting already required at each step. Appendix E.3 gives the scoring complexity.

5.2 Results Analysis

Overall trends. Figure 2 and Table 3 show that SL^2 delivers the best aggregate performance. Its advantage is most pronounced in the low-budget regime, where experiment selection matters most: at 1% budget, it performs best on domain, data_con, and sparsity, matches the top result on vocab, and remains competitive on moe and farseer. As the budget increases, the advantage becomes more consistent. At 5% budget, it achieves the best result

Setting	lr&bsz	domain	vocab	parallel	moe	data_con	sparsity	farseer
1% Random	-0.65 ± 0.49	-0.36 ± 0.82	0.54 ± 0.57	-1.00 ± 0.00	-0.47 ± 0.66	-0.74 ± 0.48	-0.25 ± 0.63	-0.77 ± 0.55
1% Cheapest	-0.92 ± 0.29	-0.36 ± 0.82	0.30 ± 0.71	-1.00 ± 0.00	0.30 ± 0.57	-0.58 ± 0.64	-0.95 ± 0.14	-0.89 ± 0.17
1% Cost Rand	-0.89 ± 0.31	-0.36 ± 0.82	0.86 ± 0.38	-1.00 ± 0.00	-0.11 ± 0.70	-0.38 ± 0.70	-0.41 ± 0.52	-0.25 ± 0.63
1% D-opt	-0.92 ± 0.23	0.14 ± 0.80	0.95 ± 0.03	-1.00 ± 0.00	0.34 ± 0.54	0.71 ± 0.43	-0.07 ± 0.56	-0.11 ± 0.80
1% V-opt	-0.70 ± 0.47	0.57 ± 0.66	0.96 ± 0.02	-1.00 ± 0.00	0.67 ± 0.24	0.58 ± 0.53	0.12 ± 0.53	0.67 ± 0.22
1% SL ² (ours)	-0.66 ± 0.53	0.64 ± 0.58	0.96 ± 0.02	-1.00 ± 0.00	0.59 ± 0.39	0.73 ± 0.37	0.31 ± 0.39	0.60 ± 0.11
5% Random	-0.45 ± 0.61	0.41 ± 0.82	0.53 ± 0.62	0.27 ± 0.02	0.01 ± 0.72	-0.36 ± 0.71	0.37 ± 0.26	0.78 ± 0.18
5% Cheapest	-0.88 ± 0.34	0.41 ± 0.82	0.53 ± 0.62	0.42 ± 0.03	0.74 ± 0.19	-0.56 ± 0.65	0.04 ± 0.46	0.41 ± 0.40
5% Cost Rand	-0.79 ± 0.43	0.41 ± 0.82	0.89 ± 0.27	0.03 ± 0.87	0.68 ± 0.33	-0.39 ± 0.65	-0.06 ± 0.52	0.56 ± 0.38
5% D-opt	-0.66 ± 0.57	0.81 ± 0.48	0.97 ± 0.01	0.70 ± 0.52	0.55 ± 0.34	0.80 ± 0.17	0.23 ± 0.32	0.49 ± 0.18
5% V-opt	-0.06 ± 0.59	0.91 ± 0.33	0.97 ± 0.00	0.69 ± 0.53	0.80 ± 0.05	0.83 ± 0.17	0.39 ± 0.20	0.87 ± 0.01
5% SL ² (ours)	0.00 ± 0.59	0.89 ± 0.38	0.98 ± 0.00	0.77 ± 0.39	0.65 ± 0.27	0.85 ± 0.16	0.44 ± 0.17	0.88 ± 0.02
10% Random	-0.33 ± 0.66	0.70 ± 0.67	0.59 ± 0.54	0.19 ± 0.01	0.26 ± 0.65	0.12 ± 0.62	0.38 ± 0.25	0.79 ± 0.09
10% Cheapest	-0.80 ± 0.46	0.70 ± 0.67	0.55 ± 0.63	0.42 ± 0.03	0.81 ± 0.14	-0.40 ± 0.64	0.32 ± 0.20	0.68 ± 0.18
10% Cost Rand	-0.79 ± 0.39	0.70 ± 0.67	0.83 ± 0.34	0.53 ± 0.81	0.78 ± 0.23	-0.22 ± 0.67	0.12 ± 0.38	0.74 ± 0.11
10% D-opt	-0.53 ± 0.57	0.88 ± 0.43	0.97 ± 0.00	0.99 ± 0.00	0.74 ± 0.14	0.77 ± 0.16	0.19 ± 0.41	0.87 ± 0.03
10% V-opt	0.18 ± 0.54	0.95 ± 0.27	0.98 ± 0.00	0.99 ± 0.00	0.80 ± 0.11	0.85 ± 0.13	0.51 ± 0.15	0.92 ± 0.01
10% SL ² (ours)	0.22 ± 0.55	0.95 ± 0.28	0.98 ± 0.00	0.99 ± 0.00	0.83 ± 0.07	0.86 ± 0.11	0.53 ± 0.08	0.93 ± 0.00
All Data	0.04 ± 0.67	0.81 ± 0.51	0.93 ± 0.16	0.99 ± 0.00	0.81 ± 0.04	0.79 ± 0.23	0.37 ± 0.10	0.91 ± 0.01

Table 3: **Task-level target-region performance.** Each cell reports the mean and standard deviation of target-region R^2 aggregated over all scaling-law instances within the corresponding task. Higher is better, and R^2 is clipped to $[-1, 1]$. Budgets are 1%, 5%, 10% for most tasks, and 20%, 35%, 50% for domain and sparsity.

on six of the eight tasks, and at 10% budget, it matches or outperforms all baselines on every task.

A consistent pattern is that model-aware design substantially outperforms simple budget heuristics. Across most tasks and budgets, **Random**, **Cheapest**, and **Cost Rand** lag far behind, especially under tight budgets, showing that neither uninformed exploration nor naive cost preference is sufficient for reliable target-region extrapolation.

Comparison to design-based baselines. Among the design-based baselines, **D-opt** is already much stronger than the simple heuristics, highlighting the value of exploiting local sensitivity information. **V-opt** is closer in spirit to SL²: both are prediction-oriented, but **V-opt** relies on a single local linearization around the current best-fit parameter, whereas SL² maintains a mixture-of-Gaussians representation over multiple plausible parameter regions. Empirically, our method is generally more robust and achieves better overall performance, especially at moderate and higher budgets. This gap is most pronounced when the fitting landscape contains multiple plausible basins and the target region is strongly extrapolative, so that the current best fit need not be the right expansion point for design. Figure 3 illustrates such a case on one lr&bsz scaling law using t-SNE visualization (van der Maaten & Hinton, 2008).

Task and budget dependence. Performance varies substantially across tasks, reflecting the heterogeneous structure of the benchmark. Because each entry in Table 3 averages over multiple scaling-law instances within a task, these results should be interpreted as robustness over a heterogeneous family of fitting problems rather than performance on any single law. Some tasks are relatively easy on average: on vocab, all design-based methods already perform strongly at low budget, and the gap nearly closes by 5% budget; similarly, parallel is close to saturated once the budget becomes moderate. By contrast, lr&bsz, domain, data_con, and sparsity remain more challenging on average and show larger separation among methods across a wider range of budgets. Overall, low-budget

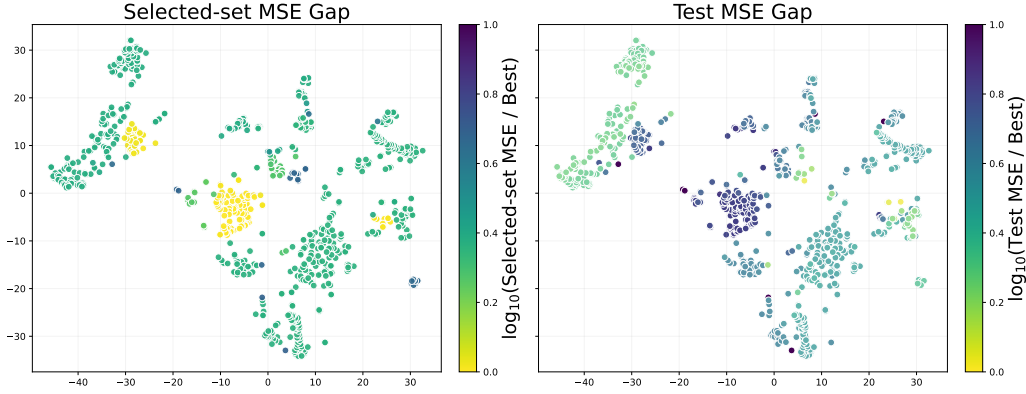


Figure 3: **Local optima and extrapolation mismatch** in lr&bsz. Parameter-space visualization for one lr&bsz scaling law (sl.5) after fitting on the cheapest 12% of training points from 2048 initializations. We embed the fitted parameters with t-SNE and color each solution by its MSE on the selected points (left) or on the held-out test region (right). Multiple separated clusters indicate many local optima, while the mismatch between the two colorings shows that low error on the observed low-cost points does not reliably imply low error on the high-cost extrapolation region. SL^2 achieves 0.71 averaged test R^2 within 5% budget compared with 0.57 for V-opt and 0.16 for D-opt.

Metric	lr&bsz	domain	vocab	parallel	moe	data.con	sparsity	farseer	Average
SL^2 @ final checkpoint	0.22	0.95	0.98	0.99	0.83	0.86	0.53	0.93	0.79
Cheap half	-0.89	0.79	0.84	1.00	0.42	0.54	-0.67	0.51	0.32
Expensive half	0.24	0.79	0.42	1.00	0.33	0.84	0.48	0.79	0.61

Table 4: **Cheap and expensive half diagnostic.** Target-region R^2 for the final-checkpoint design selected by SL^2 versus fits using only the cheapest or most expensive half of the candidate pool. Higher is better.

performance is the main differentiator across methods, while higher-budget results reveal which approaches remain robust on the harder and more heterogeneous tasks.

Why can some methods outperform All Data? Some budgeted methods occasionally outperform **All Data**. This is not contradictory, because **All Data** is not an oracle upper bound: it is simply the test-region R^2 obtained by fitting the same parametric law on the full training set. Since the target region is fully extrapolative and concentrated in the high-cost regime, the best fit on all observed training points need not be the best fit for the target region. Under model misspecification, adding more training points can even hurt extrapolation if those points are concentrated in regions whose trends are less aligned with the high-cost behavior of interest.

To test whether this gain merely comes from avoiding cheap runs, we compare SL^2 with fits using only the cheapest or most expensive half of each candidate pool. Table 4 shows that SL^2 outperforms both diagnostics on average and beats the expensive-half fit on six of eight tasks. The gain therefore comes from target-aware selection, not a simple preference for expensive runs. More broadly, when the goal is to predict much larger configurations, blindly fitting all available pilot runs may be suboptimal.

5.3 Ablation Study

To understand which parts of our acquisition function are responsible for the observed gains, we perform an ablation study by removing the two terms in the MSPE decomposition, ΔV_{inter} and ΔV_{intra} . Here, V_{inter} measures uncertainty induced by disagreement across

Setting	lr&bsz	domain	vocab	parallel	moe	data_con	sparsity	farseeer
1% w/o ΔV_{inter}	-0.70 ± 0.48	0.64 ± 0.58	0.96 ± 0.02	-1.00 ± 0.00	0.55 ± 0.44	0.69 ± 0.39	0.08 ± 0.55	0.60 ± 0.11
1% w/o ΔV_{intra}	-0.78 ± 0.44	0.39 ± 0.76	0.96 ± 0.02	-1.00 ± 0.00	0.43 ± 0.63	-0.01 ± 0.79	-0.07 ± 0.71	-0.41 ± 0.73
1% SL ² (ours)	-0.66 ± 0.53	0.64 ± 0.58	0.96 ± 0.02	-1.00 ± 0.00	0.59 ± 0.39	0.73 ± 0.37	0.31 ± 0.39	0.60 ± 0.11
5% w/o ΔV_{inter}	-0.02 ± 0.60	0.89 ± 0.38	0.97 ± 0.00	0.77 ± 0.39	0.67 ± 0.23	0.83 ± 0.17	0.33 ± 0.30	0.88 ± 0.02
5% w/o ΔV_{intra}	-0.51 ± 0.53	0.86 ± 0.43	0.98 ± 0.00	0.07 ± 0.90	0.34 ± 0.40	0.65 ± 0.47	0.37 ± 0.39	0.84 ± 0.09
5% SL ² (ours)	0.00 ± 0.59	0.89 ± 0.38	0.98 ± 0.00	0.77 ± 0.39	0.65 ± 0.27	0.85 ± 0.16	0.44 ± 0.17	0.88 ± 0.02
10% w/o ΔV_{inter}	0.20 ± 0.55	0.95 ± 0.28	0.98 ± 0.00	0.99 ± 0.01	0.83 ± 0.08	0.85 ± 0.12	0.51 ± 0.13	0.93 ± 0.00
10% w/o ΔV_{intra}	-0.13 ± 0.60	0.91 ± 0.35	0.98 ± 0.00	0.99 ± 0.00	0.71 ± 0.22	0.80 ± 0.25	0.48 ± 0.19	0.89 ± 0.04
10% SL ² (ours)	0.22 ± 0.55	0.95 ± 0.28	0.98 ± 0.00	0.99 ± 0.00	0.83 ± 0.07	0.86 ± 0.11	0.53 ± 0.08	0.93 ± 0.00

Table 5: **Acquisition-function ablation.** We remove either ΔV_{inter} or ΔV_{intra} from the acquisition score and report the target-region R^2 at different budget levels. Each cell shows the mean and standard deviation aggregated over all scaling-law instances. Higher is better. Budgets are 1%, 5%, 10% for most tasks, and 20%, 35%, 50% for domain and sparsity.

different basins, while V_{intra} measures uncertainty within each basin due to local parameter variation around a mode. Ablating them separately allows us to isolate the contributions of cross-basin and within-basin uncertainty to sequential experiment selection.

Ablation results. Table 5 shows that both terms contribute, but not equally. Removing ΔV_{intra} causes the larger and more consistent degradation, especially on `data_con`, `farseeer`, `moe`, and `lr&bsz`, indicating that within-basin uncertainty is the dominant signal for effective sequential design. By contrast, removing ΔV_{inter} usually leads to a smaller drop and in some cases leaves performance nearly unchanged, suggesting that cross-basin disagreement is more task-dependent.

This difference is consistent with the roles of the two terms. ΔV_{intra} remains useful even after the method has identified a plausible parameter region, because it continues to refine uncertainty within that basin. ΔV_{inter} is most helpful when several basins remain plausible and induce different target-region predictions, which occurs more often under tighter budgets and on more heterogeneous tasks such as `sparsity` and `data_con`. Overall, the full method is the most robust across tasks and budget levels, indicating that the two terms are complementary: ΔV_{intra} provides the dominant signal, while ΔV_{inter} yields additional gains when basin ambiguity is substantial.

6 Conclusions

Our work formulates scaling-law fitting as a budget-aware sequential experimental design problem, where each candidate run incurs a cost and the objective is to maximize predictive accuracy in a high-cost target region. We propose SL², an uncertainty-aware acquisition strategy that selects experiments according to their value for target-region extrapolation. Across a diverse benchmark of scaling-law tasks, SL² generally outperforms random, heuristic, and classical design-based baselines, and often approaches full-data performance using only a small fraction of the original training budget. These results demonstrate that treating scaling-law fitting as a problem of experimental design and budget allocation leads to new algorithmic insights.

We also note that SL² may be less reliable under severe model misspecification, particularly when abrupt phase transitions are neither visible in the pilot region nor represented by the chosen law family. Moreover, its one-step acquisition rule does not explicitly optimize long-horizon budget allocation. More robust uncertainty approximations and batched or multi-step budget-aware design are promising directions for future work.

References

- Anthony C Atkinson and David C Woods. Designs for generalized linear models. *Handbook of design and analysis of experiments*, 7:471–514, 2015.
- Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. Broken neural scaling laws, 2023. URL <https://arxiv.org/abs/2210.14891>.
- Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. Are more LLM calls all you need? towards the scaling properties of compound AI systems. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=m5106RRLgx>.
- Mouxiong Chen, Binyuan Hui, Zeyu Cui, Jiayi Yang, Dayiheng Liu, Jianling Sun, Junyang Lin, and Zhongxin Liu. Parallel scaling law for language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL <https://openreview.net/forum?id=dEi1S731lk>.
- Corinna Cortes, L. D. Jackel, Sara Solla, Vladimir Vapnik, and John Denker. Learning curves: Asymptotic values and rate of convergence. In J. Cowan, G. Tesauro, and J. Alspector (eds.), *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann, 1993. URL https://proceedings.neurips.cc/paper_files/paper/1993/file/1aa48fc4880bb0c9b8a3bf979d3b917e-Paper.pdf.
- William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022. URL <http://jmlr.org/papers/v23/21-0998.html>.
- Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10835–10866. PMLR, 23–29 Jul 2023. URL <https://proceedings.mlr.press/v202/gao23h.html>.
- Steven G Gilmour and Luzia A Trinca. Optimum design of experiments for statistical inference. *Quality control and applied statistics*, 58(3):235–236, 2013.
- Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 76232–76264. Curran Associates, Inc., 2024. doi: 10.52202/079017-2427. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/8b970e15a89bf5d12542810df8eae8fc-Paper-Conference.pdf.
- Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. Scaling laws for transfer. *arXiv preprint arXiv:2102.01293*, 2021. URL <https://arxiv.org/abs/2102.01293>.
- 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, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models, 2022. URL <https://arxiv.org/abs/2203.15556>.
- Shengding Hu, Yuge Tu, Xu Han, Ganqu Cui, Chaoqun He, Weilin Zhao, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Xinrong Zhang, Zhen Leng Thai, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, dahai li, Zhiyuan Liu, and Maosong Sun. MiniCPM: Unveiling the

- potential of small language models with scalable training strategies. In *First Conference on Language Modeling*, 2024. URL <https://openreview.net/forum?id=3X2L2TFr0f>.
- Xun Huan and Youssef M. Marzouk. Simulation-based optimal bayesian experimental design for nonlinear systems. *Journal of Computational Physics*, 232(1):288–317, January 2013. ISSN 0021-9991. doi: 10.1016/j.jcp.2012.08.013. URL <http://dx.doi.org/10.1016/j.jcp.2012.08.013>.
- Baihe Huang, Shanda Li, Tianhao Wu, Yiming Yang, Ameet Talwalkar, Kannan Ramchandran, Michael I. Jordan, and Jiantao Jiao. Sample complexity and representation ability of test-time scaling paradigms. In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=RJDIX75TEE>.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Chris Hallacy, Jan Leike, Aditya Ramesh, et al. Scaling laws for neural language models. In *arXiv preprint arXiv:2001.08361*, 2020.
- André I Khuri, Bhramar Mukherjee, Bikas K Sinha, and Malay Ghosh. Design issues for generalized linear models: A review. 2006.
- Jack Kiefer. Optimum experimental designs. *Journal of the Royal Statistical Society: Series B (Methodological)*, 21(2):272–304, 1959.
- Houyi Li, Wenzhen Zheng, Jingcheng Hu, Qiufeng Wang, Hanshan Zhang, Zili Wang, Shijie Xuyang, Yuntao Fan, Shuigeng Zhou, Xiangyu Zhang, et al. Predictable scale: Part i—optimal hyperparameter scaling law in large language model pretraining. *arXiv e-prints*, pp. arXiv–2503, 2025a.
- Houyi Li, Wenzhen Zheng, Qiufeng Wang, Zhenyu Ding, Haoying Wang, Zili Wang, Shijie Xuyang, Ning Ding, Shuigeng Zhou, Xiangyu Zhang, and Daxin Jiang. Predictable scale: Part ii, farseer: A refined scaling law in large language models, 2025b. URL <https://arxiv.org/abs/2506.10972>.
- Margaret Li, Sneha Kudugunta, and Luke Zettlemoyer. (mis)fitting: A survey of scaling laws, 2025c. URL <https://arxiv.org/abs/2502.18969>.
- Seng Pei Liew, Takuya Kato, and Sho Takase. Scaling laws for upcycling mixture-of-experts language models. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=ZBBo19jldX>.
- Haowei Lin, Baizhou Huang, Haotian Ye, Qinyu Chen, Zihao Wang, Sujian Li, Jianzhu Ma, Xiaojun Wan, James Zou, and Yitao Liang. Selecting large language model to fine-tune via rectified scaling law. In *Forty-first International Conference on Machine Learning*, 2024. URL <https://openreview.net/forum?id=Bq2THENXRr>.
- Haowei Lin, Haotian Ye, Wenzheng Feng, Quzhe Huang, Yujun Li, Hubert Lim, Zhengrui Li, Xiangyu Wang, Jianzhu Ma, Yitao Liang, and James Zou. Can language models discover scaling laws? In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=TPtWC0pGk>.
- Jan Ludziejewski, Jakub Krajewski, Kamil Adamczewski, Maciej Pióro, Michał Krutul, Szymon Antoniak, Kamil Ciebia, Krystian Król, Tomasz Odrzygóźdź, Piotr Sankowski, Marek Cygan, and Sebastian Jaszczur. Scaling laws for fine-grained mixture of experts. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 33270–33288. PMLR, 21–27 Jul 2024. URL <https://proceedings.mlr.press/v235/ludziejewski24a.html>.
- Niklas Muennighoff, Alexander Rush, Boaz Barak, Teven Le Scao, Nouamane Tazi, Aleksandra Piktus, Sampo Pyysalo, Thomas Wolf, and Colin A Raffel. Scaling data-constrained language models. In *Advances in Neural Information Processing Systems*, volume 36, pp. 50358–50376, 2023.

- Tomer Porian, Mitchell Wortsman, Jenia Jitsev, Ludwig Schmidt, and Yair Carmon. Resolving discrepancies in compute-optimal scaling of language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=4fSSqpk1sM>.
- Haoran Que, Jiaheng Liu, Ge Zhang, Chenchen Zhang, Xingwei Qu, Yinghao Ma, Feiyu Duan, Zhiqi Bai, Jiakai Wang, Yuanxing Zhang, Xu Tan, Jie Fu, Jiamang Wang, Lin Qu, Wenbo Su, and Bo Zheng. D-CPT law: Domain-specific continual pre-training scaling law for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=JzKFN5fWOk>.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, Proceedings of Machine Learning Research (PMLR), 2021. URL <https://arxiv.org/abs/2103.00020>.
- Ayan Sengupta, Yash Goel, and Tanmoy Chakraborty. How to upscale neural networks with scaling law? *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL <https://openreview.net/forum?id=AL7N0UOfgI>.
- Samuel Silvey. *Optimal design: an introduction to the theory for parameter estimation*. Springer Science & Business Media, 2013.
- Chaofan Tao, Qian Liu, Longxu Dou, Niklas Muennighoff, Zhongwei Wan, Ping Luo, Min Lin, and Ngai Wong. Scaling laws with vocabulary: Larger models deserve larger vocabularies. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=sKCKPr8cRL>.
- Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL <http://jmlr.org/papers/v9/vandermaaten08a.html>.
- Lynda V White. An extension of the general equivalence theorem to nonlinear models. *Biometrika*, 60(2):345–348, 1973.
- Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=VNckp7JEHn>.
- Ruihan Xu, Jiajin Li, and Yiping Lu. On the width scaling of neural optimizers under matrix operator norms i: Row/column normalization and hyperparameter transfer. *arXiv preprint arXiv:2603.09952*, 2026.
- Ge Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 17084–17097. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/8df7c2e3c3c3be098ef7b382bd2c37ba-Paper.pdf.
- Min Yang, Stefanie Biedermann, and Elina Tang. On optimal designs for nonlinear models: a general and efficient algorithm. *Journal of the American Statistical Association*, 108(504):1411–1420, 2013.
- Jiasheng Ye, Peiju Liu, Tianxiang Sun, Jun Zhan, Yunhua Zhou, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=jjCB27TMK3>.

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 (CVPR)*, pp. 12104–12113, June 2022.

Shijie Zhong, Wanggang Shen, Tommie Catanach, and Xun Huan. Goal-oriented bayesian optimal experimental design for nonlinear models using markov chain monte carlo. *SIAM/ASA Journal on Uncertainty Quantification*, 14(1):19–47, 2026. doi: 10.1137/24M1649344. URL <https://doi.org/10.1137/24M1649344>.

A Use of LLMs

We employ large language models (LLMs) exclusively for the purpose of assisting in the drafting and refinement of our manuscripts, with the objective of enhancing clarity and coherence.

B Detailed Statistics of the Benchmark

B.1 Task Collection Details

Our experimental benchmark covers a diverse set of practical LLM scaling scenarios, including pre-training hyperparameter tuning, data allocation, architecture design, and inference-time scaling. The first six tasks are drawn from SLDBench (Lin et al., 2026), and the remaining two from the original papers; we also refer to surveys and related literature for task coverage and context (Sengupta et al., 2025). For the six SLDBench datasets, we use the top-10 law forms on the leaderboard ranked by mean R^2 . We exclude *Supervised Finetuning Scaling Law* due to its very limited data and *U-shaped Scaling Law* due to severe misspecification that yields poor fits even on the full training set.

The collected tasks are: (1) *Parallel Scaling Law* (parallel), which studies the effect of parallelism P and model size N on language modeling loss; this setting creates P augmentations of an input and aggregates their outputs, conceptually similar to Best-of- N (Chen et al., 2025). (2) *Vocabulary Scaling Law* (vocab), which models unigram-normalized loss as a function of non-vocabulary model size N , vocabulary size V , and dataset size D (Tao et al., 2024). (3) *Domain Mixture Scaling Law* (domain), which models domain-specific pre-training loss as a function of the mixture proportions of training domains (Ye et al., 2025). (4) *Mixture of Experts Scaling Law* (moe), which relates loss to the number of dense parameters N and experts E (Ludziejewski et al., 2024). (5) *Data Constrained Scaling Law* (data_con), which models pre-training loss using network size N , dataset size D , and the number of unique tokens U . (6) *Learning Rate and Batch Size Scaling Law* (lr&bsz), adapted from the Step Law (Li et al., 2025a), which models pre-training loss as a function of learning rate l , batch size b , dataset size D , and network size N . (7) *Sparsity Scaling Law* (sparsity), which models test loss based on the ratio P between total model size N_1 and active parameters N_2 (Liew et al., 2025). (8) *Farseer Scaling Law* (farseer), which extends the Chinchilla-style formulation (Hoffmann et al., 2022) to predict loss from model size N and training data D (Li et al., 2025b).

B.2 Parametric Forms of Collected Scaling Laws

This appendix summarizes the parametric scaling-law families used in our benchmark. Our goal is not to advocate a single canonical law form, but to evaluate budget-aware sequential design across a diverse set of realistic fitting regimes. The collected laws therefore span classical power-law formulations, log-space interaction models, compositional mixture laws, hyperparameter response surfaces, and several more expressive nonlinear forms drawn from prior scaling-law studies. For the six tasks inherited from SLDBench, we use the top-ranked law families on the public leaderboard by mean R^2 ; for the remaining tasks, we adopt the parametric forms proposed in the corresponding original papers. Together, these forms define the nonlinear fitting landscapes on which all acquisition methods are evaluated.

Table 6: **Collected scaling laws.** Scaling laws grouped by task.

ID	Parametric Form	# Params	All Data test R^2
data_con			
sl_1	$L(N, D, U) = \frac{A}{N^\alpha} + \frac{B}{D^\beta} + E U^\gamma N^\delta$	7	0.94 ± 0.00

ID	Parametric Form	# Params	All Data test R^2
sl.2	$L(N, D, U) = a + bU^p + cN^q + dD^r$	7	0.73 ± 0.11
sl.3	$L(N, D, U) = \frac{A}{N_{\text{eff}}^\alpha} + \frac{B}{D_{\text{eff}}^\beta} + C, \quad U_N = \min(\rho U, N), \quad R_N = \max(N/U_N - 1, 0), \quad N_{\text{eff}} = U_N + \tau_N U_N (1 - e^{-R_N/\tau_N}), \quad D_{\text{eff}} = U + \tau_D U (1 - e^{-(D/U-1)/\tau_D})$	7	0.89 ± 0.00
sl.4	$L(N, D, U) = L_0 + AM_n^{-a} + BT_{\text{eff},n}^{-b}, \quad q = \frac{T_n}{sU_n M_n^d}, \quad T_{\text{eff},n} = \frac{T_n}{1+q}$	7	0.95 ± 0.01
sl.5	$L(N, D, U) = \frac{A}{N^\alpha} + \frac{B}{D^\beta} + E, \quad D_{\text{eff}} = U^\gamma D^{1-\gamma}$	6	0.84 ± 0.00
sl.6	$L(N, D, U) = E + AN^{-\alpha} + BD_{\text{eff}}^{-\beta}, \quad D_{\text{eff}} = \frac{D}{1+C \max(D/U-1, 0)^c N^d}$	8	0.88 ± 0.08
sl.7	$L(N, D, U) = L_0 + A(NU)^{\alpha_{pu}} + BD^{\alpha_t} + CN^{\alpha_p}$	7	0.94 ± 0.03
sl.8	$L(N, D, U) = a + \frac{b}{D^\alpha} + \frac{c}{N^\beta} + d \left \log\left(\frac{U}{D} + 1\right) \right ^\gamma$	7	-0.07 ± 0.19
sl.9	$L(N, D, U) = \frac{A}{N^\alpha} + \frac{B}{D^\beta} \left(1 + \frac{C}{U^\gamma}\right) + L_{\text{inf}}$	7	0.84 ± 0.08
sl.10	$L(N, D, U) = L_0 + AN^{-a} + B \left(D^{-bq} + (kU)^{-bq}\right)^{1/q}$	7	0.81 ± 0.03
domain			
sl.1	$L_i(\mathbf{r}) = a_i + b_i \log(r_i + \varepsilon) + \sum_{j \neq i} c_{ij} r_j$	30	1.00 ± 0.00
sl.2	$L_i(\mathbf{r}) = A_i (r_i + \varepsilon_i)^{-\alpha_i} \exp\left(\sum_{j \neq i} w_{ij} r_j\right)$	35	1.00 ± 0.00
sl.3	$L_i(\mathbf{r}) = \text{base}_i + \text{coeff}_i r_i^{\text{exp}_i} + \sum_{j \neq i} W_{ij} r_j$	35	1.00 ± 0.00
sl.4	$L_i(\mathbf{r}) = \exp\left(\sum_k C_{ik} r_k^{\alpha_k} + \text{bias}_i\right)$	35	1.00 ± 0.00
sl.5	$L_i(\mathbf{r}) = b_i + \sum_j W_{ij} r_j^{\frac{1}{\alpha_j}}$	35	1.00 ± 0.00
sl.6	$L_i(\mathbf{r}) = C_i + A_i \left(\sum_j T_{ij} r_j\right)^{-\alpha_i}$	35	-0.81 ± 0.58
sl.7	$L_i(\mathbf{r}) = \text{intercept}_i + \sum_j \left(c_{ij}^{\text{lin}} r_j + c_{ij}^{\text{log}} \log(r_j + \varepsilon)\right)$	40	1.00 ± 0.00
sl.8	$L_i(\mathbf{r}) = c_i - a_i r_i^{b_i}$	15	1.00 ± 0.00
sl.9	$L_i(\mathbf{r}) = a_i + b_i \log(r_i + \varepsilon) + c_i [\log(r_i + \varepsilon)]^2$	15	1.00 ± 0.00
sl.10	$L_i(\mathbf{r}) = a_i + \frac{b_i}{r_i + \varepsilon_i}$	15	1.00 ± 0.00
farseer			
sl.1	$L(N, D) = e^{sN^q + S} + e^{BN^b + Q} D^{-e^{AN^q + E}}$	9	0.92 ± 0.01
lr&bsz			
sl.1	$L(l, b, N, D) = \exp(\text{poly}_2(\log l, \log b, \log D, \log N))$	15	0.61 ± 0.19
sl.2	$L(l, b, N, D) = L_{\text{inf}} + C_p e^{-a_p n} + C_d e^{-a_d s} + C_{dp} e^{-a_{dp}(s-kn)} + C_{bb} e^{-a_{bb} v} + c_L \Delta u^2 + c_B \Delta v^2 + 2\rho \sqrt{c_L c_B} \Delta u \Delta v, \quad n = \log N, \quad s = \log D, \quad u = \log l, \quad v = \log b$	26	-0.15 ± 0.30
sl.3	$L(l, b, N, D) = E + AN^{-\alpha} + BD^{-\beta} + \frac{F}{N^w N^D w_D} + C_{\text{eff}}(\log l - \text{opt}_l)^2 + G_{\text{eff}}(\log b - \text{opt}_b)^2$	24	-0.99 ± 0.04

ID	Parametric Form	# Params	All Data test R^2
sl.4	$L(l, b, N, D) = \exp(w_0 + w_1 \log l + w_2 \log b + w_3 \log D + w_4 \log N + w_5 \log^2 l + w_6 \log^2 b + w_7 \log^2 D + w_8 \log^2 N + w_9 \log l \log b + w_{10} \log l \log D + w_{11} \log l \log N + w_{12} \log b \log D + w_{13} \log b \log N + w_{14} \log D \log N + w_{15} (\log D - \log N) + w_{16}/b + w_{17}/b^2 + w_{18}/D + w_{19}/N)$	20	-0.88 ± 0.36
sl.5	$L(l, b, N, D) = L_0 + A_N e^{-a_N n} + A_D e^{-a_D s} + A_B e^{-a_B v} + A_R e^{-a_R (s-n)^2} + A_X e^{-a_X (s-v)} + c_{lr,0} e^{-w_b v - w_n n - w_s s} (u - u_*)^2, \quad u = \log l, v = \log b, s = \log D, n = \log N$	19	-0.10 ± 0.66
sl.6	$L(l, b, N, D) = L_{\inf} + \exp(w_0 + w_d \log D + w_p \log N + w_{dp} \log D \log N + w_{lr} \log l + w_{lr^2} \log^2 l + w_{bsz} \log b + w_{bsz^2} \log^2 b + w_{lr,bsz} \log l \log b + w_{lr,D} \log l \log D + w_{lr,N} \log l \log N + w_{bsz,D} \log b \log D + w_{bsz,N} \log b \log N)$	14	0.47 ± 0.14
sl.7	$L(l, b, N, D) = E + \exp(w_{1,0} + w_{1,1} x_1 + w_{1,2} x_2 + w_{1,3} x_3 + w_{1,4} x_4 + w_{1,5} x_1^2 + w_{1,6} x_2^2 + w_{1,7} x_3^2 + w_{1,8} x_4^2 + w_{1,9} x_1 x_2 + w_{1,10} x_1 x_3 + w_{1,11} x_1 x_4 + w_{1,12} x_2 x_3 + w_{1,13} x_2 x_4 + w_{1,14} x_3 x_4) + \exp(w_{2,0} + w_{2,1} x_1 + w_{2,2} x_2 + w_{2,3} x_3 + w_{2,4} x_4 + w_{2,5} x_1^2 + w_{2,6} x_2^2 + w_{2,7} x_3^2 + w_{2,8} x_4^2 + w_{2,9} x_1 x_2 + w_{2,10} x_1 x_3 + w_{2,11} x_1 x_4 + w_{2,12} x_2 x_3 + w_{2,13} x_2 x_4 + w_{2,14} x_3 x_4), \quad x_1 = \log l, x_2 = \log b, x_3 = \log D, x_4 = \log N$	31	0.75 ± 0.19
sl.8	$L(l, b, N, D) = L_0 + c_P e^{-a_P n} + c_D e^{-a_D s} + c_R e^{-a_R (s-n)} + k_{lr} \delta_{lr}^2 (1 + a_{lr} \tanh \delta_{lr}) + k_{ns} \delta_{ns}^2 (1 + a_{ns} \tanh \delta_{ns}) + k_{dp} ((s - n) - \delta_0)^2, \quad n = \log N, s = \log D$	20	0.49 ± 0.33
sl.9	$L(l, b, N, D) = \text{poly}_2(\log_{10} l, \log_{10} b, \log_{10} D, \log_{10} N)$	15	0.36 ± 0.09
sl.10	$L(l, b, N, D) = \text{poly}_2(\log l, \log b, \log D, \log N) + w_D D^{-1/2} + w_N N^{-1/2} + w_b b^{-1}$	18	-0.07 ± 0.64
moe			
sl.1	$L(N, E) = L_{\inf} + \frac{B}{N^\alpha E^\beta}$	4	0.83 ± 0.00
sl.2	$L(N, E) = L + K(N^\alpha E^\beta)^{-\gamma}$	5	0.83 ± 0.00
sl.3	$L(N, E) = \frac{AN^\alpha}{1+BE^\beta} + CN^{0.6\alpha} + D$	6	0.90 ± 0.00
sl.4	$L(N, E) = \frac{a}{N^\alpha(1+bE)^\gamma} + c + d(\log N - 0.4 \log(1 + E))$	6	0.83 ± 0.00
sl.5	$L(N, E) = p_0 + \exp(p_1 + p_2 \log E + p_3 \log N + p_4 \log E \log N) + p_5 \log E$	6	0.80 ± 0.00
sl.6	$L(N, E) = aN^{-b}(1 + cE^{-d}) + e + \frac{f}{EN^{0.05}}$	6	0.83 ± 0.00
sl.7	$L(N, E) = p_0 E^{p_1} N^{p_2} + p_3 N^{p_4} + p_5$	6	0.74 ± 0.00
sl.8	$L(N, E) = aN^b E^c + d$	4	0.83 ± 0.00
sl.9	$L(N, E) = c_0 + A(NE^g)^{-a}$	4	0.83 ± 0.00
sl.10	$L(N, E) = \text{bias} + A(N/10^9)^{-\alpha} \left(\frac{1+BE^\gamma}{1+B} \right)^{-\beta}$	6	0.77 ± 0.00
parallel			
sl.1	$L(N, P) = c_0 + c_N N^{-\alpha} + c_P P^{-\beta} + c_{NP} N^{-\alpha} P^{-\beta}$	6	1.00 ± 0.00
sl.2	$L(N, P) = c_0 + c_N N^{-\alpha} + c_P P^{-\beta}$	5	1.00 ± 0.00

ID	Parametric Form	# Params	All Data test R^2
s1.3	$L(N, P) = aN^b + \frac{c}{1+P} + d$	4	1.00 ± 0.00
s1.4	$L(N, P) = aN^b + cP^{-1/2} + d$	4	1.00 ± 0.00
s1.5	$L(N, P) = \left(\frac{A}{N(k \log P + 1)} \right)^\alpha + E$	4	1.00 ± 0.00
s1.6	$L(N, P) = c_0 + c_1(N^{-\alpha} + P^{-\beta})$	4	1.00 ± 0.00
s1.7	$L(N, P) = L_0 + \frac{AN^{-\alpha}}{1+k \ln P}$	4	1.00 ± 0.00
s1.8	$L(N, P) = \frac{aN^b + c}{1+d \log P}$	4	1.00 ± 0.00
s1.9	$L(N, P) = AN^{-\alpha} P^{-\beta}$	3	1.00 ± 0.00
s1.10	$L(N, P) = (AN^{-\alpha} + E)P^{-\beta}$	4	1.00 ± 0.00
sparsity			
s1.1	$L(P, N_2) = e^{d_1} P^{-a} N_2^{-b} e^{c \log P \log N_2} + e^{d_3}$	5	0.28 ± 0.00
s1.2	$L(P, N_2) = e^{d_1} P^{-a} N_2^{-b} + e^{d_3}$	4	0.52 ± 0.00
s1.3	$L(P, N_2) = e^{d_1} P^{-a} + e^{d_2} N_2^{-b} e^{c \log P \log N_2} + e^{d_3}$	6	0.27 ± 0.00
s1.4	$L(P, N_2) = e^{d_1} P^{-a} + e^{d_2} N_2^{-b} + e^{d_3}$	5	0.40 ± 0.00
vocab			
s1.1	$L(N, V, D) = c_0 + AV^b N^e D^g$	5	0.89 ± 0.30
s1.2	$L(N, V, D) = L + A M_r(N^{-\alpha}, D^{-\beta})(1 + C(\log V - v_0)^2)$	7	0.99 ± 0.00
s1.3	$L(N, V, D) = L_0 + \left((aN^{-\alpha})^q + (b(DV^\phi)^{-\beta})^q \right)^{1/q}$	7	0.99 ± 0.00
s1.4	$L(N, V, D) = L_{\inf} + A \max(N^a, \lambda D^b)^{-d} V^{-g}$	7	0.83 ± 0.19
s1.5	$L(N, V, D) = p_0 N^{p_1} V^{p_2} D^{p_3} + p_4 N^{p_5} + p_6$	7	0.99 ± 0.00
s1.6	$L(N, V, D) = A(NV^{k_1})^{-\alpha} + B(DV^{k_2})^{-\beta} + c_0$	7	0.77 ± 0.31
s1.7	$L(N, V, D) = AN^{-\alpha} D^{-\beta} + BV^\gamma D^{-\delta} + c_0$	7	0.98 ± 0.02
s1.8	$L(N, V, D) = c_0 + c_1 \log V + V^\beta (c_2 N^{-\alpha} + c_3 D^{-\gamma})$	7	0.97 ± 0.04
s1.9	$L(N, V, D) = AN^{-\alpha} D^{-\beta} (1 + \gamma \log V) + \delta V^\epsilon + L_{\inf}$	7	0.98 ± 0.00
s1.10	$L(N, V, D) = L_{\min} + \exp(a + b_p \log N + b_{V1} \log V + b_{V2} \log^2 V + b_D \log D + b_{VD} \log V \log D)$	7	0.98 ± 0.00

C Basin Estimation and Posterior Approximation

We assume access to a set of locally optimal parameter vectors obtained by refitting the scaling law from multiple initializations on the current dataset $\mathcal{D}_t$. This appendix explains how these local solutions are converted into the basin approximation

$$p(\theta \mid \mathcal{D}_t) \approx \sum_{k=1}^K w_k \mathcal{N}(\theta_k, \Sigma_k).$$

C.1 Local covariance approximation

We assume access to a collection of locally optimal parameter vectors

$$\{\tilde{\theta}_m\}_{m=1}^M$$

obtained by refitting the scaling law from multiple initializations on the current dataset $\mathcal{D}_t$. This subsection explains how each local solution is converted into a local Gaussian approximation, which will later be consolidated into the basin mixture

$$p(\theta \mid \mathcal{D}_t) \approx \sum_{k=1}^K w_k \mathcal{N}(\theta_k, \Sigma_k).$$

For a local optimum $\tilde{\theta}_m$, let

$$\mathcal{L}_t(\theta) = \frac{1}{|\mathcal{D}_t|} \sum_{(x_i, y_i) \in \mathcal{D}_t} \|f(x_i; \theta) - y_i\|_2^2$$

denote the empirical mean-squared error on the currently observed data. Around $\tilde{\theta}_m$, we approximate the local curvature of this objective using a Gauss–Newton/Fisher-style matrix. Concretely, let

$$J_m = J(\tilde{\theta}_m; \mathcal{D}_t)$$

be the Jacobian of model predictions with respect to the parameters, evaluated on all observations in $\mathcal{D}_t$ and flattened across data points (and output dimensions, when applicable). Given the current noise variance estimate σ^2 , we define

$$H_m = \frac{1}{\sigma^2} J_m^\top J_m + \Lambda_{\text{prior}},$$

where Λ_{prior} is a diagonal prior-precision matrix used to stabilize weakly identified directions. We then approximate the local parameter uncertainty around $\tilde{\theta}_m$ by

$$\tilde{\Sigma}_m = H_m^{-1}.$$

This yields the local Gaussian approximation

$$q_m(\theta) = \mathcal{N}(\tilde{\theta}_m, \tilde{\Sigma}_m)$$

for each candidate local optimum.

For parameters constrained to be positive, we additionally rescale the corresponding Jacobian columns by the current parameter values before forming H_m . Equivalently, this amounts to measuring local sensitivity in a log-parameterization for those coordinates, which improves numerical stability when different parameters operate on very different scales. In implementation, the inversion of H_m is further stabilized by standard spectral regularization, including eigenvalue flooring and small diagonal correction when necessary.

At this stage, each local refit $\tilde{\theta}_m$ is associated with a Gaussian approximation $q_m(\theta)$. These candidate components are not yet the final basins in the mixture posterior, because several local optima may induce nearly identical predictive behavior. The next subsection therefore clusters these local solutions in prediction space and consolidates them into a smaller set of representative basins.

C.2 Prediction-space clustering of local optima

The local Gaussian approximations constructed above are still over-complete: different local optima may correspond to essentially the same extrapolative behavior, even when their parameter values differ substantially. This is particularly common for nonlinear scaling laws with weakly identifiable directions. Since our downstream objective is prediction accuracy on a target region rather than parameter recovery itself, we consolidate local optima in prediction space rather than in parameter space.

For each local solution $\tilde{\theta}_m$ with covariance $\tilde{\Sigma}_m$, we evaluate its induced predictive distribution on an evaluation set $\mathcal{X}_{\text{eval}}$. In our implementation, $\mathcal{X}_{\text{eval}}$ is chosen to be the target region used by the acquisition function. Under the local linear approximation, the predictive distribution at a point $x \in \mathcal{X}_{\text{eval}}$ is

$$f(x; \theta) \mid \theta \sim q_m \approx \mathcal{N}(\mu_m(x), v_m(x)),$$

where

$$\mu_m(x) = f(x; \tilde{\theta}_m), \quad v_m(x) = J_x(\tilde{\theta}_m) \tilde{\Sigma}_m J_x(\tilde{\theta}_m)^\top + \sigma^2,$$

and $J_x(\tilde{\theta}_m) = \left. \frac{\partial f(x; \theta)}{\partial \theta} \right|_{\theta=\tilde{\theta}_m}$ is the parameter Jacobian at x .

We then measure the discrepancy between two local optima $\tilde{\theta}_m$ and $\tilde{\theta}_n$ by comparing their predictive Gaussians across $\mathcal{X}_{\text{eval}}$. For scalar outputs, the pointwise symmetric KL divergence between

$$\mathcal{N}(\mu_m(x), v_m(x)) \quad \text{and} \quad \mathcal{N}(\mu_n(x), v_n(x))$$

is

$$\text{SKL}_{m,n}(x) = \frac{1}{4} \left(\frac{v_m(x)}{v_n(x)} + \frac{v_n(x)}{v_m(x)} - 2 + (\mu_m(x) - \mu_n(x))^2 \left(\frac{1}{v_m(x)} + \frac{1}{v_n(x)} \right) \right).$$

Averaging over the evaluation region gives the prediction-space dissimilarity

$$d_{mn} = \frac{1}{|\mathcal{X}_{\text{eval}}|} \sum_{x \in \mathcal{X}_{\text{eval}}} \text{SKL}_{m,n}(x).$$

Using the pairwise dissimilarity matrix $\{d_{mn}\}$, we perform agglomerative hierarchical clustering to group local optima with similar predictive behavior. The clustering threshold is selected data-adaptively by maximizing the silhouette score over candidate cuts of the hierarchy. This procedure yields a partition of the local optima into clusters

$$\mathcal{C}_1, \dots, \mathcal{C}_K,$$

which we interpret as the final basins.

For each cluster $\mathcal{C}_k$, we choose a single representative local optimum as the basin center. Specifically, we select the member with the smallest empirical fitting error on the current dataset:

$$m_k = \arg \min_{m \in \mathcal{C}_k} \mathcal{L}_t(\tilde{\theta}_m), \quad \theta_k = \tilde{\theta}_{m_k}.$$

Its associated covariance approximation is inherited from the same representative:

$$\Sigma_k = \tilde{\Sigma}_{m_k}.$$

The output of this step is therefore a reduced collection of representative basins

$$\{(\theta_k, \Sigma_k)\}_{k=1}^K,$$

which preserves distinct extrapolative behaviors while removing redundant local optima. The remaining ingredient is to assign mixture weights to these representative basins; this is described in the next subsection.

C.3 Representative basins and mixture weights

After clustering the local optima in prediction space, we obtain a reduced set of representative basins

$$\{(\theta_k, \Sigma_k)\}_{k=1}^K,$$

where each representative parameter θ_k is selected from one prediction-space cluster, and Σ_k is the corresponding local covariance approximation inherited from that representative.

To complete the mixture approximation, we associate each basin with a weight

$$w_k \approx p(B = k \mid \mathcal{D}_t),$$

where $B \in \{1, \dots, K\}$ is a latent basin indicator specifying which basin generated the current local posterior approximation. Thus, w_k represents the posterior probability that basin k is the relevant mode given the observations collected so far. In practice, exact computation of $p(B = k \mid \mathcal{D}_t)$ is generally intractable, so we approximate it using a basin-level evidence score. We consider two natural choices.

Option 1: BIC-style approximation. A simple approximation is to rank basins by an information criterion derived from their empirical fit on the current dataset. Let

$$\text{MSE}_k = \mathcal{L}_t(\theta_k) = \frac{1}{|\mathcal{D}_t|} \sum_{(x_i, y_i) \in \mathcal{D}_t} \|f(x_i; \theta_k) - y_i\|_2^2$$

and let $n_{\text{obs}} = |\mathcal{D}_t|$. We define

$$\text{BIC}_k = n_{\text{obs}} \log(\text{MSE}_k) + p \log n_{\text{obs}},$$

where p is the number of free parameters. The basin probabilities are then approximated by

$$w_k = \frac{\exp\left(-\frac{\text{BIC}_k}{2T}\right)}{\sum_{\ell=1}^K \exp\left(-\frac{\text{BIC}_\ell}{2T}\right)},$$

where $T > 0$ is a temperature parameter.

Option 2: Laplace-approximate basin posterior. A more direct approximation is obtained by locally approximating the contribution of each basin to the posterior normalizing constant. Let H_k denote the local curvature matrix at θ_k , with $\Sigma_k \approx H_k^{-1}$. Then, up to a common normalization constant,

$$p(B = k \mid \mathcal{D}_t) \propto \pi(\theta_k) \exp\left(-\frac{|\mathcal{D}_t|}{2\sigma^2} \mathcal{L}_t(\theta_k)\right) |H_k|^{-1/2},$$

which yields the normalized weight

$$w_k = \frac{\pi(\theta_k) \exp\left(-\frac{|\mathcal{D}_t|}{2\sigma^2} \mathcal{L}_t(\theta_k)\right) |H_k|^{-1/2}}{\sum_{\ell=1}^K \pi(\theta_\ell) \exp\left(-\frac{|\mathcal{D}_t|}{2\sigma^2} \mathcal{L}_t(\theta_\ell)\right) |H_\ell|^{-1/2}}.$$

In our experiments, we use the BIC-style approximation for robustness, while the Laplace form provides a more principled local-evidence interpretation of the same quantity.

D Derivation of the Acquisition Function

This appendix derives the target-aware acquisition function used in the main text. Starting from the basin-mixture approximation

$$p(\theta \mid \mathcal{D}_t) \approx \sum_{k=1}^K w_k \mathcal{N}(\theta_k, \Sigma_k),$$

we show how the target-region uncertainty measure

$$\text{MSPE}_{\text{tar}} = \frac{1}{|\mathcal{X}_{\text{tar}}|} \mathbb{E}_{\theta \sim p(\theta \mid \mathcal{D}_t)} [\|F(\theta) - \tilde{f}\|_2^2]$$

decomposes into intra-basin and inter-basin terms, and how this decomposition leads to the candidate utility

$$\Delta \text{MSPE}_{\text{tar}}(x) = \Delta V_{\text{intra}}(x) + \Delta V_{\text{inter}}(x).$$

Throughout, the key approximation is local: within each basin, we linearize the predictor around its representative parameter θ_k , while retaining the multimodal mixture structure across basins.

D.1 Local linearization within each basin

Fix a basin k with local posterior approximation

$$\theta \mid (B = k, \mathcal{D}_t) \approx \mathcal{N}(\theta_k, \Sigma_k).$$

To obtain tractable expressions for predictive uncertainty and posterior updates after adding a new observation, we linearize the scaling law around the basin representative θ_k .

For the target region $\mathcal{X}_{\text{tar}}$, recall the prediction map

$$F(\theta) = (f(x; \theta))_{x \in \mathcal{X}_{\text{tar}}} \in \mathbb{R}^{|\mathcal{X}_{\text{tar}}|}.$$

A first-order Taylor expansion around θ_k gives

$$F(\theta) \approx F(\theta_k) + J_k(\theta - \theta_k),$$

where

$$J_k = \left. \frac{\partial F(\theta)}{\partial \theta} \right|_{\theta=\theta_k} \in \mathbb{R}^{|\mathcal{X}_{\text{tar}}| \times p}.$$

Writing

$$\hat{f}_k = F(\theta_k),$$

we obtain the Gaussian approximation

$$F(\theta) \mid (B = k, \mathcal{D}_t) \approx \mathcal{N}(\hat{f}_k, J_k \Sigma_k J_k^\top).$$

Similarly, for a candidate experiment $x \in \mathcal{X}_{\text{cand}}$, define the scalar predictive mean

$$m_k(x) = f(x; \theta_k)$$

and the parameter Jacobian

$$j_k(x) = \left. \frac{\partial f(x; \theta)}{\partial \theta} \right|_{\theta=\theta_k} \in \mathbb{R}^p.$$

Then the same first-order expansion yields

$$f(x; \theta) \approx m_k(x) + j_k(x)^\top (\theta - \theta_k),$$

so under the local Gaussian approximation,

$$y \mid (x, B = k, \mathcal{D}_t) \approx \mathcal{N}(m_k(x), s_k^2(x)), \quad s_k^2(x) = \sigma^2 + j_k(x)^\top \Sigma_k j_k(x).$$

The corresponding posterior update within basin k remains Gaussian. After observing (x, y) , the updated covariance is given by the standard rank-one linear-Gaussian update

$$\Sigma_k^+(x) = \Sigma_k - \frac{\Sigma_k j_k(x) j_k(x)^\top \Sigma_k}{s_k^2(x)},$$

and the posterior mean of the target-region prediction vector updates as

$$\hat{f}_k^+(x, y) = \hat{f}_k + g_k(x)(y - m_k(x)), \quad g_k(x) = \frac{J_k \Sigma_k j_k(x)}{s_k^2(x)} \in \mathbb{R}^{|\mathcal{X}_{\text{tar}}|}.$$

These local update formulas are the basic ingredients for the derivations below: $\Sigma_k^+(x)$ determines the reduction in within-basin predictive variance, while $\hat{f}_k^+(x, y)$ determines how a new observation changes the disagreement between basins.

D.2 Decomposition of target-region MSPE

We now derive the decomposition of the target-region uncertainty objective used in the main text. Recall that

$$\text{MSPE}_{\text{tar}} = \frac{1}{|\mathcal{X}_{\text{tar}}|} \mathbb{E}_{\theta \sim p(\theta | \mathcal{D}_t)} [\|F(\theta) - \bar{f}\|_2^2], \quad \bar{f} = \mathbb{E}_{\theta \sim p(\theta | \mathcal{D}_t)} [F(\theta)].$$

Introducing the latent basin indicator $B \in \{1, \dots, K\}$, the mixture approximation can be written as

$$p(\theta | \mathcal{D}_t) \approx \sum_{k=1}^K p(B = k | \mathcal{D}_t) p(\theta | B = k, \mathcal{D}_t),$$

with

$$p(B = k | \mathcal{D}_t) \approx w_k, \quad p(\theta | B = k, \mathcal{D}_t) \approx \mathcal{N}(\theta_k, \Sigma_k).$$

Under the local linearization from the previous subsection,

$$F(\theta) | (B = k, \mathcal{D}_t) \approx \mathcal{N}(\hat{f}_k, J_k \Sigma_k J_k^\top), \quad \hat{f}_k = F(\theta_k).$$

The target-region MSPE is simply the total variance of the random vector $F(\theta)$, normalized by $|\mathcal{X}_{\text{tar}}|$. Applying the law of total variance with respect to the basin indicator B gives

$$\text{MSPE}_{\text{tar}} = \frac{1}{|\mathcal{X}_{\text{tar}}|} (\mathbb{E}_B [\text{tr}(\text{Cov}(F(\theta) | B, \mathcal{D}_t))] + \text{tr}(\text{Cov}(\mathbb{E}[F(\theta) | B, \mathcal{D}_t]))).$$

We evaluate the two terms separately.

For the first term, conditioning on basin k yields

$$\text{Cov}(F(\theta) | B = k, \mathcal{D}_t) \approx J_k \Sigma_k J_k^\top,$$

so

$$\mathbb{E}_B [\text{tr}(\text{Cov}(F(\theta) | B, \mathcal{D}_t))] \approx \sum_{k=1}^K w_k \text{tr}(J_k \Sigma_k J_k^\top).$$

For the second term, the conditional mean is

$$\mathbb{E}[F(\theta) | B = k, \mathcal{D}_t] \approx \hat{f}_k,$$

and the global posterior mean is therefore

$$\bar{f} = \mathbb{E}[F(\theta) | \mathcal{D}_t] \approx \sum_{k=1}^K w_k \hat{f}_k.$$

Hence,

$$\text{tr}(\text{Cov}(\mathbb{E}[F(\theta) | B, \mathcal{D}_t])) = \sum_{k=1}^K w_k \|\hat{f}_k - \bar{f}\|_2^2.$$

Combining the two terms yields

$$\text{MSPE}_{\text{tar}} = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K w_k \text{tr}(J_k \Sigma_k J_k^\top) + \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K w_k \|\hat{f}_k - \bar{f}\|_2^2.$$

We therefore define

$$V_{\text{intra}} = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K w_k \text{tr}(J_k \Sigma_k J_k^\top), \quad V_{\text{inter}} = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K w_k \|\hat{f}_k - \bar{f}\|_2^2,$$

so that

$$\text{MSPE}_{\text{tar}} = V_{\text{intra}} + V_{\text{inter}}.$$

The term V_{intra} measures the average predictive variance that remains within each basin after conditioning on which local mode is correct, while V_{inter} measures the residual disagreement between basin-level extrapolations. This decomposition is the basis for our acquisition function: a useful new experiment should either reduce uncertainty within a plausible basin or help distinguish between basins that extrapolate differently.

D.3 Derivation of the intra-basin utility

We now derive the reduction in the within-basin term

$$V_{\text{intra}} = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K w_k \text{tr}(J_k \Sigma_k J_k^\top)$$

after querying a candidate experiment x .

After observing an outcome y at x , both the basin posterior probabilities and the within-basin covariances are updated. The exact updated intra-basin term is therefore

$$V_{\text{intra}}^+(x, y) = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K w_k^+(x, y) \text{tr}(J_k \Sigma_k^+(x) J_k^\top),$$

where

$$w_k^+(x, y) = p(B = k \mid x, y, \mathcal{D}_t)$$

and

$$\Sigma_k^+(x) = \Sigma_k - \frac{\Sigma_k j_k(x) j_k(x)^\top \Sigma_k}{s_k^2(x)}, \quad s_k^2(x) = \sigma^2 + j_k(x)^\top \Sigma_k j_k(x).$$

Note that $\Sigma_k^+(x)$ depends on the queried location x but not on the realized value y .

The intra-basin utility is defined by

$$\Delta V_{\text{intra}}(x) = V_{\text{intra}} - \mathbb{E}_{y|x, \mathcal{D}_t} [V_{\text{intra}}^+(x, y)].$$

Substituting the expression above gives

$$\mathbb{E}_{y|x, \mathcal{D}_t} [V_{\text{intra}}^+(x, y)] = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K \mathbb{E}_{y|x, \mathcal{D}_t} [w_k^+(x, y)] \text{tr}(J_k \Sigma_k^+(x) J_k^\top).$$

It remains to evaluate the expectation of the updated basin weights.

By Bayes' rule,

$$w_k^+(x, y) = \frac{w_k p(y \mid x, B = k, \mathcal{D}_t)}{p(y \mid x, \mathcal{D}_t)}.$$

Therefore,

$$\begin{aligned} \mathbb{E}_{y|x, \mathcal{D}_t} [w_k^+(x, y)] &= \int \frac{w_k p(y \mid x, B = k, \mathcal{D}_t)}{p(y \mid x, \mathcal{D}_t)} p(y \mid x, \mathcal{D}_t) dy \\ &= w_k \int p(y \mid x, B = k, \mathcal{D}_t) dy = w_k. \end{aligned}$$

Hence,

$$\mathbb{E}_{y|x, \mathcal{D}_t} [V_{\text{intra}}^+(x, y)] = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K w_k \text{tr}(J_k \Sigma_k^+(x) J_k^\top),$$

and thus

$$\Delta V_{\text{intra}}(x) = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K w_k \left[\text{tr}(J_k \Sigma_k J_k^\top) - \text{tr}(J_k \Sigma_k^+(x) J_k^\top) \right].$$

Substituting the rank-one update for $\Sigma_k^+(x)$ yields

$$\Delta V_{\text{intra}}(x) = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K w_k \text{tr} \left(J_k \frac{\Sigma_k j_k(x) j_k(x)^\top \Sigma_k}{s_k^2(x)} J_k^\top \right).$$

Using cyclic invariance of the trace,

$$\text{tr} \left(J_k \Sigma_k j_k(x) j_k(x)^\top \Sigma_k J_k^\top \right) = j_k(x)^\top \Sigma_k J_k^\top J_k \Sigma_k j_k(x),$$

so

$$\Delta V_{\text{intra}}(x) = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K w_k \frac{j_k(x)^\top \Sigma_k J_k^\top J_k \Sigma_k j_k(x)}{s_k^2(x)}.$$

Equivalently,

$$\Delta V_{\text{intra}}(x) = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K w_k \frac{\|J_k \Sigma_k j_k(x)\|_2^2}{\sigma^2 + j_k(x)^\top \Sigma_k j_k(x)}.$$

D.4 Derivation of the inter-basin utility

We now derive the reduction in the between-basin term

$$V_{\text{inter}} = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K w_k \|\hat{f}_k - \bar{f}\|_2^2, \quad \bar{f} = \sum_{k=1}^K w_k \hat{f}_k,$$

after querying a candidate experiment x .

Unlike the intra-basin term, the inter-basin term depends on the relative positions and weights of the basin-level predictions. After observing an outcome y at x , both quantities change: the posterior probability of each basin is updated, and within each basin the target-region prediction mean is shifted by the new observation.

Updated basin weights. By Bayes' rule, the posterior probability of basin k after observing (x, y) is

$$w_k^+(x, y) = p(B = k | x, y, \mathcal{D}_t) = \frac{w_k p(y | x, B = k, \mathcal{D}_t)}{p(y | x, \mathcal{D}_t)}.$$

Under the local linear-Gaussian approximation,

$$p(y | x, B = k, \mathcal{D}_t) \approx \phi(y; m_k(x), s_k^2(x)),$$

where

$$m_k(x) = f(x; \theta_k), \quad s_k^2(x) = \sigma^2 + j_k(x)^\top \Sigma_k j_k(x),$$

and $\phi(\cdot; \mu, \nu)$ denotes the Gaussian density with mean μ and variance ν . Therefore,

$$w_k^+(x, y) = \frac{w_k \phi(y; m_k(x), s_k^2(x))}{\sum_{\ell=1}^K w_\ell \phi(y; m_\ell(x), s_\ell^2(x))}.$$

Updated basin-level target predictions. Within basin k , the posterior mean of the target-region prediction vector updates according to

$$\hat{f}_k^+(x, y) = \hat{f}_k + g_k(x)(y - m_k(x)), \quad g_k(x) = \frac{J_k \Sigma_k j_k(x)}{s_k^2(x)} \in \mathbb{R}^{|\mathcal{X}_{\text{tar}}|}.$$

Thus the updated global posterior mean on the target region is

$$\bar{f}^+(x, y) = \sum_{k=1}^K w_k^+(x, y) \hat{f}_k^+(x, y).$$

Updated inter-basin uncertainty. Conditioned on the new observation (x, y) , the between-basin term becomes

$$V_{\text{inter}}^+(x, y) = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{k=1}^K w_k^+(x, y) \left\| \hat{f}_k^+(x, y) - \bar{f}^+(x, y) \right\|_2^2.$$

Accordingly, the inter-basin utility of candidate x is

$$\Delta V_{\text{inter}}(x) = V_{\text{inter}} - \mathbb{E}_{y|x, \mathcal{D}_t} [V_{\text{inter}}^+(x, y)].$$

The predictive distribution of y under the current mixture is

$$p(y \mid x, \mathcal{D}_t) = \sum_{k=1}^K w_k \phi(y; m_k(x), s_k^2(x)),$$

so the expectation above can be written explicitly as the one-dimensional integral

$$\mathbb{E}_{y \mid x, \mathcal{D}_t} [V_{\text{inter}}^+(x, y)] = \int V_{\text{inter}}^+(x, y) p(y \mid x, \mathcal{D}_t) dy.$$

In contrast to the intra-basin case, this expectation does not collapse to a simpler weighted average, because both the updated responsibilities $w_k^+(x, y)$ and the updated basin predictions $\hat{f}_k^+(x, y)$ depend nonlinearly on the realized outcome y . Intuitively, a candidate receives high inter-basin utility when different basins predict substantially different outcomes at x , so that observing y is likely to either reweight the basin probabilities or pull their target-region predictions closer together. In the next subsection, we show that this integral admits an efficient pairwise form that can be evaluated by one-dimensional numerical quadrature.

D.5 Pairwise form and one-dimensional quadrature

A convenient identity for the between-basin variance is

$$\sum_{k=1}^K w_k \|\hat{f}_k - \bar{f}\|_2^2 = \sum_{1 \leq k < \ell \leq K} w_k w_\ell \|\hat{f}_k - \hat{f}_\ell\|_2^2.$$

Applying this identity to the updated basin mixture yields

$$V_{\text{inter}}^+(x, y) = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{1 \leq k < \ell \leq K} w_k^+(x, y) w_\ell^+(x, y) \|\hat{f}_k^+(x, y) - \hat{f}_\ell^+(x, y)\|_2^2.$$

This pairwise form is more convenient than the centered form because it separates the contribution of each basin pair and avoids recomputing the global mean explicitly.

Using the linear update

$$\hat{f}_k^+(x, y) = \hat{f}_k + g_k(x)(y - m_k(x)),$$

the difference between two updated basin means is

$$\hat{f}_k^+(x, y) - \hat{f}_\ell^+(x, y) = \underbrace{\hat{f}_k - \hat{f}_\ell - g_k(x)m_k(x) + g_\ell(x)m_\ell(x)}_{a_{k\ell}(x)} + \underbrace{(g_k(x) - g_\ell(x))}_{b_{k\ell}(x)} y.$$

Hence, for each pair (k, ℓ) ,

$$\|\hat{f}_k^+(x, y) - \hat{f}_\ell^+(x, y)\|_2^2 = \|a_{k\ell}(x) + b_{k\ell}(x)y\|_2^2,$$

which is a quadratic polynomial in y :

$$\|\hat{f}_k^+(x, y) - \hat{f}_\ell^+(x, y)\|_2^2 = A_{k\ell}(x) + B_{k\ell}(x)y + C_{k\ell}(x)y^2,$$

with coefficients

$$A_{k\ell}(x) = \|a_{k\ell}(x)\|_2^2, \quad B_{k\ell}(x) = 2 a_{k\ell}(x)^\top b_{k\ell}(x), \quad C_{k\ell}(x) = \|b_{k\ell}(x)\|_2^2.$$

Next, recall that the updated basin weights satisfy

$$w_k^+(x, y) = \frac{w_k \phi_k(y; x)}{\sum_{r=1}^K w_r \phi_r(y; x)}, \quad \phi_k(y; x) = \phi(y; m_k(x), s_k^2(x)).$$

Therefore, for any pair (k, ℓ) ,

$$w_k^+(x, y) w_\ell^+(x, y) p(y | x, \mathcal{D}_t) = \frac{w_k w_\ell \phi_k(y; x) \phi_\ell(y; x)}{\sum_{r=1}^K w_r \phi_r(y; x)}.$$

Substituting this into the expectation of the updated inter-basin term gives

$$\begin{aligned} & \mathbb{E}_{y|x, \mathcal{D}_t} [V_{\text{inter}}^+(x, y)] \\ &= \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{1 \leq k < \ell \leq K} w_k w_\ell \int \frac{\phi_k(y; x) \phi_\ell(y; x)}{\sum_{r=1}^K w_r \phi_r(y; x)} (A_{k\ell}(x) + B_{k\ell}(x) y + C_{k\ell}(x) y^2) dy. \end{aligned}$$

This is the form used in our implementation. For a fixed candidate x , the expectation reduces to a one-dimensional integral over the scalar observation y , and the only dependence on the target region enters through the precomputed coefficient vectors $a_{k\ell}(x)$ and $b_{k\ell}(x)$. Consequently, the inter-basin utility can be evaluated efficiently by numerical quadrature even when the target region contains many points.

Finally, combining this expression with

$$V_{\text{inter}} = \frac{1}{|\mathcal{X}_{\text{tar}}|} \sum_{1 \leq k < \ell \leq K} w_k w_\ell \|\hat{f}_k - \hat{f}_\ell\|_2^2$$

gives

$$\Delta V_{\text{inter}}(x) = V_{\text{inter}} - \mathbb{E}_{y|x, \mathcal{D}_t} [V_{\text{inter}}^+(x, y)].$$

In practice, we evaluate the one-dimensional integral numerically on a finite grid covering the predictive support of the current basin mixture at the candidate point.

D.6 Final cost-aware acquisition score

Combining the two components derived above, the total expected reduction in target-region uncertainty from querying candidate x is

$$\Delta \text{MSPE}_{\text{tar}}(x) = \Delta V_{\text{intra}}(x) + \Delta V_{\text{inter}}(x).$$

The intra-basin term captures how much the new observation is expected to reduce local predictive variance within each plausible basin, while the inter-basin term captures how much it is expected to reduce disagreement across basins.

To account for heterogeneous experiment costs, we rank candidates using the cost-aware score

$$S(x) = \frac{\Delta \text{MSPE}_{\text{tar}}(x)}{c(x)^\alpha} = \frac{\Delta V_{\text{intra}}(x) + \Delta V_{\text{inter}}(x)}{c(x)^\alpha},$$

where $c(x)$ is the cost of running experiment x , and $\alpha \geq 0$ controls the strength of cost penalization.

This form has a natural interpretation. When basin ambiguity is large, the inter-basin term tends to dominate, so the design prefers experiments that distinguish between qualitatively different extrapolations. Once the correct basin is largely identified, the acquisition increasingly behaves like a target-aware local optimal-design criterion, favoring experiments that most reduce predictive variance on $\mathcal{X}_{\text{tar}}$ per unit cost. This is exactly the behavior desired in budget-constrained scaling-law fitting: early experiments should resolve global ambiguity, while later experiments should refine the locally relevant scaling trend.

E Evaluation Details and Robustness Analyses

This appendix collects additional robustness and implementation details for the SL^2 sequential design procedure, including sensitivity to the cost exponent, robustness to the warm-start size, the cost of the warm-start phase, and acquisition-scoring complexity.

E.1 Sensitivity to the Cost Exponent

The cost exponent α controls the strength of cost penalization in the acquisition score. We use $\alpha = 0.4$ as a single global value for all tasks and scaling-law instances, rather than tuning it per task. Table 7 evaluates $\alpha \in \{0, 0.25, 0.4, 0.5, 0.75, 1.0\}$ and shows that the method is stable across a broad middle range. The extreme settings behave as expected: $\alpha = 0$ can spend budget too aggressively on expensive points in the low-budget regime, while $\alpha = 1$ can over-prioritize cheap points and underperform at larger budgets.

Budget	$\alpha = 0$	$\alpha = 0.25$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.75$	$\alpha = 1$
1%	-0.350 ± 0.389	0.288 ± 0.317	0.271 ± 0.439	0.212 ± 0.346	0.234 ± 0.352	0.117 ± 0.351
5%	0.357 ± 0.166	0.694 ± 0.090	0.683 ± 0.091	0.645 ± 0.116	0.614 ± 0.175	0.553 ± 0.249
10%	0.756 ± 0.057	0.796 ± 0.055	0.786 ± 0.065	0.762 ± 0.086	0.718 ± 0.135	0.682 ± 0.204

Table 7: **Cost-exponent sensitivity.** Sensitivity of SL^2 to the cost-penalty exponent α , aggregated over benchmark tasks and scaling-law instances. Higher target-region R^2 is better.

E.2 Warm-Start Cost and Size

The design-based methods require an initial nonlinear fit before local Jacobian and covariance information can be used. We therefore initialize **D-opt**, **V-opt**, and SL^2 with the same $2.5p$ lowest-cost observations, where p is the number of law parameters, and count this initialization toward the total budget. Table 8 reports the fraction of budget consumed by this warm start. The fraction is usually tiny relative to the full pool, though it is larger for tasks with more uniform costs and small candidate pools.

Metric	lr&bsz	domain	vocab	parallel	moe	data_con	sparsity	farseer
Warm-start cost / full pool	0.023%	5.000%	0.001%	1.615%	0.020%	0.007%	1.816%	0.028%
Warm-start cost / final checkpoint	0.23%	10.00%	0.01%	16.15%	0.20%	0.07%	3.63%	0.28%

Table 8: **Warm-start cost fraction.** Cost of the shared $2.5p$ warm-start phase. The final checkpoint is 10% of the full pool for most tasks and 50% for domain and sparsity.

Table 9 varies the number of warm-start observations. The results are similar for p , $2.5p$, and $5p$, suggesting that performance is driven mainly by the acquisition strategy rather than by the precise initialization size.

Budget	warmup=1	warmup= p	warmup= $2.5p$	warmup= $5p$
1%	0.194 ± 0.337	0.229 ± 0.386	0.271 ± 0.439	0.218 ± 0.391
5%	0.674 ± 0.086	0.658 ± 0.094	0.683 ± 0.091	0.616 ± 0.121
10%	0.788 ± 0.061	0.785 ± 0.065	0.786 ± 0.065	0.770 ± 0.070

Table 9: **Warm-start size ablation.** Ablation over warm-start sizes, aggregated over benchmark tasks and scaling-law instances. Higher target-region R^2 is better.

E.3 Computational Overhead

The budget percentages in the main text measure executed training-run cost, while acquisition overhead is CPU wall-clock time spent on small tabular pilot data and low-dimensional scaling-law parameters. In our implementation, the dominant overhead is repeated refitting: each decision performs $S = 64$ L-BFGS-B multi-start refits on the selected observations. The additional acquisition-scoring cost is approximately

$$O\left(M_c \left[KM_t P + K^2(M_t + Q) \right] \right),$$

where M_c is the candidate-pool size, M_t is the target-region size, P is the parameter dimension, and $Q = 500$ is the one-dimensional quadrature grid size.

We profile `lr&bsz`, one of the most computationally demanding benchmark tasks because it has the largest candidate pool and relatively high-dimensional laws. The measured decision-time breakdown is reported in Table 2 in the main text.