# IDENTIFICATION IS NOT REWARD: DISCOVERING TRAINING FEASIBILITY BOUNDARIES

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

A successful hyperparameter search finds configurations that train well, but may reveal little about where training ceases to be feasible. We study how to recover this boundary from a fixed budget of short runs and turn it into a usable learning-rate scale. Our approach fits a finite grammar of threshold models, selects experiments through template disagreement, and compiles the fitted boundary into a schedule. The protocol evaluates boundary identification and collected-run reward separately. A threshold-identification analysis explains why reward-optimal sampling can leave the boundary unidentified, while monotone inversion connects an identified cut to a control scale. On synthetic tasks, active discovery recovers planted cuts to the reported precision while reward-seeking Bayesian optimization achieves higher sample reward. On CIFAR-10, the same protocol improves identification in harder proxy-label settings and supports a separate sharpness-based feasibility probe. Under label-noise stress, compiled scales improve over hand-set cosine schedules; a scale-by-shape comparison attributes the improvement primarily to the scale. These results establish a practical route from short-run experiments to explicit, actionable descriptions of training feasibility.

## 1 INTRODUCTION

Choosing a learning rate is usually framed as finding a configuration that trains well. That configuration answers only one part of a broader question: how does the feasible training range change with the environment? A schedule selected at one noise level is a point solution. An explicit boundary in learning rate and noise describes a family of admissible choices and provides a scale for subsequent training. Recovering this boundary requires experiments that distinguish feasible from infeasible regions, including runs that a reward-seeking search would have little reason to sample.

Current practice offers strong tools for optimization. Cosine schedules and LR-range or 1cycle methods encode useful learning-rate structure (Loshchilov & Hutter, 2017; Smith, 2017; Smith & Topin, 2017). Bayesian optimization (BO), multi-fidelity search, and learned optimizers allocate computation to improve validation performance (Snoek et al., 2012; Li et al., 2018b; Falkner et al., 2018; Andrychowicz et al., 2016; Metz et al., 2022). These methods can optimize the observed objective without establishing that the resulting experiments identify a feasibility boundary. When reward peaks inside a feasible region, repeatedly sampling that region can improve the observed score while leaving nearby candidate boundaries indistinguishable. The distinction matters whenever the desired output is an explicit relation that can be inspected, tested under interventions, and used to set a control scale.

Training dynamics provide a reason to look for this structure. Edge-of-stability analyses connect learning rates to curvature (Cohen et al., 2021; Wang et al., 2022); studies of trainability frontiers reveal more complex feasible regions (Sohl-Dickstein, 2024; Kalra et al., 2025). Online controllers exploit dynamical signals during training (Défossez et al., 2023; Lewkowycz, 2021). These observations motivate an offline question: given a fixed budget of short runs, can we identify a compact description of feasibility and then use it to choose a learning-rate scale?

We answer this question with a finite-grammar discovery loop. Each candidate maps the training controls to a scalar feature and predicts feasibility by thresholding that feature. A bootstrap ensemble exposes disagreement among fitted models, and an experimentalist queries controls that help resolve



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

it. The fitted threshold is then inverted into a learning-rate scale and combined with a fixed schedule shape. The intermediate model adds an inspectable object to a selected hyperparameter: it specifies which controls are predicted to be feasible and how that prediction changes with the intervention.

**Identification and reward require separate measurements.** Our central comparison fixes the short-run budget and evaluates two outcomes: boundary recovery and reward accumulated by the sampled runs. On synthetic tasks with known cuts, active discovery concentrates near the boundary and estimates its location accurately; reward-seeking BO obtains higher reward while leaving larger boundary error (Figure 2). We call treating a reward win as evidence of boundary recovery the *reward fallacy*. The term identifies an evaluation mistake: either objective can be appropriate, but evidence for one does not establish the other. This is the experimental-design distinction between learning a system and optimizing its response (MacKay, 1992; Chaloner & Verdinelli, 1995).

**From a boundary to a training scale.** The compiler tests whether the recovered model supports a downstream control decision. For a product feature $x = \eta(1+T)$, for example, the fitted cut gives the scale $\hat{\eta}_c(T) = \hat{C}/(1+T)$. This relation can be evaluated at a target noise level and inserted into a schedule without another discovery campaign. On CIFAR-10, we distinguish an early-accuracy feasibility proxy from a separate sharpness-based label. Under label-noise stress, the compiled scale improves over hand-set cosine, and a factorial comparison shows that the benefit follows the scale across schedule shapes. Schedule search achieves comparable accuracy, providing a complementary way to choose a controller; the discovery loop additionally returns an explicit boundary model.

**Contributions.**

1. **A protocol for actionable boundary identification.** We connect a finite threshold grammar, disagreement-based experimental design, and an inversion interface. The protocol measures identification, sample reward, and downstream control separately.
2. **An explanation of the objective mismatch.** A threshold analysis characterizes the ambiguity left by boundary-avoiding queries and gives conditions under which inversion recovers the feasible upper scale.
3. **Evidence across controlled and neural-network settings.** Synthetic experiments isolate identification from reward. CIFAR-10 experiments examine harder proxy cuts, transfer across query designs, an independent evaluation grid, and a sharpness-based cut. Control and ablation results connect the recovered model to an effective learning-rate scale.

Together, these contributions make boundary recovery a measurable output of training experiments, alongside the performance of the configurations produced by those experiments.

## 2 Related work

**Active identification of feasibility.** Level-set estimation (Bryan et al., 2006; Gotovos et al., 2013) and active plan-feasibility learning (Noseworthy et al., 2021) target feasibility directly. Koyama & Ishikawa (2026) learn a failure boundary before safety-weighted BO; Li et al. (2025a) combine active constraint learning with objective optimization. Our distinction is therefore the finite, invertible boundary representation and its separate evaluation, rather than the use of feasibility information itself. Nikova et al. (2026) study acquisition trade-offs in feasible-region identification; our fixed disagreement, diversity, and margin terms instantiate a heuristic balance of global exploration and local refinement.

**Budgeted schedules and adaptive controllers.** Stepsize Anything (Tang et al., 2025), REX (Chen et al., 2021), and the explore-exploit schedule (Iyer et al., 2020) design schedules for limited training budgets. BrachistoneLR (Roni et al., 2026) studies schedule policies through a controlled benchmark. These works complement our scale-versus-shape evaluation: they prescribe how training progresses, while our intermediate model describes which controls are feasible. GreedyLR (Subramanian & Ganapathiraman, 2023), Mechanic (Défossez et al., 2023), and ABEL (Lewkowycz, 2021) adapt rates online; our controller is compiled from offline observations. Learned optimizers and HPO pursue performance directly (Andrychowicz et al., 2016; Metz et al., 2022; Jin et al., 2021; Li et al., 2018b; Falkner et al., 2018).

**Training laws and horizon dependence.** Scaling-law fitting (Kaplan et al., 2020; Hoffmann et al., 2022; Li et al., 2026) and symbolic regression (Udrescu & Tegmark, 2020; Brunton et al., 2016) motivate compact models. Zhou et al. (2026) extrapolate learning rates from low-cost pre-training experiments; we instead estimate a binary feasibility boundary and invert it to a control scale. Schaipp et al. (2025) connect convex optimization bounds to practical schedule behavior, providing context for the dependence of a useful schedule on its horizon. Our short-run label and long-horizon control performance remain separate measurements. See Shen et al. (2023) for broader efficient-training context and Appendix C for extended comparisons.

## 3 Problem setup and metrics

We treat training as a controlled dynamical system. An environment $e$ fixes architecture, data, and code. A run applies controls $u_t$ (learning rate, weight decay, label noise, ...) and produces coarse outcomes $z_t$ (losses, explosion flags, validation accuracy). An *experiment* is one such trajectory. We deliberately allow atypical controls, such as large $\eta$ or high noise, because these points inform the boundary even when they correspond to poor hyperparameters for production training. A complete symbol table appears in Appendix A.

**Definition 1** (Control, label, and short-run budget)**.** *Let $\mathcal{U}$ be a compact control box (e.g. $(\eta, T[, w])$). A short run returns a binary feasibility label $y(u) \in \{0, 1\}$ and an optional scalar reward $r(u) \in \mathbb{R}$ (e.g. early validation accuracy). A campaign of size $B$ is a sequence $u_1, \dots, u_B \in \mathcal{U}$ together with the observed pairs $\{(u_i, y_i, r_i)\}_{i=1}^{B}$.*

**Definition 2** (Finite-grammar machine law)**.** *A* machine law $\mathcal{L} = (x, C)$ *consists of a scalar feature $x : \mathcal{U} \to \mathbb{R}$ from a finite template bank $\mathcal{G}$ (Table 1) and a cut $C \in \mathbb{R}$, predicting*

$$\hat{y}_{\mathcal{L}}(u) = \mathbf{1}\{x(u) < C\}.$$

*We evaluate it through three operational criteria:*

**(C1) Compactness.** *The feature comes from a finite grammar and is selected by a complexity-penalized balanced error (Eq. 3).*

**(C2) Predictive validity.** *Its feasible/infeasible predictions are evaluated on held-out or interventional controls using balanced error (Eq. 1). The label is specified separately for each experiment.*

**(C3) Actionability.** *The fitted model admits inversion to a learning-rate scale, whose use in a fixed schedule family is evaluated downstream (Eq. 5).*

$$\mathcal{L}(\mathcal{L}) = \tfrac{1}{2}\big(\mathrm{FNR}(\mathcal{L}) + \mathrm{FPR}(\mathcal{L})\big). \tag{1}$$

**Definition 3** (Identification and reward objectives)**.** *Let $R_{\mathrm{id}}$ denote an identification error, such as $|\hat{C} - C^\star|$, held-out balanced error, or interventional label error, and define identification utility as $U_{\mathrm{id}} = -R_{\mathrm{id}}$. The collected-run reward is $U_{\mathrm{rew}} = B^{-1} \sum_{i=1}^{B} r(u_i)$. The two objectives score the same campaign from different perspectives.*

Boundary error measures the fitted model; sample reward measures the queried runs. Long-horizon controller accuracy is a third quantity, measured after compilation. Keeping these measurements separate allows a campaign to recover an informative boundary even when its exploratory runs have modest reward. Conversely, high-reward runs may leave multiple boundaries consistent with the observations.

## 4 Theoretical framework

We analyze noiseless thresholds to isolate the information supplied by a query. The results explain the objective mismatch and the inversion step; the finite-budget behavior of the implemented acquisition is evaluated empirically. Proofs appear in Appendix B.

**Assumption 1** (Planted cut and interior reward optimum)**.** *The label is $y(u) = \mathbf{1}\{x^\star(u) < C^\star\}$ for a feature in $\mathcal{G}$. The control box is compact, $x^\star$ and $r$ are continuous, and every maximizer of $r$ satisfies $x^\star(u) < C^\star - \delta$ for some $\delta > 0$.*

**Proposition 1** (Disagreement localizes threshold uncertainty)**.** *For $M$ models sharing feature $x$ with cuts $C^{(b)} \in [\bar{C}-\varepsilon, \bar{C}+\varepsilon]$, write $p(u) = M^{-1}\sum_b \mathbf{1}\{x(u) < C^{(b)}\}$. Their population label variance is $p(u)(1-p(u))$. It vanishes outside the feature band $|x(u) - \bar{C}| \le \varepsilon$ and is largest at attainable points whose vote fractions are closest to $1/2$.*

**Proposition 2** (Boundary-avoiding queries leave ambiguity)**.** *Suppose the queried feature values avoid $(C^\star - \varepsilon, C^\star + \varepsilon)$ for some $\varepsilon > 0$. All cuts in that interval produce the same labels at the queried controls. If the reward function is also unchanged across these candidate cuts, no estimator based on these observations can be uniformly consistent over them. Under Assumption 1, querying only reward maximizers has this ambiguity.*

**Proposition 3** (Reward optimality does not ensure identification)**.** *Under Assumption 1, a campaign that queries only a reward maximizer attains maximal mean reward but need not identify $C^\star$. By contrast, for a known feature with feasible and infeasible values bracketing the cut, realizable midpoint queries reduce a bracket of width $W$ to $W/2^n$ after $n$ noiseless queries.*

Reward-seeking methods can still identify a cut if they explore sufficiently: concentration in the interior does not exclude an increasing number of boundary queries. Our BO comparisons test the implemented finite-budget allocation.

**Theorem 1** (Inversion of a monotone feasibility model)**.** *Fix $(T, w)$ and let $x(\eta; T, w)$ be continuous and strictly increasing on $[\eta_{\min}, \eta_{\max}]$. If the fitted feature and cut equal the plant and the feasible set is nonempty, then*

$$\eta_{\sup}(T, w) = \sup\{\eta \in [\eta_{\min}, \eta_{\max}] : x(\eta; T, w) < \hat{C}\}$$

*is the upper boundary of feasible rates in that interval. An interior boundary satisfies $x(\eta_{\sup}; T, w) = C^\star$ and need not itself be feasible. For $x = \eta(1+T)$, the untruncated boundary is $C^\star/(1+T)$.*

*Remark* 1 (From inversion to deployment). A multiplicative factor below one moves an interior monotone boundary to the feasible side, provided the resulting rate stays in the modeled domain. Empty feasible sets and nonmonotone band features require separate handling (Appendix F). The theorem concerns the fitted short-run label; downstream training is evaluated separately because its horizon and trajectory differ.

## 5 METHOD

Figure 1 summarizes the discovery loop. A collector executes short runs, a discoverer fits boundary models, an experimentalist selects the next controls, and a compiler converts the fitted boundary into a schedule scale. Synthetic ground-truth constants are available only to evaluation.

**1. Collector.** We run standard training and log $(\eta, T, w)$, explosion flags, and validation accuracy. On CIFAR-10, a short discovery run (default 15 epochs) is labeled

$$y = \mathbf{1}\{\text{no explosion}\} \wedge \mathbf{1}\{\text{val-acc} \ge \tau\}, \qquad \tau = 0.25.$$

This is an *operational proxy* label. In the original 180 discovery runs, no explosion is recorded, so the accuracy threshold determines the label. The separate wide-$\eta$ probe uses the sharpness criterion $y_{\text{sharp}} = \mathbf{1}\{\lambda_{\max} < 2/\eta\}$ (Appendix D). We distinguish these two targets throughout. Synthetic collectors query a planted simulator that provides a true stability bit.

**Intervention tuple.** Discovery intervenes on $u = (\eta,\ T[, w])$: $\eta$ is the base learning rate, $T$ is the label-noise level on CIFAR or a temperature-like stress on synthetic tasks, $w$ is SGD weight decay, and $\rho$ is the warmup fraction in deploy shapes (Eqs. 7–8, default $\rho$=0.05). Discovery logs $(\eta, T, w)$ and the proxy bit $y$; $\rho$ is used only at compile/deploy.

**2. Discoverer.** Every template defines a scalar feature $x(\eta, T[, w])$ and predicts

$$\hat{y}(u) = \mathbf{1}\{x(u) < C\} \tag{2}$$

(stable when the feature is below the cut; orientation is fixed across the grammar). Table 1 lists fourteen base forms and four band templates used when dynamical labels induce annular feasible

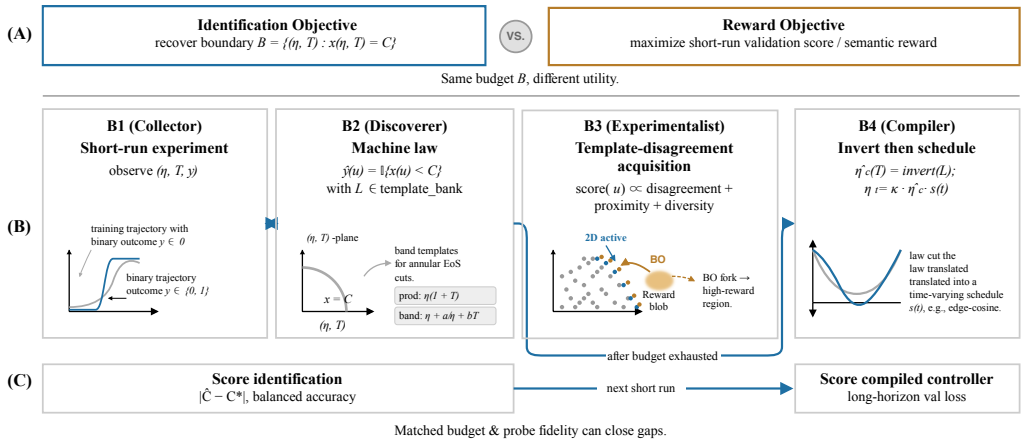


Figure 1: From short-run experiments to a compilable feasibility boundary. (A) Same short-run budget $B$, different utilities: recover a compilable cut versus maximize reward. (B) Collector $\rightarrow$ finite-grammar discoverer $\rightarrow$ template-disagreement experimentalist (BO forks to reward) $\rightarrow$ invert-and-schedule compiler, with a discovery loop for the next short run. (C) Score *identification* and *compiled control* separately so boundary recovery is not confused with long-horizon reward.

regions. Appendix G gives the exact grids for every free scalar. For each template and each grid point of its free scalars, $C$ is chosen by midpoint search over the sorted unique feature values on the training set to minimize Eq. 1; ties break toward the median feature. Candidates are ranked by the heuristic

$$\text{score}(\mathcal{L}) = \mathcal{L}(\mathcal{L}) + \alpha\, n_{\text{params}}(\mathcal{L}), \qquad \alpha = 0.03 \tag{3}$$

where $n_{\text{params}}$ includes the fitted cut. This fixed penalty favors compact predictors; it is a heuristic rather than a likelihood-based information criterion. The shape-grid search and functional overlap among templates are not separately penalized, so equivalent fits need not identify a unique feature family. When $w$ is not an intervention, drop the four $w$-coupled templates. Counts: 206 shape settings (no-$w$) or 366 (with $w$), each with one fitted $C$.

**3. Experimentalist.** Draw $n_{\text{cand}}$=2500 candidates from the configured control box, using its linear or logarithmic coordinates. Until at least six observations include both labels, the implementation samples passively. Fit the full grammar on the labeled history and $M$=10 bootstrap refits (resample indices with replacement; refit best law each time). For candidate $u$, let $\hat{y}^{(b)}(u) \in \{0, 1\}$ be the hard label of bootstrap law $b$, $\bar{p}(u) = \frac{1}{M}\sum_b \hat{y}^{(b)}(u)$, and $m(u)$ the margin of the current best law. The default implementation normalizes $x(u) - C$ by the standard deviation of absolute historical margins plus $10^{-6}$. Embed controls by min-max scaling each coordinate of $(\eta, T[, \log w])$ into $[0, 1]$; let $d_{\min}(u)$ be Euclidean distance in that embedding to the nearest past query. Let $v(u)$ be the sum of bootstrap label variance and label variance among the five best full-history fits (or all available fits if fewer than five). Score

$$s(u) = 1.4\, v(u) + 0.9\big(1 - |2\bar{p} - 1|\big) + 0.35\, d_{\min} + 0.7\, e^{-|m|/0.25}. \tag{4}$$

Select a batch of size $k$ (default $k$=1 per proposal call) by greedy diverse top-$k$. The first two terms favor uncertain labels; the distance term spreads queries through the control box, and the last term favors the current boundary. Bootstrap disagreement reflects sensitivity to the observations, while disagreement among the best full-history fits exposes competing template explanations. As identification baselines we also evaluate a grammar version-space splitter and GP level-set acquisitions (straddle, randomized straddle with $\beta \sim \chi^2(2)$, cUCB; Tables 9 and 3). BO maximizes GP-UCB on reward; passive samples uniformly. Table 9 compares full and near-boundary-only acquisition on the product plant.

**4. Compiler and deployment modes.** Invert the fitted classifier to a base scale $\hat{\eta}_c$ and place it inside a hand-chosen schedule shape using *only* discovery logs:

$$\eta_t = \text{clip}_{[0.02,\, 0.2]}\big(\kappa(T)\, \hat{\eta}_c(T, w)\big)\, s(t), \quad \kappa \in \{0.55, 0.7, 0.85\}. \tag{5}$$

Table 1: Phase-template grammar. Predict stable iff $x < C$. Grids in Appendix G. Band templates (`band-*` / `logband*`) target annular EoS-style feasible sets; `inv-wd` and `harm-wd` share $x = \eta(1+T)/(1+\delta w)$.

| **Template** | **Feature** $x$ | **Parameters** | **Settings** |
|---|---|---|---|
| prod | $\eta(1+T)$ | 1 | 1 |
| linear | $\eta + \lambda T$ | 2 | 21 |
| power | $\eta(1+\lambda T)^p$ | 3 | 56 |
| exp | $\eta e^{\gamma T}$ | 2 | 16 |
| sqrt | $\eta\sqrt{1+\lambda T}$ | 2 | 16 |
| log1p | $\eta(1+\gamma\log(1+T))$ | 2 | 16 |
| prod-pow | $\eta(1+T)^p$ | 2 | 13 |
| bilinear | $\eta(1+\mu T)+\lambda T$ | 3 | 49 |
| eta-T2 | $\eta(1+\lambda T^2)$ | 2 | 17 |
| eta-only | $\eta$ | 1 | 1 |
| band-eta | $\eta + a/\eta$ | 2 | 18 |
| band-eta-T | $\eta + a/\eta + bT$ | 3 | 80 |
| logband | $\lvert\log\eta - m\rvert + bT$ | 3 | 112 |
| logband-sq | $(\log\eta - m)^2 + bT$ | 3 | 84 |
| inv-wd | $\eta(1+T)/(1+\delta w)$ | 2 | 12 |
| prod-wd | $\eta(1+T)(1+\delta w/5{\times}10^{-4})$ | 2 | 11 |
| power-wd | $\eta(1+\lambda T)^p/(1+\delta w)$ | 4 | 125 |
| harm-wd | $\eta(1+T)/(1+\delta w)$ | 2 | 12 |

Let $t = e/(E-1)$ for epoch $e \in \{0, \ldots, E-1\}$ and total epochs $E$. The three shapes (warmup fraction $\rho$=0.05 for edge-warmup) are

$$s_{\text{edge-cos}}(t) = \tfrac{1}{2}\big(1 + \cos(\pi t)\big), \tag{6}$$

$$s_{\text{edge-warm}}(t) = \begin{cases} \max(t/\rho,\, 1/E) & t \le \rho, \\ \tfrac{1}{2}\big(1 + \cos(\pi \tfrac{t-\rho}{1-\rho})\big) & t > \rho, \end{cases} \tag{7}$$

$$s_{\text{piece}}(t) = \begin{cases} 1 & t < 0.4, \\ \tfrac{1}{2}\big(1 + \cos(\pi \tfrac{t-0.4}{0.6})\big) & t \ge 0.4. \end{cases} \tag{8}$$

The default report uses `edge-warmup`. Here $\hat{\eta}_c$ is the analytic upper-boundary scale before deployment clipping (for band templates, the upper root in Appendix F). The CIFAR compiler multiplies this scale by $\kappa$ and then clips the base rate to $[0.02, 0.2]$, as in the implementation. **Single-$\kappa$ mode** (primary): choose $\kappa^\star$ by weighted match to high-accuracy discovery runs; compile at the deploy $T$ (Env 3: $T$=0.2). Env 1/3 control accuracies never enter template, $\kappa$, or shape selection. Cosine baselines use a hand-set base of 0.1 with $s_{\text{edge-cos}}$. For each candidate $\kappa$, evaluate the compiled base $a_i(\kappa)$ at the controls $(T_i, w_i)$ of each logged run. The selector maximizes a weighted average of logged rewards, with weights $\exp(-|\eta_i - a_i(\kappa)|/0.25)$ multiplied by $0.25 + 0.75y_i$ when feasible observations are available. This favors factors whose implied rates are close to successful discovery runs. The factor is fixed before the long-horizon evaluation; the compiler uses neither target-run outcomes nor additional probes.

## 6 Experiments

We organize the evidence around three questions: does active discovery identify known boundaries, does it recover useful cuts in neural-network training, and does a recovered cut provide an effective control scale? Within each suite, identification and reward use matched short-run budgets.

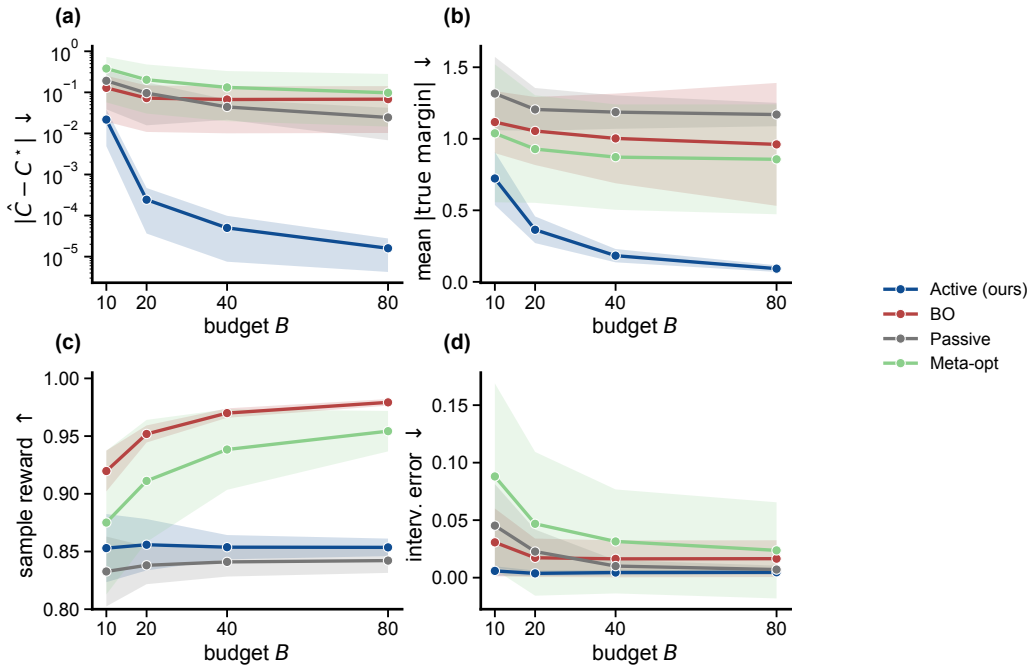


Figure 2: Synthetic phase plant under a shared budget. (a) $|\hat{C} - C^\star|$ (log). (b) Proximity to the true margin. (c) Sample reward (BO wins). (d) Held-out interventional error. Means $\pm$ std over 20 seeds.

### 6.1 Experimental setup

**Synthetic.** Planted in-grammar cuts in $(\eta, T)$ with known $C^\star$; Active / Passive / BO / meta-opt share budget $B$ (default report $B$=80, 20 seeds). Identification: $|\hat{C} - C^\star|$, margin proximity, interventional stability error. Reward: mean sample semantic recovery (BO's objective).

**Vision.** CIFAR-10 (Krizhevsky, 2009), ResNet-20/32 (He et al., 2016), SGD. Discovery: 15-epoch short runs with proxy bit $y$ (non-explosion and early val-acc $\geq \tau$; explosions almost never fire in the default box). A separate wide-$\eta$ probe uses an EoS-style sharpness bit. Control: 200 epochs; Env 3 uses 20% label noise. Identification baselines: Passive, reward BO, GP level-set (straddle / cUCB). Control comparisons include hand cosine, a shape$\times$scale factorial, and equal-budget schedule search (Appendix E.3). Discovery uses a 45k/5k train/validation split; the separate locked-controller check also evaluates the official test set (Appendix D).

### 6.2 Main results

**Boundary recovery and sample reward favor different designs.** With $B = 80$ short runs, Active recovers the planted cut to the three-decimal precision reported in Table 2; BO has mean cut error 0.068. BO obtains the highest sample reward (0.979 versus 0.854 for Active). Figure 2 connects this contrast to where experiments are placed: Active approaches the true margin, while BO samples higher-reward regions. Active also lowers interventional label error from 0.017 to 0.005. Reporting both measurements separates accurate boundary reconstruction from successful reward optimization. Appendix E.1 extends the comparison to other planted families and out-of-grammar cuts.

**Harder CIFAR settings expose differences in identification.** In the default ResNet-20 box, Active, Passive, and BO all reach about 0.97 balanced accuracy (Table 3). ResNet-32 separates them: Active reaches 0.957, compared with 0.925 for Passive and 0.921 for BO. Fitting on one design's runs and evaluating on the other designs gives 0.862 for Active and 0.491 for BO. This leave-other comparison diagnoses sensitivity to the query distribution; an independent-grid evaluation

Table 2: Synthetic dissociation at $B$=80 (mean ± std, 20 seeds). Boundary metrics favor Active; sample reward favors BO. Bold marks the best reported mean; zeros are rounded to three decimal places.

| **Design** | $\lvert\hat{C} - C^\star\rvert \downarrow$ | Interv. ↓ | Reward ↑ | $\lvert$margin$\rvert \downarrow$ |
|---|---|---|---|---|
| Passive | 0.025±0.018 | 0.007±0.003 | 0.842±0.011 | 1.169±0.082 |
| Active (ours) | **0.000**±0.000 | **0.005**±0.002 | 0.854±0.008 | **0.092**±0.023 |
| BO | 0.068±0.074 | 0.017±0.016 | **0.979**±0.003 | 0.961±0.430 |
| Meta-opt | 0.098±0.183 | 0.024±0.042 | 0.954±0.018 | 0.857±0.384 |

Table 3: Identification (balanced accuracy; bold: best mean, not a significance test). Proxy columns: default-box held-out bal. acc. ($B$=32, 5 seeds). Leave-other: fit on one design's 60 runs, score on the other two. Dyn.: wide-$\eta$ sharpness label with band grammar. Dash: not applicable.

| | **Early-accuracy proxy** | | | **Dynamical label** |
|---|---|---|---|---|
| **Design** | R20 | R32 | Transfer | Sharpness |
| Passive | 0.970±0.022 | 0.925±0.040 | 0.627 | 0.716±0.142 |
| Active | 0.966±0.023 | **0.957**±0.023 | **0.862** | **0.752**±0.055 |
| BO | **0.971**±0.026 | 0.921±0.039 | 0.491 | 0.736±0.087 |
| GP-straddle | 0.908±0.047 | 0.923±0.045 | — | 0.748±0.111 |
| GP-cUCB | 0.920±0.033 | 0.918±0.014 | — | 0.651±0.162 |

provides an additional check, with 0.984 for Active versus 0.950 for Passive and 0.920 for BO (Appendix E.2). On the sharpness label, Active and GP-straddle have similar mean accuracy (0.752 and 0.748), with substantial variability. The strongest active advantage is therefore in the harder proxy-label settings, while the sharpness probe demonstrates applicability to a distinct target.

**The identified cut supplies an effective learning-rate scale.** Under Env 3 (20% label noise), the single-$\kappa$ compiler reaches approximately 0.868 accuracy, compared with 0.857–0.860 for hand-set cosine (Table 4). The reported five-seed comparison gives a 0.95 percentage-point mean improvement, with a bootstrap 95% interval of $[0.55, 1.36]$ points (Appendix I). In the scale-by-shape factorial, law-scale cells remain in the 0.867–0.872 range across acquisition strategies and schedule shapes; fixed-0.1 cells reach 0.860–0.862. Within this comparison, the benefit follows the compiled scale. Actual-schedule BO probes achieve 0.869–0.872 (Appendix E.3), showing that direct schedule search is also effective at the control objective.

### 6.3 Analysis

**What the acquisition needs to resolve.** On the planted product boundary at $B = 40$, the full template acquisition obtains $4.2 \times 10^{-5}$ cut error, while near-boundary-only sampling obtains $3.1 \times 10^{-5}$. Both are below the errors of the GP level-set baselines (0.0035–0.0046) and BO (0.107; Appendix E.4). On this simple plant, concentrating queries near an informative model boundary accounts for the recovery; the full weighted acquisition is not required to reach that precision. The version-space splitter also improves on passive sampling, supporting the use of the finite grammar as an experimental-design prior.

**Interpreting the fitted family.** Prediction and symbolic family recovery answer different questions. Narrow intervention ranges can make several features nearly equivalent, and the fixed complexity penalty selects among them without correcting for every shape-grid comparison. The independent CIFAR campaigns select several families while retaining good held-out prediction (Appendix K). Accordingly, the fitted relation is an operational model of the measured label in the intervention box. Its value is assessed by boundary prediction and downstream use, rather than by treating the selected expression as a unique dynamical explanation.

Table 4: Env 3 compile interface (200-epoch control). Top: law vs. hand cosine. Bottom: factorial summary over acquisition×shape at fixed scale families ($B$=60 discovery).

| **Controller / scale** | **Accuracy** | **Setting** |
|---|---|---|
| Law-single-$\kappa$ | 0.868 ($B$=24/60) | compiled from discovery cut |
| Cosine (base 0.1) | 0.857–0.860 | no short-run search |
| Law-scale (any acq./shape) | 0.867–0.872 | factorial high band |
| Fixed-0.1 scale | 0.860–0.862 | same shapes |
| BO-best / random-stable scale | 0.845–0.849 | factorial low band |

## 7 Discussion

The experiments support a division of labor. Reward search allocates runs toward configurations that score well, whereas boundary discovery allocates them toward distinctions that the model must resolve. A compiled controller connects the two stages: identification supplies a scale, and a schedule determines how that scale is used over time. Budgeted schedules and online adaptation provide additional downstream comparators; our control evidence covers the explicitly reported schedule family. This distinction explains why similar final accuracy can coexist with different informational outputs. The explicit boundary remains available for inspection and evaluation when a searched schedule performs equally well.

The label determines what has been identified. The default CIFAR proxy asks whether a short run achieves an early-accuracy threshold without exploding; in the recorded default campaigns, the accuracy condition is decisive. The sharpness probe asks a different question about the relation between curvature and learning rate. Reporting these targets separately avoids interpreting an accuracy frontier as a dynamical stability certificate. Schedule performance can depend systematically on the training horizon (Schaipp et al., 2025). The compilation theorem therefore describes inversion of the short-run model; its long-horizon usefulness is established by the control experiments.

**Transfer and budget.** Leave-other and independent-grid evaluations test prediction away from the fitting design; the factorial tests the scale across schedule shapes. Joint feasibility-aware BO (Koyama & Ishikawa, 2026; Li et al., 2025a) is a natural additional comparator. Such comparisons should match probe budgets, deployment horizons, and tuning protocols; the present results cover the explicitly evaluated methods.

**Limitations and next questions.** The grammar supplies a strong prior and limits the boundaries that can be represented. Out-of-grammar experiments quantify some resulting error, but do not establish coverage of arbitrary training regimes. The acquisition weights and complexity penalty are fixed heuristics, and the neural-network evidence is limited to CIFAR-10. No general guarantee connects a short-run feasibility label to long-horizon behavior. Larger-scale evaluations, uncertainty calibration across template families, and criteria for detecting an inadequate grammar are natural next steps.

## 8 Conclusion

A short-run campaign can be valuable for the boundary it identifies as well as for the reward attained by its best configurations. We make that distinction explicit with a finite-grammar discovery protocol that measures identification and reward separately. Synthetic experiments reveal the allocation mismatch directly; CIFAR-10 experiments show useful proxy-boundary recovery and a separate sharpness-based application. Compiling the fitted relation supplies an effective learning-rate scale under label-noise stress. The resulting boundary model is a concrete, testable output of training experiments that complements performance-oriented hyperparameter search.

### AI USE STATEMENT

Generative AI tools were used for implementation assistance, plotting, LaTeX drafting, and language editing. The original workflow used OpenRouter and Moonshot Kimi; OpenAI Codex assisted with manuscript revision, mathematical wording, and layout. No additional experiments were generated for this revision, and AI tools were not used to invent experimental measurements. The authors are responsible for reviewing AI-assisted text and code and for the accuracy of the final claims and reported results.

### ETHICS STATEMENT

This work uses CIFAR-10 to study image-classification training and does not involve new human-subject data collection. The method allocates training experiments and may reduce the cost of boundary estimation, although exploratory runs also consume computation. Its empirical conclusions concern the reported training settings; the learned feasibility models are not safety certificates for deployed systems.

### REPRODUCIBILITY STATEMENT

Section 5 specifies the model-selection and acquisition rules. Appendix B provides proofs, Appendix D describes data splits and training protocols, and Appendices F–G give inversion rules and parameter grids. Additional results, controller selection details, statistical reporting, and template frequencies appear in Appendices E–K. The accompanying project code uses `pipeline.py` for the core discovery and control pipeline, with separate scripts for the additional evaluation suites.

## REFERENCES

Atish Agarwala, Fabian Pedregosa, and Jeffrey Pennington. Second-order regression models exhibit progressive sharpening to the edge of stability. *arXiv preprint arXiv:2210.04860*, 2023.

Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando de Freitas. Learning to learn by gradient descent by gradient descent. *Advances in Neural Information Processing Systems*, 29, 2016.

Atticus J Beachy and Ramana V Grandhi. Active learning of constraint boundaries using expected magnitude of incorrectness and neural networks. *Structural and Multidisciplinary Optimization*, 2026. doi: 10.1007/s00158-026-04352-4.

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, 2011.

Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

Brent Bryan, Robert C. Nichol, Jeff Schneider, Christopher Genovese, Larry Wasserman, and Christopher J. Miller. Active learning for identifying function threshold boundaries. In *Advances in Neural Information Processing Systems*, 2006. Straddle heuristic for level-set / contour estimation.

Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995.

John Chen, Cameron Wolfe, and Anastasios Kyrillidis. REX: Revisiting budgeted training with an improved schedule. *arXiv preprint arXiv:2107.04197*, 2021.

Zhanzhan Cheng, Yuyi Cheng, Chenbo Zhang, Xingbo Li, Jihong Guan, Fei Wu, and Shuigeng Zhou. Automated deep learning by recurrent hyperparameter optimization. *Nature Communications*, 2026. doi: 10.1038/s41467-026-72413-9.

Jeremy M Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021.

Alex Damian, Eshaan Nichani, and Jason D Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. In *International Conference on Learning Representations*, 2023.

Stéphane d'Ascoli, Maria Refinetti, and Giulio Biroli. Optimal learning rate schedules in high-dimensional non-convex optimization problems. *arXiv preprint arXiv:2202.04509*, 2022.

Aaron Defazio, Ashok Cutkosky, Harsh Mehta, and Konstantin Mishchenko. Optimal linear decay learning rate schedules and further refinements. *arXiv preprint arXiv:2310.07831*, 2023.

Alexandre Défossez, Léon Bottou, Francis Bach, and Nicolas Usunier. Mechanic: A learning rate tuner. In *Advances in Neural Information Processing Systems*, 2023.

Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: Robust and efficient hyperparameter optimization at scale. In *International Conference on Machine Learning*, 2018.

Alkis Gotovos, Nathalie Casati, Gregory Hitz, and Andreas Krause. Active learning for level set estimation. In *International Joint Conference on Artificial Intelligence*, pp. 1344–1350, 2013.

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*, 2016.

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. *Advances in Neural Information Processing Systems*, 2022.

Carl Hvarfner, Erik Orm Hellsten, Frank Hutter, and Luigi Nardi. Self-correcting bayesian optimization through bayesian active learning. In *Advances in Neural Information Processing Systems*, 2023.

Nikhil Iyer, V. Thejas, Nipun Kwatra, Ramachandran Ramjee, and Muthian Sivathanu. Wide-minima density hypothesis and the explore-exploit learning rate schedule. *arXiv preprint arXiv:2003.03977*, 2020.

Yuchen Jin, Tianjian Zhou, Liangyu Zhao, Yibo Zhu, Chuanxiong Liu, Lidong Mathy, and Rex Cheng. AutoLRS: Automatic learning-rate schedule by Bayesian optimization on the fly. In *International Conference on Learning Representations*, 2021.

Arlind Kadra, Maciej Januszewski, Josif Grabocka, and Frank Hutter. Scaling laws for hyperparameter optimization. In *Advances in Neural Information Processing Systems*, 2023.

Dayal Singh Kalra and Maissam Barkeshli. Phase diagram of early training dynamics in deep neural networks: effect of the learning rate, depth, and width. In *Advances in Neural Information Processing Systems*, 2023.

Dayal Singh Kalra, Tianyu He, and Maissam Barkeshli. Universal sharpness dynamics in neural network training: Fixed point analysis, edge of stability, and route to chaos. In *International Conference on Learning Representations*, 2025.

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.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017.

Motoki Koyama and Masato Ishikawa. Failure-avoidance Bayesian optimization with failure-boundary search for automatic exploration of digging control parameters. *Scientific Reports*, 16:15580, 2026. doi: 10.1038/s41598-026-45046-7.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

HyunJae Lee, Gihyeon Lee, Junhwan Kim, Sungjun Cho, Dohyun Kim, and Donggeun Yoo. Improving multi-fidelity optimization with a recurring learning rate for hyperparameter tuning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2309–2318, 2023.

Aitor Lewkowycz. How to decay your learning rate. *arXiv preprint arXiv:2103.12682*, 2021. Introduces ABEL.

Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv preprint arXiv:2003.02218*, 2020.

Guoyan Li, Yujia Wang, Swastik Kar, and Xiaoning Jin. Bayesian optimization with active constraint learning for advanced manufacturing process design. *IISE Transactions*, 2025a. doi: 10.1080/24725854.2025.2475505.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, 2018a.

Houyi Li et al. Predictable scale: Part I, step Law — optimal hyperparameter scaling law in large language model pretraining. *arXiv preprint arXiv:2503.04715*, 2025b.

Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. A system for massively parallel hyperparameter tuning. In *Proceedings of Machine Learning and Systems (MLSys)*, 2020.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52, 2018b.

Sijie Li, Shanda Li, Haowei Lin, Weiwei Sun, Ameet Talwalkar, and Yiming Yang. Spend less, fit better: Budget-efficient scaling law fitting via active experiment selection. *arXiv preprint arXiv:2604.22753*, 2026.

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? *arXiv preprint arXiv:2507.21184*, 2025.

Liming Liu, Zixuan Zhang, Simon Shaolei Du, and Tuo Zhao. A minimalist example of edge-of-stability and progressive sharpening. In *Advances in Neural Information Processing Systems*, 2025.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*, 2017.

David JC MacKay. Information-based objective functions for active data selection. *Neural Computation*, 4(4):590–604, 1992.

Niru Maheswaranathan, David Sussillo, Luke Metz, Ruoxi Sun, and Jascha Sohl-Dickstein. Reverse engineering learned optimizers reveals known and novel mechanisms. In *Advances in Neural Information Processing Systems*, 2021.

Luke Metz, Niru Maheswaranathan, C Daniel Freeman, Ben Poole, and Jascha Sohl-Dickstein. Using a thousand optimization tasks to learn hyperparameter search strategies. *arXiv preprint arXiv:2002.11887*, 2020.

Luke Metz, James Harrison, C Daniel Freeman, Adam Merchant, Luke Beyer, James Bradbury, Niki Agrawal, Ben Poole, Igor Mordatch, Adam Roberts, et al. VeLO: Training versatile learned optimizers by scaling up. *arXiv preprint arXiv:2211.09760*, 2022.

Paul Micaelli and Amos Storkey. Gradient-based hyperparameter optimization over long horizons. In *Advances in Neural Information Processing Systems*, 2021.

Hiroki Naganuma, Atish Agarwala, Priya Kasimbeg, and George E Dahl. What do near-optimal learning rate schedules look like? *arXiv preprint arXiv:2603.10301*, 2026.

Ioana Nikova, Sebastian Rojas Gonzalez, Tom Dhaene, and Ivo Couckuyt. Trade-offs in Bayesian active learning for feasible region identification. *Journal of Intelligent Manufacturing*, 2026. doi: 10.1007/s10845-025-02632-2.

Michael Noseworthy, Caris Moses, Isaiah Brand, Sebastian Castro, Leslie Kaelbling, Tomás Lozano-Pérez, and Nicholas Roy. Active learning of abstract plan feasibility. In *Robotics: Science and Systems XVII*, 2021. doi: 10.15607/RSS.2021.XVII.043.

Herilalaina Rakotoarison, Steven Adriaensen, Neeratyoy Mallik, Samir Garibov, Edward Bergman, and Frank Hutter. In-context freeze-thaw bayesian optimization for hyperparameter optimization. In *International Conference on Machine Learning*, 2024.

Md. Sadekur Rahman Roni, Md. Jalal Uddin Chowdhury, and Moutusi Dash Nimi. BrachistoneLR: A brachistochrone-inspired learning-rate schedule and a controlled benchmark of scheduling policies. *arXiv preprint arXiv:2609.08069*, 2026.

Fabian Schaipp, Alexander Hägele, Adrien Taylor, Umut Simsekli, and Francis Bach. The surprising agreement between convex optimization theory and learning-rate scheduling for large model training. *arXiv preprint arXiv:2501.18965*, 2025.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, 2009.

Li Shen, Yan Sun, Zhiyuan Yu, Liang Ding, Xinmei Tian, and Dacheng Tao. On efficient training of large-scale deep learning models: A literature review. *arXiv preprint arXiv:2304.03589*, 2023.

Leslie N Smith. Cyclical learning rates for training neural networks. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. *arXiv preprint arXiv:1708.07120*, 2017.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, 2012.

Jascha Sohl-Dickstein. The boundary of neural network trainability is fractal. arXiv preprint arXiv:2402.06184, 2024.

Shreyas Subramanian and Vignesh Ganapathiraman. Zeroth order GreedyLR: An adaptive learning rate scheduler for deep neural network training. In *IEEE International Conference on Pattern Recognition and Machine Learning*, 2023. doi: 10.1109/PRML59573.2023.10348370.

Anda Tang, Yiming Dong, Yutao Zeng, Zhou Xun, and Zhouchen Lin. Stepsize anything: A unified learning rate schedule for budgeted-iteration training. *arXiv preprint arXiv:2505.24452*, 2025.

Silviu-Marian Udrescu and Max Tegmark. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, 2020.

Zixuan Wang, Zhouzi Li, and Jian Li. Analyzing sharpness along GD trajectory: Progressive sharpening and edge of stability. In *Advances in Neural Information Processing Systems*, 2022.

Yunhua Zhou, Shuhao Xing, Junhao Huang, Xipeng Qiu, and Qipeng Guo. How to set the learning rate for large-scale pre-training? *arXiv preprint arXiv:2601.05049*, 2026.

## APPENDIX CONTENTS

The appendix provides proofs, implementation details, and the full supporting experimental record.

A Notation . . . 15
B Proofs . . . 15
C Extended related work . . . 16
D Implementation details . . . 17
E Additional experimental results . . . 17
E.1 Synthetic misspecification . . . 17
E.2 Additional identification results . . . 18
E.3 Additional control results . . . 18
E.4 Acquisition ablations . . . 19
F Band-template inversion rules . . . 19
G Template grids and search size . . . 19
H Evaluation and controller-selection details . . . 20
I Env 3 statistical reporting . . . 21
J Schedule-search protocol . . . 21
K Template selection frequencies . . . 21

# A NOTATION

| Symbol | Meaning |
|---|---|
| $e$ | Training environment (architecture, data, code) |
| $u = (\eta, T[, w])$ | Controls: base LR, stress / label noise, weight decay |
| $y(u)$ | Binary feasibility label (proxy or dynamical) |
| $r(u)$ | Short-run reward (e.g. early val-acc) |
| $B$ | Short-run campaign budget |
| $\mathcal{G}$ | Finite template grammar (Table 1) |
| $x, C$ | Feature and cut of a law $\mathcal{L} = (x, C)$ |
| $\mathcal{L}$ | Balanced error $\frac{1}{2}(\mathrm{FNR} + \mathrm{FPR})$ |
| $U_{\mathrm{id}}, U_{\mathrm{rew}}$ | Identification / reward utilities (Def. 3) |
| $\hat{\eta}_c$ | Fitted upper-boundary scale before deployment clipping |
| $\kappa, s(t)$ | Scale factor and schedule shape (Eq. 5) |

# B PROOFS

*Proof of Proposition 1.* Throughout this proof, $\bar{C}$ denotes the same central cut as in Proposition 1. The empirical distribution over bootstrap models assigns probability $p(u)$ to label one and $1 - p(u)$ to label zero, giving population variance $p(u)(1 - p(u))$. If $x(u) < \bar{C} - \varepsilon$, all models assign label one; if $x(u) > \bar{C} + \varepsilon$, all assign zero. Thus any positive disagreement is confined to the stated feature band. The function $p(1 - p) = 1/4 - (p - 1/2)^2$ is maximized by vote fractions closest to $1/2$ among those attainable by a control in the domain. Its maximum need not equal $1/4$, for example when the ensemble size is odd or feature values cannot reach the relevant interval. This statement is in feature coordinates and requires no geometric regularity of the level set. □

*Proof of Proposition 2.* For any queried $u$, either $x^\star(u) \leq C^\star - \varepsilon$ or $x^\star(u) \geq C^\star + \varepsilon$. Every candidate $C$ in the open interval $(C^\star - \varepsilon, C^\star + \varepsilon)$ therefore gives the same label at $u$. Choose distinct $C_1, C_2$ in this interval and keep the reward function fixed. The observations are identical under these two cuts. For any common estimate $\hat{C}$, the triangle inequality gives

$$\max\{|\hat{C} - C_1|, |\hat{C} - C_2|\} \geq |C_1 - C_2|/2.$$

The same lower bound holds for the larger of the two expected errors of a randomized estimator. Thus no estimator can be uniformly consistent over these indistinguishable cuts while the query gap persists. For an adaptive design, couple its random choices across the two plants: identical histories imply identical next queries, so the argument applies whenever the queried feature values retain the stated gap. Under Assumption 1, reward-maximizing controls are separated from the cut by $\delta$; choosing, for example, $\varepsilon = \delta/2$ gives the final claim. The unchanged-reward condition is essential: a known reward model could itself encode information about the threshold. □

*Proof of Proposition 3.* Let $u^\star \in \arg\max r$. Repeating $u^\star$ yields mean reward $r(u^\star)$, the maximum possible, while Proposition 2 gives a family of cuts consistent with all resulting observations. For the second statement, let $l < C^\star \leq h$ be a bracket whose endpoints correspond to feasible and infeasible feature values. Query a control with feature value $(l + h)/2$. A feasible label replaces $l$ by the midpoint; an infeasible label replaces $h$. Each query halves the bracket and preserves $l < C^\star \leq h$. After $n$ queries its width is $W/2^n$, and its midpoint estimates the cut with error at most $W/2^{n+1}$. This comparison establishes that maximizing reward alone is insufficient to ensure identification. It does not assert inconsistency of every BO acquisition, nor an optimality guarantee for the implemented template acquisition. □

*Proof of Theorem 1.* Continuity and strict increase imply that the feasible set is an initial segment of the rate interval. When nonempty, its supremum is the upper endpoint of that segment. If the supremum is interior, continuity forces $x(\eta_{\sup}; T, w) = C^\star$: a smaller value would allow a slightly larger feasible rate, and a larger value would exclude feasible rates immediately below the supremum. The strict label inequality excludes the interior boundary itself. If the feasible set extends to the upper

edge of the domain, its supremum is $\eta_{\max}$. For the product feature, solving $\eta(1+T) = C^\star$ gives $C^\star/(1+T)$ before truncation. An empty feasible set has no feasible upper rate; assigning a clipped numerical value in that case is an implementation fallback and carries no feasibility guarantee. □

## C Extended related work

**Learned optimizers and meta-optimization.** Learning update rules from a distribution of tasks (Andrychowicz et al., 2016; Metz et al., 2020; 2022) treats training as a sequential decision problem. Reverse-engineering such optimizers recovers known mechanisms (Maheswaranathan et al., 2021). Our approach retains an explicit boundary model between the experiments and the controller, allowing its predictive validity to be evaluated independently. Recent budgeted-training schedules make the comparison especially timely: Stepsize Anything derives a unified schedule for a fixed iteration budget (Tang et al., 2025), while REX revisits schedule design under budgeted training (Chen et al., 2021). BrachistoneLR evaluates schedule policies in a controlled benchmark (Roni et al., 2026), and the explore-exploit schedule of Iyer et al. (2020) explicitly allocates a fixed training budget between exploration and exploitation. These methods optimize or prescribe a schedule; our output is a feasibility relation that can be evaluated before that downstream choice.

**Scaling laws and hyperparameter laws.** Performance as a function of scale is a standard macroscopic description (Kaplan et al., 2020; Hoffmann et al., 2022). Recent work automates fitting of such formulae (Lin et al., 2025) or uses power-law structure inside HPO (Kadra et al., 2023; Li et al., 2025b). The work closest in spirit is budget-aware active selection for scaling-law fitting (Li et al., 2026), which allocates costly runs to improve extrapolation of a performance formula. We instead allocate runs to classify a feasibility cut and require compilation into a schedule.

**Training dynamics, sharpness, and schedule shapes.** Loss-landscape geometry (Li et al., 2018a; Keskar et al., 2017) and the edge of stability (Cohen et al., 2021; Damian et al., 2023) motivate a meaningful stability boundary. Analyses of sharpness along GD trajectories (Wang et al., 2022), second-order regression models (Agarwala et al., 2023), and minimalist network examples (Liu et al., 2025) show critical learning rates in controlled settings. Fixed-point analyses (Kalra et al., 2025) and fractal trainability frontiers (Sohl-Dickstein, 2024) show why a monotone threshold may be only a local approximation. The catapult phase (Lewkowycz et al., 2020) and early-training phase diagrams (Kalra & Barkeshli, 2023) further motivate separate dynamical targets. Practitioners probe ranges with LR-range / CLR / 1cycle (Smith, 2017; Smith & Topin, 2017), while cosine (Loshchilov & Hutter, 2017) encodes schedule geometry without an intervening law. Recent shape searches (Naganuma et al., 2026) show that base learning rate often dominates fine shape tuning. That is the setting in which we compile a discovered law. Theory-backed linear decay (Defazio et al., 2023) and high-dimensional analyses of schedules (d'Ascoli et al., 2022) provide complementary shape choices. The convex-analysis account of schedule behavior by Schaipp et al. (2025) gives a useful bridge between short-budget observations and long-horizon training, while also underscoring that such a bridge depends on the modeling assumptions. A broader survey of efficient large-scale training places schedule selection alongside resource allocation and systems constraints (Shen et al., 2023). Mechanic (Défossez et al., 2023) and ABEL (Lewkowycz, 2021) adapt rates online; our compiler instead uses a model fitted before deployment.

**Schedule-aware and multi-fidelity HPO.** AutoLRS (Jin et al., 2021), Zeroth Order GreedyLR (Subramanian & Ganapathiraman, 2023), gradient-based long-horizon HPO (Micaelli & Storkey, 2021), and multi-fidelity methods such as Hyperband, BOHB, and ASHA (Li et al., 2018b; Falkner et al., 2018; Li et al., 2020) allocate experiments to improve performance. Recurring learning rates (Lee et al., 2023) address probe fidelity, while freeze-thaw BO (Rakotoarison et al., 2024) models learning curves. Recurrent HPO (Cheng et al., 2026) uses short-run feedback to guide hyperparameter choices under delayed training rewards. Its target is training performance, rather than recovery of an explicit threshold model. Our protocol uses the short-run budget to estimate a boundary and evaluates the resulting controller separately.

**Symbolic regression and active design.** AI Feynman (Udrescu & Tegmark, 2020), SINDy (Brunton et al., 2016), and related expression search (Schmidt & Lipson, 2009) recover compact laws in physical systems. We restrict to a fixed template bank. Bayesian experimental design (MacKay,

1992; Chaloner & Verdinelli, 1995) and BO (Snoek et al., 2012; Bergstra et al., 2011) differ in utility. Level-set estimation (Gotovos et al., 2013; Bryan et al., 2006), constraint-boundary active learning (Beachy & Grandhi, 2026), and active learning of abstract plan feasibility (Noseworthy et al., 2021) target cuts rather than maxima. We compare template disagreement to GP straddle / randomized straddle / cUCB. Self-correcting BO (Hvarfner et al., 2023) also uses information about surrogate structure to guide experiments. The shared design question is where to spend the next query; the distinguishing output here is a threshold model with an explicit inversion interface.

**Feasibility-aware optimization and learning-rate transfer.** Koyama & Ishikawa (2026) separate offline success/failure-boundary learning from safety-weighted reward optimization. Their Gaussian-process classifier estimates success probabilities for a subsequent BO acquisition; our finite grammar makes the fitted feature and threshold explicit and supplies an inversion interface. Li et al. (2025a) likewise couple active constraint estimation with objective optimization. Such joint methods provide a stronger comparison for simultaneous feasibility and reward objectives than reward-only BO; our experiments isolate the latter objective and do not establish superiority over all constrained BO methods.

Nikova et al. (2026) examine trade-offs among acquisitions for feasible-region identification. This is directly relevant to Eq. 4: local boundary refinement can neglect unexplored regions, while broad exploration can spend queries away from an informative cut. Our combination of disagreement, distance, and margin terms is a fixed heuristic, with its behavior assessed by the reported ablations. Zhou et al. (2026) study fitting and transfer of learning rates from inexpensive experiments, including stability and feature-learning considerations. Their desired output is an effective rate for pre-training; ours is a classifier of the measured feasibility label whose upper scale is evaluated downstream.

# D Implementation details

**Synthetic.** Phase box $\eta \in [0.1, 3]$, $T \in [0, 1]$. Active: Eq. 4, $n_{\text{cand}}$=2500, 10 bootstraps. GP-UCB: RBF, noise $10^{-2}$, $\beta$=1.4. Twenty seeds.

**CIFAR.** ResNet-20/32, SGD momentum $0.9$, batch $128$. Discovery: $15$ epochs; proxy label $y =$ non-explosion and val. acc $\geq \tau$ (default $0.25$). In the released $180$ discovery runs, explosion rate is $0$. Control: $200$ epochs. Compiled base rate in $[0.02, 0.2]$ before the schedule multiplier. $\kappa$ from discovery logs only; noise branch assumes known target $T$. **Data splits.** The $50k$ CIFAR-10 train set is split once into $45k$ train / $5k$ val; discovery labels, BO rewards, $\kappa$ selection, and template fitting use this val split. Long-horizon controller selection likewise uses val; Table 8 additionally reports the official $10k$ test set under a *fixed* compiled Env 3 controller (no test-based $\kappa$/shape selection). **Sharpness probe.** $\lambda_{\max}$ uses 12 power iterations on one fixed mini-batch per $(\eta, T)$ configuration at the final discovery epoch unless noted; label $y_{\text{sharp}} = \mathbf{1}\{\lambda_{\max} < 2/\eta\}$ (Sec. 6.2).

**Additional suites.** The released code includes misspecification / compile / acquisition / LSE ablations; ResNet-20/32 proxy rediscovery against GP level-set baselines; val/test holdout; schedule-BO reference runs; and independent-grid / multi-seed discovery suites.

# E Additional experimental results

## E.1 Synthetic misspecification

Table 5 extends the synthetic comparison beyond the default `prod` plant. On other in-grammar plants Active keeps lower interventional error; on out-of-grammar cuts exact family match is impossible, but Active still degrades more gracefully than BO.

Table 5: Misspecification (additional synthetic results). Match = planted-family recovery. OOG rows report interventional error only.

| Plant | Design | Match ↑ | Interv. error ↓ |
|---|---|---|---|
| prod | Active | 1.00 | $0.005 \pm 0.002$ |
| prod | BO | 1.00 | $0.017 \pm 0.020$ |
| linear | Active | 1.00 | $0.015 \pm 0.017$ |
| linear | BO | 0.75 | $0.088 \pm 0.068$ |
| exp | Active | 1.00 | $0.001 \pm 0.001$ |
| exp | BO | 0.20 | $0.086 \pm 0.043$ |
| sin (OOG) | Active | — | $0.089 \pm 0.019$ |
| sin (OOG) | BO | — | $0.108 \pm 0.041$ |
| piecewise (OOG) | Active | — | $0.054 \pm 0.017$ |
| piecewise (OOG) | BO | — | $0.075 \pm 0.028$ |

### E.2 Additional identification results

Table 6 evaluates identification on an independent $(\eta, T, w)$ grid not chosen by any design. Default-box explosion-only labels remain uninformative ($< 0.5\%$ explosions), which is why the main compile path uses the proxy bit and why dynamical evidence uses the separate wide-$\eta$ sharpness probe summarized in Table 3.

Table 6: Independent held-out grid (additional identification results). Three discovery seeds $\times$ 24 short runs; same 80-point grid for all fits.

| Discovery design | Grid balanced accuracy ↑ |
|---|---|
| Passive | $0.950 \pm 0.034$ |
| Active | $0.984 \pm 0.000$ |
| BO | $0.920 \pm 0.014$ |

### E.3 Additional control results

Table 7 compares compiled control with schedule search under Env 3. Actual-schedule probes achieve accuracy comparable to the compiled controller; constant-rate probes introduce a mismatch between the probed and deployed trajectories. Table 8 reports a separate locked-controller validation/test check. Its absolute accuracies come from that evaluation suite and should not be pooled with the controller comparison.

Table 7: Compiled control and equal-budget schedule search. Env 3, CIFAR-10. Actual-schedule probes evaluate the proposed schedule during discovery; constant probes keep the proposed base rate fixed throughout the short run.

| Controller | Env 3 accuracy |
|---|---|
| Law-single-$\kappa$ | 0.868 ($B$=24/60) |
| BO-schedule (actual probe) | 0.869–0.872 |
| BO-schedule (constant probe) | $0.859 \pm 0.004$ |
| Cosine | 0.857–0.860 |

Table 8: Fixed Env 3 law controller on CIFAR-10 validation vs. official test (no test-based selection).

| Evaluation split | Accuracy |
|---|---|
| Val (5k) | $0.883 \pm 0.003$ |
| Test (10k) | $0.881 \pm 0.002$ |

### E.4 ACQUISITION ABLATIONS

Table 9 compares template-Active to GP level-set acquisitions and a grammar version-space splitter on the planted `prod` plant. Template-Active improves on the GP baselines, while near-boundary-only sampling has a slightly smaller cut error than the full acquisition. The product-plant result therefore does not depend on the precise combination of terms in Eq. 4.

Table 9: Synthetic LSE / version-space vs. template-Active on planted `prod` ($B$=40, 16 seeds). Near-boundary-only acquisition already matches full Active on $|\hat{C} - C^\star|$ ($3.1\times10^{-5}$ vs. $4.2\times10^{-5}$).

| **Design** | $\lvert\hat{C} - C^\star\rvert \downarrow$ | $\lvert\text{margin}\rvert \downarrow$ |
|---|---|---|
| Active (templates) | $4.2\times10^{-5}$ | 0.198 |
| Near-only | $3.1\times10^{-5}$ | 0.193 |
| Version-space | $3.8\times10^{-4}$ | 0.637 |
| Rand. straddle | 0.0035 | 0.628 |
| GP-straddle | 0.0046 | 0.624 |
| GP-cUCB | 0.0044 | 0.619 |
| Passive | 0.044 | 1.170 |
| BO | 0.107 | 0.960 |

## F BAND-TEMPLATE INVERSION RULES

For monotone increasing features, inversion returns the supremum of feasible rates when the feasible set is nonempty. Band features require both endpoints to describe the feasible set, even though the compiler uses its upper scale.

For `band-eta`, $x = \eta + a/\eta$ with $a > 0$ and $\eta > 0$. The inequality $x < C$ is equivalent to $\eta^2 - C\eta + a < 0$. A positive feasible interval exists exactly when $C > 2\sqrt{a}$, with endpoints

$$\eta_\pm = \tfrac{1}{2}\big(C \pm \sqrt{C^2 - 4a}\big).$$

The compiler returns $\eta_+$ before clipping to its deployment range. For `band-eta-T`, replace $C$ by $C_{\text{eff}} = C - bT$; the nonempty-interval condition becomes $C_{\text{eff}} > 2\sqrt{a}$.

For `logband`, let $q = C - bT$. If $q > 0$, the feasible rates form the open interval $(e^{m-q}, e^{m+q})$, and the upper scale is $e^{m+q}$. For `logband-sq`, the corresponding interval is $(e^{m-\sqrt{q}}, e^{m+\sqrt{q}})$ and the upper scale is $e^{m+\sqrt{q}}$. Neither has a feasible interval when $q \le 0$.

**Implementation conventions.** The existing `critical_eta` implementation clamps quadratic discriminants and logarithmic widths to zero before evaluating the expressions. This produces a numerical fallback even when the strict feasible interval is empty; it does not certify feasibility. Deployment multiplies the returned scale by $\kappa$ and clips the resulting base rate to $[0.02, 0.2]$. For a nonempty band, multiplying the upper scale by $\kappa < 1$ can cross the lower endpoint, so monotone-template guarantees do not extend to band schedules. The reported main control path uses a monotone proxy model; the band grammar supports the separate sharpness-identification experiments.

## G TEMPLATE GRIDS AND SEARCH SIZE

Orientation is fixed: predict stable iff $x(\eta, T[, w]) < C$. $C$ is chosen by evaluating midpoints of the sorted unique training features (plus one point below the min and one above the max); ties prefer the threshold closest to the median feature. Free scalars use the following discrete grids (matching code):

| Template | Free scalars | Grid |
|---|---|---|
| prod / eta-only | — | singleton |
| linear | $\lambda$ | $\mathrm{linspace}(0, 4, 21)$ |
| power | $\lambda, p$ | $\lambda \in \mathrm{linspace}(0.25, 2.5, 8)$, $p \in \mathrm{linspace}(0.4, 2.0, 7)$ |
| exp | $\gamma$ | $\mathrm{linspace}(0, 3, 16)$ |
| sqrt | $\lambda$ | $\mathrm{linspace}(0.25, 4, 16)$ |
| log1p | $\gamma$ | $\mathrm{linspace}(0, 3, 16)$ |
| prod-pow | $p$ | $\mathrm{linspace}(0.4, 2.0, 13)$ |
| bilinear | $\lambda, \mu$ | each $\mathrm{linspace}(0, 2, 7)$ |
| eta-T2 | $\lambda$ | $\mathrm{linspace}(0, 4, 17)$ |
| band-eta | $a$ | $\{10^z : z \in \mathrm{linspace}(-2, 2, 18)\}$ |
| band-eta-T | $a, b$ | $a \in \{10^z : z \in \mathrm{linspace}(-2, 2, 10)\}$, $b \in \mathrm{linspace}(0, 5, 8)$ |
| logband | $m, b$ | $m \in \mathrm{linspace}(-3, 2.5, 16)$, $b \in \mathrm{linspace}(-1, 2, 7)$ |
| logband-sq | $m, b$ | $m \in \mathrm{linspace}(-3, 2.5, 14)$, $b \in \mathrm{linspace}(0, 2, 6)$ |
| inv-wd / harm-wd | $\delta$ | $\mathrm{linspace}(0, 2000, 12)$ (same feature $x$) |
| prod-wd | $\delta$ | $\mathrm{linspace}(0, 2, 11)$ |
| power-wd | $\lambda, p, \delta$ | $5\times5\times5$ on $[0.25, 2]\times[0.5, 1.5]\times[0, 1500]$ |

Total shape settings (base fourteen): 206 without $w$-templates, 366 with $w$; band templates add 294 additional shape settings for dynamical / EoS labels. Each setting yields one $(C, \mathrm{score})$; the discoverer returns the global minimum score.

## H Evaluation and controller-selection details

**Acquisition conventions.** Bootstrap predictions are hard binary labels, with $M = 10$ index-resampled refits. The current best law supplies the margin, normalized by the standard deviation of absolute historical margins plus $10^{-6}$ in the default implementation. Disagreement combines bootstrap variance with variance among the five best full-history fits. Distances are Euclidean after min–max embedding of $(\eta, T[, \log w])$. Diverse top-$k$ selection uses a minimum embedded distance of 0.07; the default proposal batch has $k = 1$.

**Evaluation distributions.** The leave-other metric fits on one design's 60 runs and evaluates on the two other designs. It measures transfer between query distributions, which need not represent independent out-of-distribution tasks. The independent 80-point grid in Table 6 evaluates all fitted models on controls not chosen by any of the three designs. Default-box explosion-only labels carry little information because explosions are rare; the proxy and sharpness targets are evaluated separately.

**Single-factor and noise-dependent compilation.** The primary single-$\kappa$ mode selects one factor from discovery logs, then evaluates the fitted boundary at the target $T$. The additional noise-dependent mode uses $(\kappa_{\mathrm{clean}}, \kappa_{\mathrm{noise}})$ and switches at $T \geq 0.1$, assuming the target noise is known. Neither mode uses Env 1/3 long-horizon accuracies to select these factors. The default deployment shape is edge-warmup.

**Template selection and scale comparisons.** The original 60-run histories select active `prod_wd`, passive `linear`, and BO `prod`. Appendix K reports the independent multi-seed campaigns. The fitted family can vary across campaigns, so the control comparison fixes the discovered model before deployment. The factorial in Table 4 compares scale families across acquisition strategies and shapes. LR-range and 1cycle use a different probing procedure and are not evaluated here; the direct searched-controller comparator is the actual-schedule protocol in Appendix J.

## I ENV 3 STATISTICAL REPORTING

For law vs. hand cosine under Env 3 (5 seeds, 200 epochs), Welch $t$-tests and nonparametric bootstrap 95% CIs ($10^4$ resamples):

| **Comparison** | $\Delta$ mean (pp) | 95% CI | Welch $p$ |
|---|---|---|---|
| law − cosine | $+0.95$ | $[+0.55, +1.36]$ | 0.004 |

Raw seed accuracies accompany the released Env 3 control logs. Table 7 reports the separate schedule-search comparison; the interval and test above apply only to the stated law-versus-cosine contrast.

## J SCHEDULE-SEARCH PROTOCOL

The budget is 24 short runs of 15 epochs on ResNet-20, using SGD with momentum 0.9, batch size 128, and weight decay $5 \times 10^{-4}$. The search varies the base learning rate over $[0.02, 0.2]$ and chooses among the cosine, warmup, and piecewise shapes in Eqs. 6–8. The original suite calls this baseline BO-schedule. Its implemented acquisition is a sequential local search: after 8 trials, 70% of proposals perturb a top-five configuration; otherwise proposals are uniform. It is distinct from the GP-UCB reward baseline used for discovery. "Actual probe" uses the proposed schedule in the short run; "constant probe" uses constant-LR at the proposed base (fidelity mismatch). These runs support Table 7 only.

## K TEMPLATE SELECTION FREQUENCIES

Independent 3-seed discovery campaigns (24 short runs each; proxy label). The control model remains active `prod_wd` from the $B = 60$ campaign. Explosion-only fits select `prod` on every seed because the explosion signal is sparse. Neither redundant weight-decay template is selected.

| **Design** | **Seed templates** (proxy) | Most frequent |
|---|---|---|
| Passive | `prod_wd` (1), `prod` (2) | `prod` |
| Active | `eta_only` (1), `prod` (2) | `prod` |
| BO | `prod` (2), `linear` (1) | `prod` |