# LONG-HORIZON SCALING: HOW MODEL CAPABILITIES SHAPE THE RETURNS TO COMPUTATION

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

## ABSTRACT

Long-horizon agents improve solutions through sustained interaction, execution, and task feedback. Scaling studies relate performance to resources and capabilities, yet how existing capabilities shape returns to extended interaction remains less understood. To address this gap, we analyze AutoLab and EdgeBench, two long-horizon benchmarks. We find that starting performance and subsequent growth are associated with different capabilities: within a task category, similar early scores can precede different later gains. To formalize this finding, we model capability–time scaling with category-specific logistic power laws shared across models. Fitted to early trajectories, these curves extrapolate the observed models' category-average scores to later computation. However, rising average scores mask narrowing improvement opportunities: later gains concentrate among fewer improving models. High final scores and continued improvement also have distinct capability profiles. Predicted mean gains estimate each model's fraction of improving tasks; averaging these estimates forecasts the average share of improving models. These uneven returns motivate deciding whether a specific run should continue. We therefore derive a continuation policy to save time and compute with limited score loss. The policy conditions growth predictions on the run's observed progress and weighs immediate and delayed gains against computation costs. In replay with training and price calibration based on other models' histories, the policy saves roughly one-third of full-run time, with relative score losses of $2.4\%$ on AutoLab individual runs and $3.3\%$ on EdgeBench published mean curves.

## 1 INTRODUCTION

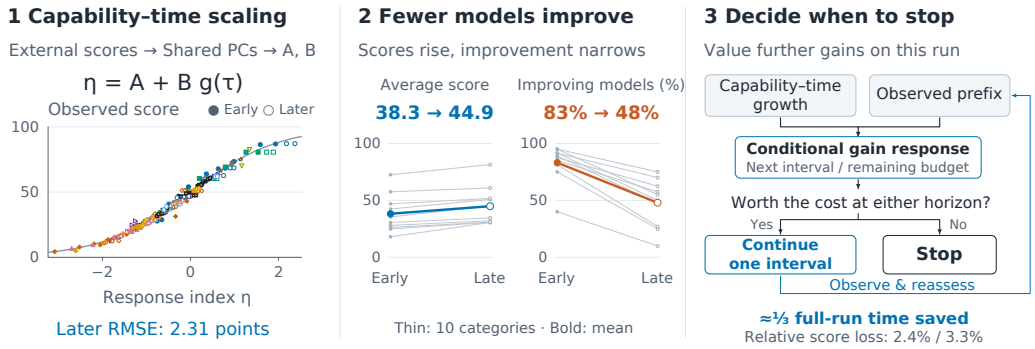



Figure 1: From capability–time scaling to adaptive continuation. (1) All $254$ observations versus the fitted index. The gray curve shows the logistic link; filled/open markers denote early/later observations. RMSE is averaged across categories. (2) Thin lines show ten categories; bold lines show their means. Scores use first/last interval endpoints (AutoLab $40/100\%$ budget; EdgeBench $4/12$ h). (3) Growth fitted on other models and the current prefix guide stopping. Reported tradeoffs use prices calibrated on other models and then fixed (AutoLab/EdgeBench; Section 5.3).



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.

Language models increasingly undertake research and engineering through sustained interaction, execution, and refinement (Lu et al., 2026; Wijk et al., 2025). Understanding these capabilities requires measuring growth with computation, beyond final scores. Scaling laws frame this question: training scaling relates performance to model size, data, and compute (Kaplan et al., 2020; Hoffmann et al., 2022); observational scaling uses shared capability representations (Ruan et al., 2024; Maia Polo et al., 2025); test-time scaling examines inference computation (Snell et al., 2025; Schaeffer et al., 2025). Recent work models growth during sustained interaction (Zhu et al., 2026), but its relation to model capabilities remains less understood. We therefore ask: *How do shared model capabilities shape long-horizon growth and the distribution of gains from further computation?*

Our central finding is that a shared capability space organizes both starting performance and subsequent growth through task-specific combinations (Figure 1, left). Task trajectories measure progress on particular tasks. We use external benchmarks as a common reference to compare model capabilities across tasks. Their correlated scores support a compact representation with multiple directions of model variation (Ruan et al., 2024; Zeng & Papailiopoulos, 2026). For each task category, two learned sets of weights combine a model's coordinates to describe starting performance and growth over time, respectively. Both sets are shared across models. These combinations parameterize a logistic power law, allowing similar starting scores to precede different later gains. The fitted weights express these growth patterns as external benchmark combinations. This shared representation lets us compare how capabilities relate to performance across task categories and budgets. Fits to early trajectories predict the observed models' later category-average scores.

The growth model above also helps explain why average scores rise while fewer models continue to improve (Figure 1, middle). On average across tasks in each category, declining participation (the share of improving models) concentrates gains, partly offset by more balanced gains among those that improve. A relation calibrated on early intervals uses predicted mean gains to estimate the fraction of tasks on which each model later improves. Averaging these forecasts across models estimates task-averaged participation. External benchmark associations help interpret these patterns. In systems and software engineering, repository and execution scores correlate more strongly with final scores than with continued improvement, whereas scientific reasoning scores correlate more strongly with continued improvement than with final scores. Gain timing shows that models with fewer late improvements may have already realized most of their observed gains.

This finding—that fewer models improve in later intervals—motivates deciding whether a particular model should continue its current task run. Runs sharing a category-level reference can differ in attained gains and recent stagnation. Conditioning capability–time growth learned from other models on the current run's observed history yields a bounded curve for next-interval and remaining-budget gains (Figure 1, right). The policy purchases one interval when either forecast justifies its computation cost (Hay et al., 2012), then reassesses. The remaining-budget forecast allows the policy to wait for delayed gains even when the next interval alone does not justify its cost.

We examine these growth patterns and evaluate the resulting continuation policy on AutoLab and EdgeBench, whose research and engineering tasks require sustained execution and task feedback (Xu et al., 2026; Zhu et al., 2026). Our AutoLab individual-run trajectories and official EdgeBench task–model mean curves support category-level analysis of ten categories, ten model versions, and 71 tasks. The category analysis tests temporal extrapolation for observed models and forecasts later improvement frequencies. Continuation training and price calibration exclude target models' long-horizon histories; decisions use external profiles and observed prefixes.

This joint analysis yields three connected contributions:

1. **Capability-dependent growth and gain concentration.** Task-specific capability combinations describe starting performance and growth, and help explain shrinking participation. Benchmark profiles distinguish final scores from continued improvement (Sections 3–4).
2. **A continuation policy derived from capability–time growth.** Conditioning capability-dependent progress on the current run yields a bounded curve for immediate and delayed gains, guiding further computation through sequential reassessment (Section 5).
3. **Empirical validation across two long-horizon benchmarks.** Early fits predict later scores with mean category $R^2 = 0.844$ and RMSE 2.31 points. Mean participation falls 35.2 percentage points from first to last interval. In replay, the policy saves roughly one-third of full-run time at $2.4\%$ relative score loss on AutoLab and $3.3\%$ on EdgeBench.

## 2 Measurement Framework and Experimental Setting

We study how verified solutions improve over sustained computation. AutoLab covers CUDA kernel optimization, model development, puzzles and challenges, and system optimization (Xu et al., 2026). EdgeBench includes formal mathematics, games, knowledge work, optimization, scientific and machine-learning tasks, and systems and software engineering (Zhu et al., 2026). Growth describes score trajectories over computation; gains are interval increments, and improvements are positive increments. Checkpoints reveal their timing and distribution.

Let $m$ index a model version and $i$ a concrete task in category $c(i)$. Runs have a predeclared budget $T$ and normalized time $\tau = t/T$. At predeclared checkpoints ending at $\tau_N = 1$, trajectories record their best verified score in $[0, 1]$; natural termination is absorbing. Write $s_{mi}(\tau)$ for a task–model mean curve. For fixed tasks $\mathcal{I}_c$ shared by models $\mathcal{M}_c$, the category response targets

$$\bar{s}_{mc}(\tau) = \frac{1}{|\mathcal{I}_c|} \sum_{i \in \mathcal{I}_c} s_{mi}(\tau). \tag{1}$$

Our category panels contain 71 tasks, four AutoLab and six EdgeBench categories, and four or five models per category (10 distinct versions overall). Membership depends only on trajectory availability. AutoLab curves average available replicates from the evaluations of Li et al. (2026); EdgeBench curves are published task–model means. Equations use $[0, 1]$ scores; results report scores and errors on 0–100. At checkpoint $k$, continuation uses score $Y_k$ and observed history $\mathcal{F}_k$.

Fixed membership keeps temporal comparisons on the same tasks and models within each category. Continuation replay uses all eligible histories under its held-out-model protocol (Section 5.3); it does not require every task to be shared by the category panel.

## 3 Capability–Time Scaling of Long-Horizon Performance

### 3.1 Shared Capabilities, Category-Specific Growth

Shared capability coordinates organize the level and timing of progress through category-specific readouts. Let $\mathbf{x}_m \in \mathbb{R}^{29}$ be model $m$'s preprocessed profile of 29 external benchmark scores (Appendix A). We estimate a regularized covariance from 82 reference entries excluding evaluated versions and aliases, and retain five principal components (PCs) (Jolliffe & Cadima, 2016):

$$\mathbf{z}_m = \mathbf{V}_5^\mathsf{T} \mathbf{x}_m \in \mathbb{R}^5. \tag{2}$$

The columns of $\mathbf{V}_5 \in \mathbb{R}^{29\times 5}$ are the retained covariance eigenvectors. Each PC combines the external measurements, and the same model coordinates serve all categories. PC1 is a broad positive direction across all 29 benchmarks; PC2 contrasts agent-execution with knowledge and reasoning measurements. Figure 9 shows all loadings and model coordinates; external-only missing-data estimation and preprocessing appear in Appendix A.

For first checkpoint $0 < \tau_0 < 1$, define normalized logarithmic time

$$g(\tau) = \frac{\log(\tau/\tau_0)}{\log(1/\tau_0)}, \qquad g(\tau_0) = 0, \quad g(1) = 1. \tag{3}$$

Logarithmic time measures proportional compute increases from the first checkpoint to the endpoint. Category-specific capability combinations determine early performance and temporal response:

$$A_{mc} = a_c + \mathbf{w}_c^\mathsf{T} \mathbf{z}_m, \qquad B_{mc} = b_c + \mathbf{v}_c^\mathsf{T} \mathbf{z}_m, \tag{4}$$

$$\eta_{mc}(\tau) = A_{mc} + B_{mc} g(\tau), \qquad \widehat{q}_{mc}(\tau) = \sigma(\eta_{mc}(\tau)), \tag{5}$$

Here $\widehat{q}_{mc}$ predicts the category mean $\bar{s}_{mc}$, and $\sigma(u) = (1 + e^{-u})^{-1}$ maps the response index $\eta$ to $[0, 1]$. $A_{mc}$ is the predicted first-checkpoint score logit; $B_{mc}$ is the modeled logit change to the endpoint. Each category shares its intercepts $a_c, b_c$ and weights $\mathbf{w}_c, \mathbf{v}_c$ across models; substituting a model's coordinates yields its $A_{mc}, B_{mc}$. Separate readouts allow similar starting performance to coexist with different subsequent growth.

The response is a *logistic power law*, since

$$\frac{\widehat{q}_{mc}(\tau)}{1 - \widehat{q}_{mc}(\tau)} = e^{A_{mc}} \left(\frac{\tau}{\tau_0}\right)^{\kappa_{mc}}, \qquad \kappa_{mc} = \frac{B_{mc}}{\log(1/\tau_0)}. \tag{6}$$

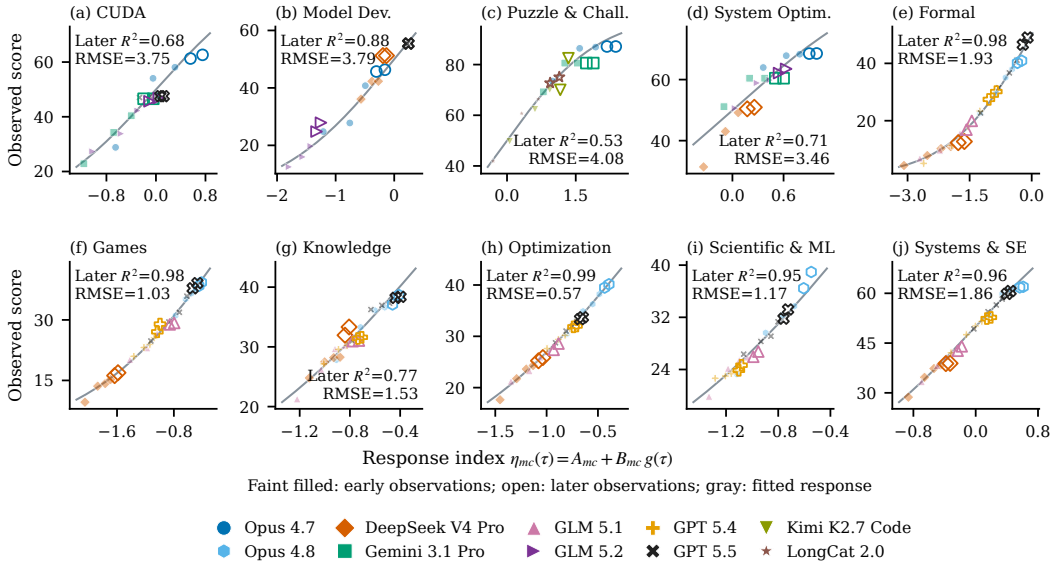



Figure 2: Shared capabilities organize category–time responses across all ten categories: AutoLab (a–d), EdgeBench (e–j). Filled/open markers show early/later category means, with model identity fixed by color and shape; all 254 means are retained. Gray curves show $100\sigma(\eta)$, and annotations report later $R^2$ and RMSE in score points. Section 3.2 specifies the early/later checkpoints and five-PC fitting protocol; Table 7 lists the category panels. Axis ranges vary by category.

The initial score odds are $e^{A_{mc}}$; doubling computation multiplies them by $2^{\kappa_{mc}}$. The score grows at $\kappa_{mc}\widehat{q}_{mc}(1-\widehat{q}_{mc})$ per unit log time, so gains diminish near saturation even with a fixed exponent. A matched comparison with all 29 external inputs favors the logistic over the affine log-time response in eight of ten categories (Appendix A.2).

The five-PC projection yields benchmark combinations that depend on category and time:

$$\eta_{mc}(\tau) = a_c + b_c g(\tau) + \left\{\mathbf{V}_5\left[\mathbf{w}_c + g(\tau)\mathbf{v}_c\right]\right\}^{\mathsf{T}} \mathbf{x}_m. \tag{7}$$

The effective weights $\mathbf{V}_5[\mathbf{w}_c + g(\tau)\mathbf{v}_c]$ combine all 29 measurements through the shared five-PC space. Each category's $\mathbf{w}_c$ sets the initial weights; $\mathbf{v}_c$ determines how they change with time.

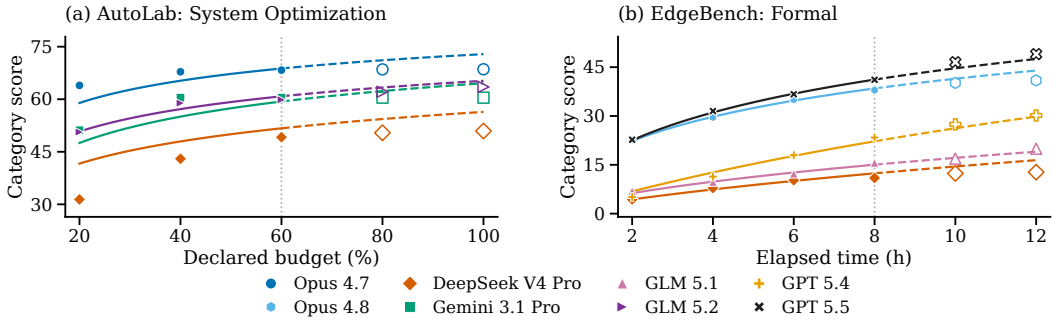



Figure 3: Observed trajectories show early gain realization and sustained growth. Fits from Figure 2; all 50 model–checkpoint means: (a) 12 tasks, four models; (b) eight tasks, five models. Filled/solid: early observations/fits; open/dashed: later observations/predictions. Dotted: fitting cutoffs.

### 3.2 Forecasting Later Category Performance

Within each category, we fit coefficients shared across models by regularized squared error on early scores: AutoLab $20\%, 40\%, 60\%$ budget and EdgeBench $2, 4, 6, 8$ hours. Later evaluation uses

$80\%, 100\%$ and $10, 12$ hours, respectively. At a fixed dimension, coefficients and regularization use early checkpoints only; Appendix A details dimension selection and comparisons.

Across all 90 later model–checkpoint means, mean category $R^2$ is $0.844$, weighting categories equally (Figure 2). These forecasts extend the observed models' early trajectories to later computation. Category RMSE ranges from $0.57$ to $4.08$ points, with five categories exceeding $R^2 = 0.95$.

On the same later points, retaining the last early score gives mean RMSE $3.10$; fitting an independent logistic time curve to each model gives $2.52$; replacing PCs with regularized model indicators gives $2.66$. Five PCs attain $2.31$, improving on independent curves in seven categories (Appendix A.4).

Figure 3 places these responses on budget and time axes to expose when gains occur. System Optimization shows substantial early gains followed by flatter observed trajectories; in Formal, Opus 4.8 and GPT 5.5 start at similar scores but diverge as computation proceeds.

## 4 Shrinking Improvement Opportunities and Concentrated Gains

For each interval, *participation* measures the fraction of models improving on a task; *improvement frequency* measures the fraction of tasks improved by a model.

### 4.1 Later Gains Reach Fewer Models

The growth curves describe how much each model improves on average across tasks. To understand who benefits from further computation, we examine how widely models share gains on each concrete task. Let $\Delta_{mi,k} = s_{mi}(\tau_{k+1}) - s_{mi}(\tau_k) \geq 0$ and $M_c = |\mathcal{M}_c|$. Define participation $P$ and effective gain coverage $C$ (Hill, 1973) by

$$P_{i,k} = \frac{1}{M_c} \sum_m \mathbf{1}\{\Delta_{mi,k} > 0\}, \qquad C_{i,k} = \frac{(\sum_m \Delta_{mi,k})^2}{M_c \sum_m \Delta_{mi,k}^2}. \tag{8}$$

Sums range over $\mathcal{M}_c$. Coverage is one for equal positive gains across all models and zero when none improves. For positive gains, their population coefficient of variation gives the exact identity

$$C_{i,k} = P_{i,k} U_{i,k}, \qquad U_{i,k} = \frac{1}{1 + \mathrm{CV}_{+,i,k}^2}, \qquad C_{i,k} \leq P_{i,k}. \tag{9}$$

Participation measures how widely improvement occurs; $U$ measures how evenly positive gains are distributed among improving models. Their product separates these two sources of concentration and is unchanged by a common rescaling of gain magnitudes.

Category-mean scores rise at every recorded interval in all ten categories. Yet across the 71 fixed tasks, task-averaged participation falls in every category from the first to the last interval, by $35.2$ percentage points on a ten-category average (Figure 4(a)). Positive gains become more balanced in seven categories. For tasks with positive gains in both intervals, fewer improving models drive the coverage decline; more evenly shared gains partly offset it. Requiring gains above $0.1$ points preserves the decline in all ten categories. Separately, expanding the positive-gain analysis to all available complete task–model curves yields declines in all ten categories, averaging $33.5$ percentage points (Appendix B).

### 4.2 From Mean Gains to Improvement Frequency

For a given model, task category, and interval, mean gain equals the fraction of tasks that improve multiplied by their mean gain, provided at least one task improves. A smaller mean can reflect fewer improving tasks, smaller improvements, or both. The growth curves predict mean gains for each model and task category; we use those gains to estimate the fraction of tasks that improve.

Let $\widehat{\mu}_{mc,k} = \widehat{q}_{mc}(\tau_{k+1}) - \widehat{q}_{mc}(\tau_k)$ be the predicted category-mean increment. Using the positive increments of the fitted curves, we model the fraction of tasks with positive gain as

$$\widehat{\pi}_{mc,k} = \frac{1}{1 + (\mu_c^\star / \widehat{\mu}_{mc,k})^\gamma}, \qquad \mu_c^\star > 0. \tag{10}$$

The category scales $\mu_c^\star$ and shared exponent $\gamma = 1.75$ are fitted to early improvement frequencies. At $\widehat{\mu} = \mu_c^\star$, half the tasks are predicted to improve.

Figure 4(b) groups later model–category–interval observations with similar predictions. The grouped predictions track the observed fractions of improving tasks.

In the fitted low-gain regime, improvement frequency falls proportionally faster than mean gain. When $\widehat{\mu} \ll \mu_c^\star$, $\widehat{\pi} \approx (\widehat{\mu}/\mu_c^\star)^{1.75}$: halving the predicted mean increment leaves about 30% of the predicted improvement frequency. Mean scores can thus rise while improvement reaches a shrinking share of tasks. Starting level $A_{mc}$ and temporal response $B_{mc}$ jointly set the interval gain: near saturation, further logit growth yields smaller score increments. The scale $\mu_c^\star$ determines how these increments translate into improvement frequency for that task category. These capability-dependent growth histories place models in the low-opportunity regime at different times.

Within a category, averaging improvement frequency over models equals averaging participation over tasks. The mean predicted frequency therefore estimates task-averaged participation. Fewer improving models reduce the upper bound on gain coverage, $C \leq P$; the balance of gains among them determines $U$ in $C = PU$. Appendix B.2 evaluates later calibration and compares forecast inputs under the same early-to-late split.

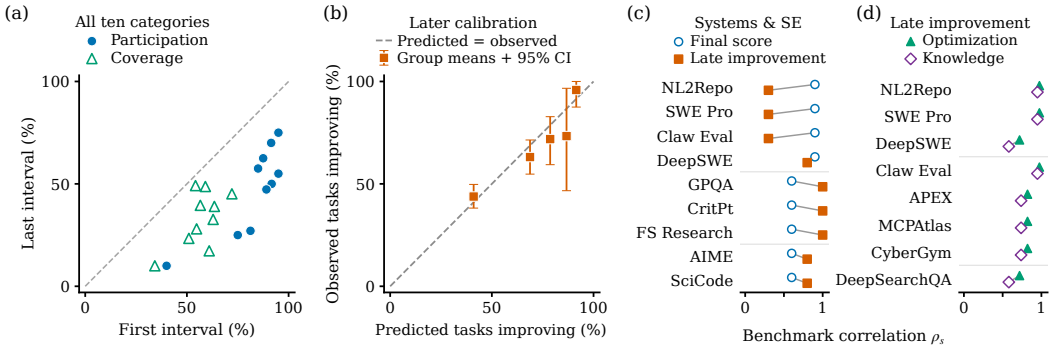



Figure 4: Fewer models improve later, with category-specific capability profiles. (a) First/last-interval participation and gain coverage across ten categories. (b) Task-weighted group means of predicted and observed improvement frequencies for all 90 later observations. Groups use early prediction quantiles; bars show 95% whole-task bootstrap intervals. Dashed lines indicate equality. (c) Systems & SE: benchmark correlations with final score and terminal improvement frequency. (d) Optimization's eight benchmarks with $\rho_s \geq 0.70$ for terminal frequency, compared with Knowledge. Correlations use five models per category; terminal frequency averages the last two intervals. Lines pair the same benchmark. Details and measurements: Appendix B.

### 4.3 Capability Profiles of Long-Horizon Progress

We next examine which external benchmarks are associated with high final scores, continued improvement, and earlier gains. We compare all 29 external measurements with final scores and improvement frequencies using within-category model rank correlations. These associations describe capability profiles rather than isolated benchmark effects. Terminal improvement frequency averages the fraction of tasks improving in the last two intervals. Gain timing distinguishes persistent progress from earlier gain realization (sources: Table 16; full profiles: Appendix B.3).

**Attainment and continued improvement have distinct capability profiles.** Across all 29 benchmarks, the average signed rank correlation across categories is more positive for final scores than for terminal improvement frequency. This difference persists when removing each model across categories and when varying the terminal window and improvement threshold (Appendix B.3). In Systems & SE, repository engineering and execution measurements—NL2Repo, SWE-bench Pro (Ding et al., 2025; Deng et al., 2025), and Claw Eval—track final scores more closely than terminal improvement frequency (Figure 4(c)). Scientific reasoning (GPQA, CritPt, FrontierScience Research) tracks continued improvement more closely than final scores; mathematics and coding (AIME, SciCode, DeepSWE) also track improvement (Rein et al., 2024; Tian et al., 2024).

**Engineering and execution recur across task categories.** Optimization exhibits a composite profile spanning repository engineering (NL2Repo, SWE-bench Pro, DeepSWE), execution and tool use (Claw Eval, APEX Agents, MCPAtlas, CyberGym), and information seeking (DeepSearchQA). All eight align with continued improvement in executable solvers; seven retain positive associations across nine choices of time window and improvement threshold. Six also track continued progress in Knowledge, where three of four tasks deliver domain-specific systems (Figure 4(d)). The shared profile reflects implementation and revision work across different subject matter.

**Fewer late improvements can accompany earlier gain realization.** In AutoLab System Optimization, AA-LCR, GPQA, SciCode, SimpleQA, MMLU-Pro, and MathArena associate with both fewer terminal improvements and earlier gains. By $60\%$ budget, Opus 4.7, Gemini 3.1 Pro, and DeepSeek V4 Pro have realized 91–$100\%$ of their observed 20–$100\%$ gains, versus $71.5\%$ for GLM 5.2 (Figure 3a). Full-trajectory timing preserves this profile (Appendix B.4).

## 5 From Scaling to Adaptive Continuation

Continued category growth can coexist with little progress on the current task. We evaluate further computation using a response to capability–time growth conditioned on the run's observed state.

### 5.1 Conditioning Growth on the Current Run

At each checkpoint, one bounded curve predicts next-interval and remaining-budget gains. Capability–time growth defines reference progress; attained score, recent improvement, and stagnation condition how much the current run gains from that progress.

For each held-out model, we refit capability–time growth on other models' completed histories, with fixed external PCs and a positive time coefficient. Write $r_{mc}(\tau)$ for this reference and $H_k = 1 - Y_k$ for the remaining score space. Suppressing $m, c$, we use a Weibull response (Weibull, 1951):

$$\begin{aligned} D_k(\tau) &= \log \frac{1 - r(\tau_k)}{1 - r(\tau)}, && \tau_k \le \tau \le 1, \\ \widehat{G}_k(\tau) &= H_k \left[1 - \exp\{-\alpha_k D_k(\tau)^{\beta_k}\}\right], && \alpha_k, \beta_k > 0. \end{aligned} \tag{11}$$

$D_k$ measures the reference's reduction in remaining score space on a log scale. At $\alpha_k = \beta_k = 1$, the curve transfers its fractional reduction, $1 - [1 - r(\tau)]/[1 - r(\tau_k)]$, to the current run's remaining space. Intensity $\alpha_k$ adjusts gain magnitude; shape $\beta_k$ adjusts its dependence on reference progress. The curve starts at zero, increases with horizon, and stays bounded by $H_k$.

The positive coefficients depend on seven features of the observed prefix, with task effects learned from other models' histories. Equal time intervals can represent different reference progress for different model–category pairs. Runs of the same model on the same task share a reference but receive different forecasts as their observed progress diverges.

We fit this single curve jointly to next-interval and remaining-budget gains, including zero-gain histories. All policy fitting excludes the target model's long-horizon histories; each source model is also excluded when constructing its training references. Source histories on the same task are allowed, while the target run contributes only its observed prefix. Appendix C.2 gives the feature definitions, regression, and training objective.

### 5.2 Purchase One Interval and Reassess

For $h \in \{\mathrm{n}, \mathrm{e}\}$, set $\tau_{k,\mathrm{n}} = \tau_{k+1}$ and $\tau_{k,\mathrm{e}} = 1$. The conditional curve gives ordered gains:

$$\widehat{g}_{k,h} = \widehat{G}_k(\tau_{k,h}), \qquad 0 \le \widehat{g}_{k,\mathrm{n}} \le \widehat{g}_{k,\mathrm{e}} \le H_k. \tag{12}$$

Let $\lambda \ge 0$ price each unit of declared budget in score gain. Subtracting computation costs from predicted gains gives the net values (Hay et al., 2012):

$$\widehat{Q}_{k,h} = \widehat{g}_{k,h} - \lambda c_{k,h}, \qquad c_{k,h} = \tau_{k,h} - \tau_k. \tag{13}$$

If $\max_h \widehat{Q}_{k,h} > 0$, purchase the next interval and reassess. The remaining-budget forecast allows the policy to continue even when the next interval alone is not worth its cost. Full-score and naturally

terminated runs stop. We use source histories to select a nonnegative price or the run-to-end option and keep that choice fixed for the target model. Realized duration is used only for cost evaluation.

Let $Q_{k,h}$ denote equation 13 evaluated with true conditional mean gains given $\mathcal{F}_k$. With these means at every visited checkpoint, the expected net value of reassessment satisfies

$$V_k^{\text{seq}} \geq \max(0, Q_{k,\text{n}}, Q_{k,\text{e}}). \tag{14}$$

Here $V_k^{\text{seq}}$ is the conditional expectation of subsequent gain minus priced declared cost. The plans—stop, one interval, and endpoint—use full declared costs, with $\lambda(1-\tau_k)$ for the endpoint. The bound assumes a finite declared grid, additive costs, history-based actions, and stopping that truncates the same potential trajectory. Proposition 1 gives the proof and bounds losses from gain-sign errors.

### 5.3 Cost–Performance Tradeoffs

We replay 492 individual AutoLab seed runs over seven target models and 252 published EdgeBench mean curves over five. Each target provides its external profile and current prefix, with its long-horizon histories held out from training.

Fixed-budget stopping uses a preset cutoff independent of observed progress. Time-based patience stops after a threshold fraction of the declared horizon since the last improvement, while recent-gain rules threshold the gain rate over one or two observed intervals.

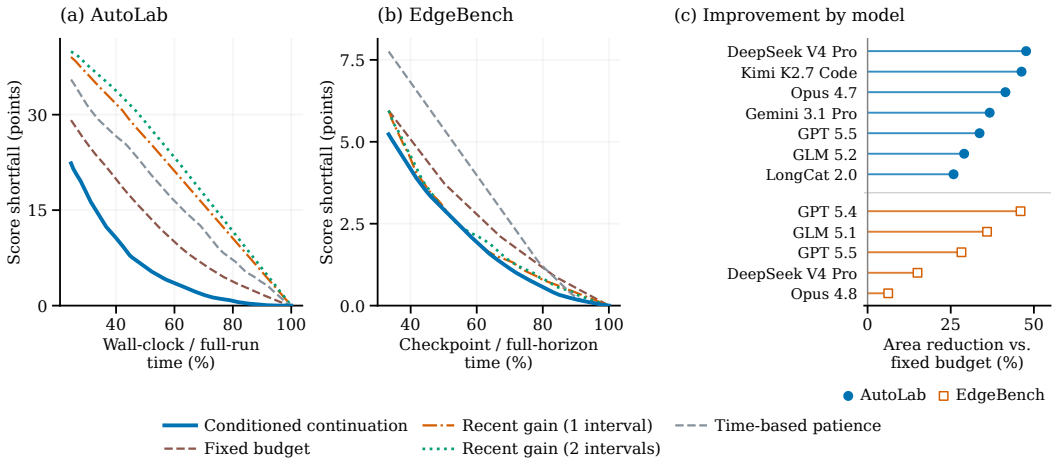



Figure 5: Conditioned continuation improves cost–loss tradeoffs. (a–b) Frontiers averaged over replicates then models. (c) Reductions in mean score loss versus fixed-budget stopping; all seven AutoLab and five EdgeBench evaluations improve. Comparisons use the same cost range across policies for each model and replicate; (a–b) display the range shared across each suite's evaluations.

In Figure 5, cost is charged time including initial observation, divided by full-run time: wall-clock time for AutoLab, checkpoint hours for EdgeBench. Loss is the score shortfall relative to completing the same trajectory. We average loss over the policies' shared cost range within each model–replicate, then over replicates and models (normalized frontier area; Appendix C.3).

Conditioned continuation has the lowest mean loss over these cost ranges in both suites. Relative to fixed-budget stopping, mean loss falls from 14.38 to 8.83 points on AutoLab and from 2.44 to 1.77 on EdgeBench, reductions of 38.6% and 27.4% (Table 13). Mean stopping loss falls for every model (Figure 5c). The available EdgeBench trajectories do not retain run-to-run variation in improvement timing within a task–model pair. This limits the run-specific information available for adaptation and may narrow the margin over fixed-budget stopping.

Saving half the time retains 90.5% of the full-run score on AutoLab and 91.8% on EdgeBench, versus 77.1% and 89.5% for fixed-budget stopping (Table 1). At 95% score retention, time savings are 38.1% on AutoLab and 38.4% on EdgeBench, versus 17.2% and 28.5% for fixed-budget stopping.

To choose an operating point before observing the target run, we calibrate the price on other models' runs to meet a target mean fraction of declared budget. We then fix that price for the target model.

Table 1: Matched operating points, interpolated from the mean frontiers in Figure 5. Score retention is relative to the mean full-run score. Ours denotes conditioned continuation; higher is better. Table 14 compares all baselines.

| | | AutoLab | | EdgeBench | |
|---|---|---|---|---|---|
| Matched condition | Reported outcome | Fixed budget | Ours | Fixed budget | Ours |
| Save 30% time | Score retained (%) | 89.9 | **97.3** | 94.7 | **96.8** |
| Save 50% time | Score retained (%) | 77.1 | **90.5** | 89.5 | **91.8** |
| Retain 95% score | Time saved (%) | 17.2 | **38.1** | 28.5 | **38.4** |
| Retain 90% score | Time saved (%) | 29.7 | **50.9** | 48.2 | **55.9** |

At these operating points, continuation saves $32.8\%$ of full-run time on AutoLab with a loss of $1.52$ points ($2.4\%$ relative to full-run scores), and $30.9\%$ on EdgeBench with a loss of $1.19$ points ($3.3\%$). Table 15 reports the source targets and complete operating points.

## 6 Related Work

**Capability-based scaling.** Downstream scaling relates benchmark performance to pretraining loss or perplexity (Gadre et al., 2025; Du et al., 2024). Observational scaling and Sloth instead organize performance through latent capabilities, accounting for family-specific compute efficiencies (Ruan et al., 2024; Maia Polo et al., 2025). Shared performance structure also enables efficient evaluation: tinyBenchmarks estimates scores from selected examples (Maia Polo et al., 2024), while BenchPress predicts missing benchmark scores from low-rank public score matrices (Zeng & Papailiopoulos, 2026). We use external capability coordinates to describe both starting performance and within-task temporal growth, with separate readouts for each task category.

**Long-horizon performance and compute returns.** Long-horizon evaluations measure agent reliability in units of human task duration (Kwa et al., 2025) and compare performance across computation budgets. RE-Bench compares best-of-$k$ agents with human experts across budgets (Wijk et al., 2025); AutoLab evaluates sustained research and engineering (Xu et al., 2026). On AutoLab, Li et al. (2026) study process behavior, experience reuse, and harness effects. EdgeBench fits benchmark- and task-family growth curves and tests temporal extrapolation (Zhu et al., 2026). Test-time scaling increases computation through repeated sampling (Brown et al., 2024) or longer reasoning (Muennighoff et al., 2025); allocation depends on problem difficulty (Snell et al., 2025). Heterogeneous success probabilities explain sampling power laws (Schaeffer et al., 2025), while transition models describe iterative self-correction (Yang et al., 2025). We study capability-dependent growth, gain concentration, and continuation across tasks and models.

**Adaptive stopping and computation value.** Rational metareasoning compares expected decision-quality gains with compute costs (Hay et al., 2012). AgentStop predicts unsuccessful episodes using token probabilities and trace features (Pham et al., 2026), while Ruan et al. (2026) use a recall-controlled cascade of hidden-state probes. Our policy prices next-interval and remaining-budget gains from a bounded, prefix-conditioned response to capability-dependent progress.

## 7 Discussion and Conclusion

Model capabilities organize the level and timing of long-horizon progress. Average scores rise across the ten categories while participation falls from the first to last interval. Shrinking participation concentrates gains; capability-dependent growth helps explain each model's later improvement frequency. Evaluations should report performance, gain timing, and the breadth of improvement.

Conditioning category-level growth on observed progress forecasts immediate and delayed gains for a specific run. The resulting policy improves time–score tradeoffs in replay on AutoLab individual runs and EdgeBench published mean curves. Broader model coverage and individual EdgeBench trajectories would extend this evaluation. Capability-based scaling can therefore describe long-horizon progress and guide decisions about the value of further computation.

## Reproducibility Statement

Appendices A–C describe the data preprocessing, evaluation splits, fitting procedures, hyperparameters, and continuation replay protocol, and provide the mathematical proofs. The EdgeBench analyses use officially released task–model mean checkpoint scores. Upon publication, we will release the complete AutoLab trajectories used in this study, together with processed analysis data and core fitting and replay scripts.

## References

Glenn W. Brier. Verification of forecasts expressed in terms of probability. *Monthly Weather Review*, 78(1):1–3, 1950. doi: 10.1175/1520-0493(1950)078⟨0001:VOFEIT⟩2.0.CO;2.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. arXiv preprint arXiv:2407.21787, 2024. URL https://arxiv.org/abs/2407.21787.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1):1–38, 1977. doi: 10.1111/j.2517-6161.1977.tb01600.x.

Xiang Deng, Jeff Da, Edwin Pan, Yannis Yiming He, Charles Ide, Kanak Garg, Niklas Lauffer, Andrew Park, Nitin Pasari, Chetan Rane, Karmini Sampath, Maya Krishnan, Srivatsa Kundurthy, Sean Hendryx, Zifan Wang, Vijay Bharadwaj, Jeff Holm, Raja Aluri, Chen Bo Calvin Zhang, Noah Jacobson, Bing Liu, and Brad Kenstler. SWE-Bench Pro: Can AI agents solve long-horizon software engineering tasks? arXiv preprint arXiv:2509.16941, 2025. URL https://arxiv.org/abs/2509.16941.

Jingzhe Ding, Shengda Long, Changxin Pu, Huan Zhou, Hongwan Gao, Xiang Gao, et al. NL2Repo-Bench: Towards long-horizon repository generation evaluation of coding agents. arXiv preprint arXiv:2512.12730, 2025. URL https://arxiv.org/abs/2512.12730.

Zhengxiao Du, Aohan Zeng, Yuxiao Dong, and Jie Tang. Understanding emergent abilities of language models from the loss perspective. 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. 53138–53167. Curran Associates, Inc., 2024. doi: 10.52202/079017-1683. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/5f1eee2509599faeeb3570a887016a64-Paper-Conference.pdf.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979. doi: 10.1214/aos/1176344552.

Samir Yitzhak Gadre, Georgios Smyrnis, Vaishaal Shankar, Suchin Gururangan, Mitchell Wortsman, Rulin Shao, Jean Mercat, Alex Fang, Jeffrey Li, Sedrick Keh, Rui Xin, Marianna Nezhurina, Igor Vasiljevic, Luca Soldaini, Jenia Jitsev, Alex Dimakis, Gabriel Ilharco, Pang Wei Koh, Shuran Song, Thomas Kollar, Yair Carmon, Achal Dave, Reinhard Heckel, Niklas Muennighoff, and Ludwig Schmidt. Language models scale reliably with over-training and on downstream tasks. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 67661–67682, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/a91869936a63d814971b6423990ecf6e-Paper-Conference.pdf.

Nicholas Hay, Stuart Russell, David Tolpin, and Solomon Eyal Shimony. Selecting computations: Theory and applications. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, 2012. doi: 10.48550/ARXIV.1207.5879. URL https://arxiv.org/abs/1207.5879.

Mark O. Hill. Diversity and evenness: A unifying notation and its consequences. *Ecology*, 54(2): 427–432, 1973. doi: 10.2307/1934352.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30016–30030, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/c1e2faff6f588870935f114ebe04a3e5-Abstract-Conference.html.

Ian T. Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A*, 374(2065):20150202, 2016. doi: 10.1098/rsta.2015.0202.

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. URL https://arxiv.org/abs/2001.08361.

Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, Ryan Bloom, Thomas Broadley, Haoxing Du, Brian Goodrich, Nikola Jurkovic, Luke Harold Miles, Seraphina Nix, Tao Lin, Chris Painter, Neev Parikh, David Rein, Lucas Jun Koba Sato, Hjalmar Wijk, Daniel M. Ziegler, Elizabeth Barnes, and Lawrence Chan. Measuring AI ability to complete long software tasks. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/85069585133c4c168c865e65d72e9775-Abstract-Conference.html.

Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004. doi: 10.1016/S0047-259X(03)00096-4.

Yiwei Li, Wanli Yang, Hexiang Tan, Xiangzhou Huang, Zhengyu Chen, Ziran Li, Borun Chen, Shanglin Lei, Huaisheng Zhu, Hao Tian, Fei Sun, Xunliang Cai, and Jingang Wang. Beyond final scores: A systematic evaluation of agents for long-horizon AI research and development. arXiv preprint arXiv:2608.13417, 2026.

Chris Lu, Cong Lu, Robert Tjarko Lange, Yutaro Yamada, Shengran Hu, Jakob Foerster, David Ha, and Jeff Clune. Towards end-to-end automation of AI research. *Nature*, 651:914–919, 2026. doi: 10.1038/s41586-026-10265-5. URL https://www.nature.com/articles/s41586-026-10265-5.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: evaluating LLMs with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 34303–34326. PMLR, 2024. URL https://proceedings.mlr.press/v235/maia-polo24a.html.

Felipe Maia Polo, Seamus Somerstep, Leshem Choshen, Yuekai Sun, and Mikhail Yurochkin. Sloth: scaling laws for LLM skills to predict multi-benchmark performance across families. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 39511–39556. Curran Associates, Inc., 2025. doi: 10.52202/085713-1318.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1025. URL https://aclanthology.org/2025.emnlp-main.1025/.

Dzung Pham, Kleomenis Katevas, Ali Shahin Shamsabadi, and Hamed Haddadi. AgentStop: Terminating local AI agents early to save energy in consumer devices. In *Proceedings of the ACM Conference on AI and Agentic Systems*, pp. 1051–1069. ACM, 2026. doi: 10.1145/3786335.3813163. URL http://dx.doi.org/10.1145/3786335.3813163.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Kai Ruan, Zihe Huang, Ziqi Zhou, Qianshan Wei, Jinghao Lin, Xuan Wang, and Hao Sun. Doomed from the start: Early abort of LLM agent episodes via a recall-controlled probe cascade. arXiv preprint arXiv:2607.06503, 2026. URL https://arxiv.org/abs/2607.06503.

Yangjun Ruan, Chris J. Maddison, and Tatsunori Hashimoto. Observational scaling laws and the predictability of language model performance. In *Advances in Neural Information Processing Systems*, volume 37, pp. 15841–15892. Curran Associates, Inc., 2024. doi: 10.52202/079017-0506.

Rylan Schaeffer, Joshua Kazdan, John Hughes, Jordan Juravsky, Sara Price, Aengus Lynch, Erik Jones, Robert Kirk, Azalia Mirhoseini, and Sanmi Koyejo. How do large language monkeys get their power (Laws)? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 53132–53176. PMLR, 2025. URL https://proceedings.mlr.press/v267/schaeffer25a.html.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 10131–10165, 2025.

Minyang Tian, Luyu Gao, Shizhuo Dylan Zhang, Xinan Chen, Cunwei Fan, Xuefei Guo, Roland Haas, Pan Ji, Kittithat Krongchon, Yao Li, Shengyan Liu, Di Luo, Yutao Ma, Hao Tong, Kha Trinh, Chenyu Tian, Zihan Wang, Bohao Wu, Yanyu Xiong, Shengzhu Yin, Minhui Zhu, Kilian Lieret, Yanxin Lu, Genglin Liu, Yufeng Du, Tianhua Tao, Ofir Press, Jamie Callan, Eliu Huerta, and Hao Peng. SciCode: A research coding benchmark curated by scientists. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/36850592258c8c41cecdaa3dea5ff7de-Abstract-Datasets_and_Benchmarks_Track.html.

Bertie Vidgen, Austin Mann, Abby Fennelly, John Wright Stanly, Lucas Rothman, Marco Burstein, Julien Benchek, David Ostrofsky, Anirudh Ravichandran, Debnil Sur, Neel Venugopal, Alannah Hsia, Isaac Robinson, Calix Huang, Olivia Varones, Daniyal Khan, Michael Haines, Zach Richards, Chirag Mahapatra, Brendan Foody, and Osvald Nitski. APEX-Agents. arXiv preprint arXiv:2601.14242, 2026. URL https://arxiv.org/abs/2601.14242.

Miles Wang, Robi Lin, Kat Hu, Joy Jiao, Neil Chowdhury, Ethan Chang, and Tejal Patwardhan. FrontierScience: Evaluating AI's ability to perform expert-level scientific tasks. arXiv preprint arXiv:2601.21165, 2026. URL https://arxiv.org/abs/2601.21165.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://arxiv.org/abs/2406.01574.

Waloddi Weibull. A statistical distribution function of wide applicability. *Journal of Applied Mechanics*, 18(3):293–297, 1951. doi: 10.1115/1.4010337.

Hjalmar Wijk, Tao Roa Lin, Joel Becker, Sami Jawhar, Neev Parikh, Thomas Broadley, Lawrence Chan, Michael Chen, Joshua M Clymer, Jai Dhyani, Elena Ericheva, Katharyn Garcia, Brian Goodrich, Nikola Jurkovic, Megan Kinniment, Aron Lajko, Seraphina Nix, Lucas Jun Koba Sato, William Saunders, Maksym Taran, Ben West, and Elizabeth Barnes. RE-bench: Evaluating frontier AI R&D capabilities of language model agents against human experts. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 66772–66832. PMLR, 2025. URL https://proceedings.mlr.press/v267/wijk25a.html.

Zhangchen Xu, Junda Chen, Yue Huang, Dongfu Jiang, Jiefeng Chen, Hang Hua, Zijian Wu, Zheyuan Liu, Zexue He, Lichi Li, Shizhe Diao, Jiaxin Pei, Jinsung Yoon, Hao Zhang, Mengdi

Wang, Radha Poovendran, Misha Sra, Alex Pentland, and Zichen Chen. AutoLab: Can frontier models solve long-horizon auto research and engineering tasks? arXiv preprint arXiv:2606.05080, 2026.

Zhe Yang, Yichang Zhang, Yudong Wang, Ziyao Xu, Junyang Lin, and Zhifang Sui. A probabilistic inference scaling theory for LLM self-correction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 13573–13587. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.685. URL https://aclanthology.org/2025.emnlp-main.685/.

Yuchen Zeng and Dimitris Papailiopoulos. You don't need to run every eval. arXiv preprint arXiv:2606.24020, 2026. URL https://arxiv.org/abs/2606.24020.

Deyao Zhu, Xin Zhou, Shengling Qin, Xuekai Zhu, Hangliang Ding, Shu Zhong, Zixin Wen, Zhonglin Xie, Chenhui Gou, Linxuan Ren, Yueyang Wang, Junfeng Zhong, Rui Liu, Tian Gao, Yangguang Lin, Jingyuan Zhang, Maojia Song, Xuan Qi, Jinhong Wu, Chenyang Zhang, Yinzhu Piao, Ziru Niu, Hongbin Lin, Lingxiang Meng, Peng Tang, Chengyao Tang, Shanyu Wu, Huanyu Zheng, Yu Liu, Liya Zhu, He Wang, Ming Ding, Ziyu Wan, Hao Liu, Sibo Wang, Haotian Zhu, Xintian Zhang, Nan Chai, Yipeng Liu, Panhao Lai, Sihang Yuan, Zixin Su, Ge Zhang, Wangchunshu Zhou, Yantao Du, Wenhao Huang, and Guang Shi. EdgeBench: Unveiling scaling laws of learning from real-world environments. arXiv preprint arXiv:2607.05155, 2026.

# A Representation and Fitting Details

## A.1 External Measurements and the Shared PCs

External scores are assembled from BenchPress (Zeng & Papailiopoulos, 2026) and supplementary published benchmark results. Table 16 lists the 29 measurements and links their benchmark papers or official releases. External benchmark scores are put on a $[0, 1]$ scale. Let $\psi$ denote the logit transformation after clipping the score to $[0.005, 0.995]$. For benchmark $j \in \{1, \ldots, J\}$ with $J = 29$, let $\bar{\psi}_j$ and $s_j$ be the reference mean and standard deviation of this transformed score. Let $\omega_j$ be the inverse square root of the number of measurements in its benchmark family. An observed entry $b_{mj}$ produces

$$x_{mj} = \frac{\omega_j}{\sqrt{\sum_{\ell=1}^{J} \omega_\ell^2}} \frac{\psi(b_{mj}) - \bar{\psi}_j}{s_j}. \tag{15}$$

Family weighting prevents several related benchmark variants from receiving proportionally more total squared weight simply because more variants are available. FrontierScience and HMMT each have two measurements; the SWE family has three; other measurements receive unit family weight.

The $82$ reference entries provide $901$ observed cells out of $82 \times 29$. For the ten target model versions, $248$ of $290$ cells use published scores and $42$ use frozen external-only estimates. The missing-entry estimator uses a rank-two external fit and regularized projection with coefficient $0.1$; observed entries are preserved exactly. This estimator supplies missing values, whereas the five downstream PCs come from a separately estimated full-rank covariance. The covariance update uses conditional first and second moments for missing entries, following the expectation–maximization principle (Dempster et al., 1977), and linear shrinkage toward the identity (Ledoit & Wolf, 2004). Its shrinkage coefficient $0.1$ is selected by hiding observed external entries in reference-model validation splits. The covariance is then family-weighted before extracting its leading eigenvectors. The shared PC coordinates are not whitened. All external preprocessing and target completion are fixed before long-horizon response fitting.

**Dimensionality and interpretation.** The fixed-dimension sweep in Table 2 informs our five-PC choice through compactness and later-checkpoint errors. At each dimension, response coefficients, regularization, and temporal structure use only early checkpoints. Selecting dimension from last-early-checkpoint errors alone, over one to 29 PCs, gives either a common dimension of 19 or category-specific dimensions; the table reports both protocols. Individual PC names interpret the fitted external loadings: PC1 is a broad positive direction; subsequent PCs describe relative benchmark patterns. PC signs are conventional, with contrast loadings expressing relative performance across measurements.

Table 2: Dimension sweep and early-only dimension selection on the same temporal panels. Metrics average the ten categories equally; RMSE is in score points.

| Number of PCs | 1 | 3 | 5 | 7 | 29 |
|---|---|---|---|---|---|
| Mean later-score RMSE | 4.861 | 3.180 | 2.315 | 2.288 | 2.504 |

| Early-only dimension selection | Selected PCs | RMSE | $R^2$ |
|---|---|---|---|
| Common across categories | 19 | 2.486 | 0.824 |
| Category-specific | 2–23 | 2.434 | 0.830 |

## A.2 Response on the Score Scale

Differentiating the logistic power-law response with respect to log time gives

$$\frac{\partial \widehat{q}_{mc}(\tau)}{\partial \log \tau} = \kappa_{mc} \widehat{q}_{mc}(\tau)\big[1 - \widehat{q}_{mc}(\tau)\big]. \tag{16}$$

The same fitted logit growth coefficient yields different score changes depending on the current performance level, with changes diminishing in magnitude near saturation. Equation 16 translates the temporal coefficient estimated from category trajectories into score gains along the fitted curve.

**Matched response-family comparison.** We compare the logistic response with its affine counterpart, $\widehat{q} = A + Bg(\tau)$, using the complete 29-dimensional external profile in both cases. The task and model panels, early checkpoints, native-score squared-error objective, and 90 later evaluation means match the five-PC analysis. Each family selects its regularization and shared-versus-capability-dependent temporal coefficient on early checkpoints, then refits on the full prefix. Table 3 reports all ten categories. The logistic family reduces mean category RMSE from 2.83 to 2.50 points (11.4%), improving eight categories. This comparison evaluates the response family with the external input space held fixed; the five-PC dimension comparison appears in Table 2.

Table 3: Matched link-family comparison using all 29 external inputs. Both families select regularization and slope structure on early checkpoints and predict the same 90 later means. Errors are RMSE in score points.

| Category | Affine in log time | Logistic power law |
|---|---|---|
| CUDA | 4.42 | 4.02 |
| Model Development | 5.68 | 5.73 |
| Puzzle & Challenge | 5.86 | 4.28 |
| System Optimization | 3.91 | 3.48 |
| Formal | 2.03 | 1.53 |
| Games | 1.09 | 1.07 |
| Knowledge | 1.65 | 1.57 |
| Optimization | 0.41 | 0.54 |
| Scientific & ML | 1.33 | 0.97 |
| Systems & SE | 1.87 | 1.82 |
| Macro average | 2.83 | 2.50 |

**From PC readouts to trajectories.** Figure 6 connects external loadings, category readouts, and growth in actual time. In the regularized Formal fit, PC2 enters the early-performance readout positively and the temporal readout negatively: its association with the starting level differs from its association with subsequent change. System Optimization illustrates how much of the observed gain occurs before the later evaluation checkpoints (Section B.4).

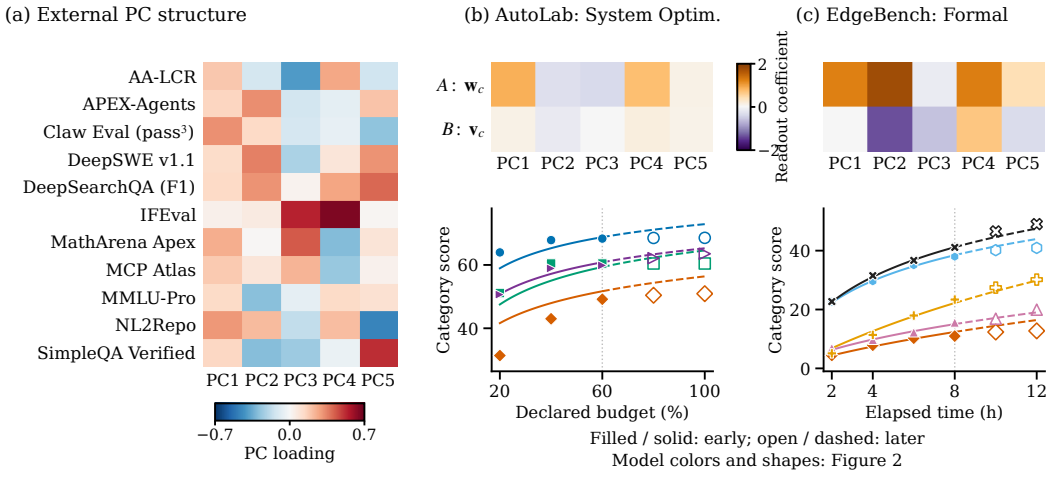



Figure 6: From external capabilities to time responses. (a) The union of the two strongest positive and negative loadings per PC shows 11 benchmarks; PC1 has only positive loadings. All 29 benchmarks enter the representation; full loadings appear in Figure 9. (b–c) Illustrative categories: top, fitted PC coefficients $\mathbf{w}_c, \mathbf{v}_c$ for $A, B$ (intercepts omitted); bottom, all 50 observed category-mean scores across four/five models on $12/8$ fixed tasks and their fitted curves. Filled/solid denotes early observations/fits; open/dashed denotes later observations/predictions. Both use the five-PC response of Figure 2. Loadings and readout coefficients use separate color scales.

### A.3 Category Response Estimation and Metrics

**Fixed trajectory panels.** We first retain task–model mean curves with observed scores at every analysis checkpoint in Table 4. Within each category, we enumerate model subsets of size at least four and intersect their available tasks. The primary panel maximizes the number of observed task–model pairs; ties favor more tasks, then lexicographic task and model identifiers. Selection uses availability, not score values. This yields the fixed panels in Table 7, shared by the temporal forecasts and gain-distribution analysis. AutoLab means average available replicates; EdgeBench supplies task–model means. The same tasks and models enter every checkpoint within a category. Continuation uses the separately defined replay population below (Appendix C.3).

Table 4: Temporal prediction protocol. AutoLab time is a fraction of declared budget; EdgeBench time is elapsed hours.

| Benchmark | Early coefficient fit | Later evaluation |
|---|---|---|
| AutoLab | $20\%, 40\%, 60\%$ | $80\%, 100\%$ |
| EdgeBench | $2, 4, 6, 8$ h | $10, 12$ h |

For category $c$, let $\mathcal{E}_c$ index its early checkpoints and $n_c = |\mathcal{M}_c||\mathcal{E}_c|$. The response parameters are $\theta_c = (a_c, b_c, \mathbf{w}_c, \mathbf{v}_c)$. Their fitting objective is

$$\mathcal{L}_c(\theta_c) = \frac{1}{n_c} \sum_{m \in \mathcal{M}_c} \sum_{k \in \mathcal{E}_c} \left[\widehat{q}_{mc}(\tau_k) - \bar{s}_{mc}(\tau_k)\right]^2 + \rho_c\big(\|\mathbf{w}_c\|_2^2 + \|\mathbf{v}_c\|_2^2\big) + 10^{-9} a_c^2 + 10^{-7} b_c^2. \quad (17)$$

The small intercept penalties stabilize the numerical fit. We compare the full response with its restriction $\mathbf{v}_c = \mathbf{0}$, and use $\rho_c \in \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. Each candidate is fitted before the last early checkpoint and scored at that checkpoint; ties prefer the restricted response and then stronger regularization. The selected specification is refitted on all early points. The temporal fit leaves $B_{mc}$ unconstrained; all increments used in the opportunity analysis are positive in the selected fits. The continuation reference instead imposes a positive slope by construction.

For the later model–checkpoint set $\mathcal{T}_c$, write $y_{mk} = \bar{s}_{mc}(\tau_k)$ and $\bar{y}_c = |\mathcal{T}_c|^{-1} \sum_{(m,k) \in \mathcal{T}_c} y_{mk}$. We report

$$\mathrm{MSE}_c = \frac{1}{|\mathcal{T}_c|} \sum_{(m,k) \in \mathcal{T}_c} [\widehat{q}_{mc}(\tau_k) - y_{mk}]^2, \qquad \mathrm{RMSE}_c = 100\sqrt{\mathrm{MSE}_c}, \quad (18)$$

$$R_c^2 = 1 - \frac{\sum_{(m,k) \in \mathcal{T}_c} [\widehat{q}_{mc}(\tau_k) - y_{mk}]^2}{\sum_{(m,k) \in \mathcal{T}_c} (y_{mk} - \bar{y}_c)^2}. \quad (19)$$

Macro metrics average category values equally, with the square root taken within each category for RMSE.

### A.4 Temporal Controls without External Scores

All controls use the identical fixed model–task panels, early checkpoints, score-space squared-error loss, and later evaluation points. The persistence control retains the last early score. Independent time curves fit $\sigma(A_m + B_m g(\tau))$ separately to each model's early category means, using only the numerical intercept and slope penalties in equation 17. The model-indicator control replaces the five PC coordinates with one-hot model indicators and uses the same early-only regularization and shared-versus-model-dependent slope selection as the capability response. Its final coefficients are refitted on all early points. Table 5 retains every category.

With four or five models per category, the intercept-plus-PC design has full row rank; regularization selects among coefficient vectors with identical fitted indices. The coefficients describe regularized associations, with predictive performance evaluated by the later-score controls in Table 5.

Table 5: Later-score RMSE in points under matched temporal evaluation. Indicators denotes regularized model identities; PCs denotes the selected five-PC response.

| Category | Last score | Independent | Indicators | PCs |
|---|---|---|---|---|
| CUDA | 4.19 | 4.95 | 4.18 | 3.75 |
| Model Development | 6.19 | 3.46 | 6.06 | 3.79 |
| Puzzle & Challenge | 5.83 | 4.27 | 4.41 | 4.08 |
| System Optimization | 1.71 | 3.97 | 3.41 | 3.46 |
| Formal | 4.40 | 1.86 | 1.66 | 1.93 |
| Games | 1.95 | 1.10 | 1.01 | 1.03 |
| Knowledge | 2.17 | 2.05 | 1.90 | 1.53 |
| Optimization | 1.11 | 0.58 | 0.50 | 0.57 |
| Scientific & ML | 2.25 | 1.04 | 1.50 | 1.17 |
| Systems & SE | 1.19 | 1.90 | 1.91 | 1.86 |
| Macro average | 3.10 | 2.52 | 2.66 | 2.31 |

# B Gain Decomposition and Opportunity Modeling

## B.1 The Coverage Identity and Its Change over Time

For positive gains, $M_c C$ is the order-two Hill effective number (Hill, 1973), $(\sum_m \Delta_m)^2 / \sum_m \Delta_m^2$; dividing by $M_c$ gives normalized gain coverage. Fix a task and interval, and suppress their indices. Let $K_+$ be the number of positive increments, $\bar{\Delta}_+$ their mean, and $v_+$ their variance with divisor $K_+$. For $K_+ > 0$,

$$C = \frac{(K_+ \bar{\Delta}_+)^2}{M_c K_+ (\bar{\Delta}_+^2 + v_+)} = \frac{K_+}{M_c} \frac{1}{1 + v_+ / \bar{\Delta}_+^2} = P\,U. \tag{20}$$

When $K_+ = 0$, we set $P = C = 0$ and leave the positive-gain balance undefined. For a task with positive gains in both its early and late intervals, denoted by subscripts $\mathrm{early}$ and $\mathrm{late}$, the exact symmetric decomposition is

$$\begin{aligned} C_{\mathrm{late}} - C_{\mathrm{early}} = (P_{\mathrm{late}} - P_{\mathrm{early}}) \frac{U_{\mathrm{late}} + U_{\mathrm{early}}}{2} \\ + (U_{\mathrm{late}} - U_{\mathrm{early}}) \frac{P_{\mathrm{late}} + P_{\mathrm{early}}}{2}. \end{aligned} \tag{21}$$

The two terms exactly decompose the observed coverage change into participation and balance components. We compute them per task, then average within category and across categories. The early and late intervals are 20–40% and 80–100% for AutoLab, and 2–4 h and 10–12 h for EdgeBench. Participation uses all 71 fixed tasks. The paired decomposition uses the 62 tasks with positive gains in both intervals, giving contributions of $-23.95$ percentage points from participation and $+2.92$ from balance. A nonparametric bootstrap (Efron, 1979) resamples whole tasks within categories, keeping the model set fixed, and gives a 95% interval of 29.5–40.8 percentage points for the macro participation decline. Raising the improvement threshold from numerical zero to 0.01 or 0.1 score points preserves the decline in all ten category means.

**Expanded trajectory panels.** Table 6 extends the analysis from the primary panels to all complete task–model curves available at the panel-selection stage. For each task $i$, participation divides the number of improving models by the number observed on that task; this model set stays fixed across the first and last intervals. We average tasks within categories, then the ten categories equally. The expansion increases coverage from 71 tasks and 329 task–model pairs to 80 tasks and 402 pairs, without imputing incomplete curves. Participation declines in every category under both definitions, averaging 35.2 and 33.5 percentage points, respectively.

Table 6: Participation declines under both panel definitions. Counts are tasks; decline is the task-averaged first-to-last reduction in percentage points. Expanded panels retain every available complete task–model curve, with each task's model set fixed across intervals.

| | Fixed primary panel | | Expanded panel | |
|---|---|---|---|---|
| Category | Tasks | Decline (pp) | Tasks | Decline (pp) |
| CUDA | 3 | 41.7 | 4 | 32.5 |
| Model Development | 2 | 50.0 | 2 | 42.5 |
| Puzzle & Challenge | 5 | 30.0 | 9 | 43.0 |
| System Optimization | 12 | 54.2 | 14 | 48.3 |
| Formal | 8 | 20.0 | 8 | 20.0 |
| Games | 8 | 27.5 | 8 | 27.5 |
| Knowledge | 4 | 40.0 | 4 | 40.0 |
| Optimization | 14 | 21.4 | 15 | 20.0 |
| Scientific & ML | 4 | 25.0 | 4 | 21.2 |
| Systems & SE | 11 | 41.8 | 12 | 40.4 |
| Ten-category mean | | 35.2 | | 33.5 |

### B.2 Opportunity Response and Later Calibration

For each model–category–interval observation, let $f_{mc,k}$ be its empirical fraction of tasks with positive increments. The opportunity model is logistic regression on log predicted mean gain:

$$\operatorname{logit} \widehat{\pi}_{mc,k} = \gamma\big(\log \widehat{\mu}_{mc,k} - \log \mu_c^{\star}\big). \tag{22}$$

In implementation, log gain is standardized on early observations; the reported scale and exponent absorb that standardization. A category intercept and one shared slope minimize binomial cross-entropy against $f_{mc,k}$, with equal total category weight and weights proportional to task counts within category. Only the standardized slope receives an L2 penalty, selected from $\{0, 0.001, 0.01, 0.1\}$ using the last early interval and category-macro Brier score (Brier, 1950). The slope is fitted without a sign constraint and is positive in the selected fit. Refitting the opportunity parameters under whole-task resampling gives a $95\%$ interval of $1.54$–$2.05$ for $\gamma$, conditional on the fixed capability curves and selected regularization.

Figure 7a uses the $20, 40, 60, 80$th percentiles of early log gain ratios as five-bin boundaries, extending the outer bins to include all later points. Each square places the geometric mean gain ratio against the unweighted mean observed task fraction within that bin. From left to right, the groups contain $60, 18, 5, 6, 1$ later observations; all $90$ enter these descriptive summaries. The orange segments connect adjacent group averages in increasing predicted-gain order; they do not represent a trajectory through time. The black response curve is unchanged.

Figure 4(b) evaluates later calibration with six bins defined by the early predicted-probability quantiles at $j/6$, $j = 1, \ldots, 5$, extending the outer boundaries to include all later predictions. Five bins contain later observations. Squares show task-count-weighted mean predicted probabilities and pooled observed improvement fractions. Figure 7b adds all $90$ model–category–interval fractions as individual points. The groups contain $51, 22, 9, 5, 3$ fractions, representing $372, 168, 64, 30, 24$ task–model–interval entries, respectively. Error bars are $95\%$ percentile intervals from $5{,}000$ whole-task bootstrap replicates within categories, retaining each sampled task's models and intervals; predictions and bin boundaries remain fixed.

**Matched input comparison.** To evaluate the input to equation 10, we replace log predicted gain with log interval midpoint, retaining category intercepts, a shared slope, the early fitting intervals, and the regularization-selection procedure. Both responses use these category intercepts and a single input, without model-specific intercepts. On all $90$ later model–category–interval observations, category-macro Brier scores are $0.2071$ for predicted gain and $0.2187$ for time. For an observed improving-task fraction $f$ and prediction $\widehat{\pi}$, the Brier contribution is $f(1-\widehat{\pi})^2 + (1-f)\widehat{\pi}^2$; this averages squared probability error over the tasks' binary improvement outcomes. We average contributions within each category and then across the ten categories.

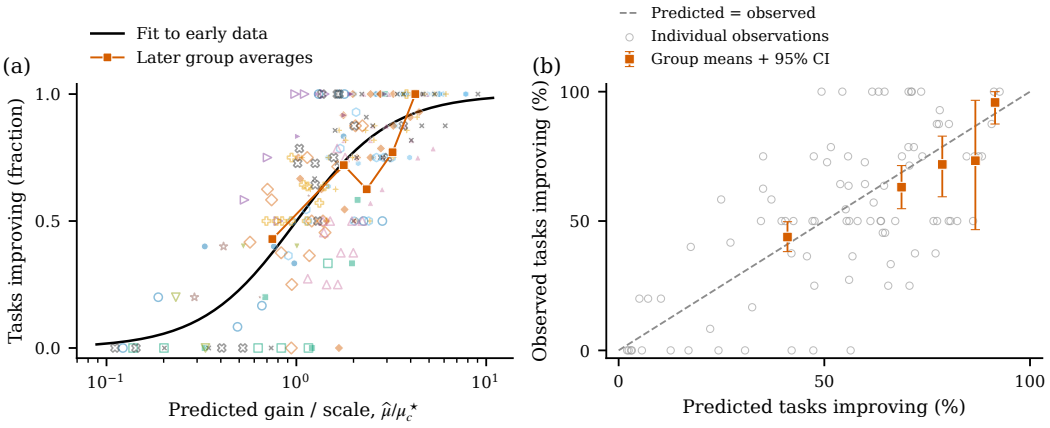



Figure 7: The gain–opportunity response and later calibration. (a) All 209 model–category–interval observations (filled: early; open: later). The black curve fits early improvement frequencies; orange squares connect later averages in five groups defined by early gain-ratio quantiles. (b) All 90 later fractions (gray), with the same task-weighted means and 95% whole-task bootstrap intervals as Figure 4(b); dashed line: predicted = observed.

### B.3 Task-Grounded Interpretation of Progress

The task-signature analysis relates the same 29 external measurements to observed trajectories on the fixed 71-task panel. Table 7 summarizes the work performed in each category. All external measurements named in the main-text cases have published scores for every model in their respective panel. Spearman $\rho_s$ is computed across the four or five models within a category, with ties assigned average ranks. These exploratory associations describe task- and stage-related performance patterns; the task counts do not constitute additional independent model samples.

Table 7: Work performed by the fixed task panels. Counts are models/tasks. The first four categories are AutoLab; the remaining six are EdgeBench. The machine-readable supplement retains all 71 task descriptions.

| Category | Models/tasks | Task process |
|---|---|---|
| CUDA | 4/3 | Optimize decoding, geometric correspondence, and elliptic-curve CUDA kernels. |
| Model Development | 4/2 | Select fine-tuning data and optimize an online serving engine. |
| Puzzle & Challenge | 4/5 | Reduce program or model complexity under correctness and accuracy constraints. |
| System Optimization | 4/12 | Improve implementations of cryptography, retrieval, storage, and numerical kernels. |
| Formal | 5/8 | Complete interdependent Lean/Coq proofs in multi-file projects. |
| Games | 5/8 | Explore interactive fiction or implement game-playing and management agents. |
| Knowledge | 5/4 | Build three domain-specific systems and produce one examination-document suite. |
| Optimization | 5/14 | Implement and revise solvers for routing, packing, constraints, and search. |
| Scientific & ML | 4/4 | Implement and evaluate reinforcement learning, inverse models, and graph learning. |
| Systems & SE | 5/11 | Implement repositories and specified features; optimize throughput and kernels. |

For a terminal window containing $w$ intervals and an improvement threshold $\epsilon$ in score points, define

$$f_{mc}^{(w,\epsilon)} = \frac{1}{w|\mathcal{I}_c|} \sum_{k=N-w}^{N-1} \sum_{i \in \mathcal{I}_c} \mathbf{1}\{100\Delta_{mi,k} > \epsilon\}. \tag{23}$$

Thus each interval contributes its fraction of tasks improving, and the selected intervals receive equal weight. The main setting uses $w = 2$ and positive gain. Sensitivity checks cross $w \in \{1, 2, 3\}$ with $\epsilon \in \{0, 0.1, 0.5\}$, retaining every task and model. The three suffixes cover 80–100%, 60–100%, and 40–100% for AutoLab, and 10–12, 8–12, and 6–12 hours for EdgeBench. They overlap; the three-interval suffix can contain fitted checkpoints. This is a descriptive window check, with the temporal fitting protocol unchanged.

**Aggregate profiles across all external measurements.** Table 8 averages the signed Spearman correlations over all 29 benchmarks within each category, then across categories with equal weight. Final scores have a mean association of $0.454$, compared with $0.186$ for terminal improvement frequency, a difference of $0.268$. The difference is positive in seven categories; Games, Scientific & ML, and Systems & SE have more positive terminal-frequency associations. The aggregate contrast motivates separate profiles for attainment and continued improvement, with the constituent benchmarks varying by task category.

The difference remains positive in four descriptive sensitivity checks. Removing each model from every category in which it appears and recomputing the correlations gives differences of $0.089$–$0.355$. The nine window–threshold combinations above give $0.268$–$0.432$. Giving equal weight to distinct benchmark rankings within each category yields $0.202$. Retaining only benchmarks with published scores for every model in the category yields $0.279$, using 9–27 measurements per category. These checks preserve equal category weighting; undefined correlations from constant rankings are omitted within a category. They measure the stability of the observed aggregate contrast across model membership and measurement choices.

Table 8: Attainment and terminal improvement have different external benchmark profiles. Each entry averages signed model-rank correlations over all 29 measurements within a category; the final row averages categories equally. Difference is final-score minus terminal-frequency association.

| Category | Final score | Terminal frequency | Difference |
|---|---|---|---|
| CUDA | 0.290 | −0.222 | +0.511 |
| Model Development | 0.250 | −0.250 | +0.500 |
| Puzzle & Challenge | 0.473 | −0.364 | +0.837 |
| System Optimization | 0.304 | −0.051 | +0.354 |
| Formal | 0.671 | 0.171 | +0.499 |
| Games | 0.444 | 0.601 | −0.156 |
| Knowledge | 0.538 | 0.259 | +0.279 |
| Optimization | 0.616 | 0.392 | +0.224 |
| Scientific & ML | 0.339 | 0.657 | −0.319 |
| Systems & SE | 0.616 | 0.671 | −0.055 |
| Ten-category mean | 0.454 | 0.186 | +0.268 |

**Composite profiles and their constituent measurements.** Tables 9 and 11 list every benchmark with $|\rho_s| \geq 0.70$ under the stated outcome, grouping equal rounded correlations for readability. This cutoff indexes the descriptive scan; it neither selects predictor inputs nor limits the number of profile members. All 290 category–benchmark pairs, their nine-condition results, published-score masks, and within-panel benchmark rank equivalences are retained in the numerical supplement. For example, Optimization's NL2Repo, SWE-bench Pro, and Claw Eval share one ranking, while APEX Agents and CyberGym share another. Their different task content broadens the interpretation of the profile; the benchmark count is not an independent-sample count.

Optimization's eight-member profile spans repository construction, agent execution, and information seeking. Seven members stay positive in all nine conditions; DeepSWE is positive in seven. In Systems & SE, final-score correlations are $0.90$ for NL2Repo, SWE-bench Pro, Claw Eval, Deep-

Table 9: Composite profiles in the main-text categories. All external measurements with $|\rho_s| \geq 0.70$ for terminal improvement frequency are listed, with no cap on profile size. Parentheses give Spearman correlations; category counts are models/tasks. The default uses the last two intervals and positive gain; the additional increment threshold is stated explicitly. An asterisk identifies a benchmark with at least one frozen estimated input in that panel. Correlated members describe a joint performance profile, not separate effects.

| Category | Concrete benchmark members and observed associations |
|---|---|
| Systems & SE (5/11) | BrowseComp, FrontierScience-Olympiad*, HLE, IMO-AnswerBench, SimpleQA, SWE-bench Multilingual, Terminal-Bench, Toolathlon (+0.70); AIME 2026, DeepSWE, SciCode (+0.80); HMMT Feb 2026, HMMT Nov 2025 (+0.82); MathArena, $\tau^3$-Banking (+0.90); CritPt, FrontierScience-Research, GPQA (+1.00) |
| Optimization (5/14) | DeepSWE, DeepSearchQA (+0.72); APEX Agents, CyberGym, MCPAtlas (+0.82); Claw Eval, NL2Repo, SWE-bench Pro (+0.97) |
| Knowledge (5/4) | AA-LCR (-0.74); APEX Agents, CyberGym, MCPAtlas (+0.74); Claw Eval, NL2Repo, SWE-bench Pro (+0.95) |
| System Optimization (4/12) | MathArena, MMLU-Pro (-1.00); GPQA, SciCode, SimpleQA (-0.80); AA-LCR, HMMT Nov 2025* (-0.74); IMO-AnswerBench (+1.00) |

SWE, Terminal-Bench, and SWE-bench Multilingual. The first three have terminal-frequency correlations of $0.30$, while DeepSWE remains at $0.80$. Scientific reasoning (GPQA, CritPt, FrontierScience Research), mathematics (MathArena, AIME), and scientific programming (SciCode) are all positive across the nine conditions. This is an overlapping capability profile, with distinct associations for attainment and continued improvement.

Formal exhibits increment-size dependence within the same terminal window. AA-LCR correlations are $0.80$, $0.87$, and $0.67$ at thresholds $0$, $0.1$, and $0.5$ points. At $> 0.5$ points, AIME, SciCode, and $\tau^3$-Banking each correlate at $0.97$; HMMT February at $0.95$; GPQA, CritPt, and FrontierScience Research at $0.87$. Thus long-input integration, mathematical/scientific reasoning, programming, and multi-step execution provide complementary task interpretations. The thresholds distinguish observed increment sizes, not time stages or proof difficulty.

Table 10 provides individual-member window checks. These retain the original measurements and relate task processes to their observed progress patterns.

Table 10: Window and gain-threshold sensitivity of the representative associations. Entries are Spearman correlations across fixed model panels with the fraction of tasks improving, averaged over the indicated final intervals. Last two is the main window; thresholds are in 0–100 score points. The first five rows concern AutoLab, the remainder EdgeBench. Every displayed external measurement is published for all models in its panel.

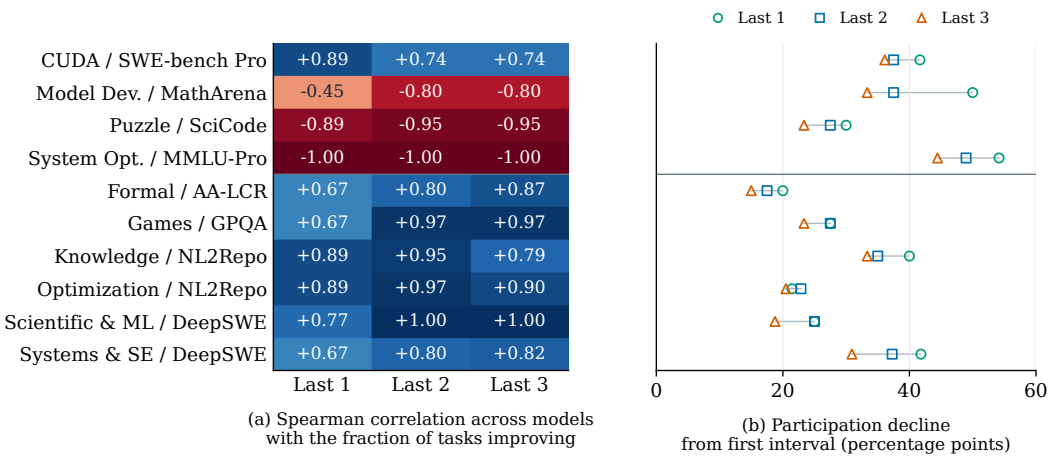
| Category | Benchmark | $100\Delta > 0$ | | | $100\Delta > 0.1$ | | | $100\Delta > 0.5$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Last 1 | Last 2 | Last 3 | Last 1 | Last 2 | Last 3 | Last 1 | Last 2 | Last 3 |
| CUDA | SWE Pro | $+0.89$ | $+0.74$ | $+0.74$ | $+0.74$ | $+0.60$ | $+0.60$ | $+0.74$ | $+0.60$ | $+0.60$ |
| Model Dev. | MathArena | $-0.45$ | $-0.80$ | $-0.80$ | $-0.45$ | $-0.80$ | $-0.80$ | $-0.45$ | $-0.63$ | $-0.40$ |
| Puzzle | SciCode | $-0.89$ | $-0.95$ | $-0.95$ | $-0.89$ | $-0.95$ | $-0.95$ | $-0.89$ | $-0.95$ | $-0.77$ |
| System Opt. | MMLU-Pro | $-1.00$ | $-1.00$ | $-1.00$ | $-1.00$ | $-1.00$ | $-1.00$ | $-0.95$ | $-1.00$ | $-1.00$ |
| System Opt. | AA-LCR | $-0.74$ | $-0.74$ | $-0.74$ | $-0.74$ | $-0.74$ | $-0.74$ | $-0.89$ | $-0.74$ | $-0.74$ |
| Formal | AA-LCR | $+0.67$ | $+0.80$ | $+0.87$ | $+0.67$ | $+0.87$ | $+0.82$ | $+0.67$ | $+0.67$ | $+0.60$ |
| Games | GPQA | $+0.67$ | $+0.97$ | $+0.97$ | $+0.89$ | $+0.97$ | $+1.00$ | $+0.90$ | $+0.90$ | $+0.97$ |
| Knowledge | NL2Repo | $+0.89$ | $+0.95$ | $+0.79$ | $+0.53$ | $+0.67$ | $+0.74$ | $+0.53$ | $+0.36$ | $+0.82$ |
| Optimization | NL2Repo | $+0.89$ | $+0.97$ | $+0.90$ | $+0.67$ | $+0.40$ | $+0.82$ | $+0.58$ | $+0.50$ | $+0.80$ |
| Scientific/ML | DeepSWE | $+0.77$ | $+1.00$ | $+1.00$ | $+0.00$ | $+0.63$ | $+0.80$ | $-0.77$ | $+0.89$ | $+0.74$ |
| Systems/SE | DeepSWE | $+0.67$ | $+0.80$ | $+0.82$ | $+0.67$ | $+0.67$ | $+0.56$ | $+0.70$ | $+0.36$ | $+0.31$ |
| Systems/SE | SWE Pro | $+0.10$ | $+0.30$ | $+0.41$ | $+0.10$ | $+0.10$ | $+0.10$ | $+0.30$ | $-0.21$ | $-0.21$ |

**Shared processes and additional task profiles.** Knowledge's six positive members combine repository engineering (NL2Repo, SWE-bench Pro) with execution reliability and tool-mediated work (Claw Eval, APEX Agents, CyberGym, MCPAtlas); all remain positive across the nine conditions. This profile matches three system-building tasks among four deliverables. Games combines reasoning, search, and interaction across thirteen published positive members, all positive in the nine conditions; its three interactive-fiction tasks and five programmed agents share exploration while differing in implementation demands. Scientific & ML has seventeen published positive members spanning mathematics, science, implementation, and execution. Its profile depends on increment size and window: DeepSWE reverses to $-0.77$ for final-interval increments above $0.5$ points. CUDA and Model Development retain their engineering–tool profiles and opposing mathematical or knowledge associations in Table 11; their three- and two-task panels also admit task-level interpretation. Puzzle & Challenge's published negative members, SciCode, GPQA, and CritPt, all correlate positively with initial performance ($0.80$) and negatively with terminal frequency ($-0.95$).

Table 11: Composite profiles in the remaining six categories. All external measurements with $|\rho_s| \geq 0.70$ for terminal improvement frequency are listed, with no cap on profile size. Parentheses give Spearman correlations; category counts are models/tasks. The default uses the last two intervals and positive gain; the additional increment threshold is stated explicitly. An asterisk identifies a benchmark with at least one frozen estimated input in that panel. Correlated members describe a joint performance profile, not separate effects.

| Category | Concrete benchmark members and observed associations |
|---|---|
| Formal (5/8) | $\tau^3$-Banking (+0.70); AA-LCR (+0.80)<br>$> 0.5$ points: CritPt, FrontierScience-Research, FrontierScience-Olympiad*, GPQA (+0.87); HMMT Feb 2026 (+0.95); AIME 2026, SciCode, $\tau^3$-Banking (+0.97) |
| Games (5/8) | HMMT Nov 2025 (+0.71); AA-LCR, AIME 2026, SciCode (+0.72); HMMT Feb 2026 (+0.76); BrowseComp, IMO-AnswerBench, MathArena, SimpleQA (+0.82); $\tau^3$-Banking (+0.87); CritPt, FrontierScience-Research, GPQA (+0.97) |
| Scientific & ML (4/4) | APEX Agents, BrowseComp, CritPt, CyberGym, DeepSearchQA, FrontierScience-Research, GPQA, HLE, MMLU-Pro, SimpleQA, SWE-bench Multilingual, SWE-bench Verified, Terminal-Bench, Toolathlon (+0.80); HMMT Nov 2025 (+0.95); DeepSWE, MathArena (+1.00) |
| CUDA (4/3) | HMMT Feb 2026, MathArena (-0.95); BrowseComp*, FrontierScience-Olympiad, IFEval*, SimpleQA (-0.74); MCPAtlas, SWE-bench Multilingual, SWE-bench Pro (+0.74) |
| Model Development (4/2) | FrontierScience-Research, HMMT Feb 2026, SimpleQA (-1.00); BrowseComp*, MathArena (-0.80); SWE-bench Multilingual (+0.80)<br>$> 0.5$ points: BrowseComp*, FrontierScience-Research, HMMT Feb 2026, SimpleQA (-0.95); MCPAtlas, SWE-bench Multilingual, SWE-bench Pro (+0.74) |
| Puzzle & Challenge (4/5) | AIME 2026*, CritPt, FrontierScience-Olympiad*, GPQA, HMMT Feb 2026*, MathArena*, MMLU-Pro*, SciCode, SimpleQA* (-0.95) |

Across all three windows, category-mean participation and effective gain coverage remain below their first-interval values in every category. For the last one, two, and three intervals, macro participation declines are $35.2$, $31.7$, and $27.9$ percentage points; coverage declines are $23.6$, $20.5$, and $18.2$ points. Coverage is computed per task and interval before averaging, retaining all-zero cases as zero.




(a) Spearman correlation across models with the fraction of tasks improving

(b) Participation decline from first interval (percentage points)

Figure 8: Window sensitivity on all ten categories. (a) Benchmark correlations with the equally weighted mean fraction of tasks improving across the selected final intervals, using positive gain. The displayed pairs come from the task-interpretation analysis. (b) Task-averaged model participation decline relative to the first interval, in percentage points. The horizontal rule separates AutoLab from EdgeBench; model/task counts follow Table 7. Last two is the main window. All tasks and models are retained, and the response models are unchanged.

### B.4 Gain Realization over the Observed Trajectory

To describe gain timing without an early/late cutoff, normalize the observed window by $u_k = (\tau_k - \tau_0)/(1 - \tau_0)$ and let $\overline{\Delta}_{mc,k} = \bar{s}_{mc}(\tau_{k+1}) - \bar{s}_{mc}(\tau_k)$. The gain-weighted interval midpoint is

$$\ell_{mc}^{\text{gain}} = \frac{\sum_{k=0}^{N-1} \overline{\Delta}_{mc,k}(u_k + u_{k+1})/2}{\sum_{k=0}^{N-1} \overline{\Delta}_{mc,k}}. \tag{24}$$

Smaller values indicate earlier realization of the model's observed gains; constant-rate improvement gives 0.5. The statistic locates gains at recorded interval midpoints rather than asserting within-interval event times. It is defined for 44 of the 45 model–category pairs. GPT-5.5 has zero total gain in Model Development, leaving three effective models for that category's timing correlations. Supplementary tables report effective counts and attainment correlations on the same effective model set.

Table 12: Gain realization on all four models in AutoLab System Optimization (12 tasks). Scores use the 0–100 scale. The fourth column is the fraction of each model's observed 20–100% gain realized by 60% budget. The timing index $\ell^{\text{gain}}$ weights normalized interval midpoints by category mean-score increments; smaller values indicate earlier gains.

| Model | First score | Final score | Gain by 60% | $\ell^{\text{gain}}$ |
|---|---|---|---|---|
| Opus 4.7 | 63.96 | 68.55 | 94.8% | 0.179 |
| DeepSeek V4 Pro | 31.40 | 50.92 | 91.0% | 0.255 |
| Gemini 3.1 Pro | 51.13 | 60.45 | 100.0% | 0.125 |
| GLM 5.2 | 50.63 | 63.52 | 71.5% | 0.313 |

Table 12 reports all four models in System Optimization. The same six published benchmarks associate negatively with both terminal improvement frequency and $\ell^{\text{gain}}$: AA-LCR at $-0.74$, GPQA/SciCode/SimpleQA at $-0.80$, and MMLU-Pro/MathArena at $-1.00$. All six retain negative terminal-frequency associations in every window–threshold condition. Their first-score correlations differ: AA-LCR is 0.95, GPQA/SciCode 0.80, MMLU-Pro/MathArena 0.60, and SimpleQA 0.00. The shared profile thus concerns gain timing, without requiring identical initial-attainment relationships. The fraction of observed 20–100% gain realized by 60% budget ranges from 71.5% to 100%. IMO-AnswerBench retains the opposite timing association (1.00), distinguishing this profile from

a uniform mathematics effect. Puzzle & Challenge provides a second example: SciCode correlates positively with initial performance (0.80) and negatively with $\ell^{\text{gain}}$ ($-0.80$). Timing uses complete observed trajectories for retrospective interpretation; continuation conditions on the currently visible prefix.

### B.5 Joint Readouts and Observed Benchmark Profiles

The joint predictor combines all 29 inputs through the five PCs. Expanding its readouts gives $\boldsymbol{\beta}_c^A = \mathbf{V}_5\mathbf{w}_c$ and $\boldsymbol{\beta}_c^B = \mathbf{V}_5\mathbf{v}_c$ in equation 7; the numerical supplement supplies both full vectors. Direct PC evaluation and this expansion agree within $9 \times 10^{-16}$ in the fitted indices. The joint readouts predict growth, while empirical benchmark profiles characterize task demands and gain timing through associations with observed outcomes. These profiles are descriptive associations rather than independent benchmark effects; larger-increment profiles in Table 11 concern a different outcome from the opportunity response's any-positive improvement.

## C Continuation Fitting and the Sequential Value Bound

### C.1 The Monotone Decision Reference

The decision reference uses the same external coordinates and log-time functional form as the category response, with an analysis-specific first checkpoint. It is refitted on source-model histories with a positive slope:

$$A_{mc}^{\text{ref}} = a_c^{\text{ref}} + (\mathbf{w}_c^{\text{ref}})^{\mathsf{T}}\mathbf{z}_m, \qquad B_{mc}^{\text{ref}} = \text{softplus}\big(b_c^{\text{ref}} + (\mathbf{v}_c^{\text{ref}})^{\mathsf{T}}\mathbf{z}_m\big), \tag{25}$$

$$r_{mc}(\tau) = \sigma\big(A_{mc}^{\text{ref}} + B_{mc}^{\text{ref}} g(\tau)\big). \tag{26}$$

Here $\text{softplus}(u) = \log(1 + e^u)$ guarantees $B_{mc}^{\text{ref}} > 0$. AutoLab uses $\tau_0 = 0.05$ for the first declared decision checkpoint and $\tau_0 = 0.2$ for category forecasting; EdgeBench uses $\tau_0 = 2/12$ for both. These decision grids and source-history fits are distinct from the early-checkpoint category forecasts in Table 4. The fitted category response $\widehat{q}$ and the decision reference $r$ share their external representation and functional structure, with coefficients learned under their respective information conditions.

### C.2 Source-Only Estimation

We fit a separate continuation model within each benchmark suite. Source category means average replicates within a model–task–checkpoint and then average available tasks within category. The monotone reference in equation 26 minimizes score squared error with equal total model weight within category and coefficient penalty 0.01 on PC readouts; intercepts are unpenalized. Reference fits use completed source histories, including declared endpoints. Missing earlier observations are not filled with future scores.

Let $\widetilde{\mathbf{h}}_k \in \mathbb{R}^7$ contain standardized budget progress, current and initial scores, last-interval and last-two-interval gains, observed-prefix improvement rate, and trailing-stagnation fraction. The positive response coefficients are

$$\log \alpha_k = u_{\alpha,0} + \mathbf{u}_\alpha^{\mathsf{T}}\widetilde{\mathbf{h}}_k + d_i^\alpha, \qquad \log \beta_k = u_{\beta,0} + \mathbf{u}_\beta^{\mathsf{T}}\widetilde{\mathbf{h}}_k + d_i^\beta. \tag{27}$$

Within each suite, the weights are shared across source models and tasks. Regularized task effects $d_i^\alpha, d_i^\beta$ are learned from other models' histories on task $i$. We jointly fit the next-checkpoint (n) and endpoint (e) gains in completed source runs. For $H_k > 0$, their normalized targets are

$$Z_{k,\text{n}} = \frac{Y_{k+1} - Y_k}{H_k}, \qquad Z_{k,\text{e}} = \frac{Y_N - Y_k}{H_k}, \tag{28}$$

Both targets are zero when no score space remains. Equal-weight squared errors on these two fractions train the same curve, including zero-gain histories, with parameter regularization and source-only model selection.

Index active training states by $j$ and let $\boldsymbol{\phi}_j$ concatenate an intercept, the seven source-standardized state features, and task indicators multiplied by $\sqrt{n_{\text{task}}}$, where $n_{\text{task}}$ is the number of source tasks.

Let $\boldsymbol{\theta}_\alpha, \boldsymbol{\theta}_\beta$ be the corresponding coefficient vectors. This design implements equation 27; its task effects satisfy $d_i^\alpha = \sqrt{n_{\text{task}}}\,\theta_{\alpha,i}$ and likewise for the shape. Write $F_{j,h} = 1 - \exp[-\alpha_j D_j(\tau_{j,h})^{\beta_j}]$. The objective is

$$\mathcal{L}_{\text{gain}} = \frac{1}{2}\sum_j \nu_j \sum_{h\in\{\text{n,e}\}} (F_{j,h} - Z_{j,h})^2 + \rho\left(\|\boldsymbol{\theta}_{\alpha,\text{pen}}\|_2^2 + \|\boldsymbol{\theta}_\beta\|_2^2\right). \tag{29}$$

The subscript pen denotes the intensity coefficients excluding the intercept; all shape coefficients are penalized. The fixed-shape case sets $\boldsymbol{\theta}_\beta = \mathbf{0}$. State weights $\nu_j$ are inverse trajectory lengths, rescaled so that each source model has equal total mass and their mean over states is one. The observed-prefix improvement-rate feature divides the number of positive observed increments by the number of observed intervals; the stagnation feature uses the trailing number of zero-gain intervals with the same denominator. Both are zero when no interval has yet been observed. Active full-score states remain in training with zero normalized targets. Numerically, remaining score space is floored at $10^{-10}$ for division and rescaling, and the decision uses a positive-value tolerance of $10^{-10}$; full-score predictions are therefore at most this tolerance and stop. Naturally ended states return exactly zero gain.

For each outer target-model holdout, we consider $\rho \in \{1, 10, 100\}$ and learned or fixed shape. Each candidate is evaluated by leaving out source models in turn. Its source-only selection score averages disagreement between the signs of $\max_h \widehat{Q}_{k,h}$ and the realized value $\max_h[H_k Z_{k,h} - \lambda c_{k,h}]$, weighted by the absolute realized value and averaged across prices. This is a realized-label selection criterion; the conditional-mean error in the theorem below is a different quantity. Ties prefer stronger regularization and then fixed shape. To construct each source training state's reference, we further exclude that state's model from reference fitting. The target reference is refitted using all outer source models. Source predictions used for price calibration repeat the model selection with each calibration model excluded.

### C.3 Replay Metrics and Operating Points

**Replay population and visible prefixes.** AutoLab retains runs with a verified natural or budget termination, complete recorded evaluation outputs, a valid elapsed duration, and explicit initial-score feedback. A task–model pair enters replay when all three replicate runs meet these conditions, yielding 492 runs. Decisions start at the first declared checkpoint at or after initial feedback becomes visible; each prefix score uses only evaluations recorded by that checkpoint. Runs without an available continuation checkpoint contribute terminal accounting only. Naturally terminated runs retain their final score and incur no further cost. EdgeBench retains the 252 published task–model mean curves with all six two-hour checkpoints. Missing earlier observations are never filled with later scores. Replay retains eligible pairs without requiring a common task intersection across all models.

AutoLab continuation uses checkpoints at 5%, 10%, 15%, 25%, 50%, 75%, and 100% of declared budget. EdgeBench uses two-hour checkpoints from 2 to 12 hours. The price grid contains zero and 61 logarithmically spaced positive values from $10^{-5}$ to 30 on the $[0, 1]$ score scale, plus a run-to-end option.

For model $m$ and replicate $r$, cost $x$ is the sum of charged task hours divided by the sum of full-run hours for the same tasks, including the initial observation. Loss is the task-mean score shortfall in points. We retain nondominated parameter-sweep points and linearly interpolate the frontier $\ell_{mr}^p(x)$ for policy $p$. Let $[a_{mr}, b_{mr}]$ be the reachable cost interval common to the compared policies within that model–replicate pair. The reported normalized frontier area is

$$\mathcal{A}^p = \frac{1}{|\mathcal{M}|}\sum_{m\in\mathcal{M}} \frac{1}{R_m}\sum_{r=1}^{R_m} \frac{1}{b_{mr} - a_{mr}} \int_{a_{mr}}^{b_{mr}} \ell_{mr}^p(x)\,dx, \tag{30}$$

Here $\mathcal{M}$ is the suite's evaluated model set, with $R_m = 3$ for AutoLab and $R_m = 1$ for EdgeBench's published mean trajectories. Dividing by interval width gives an average shortfall in score points; averaging then proceeds over replicates and models. Figure 5a–b displays mean curves on the intersection of these intervals across models and replicates.

Table 1 reads matched operating points from these mean curves. At time savings $s \in \{0.30, 0.50\}$, retained score is $100[1 - \bar{\ell}^p(1-s)/\bar{S}_{\text{full}}]$, where $\bar{\ell}^p$ averages the interpolated replicate curves within

models and then models. At retained fractions $q \in \{0.95, 0.90\}$, we find the smallest cost $x$ satisfying $\bar{\ell}^p(x) \leq (1-q)\bar{S}_{\text{full}}$ and report $100(1-x)$. All crossings lie within the shared cost range; no extrapolation is used. These parameter-sweep comparisons summarize the frontiers, whereas Table 15 evaluates source-calibrated prices frozen for each target.

Table 13: Mean score loss over matched cost ranges (score points; lower is better). Ranges are shared across policies for each model and replicate; results average replicates then models.

| Policy | AutoLab | EdgeBench |
|---|---|---|
| Fixed budget | 14.384 | 2.442 |
| Time-based patience | 19.372 | 3.233 |
| Recent gain (1 interval) | 22.907 | 1.946 |
| Recent gain (2 intervals) | 24.271 | 1.982 |
| Conditioned continuation | **8.825** | **1.774** |

Table 14: Matched tradeoffs on the mean frontiers in Figure 5. For each suite: full-run score retained at $30\%/50\%$ time savings, and time saved at $95\%/90\%$ score retention. All entries are percentages; higher is better, best at displayed precision in bold. Values use interpolation within shared cost ranges.

| | **AutoLab** | | | | **EdgeBench** | | | |
|---|---|---|---|---|---|---|---|---|
| | Score retained | | Time saved | | Score retained | | Time saved | |
| Method | 30% | 50% | 95% | 90% | 30% | 50% | 95% | 90% |
| Fixed budget | 89.9 | 77.1 | 17.2 | 29.7 | 94.7 | 89.5 | 28.5 | 48.2 |
| Time-based patience | 81.3 | 65.9 | 8.2 | 18.0 | 92.8 | 84.9 | 24.6 | 37.2 |
| Recent gain (1 interval) | 75.1 | 58.6 | 6.0 | 12.1 | 96.2 | 91.6 | 38.1 | 54.9 |
| Recent gain (2 intervals) | 72.6 | 54.7 | 5.5 | 10.9 | 96.2 | **91.8** | 34.9 | 54.6 |
| Conditioned continuation | **97.3** | **90.5** | **38.1** | **50.9** | **96.8** | **91.8** | **38.4** | **55.9** |

For operating points, candidate nonnegative prices and a run-to-end option are replayed on nested source predictions. For each reported source budget target $(25\%, 37.5\%, 50\%, 62.5\%, 75\%)$, selection minimizes source score shortfall among candidates whose mean declared cost is within that preference; if none is feasible, it uses the least-cost candidate. Ties prefer higher cost and then the smaller price parameter. The selected option is frozen for the target. Across the seven AutoLab source panels, running every trajectory to its recorded end uses $52.06\%$–$62.45\%$ of declared budget on average. Run-to-end therefore satisfies both the $62.5\%$ and $75\%$ source constraints with zero score loss and is selected for every target model; the other reported settings select nonnegative prices. Fixed-budget stopping uses the same preference as a declared cutoff, stopping at the last decision checkpoint no later than that cutoff. A desired source budget fraction need not equal the realized target cost fraction. All costs entering the rule are normalized by the declared budget of the run; normalization by eventual full-run duration is used only for reporting realized costs. EdgeBench replay uses published task–model mean curves; AutoLab uses individual replicates.

Relative score loss is $100L/\bar{S}_{\text{full}}$, where $L$ is the reported mean loss in points and $\bar{S}_{\text{full}}$ is the mean full-run score under the same task, replicate, and model averaging. The full-run means are $63.78$ points on AutoLab and $35.68$ on EdgeBench. Time saved is $100(1-\bar{x})$, where $\bar{x}$ is the reported mean actual-cost fraction.

### C.4 Sequential Reassessment with Imperfect Gain Estimates

Consider a finite declared grid $\tau_0 < \cdots < \tau_N = 1$, a bounded best-so-far potential score process $Y_k$, and an increasing filtration $\mathcal{F}_k$ containing its visible prefix and fixed source information. Stopping truncates this potential trajectory; natural termination is absorbing. The price $\lambda \geq 0$ is fixed before the target run. Purchased intervals have additive declared costs $c_k = \tau_{k+1} - \tau_k$. A policy's continue indicator $a_k \in \{0, 1\}$ is measurable with respect to $\mathcal{F}_k$, with forced stopping at natural termination and $N$.

Table 15: Frozen operating points. Each target supplies the conditioned policy's source budget constraint and the fixed policy's declared cutoff. Cost is a fraction of full-run time. Loss is in points, with relative score loss in parentheses. Results average replicates within model, then models.

| Target | Policy | AutoLab | | EdgeBench | |
|---|---|---|---|---|---|
| | | Cost | Loss (pt; %) | Cost | Loss (pt; %) |
| 25% | Fixed budget | 46.3% | 15.77 (24.73%) | 16.7% | 10.12 (28.36%) |
| | Conditioned | 46.5% | 6.99 (10.96%) | 23.3% | 7.66 (21.46%) |
| 37.5% | Fixed budget | 46.3% | 15.77 (24.73%) | 33.3% | 5.95 (16.68%) |
| | Conditioned | 67.2% | 1.52 (2.39%) | 35.5% | 4.96 (13.90%) |
| 50% | Fixed budget | 70.1% | 5.34 (8.38%) | 50.0% | 3.74 (10.49%) |
| | Conditioned | 89.0% | 0.12 (0.19%) | 44.6% | 3.47 (9.71%) |
| 62.5% | Fixed budget | 70.1% | 5.34 (8.38%) | 50.0% | 3.74 (10.49%) |
| | Conditioned | 100.0% | 0.00 (0.00%) | 60.7% | 1.87 (5.25%) |
| 75% | Fixed budget | 87.2% | 2.10 (3.30%) | 66.7% | 2.14 (6.01%) |
| | Conditioned | 100.0% | 0.00 (0.00%) | 69.1% | 1.19 (3.34%) |

For $k < N$, define the true conditional gain means and fixed-plan values as

$$g_{k,\mathrm{n}} = \mathbb{E}[Y_{k+1} - Y_k \mid \mathcal{F}_k], \qquad Q_{k,\mathrm{n}} = g_{k,\mathrm{n}} - \lambda c_k, \tag{31}$$

$$g_{k,\mathrm{e}} = \mathbb{E}[Y_N - Y_k \mid \mathcal{F}_k], \qquad Q_{k,\mathrm{e}} = g_{k,\mathrm{e}} - \lambda(1 - \tau_k), \tag{32}$$

$$Q_k^{\max} = \max(Q_{k,\mathrm{n}}, Q_{k,\mathrm{e}}), \qquad J_k = \max(0, Q_k^{\max}). \tag{33}$$

At the endpoint set both $Q$ values to zero. At earlier absorbing states, the score gains are zero and the virtual endpoint value remains $-\lambda(1 - \tau_k)$; $J_k = 0$. These virtual costs retain the fixed-plan definition and are not charged to the stopped policy.

**Proposition 1** (Fixed-plan bound with sign loss)**.** *Let $V_k^\pi$ be the conditional expected score gain from checkpoint $k$ until policy stopping, minus $\lambda$ times the sum of purchased declared interval costs. At a visited checkpoint, define*

$$L_k = J_k - a_k Q_k^{\max} \geq 0. \tag{34}$$

*The nonnegative sign loss is zero for a correct continue/stop decision. If $\mathcal{V}_k^\pi$ is the set of visited checkpoints starting at $k$, including the checkpoint where stopping is selected, then*

$$V_k^\pi \geq J_k - \mathbb{E}_\pi \left[ \sum_{j \in \mathcal{V}_k^\pi} L_j \,\middle|\, \mathcal{F}_k \right]. \tag{35}$$

*In particular, if $a_j = \mathbf{1}\{Q_j^{\max} > 0\}$ at every visited checkpoint $j$, then $V_k^\pi \geq J_k$.*

*Proof.* Let $\mathcal{D}_k^\pi$ denote the conditional expected loss sum in equation 35. At the endpoint and natural termination, $V_k^\pi = J_k = \mathcal{D}_k^\pi = 0$. For an active checkpoint, additivity and the tower property yield

$$Q_{k,\mathrm{e}} = Q_{k,\mathrm{n}} + \mathbb{E}[Q_{k+1,\mathrm{e}} \mid \mathcal{F}_k]. \tag{36}$$

Proceed backward in $k$. If $a_k = 0$, the policy stops, $V_k^\pi = 0$, $L_k = J_k$, and there are no subsequent visits, giving equality. If $a_k = 1$, the induction hypothesis implies

$$\begin{aligned} V_k^\pi &= Q_{k,\mathrm{n}} + \mathbb{E}[V_{k+1}^\pi \mid \mathcal{F}_k] \\ &\geq Q_{k,\mathrm{n}} + \mathbb{E}[J_{k+1} - \mathcal{D}_{k+1}^\pi \mid \mathcal{F}_k] \\ &\geq \max(Q_{k,\mathrm{n}}, Q_{k,\mathrm{n}} + \mathbb{E}[Q_{k+1,\mathrm{e}} \mid \mathcal{F}_k]) - \mathbb{E}[\mathcal{D}_{k+1}^\pi \mid \mathcal{F}_k] \\ &= Q_k^{\max} - \mathbb{E}[\mathcal{D}_{k+1}^\pi \mid \mathcal{F}_k] = J_k - \mathcal{D}_k^\pi. \end{aligned} \tag{37}$$

The second inequality uses both $J_{k+1} \geq 0$ and $J_{k+1} \geq Q_{k+1,\mathrm{e}}$. The final equality uses $L_k = J_k - Q_k^{\max}$ when continuing and $\mathcal{D}_k^\pi = L_k + \mathbb{E}[\mathcal{D}_{k+1}^\pi \mid \mathcal{F}_k]$. This completes the induction. □

**Effect of prediction error.** Define $\epsilon_k = \max_h |\widehat{g}_{k,h} - g_{k,h}|$. Since the costs are identical in the true and estimated values, $|\max_h \widehat{Q}_{k,h} - Q_k^{\max}| \leq \epsilon_k$. The estimated-value rule consequently satisfies

$$L_k \leq \epsilon_k \, \mathbf{1}\{a_k \neq \mathbf{1}\{Q_k^{\max} > 0\}\}. \tag{38}$$

Error contributes only when it changes the sign decision, and only along visited checkpoints. A positive numerical decision tolerance $\delta$ replaces $\epsilon_k$ in this inequality by $\epsilon_k + \delta$. Here $\epsilon_k$ includes finite-data fitting error, reference mismatch, and information lost by compressing the full prefix into seven features. The tower identity applies to the true conditional means; the separately updated fitted curves need not satisfy it exactly. The theorem compares against the three specified fixed plans and makes no assumption that the empirical regression attains the conditional means. If each purchased interval's realized cost is at most its declared interval cost, the same lower bound also holds for net value computed with realized costs.

## D External Capability Representation

Figure 9 displays the benchmark loadings and model coordinates used throughout the analysis. Table 16 identifies the corresponding external measurements and their sources.

Table 16: Sources of the 29 external measurements. Rows identify the versions or metrics used; links point to benchmark papers, dataset releases, or evaluation documentation.

| Measurement | Reference or source |
|---|---|
| AA-LCR | Artificial Analysis methodology |
| AIME 2026 | MathArena dataset |
| APEX-Agents | Vidgen et al. (2026) |
| BrowseComp | OpenAI evaluation release |
| Claw Eval (pass$^3$) | Claw-Eval release |
| CritPt | CritPt dataset |
| CyberGym | CyberGym project |
| DeepSWE v1.1 | DataCurve benchmark release |
| DeepSearchQA (F1) | Google dataset |
| FrontierScience Research (FS Research) | Wang et al. (2026) |
| FrontierScience Olympiad | Wang et al. (2026) |
| GPQA Diamond | Rein et al. (2024) |
| Humanity's Last Exam (text) | Text-only evaluation protocol |
| HMMT February 2026 | MathArena dataset |
| HMMT November 2025 | MathArena dataset |
| IFEval | Instruction-Following Evaluation |
| IMO-AnswerBench | IMO-Bench project |
| MathArena Apex 2025 | MathArena Apex release |
| MCPAtlas Public | MCP-Atlas public dataset |
| MMLU-Pro | Wang et al. (2024) |
| NL2Repo-Bench | Ding et al. (2025) |
| SciCode | Tian et al. (2024) |
| SimpleQA-Verified | Verified benchmark documentation |
| SWE-bench Multilingual | Multilingual dataset |
| SWE-bench Pro | Deng et al. (2025) |
| SWE-bench Verified | Verified dataset |
| $\tau^3$-Banking | Sierra banking evaluation |
| Terminal-Bench 2.1 | Terminal-Bench 2.1 release |
| Toolathlon (original) | Toolathlon project |

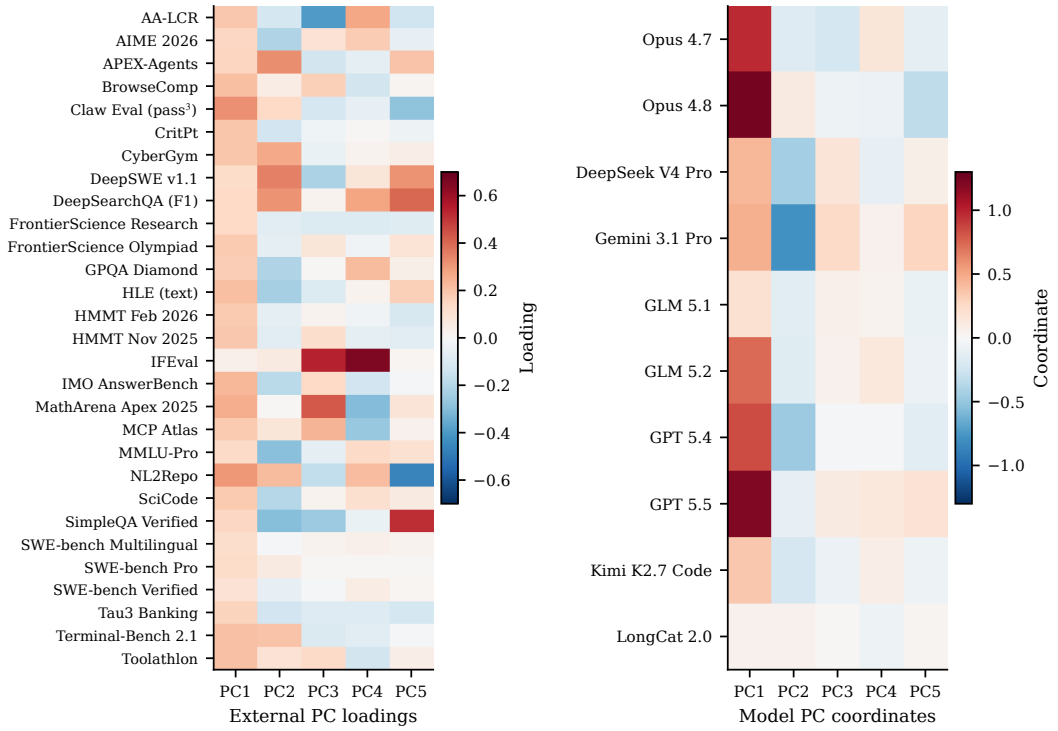



Figure 9: External representation used by the capability–time model. Left: all 29 benchmark loadings on five PCs estimated from 82 reference model entries, excluding evaluated versions and configuration aliases. Right: the fixed coordinates of all ten evaluated model versions. Loadings and coordinates have separate color scales. The loadings summarize covariance patterns among external measurements; missing-entry estimation is specified in Appendix A.

## AI USE STATEMENT

AI tools provided iterative assistance with research and mathematical development, including hypotheses and proofs; experimental implementation and data analysis; and language polishing and consistency checks. Numerical results were checked against recorded experiment outputs. The authors take responsibility for the work.