REGULARIZED AUTO-RESEARCH WITH CAUSAL HARNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Auto-research agents design, run, and learn from ML experiments on their own, and each experiment costs a training run. We find that such agents overfit to their own experimental history in two ways. On our primary task an unassisted agent locks onto its incumbent, spending half of each wave on one-parameter perturbations of it, and sessions given the same instructions end far apart. Given a correlational dashboard, it reads parameter rankings that its own co-modifications confound, and the rankings degrade as the session proceeds. We introduce CAUSALHARNESS, a harness that treats every edit as an intervention and regularizes the search with a Bayesian posterior over causal structure and mechanisms, organized in three layers: an LLM prior sets where the posterior starts, the agent's interventions update it, and the posterior returns to the agent as an information-gain ranking of candidate experiments that the agent is free to ignore. In a study in which agents are not told their condition, with multiple independent sessions of about 300 experiments per condition on up to 64 GPUs, the worst CAUSALHARNESS session ends with a lower convergence AUC than the median session of every baseline in all four settings. On the primary task every CAUSALHARNESS session converges faster than every session of a matched correlational dashboard (Hedges' $g = -2.68$, exact $p = 1/924$), reaches the target in every session, and beats the unassisted agent. At matched cadence every session also beats every run of a trust-region Bayesian optimizer, and removing any one layer significantly weakens the advantage. The gains hold on two further architectures run with a second agent LLM, and the selected configurations keep their lead under full-length pretraining.

1 INTRODUCTION

Automated research systems cover increasingly complete scientific workflows, from idea generation and experimental execution to manuscript preparation (Wang et al., 2023; Wei et al., 2025; Lu et al., 2026; Yamada et al., 2025; Schmidgall et al., 2025; Tang et al., 2025; Gottweis et al., 2026), yet end-to-end execution says little about how efficiently a system turns its own experimental outcomes into further progress. Automated ML research makes this efficiency measurable (Huang et al., 2024; Chan et al., 2025; Nathani et al., 2025; Toledo et al., 2025): model training involves interacting choices about architecture, optimization, and hyperparameters, and each experiment consumes compute and returns a measurable outcome. Karpathy's autoresearch (Karpathy, 2026) offers a compact instance, in which an LLM-driven agent reads training code, proposes changes, runs experiments, evaluates outcomes, and iterates. Under a fixed budget, what matters is how much progress the agent makes from its previous runs, and this matters more as training costs rise, because every inefficient research decision is wasted compute.

That progress depends on what the agent learns from its own experimental history, and the history is a poor training set: it is small, it is collected adaptively by the agent itself, and it is confounded because the agent changes several hyperparameters at once. An agent that learns from it without constraint overfits to it, and we observe two failure modes. **Incumbent lock-in:** on our primary task, from the fourth wave onward half of an unassisted agent's experiments change at most one parameter relative to its current best, and within the common horizon two of six sessions given the same instructions miss a target that every session with our harness reaches. **Self-confounding:** a correlational dashboard computed on the same history ranks the parameters well after the first wave,

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.

when the agent varies one parameter at a time, and progressively worse once it changes several together, consistent with the ranking absorbing the agent's own co-modifications.

Neither more feedback nor a generic regularizer fixes this. Matched to ours in form but not in causal content, a correlational dashboard has no consistent effect across four settings, and a trust-region Bayesian optimizer confined to a 13-dimensional box is not separated from the unassisted agent. Without a model of its own interventions, the agent isolates changes one parameter at a time or reads correlations its co-modifications confound. What it needs is a causal model that attributes each outcome to the parameters changed jointly and says which experiment is worth running next.

We propose CAUSALHARNESS, a middleware between model training and agent decisions in the auto-research loop. It treats every agent edit as an intervention on an unknown causal graph over hyperparameters and maintains one object, a posterior over causal structure and mechanisms, in three layers: an LLM-elicited prior fixes where the posterior starts, the agent's interventions update it, and the updated posterior returns to the agent as an information-gain ranking of candidate interventions. The harness wraps an unmodified agent and never acts on its behalf. It regularizes the search in the sense of an inductive bias: it shapes which experiments the agent considers worth running, without restricting what it may run. In all four settings the worst CAUSALHARNESS session ends with a lower convergence AUC than the median session of every baseline.

Our contributions are threefold.

- **Diagnosis.** We characterize two failure modes of auto-research that share one cause, overfitting to self-generated history: *incumbent lock-in*, where sessions settle wherever their incumbent happened to be, and *self-confounding*, where correlational summaries of the agent's own data degrade as the agent co-modifies parameters.
- **Method.** CAUSALHARNESS returns a posterior over causal structure and mechanisms, learned from the agent's own interventions, as advice the agent may ignore. An idealized analysis shows that a wrong prior inflates only the cost of *finding* the causal parent set, not the rate of optimizing within it (theorem 1).
- **Reliable gains.** Across four settings and two agent LLMs, CAUSALHARNESS makes auto-research reliable and fast. On the primary task every CAUSALHARNESS session converges faster than every session of the matched correlational dashboard (Hedges' $g{=}-2.68$, exact $p{=}1/924$) and than the unassisted agent on average ($g{=}-1.84$), and reaches the target in every session. The selected configurations keep their lead under full-length pretraining, and each single-layer ablation is significantly worse than the full system.

2 THE AUTO-RESEARCH LOOP

In the auto-research setup (Karpathy, 2026), a coding agent improves a small but real language-model pretraining script by editing `train.py`, trains each edit for a fixed wall-clock budget, and keeps or reverts it according to validation bits-per-byte. Unlike judged reports (Lu et al., 2026) or heterogeneous task suites (Huang et al., 2024; Chan et al., 2025), the setup poses the same problem with a scalar objective in every session, so differences between sessions reflect the agent's search rather than task difficulty. We run the loop on Nanochat and on re-implementations for OLMo 2 and Mamba, and extend the original single-GPU loop to parallel waves on a cluster, as recent orchestration work does (Kim & Bhardwaj, 2026), while other work optimizes the loop's own search mechanism (Qu & Lu, 2026).

At each wave the agent selects B hyperparameter configurations $\mathbf{x} \in \mathbb{R}^d$, executes them in parallel, and observes a scalar training outcome y. The d hyperparameters $\mathcal{V} = \{V_1, \ldots, V_d\}$ and the target Y (validation bits-per-byte; lower is better) are connected by an unknown directed acyclic graph (DAG) G, and the variable set may grow as the agent introduces new parameters. Experiments proceed in W waves of at most B parallel trials, and T denotes the total number of experiments. Because every experiment costs a training run, we score a session by how quickly it improves, through the area under its running-best curve,

$$\mathrm{AUC}_n \;=\; \frac{1}{n}\sum_{t=1}^{n} \min_{s \le t} y_s, \tag{1}$$

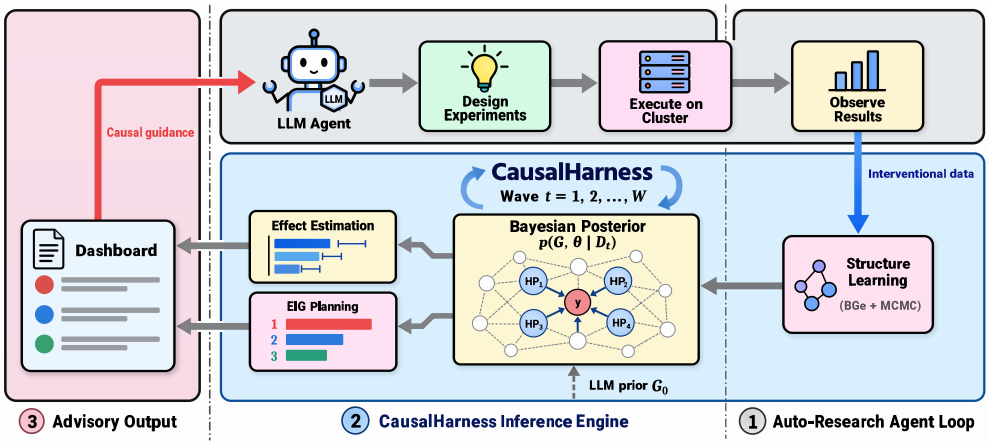

Figure 1: CAUSALHARNESS beside the auto-research loop. (1) The agent designs, runs, and observes a wave of experiments. (2) Their interventional data update one posterior $p(G, \theta \mid \mathcal{D}_t)$ over causal structure and mechanisms, starting from an LLM prior. (3) Effect estimates and a ranking of candidate interventions return to the agent as a non-binding dashboard. Whatever the agent then runs, advised or not, is the next intervention.

where t runs over unique valid evaluations (experiments that returned a finite metric) up to the horizon n. A lower value means faster convergence within the horizon. Hyperparameter importance in such spaces tends to concentrate on a few dimensions (Bergstra & Bengio, 2012; Hutter et al., 2014; Pushak & Hoos, 2022; Bischl et al., 2023). We write $k_{\text{eff}}(\epsilon)$ for the smallest number of parameters that carry a $1 - \epsilon$ share of importance. On our primary task a session-held-out random-forest fit attributes 79% of its permutation relevance to four of seventeen parameters (appendix B).

3 CAUSAL HARNESS

Every edit the agent makes in the loop of section 2 is an intervention $\mathrm{do}(V_j{=}x_j)$ on the unknown graph G. Which of the co-modified hyperparameters caused an improvement is therefore a causal question, and a marginal reading of the agent's logs cannot answer it: its policy is a common cause of the parameters it changes together, so the marginal association of one parameter with the outcome carries the effects of the others. This is the self-confounding of section 1. Adjusting for the co-modified parameters closes this back-door path, and the graph determines which parameters are parents of Y and must be adjusted for. CAUSALHARNESS therefore maintains one object, the posterior $p(G, \theta \mid \mathcal{D}_t)$ over structure G and mechanisms θ, and organizes the loop around it in three layers (fig. 1; algorithm 1 in appendix F). An LLM prior fixes where the posterior starts (section 3.1), the agent's interventions update it (section 3.2), and the updated posterior returns to the agent as a ranking of candidate interventions whose outcomes form the next update (section 3.3).

3.1 PRIOR LAYER: WHERE THE POSTERIOR STARTS

Before any experiment the LLM emits edge beliefs $p_0(e_{ij})$, that is, which hyperparameters it expects to be parents of Y, entering as a product-form prior over graphs,

$$\pi(G) \;=\; \prod_{(i,j)} p_0(e_{ij})^{\mathbf{1}[e_{ij} \in G]} \left(1 - p_0(e_{ij})\right)^{\mathbf{1}[e_{ij} \notin G]}. \tag{2}$$

This is a prior over causal structure, not over good configurations, and it is multiplied into the posterior rather than imposed, so interventional evidence overrides it. A budget of T experiments serves two tasks, finding the parameters that matter among d hyperparameters and optimizing within them, and the prior acts only on the first. With G_0^{LLM} the graph encoded by the prior and G^* the true graph, the following idealized analysis states what a wrong prior costs.

Theorem 1 (Prior Quality Tradeoff)**.** *Let $s = \mathrm{SHD}(G_0^{\text{LLM}}, G^*)$ denote the structural Hamming distance between the LLM prior and the true graph. In an idealized two-phase system satisfying*

Assumptions A1–A3 (appendices B and C) and allocating T_1 experiments to structure learning and $T_2 = T - T_1$ to optimization, the simple regret decomposes as:

$$r_T \;\le\; \underbrace{\mathcal{O}\!\left(\frac{s \log d}{T_1}\right)}_{\textit{structure learning cost}} \;+\; \underbrace{\mathcal{O}\!\left(T_2^{-\frac{2}{k_{\text{eff}}+2}}\right)}_{\textit{optimization in eff. subspace}} \;+ \mathcal{O}(\epsilon). \tag{3}$$

The optimization term scales with k_{eff} rather than d, which is what structure buys over black-box search. Prior error s enters only the structure term, which at the optimal allocation of T_1 decays faster than the optimization term (appendix C). Prior quality therefore sets the exploration cost paid before the reduced-dimensional rate is reached, not the rate itself.

3.2 Structure Layer: How Interventions Update the Posterior

Under $\mathrm{do}(V_j{=}x_j)$ the truncated factorization removes V_j's own mechanism while every other node keeps its conditional (Pearl, 2009). The BGe score (Geiger & Heckerman, 1994) implements this on the agent's data (Hauser & Bühlmann, 2012). It decomposes over nodes,

$$\log p(\mathcal{D}_t \mid G) \;=\; \sum_{v \in \mathcal{V} \cup \{Y\}} \log p(\mathbf{D}_v \mid \mathrm{pa}(v)), \tag{4}$$

and the local term of V_j uses only the rows in which the submission did not set V_j explicitly (its value was inherited from the variant file), so the agent's choice of V_j does not enter V_j's own term, while the set values are retained in the terms of V_j's children, where they act through causal paths. A correlational summary of the same rows, including the agent's own reading of its logs, models the search policy instead. DAGs are sampled by Order MCMC (Friedman & Koller, 2003) with four chains under $\hat{R}$ monitoring (Gelman & Rubin, 1992). Because early waves supply few rows relative to d, the edge marginals are shrunk toward 0.5 with weight $1 - \lambda_t$, $\lambda_t = \min(1,\, N_t/3d)$, so they cannot concentrate before the data support it. Edge marginals over the S retained samples,

$$\hat{p}(e_{ij}) = \frac{1}{S} \sum_{k=1}^{S} \mathbf{1}[e_{ij} \in G_k], \tag{5}$$

tell the agent which parameters are causes: $\hat{p}(e_{j \to Y}) \ge 0.8$ marks V_j as a confirmed parent of Y.

The effect of V_j on Y is $\partial\, \mathbb{E}[Y \mid \mathrm{do}(\mathbf{x})]/\partial x_j$, which under a given G depends only on $\mathrm{pa}(Y)$. Because G is uncertain, so is the parent set. For each plausible parent set S_m we fit a Gaussian process with an ARD-RBF kernel, $\hat{f}_m$, and marginalize over structure,

$$\hat{\tau}_j \;=\; \sum_{m:\, j \in S_m} w_m \left.\frac{\partial \hat{f}_m}{\partial x_j}\right|_{\bar{\mathbf{x}}}, \tag{6}$$

with $w_m = p(S_m \mid \mathcal{D}_t)$ and the derivative taken at the mean configuration $\bar{\mathbf{x}}$. The mixture mean is the Bayes-optimal point estimate under squared loss, and its credible interval carries both mechanism and structural uncertainty by the law of total variance (eq. (8) in appendix F.3). Multi-variable interventions are attributed per variable by counterfactual decomposition (appendix F.6), and architecture variants share strength through a hierarchical GP (appendix F.5).

3.3 Behavior Layer: How the Posterior Shapes the Next Intervention

A candidate $\mathbf{x}^*$ is valued by what its outcome would teach the posterior: about G while structure is unresolved, and about the mechanism on $\mathrm{pa}(Y)$ once it is,

$$\mathrm{EIG}(\mathbf{x}^*) = (1-\alpha)\ \mathrm{EIG}_{\mathrm{struct}}(\mathbf{x}^*) + \alpha\ \mathrm{EIG}_{\mathrm{mech}}(\mathbf{x}^*), \qquad \alpha = \bar{c}^{\,2},\ \ \bar{c} = \underset{i:\, n_i \ge 2}{\mathrm{mean}} \left|2\hat{p}(e_{i \to Y}) - 1\right|. \tag{7}$$

$\mathrm{EIG}_{\mathrm{struct}}$ is the expected entropy reduction of the DAG posterior and favors interventions that split competing parent sets. $\mathrm{EIG}_{\mathrm{mech}}$ is the GP predictive standard deviation of the standardized outcome under the MAP parent set, capped at one (appendix F.4), and favors configurations far from the observed data. The weight α is the squared mean certainty over edges tested at least twice: zero

while every tested edge is maximally uncertain and one once every edge is resolved, so the balance between discovery and optimization is read off the posterior rather than scheduled.

After each wave the agent receives edge marginals, effects with credible intervals, counterfactual decompositions, and the EIG ranking, and selects its next B experiments. The posterior update takes under 30 s of CPU per wave. The ranking is advisory. Whatever the agent runs is an intervention that updates the posterior of section 3.2, so the loop closes through the agent, not around it. Section 5.3 ablates each layer.

4 Experimental Setup

The primary task is Nanochat, language-model pretraining of a transformer from scratch with $d{=}17$ tunable hyperparameters (appendix A), a task that shows the concentrated hyperparameter importance reported for other training problems (Hutter et al., 2014; Pushak & Hoos, 2022). Each run trains for 5 minutes (10 on OLMo 2) on one NVIDIA H200 GPU from a human-tuned genesis configuration ($y_{\text{genesis}} = 0.992$). We run it at two cadences, waves of 64 and of 16 experiments, and replicate the design on OLMo 2 1B and on a Mamba-1 state-space model at 16 GPUs. The study used about 3,000 H200 GPU-hours (table 5).

Agents are not told their condition and receive the same task instructions and budgets. Conditions differ in the analysis they provide: all but the unassisted agent describe their analysis system at initialization and show a dashboard after each wave, and conditions with an LLM prior also print the prior at initialization. **VANILLA** (VAN) sees raw results only. **STATFEEDBACK** (SF) receives a correlational dashboard (sensitivity rankings, coverage gaps, interactions, and suggested experiments) matched to ours in tone, structure, and length (examples in appendix H). **CAUSALHARNESS** (CH) receives the causal dashboard of section 3. Three ablations each remove one layer: CH-NOPRIOR replaces the LLM prior with a uniform one, CH-NODAG replaces the learned structure posterior with a fixed star graph and per-parameter GPs, and CH-NOEIG drops the information-gain ranking. As a non-agent reference, a genesis-initialized TuRBO-1 batch Bayesian optimizer (Eriksson et al., 2019) runs eight times at the 16-GPU cadence over the 13 numeric hyperparameters. It cannot tune the two categorical and two integer parameters or create architecture variants (appendix G). Nanochat agents use Claude, and the OLMo 2 and Mamba agents use DeepSeek V4 (DeepSeek-AI, 2026), including prior elicitation. Each condition runs $n{=}6$ independent sessions of about 300 experiments.

The primary endpoint is convergence AUC (eq. (1)). Reliability, behavior, and transfer are reported descriptively. AUC is taken at a horizon of K unique valid evaluations, the common support of all sessions in a setting: $K{=}225$ at 64 GPUs, three quarters of the 300-experiment budget, and 254, 219, and 243 for the 16-GPU, OLMo 2, and Mamba settings. The primary AUC comparisons use one-sided full-enumeration exact permutation tests over all 924 balanced labelings, with Hedges' g as CAUSALHARNESS minus baseline (negative favors CAUSALHARNESS) and Holm–Bonferroni correction within each prespecified family (appendix D).

5 Results and Insights

5.1 Causal Harness Buys Reliability and Speed

The gain sits in the tail. On the primary task every CAUSALHARNESS session reaches a lower AUC than every STATFEEDBACK session (Hedges' $g{=}-2.68$, exact $p{=}1/924$) and CAUSALHARNESS beats the unassisted agent ($g{=}-1.84$, $p{=}3/924$), both rejecting after Holm correction and at every horizon from 72 (against VANILLA) and 86 (against STATFEEDBACK) valid evaluations onward (table 1; appendix D.1). What moves most is the bad end of the distribution. The spread across CAUSALHARNESS sessions is smaller than for either baseline in every setting, about half on Nanochat at 64 GPUs (appendix E), and in all four settings the worst CAUSALHARNESS session beats the median session of every baseline (fig. 2). A practitioner runs one session, not six, so removing the sessions that stall is worth more than moving the mean. The pattern holds at the 16-GPU cadence and on OLMo 2 and Mamba, both run with the second agent LLM, and on OLMo 2 every CAUSALHARNESS session beats every baseline session (table 2; per-session tables in appendices I and M).

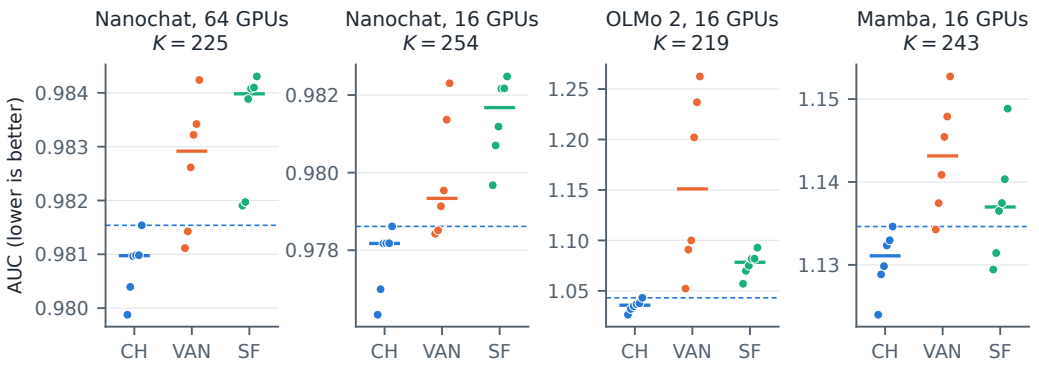

Figure 2: Per-session convergence AUC in the four settings (one dot per session, bar at the median, n=6 per condition). The dashed line marks the worst CAUSALHARNESS session, which lies below the median session of every baseline in all four settings.

Table 1: Nanochat, 64 GPUs, K=225 valid experiments, n=6 sessions per condition. AUC and final best are means ± SD over sessions. Final best is given within K and within each session's full budget (full). Worst: the highest session AUC. Reach: sessions with `val_bpb` ≤ 0.978 within K (median evaluations among them). Hedges' g is CAUSALHARNESS minus the row condition, and p is the one-sided full-enumeration exact value. **Rejects after Holm correction within its family.

Condition	AUC ↓	Worst ↓	Final (K) ↓	Final (full) ↓	Reach	g (CH−row)	exact p
CAUSALHARNESS	**0.9808 ± .0006**	**0.9815**	**0.9756 ± .0016**	**0.9752 ± .0015**	**6/6** (144)	—	—
VANILLA	0.9827 ± .0012	0.9842	0.9780 ± .0018	0.9763 ± .0022	4/6 (180)	−1.84	$3/924^{**}$
STATFEEDBACK	0.9834 ± .0011	0.9843	0.9799 ± .0023	0.9778 ± .0030	2/6 (163.5)	−2.68	$1/924^{**}$
Removing one layer							
CH-NOPRIOR	0.9826 ± .0011	0.9844	0.9780 ± .0011	0.9769 ± .0006	4/6 (165)	−1.88	$2/924^{**}$
CH-NODAG	0.9823 ± .0010	0.9835	0.9776 ± .0012	0.9762 ± .0007	3/6 (153)	−1.77	$6/924^{**}$
CH-NOEIG	0.9823 ± .0009	0.9834	0.9778 ± .0014	0.9766 ± .0017	4/6 (191)	−1.80	$2/924^{**}$

Faster to the same target, and the lead survives full-length training. Every CAUSALHARNESS session reaches the 0.978 target, essentially the level at which the median unassisted session ends its horizon (0.9778), after a median of 144 evaluations, 64% of the horizon and 20% fewer than the four unassisted sessions that reach it within the horizon (180, fig. 3b). At the end of the first wave, before the agent has seen a dashboard, CAUSALHARNESS and the unassisted agent are level (fig. 3a; appendix D.1), so the gap opens once the agent starts acting on the posterior. The lead is not an artifact of the five-minute proxy. We retrained each session's best configuration within its first 100 evaluations, a fixed early point that reflects search efficiency rather than the final budget, for 7,200 s on Nanochat and on Mamba. The CAUSALHARNESS configurations reach the lowest median full-pretraining `val_bpb` and the highest LAMBADA and HellaSwag accuracy on both templates (table 2), and proxy and full-length `val_bpb` correlate within each template (Spearman r=0.674 on Nanochat, 0.975 on Mamba).

5.2 TWO FAILURE MODES OF UNREGULARIZED SEARCH

Incumbent lock-in. On the primary task, from the fourth wave onward, 50.8% of an unassisted agent's experiments change at most one of the 17 parameters relative to its incumbent (session range 37.5–65.4%), against 24.9% for CAUSALHARNESS (1.8–46.7%; fig. 4a). One-at-a-time perturbation is the analysis an agent can still trust without a model: when only one knob moves, the outcome can be credited to it. The price is exploration. An agent that spends half of each wave this way explores little beyond its incumbent, and two of six unassisted sessions never reach the target within the horizon. CAUSALHARNESS halves the share even though its own dashboard suggests isolation tests for uncertain parameters. With a model that attributes the outcome of a joint change, the agent can afford to change several parameters at once, and its experiments move farther from the incumbent (appendix L.1).

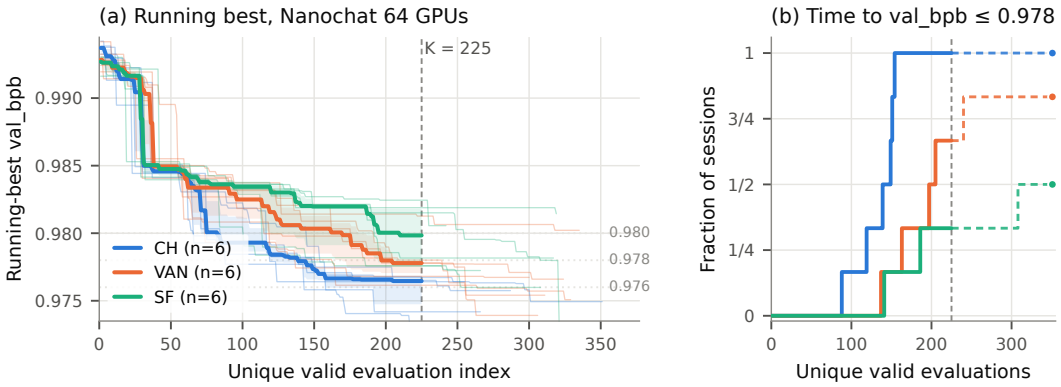

Figure 3: CAUSALHARNESS gets there first. **(a)** Running-best `val_bpb` on Nanochat at 64 GPUs: median and interquartile band over six sessions per condition up to the common support K=225 (dashed). Thin lines are individual sessions over their full budgets. **(b)** Fraction of sessions that have reached `val_bpb` ≤ 0.978, solid within K and dashed within each session's full budget.

Table 2: Replication and transfer. *Left:* convergence AUC, Hedges' g of CAUSALHARNESS minus each baseline (negative favors CAUSALHARNESS) at each setting's common support K, with the one-sided full-enumeration exact p over 924 labelings; every comparison rejects after Holm correction within its setting. *Right:* each session's best configuration within its first 100 evaluations, retrained at full length (7,200 s); `val_bpb` is the median and benchmark accuracies (%) the mean over six sessions (all benchmarks in table 24).

Setting	vs. VANILLA	vs. STATFEEDBACK
Nanochat, 64 GPUs	−1.84 (3/924)	−2.68 (1/924)
Nanochat, 16 GPUs	−1.53 (3/924)	−3.42 (1/924)
OLMo 2, 16 GPUs	−1.82 (1/924)	−3.99 (1/924)
Mamba, 16 GPUs	−2.11 (2/924)	−1.14 (24/924)

		CAUSALHARNESS	VANILLA	STATFEEDBACK
Nanochat	`val_bpb` ↓	**0.8830**	0.8975	0.9010
	LAMBADA ↑	**27.4**	26.7	25.9
	HellaSwag ↑	**32.1**	31.5	31.4
Mamba	`val_bpb` ↓	**1.1164**	1.1307	1.1243
	LAMBADA ↑	**8.8**	8.3	8.0
	HellaSwag ↑	**26.4**	26.1	26.2

Self-confounding. The correlational dashboard summarizes data the agent generated, so its ranking reflects the agent's co-modifications as well as the response. After the first wave, which mostly varies one parameter at a time from genesis, the STATFEEDBACK ranking agrees well with the response-surface reference (median NDCG@3 0.87). As the agent begins to change several parameters together, agreement falls to 0.40 by the fourth wave and does not recover (fig. 4b). An estimate that gets worse as data accumulate is what confounding predicts and noise does not, since noise would average out. The two failure modes are two halves of one trap. An agent that isolates its changes learns reliably but explores little, and an agent that changes several parameters at once explores but, read marginally, learns the wrong ranking. At the level of outcomes, STATFEEDBACK is never separated from VANILLA across the four settings (two-sided $p \geq 62/924$), with a direction that flips between tasks. More feedback does not help when the feedback is confounded.

5.3 WHAT THE GAIN REQUIRES

Where the search is pointed, not how far it may move. At the 16-GPU cadence every one of the six CAUSALHARNESS sessions reaches a lower AUC than every one of the eight TuRBO runs (g=+2.12 in the TuRBO-minus-CAUSALHARNESS direction, exact p=1/3003), with half its spread (SD 0.0009 against 0.0017). VANILLA and STATFEEDBACK are not separated from TuRBO (two-sided p=797/3003 and 1597/3003), so the agents' wider action space alone does not produce the gap (fig. 5a). A trust region regularizes by keeping each batch near the incumbent, which is what the unassisted agent already does. CAUSALHARNESS regularizes by where it points the search.

Every layer is needed, and none of them decides for the agent. Removing the prior, the structure posterior, or the information-gain ranking each makes the harness significantly worse (g=−1.88, −1.77, and −1.80 against the full system, exact p=2/924, 6/924, and 2/924, all rejecting after

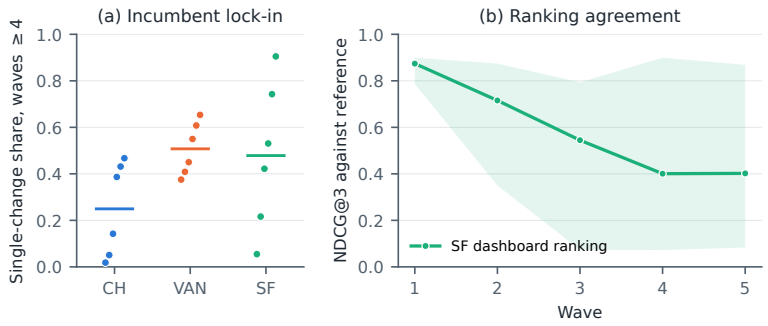

Figure 4: Two failure modes on Nanochat at 64 GPUs. **(a)** Incumbent lock-in: share of a wave's experiments that change at most one of the 17 parameters relative to the best configuration completed before the wave, averaged over waves 4 onward (one dot per session, bar at the mean; waves are batches of at least four experiments). **(b)** Self-confounding: agreement (NDCG@3) of the StatFeedback sensitivity ranking the agent saw with the session-held-out response-surface reference, per wave (ties averaged). The line is the median, the band the range over the StatFeedback sessions.

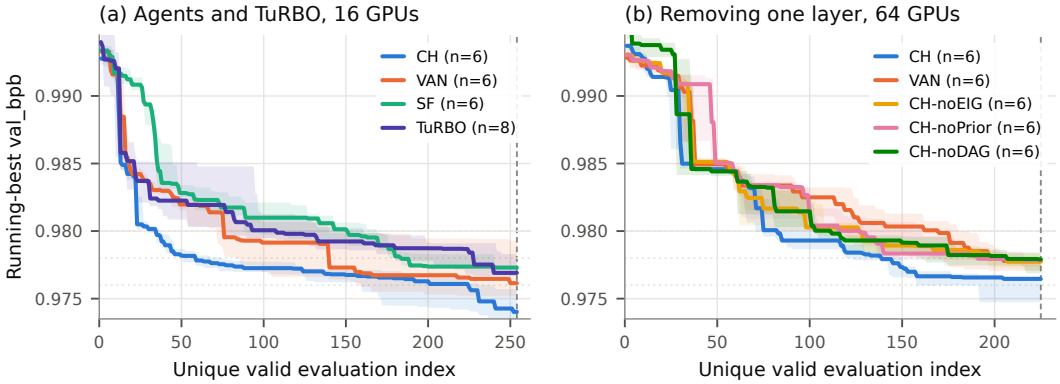

Figure 5: What the gain requires. Median running-best `val_bpb` with interquartile bands. **(a)** At the 16-GPU cadence (common support $K=254$), CausalHarness against the unassisted agent, the correlational dashboard, and eight TuRBO runs. **(b)** At 64 GPUs ($K=225$), CausalHarness against the unassisted agent and the three single-layer ablations.

Holm correction within the ablation family; table 1, fig. 5b, appendix J), and each ablation reaches the target within the horizon in three or four of six sessions against six for the full system. The layers depend on one another. The prior fixes where the posterior starts, the structure posterior decides which parameters the ranking considers, and the ranking turns the posterior into a next experiment. The structure posterior settles early: by the fourth update, three of the four parameters that carry 79% of the reference's relevance hold edge marginals above 0.9 in at least five of six sessions (appendix K). On synthetic graphs, waves to full recovery rise from 1.0 to 8.0 with the prior's structural error, consistent with theorem 1 (appendix K.1). Yet only 28% of the agent's submissions after the first dashboard follow a ranked recommendation (appendix L.2). The harness does not win by choosing experiments for the agent. It changes which experiments the agent considers worth running, which is the sense in which it regularizes the search.

6 Related Work

Agent workflows for automated scientific research. Research agents now cover the pipeline from ideation (Baek et al., 2025; Gottweis et al., 2026) to end-to-end paper production (Lu et al., 2026; Yamada et al., 2025; Schmidgall et al., 2025; Tang et al., 2025; Jansen et al., 2025), and benchmarks measure this coverage (Starace et al., 2025; Wijk et al., 2025; Kon et al., 2026; Chen et al., 2025).

None of them measures how efficiently an agent converts the experiments it has already run into its next decision, which is the question CAUSALHARNESS targets.

Automated ML research. LLMs propose hyperparameters from run histories (Zhang et al., 2023; Liu et al., 2025), search over code (Jiang et al., 2025), or improve themselves (Fang et al., 2025; Novikov et al., 2025; Zhou et al., 2026). Each learns from data its own choices produced, a setting in which selecting among noisy evaluations biases the reported optimum (Cawley & Talbot, 2010) and training on self-generated data can degrade a model (Shumailov et al., 2024). None keeps a statistical model of which parameters drive the outcome, estimated from the agent's own interventions. Correcting the history for how it was collected, as interventional scoring does, is the regularizer we propose, and CAUSALHARNESS can supply that analysis to any of them (appendix N).

Causal discovery and hyperparameter optimization. Causal Bayesian optimization (Aglietti et al., 2020; Sussex et al., 2023; Aglietti et al., 2021; 2023) and its unknown-graph variants (Branchini et al., 2023; Durand et al., 2025) choose interventions on a fixed variable set by algorithm. In CAUSALHARNESS the agent submits multi-variable batches, can introduce new parameters and architecture changes at any wave, and keeps full decision authority (appendix O).

Experimental design and LLM training. Expected information gain (Lindley, 1956) drives active causal discovery (Tong & Koller, 2001; Tigas et al., 2022; Toth et al., 2022), and Bayesian optimization (Eriksson et al., 2019; Liu et al., 2024) searches a space specified before the run. These methods choose and run the experiments themselves, whereas CAUSALHARNESS returns its analysis to an agent that decides. LLM pretraining sets model size, batch size, and learning rate before the main run, from scaling laws fitted to planned sweeps (Hoffmann et al., 2022; DeepSeek-AI, 2024) or by transfer from a smaller model (Yang et al., 2021). Auto-research learns from short proxy runs it chooses itself, and section 5.1 tests whether its choices hold at full length.

7 DISCUSSION AND LIMITATIONS

An auto-research agent learns from a history its own search has confounded, and giving it more information about that history is not the same as helping it. What helped was a causal model of the history, estimated from the agent's interventions and returned as advice it can ignore.

Why a causal model, and why as advice. An LLM agent already reads its logs, but without a model it can credit an outcome to a parameter only when that parameter changed alone, consistent with the unassisted agent isolating its changes. A summary of the same logs does not escape the problem, because in a history the agent wrote itself the marginal association of each parameter carries the agent's own choices. The harness supplies the step the agent cannot perform reliably in context, attributing each outcome jointly to the parameters that changed with it, and leaves the decision to the agent. This division of labor explains how a dashboard followed in fewer than a third of submissions can still change outcomes: it changes what the agent believes about its history, and the agent's own choices follow. Nothing in the argument depends on the search space. Whenever an agent's own choices generate the data it learns from and several knobs change per experiment, as in data-mixture or architecture search, its history is confounded by its policy, and recording which variables each experiment sets is enough to score them as interventions.

Limitations. All four settings are language-model pretraining with short proxy runs, and other research loops remain to be tested. Per-condition samples are small by design, which is why every primary comparison uses a full-enumeration exact test. The linear-Gaussian BGe score leaves nonlinear dependence to the GP layer, and a nonparametric structure score is a direct extension.

Finally, the harness helps most in the worst session, which is what a single expensive run risks. We suggest that evaluations of auto-research systems report the distribution over sessions, including the worst session and the share of sessions that reach a target, rather than the mean alone.

AI USE STATEMENT

LLM-driven agents are the object of study in this paper: the experiments were carried out by LLM agents whose configuration is described in section 4 and appendix A. In preparing the manuscript, the authors used AI tools for language editing; all technical content, analyses, and citations were written and verified by the authors.

Reproducibility statement

The experimental protocol, hyperparameter space, a summary of the agent instructions, and analysis settings are given in appendices A and F; the proof of theorem 1 is in appendix C. Anonymized code for CausalHarness, the workspace manager, the Nanochat task, and the TuRBO baseline is included in the supplementary material.

Ethics statement

This work studies hyperparameter search for language-model pretraining carried out by LLM agents. It involves no human subjects and no personal data, and the compute it consumed is reported in table 5.

References

Virginia Aglietti, Xiaoyu Lu, Andrei Paleyes, and Javier González. Causal Bayesian optimization. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3155–3164. PMLR, 2020.

Virginia Aglietti, Neil Dhir, Javier González, and Theodoros Damoulas. Dynamic causal Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10549–10560, 2021.

Virginia Aglietti, Alan Malek, Ira Ktena, and Silvia Chiappa. Constrained causal Bayesian optimization. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*, pp. 304–321. PMLR, 2023.

Raj Agrawal, Chandler Squires, Karren Yang, Karthikeyan Shanmugam, and Caroline Uhler. ABCD-Strategy: Budgeted experimental design for targeted causal structure discovery. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 89 of *Proceedings of Machine Learning Research*, pp. 3400–3409. PMLR, 2019.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, pp. 2623–2631. ACM, 2019. doi: 10.1145/3292500.3330701.

Jinheon Baek, Sujay Kumar Jauhar, Silviu Cucerzan, and Sung Ju Hwang. ResearchAgent: Iterative research idea generation over scientific literature with large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6709–6738, 2025. doi: 10.18653/v1/2025.naacl-long.342.

Maria Maddalena Barbieri and James O. Berger. Optimal predictive model selection. *Annals of Statistics*, 32(3):870–897, 2004.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

Bernd Bischl, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, Theresa Ullmann, Marc Becker, Anne-Laure Boulesteix, Difan Deng, and Marius Lindauer. Hyperparameter optimization: Foundations, algorithms, best practices, and open challenges. *WIREs Data Mining and Knowledge Discovery*, 13(2):e1484, 2023. doi: 10.1002/widm.1484.

Nicola Branchini, Virginia Aglietti, Neil Dhir, and Theodoros Damoulas. Causal entropy optimization. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 206 of *Proceedings of Machine Learning Research*, pp. 8586–8605. PMLR, 2023.

Adam D. Bull. Convergence rates of efficient global optimization algorithms. *Journal of Machine Learning Research*, 12:2879–2904, 2011.

Gavin C. Cawley and Nicola L. C. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(70):2079–2107, 2010.

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

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, Lilian Weng, and Aleksander Madry. MLE-bench: Evaluating machine learning agents on machine learning engineering. In *International Conference on Learning Representations (ICLR)*, 2025. URL `https://openreview.net/forum?id=6s5uXNWGIh`.

Hui Chen, Miao Xiong, Yujie Lu, Wei Han, Ailin Deng, Yufei He, Jiaying Wu, Yibo Li, Yue Liu, and Bryan Hooi. MLR-Bench: Evaluating AI agents on open-ended machine learning research. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 38, 2025.

Alexander I. Cowen-Rivers, Wenlong Lyu, Rasul Tutunov, Zhi Wang, Antoine Grosnit, Ryan Rhys Griffiths, Alexandre Max Maraval, Jianye Hao, Jun Wang, Jan Peters, and Haitham Bou-Ammar. HEBO: Pushing the limits of sample-efficient hyperparameter optimisation. *Journal of Artificial Intelligence Research*, 74:1269–1349, 2022. doi: 10.1613/jair.1.13643.

DeepSeek-AI. DeepSeek LLM: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*, 2024.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Jean Durand, Yashas Annadani, Stefan Bauer, and Sonali Parbhoo. Causal Bayesian optimization with unknown graphs. *arXiv preprint arXiv:2503.19554*, 2025.

David Eriksson and Martin Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 493–503. PMLR, 2021.

David Eriksson, Michael Pearce, Jacob Gardner, Ryan D. Turner, and Matthias Poloczek. Scalable global optimization via local Bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Stefan Falkner, Aaron Klein, and Frank Hutter. BOHB: Robust and efficient hyperparameter optimization at scale. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1437–1446. PMLR, 2018.

Jinyuan Fang, Yanwen Peng, Xi Zhang, Yingxu Wang, Xinhao Yi, Guibin Zhang, Yi Xu, Bin Wu, Siwei Liu, Zihao Li, Zhaochun Ren, Nikos Aletras, Xi Wang, Han Zhou, and Zaiqiao Meng. A comprehensive survey of self-evolving AI agents: A new paradigm bridging foundation models and lifelong agentic systems. *arXiv preprint arXiv:2508.07407*, 2025.

Nir Friedman and Daphne Koller. Being Bayesian about network structure: A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, 50(1-2):95–125, 2003.

Dan Geiger and David Heckerman. Learning Gaussian networks. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 235–243, 1994.

Andrew Gelman and Donald B. Rubin. Inference from iterative simulation using multiple sequences. *Statistical Science*, 7(4):457–472, 1992.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, Anil Palepu, Keran Rong, Ryutaro Tanno, Khaled Saab, Fan Zhang, Jacob Blum, Andrew Carroll, Kavita Kulkarni, Nenad Tomašev, Dina Zverinski, Ivor Rendulic, Elahe Vedadi, Florian Hasler, Luka Rimanic, Marina Boia, Ivan Budiselic, Ben Feinstein, Mathias Bellaiche, Tom Sheffer, Jan Freyberg, Jeremy Ratcliff,

Ottavia Bertolli, Katherine Chou, Avinatan Hassidim, Burak Gokturk, Amin Vahdat, Yuan Guan, Vikram Dhillon, Eeshit Dhaval Vaishnav, Byron Lee, Tiago R. D. Costa, José R. Penadés, Gary Peltz, Yossi Matias, James Manyika, Demis Hassabis, Yunhan Xu, Pushmeet Kohli, Annalisa Pawlosky, Alan Karthikesalingam, and Vivek Natarajan. Accelerating scientific discovery with Co-Scientist. *Nature*, 655(8122):487–496, 2026. doi: 10.1038/s41586-026-10644-y.

Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional Markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13:2409–2464, 2012.

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. doi: 10.52202/068431-2176.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating language agents on machine learning experimentation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20271–20309. PMLR, 2024.

Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. An efficient approach for assessing hyperparameter importance. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, pp. 754–762, 2014.

Carl Hvarfner, Danny Stoll, Artur Souza, Marius Lindauer, Frank Hutter, and Luigi Nardi. πBO: Augmenting acquisition functions with user beliefs for Bayesian optimization. In *International Conference on Learning Representations (ICLR)*, 2022. URL `https://openreview.net/forum?id=MMAeCXIa89`.

Peter Jansen, Oyvind Tafjord, Marissa Radensky, Pao Siangliulue, Tom Hope, Bhavana Dalvi Mishra, Bodhisattwa Prasad Majumder, Daniel S. Weld, and Peter Clark. CodeScientist: End-to-end semi-automated scientific discovery with code-based experimentation. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 13370–13467, 2025. doi: 10.18653/v1/2025.findings-acl.692.

Zhengyao Jiang, Dominik Schmidt, Dhruv Srikanth, Dixing Xu, Ian Kaplan, Deniss Jacenko, and Yuxiang Wu. AIDE: AI-driven exploration in the space of code. *arXiv preprint arXiv:2502.13138*, 2025.

Andrej Karpathy. AutoResearch: AI agents running research on single-GPU nanochat training automatically. `https://github.com/karpathy/autoresearch`, 2026. GitHub repository, created March 6, 2026.

Alex Kim and Romil Bhardwaj. Scaling Karpathy's autoresearch: What happens when the agent gets a GPU cluster. `https://blog.skypilot.co/scaling-autoresearch/`, 2026. Technical blog post, March 18, 2026.

Patrick Tser Jern Kon, Qiuyi Ding, Jiachen Liu, Xinyi Zhu, Jingjia Peng, Jiarong Xing, Yibo Huang, Yiming Qiu, Jayanth Srinivasa, Myungjin Lee, Mosharaf Chowdhury, Matei Zaharia, and Ang Chen. EXP-Bench: Can AI conduct AI research experiments? In *International Conference on Learning Representations (ICLR)*, 2026.

Finnian Lattimore, Tor Lattimore, and Mark D. Reid. Causal bandits: Learning good interventions via causal inference. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. SMAC3: A versatile Bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23(54):1–9, 2022.

D. V. Lindley. On a measure of the information provided by an experiment. *Annals of Mathematical Statistics*, 27(4):986–1005, 1956.

Siyi Liu, Chen Gao, and Yong Li. AgentHPO: Large language model agent for hyper-parameter optimization. In Beidi Chen, Shijia Liu, Mert Pilanci, Weijie Su, Jeremias Sulam, Yuxiang Wang, and Zhihui Zhu (eds.), *Proceedings of the Second Conference on Parsimony and Learning (CPAL)*, volume 280 of *Proceedings of Machine Learning Research*, pp. 1146–1169. PMLR, 2025.

Tennison Liu, Nicolás Astorga, Nabeel Seedat, and Mihaela van der Schaar. Large language models to enhance Bayesian optimization. In *International Conference on Learning Representations (ICLR)*, 2024.

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(8107):914–919, 2026. doi: 10.1038/s41586-026-10265-5.

Deepak Nathani, Lovish Madaan, Nicholas Roberts, Nikolay Bashlykov, Ajay Menon, Vincent Moens, Mikhail Plekhanov, Amar Budhiraja, Despoina Magka, Vladislav Vorotilov, Gaurav Chaurasia, Dieuwke Hupkes, Ricardo Silveira Cabral, Tatiana Shavrina, Jakob Nicolaus Foerster, Yoram Bachrach, William Yang Wang, and Roberta Raileanu. MLGym: A new framework and benchmark for advancing AI research agents. In *Second Conference on Language Modeling (COLM)*, 2025. URL https://openreview.net/forum?id=ryTr83DxRq.

Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. AlphaEvolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition, 2009.

Yasha Pushak and Holger H. Hoos. AutoML loss landscapes. *ACM Transactions on Evolutionary Learning and Optimization*, 2(3):1–30, 2022. doi: 10.1145/3558774.

Yaonan Qu and Meng Lu. Bilevel autoresearch: Meta-autoresearching itself. *arXiv preprint arXiv:2603.23420*, 2026.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. Agent Laboratory: Using LLM agents as research assistants. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 5977–6043, 2025. doi: 10.18653/v1/2025.findings-emnlp.320.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. AI models collapse when trained on recursively generated data. *Nature*, 631(8022):755–759, 2024. doi: 10.1038/s41586-024-07566-y.

Artur Souza, Luigi Nardi, Leonardo B. Oliveira, Kunle Olukotun, Marius Lindauer, and Frank Hutter. Bayesian optimization with a prior for the optimum. In Nuria Oliver, Fernando Pérez-Cruz, Stefan Kramer, Jesse Read, and Jose A. Lozano (eds.), *Machine Learning and Knowledge Discovery in Databases. Research Track (ECML PKDD 2021), Part III*, volume 12977 of *Lecture Notes in Computer Science*, pp. 265–296. Springer, 2021. doi: 10.1007/978-3-030-86523-8_17.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pp. 1015–1022, 2010.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. PaperBench: Evaluating AI's ability to replicate AI research. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 56843–56873. PMLR, 2025.

Scott Sussex, Anastasiia Makarova, and Andreas Krause. Model-based causal Bayesian optimization. In *International Conference on Learning Representations (ICLR)*, 2023.

Jiabin Tang, Lianghao Xia, Zhonghang Li, and Chao Huang. AI-Researcher: Autonomous scientific innovation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0320.

Panagiotis Tigas, Yashas Annadani, Andrew Jesson, Bernhard Schölkopf, Yarin Gal, and Stefan Bauer. Interventions, where and how? Experimental design for causal models at scale. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Edan Toledo, Karen Hambardzumyan, Martin Josifoski, Rishi Hazra, Nicolas Baldwin, Alexis Audran-Reiss, Michael Kuchnik, Despoina Magka, Minqi Jiang, Alisia Maria Lupidi, Andrei Lupu, Roberta Raileanu, Kelvin Niu, Tatiana Shavrina, Jean-Christophe Gagnon-Audet, Michael Shvartsman, Shagun Sodhani, Alexander H. Miller, Abhishek Charnalia, Derek Dunfield, Carole-Jean Wu, Pontus Stenetorp, Nicola Cancedda, Jakob Nicolaus Foerster, and Yoram Bachrach. AI research agents for machine learning: Search, exploration, and generalization in MLE-bench. In *Advances in Neural Information Processing Systems*, volume 38, pp. 35309–35348, 2025.

Simon Tong and Daphne Koller. Active learning for structure in Bayesian networks. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 863–869, 2001.

Christian Toth, Lars Lorch, Christian Knoll, Andreas Krause, Franz Pernkopf, Robert Peharz, and Julius von Kügelgen. Active Bayesian causal inference. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 16261–16275. Curran Associates, Inc., 2022. doi: 10.52202/068431-1183.

Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, Anima Anandkumar, Karianne Bergen, Carla P. Gomes, Shirley Ho, Pushmeet Kohli, Joan Lasenby, Jure Leskovec, Tie-Yan Liu, Arjun Manrai, Debora Marks, Bharath Ramsundar, Le Song, Jimeng Sun, Jian Tang, Petar Veličković, Max Welling, Linfeng Zhang, Connor W. Coley, Yoshua Bengio, and Marinka Zitnik. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023. doi: 10.1038/s41586-023-06221-2.

Jiaqi Wei, Yuejin Yang, Xiang Zhang, Yuhan Chen, Xiang Zhuang, Zhangyang Gao, Dongzhan Zhou, Guangshuai Wang, Zhiqiang Gao, Juntai Cao, Zijie Qiu, Ming Hu, Chenglong Ma, Shixiang Tang, Junjun He, Chunfeng Song, Xuming He, Qiang Zhang, Chenyu You, Shuangjia Zheng, Ning Ding, Wanli Ouyang, Nanqing Dong, Yu Cheng, Siqi Sun, Lei Bai, and Bowen Zhou. From AI for science to agentic science: A survey on autonomous scientific discovery. *arXiv preprint arXiv:2508.14111*, 2025.

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 (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 66772–66832. PMLR, 2025.

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The AI Scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025.

Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V. Le, Denny Zhou, and Xinyun Chen. Large language models as optimizers. In *International Conference on Learning Representations (ICLR)*, 2024.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. In *Advances in Neural Information Processing Systems*, volume 34, pp. 17084–17097, 2021.

Michael R. Zhang, Nishkrit Desai, Juhan Bae, Jonathan Lorraine, and Jimmy Ba. Using large language models for hyperparameter optimization. *arXiv preprint arXiv:2312.04528*, 2023.

Huichi Zhou, Siyuan Guo, Anjie Liu, Zhongwei Yu, Ziqin Gong, Bowen Zhao, Zhixun Chen, Menglong Zhang, Yihang Chen, Jinsong Li, Runyu Yang, Qiangbin Liu, Xinlei Yu, Jianmin Zhou, Na Wang, Chunyang Sun, and Jun Wang. Memento-Skills: Let agents design agents. *arXiv preprint arXiv:2603.18743*, 2026.

A Experimental Details

A.1 Task and Hyperparameter Space

The Nanochat task trains a small GPT-style transformer from scratch on a fixed subset of a web pretraining corpus, with an 8,192-token BPE vocabulary, rotary embeddings, sliding-window attention, value embeddings, a ReLU^2 MLP, and the Muon and AdamW optimizers. Each experiment trains for a fixed 5-minute budget on a single NVIDIA H200 GPU and reports validation bits-per-byte (`val_bpb`; lower is better). The genesis (default) configuration achieves $y_{\text{genesis}} = 0.992$.

The hyperparameter space comprises $d = 17$ tunable parameters spanning architecture, optimization, and scheduling decisions. Table 3 lists all parameters with their genesis values, types, and the bounds of the TuRBO search box. Agent conditions may move beyond these bounds.

Table 3: Complete hyperparameter space ($d = 17$). Genesis values are from the human-tuned starting configuration. Type: C = continuous, I = integer, Cat = categorical. Range shows the bounds used by TuRBO where applicable. Agent conditions can explore beyond these bounds. The four parameters marked with † are outside the TuRBO search space (see appendix G).

Parameter	Type	Genesis	TuRBO Range	Description
`DEPTH`	I	8	$[4, 12]$	Number of transformer layers
`ASPECT_RATIO`	I	64	$[32, 128]$	Model dim = `DEPTH` × `AR`, rounded to `HEAD_DIM`
`MLP_RATIO`	I	4	$[2, 8]$	MLP hidden dim multiplier
`SOFTCAP`	I	15	$[0, 50]$	Logit soft-capping magnitude (0 = disabled)
`TOTAL_BATCH_SIZE`	I	2^{19}	$[2^{18}, 2^{21}]$	Tokens per optimizer step (a multiple of the per-step micro-batch)
`MATRIX_LR`	C	0.04	$[0.005, 0.2]$	Learning rate for matrix parameters (Muon)
`EMBEDDING_LR`	C	0.6	$[0.05, 3.0]$	Learning rate for token embeddings (Adam)
`UNEMBEDDING_LR`	C	0.004	$[5 \times 10^{-4}, 0.05]$	Learning rate for `lm_head` (Adam)
`SCALAR_LR`	C	0.5	$[0.05, 3.0]$	Learning rate for per-layer scalars (Adam)
`WEIGHT_DECAY`	C	0.2	$[0.0, 0.5]$	Cautious weight decay (Muon)
`WARMUP_RATIO`	C	0.0	$[0.0, 0.3]$	Fraction of budget for LR warmup
`WARMDOWN_RATIO`	C	0.5	$[0.0, 1.0]$	Fraction of budget for LR warmdown
`FINAL_LR_FRAC`	C	0.0	$[0.0, 0.5]$	Final LR as fraction of initial
`DEVICE_BATCH_SIZE`†	I	128	—	Per-device batch size (memory management)
`HEAD_DIM`†	I	128	—	Attention head dimension
`WINDOW_PATTERN`†	Cat	`SSSL`	—	Sliding window pattern (L=full, S=half)
`ADAM_BETAS`†	Cat	$(0.8, 0.95)$	—	Adam β_1, β_2

A.2 Agent Configuration and Wave Protocol

For the Nanochat experiments all agent conditions use the same LLM, Claude Sonnet 4.6 (Anthropic) at medium reasoning effort with default sampling, within the Claude Code framework. The OLMo 2 and Mamba replications use DeepSeek V4 Pro (DeepSeek-AI, 2026) as the agent LLM, including prior elicitation, under the same instructions and wave protocol. Each agent session receives the same experiment rules (`program.md`) and task instructions (`instructions.md`). For StatFeedback and the causal conditions the instructions also describe the dashboard, and initialization prints a description of the analysis system. Conditions with a prior (CausalHarness, CH-noEIG, CH-noDAG) also print the elicited prior, the parameters the LLM expects to matter with its confidence, which a separate one-shot session of the same LLM produced from the training code (appendix A.3). CH-noPrior receives the description but no prior, and Vanilla receives no description. The instructions describe the task, available commands, and the synchronous wave protocol. They do not mention any experimental condition or the existence of other sessions.

The wave protocol enforces a strict SUBMIT→WAIT→ANALYZE loop. In the 64-GPU setting, the agent submits up to 64 experiments per wave, waits for 100% completion, then reviews all results before designing the next wave. In the 16-GPU setting, waves contain up to 16 experiments. Each session is budgeted at about 300 experiments, which the agent is not told. The budget is checked between waves, so a final in-flight wave can run past it. The agent manages architecture variants through a version-controlled system: the frozen genesis configuration (`train_genesis.py`) serves as the immutable reference, and the agent creates named variants with explicit lineage tracking.

A.3 BLINDING PROTOCOL

Blinding is enforced at the infrastructure level:

- **Condition assignment.** The experimental condition (including the ablations) is encoded as bit flags in a zlib-compressed binary blob (`.expctl_calibration.dat`) in the workspace root. No text-readable condition labels exist in the workspace. The agent's instructions never mention this file.
- **Dashboard matching.** The STATFEEDBACK and CAUSALHARNESS dashboards are matched in tone, structure, and approximate length. Both are appended to the standard experiment results output. VANILLA agents receive only raw results.
- **Workspace isolation.** Each agent session runs in a separate workspace with a random 6-character hex identifier. Each agent is launched inside its own workspace, which contains no reference to other workspaces or to the source repository.
- **Prior elicitation (conditions with a prior).** An independent one-shot LLM session generates the structural prior, writes it to the binary blob, and all session artifacts are removed. The experiment agent never observes the prior generation process.

Table 4 summarizes the information available to each condition.

Table 4: Condition feature matrix. Each cell indicates whether the agent receives that information channel.

Information Channel	VANILLA	STATFEEDBACK	CAUSALHARNESS
Raw experiment results (metric values, logs)	✓	✓	✓
Sensitivity rankings (Spearman correlations)		✓	
Coverage gap analysis		✓	
Pairwise interaction effects		✓	✓
Suggested next experiments		✓	✓
Edge posterior marginals			✓
Causal effect estimates + 95% credible intervals			✓
EIG-ranked candidate experiments			✓
Counterfactual decompositions			✓
LLM structural prior (shown at start, enters posterior)			✓

A.4 HARDWARE AND COMPUTE

All experiments run on NVIDIA H200 GPUs on a Slurm-managed cluster. Flash Attention 3 is required for the Nanochat task (Hopper architecture). Each experiment trains on one GPU for a fixed budget of 5 minutes (Nanochat, Mamba) or 10 minutes (OLMo 2).

The CAUSALHARNESS posterior update (Order MCMC + GP mixture + EIG computation) executes in under 30 seconds per wave on CPU, small against a wave of five-minute training runs and their queueing time.

Table 5: Compute summary by study. Each agent condition has six independent sessions, TURBO has eight independent runs, and full pretraining retrains one genesis plus six configurations per agent condition. Each proxy experiment runs on one GPU with a training-time budget of 5 min (Nanochat, Mamba) or 10 min (OLMo). GPU-hours are experiments × budget and exclude environment setup, compilation, and evaluation overhead. Full-pretraining hours use the configured 7,200 s training budget per run.

Study	Conditions	GPUs	Sessions/runs	Experiments	GPU-h
Nanochat 64-GPU (main + ablations)	6	64	36	11,518	959.8
Nanochat 16-GPU	3	16	18	5,479	456.6
OLMo 2 1B 16-GPU	3	16	18	5,328	888.0
Mamba-1 16-GPU	3	16	18	5,471	455.9
TURBO batch BO, 16-GPU cadence	1	16	8	2,400	200.0
Nanochat full pretraining	3	1	19	19	38.0
Mamba full pretraining	3	1	19	19	38.0
Total				**30,234**	**3,036.3**

B IMPORTANCE CONCENTRATION

We quantify importance concentration by the effective dimensionality $k_{\text{eff}}(\epsilon) = \min\{k : \sum_{j=1}^{k} I_{(j)} \geq (1-\epsilon)\sum_{j=1}^{d} I_j\}$, where $I_{(j)}$ are importance values sorted in decreasing order. On the primary task the session-held-out random-forest reference of appendix K attributes 79% of its permutation relevance to four of the seventeen parameters (`TOTAL_BATCH_SIZE`, `DEPTH`, `ASPECT_RATIO`, `DEVICE_BATCH_SIZE`), in line with the concentration reported for other training problems (Bergstra & Bengio, 2012; Hutter et al., 2014; Pushak & Hoos, 2022; Bischl et al., 2023). The statement below makes the payoff of optimizing in a low-dimensional subset explicit once that subset is given, under two assumptions that are also used in theorem 1. It is background for the theorem, which adds the cost of finding the subset.

Assumption A2 (dimension-dependent optimization rate). On an m-dimensional search domain the optimizer attains simple regret at most $C_2\, T^{-2/(m+2)}$ after T evaluations. The exponent is illustrative. What the argument uses is that it degrades with the dimension of the searched domain, as the exponent ν/m does in the GP-optimization rates of Bull (2011).

Assumption A3 (additive importance). Fixing the $d - k_{\text{eff}}$ tail parameters at any in-range values changes the attainable optimum by at most $C_3\,\epsilon$, where ϵ is the tail's share of importance.

Proposition 1 (Importance Concentration)**.** *Under A2 and A3, if the top-k_{eff} parameters are known and optimization is restricted to them (with $k_{\text{eff}}(\epsilon) \ll d$), the simple regret satisfies $r_T \leq \mathcal{O}(T^{-2/(k_{\text{eff}}+2)}) + \mathcal{O}(\epsilon)$, compared to $\mathcal{O}(T^{-2/(d+2)})$ for the full d-dimensional space.*

Both terms follow directly: A2 with $m = k_{\text{eff}}$ bounds the regret within the subset, and A3 bounds the loss from leaving the tail unoptimized.

C PROOFS

C.1 PROOF OF THEOREM 1 (PRIOR QUALITY TRADEOFF)

We restate the theorem and make the optimal allocation explicit.

Theorem (Restated)**.** *Let $s = \text{SHD}(G_0^{\text{LLM}}, G^*)$ denote the structural Hamming distance between the LLM prior and the true graph, d the number of hyperparameters, $k_{\text{eff}}(\epsilon)$ the effective dimensionality (section 2), and T the total experiment budget. In an idealized two-phase system satisfying Assumptions A1–A3 that allocates T_1 experiments to structure learning and $T_2 = T - T_1$ to optimization in the identified effective subspace, the simple regret decomposes as:*

$$r_T \leq \underbrace{\mathcal{O}\left(\frac{s \log d}{T_1}\right)}_{\textit{structure learning cost}} + \underbrace{\mathcal{O}\left(T_2^{-\frac{2}{k_{\text{eff}}+2}}\right)}_{\textit{optimization in eff. subspace}} + \mathcal{O}(\epsilon).$$

The optimal allocation $T_1^* = \Theta\big(\sqrt{s \log d} \cdot T^{(k_{\text{eff}}+4)/(2(k_{\text{eff}}+2))}\big)$ *yields:*

$$r_T^* = \mathcal{O}\left(\sqrt{s \log d} \cdot T^{-\frac{k_{\text{eff}}+4}{2(k_{\text{eff}}+2)}} + T^{-\frac{2}{k_{\text{eff}}+2}} + \epsilon \right).$$

Proof. We proceed in four steps.

Step 1: Structure learning cost. The LLM prior graph G_0^{LLM} differs from the true graph G^* in s edges (structural Hamming distance), and each erroneous edge must be corrected by interventional evidence. We abstract the structure phase by the following assumption.

Assumption A1 (edge resolution). Starting from a prior at structural Hamming distance s from G^*, after T_1 structure-phase experiments the inferred graph satisfies $\Pr[\hat{G} \neq G^* \mid \mathcal{D}_{T_1}] \leq C_1' \, s \log d / T_1$.

A1 has the form of sparse support-recovery results, in which the number of observations needed to single out the correct parent set among d candidates grows logarithmically in d, here counted per erroneous edge. On synthetic star graphs the production structure layer shows the qualitative dependence on s that A1 encodes (appendix K.1). When the inferred graph is incorrect, optimization proceeds in a misspecified subspace, contributing $\mathcal{O}(1)$ to simple regret (bounded by the metric range). Therefore:

$$r_{\text{struct}} \leq \frac{C_1 \, s \log d}{T_1},$$

where C_1 absorbs the metric-range constant and C_1'.

Step 2: Optimization in the effective subspace. After T_1 experiments, the structure is approximately known. Optimization proceeds in the identified k_{eff}-dimensional effective subspace. By proposition 1 (Assumption A2 applied to the reduced space), the simple regret after $T_2 = T - T_1$ optimization experiments satisfies:

$$r_{\text{opt}} \leq C_2 \cdot T_2^{-\frac{2}{k_{\text{eff}}+2}},$$

where C_2 is the constant of Assumption A2.

By Assumption A3, the residual contribution from the $d - k_{\text{eff}}$ tail parameters is at most $C_3 \, \epsilon$.

Step 3: Optimal allocation. The total regret is:

$$r_T \leq \frac{C_1 \, s \log d}{T_1} + C_2 (T - T_1)^{-\frac{2}{k_{\text{eff}}+2}} + C_3 \epsilon.$$

Taking the derivative with respect to T_1 and setting it to zero:

$$\frac{\partial r_T}{\partial T_1} = -\frac{C_1 \, s \log d}{T_1^2} + \frac{2 C_2}{k_{\text{eff}} + 2} (T - T_1)^{-\frac{k_{\text{eff}}+4}{k_{\text{eff}}+2}} = 0.$$

Let $\beta = \frac{2}{k_{\text{eff}}+2}$. Balancing the two terms at $T_1 \ll T$ (so $T - T_1 \approx T$):

$$\frac{s \log d}{T_1^2} \asymp T^{-(\beta+1)},$$

which gives $T_1^2 \asymp s \log d \cdot T^{\beta+1}$ and therefore:

$$T_1^* = \Theta\left(\sqrt{s \log d} \cdot T^{\frac{\beta+1}{2}} \right) = \Theta\left(\sqrt{s \log d} \cdot T^{\frac{k_{\text{eff}}+4}{2(k_{\text{eff}}+2)}} \right),$$

where the last equality substitutes $\beta = \frac{2}{k_{\text{eff}}+2}$.

Step 4: Substitution. Plugging T_1^* into the structure learning term:

$$r_{\text{struct}} \leq \frac{C_1 \, s \log d}{T_1^*} = \mathcal{O}\left(\sqrt{s \log d} \cdot T^{-\frac{k_{\text{eff}}+4}{2(k_{\text{eff}}+2)}} \right).$$

Plugging $T_2 = T - T_1^* \approx T$ into the optimization term:

$$r_{\text{opt}} \leq C_2 T^{-\frac{2}{k_{\text{eff}}+2}}.$$

The combined optimal regret is:

$$r_T^* = \mathcal{O}\left(\sqrt{s \log d} \cdot T^{-\frac{k_{\text{eff}}+4}{2(k_{\text{eff}}+2)}} + T^{-\frac{2}{k_{\text{eff}}+2}}\right) + \mathcal{O}(\epsilon).$$

For large T, the optimization term dominates (since $\frac{k_{\text{eff}}+4}{2(k_{\text{eff}}+2)} > \frac{2}{k_{\text{eff}}+2}$ for all $k_{\text{eff}} > 0$), recovering the rate of proposition 1. When $s = 0$ (perfect prior), no structure learning is needed ($T_1^* = 0$) and the bound reduces directly to this rate. □

Discussion. The theorem analyzes an idealized two-phase schedule. CAUSALHARNESS does not switch phases. It blends the structural and mechanism terms continuously through $\alpha = \bar{c}^2$ (eq. (7)), a weight that tracks how resolved the edge marginals are, so the theorem describes the tradeoff the blend addresses rather than the deployed schedule. The theorem says that prior quality moves the exploration budget rather than the asymptotic rate. The synthetic prior-quality sweep, which injects confidently wrong structure, shows the corresponding rise in the number of waves needed for recovery (appendix K.1).

Remark (Identifiability vs. sample complexity). Hauser & Bühlmann (2012) establish a *combinatorial identifiability* result: a graphical criterion for when two DAGs are equivalent under a given family of intervention targets, which determines how far a set of interventions can resolve the interventional Markov equivalence class. Assumption A1 concerns a separate, *statistical* quantity, the number of observations the posterior needs to concentrate on the correct parent set. Identifiability guarantees that a resolving intervention set *exists*, and A1 posits how many observations resolving it takes.

D ADDITIONAL STATISTICAL RESULTS

Holm–Bonferroni correction is applied within each of the following families. At each Nanochat cadence (64 and 16 GPUs): CAUSALHARNESS−STATFEEDBACK, CAUSALHARNESS−VANILLA, and the two-sided STATFEEDBACK−VANILLA. On OLMo 2 and on Mamba separately: the two CAUSALHARNESS comparisons. The three ablations form one family, and the TuRBO comparison stands alone.

D.1 AUC CUTOFF SENSITIVITY ANALYSIS

The main text uses the endpoint $K{=}225$ unique valid experiments, three quarters of the 300-experiment session budget and the common support across all 64-GPU sessions (every session provides at least 225 unique valid evaluations). The scan below shows that the conclusions do not depend on this choice. Figure 6 and table 6 report the three comparisons at every cutoff from 50 to 225. After Holm correction within each cutoff's three-comparison family, CAUSALHARNESS vs VANILLA rejects continuously from cutoff 72 and CAUSALHARNESS vs STATFEEDBACK from cutoff 86, with no later reversal, while STATFEEDBACK vs VANILLA never rejects. The CAUSALHARNESS–STATFEEDBACK effect grows in magnitude from $g{=}-1.82$ (cutoff 100) to -2.68 (225), and CAUSALHARNESS–VANILLA stays between -1.63 and -1.84 at cutoffs of 100 and above. The scan is a sensitivity analysis, not a family of confirmatory tests.

Balance at evaluation 64. At unique valid evaluation 64, roughly the end of the first wave, the running best of CAUSALHARNESS and VANILLA is level (0.9832 each, CAUSALHARNESS−VANILLA $g{=}+0.02$, exact $p{=}475/924$). Over evaluations 65–225 alone, CAUSALHARNESS−STATFEEDBACK is $g{=}-2.44$ (exact $p{=}2/924$) and CAUSALHARNESS−VANILLA is $g{=}-1.59$ ($p{=}10/924$).

D.2 PER-SESSION RESULT TABLES

Table 7 lists the individual sessions of the 64-GPU main experiment.

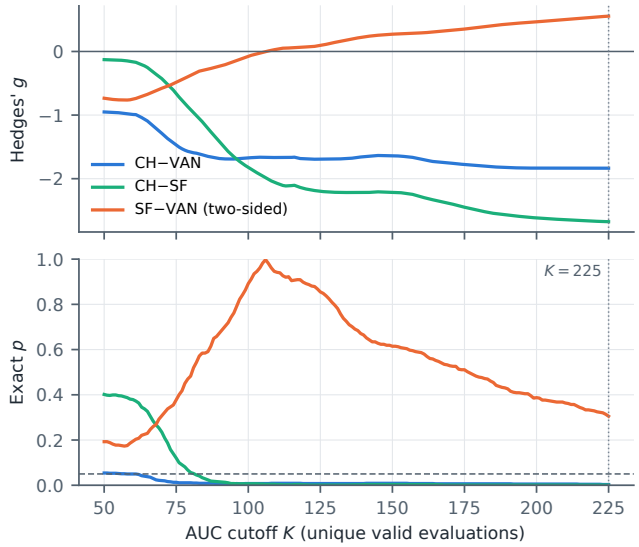

Figure 6: AUC cutoff sensitivity (64-GPU, n=6 sessions per condition). Hedges' g and full-enumeration exact p as functions of the AUC cutoff. The dashed line marks p=0.05 before adjustment. After Holm correction within each cutoff's three-comparison family, the CAUSALHARNESS advantage over STATFEEDBACK rejects from cutoff 86 onward and over VANILLA from cutoff 72 onward (table 6).

Table 6: Cutoff sensitivity, 64-GPU, n=6 sessions per condition. Hedges' g in the displayed order (negative CAUSALHARNESS−baseline favors CAUSALHARNESS; positive STATFEEDBACK−VANILLA favors VANILLA). CAUSALHARNESS comparisons use one-sided full-enumeration exact tests, STATFEEDBACK−VANILLA is two-sided, and Holm is applied within each cutoff's three-comparison family.

K	CH−SF g	p	Holm	CH−VAN g	p	Holm	SF−VAN g	p (two-sided)	Holm
50	−0.13	371/924	no	−0.95	50/924	no	−0.74	178/924	no
75	−0.66	111/924	no	−1.47	12/924	yes	−0.48	350/924	no
100	−1.82	6/924	yes	−1.67	7/924	yes	−0.07	824/924	no
125	−2.20	1/924	yes	−1.69	7/924	yes	+0.09	790/924	no
150	−2.21	3/924	yes	−1.64	8/924	yes	+0.27	568/924	no
175	−2.45	1/924	yes	−1.78	6/924	yes	+0.35	472/924	no
200	−2.62	1/924	yes	−1.83	5/924	yes	+0.47	358/924	no
225	−2.68	1/924	yes	−1.84	3/924	yes	+0.56	282/924	no

E RELIABILITY ACROSS SESSIONS

Table 8 summarizes the spread of per-session AUC in the four settings. All quantities are descriptive and computed from the per-session tables of appendices D.2, I and M.

Ranking agreement per wave. Table 9 lists the agreement (NDCG@3) of the STATFEEDBACK sensitivity ranking, replayed from the agent's logs, with the response-surface reference of appendix K at the dashboard shown after each wave.

F ALGORITHMIC DETAILS AND HYPERPARAMETERS

This section provides the full algorithmic specification required for reproducibility. All parameter values below are fixed defaults shared by every task, and none was tuned per task.

Table 7: Per-session results (64-GPU; CH = CAUSALHARNESS, VAN = VANILLA, SF = STATFEEDBACK). Unique valid experiments define the AUC index, and the AUC cutoff is $K = 225$. Session labels S1–S6 are anonymous within condition. AUC is printed to six decimals so that every exact p reproduces from this table. A full-enumeration p counts the labelings whose mean difference is at least as extreme as the observed one, ties included.

Session	Cond	AUC
S1	CH	0.980967
S2	CH	0.980983
S3	CH	0.979877
S4	CH	0.981539
S5	CH	0.980983
S6	CH	0.980394
S1	VAN	0.983421
S2	VAN	0.981115
S3	VAN	0.983220
S4	VAN	0.982615
S5	VAN	0.981425
S6	VAN	0.984240
S1	SF	0.983886
S2	SF	0.984305
S3	SF	0.981971
S4	SF	0.981905
S5	SF	0.984078
S6	SF	0.984099

Table 8: Per-session AUC spread. SD ratio and range ratio: baseline over CAUSALHARNESS. "Worst CAUSALHARNESS < median": the highest CAUSALHARNESS session AUC is below the median session AUC of the baseline.

	vs. VANILLA			vs. STATFEEDBACK		
Setting	SD ratio	Range ratio	Worst CH < median	SD ratio	Range ratio	Worst CH < median
Nanochat, 64 GPUs	2.10	1.88	yes	1.95	1.44	yes
Nanochat, 16 GPUs	1.82	1.70	yes	1.23	1.23	yes
OLMo 2, 16 GPUs	15.44	12.52	yes	2.16	2.13	yes
Mamba, 16 GPUs	1.81	1.74	yes	1.82	1.82	yes

F.1 ORDER MCMC CONFIGURATION

Table 10 lists all MCMC parameters. The implementation runs four parallel chains, each performing up to 20,000 iterations with adjacent-transposition proposals on the node ordering. Convergence is monitored via $\hat{R}$ (Gelman & Rubin, 1992) every 2,000 iterations, and chains auto-stop when $\hat{R} < 1.05$ across all monitored edge marginals. The first 20% of each chain is discarded as burn-in, and samples are thinned by keeping every 50th iteration.

F.2 BGE SCORING DETAILS

The BGe score (eq. (4)) assumes local linear-Gaussian mechanisms with a conjugate Normal-Wishart prior. Data are standardized (zero mean, unit variance) before scoring. The prior parameters are: mean precision $\alpha_\mu = 1.0$ and degrees-of-freedom offset $\alpha_W = d_{\text{local}} + 2$, where d_{local} is the number of variables in the local score computation (child + parents). These are our settings for the Normal-Wishart prior of Geiger & Heckerman (1994).

When a node v is directly intervened (i.e., the agent explicitly set its value), that observation is excluded from v's own local score computation but retained for scoring other nodes that have v as a parent. This follows the interventional scoring approach of Hauser & Bühlmann (2012) and prevents the intervened value from inflating the apparent dependence between v and its parents. Intervention targets are automatically extracted from code diffs via a section-aware parser that compares each experiment's training script to the genesis configuration.

Table 9: NDCG@3 of the StatFeedback ranking against the response-surface reference, median [range] over sessions (Nanochat, 64 GPUs).

End of wave	1	2	3	4	5
StatFeedback dashboard	0.87 [0.79, 0.90]	0.72 [0.35, 0.87]	0.55 [0.07, 0.79]	0.40 [0.07, 0.90]	0.40 [0.08, 0.87]

Algorithm 1: CausalHarness: the causally informed experiment loop

Input: agent A, HP space $\mathcal{V}$, target Y, waves W, batch size B, prior graph G_0^{LLM}

Initialize $\mathcal{D}_0 \leftarrow \emptyset$ and $p(G \mid \mathcal{D}_0)$ from the prior — eq. (2)

for $t = 1$ **to** W **do**

- Order MCMC with BGe scoring $\rightarrow$ edge marginals $\hat{p}(e_{ij})$ — eqs. (4) and (5)
- Posterior-weighted GP mixture $\rightarrow$ effects $\hat{\tau}_j$ with credible intervals — eqs. (6) and (8)
- Weight α from edge certainty $\rightarrow$ candidates ranked by $\mathrm{EIG}(\mathbf{x}^*)$ — eq. (7)
- Present the analysis to A; A selects and runs B experiments; append outcomes to $\mathcal{D}_t$

Output: the trajectory $\mathcal{D}_T$, scored by eq. (1); the best configuration is $\arg\min_{(\mathbf{x},y)\in\mathcal{D}_T} y$

Posterior tempering. With small sample sizes (fewer than $3 \times d$ observations), the BGe posterior can over-concentrate due to limited data. We apply posterior tempering by mixing MCMC edge marginals toward the uniform prior ($p = 0.5$) at a rate that decays as data accumulates:

$$\hat{p}_{\text{tempered}}(e_{ij}) = \lambda \cdot \hat{p}(e_{ij}) + (1 - \lambda) \cdot 0.5, \quad \lambda = \min\left(1, \frac{N}{3d}\right),$$

where N is the total observation count. With d=17 this shrinkage is active while $N < 51$, which at 16 GPUs covers the first three waves. At 64 GPUs the first update already has more observations.

F.3 GP Mechanism Model Configuration

Table 11 lists the GP configuration. The credible interval of each effect combines mechanism and structural uncertainty,

$$\mathrm{Var}[\hat{\tau}_j] \;=\; \underbrace{\sum_{m:\, j\in S_m} w_m\, \sigma_m^2}_{\text{mechanism}} \;+\; \underbrace{\sum_{m:\, j\in S_m} w_m\, \mu_m^2 - \hat{\tau}_j^2}_{\text{structure}}, \tag{8}$$

where (μ_m, σ_m^2) are the predictive mean and variance of $\partial \hat{f}_m / \partial x_j$ at $\bar{\mathbf{x}}$. Conditioning on the MAP parent set alone drops the structural term.

For each distinct parent set S_m in the structure posterior with weight $w_m \geq 0.05$, a separate GP is fitted using scikit-learn's `GaussianProcessRegressor`, and the surviving weights are renormalized to sum to one before entering eqs. (6) and (8). The ARD (Automatic Relevance Determination) RBF kernel assigns an independent lengthscale to each input dimension, enabling the GP to automatically down-weight irrelevant parents within each parent set. GP fits are cached by (parent set, data hash) to avoid redundant computation, and cache entries are evicted when the corresponding parent set drops below the weight threshold.

Effect estimates (eq. (6)) use half-standard-deviation perturbations ($\Delta = 0.5\sigma$) around the mean configuration, with credible intervals computed via the law of total variance (eq. (8)).

F.4 EIG Computation Details

Table 12 lists the EIG hyperparameters.

Candidate generation. Candidates are generated in three classes: (1) single-variable grid sweeps at 5 levels per continuous/integer parameter, (2) pairwise interaction candidates for parameter pairs with edge marginal ≥ 0.3, and (3) categorical level candidates (one per known level per active categorical parameter). The candidate set size varies by wave, typically 50–150 candidates.

Table 10: MCMC hyperparameters for Order MCMC structure learning.

Parameter	Value	Rationale
Chains	4	Multi-chain $\hat{R}$ diagnostic (Gelman & Rubin, 1992)
Max iterations per chain	20,000	Auto-stops via $\hat{R}$ convergence
Burn-in fraction	0.20	Standard practice (first 20% discarded)
Thinning interval	50	Reduces autocorrelation in retained samples
$\hat{R}$ threshold	1.05	Convergence criterion (Gelman & Rubin, 1992)
Convergence check interval	2,000	Balance between overhead and early stopping
Max parent set size	8	Admits secondary parameters to parent sets
Top-k parent enumeration	10	Candidate parent sets for BGe scoring
Min new observations for re-run	5	Skip MCMC if fewer than 5 new data points

Table 11: GP configuration for mechanism modeling.

Parameter	Value	Notes
Kernel	$C \cdot \text{RBF}_{\text{ARD}} + \text{White}$	Constant $\times$ ARD RBF + noise
Signal amplitude bounds	$[10^{-3}, 10^{3}]$	Optimized within bounds
RBF lengthscale bounds	$[0.1, 10.0]$	In standardized space
Noise level (initial)	0.01	Homoscedastic Gaussian
Noise bounds	$[10^{-5}, 1.0]$	Learned via MLE
Optimizer restarts	5	For marginal likelihood optimization
Min observations for fitting	5	Below this, GP is not fitted
Parent set weight threshold	0.05	Sets with $w_m < 0.05$ are not fitted

Structural EIG. Structural EIG is computed via importance reweighting on the MCMC DAG samples. For each candidate $\mathbf{x}^*$, 30 outcome values $\{y_1, \ldots, y_{30}\}$ are simulated from the GP predictive distribution under the dominant parent set. For each simulated outcome, the DAG sample weights are reweighted by the likelihood of observing y_i under each DAG's mechanism model. The structural EIG is then the difference between the current structural entropy and the expected posterior entropy after observing the candidate:

$$\text{EIG}_{\text{struct}}(\mathbf{x}^*) = \text{H}[G \mid \mathcal{D}_t] - \frac{1}{30}\sum_{i=1}^{30} \text{H}[G \mid \mathcal{D}_t \cup \{(\mathbf{x}^*, y_i)\}].$$

When the effective sample size of the reweighted distribution drops below $0.2S$ (where S is the number of retained DAG samples), the system uses the mechanism term alone. The mechanism term is the GP predictive standard deviation of the standardized outcome at the candidate under the MAP parent set, capped at one, so that it is dimensionless like the structural term.

Blending. The structure certainty $\bar{c}$ is computed as the mean $|2\hat{p}(e_{i\to Y}) - 1|$ across all parameters the agent has intervened on (set explicitly) in at least two submissions. The blending weight $\alpha = \bar{c}^2$ is quadratic to ensure that structural EIG remains dominant until the posterior is substantially converged.

F.5 Hierarchical Branch Model

When an agent creates a new architecture variant (modifying the training code structure, not just parameter values), the mechanism model decomposes as:

$$f_v(\mathbf{x}) = f_{\text{shared}}(\mathbf{x}) + \delta_v(\mathbf{x}),$$

where f_{shared} is a global GP trained on data from all variants, and δ_v is a variant-specific residual GP with an automatic shrinkage prior. The amplitude of δ_v is bounded by the global GP's fitted amplitude, causing the residual to collapse toward zero when the variant behaves identically to the baseline. A minimum of 5 observations per variant is required before fitting the residual GP. Below this threshold, the variant pools with the global model.

Categorical variables (e.g., `WINDOW_PATTERN`) are shown to the agent as per-level summaries of the target, for up to 5 distinct levels observed within a session.

Table 12: EIG computation hyperparameters.

Parameter	Value	Notes
Grid levels per parameter	5	For single-variable sweeps
Candidate range (relative to genesis)	$[0.5\times, 2.0\times]$	Expanded based on observed range
Pairwise interaction threshold	$p \geq 0.3$	Edge marginal for pairwise candidates
Outcome simulations per candidate	30	For structural EIG estimation
ESS threshold for valid reweighting	$0.2 \times S$	Below this, fall back to the mechanism term
Fallback ratio	$1/3$	Use variance proxy if $< 33\%$ valid
Max candidates in ranking	10	Output to agent dashboard
Pairwise EIG boost	$1.1\times$	Slight boost for interaction candidates

F.6 Counterfactual Decomposition

For multi-variable interventions whose outcome is more than 0.5% worse than the batch median, the system performs noise-abduction counterfactual analysis following Pearl's structural counterfactual framework (Pearl, 2009).

Given an observed outcome y_{obs} under configuration $\mathbf{x}_{\text{obs}}$:

1. **Abduction**: Infer the exogenous noise $U = y_{\text{obs}} - \mu_{\text{GP}}(\mathbf{x}_{\text{obs}})$.
2. **Action**: For each modified variable V_j, construct the counterfactual configuration $\mathbf{x}_{V_j \leftarrow v'_j}$ where V_j is reverted to its genesis value and all other variables retain their observed values.
3. **Prediction**: Compute $Y_{V_j \leftarrow v'_j} = \mu_{\text{GP}}(\mathbf{x}_{V_j \leftarrow v'_j}) + U$.

The counterfactual effect of variable V_j is $\Delta_j = Y_{V_j \leftarrow v'_j} - y_{\text{obs}}$. Positive Δ_j indicates that reverting V_j would have improved the outcome, meaning the agent's modification of V_j was detrimental. At most 5 such experiments per wave are decomposed.

G TuRBO Baseline Analysis

G.1 Configuration

The TuRBO baseline is a genesis-initialized TuRBO-1 batch Bayesian optimizer (Eriksson et al., 2019): wave 0 evaluates the genesis configuration plus 15 Latin-hypercube points, and subsequent waves draw batches of q=16 by Thompson sampling inside a trust region that expands on success and shrinks on failure, until the 300-experiment budget is reached (the last batch is trimmed to fit), for eight independent runs (table 13). It searches the same 13 numeric hyperparameters as table 3, with `TOTAL_BATCH_SIZE` quantized to multiples of 262,144. It cannot tune `WINDOW_PATTERN`, `ADAM_BETAS`, `HEAD_DIM`, or `DEVICE_BATCH_SIZE` and cannot create architecture variants, so the agent's action space is strictly wider. The baseline runs the same training and evaluation recipe as the agent conditions on the same cluster.

Table 13: TuRBO baseline configuration.

Parameter	Value
Kernel	GP with ARD Matérn(ν=2.5)
Acquisition	Thompson sampling, q=16 per wave
Trust region	TuRBO-1 (expand on success, shrink on failure)
Search dimensions	13 numeric
Initialization	genesis + 15 LHS points
Runs	8 independent
Budget	300 experiments per run
Variant creation	not supported

Table 14: TuRBO per-run AUC at the matched 16-GPU cadence (cutoff $K{=}254$). Each run has 300 valid experiments. Runs are labeled R1–R8; mean 0.9809, SD 0.0017.

	R1	R2	R3	R4	R5	R6	R7	R8
AUC	0.980680	0.982148	0.979450	0.983490	0.979905	0.979433	0.982810	0.979350

G.2 Per-Run Results and Matched-Cadence Comparison

Table 14 lists the eight runs. Against the six 16-GPU CausalHarness sessions (mean AUC 0.9777, SD 0.0009) every CausalHarness session has lower AUC than every TuRBO run: Hedges' $g = +2.12$ in the BO minus CausalHarness convention and exact one-sided $p = 1/3003$ over all $\binom{14}{6} = 3003$ allocations of the six agent observations among the fourteen. The comparison is reported only at the matched 16-GPU cadence, where both systems update after every batch of 16. Vanilla and StatFeedback are not separated from TuRBO at matched cadence (two-sided exact $p{=}797/3003$ and $1597/3003$). Like the agents, TuRBO starts from the genesis configuration, so initialization does not explain the gap.

H Dashboard Output Examples

This section shows the final dashboard of one CausalHarness session and one StatFeedback session.

H.1 CausalHarness Dashboard Example

The following is the actual CausalHarness dashboard output from 64-GPU session S1 at the final wave (255 valid observations), abridged from the `expctl results` command. The status line counts the parameters still displayed (edge marginal ≥ 0.05) and how many of them have $\hat{p} \geq 0.8$. Parameters whose marginal has fallen below 0.05 are omitted. Categorical parameters are summarized in their own block, and the UNCERTAIN block lists parameters with edge marginal below 0.8 together with categorical parameters that have untested or rarely tested levels, whatever their marginal.

```
=== Causal Analysis (255 observations, posterior updated) ===

STATUS: 8/8 parameters resolved.  Best val_bpb = 0.9729.

CONFIRMED PARENTS (p >= 0.80):
  TOTAL_BATCH_SIZE   p=1.00  tested [65536-1048576]  GP best ~1.3e+05
  WARMDOWN_RATIO     p=1.00  tested [0.2-0.9]  GP best ~0.828
  MLP_RATIO          p=1.00  tested [2-12]  GP best ~5.79
  DEPTH              p=1.00  tested [4-14]  GP best ~7.79
  MATRIX_LR          p=0.98  tested [0.01-0.12]  GP best ~0.044
  ASPECT_RATIO       p=0.91  tested [32-96]  GP best ~65.1

UNCERTAIN:
  WINDOW_PATTERN = "LLLL"  (categorical, untested level)
  ADAM_BETAS = (0.9, 0.95)  (categorical, few tests)

INTERACTIONS:
  MATRIX_LR x WARMDOWN_RATIO: strength -0.080 (strong)
  MATRIX_LR x DEPTH: strength -0.080 (strong)

ADAM_BETAS (categorical, 4 levels):
  (0.7, 0.95):  mean = 0.9783 (n=3)  best
  (0.8, 0.95):  mean = 0.9875 (n=247)
  (0.8, 0.98):  mean = 0.9801 (n=4)
  [one further level omitted]
```

The excerpt shows four of the dashboard's blocks: (i) confirmed causal parents with posterior probabilities and GP-estimated optima, (ii) uncertain parameters ranked by expected information gain, (iii) interaction effects between parameters, and (iv) categorical parameter breakdowns. The effect estimates with credible intervals, the ranked candidate experiments, the counterfactual decompositions, and the guidance section are omitted for space. The guidance prompts the agent to align defaults with GP optima, run isolation tests for uncertain parameters, and test flagged interactions.

H.2 StatFeedback Dashboard Example

The StatFeedback dashboard provides correlational analysis without causal structure. The following is an abridged excerpt of the final dashboard of one 64-GPU StatFeedback session (S1, 317 valid observations).

```
=== Statistical Analysis (317 observations) ===


STATUS: Best val_bpb = 0.9782.


SENSITIVITY (which parameters most influence val_bpb):
  1.  FINAL_LR_FRAC        |rho| = 0.69  (strong, higher is better)
  2.  SCALAR_LR            |rho| = 0.64  (strong, lower is better)
  3.  WARMDOWN_RATIO       |rho| = 0.63  (strong, higher is better)
  4.  MLP_RATIO            |rho| = 0.61  (strong, higher is better)
  5.  MATRIX_LR            |rho| = 0.55  (moderate, lower is better)
  6.  TOTAL_BATCH_SIZE     |rho| = 0.38  (moderate, lower is better)


COVERAGE (under-explored):
  WINDOW_PATTERN     never varied
  ADAM_BETAS         never varied


INTERACTIONS:
  TOTAL_BATCH_SIZE x MATRIX_LR: R^2 = 0.09 (moderate)
  TOTAL_BATCH_SIZE x DEVICE_BATCH_SIZE: R^2 = 0.07 (moderate)


SUGGESTED EXPERIMENTS:
  - Test WINDOW_PATTERN (never varied)
  - Test ADAM_BETAS (never varied)
  - TOTAL_BATCH_SIZE x MATRIX_LR factorial (4 experiments)
```

The StatFeedback dashboard ranks parameters by correlation strength but cannot distinguish causal from spurious associations, attach credible intervals to its effect estimates, or decompose multi-variable outcomes.

I 16-GPU Cadence: Full Results

I.1 Full Statistical Tables

The 16-GPU experiment uses $n = 6$ sessions per condition (CausalHarness, Vanilla, StatFeedback) with AUC cutoff $K = 254$ unique valid experiments, the largest endpoint supported by every 16-GPU session (table 16). Figures 7 and 8 show the running-best curves and threshold crossings.

Table 15: 16-GPU pairwise comparisons (AUC, cutoff $K = 254$). Full-enumeration exact permutation tests over all 924 balanced labelings. CausalHarness comparisons are one-sided, StatFeedback−Vanilla is two-sided. Holm is applied within the three-comparison family.

Comparison	Hedges' g	Exact p	Holm-adjusted	Decision
CausalHarness−Vanilla	-1.53	3/924	6/924	reject
CausalHarness−StatFeedback	-3.42	1/924	3/924	reject
StatFeedback−Vanilla (two-sided)	$+1.03$	84/924	84/924	retain

I.2 Per-Session Tables

Table 16 lists the 16-GPU sessions behind table 15.

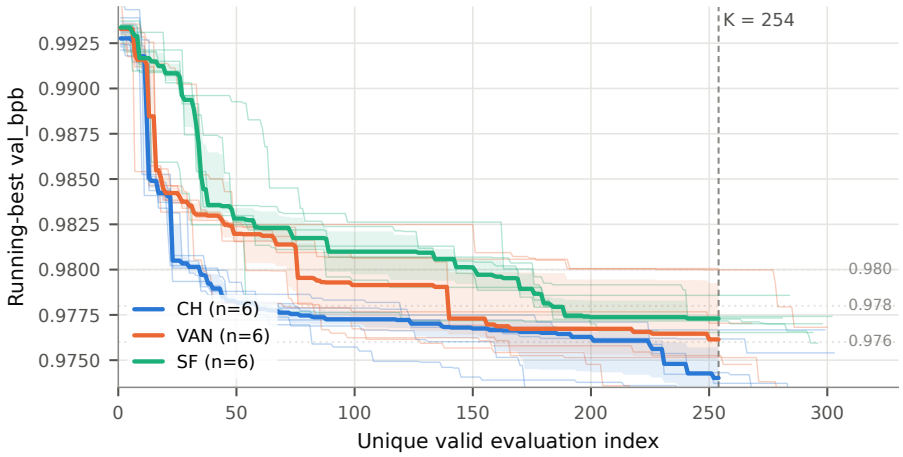

Figure 7: 16-GPU convergence curves: median running-best `val_bpb` with IQR shading (n=6 sessions per condition, K=254). Thin lines are individual sessions over their own valid length, and the median and IQR are drawn over the common support.

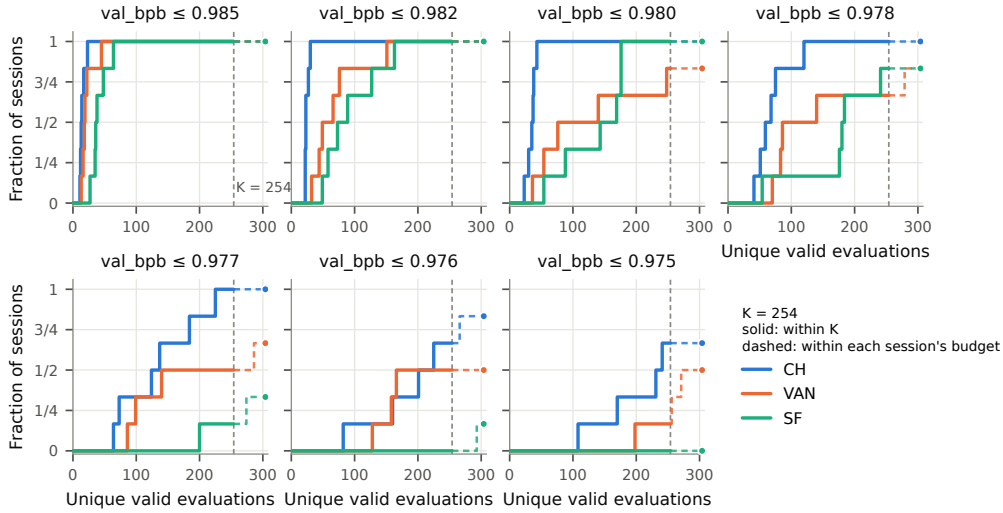

Figure 8: 16-GPU empirical CDF of experiments to reach each `val_bpb` threshold across sessions. Solid steps count first crossings within K=254 and dashed steps continue within each session's own budget. CAUSALHARNESS reaches every threshold down to 0.977 in all six sessions within K. At 0.978 within K the counts are CAUSALHARNESS 6/6, VANILLA 4/6, and STATFEEDBACK 5/6.

Table 16: Per-session AUC at K=254 (16-GPU, n=6 per condition). Labels S1–S6 are anonymous within condition. The matched-cadence TURBO comparison against the CAUSALHARNESS sessions is in appendix G.

Condition	S1	S2	S3	S4	S5	S6
CAUSALHARNESS	0.976337	0.978178	0.978172	0.978181	0.976995	0.978615
VANILLA	0.979134	0.978426	0.978515	0.981366	0.982304	0.979540
STATFEEDBACK	0.982480	0.982166	0.980700	0.979676	0.982171	0.981187

J ABLATION: FULL RESULTS

Each ablation removes one layer and keeps the rest of the dashboard. CH-NOPRIOR starts the structure posterior from uniform edge beliefs (p_0=0.5) and prints no prior. CH-NOEIG keeps the structure and mechanism posteriors and the effect estimates but ranks no candidates, suggesting single-parameter isolation tests for unresolved parameters instead. CH-NODAG runs no structure inference. Each parameter's effect, with credible intervals and the GP optimum, comes from its own one-dimensional GP. The LLM prior seeds the per-parameter importance, and candidates are ranked by GP predictive uncertainty alone.

J.1 PAIRWISE COMPARISONS

Table 17: Ablation comparisons (64-GPU, AUC cutoff $K = 225$). Hedges' g is the first condition minus the second, and p is the full-enumeration exact value. The three CAUSALHARNESS comparisons form one Holm family.

Comparison	Hedges' g	Exact p	Holm-adjusted	Decision
CAUSALHARNESS−CH-NOEIG	−1.80	2/924	6/924	reject
CAUSALHARNESS−CH-NOPRIOR	−1.88	2/924	6/924	reject
CAUSALHARNESS−CH-NODAG	−1.77	6/924	6/924	reject

All three ablations are significantly worse than CAUSALHARNESS after Holm correction (table 17).

J.2 PER-SESSION RESULTS

Table 18 lists the ablation sessions behind table 17. Figures 9 and 10 show their running-best curves and threshold crossings.

Table 18: Ablation per-session results (64-GPU). AUC at $K = 225$. Labels S1–S6 are anonymous within condition.

Condition	S1	S2	S3	S4	S5	S6
CH-NOEIG	0.983132	0.982285	0.982059	0.983443	0.981720	0.980994
CH-NOPRIOR	0.981863	0.983050	0.982599	0.981132	0.982469	0.984360
CH-NODAG	0.981198	0.983451	0.982456	0.982515	0.981130	0.983172

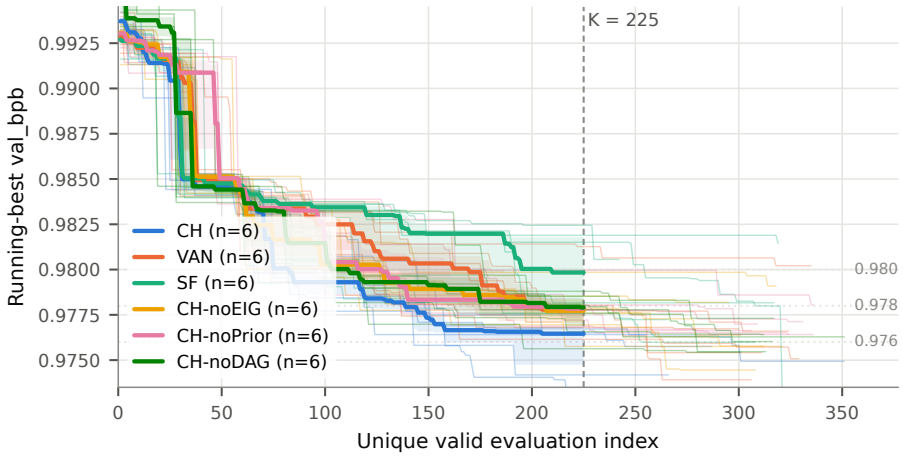

Figure 9: Ablation convergence curves (64-GPU). Six conditions are shown, n=6 sessions each, K=225: median running-best with IQR shading over the common support and thin per-session lines. The curves separate after the first wave, and at K the CAUSALHARNESS median is the lowest of the six conditions.

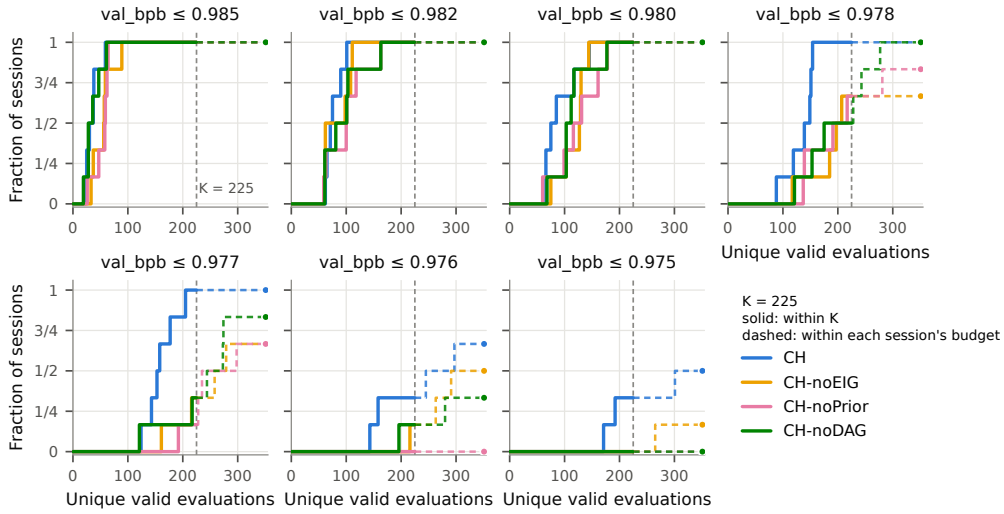

Figure 10: Ablation empirical CDF of experiments to reach each `val_bpb` threshold (CAUSALHARNESS, CH-NOEIG, CH-NOPRIOR, CH-NODAG). Solid steps count first crossings within K=225 and dashed steps continue within each session's own budget. Within K, all three ablations lag CAUSALHARNESS at every threshold $\leq$ 0.978. At 0.978 within K the counts are CAUSALHARNESS 6/6, CH-NOEIG 4/6, CH-NOPRIOR 4/6, and CH-NODAG 3/6.

K CAUSAL STRUCTURE VALIDATION

Figure 11 shows the learned structure aggregated across the six CAUSALHARNESS sessions. Directed edges are included via the median probability model ($\hat{p} \geq 0.5$; Barbieri & Berger, 2004). The posterior places hyperparameter$\rightarrow y$ marginals of 1.00, 1.00, and 0.95 on `TOTAL_BATCH_SIZE`, `DEPTH`, and `ASPECT_RATIO`, the three parameters the reference below ranks highest. As an empirical reference we fit a session-weighted, session-held-out random forest to the 4,730 64-GPU observations pooled across all three conditions (4,726 in the operational viable region, weighted $R^2 = 0.77$). Its permutation relevance ranks `TOTAL_BATCH_SIZE`, `DEPTH`, `ASPECT_RATIO`, `DEVICE_BATCH_SIZE`, and `WARMDOWN_RATIO` highest, and the top four carry 79% of relevance (d=17). The reference is a response surface under the observed intervention policies, not causal ground truth.

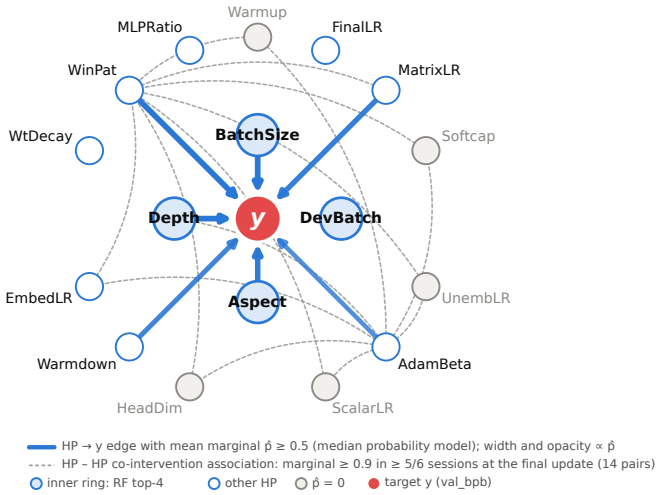

Figure 11: Learned causal structure (mean edge posteriors across the six CAUSALHARNESS sessions). Directed edges: $\hat{p}(\text{HP}\to y \mid \mathcal{D})\geq 0.5$ (median probability model; Barbieri & Berger, 2004); width and opacity proportional to posterior. Inner-ring nodes are the top-4 hyperparameters by response-surface relevance (79% of relevance share). Dashed arcs: the 14 inter-hyperparameter pairs whose undirected marginal is ≥ 0.9 in at least five of the six sessions at the final update; they are not interpreted as mechanisms.

Posterior convergence. By the fourth posterior update, `DEPTH` and `TOTAL_BATCH_SIZE` carry edge marginals above 0.9 in every session and `ASPECT_RATIO` in five of six, and all three end above the confirmed-parent threshold ($\hat{p} \geq 0.8$) in every session. The structure the agent acts on is therefore in place early.

K.1 STRUCTURE RECOVERY ON SYNTHETIC GRAPHS

We run the production structure layer (BGe scoring and Order MCMC with the defaults of appendix F, no per-run tuning) on synthetic star graphs with d hyperparameters, of which four are true parents of the target (two strong, two moderate effects) and $d-4$ are nulls. Noise is set so that single-variable sweeps of a strong parent reach $R^2 \approx 0.85$. The intervention sequence imitates an agent: early waves are single-variable sweeps and later waves are multi-variable perturbations around the incumbent. It is scripted and identical across prior conditions, so the study measures recovery and cost, not optimization. Posterior updates run at each wave on the cumulative data, on a two-core CPU.

Table 19: Synthetic structure recovery. Left: scaling with d (uniform prior, three replications per d), with AUROC of the edge marginals against the true graph and wall-clock per posterior update at the last wave. Right: prior-quality sweep at d=20 (five replications per s), where the prior confidently omits s of the four true parents.

d	Observations	AUROC	Time per update
20	149	0.98	4.0 s
50	331	0.995	10.3 s
100	613	0.96	23.1 s

Prior error s	Waves to full recovery	Fully recovered
0	1.0	5/5
1	5.4	4/5
2	6.6	4/5
3	8.0	4/5
4	8.0	4/5

The number of waves needed to recover the structure rises with the prior's structural error (table 19, right), and at every s at least four of five replications recover the full structure. In a variant with a genuine dependence between two hyperparameters, the edge is recovered with the correct orientation in 3 of 3 replications.

K.2 Response-Surface Reference Methodology

We fit a random forest (500 trees) to the 4,730 64-GPU observations pooled across the CausalHarness, Vanilla, and StatFeedback sessions (experiments that keep the genesis training code and parameter set, so architecture variants are excluded), with equal total weight per session and six-fold session-held-out validation (weighted $R^2 = 0.77$). `WINDOW_PATTERN` is encoded losslessly over its 20 observed categories and its columns are permuted jointly. The operational fit excludes the four observations with $0 <$ `SOFTCAP` < 8. This boundary is a data-derived diagnostic chosen from sparse level medians, not an identified causal threshold, and the unfiltered fit (which coincides with boundary 5, since the four excluded observations all have `SOFTCAP`$=5$) and boundaries 10 and 12 give the same top three parameters, with rank correlation at least 0.983 to the primary fit (table 20).

Table 20: Viable-region definition sensitivity for the response-surface reference. Each row refits the random forest under a different `SOFTCAP` boundary (0 always means capping disabled). Spearman ρ is the rank correlation of relevance shares with the primary fit. Sensitivity fits use 180 trees, the primary fit 500. The leading ranking is stable.

Definition	N	Weighted R^2	ρ to primary
Primary (`SOFTCAP`$=0$ or ≥ 8)	4,726	0.774	1.000
Unfiltered (all finite; equals boundary 5)	4,730	0.766	0.983
Boundary 10	4,719	0.763	0.995
Boundary 12	4,448	0.773	0.990

K.3 Comparison on the Reference's Leading Parameters

Table 21 compares the reference's five most relevant parameters. The posterior assigns mean marginals ≥ 0.95 to three of the reference's top four (`TOTAL_BATCH_SIZE`, `DEPTH`, `ASPECT_RATIO`; per-session minima 1.00, 0.997, and 0.81).

Table 21: The five most relevant parameters of the pooled 64-GPU response surface (appendix K.2), which together carry 84% of its relevance. RF rank and RF share = random-forest permutation relevance of the primary fit (share of positive ΔMSE). CH $\bar{p}$ = mean final CausalHarness edge marginal $p(\text{HP} \to y \mid \mathcal{D})$ across the six 64-GPU sessions. Parameters sorted by RF rank.

Parameter	RF rank	RF share	CH $\bar{p}$	Posterior
`TOTAL_BATCH_SIZE`	1	0.319	1.000	nonzero in 6/6 sessions
`DEPTH`	2	0.223	0.999	nonzero in 6/6 sessions
`ASPECT_RATIO`	3	0.180	0.953	nonzero in 6/6 sessions
`DEVICE_BATCH_SIZE`	4	0.070	0.4997	nonzero in 3/6 sessions
`WARMDOWN_RATIO`	5	0.047	0.832	nonzero in 6/6 sessions

What the comparison shows. The posterior recovers three of the reference's top four parameters in every session. The harness uses it as a ranking of what to test next, not as a calibrated estimate of each parameter's relevance.

L Behavioral Analysis Details

L.1 Within-Wave Diversity Analysis

We quantify the search behavior of the three agent conditions with two per-wave metrics. A wave is a batch with at least four experiments, the incumbent is the best completed result before the wave, and all logged submissions enter the denominator.

- **Single-change share**: the share of a wave's experiments that differ from the incumbent in at most one of the 17 genesis parameters.
- **Mean Hamming distance**: the mean normalized Hamming distance (over the 17 parameters) between a wave's experiments and the incumbent.

Both dashboards suggest single-variable isolation tests for parameters whose effect is uncertain, while the unassisted agent receives no such guidance. VANILLA agents nevertheless cluster around the incumbent: from the fourth wave onward, 50.8% of their experiments change at most one of the 17 parameters relative to the best configuration completed before the wave (session range 37.5–65.4%), against 24.9% for CAUSALHARNESS (1.8–46.7%). STATFEEDBACK, which receives a dashboard but no causal ranking, clusters like VANILLA (47.8%, 5.5–90.5%). The mean normalized Hamming distance to the incumbent over waves 4–6 is 0.167 for CAUSALHARNESS, 0.113 for VANILLA, and 0.122 for STATFEEDBACK: the guided agent makes larger moves away from its incumbent (fig. 12).

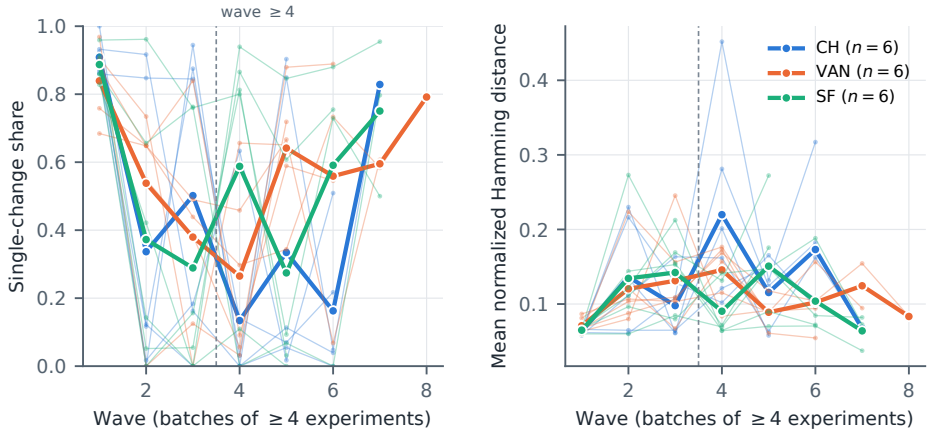

Figure 12: Search behavior per wave, 64-GPU CAUSALHARNESS, VANILLA, and STATFEEDBACK (n=6 sessions each; thin lines are sessions, bold lines are the condition mean over the sessions that reached the wave). Left: single-change share, the share of a wave's experiments that differ from the incumbent in at most one of the 17 genesis parameters. Right: mean normalized Hamming distance to the incumbent under the same definition. From wave 4 onward the single-change share is 24.9% for CAUSALHARNESS against 50.8% for VANILLA and 47.8% for STATFEEDBACK.

L.2 EIG ADOPTION ANALYSIS

A submission adopts a recommendation when the set of hyperparameters it overrides equals the parameter, or parameter pair, of a candidate shown in the most recent dashboard. Values are not matched, and submissions that modify the architecture count as non-adopted. Over the valid submissions made after the first dashboard in the six CAUSALHARNESS sessions, 28% adopt a recommendation, and 31% of those that change hyperparameters only, consistent with an advisory dashboard.

M GENERALIZATION AND FULL-LENGTH PRETRAINING

Tables 22 and 23 list the sessions behind table 2.

Table 22: Per-session AUC for the OLMo 2 1B replication (16-GPU, K=219, n=6 per condition). Labels S1–S6 are anonymous within condition.

Condition	S1	S2	S3	S4	S5	S6
CAUSALHARNESS	1.032081	1.034783	1.036756	1.037692	1.043118	1.026323
VANILLA	1.099992	1.202067	1.052265	1.262487	1.090795	1.236893
STATFEEDBACK	1.081733	1.081877	1.069850	1.057031	1.074941	1.092840

Table 23: Per-session AUC for the Mamba-1 replication (16-GPU, K=243, n=6 per condition). Labels S1–S6 are anonymous within condition.

Condition	S1	S2	S3	S4	S5	S6
CAUSALHARNESS	1.123987	1.128851	1.134625	1.129843	1.132969	1.132345
VANILLA	1.140852	1.134269	1.152727	1.147894	1.145435	1.137453
STATFEEDBACK	1.136513	1.131431	1.129440	1.148850	1.140332	1.137456

Table 24: Full-length pretraining of each session's best configuration within its first 100 valid evaluations, six sessions per condition (one GPU, 7,200 s training budget per run). `val_bpb` is the median over the six configurations (mean of the third and fourth sorted values). Benchmark accuracies are means over the same six configurations and are descriptive.

Template	Condition	`val_bpb` (median)	LAMBADA	HellaSwag	WinoGrande	ARC-Easy
Nanochat	CAUSALHARNESS	0.8830	27.4	32.1	50.4	42.4
	VANILLA	0.8975	26.7	31.5	50.4	42.2
	STATFEEDBACK	0.9010	25.9	31.4	50.0	41.5
Mamba	CAUSALHARNESS	1.1164	8.8	26.4	50.6	36.2
	VANILLA	1.1307	8.3	26.1	50.2	35.7
	STATFEEDBACK	1.1243	8.0	26.2	50.0	36.3

Proxy-budget and full-pretraining `val_bpb` correlate within each template (Spearman r=0.674, two-sided p=0.002 on Nanochat over the 18 configurations; r=0.975, p=6.6×10^{-12} on Mamba). The two templates are on different scales and are not pooled.

N EXTENDED RELATED WORK

Agent workflows for automated scientific research. Systems for automated scientific research cover an increasing share of the research pipeline, from literature-grounded ideation (Baek et al., 2025; Gottweis et al., 2026) to end-to-end paper production (Lu et al., 2026; Yamada et al., 2025; Schmidgall et al., 2025; Tang et al., 2025; Jansen et al., 2025). Benchmarks for ML research agents measure the same coverage at the task level: replicating a paper (Starace et al., 2025), solving open-ended research-engineering tasks (Wijk et al., 2025), reproducing experiments from published papers (Kon et al., 2026), or completing a workshop-scale project (Chen et al., 2025). None of them measures how efficiently an agent converts the experiments it has already run into its next decision. CAUSALHARNESS targets this question within one stage of the pipeline, the iterative experiment loop of automated ML research, and is complementary to work that widens coverage.

Automated ML research. Beyond the autoresearch loop and its cluster-scale and meta-optimized variants (section 2), LLMs propose hyperparameters directly from run histories (Zhang et al., 2023; Liu et al., 2025), tree-search scaffolds such as AIDE (Jiang et al., 2025) organize an ML-engineering agent's exploration over code, and self-evolving frameworks (Fang et al., 2025) such as AlphaEvolve (Novikov et al., 2025) and Memento-Skills (Zhou et al., 2026) advance the agent itself. None of them keeps a statistical model of which parameters drive the outcome, estimated from the agent's own interventions, and CAUSALHARNESS can supply that analysis to any of them.

Causal discovery and hyperparameter optimization. Causal Bayesian Optimization (CBO) (Aglietti et al., 2020) and MCBO (Sussex et al., 2023) propagate interventional effects through a known graph, with extensions for dynamics (Aglietti et al., 2021) and constraints (Aglietti et al., 2023), alongside the related causal-bandit line (Lattimore et al., 2016). CEO (Branchini et al., 2023) and Durand et al. (2025) relax the known-graph assumption but retain additive noise models and fixed variable sets, and each intervention is chosen and executed by the algorithm. In CAUSALHARNESS the agent submits multi-variable batches, can introduce new parameters and architecture changes at any wave, and keeps full decision authority, and GP mixtures carry structural uncertainty into every effect estimate (comparison in appendix O).

Experimental design and LLM training. Expected information gain (EIG) (Lindley, 1956; Chaloner & Verdinelli, 1995) selects maximally informative experiments. Active causal discovery uses it to learn a graph (Tong & Koller, 2001; Agrawal et al., 2019; Tigas et al., 2022), and ABCI (Toth et al., 2022) pairs a structure posterior with GP mechanisms to target a downstream causal query. These methods choose and run the experiments themselves, whereas CAUSALHARNESS returns its analysis to an agent that decides. GP-UCB (Srinivas et al., 2010) ties exploration bonuses to information gain, which our blending of structural and mechanism uncertainty builds on. πBO (Hvarfner et al., 2022) and BOPrO (Souza et al., 2021) inject expert beliefs about where the optimum lies into the acquisition function, whereas our prior is over structure. Sparse-subspace BO (Eriksson & Jankowiak, 2021) exploits importance concentration inside a fixed parameter space, and standard HPO toolkits (Falkner et al., 2018; Lindauer et al., 2022; Akiba et al., 2019; Cowen-Rivers et al., 2022) search a space the user specifies before the run. Trust-region batch BO (Eriksson et al., 2019) serves as our non-agent baseline. LLAMBO (Liu et al., 2024) puts an LLM inside BO over a fixed search space and OPRO (Yang et al., 2024) uses one as the optimizer, but neither uses causal structure. Large-scale LLM pretraining sets model size, batch size, and learning rate before the main run, either from scaling laws fitted to sweeps planned in advance (Hoffmann et al., 2022; DeepSeek-AI, 2024) or by zero-shot transfer of hyperparameters tuned on a smaller model (Yang et al., 2021). Auto-research instead learns from short proxy runs that the agent chooses adaptively, and the full-length retraining of section 5.1 tests whether those choices transfer.

Learning from self-generated data. Selecting among many noisy evaluations overfits the selection criterion and biases the reported optimum (Cawley & Talbot, 2010), and models trained on data produced by their own predecessors degrade as the generated data narrow (Shumailov et al., 2024). An auto-research agent faces both at once: it chooses its incumbent from noisy runs and learns from a history that its own choices shaped. Correcting the history for how it was collected, as interventional scoring does, is the regularizer we propose, and reliability across sessions is the quantity we report alongside mean performance.

O COMPARISON WITH CBO METHODS

Table 25 places CAUSALHARNESS next to the causal Bayesian optimization family closest to it. The differences are the ones appendix N names: a learned rather than given graph, a variable set that grows during the run, multi-variable batch interventions, and an advisory output consumed by an agent rather than a next query executed by an algorithm.

Table 25: Comparison with methods combining causal reasoning and optimization. "Classic CBO" includes MCBO (Sussex et al., 2023), which assumes a known graph and learns the unknown mechanisms. All prior methods operate on fixed variable sets, with interventions chosen by the algorithm.

	Classic CBO[a]	**Graph-Learn. CBO**[b]	**Causal BOED**[c]	**CAUSALHARNESS (ours)**
Causal graph	Known	Learned (ANM)	Learned	Learned DAG posterior
Variable set	Fixed	Fixed	Fixed	Dynamic (grows during the run)
Intervention	Algorithmic do()	Algorithmic do()	Algorithmic do()	Multi-var batch
Decision maker	Algorithm	Algorithm	Algorithm	LLM agent
Output	Next query	Next query	Next query	Ranked dashboard
Objective	Min. regret	Min. regret	Identify graph	Min. AUC regret

[a]Aglietti et al. (2020); Sussex et al. (2023) [b]Branchini et al. (2023); Durand et al. (2025) [c]Tigas et al. (2022)