

Curating Always-Loaded Context for LLM Agents: A Capacitated Assortment Model with Censored Feedback

Zexuan Liu¹ Yuning Yang² Tiancheng Zhao³

¹University of Illinois at Urbana-Champaign

²Georgia Institute of Technology

³Saint Louis University

Abstract

At the start of every session, LLM agents load a fixed context file, such as `AGENTS.md`. Each loaded token in the file is charged again in every later round of the session, and these files can degrade performance as they grow in size. However, in practice, human or automated curators usually grow these files by appending. We formulate context curation as a capacitated assortment problem. Instructions consume tokens under a finite attention capacity; adding an instruction never raises the compliance of the others, while retained instructions incur a per-session setup cost. We prove an upper bound on the optimal file size, regardless of the number of available candidate instructions, and that appending every instruction with positive standalone value can be arbitrarily worse in net value than selecting an optimal subset. A token budget also limits the loss when the token price is underestimated. We then examine what can be learned from past sessions and how this information can guide decisions to add or remove instructions. Feedback is inherently censored: the benefits and harms of loaded instructions are observable, whereas missing instructions generate feedback only when their absence causes harm. In this setting, we show that deleting instructions ignored by agents can inevitably remove helpful ones. We characterize how much evidence should be collected before adding an instruction. Besides, we bound regret when human reviewers can inspect only a limited number of edits per period. Empirical experiments further show that irrelevant rules drawn from real context files reduce language-model compliance.

1 INTRODUCTION

A coding agent that works on a Python package reads the package’s `AGENTS.md` at the start of every session. The file holds instructions such as “run `pytest` after every edit”, “add a `CHANGELOG.md` line”, “use NumPy docstrings”, “never commit to `main`”, and “always use `conda`”. The last one breaks the environment in a repository that uses `uv`. This example illustrates why instructions must be selected carefully: an unsuitable rule can lead the agent to make mistakes.

Even instructions that cause no harm incur a setup cost in each session, regardless of their relevance to the current task. Context files can raise inference cost by a fifth or more without a matching gain in task success (Gloaguen et al., 2026), since the file remains part of the prompt throughout the session, occupying context space in every model call. As the file grows, the agent may follow its instructions less reliably. At high instruction density the more instructions the file holds, the less reliably the agent follows any one of them (Jaroslawicz et al., 2025; Harada et al., 2025). However, McMillan (2026) and Zhang et al. (2026a), studying real configuration files, observe no drop in adherence as file size grows within the small ranges they test.

In practice the file grows by appending. Across 1,867 repositories such files more than double over their life, and deletions become rarer as the file ages (Chakrabarti, 2026). Tools that write the file from past sessions add to the pressure. Every session is summarized into new candidate instructions, so the pool of candidates grows without bound while the agent’s attention does not.

These lead to two questions. What should the file contain? And how can a human or automated curator learn that from the transcripts of past sessions? The second is harder than it looks, because transcripts are censored. For instructions already in the file, transcripts reveal whether the agent complied with them and whether compliance helped or hurt. For instructions not in the file, their value is usually invisible; it becomes observable only when a failure reveals the missing instruction.

We answer the first question with an assortment model. Each candidate instruction is an item with a setup cost, a probability of being relevant, and a gain when followed. The gain can be negative as a followed instruction can harm. The agent’s limited attention is shared among all instructions. Adding an instruction may lower the compliance of every instruction already in the file, so the file is an assortment whose items compete for one resource. Every loaded instruction carries a per-session setup cost. This is the fixed-cost assortment problem of operations management (Kunnumkal and Martínez-de Albéniz, 2019; Feldman and Paul, 2019) with a capacity constraint (Rusmevichientong et al., 2010; Désir et al., 2022). Two things are new: the items compete for a shared attention that dilutes as the file grows, and the pool of items grows with time.

We answer the second question with an observation model that models the censored feedback. We ask four questions: what censored transcripts identify, when a deletion policy is safe, how much corroboration to require before adding an instruction, and what a cap on human review costs.

To our knowledge, no prior work models instruction compliance as choice under a finite attention capacity, and no existing learning result handles censored feedback under an edit cap. Classical assortment models optimize revenue under a fixed choice model (Talluri and van Ryzin, 2004; Berbeglia and Joret, 2020); we instead bound the size of the optimal file under a attention-capacity bound. Learning with one-sided feedback (Helmbold et al., 2000) and learning under a bounded number of switches (Altschuler and Talwar, 2021) have each been studied separately. The unique combination with a growing candidate pool and a curator who can review only a limited number of edits has not been studied.

Contributions. (i) A model of context curation as capacitated assortment problem with a per-session setup cost, in which compliance is diluted by a shared attention load (Section 2). (ii) A structural characterization under finite attention capacity, including a file-size bound independent of the number of candidate instructions and an unbounded accretion loss, which would be optimal without attention dilution (Section 3). (iii) What censored feedback identifies: compliance and harm identify different parameters, so deleting on non-compliance is value-blind and can remove the better instruction (Section 4). (iv) An evidence threshold for adding an instruction, and regret when a reviewer accepts at most m edits per period, with the edit-cap term matched above and below (Section 4). (v) Experiments in which the optimal file size stays flat as the candidate pool grows from 200 to 3,200 items for two models, and a simulation matching the edit-cap regret bounds (Section 5).

Section 2 defines the model, Section 3 gives the structural results, Section 4 develops the learning results, and Section 5 reports the experiments. Proofs are in the appendix; Appendix A also gives the results on token budgets. The numbered results, except those relying on the smoothness-to-bound step in Assumption C.4, have complete formal proofs in Lean 4 (supplementary material).

2 MODEL

The curation workflow. An agent runs sessions. After each session a summarizer, itself a language model, reads the transcript and extracts candidate instructions: things the agent should have known, or mistakes it should not repeat. The candidates join a pool. A curator selects a subset of the pool as the file that the agent loads at the start of every later session. The pool grows with every session; the file should not.

Primitives. The pool is $[N] = \{1, \dots, N\}$. We view it as the first N items of an endless list of candidates, so that we can ask what happens as N grows. Every bound below holds for all N . The file is a subset $S \subseteq [N]$. Instruction i has a *token cost* $c_i \geq c_{\min} > 0$, and $C(S) = \sum_{i \in S} c_i$ is the cost of the file.

Instruction i is *relevant* in a session with probability p_i . When it is relevant and the agent follows it, the session gains r_i in expectation. The gain is negative for a harmful instruction. Write $w_i = p_i r_i$ for the per-session value of always following i , and $\bar{w} = \max(0, \sup_i w_i) < \infty$, the supremum over all candidates.

Given the file S , the agent follows a relevant instruction $i \in S$ with probability $q_i(S) \in [0, 1]$, its *compliance*. Compliance depends on the whole file, not on N .

Every loaded instruction is charged once per session, when the session starts and the file enters the prompt. Later turns of the session hit the prompt cache. We call $\kappa > 0$ the *token price*, so the *setup cost* of instruction i is κc_i per session and the file's setup cost is $\kappa C(S)$. The per-session value of S is its gross value minus its setup cost,

$$R(S) = R_0(S) - \kappa C(S), \quad R_0(S) = \sum_{i \in S} w_i q_i(S). \quad (1)$$

The curator solves $\max\{R(S) : S \subseteq [N], C(S) \leq B\}$ for a budget $B \geq 0$. The empty file is feasible and $R(\emptyset) = 0$, so an optimum S^* exists and $R(S^*) \geq 0$.

An instruction with positive standalone value must be followed often enough to pay its tokens. Call $\rho = \kappa c_{\min} / \bar{w}$ the *break-even compliance* ($\rho = \infty$ if $\bar{w} = 0$). If $R(\{i\}) > 0$, then $q_i(\{i\}) > \rho$.

General compliance model. The compliance functions $q_i(S)$ are otherwise arbitrary, subject to the following structural condition: adding instructions cannot increase the compliance of an instruction already in the file.

Assumption 1 (Monotone dilution). $q_i(S \cup T) \leq q_i(S)$ for every finite S, T and every $i \in S$.

Shared-dilution subclass. The shared-dilution form is a parametric subclass of the general compliance model. Each instruction has an *intrinsic compliance* $\theta_i > 0$ and an *attention load* $a_i > 0$; the file load is $A(S) = \sum_{j \in S} a_j$. A known *dilution curve* $g : [0, \infty) \rightarrow (0, 1]$, nonincreasing with $g(0) = 1$, gives the common discount

$$q_i(S) = \theta_i g(A(S)), \quad i \in S. \quad (2)$$

Here θ_i measures how readily the agent follows an instruction, a_i measures its attention load, and g describes how attention is diluted. We require $\theta_i g(a_i) < 1$ for every i . Since g is nonincreasing, (2) satisfies monotone dilution. It also gives the ratio $q_i(S)/q_j(S) = \theta_i/\theta_j$, which does not depend on S , and a single-item file has $q_i(\{i\}) = \theta_i g(a_i)$, which is the compliance a curator observes when testing an instruction on its own.

Taking $S = \{i\}$, $q_i(S) \leq q_i(\{i\})$ for every file that contains i : no instruction is followed more often in company than alone. Under (2), equality is allowed when g is flat on the added-load interval; a strict drop occurs only when g takes a smaller value.

Compliance mass and capacity. The total compliance of a file is $Q(S) = \sum_{i \in S} q_i(S)$: the expected number of followed instructions when all are relevant. Its maximum over files of n items is the *compliance mass* $M(n) = \sup\{Q(S) : |S| = n\}$. We use two nested mass conditions. The weaker one controls average compliance:

Assumption 2 (Bounded average compliance). $\limsup_{n \rightarrow \infty} M(n)/n < \rho$.

The stronger condition is a uniform capacity bound:

Assumption 3 (Attention capacity). $Q(S) \leq \bar{Q}$ for every finite S , for a constant $\bar{Q} < \infty$.

The agent follows at most $\bar{Q}$ instructions per session on average, however many it is given. Under (2) the capacity is $\bar{Q} = \chi \bar{G}$ with $\chi = \sup_i \theta_i / a_i$ and $\bar{G} = \sup_{x \geq 0} x g(x)$, which is finite exactly when $g(x) = O(1/x)$. Since $M(n) \leq \bar{Q}$, Assumption 3 implies Assumption 2. The main size bound uses capacity; the appendix also gives the weaker result when capacity fails. Under the shared-dilution form, Assumption 2 holds whenever the loads have a floor $a_i \geq a_{\min} > 0$ and $\sup_i \theta_i \cdot \lim_{x \rightarrow \infty} g(x) < \rho$, in particular whenever g vanishes at infinity. The case $\bar{Q} = 1$ is the single-choice case of classical assortment models: the agent follows at most one instruction on average. Appendix B records which classical choice models are instances of (2), and the sharper constants they give. Assumption 3 can fail under (2) while Assumption 2 holds, for example when g decays like a power below one.

3 STRUCTURAL PROPERTIES

Each instruction incurs a setup cost, while the agent has a bounded total compliance, so the value of a file cannot keep growing with its size. This section shows that the optimal file size is bounded independently of the candidate pool size. Adding every instruction that has positive standalone value can produce an arbitrarily large accretion loss. Without dilution, however, this policy is optimal whenever the token budget accommodates all such instructions. Proofs are in Appendix A.

The size bound. The bound is based on two observations. A file with nonnegative value earns at least its setup cost: $R(S) \geq 0$ means $\kappa C(S) \leq \sum_{i \in S} w_i q_i(S)$. And each followed instruction is worth at most $\bar{w}$, while the agent follows at most $\bar{Q}$ instructions per session (Assumption 3). So the setup cost of a file worth keeping is at most $\bar{w} \bar{Q}$, whatever the pool holds.

Proposition 1 (File size bound). *Under Assumption 3, every file with $R(S) \geq 0$, and in particular every optimal file S^* , satisfies*

$$\kappa C(S) \leq \bar{w} \bar{Q} \quad \text{and} \quad |S| \leq \frac{\bar{w} \bar{Q}}{\kappa c_{\min}} = \frac{\bar{Q}}{\rho}.$$

Neither bound depends on the pool size N or the budget B . If $B \geq \bar{w} \bar{Q} / \kappa$, the budget never binds. Under (2) the capacity is $\bar{Q} = \chi \bar{G}$.

Appendix A states the full version (Proposition A.1), a bound under Assumption 2 when capacity fails, and a load bound on optima that needs neither condition (Proposition A.5).

The bound gives the curator a token budget that does not exclude any optimal file. Suppose the best instruction in the pool is worth $\bar{w}/\kappa = 10,000$ always-loaded tokens per session, and the agent follows at most $\chi\bar{G} = 3$ instructions per session on average. Then no file worth keeping costs more than 30,000 tokens, and no optimal file has more than $3/\rho$ instructions, however many sessions have been summarized.

Accretion. A curator who tests each instruction alone and appends the ones with positive standalone value is running *accretion*. Write $v_i^{\text{solo}} = R(\{i\}) = w_i q_i(\{i\}) - \kappa c_i$ for the standalone value of instruction i , which is $w_i \theta_i g(a_i) - \kappa c_i$ under (2). The accretion file is $S_N^{\text{acc}} = \{i \in [N] : v_i^{\text{solo}} > 0\}$ and its closure is $\bar{S}_N^{\text{acc}} = \{i \in [N] : v_i^{\text{solo}} \geq 0\}$. Every session adds candidates, and every candidate with positive standalone value joins the file. But the file’s gross value is capped at $\bar{w}\chi\bar{G}$ while its setup cost grows with every instruction. Past $\chi\bar{G}/\rho$ instructions, each addition increase the setup cost by at least $\kappa c_{\min}$ per session.

Corollary 1 (Accretion is unboundedly suboptimal). *Assume (2) with $\chi\bar{G} < \infty$, and let S_N^* be optimal for the pool $[N]$. Every $S \subseteq [N]$ satisfies $R(S) \leq \bar{w}\chi\bar{G} - \kappa c_{\min}|S|$, hence*

$$R(S_N^*) - R(S_N^{\text{acc}}) \geq \kappa c_{\min} |S_N^{\text{acc}}| - \bar{w}\chi\bar{G},$$

which tends to infinity whenever $|S_N^{\text{acc}}| \rightarrow \infty$.

The loss grows linearly in the size of the accretion file, at rate $\kappa c_{\min}$ per instruction beyond $\chi\bar{G}/\rho$. The hypothesis $|S_N^{\text{acc}}| \rightarrow \infty$ is met as soon as infinitely many candidates with positive standalone value, which the summarizer of Section 2 supplies. As the pool grows, so does the number of instructions with positive standalone value. Appending all of them leaves an agent that follows about $\chi\bar{G}$ of them per session while paying for every one.

Without dilution. In this case, the token price alone does not make accretion suboptimal. If compliance did not depend on the file, value would be additive and accretion would be optimal whenever the file fits within the token budget. This is the limit $a_j \rightarrow 0$ of (2), which lies outside the model, and it isolates dilution as the cause of the loss above.

Proposition 2 (Without dilution, accretion is optimal). *Suppose $q_i(S) = q_i$ for every $S \ni i$, so that $v_i^{\text{solo}} = w_i q_i - \kappa c_i$. Then $R(S) = \sum_{i \in S} v_i^{\text{solo}}$. If $B \geq C(S_N^{\text{acc}})$, then S_N^{acc} is optimal, and the optimal files are exactly the S with $S_N^{\text{acc}} \subseteq S \subseteq \bar{S}_N^{\text{acc}}$ and $C(S) \leq B$.*

Without dilution, accretion is optimal whenever the token budget allows it. Under monotone dilution, every optimal file contains only instructions with nonnegative standalone value. At least one optimal file can be obtained by deleting instructions from the accretion file (Proposition A.2 in the appendix). The next section studies how a curator can use session feedback to identify which instructions to delete. The appendix also gives a tighter size bound for files whose value cannot be increased by deleting one instruction. A token budget set at this bound limits the loss when the curator underestimates the token price (Corollary A.9).

4 LEARNING WITH CENSORED FEEDBACK

Section 3 takes the values w_i and the compliances $q_i(S)$ as known. In this section, we investigate how curators can learn from session transcripts. Transcripts are censored: they show what the

loaded file did, not what an absent instruction would have done. Formal assumptions and proofs are in Appendix C.

4.1 Setup

Each session t runs with the file S_t loaded. A reviewer reads the transcript. A curator edits the file. The reviewer approves at most m edits per *epoch* of L sessions.

A transcript carries three kinds of signal.

- *Loaded* $i \in S_t$: relevant? followed? a harm flag, raised with probability $\eta > 0$ when following i did harm.
- *Absent* $i \notin S_t$: a gap sighting with probability $\delta_i p_i$, where δ_i is the reviewer’s detection rate.
- *Observed gains*: the gain $X_{i,t} \in [-1, 1]$ of a followed instruction, shown only under *magnitude feedback*.

Assumption 4 (Censored feedback). In session t the curator observes the first two signals above. All draws are independent across instructions and sessions, and their law depends on the file only through $q_i(S_t)$. The gains r_i enter no observation. Under magnitude feedback the curator also observes $X_{i,t}$ for each followed $i \in S_t$.

What transcripts identify. Let h_i be the probability that following i does harm. The *typed* case is $h_i \in \{0, 1\}$: an instruction is always helpful or always harmful. We prove what a fixed sequence of files identifies (Theorem C.5 in Appendix C.2). The compliance $q_i(S)$ of a loaded instruction is identified. The harm rate ηh_i is identified, hence whether i ever harms. The sign of the gain r_i is identified only in the typed case. The size of a gain is never identified without magnitude feedback. An absent instruction is seen only through its gap sightings, which measure a rate, not a value.

4.2 Deletion

A natural policy deletes what the agent ignores. We prove that it is blind to value.

Proposition 3 (Deleting on non-compliance is value-blind). *Suppose Assumption 4 holds, with a fixed file S played in every session. A deletion policy that uses only compliance data has the same law of decisions under every parameter vector with the same (p, q) . Take an instance of (2) with $S = \{1, 2\}$. Instruction 1 is helpful but often ignored; instruction 2 is mostly followed but worth little or harmful. Deleting the most-ignored instruction removes instruction 1 with probability tending to one as sessions accumulate.*

In the language of partial monitoring (Bartók et al., 2014), the compliance channel is feedback whose law does not separate the hypotheses the curator must tell apart. Here, the compliance observed for an instruction also depends on the other instructions in the file through dilution. The feedback on i is $q_i(S)$, which the file S sets through dilution.

4.3 Admission

An absent instruction becomes a candidate when reviewers keep sighting its gap. A spurious candidate, one that names no real gap, is also sighted now and then. Sightings are independent across sessions at a fixed *sighting rate*: ε for a spurious candidate and $\pi = \delta_j p_j$ for a real one (Assumption C.2). The curator admits at the first session τ_k by which k sightings fall within a

Algorithm 1 ETC- m : explore-then-commit under an edit cap

Require: pool $[N]$, cap m , epoch length L , budget B , exploration length n_0

- 1: $s \leftarrow \min(\lfloor m/2 \rfloor, \lfloor B/c_{\max} \rfloor)$; partition $[N]$ into batches $\mathcal{J}_1, \dots, \mathcal{J}_J$ of at most s instructions
 - 2: **for** $j = 1, \dots, J$ **do** $\triangleright$ explore
 - 3: at the first session of an epoch, replace the loaded batch by $\mathcal{J}_j$ (at most $2s \leq m$ edits)
 - 4: play $\mathcal{J}_j$ for $\lceil n_0/L \rceil$ epochs; record compliance and realized gains
 - 5: **end for**
 - 6: from the records, estimate the load $\hat{a}_i$ and the value $\hat{\beta}_i$ of every instruction (Appendix C.5)
 - 7: $\hat{S} \leftarrow \arg \max\{\hat{R}(S) : C(S) \leq B\}$ with $\hat{R}(S) = g(\hat{A}(S)) \sum_{i \in S} \hat{\beta}_i - \kappa C(S)$ $\triangleright$ commit
 - 8: move from $\mathcal{J}_J$ to $\hat{S}$, removals first, m edits per epoch; then play $\hat{S}$ until T
-

window of W sessions. Over T sessions, a spurious candidate is falsely included with probability $P_{\text{FI}}(k) = \Pr_{\varepsilon}(\tau_k \leq T)$; a real one waits $\mathbb{E}_{\pi} \tau_k$ sessions. With N_s spurious candidates, a cost c_F per false inclusion and $c_D > 0$ per session of delay, the loss of k is $\mathcal{L}(k) = \Lambda_F P_{\text{FI}}(k) + c_D \mathbb{E}_{\pi} \tau_k$ with $\Lambda_F = c_F N_s$. Write $\Lambda_D = c_D/\pi$ for the the expected delay cost until the first sighting of a real candidate. Let $\psi_W = 1 - (1 - \pi)^{W-1}$ be the chance that two real sightings share a window. Let $x = \varepsilon W$ and $x_T = \varepsilon T$ count the spurious sightings expected per window and per horizon, and let $\omega = \Lambda_D/(\Lambda_F \varepsilon T)$.

Theorem 1 (How many sightings before adding an instruction). *Assumption C.2 with $\pi, \varepsilon \in (0, 1]$, $W \geq 2$, $\Lambda_F > 0$ and $T \geq 1$. If*

$$x < \omega(2 - 1/\psi_W) \quad \text{and} \quad \omega/\psi_W < 1 - x_T/2 - x,$$

then $k = 2$ is the unique minimizer of $\mathcal{L}$ over $k \geq 1$. As $\psi_W \rightarrow 1$ the condition reads $x < \omega < 1 - x_T/2 - x$.

Two sightings give the lowest expected loss whenever the price of one wait lies between the false-inclusion costs that a third and a second sighting would prevent.

4.4 Regret under an edit cap

Sessions run $t = 1, \dots, T$, with epoch $e(t) = \lceil t/L \rceil$. The file starts empty, an edit is one addition or one removal, and each epoch has at most m edits. Every played file satisfies $C(S_t) \leq B$. A *policy* maps the history of files and observations to the next file. Its regret against the best fixed feasible file S^* is $\text{Reg}_T = \sum_{t \leq T} (R(S^*) - \mathbb{E} R(S_t))$, and $\Delta_{\max} = \max_{C(S) \leq B} [R(S^*) - R(S)]$ is the largest per-session loss. Without magnitude feedback every policy has linear regret on some instance (Proposition C.6), so we assume it from here on.

We analyze ETC- m (Algorithm 1), an explore-then-commit policy satisfying the edit cap. It tries the candidates in $J = \lceil N/s \rceil$ batches of $s = \min(\lfloor m/2 \rfloor, \lfloor B/c_{\max} \rfloor)$ instructions, where $c_{\max} = \max_i c_i$, so that one batch can be swapped for the next within one epoch's cap. Each batch is played for n_0 sessions; then it commits to the file with the largest estimated value. The estimator is in Appendix C.5.

Predicting the value of unplayed files needs *tied loads* (Assumption C.4): $\theta_i = \lambda a_i$ with λ known, and loads bounded by a known $a_{\max}$. Let n_B be the largest number of instructions a feasible file can hold, and $p_i \geq p_{\min} > 0$.

Theorem 2 (ETC- m under an edit cap). *Assumption 4 with magnitude feedback and compliance (2) under tied loads; $m \geq 2$, $B \geq c_{\max}$. For every $n_0 \geq 8 \ln(6NT)/p_{\min}$,*

$$\text{Reg}_T \leq \Delta_{\max} L \left(J + \left\lceil \frac{s + n_B}{m} \right\rceil \right) + \Lambda_T(n_0),$$

where the learning term $\Lambda_T(n_0)$ is given in Theorem C.12.

Theorem 3 (The edit-cap term is unavoidable). *Let $n_m = \lfloor N/m \rfloor$, $T \geq Ln_m$, $N, m \geq 1$, $B > 0$. There is a family of N instances of Assumption 4 with magnitude feedback and compliance (2). They share $c \leq B$, p , θ and a , with noiseless gains and no harm. Under instance j the optimal file is $\{j\}$, missing j costs at least $\Delta = p\theta g(a) - \kappa c > 0$ per session, and $\Delta_{\max} \leq \Delta + \kappa B$. Every policy that adds at most m instructions per epoch from the empty file satisfies*

$$\max_j \text{Reg}_T(j) \geq \frac{\Delta L}{2} (n_m - 1) \geq \frac{\Delta L N}{4m} \quad (N \geq 4m).$$

For $N \geq 4m$, $\kappa B \leq \Delta$ and $m \leq 2B/c_{\max}$, the edit-cap terms of the two theorems match up to a constant factor. Both are of order $(N/m)L\Delta$.

The learning term Λ_T is of order $T^{2/3}(\ln NT)^{1/3}$, the rate of explore-then-commit, and is not claimed tight (Appendix C.5). This is not the switch-budget bound of Simchi-Levi and Xu (2023), where a global budget leaves regret linear in T . Here the cap renews each epoch, so every candidate is eventually tried.

5 EXPERIMENTS

We use three experiments to study instruction files. The first measures how file size affects compliance and value. The second measures compliance in responses from a published benchmark. The third uses simulations to measure the effect of limits on file edits. Appendix D gives the detailed designs, results, and analysis plans.

Size sweep. The first experiment tests whether a larger candidate pool requires a larger file. The pool holds 3,200 items from real repository context files. Claude Sonnet 5.5 rewrites 2,560 of them as instructions about the words in a response, so that an automatic checker can score each instruction. Most of these instructions apply only to one response format, such as an email. The other 640 items stay unchanged and have zero value in our score. The tasks are 60 ManyIFEval writing tasks (Harada et al., 2025), ten in each of six response formats. The file is the system prompt, and the task is the user prompt.

We vary the pool size $N \in \{200, 800, 3200\}$ and the file size

$$b \in \{5, 10, 20, 40, 80, 200, 400, 800, 1600, N\}, \quad b \leq N.$$

The file S_b holds the b instructions that the model receives. Pools and files are nested: each larger pool contains the smaller pools, and each larger file contains the smaller files. Claude Haiku 4.5 and Claude Sonnet 5.5 receive the same files and tasks, for 2,880 runs each.

We report three quantities. The gross value $R_0(S_b)$ is the mean number of relevant instructions that the model follows, minus the number of keyword matches expected by chance. The relevant load $\ell(S_b)$ adds, over the relevant instructions in the file, the compliance of each instruction when it is loaded alone (Appendix D.1). The value subtracts a token penalty from the gross value,

$$R_{\rho_{\text{exp}}}(S_b) = R_0(S_b) - \frac{\rho_{\text{exp}}}{c_{\text{med}}} C(S_b),$$

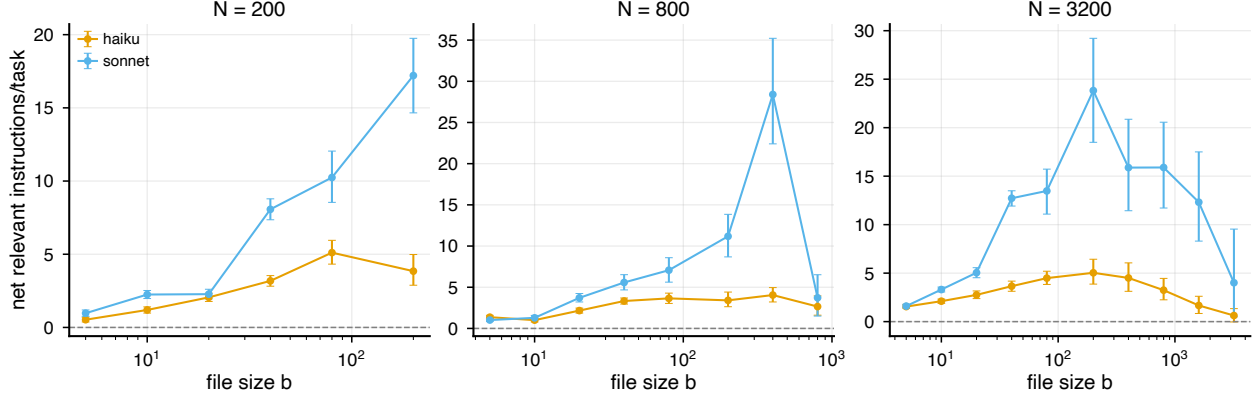


Figure 1: Gross value against file size for Claude Haiku 4.5 and Claude Sonnet 5.5. Gross value counts relevant instructions that the model follows, after adjustment for chance keyword matches. Each panel uses a different candidate pool. Points show means across 60 tasks. Bars show 95% confidence intervals from a bootstrap that resamples tasks and keeps each task’s runs together. The last point in each panel loads the entire pool.

where $C(S_b)$ is the token count of the file and $c_{\text{med}} \approx 39$ tokens is the median instruction length. Here ρ_{exp} is a normalized cost per median-length item, distinct from the theoretical break-even compliance ρ . At $\rho_{\text{exp}} = 0.1$, one median-length instruction costs 0.1 units of gross value. The measured optimum $n^*(N)$ is the size of the highest-value file among the files we test.

Size sweep: results. Large files reduce gross value in the largest pool (Figure 1). At $N = 3200$, both models reach their highest gross value at $b = 200$, where Haiku follows 5.0 relevant instructions and Sonnet follows 23.8. With the entire pool loaded, these values decrease to 0.6 and 4.0. The marginal relevant compliance, the change in gross value per unit of additional relevant load, is negative at the last step. From $b = 1600$ to $b = 3200$, it is -0.007 $[-0.013, -0.002]$ for Haiku and -0.037 $[-0.052, -0.023]$ for Sonnet (95% intervals).

The best tested file stays small as the pool grows. At $\rho_{\text{exp}} = 0.1$ and $N = 200, 800$, and 3200 , the measured optimum is 10, 5, and 10 for Haiku and 40, 20, and 40 for Sonnet. This result is consistent with the size bound of Proposition 1, although it covers only the tested files. The accretion loss, $R_{\rho_{\text{exp}}}(S_{n^*}) - R_{\rho_{\text{exp}}}(S_N)$, compares the best tested file with the file that loads the entire pool. From $N = 200$ to $N = 3200$, it increases from 15.8 to 316.7 for Haiku and from 6.0 to 320.4 for Sonnet. The loss includes both the compliance effect and the token penalty, as in Corollary 1.

Dilution also appears instruction by instruction. At $N = 3200$, the relevant instructions of the $b = 40$ file have a mean compliance of 0.27 for Haiku and 0.91 for Sonnet in that file. With the entire pool loaded, both rates decrease to 0.05, as Assumption 1 predicts at high load. The score measures instruction compliance and not the overall quality of the response.

Dilution on a published benchmark. We re-score 5,993 existing IFScale responses (Jaroslawicz et al., 2025) from 24 model configurations. Compliance is 0.92–1.00 at ten keyword requirements and 0.067–0.689 at 500 requirements. Appendix D.2 gives the fitted curves and the results for each configuration.

Regret under an edit cap. The third experiment runs ETC- m (Algorithm 1) on the instances of Theorem 3, where the optimal file is $\{j\}$ and gains have no noise. We vary $N \in \{16, 32, 64, 128, 256\}$,

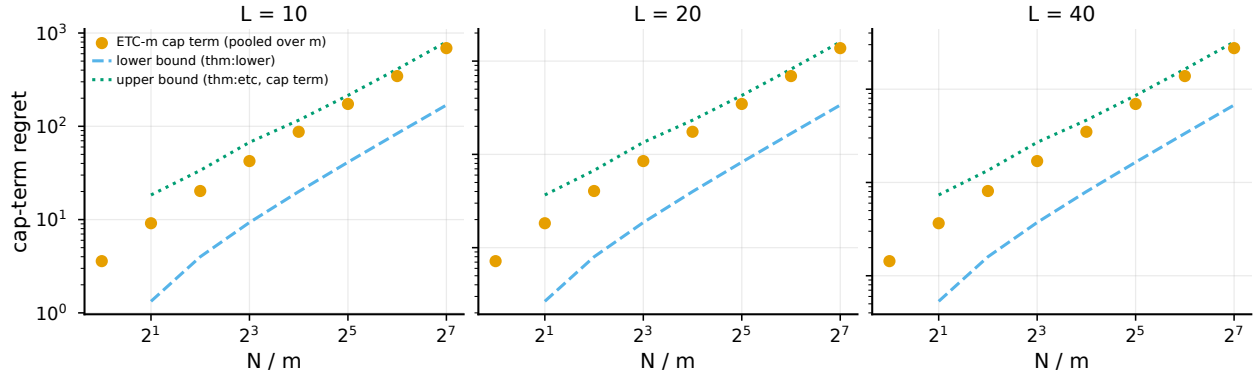


Figure 2: Regret from the edit limit against N/m . Both axes use logarithmic scales. Each panel uses a different epoch length L . Each point combines simulation settings with the same N/m . The lines show the lower bound from Theorem 3 and the cap term of the upper bound from Theorem 2. The variation across settings with the same N/m is smaller than the plotted marker.

$m \in \{2, 4, 8, 16\}$, and $L \in \{10, 20, 40\}$, which gives 60 settings (Appendix D.3). We report the cap term of the regret: the regret from the first epoch of each test batch and from the transition to the final file.

At each epoch length, the log-log slope of the cap term against N/m is 1.06, with a 95% interval of $[1.03, 1.09]$ (Figure 2). For every setting with $N \geq 4m$, the ratio of the cap term to the lower bound is between 4.1 and 5.4, and the cap term stays below the upper bound in all 60 settings. The edit-cap term is therefore of order $(N/m)L$, and the bounds of Theorems 2 and 3 differ by a constant factor.

6 RELATED WORK

Assortment with product costs and capacity. The nearest strand of operations management is assortment optimization with a per-item cost (Kunnumkal and Martínez-de Albéniz, 2019; Feldman and Paul, 2019) and with a cardinality or space constraint (Rusmevichientong et al., 2010; Désir et al., 2022). Under the choice model of Appendix B the optimal assortment is revenue-ordered (Talluri and van Ryzin, 2004); the costed and capped versions are hard, and the papers above give relaxations, approximation schemes, and learning policies. They bound revenue under a fixed choice model and supply algorithms. We bound the size of the optimal file under an attention capacity, with signed gains and a growing pool, and we study structure: the size bound, accretion, and the no-dilution converse. Hardness and approximation carry over from the cited work and are not our subject.

Other choice models. Shared dilution (2) contains the general attraction model (Gallego et al., 2015), in which an item’s share is its attraction over the sum of attractions plus an outside term. Monotone dilution (Assumption 1) is what the assortment literature calls regularity, which alone gives tight guarantees for revenue-ordered sets (Berbeglia and Joret, 2020) and places a model among regular, random-utility, and consideration-set classes (Gallego and Iravani, 2024). In attention models from choice theory the probability that an item is considered does not increase as the menu grows (Lleras et al., 2017; Cattaneo et al., 2021); the assumption there is per-item, not a cap on the sum. Multipurchase models let the total mass exceed one (Abdallah et al., 2026), as ours does, but

with additive payoffs. We know of no choice model that takes a capacity on total choice probability as a primitive.

Learning with censored feedback and edit budgets. Our observation model descends from apple tasting (Helmbold et al., 2000), in which the label is seen only after the positive action, and from partial monitoring (Bartók et al., 2014), which classifies such problems by their observation map. What is new here is that the map depends on the file through dilution. Bandits with a global switch budget have a linear regret phase when the budget is below the number of arms (Simchi-Levi and Xu, 2023; Altschuler and Talwar, 2021); switch-efficient assortment bandits treat the switch count as an output (Dong et al., 2020); batched and streaming bandits limit when a policy may change or how many arms fit in memory (Perchet et al., 2016; He et al., 2025); apple tasting with a switching cost has regret of order $\sqrt{T}$ (Cesari and Colomboni, 2026). Our cap differs from all of these: it is a per-period cap on edits that renews, so the untried part of the pool shrinks at a fixed rate, and the resulting edit-cap term is matched above and below.

LLM context and memory curation. Several recent systems edit files of the `AGENTS.md` kind from past sessions: by probing and refining the file (Shepard and Albrecht, 2026), under an edit budget with a token-cost pruner (Xia et al., 2026), by tracking which edit fixed or broke what (Ma et al., 2026), by compiling trajectories into instructions behind replay gates (Qiu et al., 2026), or by turning user corrections into enforced instructions (Zhou et al., 2026). Memory curation has also been posed as optimization: constrained retention (Kang et al., 2026), promotion under a token budget (Annapureddy and Thamalani, 2026), per-task budget allocation (Sharma et al., 2026), knapsack prompt compression (Zhang et al., 2026b), value-minus-harm scores (Wu et al., 2026), and a bandit over memory operations (Jiang et al., 2026). All use additive or submodular value, under which appending an item with positive marginal value is safe. Empirical studies document the growth of these files (Chakrabarti, 2026; Chatlatanagulchai et al., 2025), their per-session cost (Gloaguen et al., 2026; Lulla et al., 2026), and the reduction in compliance as instructions accumulate (Jaroslawicz et al., 2025; Harada et al., 2025; Vasileva, 2026), with no size effect at the sizes of real files (McMillan, 2026; Zhang et al., 2026a). None models compliance as choice: no item competes with the others for attention, no signal is censored outside the file, and no result bounds regret under a review cap.

7 CONCLUSION

We model context curation as an assortment problem with token costs, attention capacity, and censored feedback. Under our model, the optimal file size has a bound that does not depend on the candidate pool. As the number of instructions with positive standalone value grows without bound, appending all of them can become arbitrarily worse than optimal selection. The learning analysis distinguishes compliance from value. Under our feedback model, compliance and harm observations do not identify gain magnitudes. We establish conditions under which two sightings within a time window minimize the loss from false admissions and delayed admissions. With measured gains and additional identification assumptions, we derive a learning guarantee under limits on file edits. The upper and lower bounds match the edit-related regret term up to constant factors. Experiments with two language models show that large instruction files can reduce gross value, even before the token penalty. The best tested files remain small as the candidate pool grows under the chosen penalty. The published benchmark also shows lower average compliance at high instruction counts. The simulations reproduce the predicted scaling of regret caused by edit limits

within our model. An open question is which weaker conditions permit value identification when instructions interact in different ways. Another is whether curation based on measured task gains improves agent performance under actual limits on file edits.

References

- Abdallah, T., Braverman, A., and Gu, W. (2026). Multipurchase assortment optimization under a general random utility model. *Operations Research*.
- Altschuler, J. and Talwar, K. (2021). Online learning over a finite action set with limited switching. *Mathematics of Operations Research*, 46(1):179–203.
- Annapureddy, S. and Thammatani, A. P. (2026). The epistemics of agent memory: Measuring, and governing, the consolidation decision in long-horizon LLM agents.
- Bartók, G., Foster, D. P., Pál, D., Rakhlin, A., and Szepesvári, C. (2014). Partial monitoring—classification, regret bounds, and algorithms. *Mathematics of Operations Research*, 39(4):967–997.
- Berbeglia, G. and Joret, G. (2020). Assortment optimisation under a general discrete choice model: A tight analysis of revenue-ordered assortments. *Algorithmica*, 82(4):681–720.
- Cattaneo, M. D., Cheung, P., Ma, X., and Masatlioglu, Y. (2021). Attention overload.
- Cesari, T. and Colomboni, R. (2026). Two-action apple tasting with switching costs.
- Chakrabarti, K. (2026). Why does CLAUDE.md keep growing? Catastrophic remembering in agentic coding. arXiv:2608.11095.
- Chatlatanagulchai, W., Li, H., Kashiwa, Y., Reid, B., Thonglek, K., Leelaprute, P., Rungsawang, A., Manaskasemsak, B., Adams, B., Hassan, A. E., and Iida, H. (2025). Agent READMEs: An empirical study of context files for agentic coding.
- Désir, A., Goyal, V., and Zhang, J. (2022). Technical note—capacitated assortment optimization: Hardness and approximation. *Operations Research*, 70(2):893–904.
- Dong, K., Li, Y., Zhang, Q., and Zhou, Y. (2020). Multinomial logit bandit with low switching cost. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 2607–2615.
- Feldman, J. B. and Paul, A. (2019). Relating the approximability of the fixed cost and space constrained assortment problems. *Production and Operations Management*.
- Gallego, G. and Iravani, F. (2024). Axioms of rationality for discrete choice models. SSRN.
- Gallego, G., Ratliff, R., and Shebalov, S. (2015). A general attraction model and sales-based linear program for network revenue management under customer choice. *Operations Research*, 63(1):212–232.
- Galster, M., Mohsenimofidi, S., Böhme, L., Lulla, J. L., Abubakar, M. A., Treude, C., and Baltes, S. (2026). A dataset of agentic AI coding tool configurations.
- Gloaguen, T., Mündler-Sasahara, N., Müller, M. N., Raychev, V., and Vechev, M. (2026). Evaluating AGENTS.md: Are repository-level context files helpful for coding agents? arXiv:2602.11988.

- Harada, K., Yamazaki, Y., Taniguchi, M., Marrese-Taylor, E., Kojima, T., Iwasawa, Y., and Matsuo, Y. (2025). When instructions multiply: Measuring and estimating LLM capabilities of multiple instructions following. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 16506–16526.
- He, Y., Ye, Z., and Zhang, C. (2025). Understanding memory-regret trade-off for streaming stochastic multi-armed bandits. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*.
- Helmhold, D. P., Littlestone, N., and Long, P. M. (2000). Apple tasting. *Information and Computation*.
- Jaroslawicz, D., Whiting, B., Shah, P., and Maamari, K. (2025). How many instructions can LLMs follow at once? arXiv:2507.11538.
- Jiang, E. H., Zhang, Z., Wu, Y., Li, L., Liu, D., Liang, X., Sun, R., Li, Y., Sun, E., Luo, H., Kang, Z., Caliskan, A., Chang, K.-W., and Wu, Y. N. (2026). Memory as a controlled process: Learned adaptive memory management for LLM agents. arXiv:2607.13591.
- Kang, Q., Liu, M., Kai, S., Liang, K., Zhong, T., and Yuan, M. (2026). Learning what to remember: Observability-safe memory retention via constrained optimization for long-horizon language agents.
- Kunnumkal, S. and Martínez-de Albéniz, V. (2019). Tractable approximations for assortment planning with product costs. *Operations Research*.
- Lleras, J. S., Masatlioglu, Y., Nakajima, D., and Ozbay, E. Y. (2017). When more is less: Limited consideration. *Journal of Economic Theory*, 170:70–85.
- Lulla, J. L., Mohsenimofidi, S., Galster, M., Zhang, J. M., Baltes, S., and Treude, C. (2026). On the impact of AGENTS.md files on the efficiency of AI coding agents.
- Ma, X., Liu, H., Li, Y., Zhu, J., Wang, Z., Chen, Y., and Tang, X. (2026). SEPO: Evidence-grounded prompt optimization via structural editing.
- McMillan, D. (2026). Instruction adherence in coding agent configuration files: A factorial study of four file-structure variables. arXiv:2605.10039.
- Perchet, V., Rigollet, P., Chassang, S., and Snowberg, E. (2016). Batched bandit problems. *The Annals of Statistics*, 44(2).
- Pyatkin, V., Malik, S., Graf, V., Ivison, H., Huang, S., Dasigi, P., Lambert, N., and Hajishirzi, H. (2025). Generalizing verifiable instruction following.
- Qiu, L., Gao, Z., Chen, J., et al. (2026). AutoRefine: Compiling trajectories into validated typed agent artifacts.
- Rusmevichientong, P., Shen, Z.-J. M., and Shmoys, D. B. (2010). Dynamic assortment optimization with a multinomial logit choice model and capacity constraint. *Operations Research*, 58(6):1666–1680.
- Sharma, S., Pandey, N. P., Gungor, O., and Rosing, T. (2026). LIMBO: Lifelong inference-time memory and budget optimization for LLM agents.

- Shepard, A. and Albrecht, J. (2026). Probe-and-refine tuning of repository guidance for coding agents.
- Simchi-Levi, D. and Xu, Y. (2023). Phase transitions in bandits with switching constraints. *Management Science*, 69(12):7182–7201.
- Talluri, K. and van Ryzin, G. (2004). Revenue management under a general discrete choice model of consumer behavior. *Management Science*, 50(1):15–33.
- Vasileva, M. I. (2026). Large language models can follow instructions, but not many at once: Phase transitions in compositional constraint satisfaction.
- Wu, B., Ding, Z., Huang, J., and Zhao, Y. (2026). Forget to improve: On-device LLM-Agent continual learning via budget-curated memory. arXiv:2606.25115.
- Xia, P., Han, R., Wang, Z., et al. (2026). RRSI: Regularized recursive self-improvement of agent harnesses.
- Zhang, X., Wang, G., Cui, Y., Qiu, W., Li, Z., Zhu, B., and He, P. (2026a). Guardrails beat guidance: A large-scale study of rules, skills, and persistent configuration for coding agents. arXiv:2604.11088.
- Zhang, Z., Cui, S., Zhang, H., and Wang, X. (2026b). PC-SubMax: Efficient prompt compression via regularized submodular maximization.
- Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan, Y., Zhou, D., and Hou, L. (2023). Instruction-following evaluation for large language models.
- Zhou, Y., Guo, K., Zhuang, H., et al. (2026). Getting better at working with you: Compiling user corrections into runtime enforcement for coding agents.

A PROOFS AND FURTHER RESULTS FOR SECTION 3

Throughout this appendix $\Theta(S) = \sum_{i \in S} \theta_i$, $\bar{\theta} = \sup_i \theta_i$, $a_{\min} = \inf_i a_i$, $g(\infty) = \lim_{x \rightarrow \infty} g(x)$, and for $y \in (0, 1]$, $g^{-1}(y) = \sup\{x \geq 0 : g(x) \geq y\}$, which is finite iff $y > g(\infty)$. Under (2), $Q(S) = \Theta(S)g(A(S))$. Section A.1 proves the results of Section 3. Section A.2 shows that optimal files lie inside the accretion closure. Section A.3 gives the size bound under Assumption 2 when capacity fails, and a load bound that needs neither mass condition. Section A.4 sharpens the bound on files that no single deletion improves and shows that a cap at the sharpened bound insures against an underestimated token price.

A.1 Proofs for Section 3

Proposition A.1 (File size bound, full version). *Assume (2) with $\chi\bar{G} < \infty$. (i) Every S satisfies $R(S) \leq \bar{w}\Theta(S)g(A(S)) - \kappa C(S) \leq \bar{w}\chi\bar{G} - \kappa C(S)$, where $\Theta(S) = \sum_{i \in S} \theta_i$. (ii) Every S with $R(S) \geq 0$ satisfies $\kappa C(S) \leq \bar{w}\chi\bar{G}$ and $|S| \leq \chi\bar{G}/\rho$. (iii) If $B \geq \bar{w}\chi\bar{G}/\kappa$, the token budget never binds: the capped optima are exactly the uncapped optima. (iv) Every optimum S^* has $R(S^*) \geq 0$, so (ii) applies to it for every N and every B . (v) In the single-choice case $\chi\bar{G} = 1$, every S with $R(S) \geq 0$ has $C(S) \leq \bar{w}/\kappa$ and $|S| \leq 1/\rho$. The proof uses (2) only through Assumption 3 with $\bar{Q} = \chi\bar{G}$; Proposition 1 is (ii)-(iv) so stated.*

Proof of Propositions 1 and A.1. (i) Since $w_i \leq \bar{w}$, $q_i(S) \geq 0$ and $\bar{w} \geq 0$, $R_0(S) \leq \bar{w}Q(S) = \bar{w}\Theta(S)g(A(S))$. As $\theta_i \leq \chi a_i$ for every i , $\Theta(S) \leq \chi A(S)$, so $\Theta(S)g(A(S)) \leq \chi A(S)g(A(S)) \leq \chi\bar{G}$. (ii) $R(S) \geq 0$ and (i) give $\kappa C(S) \leq \bar{w}\chi\bar{G}$, and $C(S) \geq c_{\min}|S|$ gives $|S| \leq \bar{w}\chi\bar{G}/(\kappa c_{\min}) = \chi\bar{G}/\rho$. (If $\bar{w} = 0$ then $R(S) = -\kappa C(S) \geq 0$ forces $S = \emptyset$, and $\chi\bar{G}/\rho = 0$.) (iii) Let S_∞ be an uncapped optimum. It has $R(S_\infty) \geq R(\emptyset) = 0$, so $C(S_\infty) \leq \bar{w}\chi\bar{G}/\kappa \leq B$ by (ii): S_∞ is feasible under the cap, hence a capped optimum. Conversely a capped optimum S has $R(S) \geq R(S_\infty)$, so it is an uncapped optimum. (iv) $\emptyset$ is feasible for every $B \geq 0$ and $R(\emptyset) = 0$; the constants $\bar{w}$, χ and $\bar{G}$ are taken over all candidates. (v) Put $\chi\bar{G} = 1$ in (ii). $\square$

Proof of Corollary 1. Proposition A.1(i) with $C(S) \geq c_{\min}|S|$ gives the bound on $R(S)$. Apply it to $S = S_N^{\text{acc}}$ and use $R(S_N^*) \geq 0$ (Proposition A.1(iv)). $\square$

The hypothesis $|S_N^{\text{acc}}| \rightarrow \infty$ is consistent with (2) and $\chi\bar{G} < \infty$: any pool with infinitely many candidates satisfying $w_i\theta_i g(a_i) > \kappa c_i$ has it, for example identical candidates with $\theta_i = a_i = 1$ and $w_i g(1) > \kappa c_i$.

Proof of Proposition 2. Additivity is immediate from $q_i(S) = q_i$. As $v_i^{\text{solo}} > 0$ exactly on S_N^{acc} and $v_i^{\text{solo}} \leq 0$ off it, $R(S) \leq \sum_{i \in S \cap S_N^{\text{acc}}} v_i^{\text{solo}} \leq R(S_N^{\text{acc}})$, with equality iff $S_N^{\text{acc}} \subseteq S$ and $v_i^{\text{solo}} = 0$ on $S \setminus S_N^{\text{acc}}$, that is $S \subseteq \bar{S}_N^{\text{acc}}$; S_N^{acc} is feasible because $B \geq C(S_N^{\text{acc}})$. $\square$

The budget condition is needed: with $g = (1, 1)$, $c = (1, 1)$ and $B = 1$ the optima are the two singletons, and neither contains S_N^{acc} .

A.2 Optimal files lie inside the accretion closure

Under monotone dilution accretion keeps too much but never too little: an instruction with $w_i \leq 0$ is in no optimal file, and one with negative standalone value never appear in optimal file. The statement and its proof hold for any positive price in place of κ and for any budget.

Proposition A.2 (Optimal files lie inside the accretion closure). *Under Assumption 1, with w_i of any sign, every optimal S^* satisfies $S^* \subseteq \{i : w_i > 0\} \cap \bar{S}_N^{\text{acc}}$, and $S^* \cap S_N^{\text{acc}}$ is also optimal.*

Proof. Let $P = \{i : w_i > 0\}$; subsets of feasible files are feasible. Two inequalities hold for every S under Assumption 1. (L1) $R(S \cap P) \geq R(S) + \kappa C(S \setminus P)$: indeed $R(S \cap P) - R(S) = \sum_{j \in S \cap P} w_j [q_j(S \cap P) - q_j(S)] - \sum_{j \in S \setminus P} w_j q_j(S) + \kappa C(S \setminus P)$; the first sum is ≥ 0 by monotone dilution and $w_j > 0$, the second is ≤ 0 because $w_j \leq 0$. (L2) If $w_j \geq 0$ for all $j \in S$ and $i \in S$, then $R(S) - R(S \setminus \{i\}) \leq v_i^{\text{solo}}$: indeed $R(S) - R(S \setminus \{i\}) = w_i q_i(S) + \sum_{j \in S \setminus \{i\}} w_j [q_j(S) - q_j(S \setminus \{i\})] - \kappa c_i$; each bracket is ≤ 0 and $w_i q_i(S) \leq w_i q_i(\{i\})$. Now let S^* be optimal. If $S^* \not\subseteq P$, then $C(S^* \setminus P) > 0$ and (L1) gives $R(S^* \cap P) > R(S^*)$, a contradiction; so $S^* \subseteq P$. If some $i \in S^*$ had $v_i^{\text{solo}} < 0$, (L2) would give $R(S^* \setminus \{i\}) > R(S^*)$; so $S^* \subseteq \bar{S}_N^{\text{acc}}$. Finally remove the items with $v_i^{\text{solo}} = 0$ from S^* one at a time; each intermediate set S' satisfies $S' \subseteq P$, so (L2) gives $R(S' \setminus \{i\}) \geq R(S')$, the value never drops, and the last set is $S^* \cap \bar{S}_N^{\text{acc}}$. $\square$

Assumption 1 cannot be weakened to $q_i(S) \leq q_i(\{i\})$: take candidates 1, 2, 3 with $w = (1, 10, 10)$, $c_i = 1$, $\kappa = 1$ and $B = 3$, $q_1 = 0.1$ in every file, and $q_2 = q_3 = 0.9$ in every file except $\{2, 3\}$, where both are 0.1. Then $g_1 = -0.9$, yet the unique optimum is $\{1, 2, 3\}$ with $R = 15.1$: adding instruction 1 raises the compliance of 2 and 3.

A.3 Size bounds without capacity

Proposition 1 uses (2) only through the capacity $\bar{Q} = \chi \bar{G}$. When $\bar{G} = \infty$, for example when g decays like a power below one, two bounds remain. The first uses only Assumption 2 and does not require the shared-dilution parameterization. Let $\mathcal{N} = \{n \geq 0 : \kappa c_{\min} n \leq \bar{w} M(n)\}$ be the sizes at which the compliance mass can still pay for the file, and $M^*(n) = \max_{k \leq n} M(k)$.

Proposition A.3 (File size bound under bounded-average compliance). *(i) Every S satisfies $R(S) \leq \bar{w} M(|S|) - \kappa C(S)$. (ii) If $R(S) \geq 0$, then $\kappa C(S) \leq \bar{w} M(|S|)$ and $|S| \in \mathcal{N}$. (iii) Under Assumption 2, $\mathcal{N}$ is finite. With $\bar{n} = \max \mathcal{N}$, every S with $R(S) \geq 0$ has $|S| \leq \bar{n}$ and $C(S) \leq \bar{w} M^*(\bar{n})/\kappa$, and a cap $B \geq \bar{w} M^*(\bar{n})/\kappa$ never binds. (iv) Every optimal S^* has $R(S^*) \geq 0$, so (ii) and (iii) apply to it, with bounds that depend on neither N nor B .*

Proof. (i) Since $w_i \leq \bar{w}$, $q_i(S) \geq 0$ and $\bar{w} \geq 0$, $R_0(S) \leq \bar{w} Q(S) \leq \bar{w} M(|S|)$. (ii) Combine $R(S) \geq 0$ with (i) and $C(S) \geq c_{\min} |S|$. (iii) If $\bar{w} = 0$, then $\mathcal{N} = \{0\}$. Otherwise Assumption 2 gives $M(n) < \rho n$, that is $n \notin \mathcal{N}$, for all large n . Then $\kappa C(S) \leq \bar{w} M(|S|) \leq \bar{w} M^*(\bar{n})$. The cap never binds by the argument of Proposition A.1(iii), with $\bar{w} M^*(\bar{n})$ in place of $\bar{w} \chi \bar{G}$. (iv) $\emptyset$ is feasible and $R(\emptyset) = 0$; $\bar{w}$ and M are taken over all candidates. $\square$

Lemma A.4. *Under Assumption 1, $M(n+1)/(n+1) \leq M(n)/n$ for $n \geq 1$. Hence Assumption 2 holds iff $M(n_1) < \rho n_1$ for some $n_1 \geq 1$, and $\mathcal{N}$ is an initial segment of $\mathbb{N}$.*

Proof. Fix S' with $|S'| = n+1$ and let $j \in S'$ minimize $q_j(S')$. Removing the smallest of $n+1$ terms keeps at least a share $n/(n+1)$ of the sum, and Assumption 1 gives $q_i(S' \setminus \{j\}) \geq q_i(S')$, so $M(n) \geq \sum_{i \in S' \setminus \{j\}} q_i(S' \setminus \{j\}) \geq \sum_{i \in S' \setminus \{j\}} q_i(S') \geq \frac{n}{n+1} \sum_{i \in S'} q_i(S')$. Take the supremum over S' . A nonincreasing sequence has limit superior equal to its infimum, which is below ρ iff some term is. For $n \geq 1$, $n \in \mathcal{N}$ iff $M(n)/n \geq \rho$, which is preserved as n decreases, and $0 \in \mathcal{N}$. $\square$

Remarks. Without Assumption 1, $\mathcal{N}$ need not be an interval (Lemma A.4 needs the assumption), which is why $\bar{n}$ is defined as a maximum: take identical candidates with $w_i = c_i = 1$, $\kappa = 0.5$, and $q_i(S) = 0.1$ if $|S| \leq 2$, 1 if $|S| = 3$, and 0 if $|S| \geq 4$; then $\mathcal{N} = \{0, 3\}$ and $R(S) = 1.5$ for $|S| = 3$. The strict inequality in Assumption 2 cannot be relaxed: for identical instructions with $q(n) = \rho + (1-\rho)/n^2$ and $0 < \rho < 1$, Assumption 1 holds, $M(n)/n \rightarrow \rho$, every instruction is in $\bar{S}_N^{\text{acc}}$, and $R(S) = \bar{w}(1-\rho)/|S|$, so $R(\bar{S}_N^{\text{acc}}) \rightarrow 0$ while $R(S_N^*) = \bar{w}(1-\rho)$ for $B \geq c_{\min}$.

Examples. Table 1 takes identical instructions ($w_i = \bar{w} > 0$, $c_i = c_{\min}$) whose compliance $q(n)$ in a file of n is nonincreasing. Then Assumption 1 holds, $M(n) = nq(n)$, $R(S) = \bar{w} [M(n) - \rho n]$ for $|S| = n$, and $\bar{n} = \max\{n : q(n) \geq \rho\}$ (0 if $q(1) < \rho$). When M is bounded, the cost bound $\bar{w} \sup_n M(n)/\kappa$ is linear in $\bar{w}/\kappa$. Under power dilution ($0 < \beta < 1$), $\bar{n}$ grows like $(\bar{w}/\kappa)^{1/\beta}$ and, for $\alpha > \rho$, blows up as $\beta \downarrow 0$, the no-dilution limit.

Table 1: Identical instructions with compliance $q(n)$ in a file of n , where $0 < \alpha \leq 1$, $b, u > 0$, $0 < \beta < 1$, $u = v/v_0$ and $\lfloor y \rfloor^+ = \max(0, \lfloor y \rfloor)$. Assumption 2 holds in the first three rows for every ρ , and in the last iff $\alpha < \rho$. The $\bar{n}$ column assumes $\bar{w} > 0$; if $\bar{w} = 0$, then $\mathcal{N} = \{0\}$ and $\bar{n} = 0$.

Dilution	$q(n)$	$M(n)$	$\bar{n}$
Single choice	$\frac{u}{1+nu}$	< 1	$\lfloor 1/\rho - 1/u \rfloor^+$
Exponential	αe^{-bn}	$\leq \frac{\alpha}{be}$	$\lfloor \ln(\alpha/\rho)/b \rfloor^+$
Power ($\beta < 1$)	$\alpha n^{-\beta}$	$\alpha n^{1-\beta}$	$\lfloor (\alpha/\rho)^{1/\beta} \rfloor$
None	α	αn	∞ if $\alpha \geq \rho$, else 0

If the budget does not bind, the optimal size maximizes $M(n) - \rho n$, which is unimodal in each row, so it is $\lfloor n^* \rfloor$ or $\lceil n^* \rceil$ for the continuous maximizer n^* , clipped at 0: $n^* = (\sqrt{u/\rho} - 1)/u$ (single choice), $(1 - W_0(e\rho/\alpha))/b$ with W_0 the principal Lambert branch (exponential), $((1 - \beta)\alpha/\rho)^{1/\beta}$ (power), and ∞ if $\alpha > \rho$ (none). In the power row the optimal size is therefore about a fraction $n^*/\bar{n} \approx (1 - \beta)^{1/\beta} < 1/e$ of $\bar{n}$ as $\rho \rightarrow 0$. Power dilution with $\beta \geq 1$ is excluded from the table: then $M(n) = \alpha n^{1-\beta}$ is nonincreasing for $n \geq 1$, so the optimal size is 1 if $\alpha > \rho$, 0 if $\alpha < \rho$, and both 0 and 1 are optimal if $\alpha = \rho$.

The second bound applies to optima only and uses the form (2) without capacity or bounded average compliance: every instruction in an optimal file is followed often enough to cover its setup cost at the file's attention load, which bounds the load.

Proposition A.5 (Load bound on optima). *Assume (2) with $\bar{w} > 0$ and $\bar{\theta} < \infty$, and let S^* be a nonempty optimum under any budget. Then $w_i > 0$ on S^* , $\kappa c_i \leq w_i \theta_i g(A(S^*))$ for every $i \in S^*$, and $g(A(S^*)) \geq \rho/\bar{\theta}$. Hence, if $\rho/\bar{\theta} > 1$ the only optimum is $\emptyset$; and if $g(\infty) < \rho/\bar{\theta} \leq 1$ then $A(S^*) \leq g^{-1}(\rho/\bar{\theta}) < \infty$ and, when $a_{\min} > 0$, $|S^*| \leq g^{-1}(\rho/\bar{\theta})/a_{\min}$.*

Proof. By Assumption 1, $w_i > 0$ on S^* by Proposition A.2. Fix $i \in S^*$; the file $S^* \setminus \{i\}$ is feasible. By Assumption 1 and $w_l > 0$ for $l \neq i$, $R(S^* \setminus \{i\}) \geq \sum_{l \in S^* \setminus \{i\}} w_l q_l(S^*) - \kappa C(S^*) + \kappa c_i = R(S^*) - w_i q_i(S^*) + \kappa c_i$, and optimality gives $w_i q_i(S^*) \geq \kappa c_i$, that is $\kappa c_i \leq w_i \theta_i g(A(S^*))$. Since $w_i \leq \bar{w}$, $\theta_i \leq \bar{\theta}$ and $c_i \geq c_{\min}$, $g(A(S^*)) \geq \kappa c_{\min}/(\bar{w}\bar{\theta}) = \rho/\bar{\theta}$. As $g \leq 1$, this is impossible when $\rho/\bar{\theta} > 1$. When $g(\infty) < \rho/\bar{\theta} \leq 1$, $A(S^*)$ belongs to $\{x : g(x) \geq \rho/\bar{\theta}\}$, whose supremum $g^{-1}(\rho/\bar{\theta})$ is finite; and $|S^*| a_{\min} \leq A(S^*)$. $\square$

A.4 Sharpened size bound and cap insurance

Proposition 1 bounds every file with $R(S) \geq 0$. Files that no single deletion improves obey a sharper bound, because the deletion conditions bound the total setup cost by the file's own gross value. Throughout this subsection (2) holds. For a price $\mu \geq 0$ let $R_\mu(S) = R_0(S) - \mu C(S)$, so $R = R_\kappa$, and let $R_\mu^* = \max_{S \subseteq [N]} R_\mu(S)$ be the uncapped optimum at price μ . A nonempty file S is *deletion-stable* at μ if $R_\mu(S \setminus \{i\}) \leq R_\mu(S)$ for every $i \in S$. Every nonempty optimum under any budget is deletion-stable at κ , since deleting an item keeps the file feasible, and so is every nonempty file returned by a local search with deletion moves.

For a nonempty S write $A = A(S)$, $W(S) = \sum_{i \in S} w_i \theta_i$, $\phi = R_0(S) = g(A)W(S)$, $a_{\max}^S = \max_{i \in S} a_i$, and

$$\begin{aligned} e_i(S) &= \frac{1}{\theta_i} \left(\frac{1}{g(A)} - \frac{1}{g(A - a_i)} \right) \geq 0, \\ E(S) &= \sum_{i \in S} \theta_i e_i(S), \\ X_g(S) &= \Theta(S) g(A - a_{\max}^S) [1 - g(A)E(S)]. \end{aligned}$$

The quantity $e_i(S)$ is nonnegative because g is nonincreasing and $a_i > 0$. Let $\bar{X}_g = \sup X_g(S)$ over nonempty finite S with $w_i > 0$ on S ($\bar{X}_g = 0$ if there is no such S), and $a_g = \bar{w} \bar{X}_g$.

Lemma A.6 (Single deletion). *For nonempty S and $i \in S$, $R_0(S) - R_0(S \setminus \{i\}) = \theta_i g(A - a_i) [w_i - \phi e_i(S)]$.*

Proof. $R_0(S) - R_0(S \setminus \{i\}) = g(A)W(S) - g(A - a_i)(W(S) - w_i \theta_i) = w_i \theta_i g(A - a_i) - W(S)[g(A - a_i) - g(A)]$. By the definition of e_i , $g(A - a_i) - g(A) = \theta_i e_i(S) g(A) g(A - a_i)$, so the last term is $\phi \theta_i e_i(S) g(A - a_i)$. $\square$

Proposition A.7 (Sharpened size bound). *Assume (2), let w_i be of any sign, and let S be deletion-stable at $\mu > 0$. (i) $w_i > \phi e_i(S)$ for every $i \in S$, and $W(S) [1 - g(A)E(S)] > 0$. If $g(A)E(S) < 1$, in particular if $|S| = 1$, then $\phi > 0$ and $w_i > 0$ on S . (ii) If $\phi > 0$, then $w_i > 0$ on S and*

$$\mu C(S) \leq \phi \cdot \frac{g(A - a_{\max}^S)}{g(A)} [1 - g(A)E(S)] \leq \phi, \quad \text{so } R_\mu(S) \geq 0.$$

(iii) If $\phi > 0$, then $\mu C(S) \leq \bar{w} X_g(S) \leq a_g$, and $X_g(S) \leq Q(S)$; under $\chi \bar{G} < \infty$, $a_g \leq \bar{w} \chi \bar{G}$. Hence every optimum at any price $\mu > 0$ under any budget has $\mu C(S^*) \leq a_g$, and a cap $B \geq a_g / \kappa$ never binds: its optima are exactly the uncapped optima.

Proof. (i) Stability at μ and Lemma A.6 give $\theta_i g(A - a_i) [w_i - \phi e_i(S)] \geq \mu c_i > 0$, and $\theta_i g(A - a_i) > 0$, so $w_i > \phi e_i(S)$. Multiply by θ_i and sum over S : $W(S) > \phi E(S) = g(A)W(S)E(S)$, that is $W(S)[1 - g(A)E(S)] > 0$. If $1 - g(A)E(S) > 0$ then $W(S) > 0$, so $\phi = g(A)W(S) > 0$ and $w_i > \phi e_i(S) \geq 0$. For $S = \{i\}$, $E(S) = 1/g(a_i) - 1$ and $g(A)E(S) = 1 - g(a_i) < 1$. (ii) If $\phi > 0$ then $W(S) > 0$, so $1 - g(A)E(S) > 0$ by (i), and then $w_i > \phi e_i(S) \geq 0$. Sum the stability inequalities over S , using $g(A - a_i) \leq g(A - a_{\max}^S)$ (as $A - a_i \geq A - a_{\max}^S$ and g is nonincreasing) and the positive brackets: $\mu C(S) \leq g(A - a_{\max}^S) \sum_{i \in S} \theta_i [w_i - \phi e_i(S)] = g(A - a_{\max}^S) [W(S) - \phi E(S)]$, and $W(S) = \phi/g(A)$ gives the first bound. For the second, every term of $E(S)$ is nonnegative, so $E(S) \geq 1/g(A) - 1/g(A - a_{\max}^S)$, hence $1 - g(A)E(S) \leq g(A)/g(A - a_{\max}^S)$ and the first bound is at most ϕ . (iii) $W(S) \leq \bar{w} \Theta(S)$ since $w_i \leq \bar{w}$, $\theta_i > 0$ and $\bar{w} \geq 0$. Insert this in $\mu C(S) \leq g(A - a_{\max}^S) [W(S) - \phi E(S)] = g(A - a_{\max}^S) W(S) [1 - g(A)E(S)]$, where the bracket is positive by (ii), to get $\mu C(S) \leq \bar{w} X_g(S)$; by (ii) $w_i > 0$ on S , so $X_g(S) \leq \bar{X}_g$. The inequality $g(A - a_{\max}^S) [1 - g(A)E(S)] \leq g(A)$ of (ii) gives $X_g(S) \leq \Theta(S)g(A) = Q(S)$, and $Q(S) \leq \chi \bar{G}$ as in Proposition A.1(i). For the consequences, a nonempty optimum S^* at price μ is deletion-stable at μ and has $w_i > 0$ on S^* by Proposition A.2 at price μ , so $\phi > 0$ and (iii) gives $\mu C(S^*) \leq a_g$; the empty optimum has cost 0. The cap never binds by the argument of Proposition A.1(iii), with a_g in place of $\bar{w} \chi \bar{G}$. $\square$

The case split in (i) is needed. Take $g(x) = e^{-x}$, two identical instructions with $\theta_i = a_i = 1$, $w_i = -1$, $c_i = 1$, and $\mu = 0.01$. Then $R_\mu(\{1, 2\}) = -2e^{-2} - 0.02 \approx -0.291$ exceeds $R_\mu(\{1\}) = -e^{-1} - 0.01 \approx -0.378$, so $\{1, 2\}$ is deletion-stable with $\phi < 0$, both $w_i < 0$ and $R_\mu < 0$; here

$g(A)E(S) = 2(1 - e^{-1}) > 1$. Deleting a harmful instruction un-dilutes the other harmful instruction. Optima are unaffected, since they have $\phi > 0$. Under the model of Appendix B, $g(A)E(S) = Q(S) < 1$ for every file, so the sign conclusions of (i) hold for every deletion-stable file, and $X_g(S)$ has the explicit form of Appendix B; a closed form of $\bar{X}_g$ for a general g is open, and the bound $a_g \leq \bar{w}\chi\bar{G}$ is what we use.

A cap also protects a curator who misestimates the token price. One who treats tokens as free, as an appending tool does, can lose without bound; a cap at a_g/κ bounds that loss by a_g , and when the price is overestimated the cap never binds, so it adds no loss of its own. The cap must be set by whoever estimates the token price more accurately than the curator; set with the curator's own underestimate it protects nothing.

Lemma A.8 (Envelope bound). *Assume (2). For $0 < \kappa \leq \mu$, $R_\kappa^* - R_\mu^* \leq a_g \ln(\mu/\kappa)$.*

Proof. For $0 < \mu_1 \leq \mu_2$ let S_1 be an uncapped optimum at μ_1 . Then $R_{\mu_1}^* = R_{\mu_2}(S_1) + (\mu_2 - \mu_1)C(S_1) \leq R_{\mu_2}^* + (\mu_2 - \mu_1)a_g/\mu_1$, because $\mu_1 C(S_1) \leq a_g$ by Proposition A.7 applied at price μ_1 (S_1 is an optimum at μ_1 , or empty). Fix $n \geq 1$ and put $\mu_j = \kappa(\mu/\kappa)^{j/n}$ for $j = 0, \dots, n$. Summing the one-step bound over the n steps, $R_\kappa^* - R_\mu^* \leq a_g \sum_{j < n} (\mu_{j+1}/\mu_j - 1) = a_g n ((\mu/\kappa)^{1/n} - 1)$, which tends to $a_g \ln(\mu/\kappa)$ as $n \rightarrow \infty$. $\square$

Corollary A.9 (A cap insures against an underestimated token price). *Assume (2). Let the curator solve $\max\{R_0(S) - \hat{\kappa}C(S) : S \subseteq [N], C(S) \leq B\}$ exactly with an estimate $\hat{\kappa} \geq 0$ of the true price $\kappa > 0$, and call $R_\kappa^* - R(S)$ its loss. (i) If $\hat{\kappa} \leq \kappa$ and $B \geq a_g/\kappa$, the loss is at most $(\kappa - \hat{\kappa}) \min(B, a_g/\hat{\kappa})$, with $a_g/0 = \infty$; at $B = a_g/\kappa$ this is at most $a_g(1 - \hat{\kappa}/\kappa) \leq a_g$. (ii) If $\hat{\kappa} \geq \kappa$ and $B \geq a_g/\kappa$, the cap is slack ($C(S) \leq a_g/\hat{\kappa}$, and S is an uncapped optimum at $\hat{\kappa}$), and the loss is at most $\min\{a_g \ln(\hat{\kappa}/\kappa), R_\kappa^*\}$. (iii) Without a cap and with $0 < \hat{\kappa} \leq \kappa$, the loss is at most $a_g(\kappa/\hat{\kappa} - 1)$.*

Proof. Let S be the curator's file, an optimum of the capped problem at price $\hat{\kappa}$, and S_κ an uncapped optimum at κ . When $\hat{\kappa} > 0$ and $S \neq \emptyset$, S is deletion-stable at $\hat{\kappa}$ with $w_i > 0$ on S (Proposition A.2 at price $\hat{\kappa}$), so $\hat{\kappa}C(S) \leq a_g$ by Proposition A.7(iii); the same holds for $S = \emptyset$. For every file S' ,

$$R(S') - R(S) = [R_{\hat{\kappa}}(S') - R_{\hat{\kappa}}(S)] + (\kappa - \hat{\kappa})(C(S) - C(S')). \quad (3)$$

(i) $C(S_\kappa) \leq a_g/\kappa \leq B$ by Proposition A.7, so S_κ is feasible for the curator and $R_{\hat{\kappa}}(S_\kappa) \leq R_{\hat{\kappa}}(S)$. With $S' = S_\kappa$ and $\hat{\kappa} \leq \kappa$, (3) gives $R_\kappa^* - R(S) \leq (\kappa - \hat{\kappa})C(S)$. Now $C(S) \leq B$, and if $\hat{\kappa} > 0$ then $C(S) \leq a_g/\hat{\kappa}$; hence $C(S) \leq \min(B, a_g/\hat{\kappa})$. At $B = a_g/\kappa$ the bound is $(\kappa - \hat{\kappa})a_g/\kappa = a_g(1 - \hat{\kappa}/\kappa)$. (ii) $C(S) \leq a_g/\hat{\kappa}$. An uncapped optimum $S_{\hat{\kappa}}$ at $\hat{\kappa}$ has $C(S_{\hat{\kappa}}) \leq a_g/\hat{\kappa} \leq a_g/\kappa \leq B$, so it is feasible for the curator and $R_{\hat{\kappa}}(S_{\hat{\kappa}}) \leq R_{\hat{\kappa}}(S)$: S is an uncapped optimum at $\hat{\kappa}$, that is $R_{\hat{\kappa}}(S) = R_{\hat{\kappa}}^*$. Since $\hat{\kappa} \geq \kappa$, $R(S) = R_{\hat{\kappa}}(S) + (\hat{\kappa} - \kappa)C(S) \geq R_{\hat{\kappa}}^* \geq R_{\hat{\kappa}}(\emptyset) = 0$, so the loss is at most R_κ^* , and it is at most $R_\kappa^* - R_{\hat{\kappa}}^* \leq a_g \ln(\hat{\kappa}/\kappa)$ by Lemma A.8. (iii) Without a cap, S_κ is feasible, and (3) with $S' = S_\kappa$ gives $R_\kappa^* - R(S) \leq (\kappa - \hat{\kappa})C(S) \leq (\kappa - \hat{\kappa})a_g/\hat{\kappa} = a_g(\kappa/\hat{\kappa} - 1)$. $\square$

Under the model of Appendix B the bound in (iii) is attained for every $\hat{\kappa}$, so the uncapped loss is unbounded as $\hat{\kappa} \rightarrow 0$; whether it is attained for a general g is open.

B THE MULTINOMIAL LOGIT CASE

The multinomial logit model is shared dilution with $g(x) = 1/(1+x)$ and $\theta_i = a_i = u_i = v_i/v_0$.

Assumption 1' (Multinomial logit). Instruction i has weight $v_i > 0$, the agent's default behaviour has weight $v_0 > 0$, and $q_i(S) = v_i/(v_0 + V(S))$ with $V(S) = \sum_{i \in S} v_i$.

Write $U(S) = \sum_{i \in S} u_i$ and $D(S) = 1 + U(S)$, so that $q_i(S) = u_i/D(S)$ and $Q(S) = U(S)/D(S) = 1 - 1/D(S) < 1$. Every instance satisfies the requirement $\theta_i g(a_i) = u_i/(1 + u_i) < 1$ of Section 2. Here $\chi = 1$ and $\bar{G} = \sup_x x/(1+x) = 1$, so the capacity is $\bar{Q} = 1$: the agent follows at most one instruction per session on average, which is the single-choice case of Proposition A.1(v). The quantities of Appendix A.4 read, with $u_{\max} = \max_{i \in S} u_i$: $W(S) = \sum_{i \in S} w_i u_i$, $\phi = R_0(S) = W(S)/D(S)$, $e_i(S) = 1$, $E(S) = U(S)$, $g(A)E(S) = Q(S) < 1$, and

$$R_0(S) - R_0(S \setminus \{i\}) = \frac{u_i(w_i - \phi)}{D(S) - u_i}, \quad X_g(S) = \frac{U(S)}{D(S)(D(S) - u_{\max})}.$$

Since $g(A)E(S) < 1$ for every file, Proposition A.7(i) gives $w_i > \phi > 0$ on every deletion-stable file, and Proposition A.7(ii) reads $\mu C(S) \leq \phi/(D(S) - u_{\max})$.

The capped problem is capacitated multinomial logit assortment with product costs (Kunnumkal and Martínez-de Albéniz, 2019), NP-hard in general (Désir et al., 2022).

C PROOFS AND FURTHER RESULTS FOR SECTION 4

Section C.1 states the latent-variable model behind Assumption 4 and the tied-loads condition. Section C.2 states and proves Theorem C.5. Section C.3 proves Proposition 3 and gives the result on magnitude feedback. Section C.4 proves Theorem 1. Section C.5 gives the estimator of ETC- m , its error bound, the full regret bound with its tuned form, and the proofs of Theorems 2 and 3.

C.1 Formal assumptions

Assumption C.1 (Observations). The parameters are $\varpi = (p, q, h, r^+, r^-, \eta, \delta)$: relevance $p_i \in [0, 1]$; a compliance function q on pairs (S, i) with $i \in S$, with values in $[0, 1]$; harm probability $h_i \in [0, 1]$; gain $r_i^+ \geq 0$ given a helpful outcome and loss $r_i^- \geq 0$ given a harmful one, so $r_i = (1 - h_i)r_i^+ - h_i r_i^-$; harm-flag rate $\eta \in (0, 1]$; gap-detection rates $\delta_i \in (0, 1]$. A *design* is a fixed sequence of files $(S_t)_{t \geq 1}$. In session t each item i has independent latent draws $\xi_{i,t} \sim \text{Bern}(p_i)$ (relevance), $F_{i,t} \sim \text{Bern}(q_i(S_t))$ (compliance), $\zeta_{i,t} \sim \text{Bern}(h_i)$ (harmful outcome), $\vartheta_{i,t} \sim \text{Bern}(\eta)$ (flag) and $\gamma_{i,t} \sim \text{Bern}(\delta_i)$ (gap detection), independent across items and sessions, and a gain $X_{i,t}$ with $\mathbb{E}[X_{i,t} \mid \zeta_{i,t} = 0] = r_i^+$ and $\mathbb{E}[X_{i,t} \mid \zeta_{i,t} = 1] = -r_i^-$, independent of the other draws given $\zeta_{i,t}$; the gains are independent across items and sessions and, given $\zeta_{i,t}$, identically distributed. The curator observes, for $i \in S_t$: $\xi_{i,t}$; if $\xi_{i,t} = 1$, $F_{i,t}$; if $\xi_{i,t} F_{i,t} = 1$, the harm flag $H_{i,t} = \zeta_{i,t} \vartheta_{i,t}$. For $i \notin S_t$ it observes only the gap flag $\Gamma_{i,t} = \xi_{i,t} \gamma_{i,t}$. It never observes $X_{i,t}$.

The expected per-session gain of a loaded item is $p_i q_i(S) r_i = w_i q_i(S)$, as in (1). Under (2) the compliance function is $q_i(S) = \theta_i g(A(S))$ and its parameters are (θ, a) . The *typed* case is $h_i \in \{0, 1\}$ with $r_i^\pm > 0$.

Assumption C.2 (Sightings). A candidate's sighting indicators $Y_t \in \{0, 1\}$, $t \geq 1$, are i.i.d. $\text{Bern}(\pi)$, with $\pi = \varepsilon$ for a spurious candidate and $\pi = \delta_j p_j$ for a real one.

Assumption C.3 (Magnitude feedback). In addition to Assumption C.1: for $i \in S_t$ with $\xi_{i,t} F_{i,t} = 1$ the curator observes $X_{i,t} \in [-1, 1]$, so $|w_i| \leq 1$; $p_i \geq p_{\min} > 0$ for all i ; and the curator has an exact, measurable oracle for the capped problem with estimated parameters (the maximization in ETC- m below).

Assumption C.4 (Tied loads). Compliance follows (2) with $\theta_i = \lambda a_i$ for a known $\lambda > 0$, the curve g is known and continuously differentiable, and $a_i \leq a_{\max}$ for a known $a_{\max}$. With s the batch size of ETC- m , $x_s = sa_{\max}$ and $G(x) = xg(x)$,

$$\gamma_s = \min_{x \in [0, x_s]} G'(x) > 0.$$

Assumption 4 is Assumption C.1, and with magnitude feedback it is Assumption C.3. Under tied loads $Q(S) = \lambda A(S)g(A(S)) = \lambda G(A(S))$, and $\gamma_s > 0$ makes G injective on $[0, x_s]$, the range of the load of a batch. This is an identification condition, not a regularity condition: for $g(x) = e^{-\beta x}$, G peaks at $1/\beta$, and if $x_s > 1/\beta$ two loads $A_1 < 1/\beta < A_2$ with $G(A_1) = G(A_2)$ give a batch the same compliances under $a^{(1)}$ and under $a_i^{(2)} = a_i^{(1)} A_2/A_1$, while the two instances disagree on larger files.

Observable rates. Under Assumption C.1 the observations of different items and sessions are independent, so the law of the observation process under a design is the product of the laws of the single observations $O_{i,t}$. For $i \in S_t$, $O_{i,t}$ takes the values $\xi = 0$, $(\xi, F) = (1, 0)$, $(\xi, F, H) = (1, 1, 0)$ and $(1, 1, 1)$ with probabilities $1 - p_i$, $p_i(1 - q_i(S_t))$, $p_i q_i(S_t)(1 - \eta h_i)$ and $p_i q_i(S_t) \eta h_i$. For $i \notin S_t$, $O_{i,t} = \Gamma_{i,t} \sim \text{Bern}(\delta_i p_i)$. Hence two parameter vectors give the same observation law iff, for every (t, i) with $i \in S_t$, they agree on p_i , $p_i q_i(S_t)$ and $p_i q_i(S_t) \eta h_i$, and for every (t, i) with $i \notin S_t$ they agree on $\delta_i p_i$. We call these the *observable rates*. They do not involve (r^+, r^-) , and they involve q only through the values $q_i(S_t)$ on played pairs.

C.2 Identification

A parameter is *identified* under a design if any two parameter vectors with the same law of the observations agree on it.

Theorem C.5 (What censored transcripts identify). *Assumption 4 with compliance (2), under a design (S_t) . (a) For $i \in S_t$ with $p_i > 0$, $q_i(S_t)$ is identified, and nothing else about compliance is: two compliance functions that agree on the played pairs (S_t, i) give the same law. The law of the compliance data depends on the parameters only through (p, q) , so no test of $r_i > 0$ against $r_i < 0$ that uses compliance data alone has power above its size. (b) If $\{i\}$ is played and $p_i > 0$, then $\theta_i g(a_i)$ is identified. A design of singleton files identifies neither θ_i nor a_i . Within one played file, θ_i/θ_j is identified when $p_i, p_j > 0$. (c) If S and $S \cup \{i\}$ are played, $i \notin S$, and some $k \in S$ has $p_k > 0$, then $g(A(S) + a_i)/g(A(S))$ is identified, and a_i is identified once $A(S)$ is, provided the map $x \mapsto g(A(S) + x)$ is injective on the allowed interval for a_i . $A(S)$ is identified when $p_j > 0$ and $\theta_j = \lambda a_j$ on S with λ known, and $x \mapsto xg(x)$ is injective on an interval known to contain $A(S)$. (d) For $i \in S_t$ with $p_i q_i(S_t) > 0$, ηh_i is identified, hence whether i ever harms. In the typed case $\text{sign}(r_i)$ is identified; in general it is not, and $|r_i|$ never is. (e) For i never played, the observations involving i are sightings with law $\text{Bern}(\delta_i p_i)$ per session, and nothing separates δ_i from p_i . If the sighting rate δ is common and some j_0 with $p_{j_0} > 0$ is played in one session and left out in another, then δ and every p_i are identified.*

Part (a) is the key fact: compliance carries no information about value, however many sessions are observed. Testing an instruction alone measures $\theta_i g(a_i)$, but not what the instruction takes from the others. Separating the load a_i from θ_i needs two files that differ in i and a known load for the smaller one; the tied-loads condition supplies that load. Harm flags, unlike compliance, carry the sign of value in the typed case (d). One instruction played both in and out of the file calibrates the sightings of all the others (e).

Proof of Theorem C.5. (a) If $p_i > 0$, the observable rates give $q_i(S_t) = p_i q_i(S_t)/p_i$. Conversely the rates involve q only through the played pairs, so two compliance functions that agree there give the same law. The compliance data $(\xi_{i,t}, F_{i,t})$ have the product law with parameters $(p_i, q_i(S_t))$, which depend only on (p, q) . Given ϖ , let ϖ' keep (p, q, η, δ) and set $h'_i = 0$, $r_i^{+'} = 1$, $r_i^{-'} = 0$ and, for $j \neq i$, $h'_j = 1$, $r_j^{+'} = 0$, $r_j^{-'} = 1$; then $r_i' = 1 > 0 > -1 = r_j'$, and ϖ, ϖ' have the same compliance law; swapping the roles gives a ϖ'' with $r_i'' < 0$. A test based on compliance data has the same rejection probability under ϖ and under a ϖ' of the opposite sign, so its power equals its size. (b) By (a), $q_i(\{i\}) = \theta_i g(a_i)$ is identified. If only singleton files are played, item i enters the observable rates only through $\theta_i g(a_i)$; for any $a' > 0$ the pair $(\theta_i g(a_i)/g(a'), a')$ gives the same value, satisfies the requirement $\theta' g(a') = \theta_i g(a_i) < 1$, and changes no other rate, so neither θ_i nor a_i is identified. Within a played file $S \ni i, j$ with $p_i, p_j > 0$, $q_i(S)/q_j(S) = \theta_i/\theta_j$ by (2). (c) By (a), $q_k(S)$ and $q_k(S \cup \{i\})$ are identified, and their ratio is $g(A(S) + a_i)/g(A(S))$. Given $A(S)$, the injectivity condition in (c) makes a_i the unique solution. Under the stated condition every $q_j(S)$, $j \in S$, is identified, hence $Q(S) = \sum_{j \in S} \lambda a_j g(A(S)) = \lambda G(A(S))$ with $G(x) = xg(x)$, and $A(S)$ is the unique preimage of $Q(S)/\lambda$ on the known interval. (d) If $p_i q_i(S_t) > 0$, the observable rates give ηh_i , hence $\mathbf{1}\{h_i > 0\}$ as $\eta > 0$. In the typed case $h_i \in \{0, 1\}$ is read off, and $\text{sign}(r_i) = +1$ if $h_i = 0$ and -1 if $h_i = 1$, since $r_i^\pm > 0$. In general, $r_i = r_i^+(1 - h_i(1 + \lambda_i))$ with $\lambda_i = r_i^-/r_i^+$ when $r_i^+ > 0$; the observation law does not depend on (r^+, r^-) , and for $h_i \in (0, 1)$ the map $\lambda \mapsto 1 - h_i(1 + \lambda)$ takes both signs on $[0, \infty)$, so choosing $r_i^+ = 1$ and λ_i on either side of $(1 - h_i)/h_i$ gives two parameter vectors with the same law and opposite signs. Since no observable rate involves (r^+, r^-) , $|r_i|$ is not identified. (e) For an item never played, the only observations involving it are the $\Gamma_{i,t}$, i.i.d. $\text{Bern}(\delta_i p_i)$, and no observable rate of another item involves (δ_i, p_i) ; so the law depends on (δ_i, p_i) only through the product, and scaling δ_i by c and p_i by $1/c$, for c near 1, leaves it unchanged. With common δ and a bridge item j_0 , the session with $j_0 \in S_t$ identifies p_{j_0} (rate p_{j_0}), the session with $j_0 \notin S_t$ identifies δp_{j_0} , so $\delta = \delta p_{j_0}/p_{j_0}$; then every out-of-file $p_i = \delta p_i/\delta$, and every in-file p_i is an observable rate. $\square$

C.3 Deletion and the need for magnitudes

Proof of Proposition 3. Instance and claims. Take $q_1(S) = 0.1$, $q_2(S) = 0.8$, $q_1(\{1\}) = 1/2$, $q_2(\{2\}) = 8/9$, $p_1 = p_2 = 1/2$, $\kappa c_1 = \kappa c_2 = 0.001$, $r_1 = 1$ and $r_2 = \pm 0.05$, so that $R(\{1\}) = 0.249 > R(\{2\}) \in \{0.021, -0.023\}$. We show: with empirical non-compliance over each instruction's first n relevant sessions, the policy that deletes the most-ignored instruction, and the policy that deletes every instruction above a threshold $t_0 \in (0.2, 0.9)$, both delete instruction 1 and keep instruction 2 with probability tending to one as $n \rightarrow \infty$; with $r_2 = -0.05$ they leave a file worth less than the empty one; and in the typed case with $r_2 = -0.05$, the policy that deletes at the first harm flag never deletes instruction 1 and deletes instruction 2 within n' sessions with probability $1 - (1 - 0.4\eta)^{n'}$.

Law of the decision. By Theorem C.5(a) the compliance data of the first t sessions have the same law under all parameter vectors with the same (p, q) , so a measurable function of them has the same law. The ϖ' of that proof has $r_i' > 0$ and $r_j' < 0$ for $j \neq i$. The instance satisfies (2): with $g(x) = 1/(1+x)$ and $\theta = a = (1, 8)$, $q_1(\{1\}) = 1/2$, $q_2(\{2\}) = 8/9$, $q_1(S) = 1/10$, $q_2(S) = 8/10$, and $\theta_2 g(a_2) = 8/9 < 1$. Its values are $R(\{1, 2\}) = 0.05 \pm 0.02 - 0.002$, $R(\{1\}) = 0.5 \cdot 0.5 - 0.001 = 0.249$ and $R(\{2\}) = \pm 0.05 \cdot 0.5 \cdot 8/9 - 0.001 \in \{0.021, -0.023\}$. Under Assumption C.1 the latent compliance draws $(F_{i,t})_t$ of item i are i.i.d. $\text{Bern}(q_i(S))$ and independent of its relevance draws $(\xi_{i,t})_t$, so conditional on the relevance process, the draws at the first n relevant sessions are i.i.d. $\text{Bern}(q_i(S))$ (each item reaches n relevant sessions almost surely since $p_i > 0$), and Hoeffding's inequality gives, for the empirical compliance $\hat{q}_i$ over those sessions, $\Pr(\hat{q}_i \geq q_i(S) + \epsilon) \leq e^{-2n\epsilon^2}$ and $\Pr(\hat{q}_i \leq q_i(S) - \epsilon) \leq e^{-2n\epsilon^2}$. The argmax policy keeps item 1 only if $\hat{q}_1 \geq \hat{q}_2$, which requires

$\hat{q}_1 \geq 0.45$ or $\hat{q}_2 \leq 0.45$, each of probability at most $e^{-2n(0.35)^2}$; so it deletes item 1 and keeps item 2 with probability at least $1 - 2e^{-2n(0.35)^2} \rightarrow 1$. The threshold policy keeps item 1 iff $\hat{q}_1 \geq 1 - t_0 = 0.1 + (0.9 - t_0)$, of probability at most $e^{-2n(0.9-t_0)^2}$, and deletes item 2 iff $\hat{q}_2 < 1 - t_0 = 0.8 - (t_0 - 0.2)$, of probability at most $e^{-2n(t_0-0.2)^2}$; both tend to 0. With $r_2 = -0.05$ both policies end at $\{2\}$ with probability tending to 1, and $R(\{2\}) = -0.023 < 0 = R(\emptyset)$. For the harm-triggered policy in the typed case with $h_1 = 0, h_2 = 1$: item 1 is never flagged, since $\zeta_{1,t} = 0$; item 2 is flagged in a session with $S = \{1, 2\}$ with probability $p_2 q_2(S) \eta = 0.4\eta$, independently across sessions, so it is deleted within n' sessions with probability $1 - (1 - 0.4\eta)^{n'}$. $\square$

The next result is the reason Section 4.4 assumes magnitude feedback. Its setting is that of Section 4.4: a policy is a probability kernel from the history of files and observations to the next file, and given the history through session $t - 1$ and S_t , the latent draws of session t are independent of the past and follow Assumption C.1 with file S_t .

Proposition C.6 (Magnitudes are necessary). *Assumption C.1, one candidate ($N = 1$), $\kappa > 0$, $c_1 > 0$, $B \geq c_1$, $p_1 q_1(\{1\}) > 0$. There are parameter vectors ϖ, ϖ' with the same observation law such that $\emptyset$ is optimal under ϖ with margin $\Delta = R_\varpi(\emptyset) - R_\varpi(\{1\}) > 0$, $\{1\}$ is optimal under ϖ' with margin $\Delta' = R_{\varpi'}(\{1\}) - R_{\varpi'}(\emptyset) > 0$, and every policy and every horizon T have $\max(\text{Reg}_T(\varpi), \text{Reg}_T(\varpi')) \geq T \min(\Delta, \Delta')/2$.*

Proof. Take the typed helpful case ($h_1 = 0$) and two gains $r_{\text{lo}} < \kappa c_1 / (p_1 q_1(\{1\})) < r_{\text{hi}}$; let ϖ have $r_1^+ = r_{\text{lo}}$ and ϖ' have $r_1^+ = r_{\text{hi}}$, all else equal. No observation depends on r_1^+ , so the two observation laws coincide, and a policy induces the same law of $(S_t)_{t \leq T}$ under both; in particular $T_1 = \mathbb{E} \sum_{t \leq T} \mathbf{1}\{S_t = \{1\}\}$ is the same. Under ϖ , $R(\{1\}) = -\Delta < 0 = R(\emptyset)$ with $\Delta = \kappa c_1 - p_1 q_1(\{1\}) r_{\text{lo}} > 0$, so $S^* = \emptyset$ and $\text{Reg}_T(\varpi) = \Delta T_1$. Under ϖ' , $R(\{1\}) = \Delta' > 0$ with $\Delta' = p_1 q_1(\{1\}) r_{\text{hi}} - \kappa c_1$, so $S^* = \{1\}$ (feasible as $B \geq c_1$) and $\text{Reg}_T(\varpi') = \Delta'(T - T_1)$. The larger of the two is at least their average, $(\Delta T_1 + \Delta'(T - T_1))/2 \geq \min(\Delta, \Delta')T/2$. $\square$

C.4 Admission

The admission policy of Section 4.3 admits at $\tau_k = \inf\{t \geq 1 : \sum_{s=t-W+1}^t Y_s \geq k\}$, with $Y_s = 0$ for $s \leq 0$.

Lemma C.7 (Binomial tail). $\Pr(\text{Bin}(n, y) \geq j) \leq \binom{n}{j} y^j \leq (ny)^j / j!$.

Proof. If at least j of n independent trials succeed, some j -subset of the trials succeeds entirely; a union bound over the $\binom{n}{j}$ subsets gives the first inequality, and $\binom{n}{j} \leq n^j / j!$ the second. $\square$

Lemma C.8 (False inclusion). *Under Assumption C.2 with rate ε , $k \geq 1$, $W \geq 1$: (1) $P_{\text{FI}}(1) = 1 - (1 - \varepsilon)^T$; (2) $P_{\text{FI}}(k) \leq T\varepsilon \Pr(\text{Bin}(W - 1, \varepsilon) \geq k - 1) \leq T\varepsilon \binom{W-1}{k-1} \varepsilon^{k-1} \leq T\varepsilon ((W - 1)\varepsilon)^{k-1} / (k - 1)!$.*

Proof. (1) τ_1 is the first sighting. (2) The event $\tau_k = t$ requires $Y_t = 1$ and at least $k - 1$ sightings among the at most $W - 1$ sessions $t - W + 1, \dots, t - 1$, whose count is independent of Y_t and stochastically at most $\text{Bin}(W - 1, \varepsilon)$; so $\Pr(\tau_k = t) \leq \varepsilon \Pr(\text{Bin}(W - 1, \varepsilon) \geq k - 1)$. A union bound over $t \leq T$ and Lemma C.7 finish. $\square$

Lemma C.9 (Delay). *Under Assumption C.2 with rate $\pi \in (0, 1]$: (1) $\mathbb{E}\tau_k \geq k/\pi$ for all k, W ; (2) for $W \geq 2$, $\mathbb{E}\tau_2 = \frac{1}{\pi}(1 + 1/\psi_W)$ with $\psi_W = 1 - (1 - \pi)^{W-1}$; and $\mathbb{E}\tau_1 = 1/\pi$.*

Proof. (1) The policy cannot fire before the k -th sighting, whose time is a sum of k i.i.d. $\text{Geom}(\pi)$ waiting times with mean k/π . (2) Let $Z_1, Z_2, \dots$ be the i.i.d. $\text{Geom}(\pi)$ times between sightings

(Z_1 is the first sighting time). The window count changes only at sightings, and at the j -th sighting the window holds two sightings iff $Z_j \leq W - 1$. So the policy fires at the j^* -th sighting, $j^* = \min\{j \geq 2 : Z_j \leq W - 1\}$, and $\tau_2 = Z_1 + \dots + Z_{j^*}$. Here $j^* - 1 \sim \text{Geom}(\psi_W)$ with $\psi_W = \Pr(Z \leq W - 1)$, and j^* is a stopping time for $(Z_j)_{j \geq 2}$ with finite mean, so Wald's identity gives $\mathbb{E}\tau_2 = \mathbb{E}Z_1 + \mathbb{E}[j^* - 1]\mathbb{E}Z = 1/\pi + 1/(\psi_W\pi)$. Finally $\tau_1 = Z_1$. $\square$

Proof of Theorem 1. Write $\beta_\varepsilon = 1 - (1 - \varepsilon)^{W-1}$ for the chance that a spurious sighting has another inside its window. We first show that $k = 2$ is the unique minimizer under the two conditions

$$\begin{aligned} \text{(A)} \quad & \Lambda_F [1 - (1 - \varepsilon)^T - T\varepsilon\beta_\varepsilon] > \Lambda_D/\psi_W, \\ \text{(B)} \quad & \Lambda_F T\varepsilon\beta_\varepsilon < \Lambda_D(2 - 1/\psi_W), \end{aligned}$$

which need only $\varepsilon \in [0, 1]$ and $\Lambda_F \geq 0$, and then that the band implies them. By Lemmas C.8(1) and C.9, $\mathcal{L}(1) = \Lambda_F(1 - (1 - \varepsilon)^T) + \Lambda_D$. By Lemma C.8(2) with $\Pr(\text{Bin}(W - 1, \varepsilon) \geq 1) = \beta_\varepsilon$ and Lemma C.9(2), $\mathcal{L}(2) \leq \Lambda_F T\varepsilon\beta_\varepsilon + \Lambda_D(1 + 1/\psi_W)$. By Lemma C.9(1), $\mathcal{L}(k) \geq \Lambda_D k \geq 3\Lambda_D$ for $k \geq 3$. Condition (A) says exactly that the bound on $\mathcal{L}(2)$ is below $\mathcal{L}(1)$, and (B) that it is below $3\Lambda_D$; so $\mathcal{L}(2) < \mathcal{L}(k)$ for every $k \neq 2$, $k \geq 1$. For the band, use $(1 - \varepsilon)^T \leq e^{-xT}$ and $\beta_\varepsilon \leq (W - 1)\varepsilon \leq x$: (A) follows from $\Lambda_F(1 - e^{-xT} - xTx) > \Lambda_D/\psi_W$, and (B) from $\Lambda_F xTx < \Lambda_D(2 - 1/\psi_W)$. Dividing by $\Lambda_F \varepsilon T > 0$, the second is $x < \omega(2 - 1/\psi_W)$, and the first follows from $1 - e^{-xT} \geq xT - x^2 T/2$ and $1 - xT/2 - x > \omega/\psi_W$. $\square$

In the regime $W\pi \gg k$ the delay is about k/π . When false inclusion is rare, Lemma C.8(2) bounds $P_{\text{FI}}(k)$ by about $\frac{T}{W} \frac{x^k}{(k-1)!}$, and counting only the T/W disjoint windows gives the lower estimate $\frac{T}{W} \frac{x^k}{k!}$, a factor k smaller.

C.5 Regret under an edit cap

Constants. Let $c_{\max} = \max_i c_i$, $s = \min(\lfloor m/2 \rfloor, \lfloor B/c_{\max} \rfloor) \geq 1$ and $J = \lceil N/s \rceil$. Let $\bar{G} = \sup_{x \geq 0} G(x)$ and $B' = \min(B, \lambda \bar{G}/\kappa)$, read as B when $\bar{G} = \infty$; by Proposition A.1(ii), (iv) with $\chi = \lambda$ and $\bar{w} \leq 1$, some optimal file has $C(S^*) \leq B'$. Let $\mathcal{S}' = \{S \subseteq [N] : C(S) \leq B'\}$ and $n_B = \max\{|S| : S \in \mathcal{S}'\}$. With $x_s = sa_{\max}$ and $x_B = n_B a_{\max}$, put $g_s = g(x_s)$,

$$L_{1/g} = \max_{x \in [0, x_s]} \frac{|g'(x)|}{g(x)^2}, \quad K_g = \sup_{x \neq y \in [0, x_B]} \frac{x|g(x) - g(y)|}{|x - y|} \leq x_B \max_{[0, x_B]} |g'|,$$

so that $1/g$ is $L_{1/g}$ -Lipschitz on $[0, x_s]$. Finally $\epsilon_0 = \sqrt{\ln(6NT)/(n_0 p_{\min})}$ and

$$\varepsilon_{\text{SD}}(n_0) = \epsilon_0 \left[\frac{2 + K_g}{g_s} + \frac{(1 + K_g) L_{1/g} s}{\lambda \gamma_s} \right].$$

ETC- m . Fix an exploration length n_0 , in sessions per batch. Partition $[N]$ into batches $\mathcal{J}_1, \dots, \mathcal{J}_J$ of at most s items. For $j = 1, \dots, J$, play $\mathcal{J}_j$ for $\lceil n_0/L \rceil$ epochs, switching at the first session of an epoch ($\leq 2s \leq m$ edits); let I_j be the set of sessions in which $\mathcal{J}_j$ is played, $|I_j| = n \geq n_0$. For $i \in \mathcal{J}_j$ let $\hat{q}_i$ be its empirical compliance over its relevant sessions in I_j and $\bar{Y}_i$ the mean over I_j of $Y_{i,t} = \xi_{i,t} F_{i,t} X_{i,t}$. Estimate the batch load and the item parameters by

$$\hat{A}_j = G^{-1} \left(\text{clip}_{[0, G(x_s)]} \left(\frac{1}{\lambda} \sum_{l \in \mathcal{J}_j} \hat{q}_l \right) \right), \quad \hat{a}_i = \text{clip}_{[0, a_{\max}]} \left(\frac{\hat{q}_i}{\lambda g(\hat{A}_j)} \right), \quad \hat{\beta}_i = \frac{\bar{Y}_i}{g(\hat{A}_j)},$$

where G^{-1} is the inverse of G on $[0, x_s]$. For $S \in \mathcal{S}'$ let $\hat{A}(S) = \sum_{i \in S} \hat{a}_i$ and

$$\hat{R}(S) = g(\hat{A}(S)) \sum_{i \in S} \hat{\beta}_i - \kappa C(S).$$

Let $\hat{S}$ maximize $\hat{R}$ over $\mathcal{S}'$. Move from $\mathcal{J}_J$ to $\hat{S}$, removals first, m edits per epoch at the first session of each epoch, and play $\hat{S}$ thereafter.

The targets of the estimator are $\beta_i = w_i \theta_i = \lambda w_i a_i$ and $R(S) = g(A(S)) \sum_{i \in S} \beta_i - \kappa C(S)$, which is (1) under (2).

Lemma C.10 (Batch errors). *Assumptions C.3 and C.4. Let $\mathcal{G}_0$ be the event that for every $i \in [N]$, with j the batch of i : (G1) $N_i = \sum_{t \in I_j} \xi_{i,t} \geq np_i/2$; (G2) $|\hat{q}_i - q_i(\mathcal{J}_j)| \leq \epsilon_0$; (G3) $|\bar{Y}_i - \mathbb{E}Y_i| \leq 2\epsilon_0$, where $\mathbb{E}Y_i = w_i q_i(\mathcal{J}_j) = \beta_i g(A(\mathcal{J}_j))$. If $n_0 \geq 8 \ln(6NT)/p_{\min}$, then $\Pr(\mathcal{G}_0^c) \leq 5/(6T)$, and on $\mathcal{G}_0$, for every j and $i \in \mathcal{J}_j$,*

$$|\hat{A}_j - A(\mathcal{J}_j)| \leq \frac{s\epsilon_0}{\lambda\gamma_s} =: \epsilon_A, \quad |\hat{a}_i - a_i| \leq \frac{\epsilon_0}{\lambda g_s} + \frac{L_{1/g}}{\lambda} \epsilon_A =: \epsilon_a, \quad |\hat{\beta}_i - \beta_i| \leq \frac{2\epsilon_0}{g_s} + L_{1/g} \epsilon_A =: \epsilon_\beta.$$

Proof. Probability. The exploration schedule is fixed in advance, so for $i \in \mathcal{J}_j$ the latent draws $(\xi_{i,t}, F_{i,t}, X_{i,t})_{t \in I_j}$ are i.i.d. across t by Assumption C.1, and $X_{i,t}$ is independent of $(\xi_{i,t}, F_{i,t})$ because it depends on the other draws only through $\zeta_{i,t}$, which is independent of them; hence $\mathbb{E}Y_i = p_i q_i(\mathcal{J}_j) r_i = w_i q_i(\mathcal{J}_j)$. By the multiplicative Chernoff bound, $\Pr(N_i < np_i/2) \leq e^{-np_i/8} \leq e^{-n_0 p_{\min}/8} \leq 1/(6NT)$. Conditional on the relevance draws, the compliance draws at the relevant sessions are i.i.d. $\text{Bern}(q_i(\mathcal{J}_j))$, so on (G1) Hoeffding's inequality gives $\Pr(|\hat{q}_i - q_i(\mathcal{J}_j)| > \epsilon_0 \mid \xi) \leq 2e^{-2N_i \epsilon_0^2} \leq 2e^{-np_i \epsilon_0^2} \leq 2e^{-\ln(6NT)} = 1/(3NT)$. The $Y_{i,t}$ are i.i.d. in $[-1, 1]$, so Hoeffding's inequality gives $\Pr(|\bar{Y}_i - \mathbb{E}Y_i| > 2\epsilon_0) \leq 2e^{-2n\epsilon_0^2} \leq 1/(3NT)$. A union bound over the N items gives $\Pr(\mathcal{G}_0^c) \leq N(1/(6NT) + 2/(3NT)) = 5/(6T)$.

Load. Write $A_j = A(\mathcal{J}_j) \leq x_s$ and $Q_j = Q(\mathcal{J}_j) = \lambda G(A_j)$, so $Q_j/\lambda \in [0, G(x_s)]$. By (G2), $|\sum_i \hat{q}_i/\lambda - Q_j/\lambda| \leq s\epsilon_0/\lambda$, and clipping to $[0, G(x_s)]$, which contains Q_j/λ , does not increase the distance. On $[0, x_s]$, $G' \geq \gamma_s > 0$, so G is increasing there and G^{-1} is $1/\gamma_s$ -Lipschitz on $[0, G(x_s)]$; hence $|\hat{A}_j - A_j| \leq \epsilon_A$.

Items. Since $\hat{A}_j, A_j \in [0, x_s]$, $g(\hat{A}_j) \geq g_s$ and $|1/g(\hat{A}_j) - 1/g(A_j)| \leq L_{1/g} \epsilon_A$. With $q_i = q_i(\mathcal{J}_j) = \lambda a_i g(A_j) \leq 1$,

$$\frac{\hat{q}_i}{\lambda g(\hat{A}_j)} - a_i = \frac{\hat{q}_i - q_i}{\lambda g(\hat{A}_j)} + \frac{q_i}{\lambda} \left(\frac{1}{g(\hat{A}_j)} - \frac{1}{g(A_j)} \right),$$

of absolute value at most ϵ_a ; clipping to $[0, a_{\max}] \ni a_i$ does not increase the error. Likewise, with $|\mathbb{E}Y_i| \leq 1$, $\hat{\beta}_i - \beta_i = (\bar{Y}_i - \mathbb{E}Y_i)/g(\hat{A}_j) + \mathbb{E}Y_i (1/g(\hat{A}_j) - 1/g(A_j))$, of absolute value at most ϵ_β . $\square$

Lemma C.11 (Plug-in stability). *Assumptions C.3 and C.4. On $\mathcal{G}_0$, every $S \subseteq [N]$ with $|S| \leq n_B$ satisfies $|\hat{R}(S) - R(S)| \leq |S| \varepsilon_{\text{SD}}(n_0)$.*

Proof. Both $A(S) = \sum_S a_i$ and $\hat{A}(S) = \sum_S \hat{a}_i$ lie in $[0, x_B]$, and $|\hat{A}(S) - A(S)| \leq |S| \epsilon_a$ by Lemma C.10. Decompose

$$\hat{R}(S) - R(S) = g(\hat{A}(S)) \sum_{i \in S} (\hat{\beta}_i - \beta_i) + (g(\hat{A}(S)) - g(A(S))) \sum_{i \in S} \beta_i.$$

The first term is at most $|S| \epsilon_\beta$ in absolute value, as $g \leq 1$. For the second, $|\sum_S \beta_i| \leq \lambda \sum_S |w_i| a_i \leq \lambda A(S)$, and by the definition of K_g , $A(S) |g(\hat{A}(S)) - g(A(S))| \leq K_g |\hat{A}(S) - A(S)| \leq K_g |S| \epsilon_a$. So $|\hat{R}(S) - R(S)| \leq |S| (\epsilon_\beta + \lambda K_g \epsilon_a)$, and $\epsilon_\beta + \lambda K_g \epsilon_a = \varepsilon_{\text{SD}}(n_0)$ by the definitions of ϵ_a , ϵ_β and ϵ_A . $\square$

The factor $A(S)$ in front of the difference of g is what keeps the commit error linear in n_B : a per-item bound on $|\hat{q}_i(S) - q_i(S)|$ would carry n_B once more through $\hat{A}(S)$.

Theorem C.12 (ETC- m under an edit cap, full form). *Assumptions C.1, C.3 and C.4; $m \geq 2$, $B \geq c_{\max}$, $c_i > 0$, $\kappa > 0$, $N, L, T \geq 1$. If $n_0 \geq 8 \ln(6NT)/p_{\min}$, then*

$$\text{Reg}_T \leq \Delta_{\max} L \left(J + \left\lceil \frac{s + n_B}{m} \right\rceil \right) + \Lambda_T(n_0), \quad \Lambda_T(n_0) = \Delta_{\max} J n_0 + 2n_B T \varepsilon_{\text{SD}}(n_0) + \Delta_{\max}.$$

Proof. Feasibility. A batch costs at most $sc_{\max} \leq B$, the sets played during the move are subsets of the last batch or of $\hat{S}$, and $C(\hat{S}) \leq B' \leq B$; so every played file is feasible and every session's expected regret is at most $\Delta_{\max}$. A batch switch uses at most $2s \leq m$ edits and each epoch of the move at most m .

Step 1 (decomposition). Exploration lasts $JL \lceil n_0/L \rceil < J(n_0 + L)$ sessions and the move $L \lceil (s + n_B)/m \rceil$ sessions, in all at most $\Delta_{\max} [L(J + \lceil (s + n_B)/m \rceil) + Jn_0]$ regret. Every later session costs $R(S^*) - R(\hat{S})$.

Step 2 (commit error). Some optimal file lies in $\mathcal{S}'$ and every $S \in \mathcal{S}'$ has $|S| \leq n_B$. On $\mathcal{G}_0$, Lemma C.11 gives $|\hat{R}(S) - R(S)| \leq n_B \varepsilon_{\text{SD}}(n_0)$ for all $S \in \mathcal{S}'$, so $R(S^*) - R(\hat{S}) \leq [R(S^*) - \hat{R}(S^*)] + [\hat{R}(S^*) - \hat{R}(\hat{S})] + [\hat{R}(\hat{S}) - R(\hat{S})] \leq 2n_B \varepsilon_{\text{SD}}(n_0)$, the middle bracket being ≤ 0 by the choice of $\hat{S}$.

Step 3 (expectation). The commit phase lasts at most T sessions, so its regret is at most $2n_B T \varepsilon_{\text{SD}}(n_0)$ on $\mathcal{G}_0$ and at most $T \Delta_{\max}$ off $\mathcal{G}_0$, whose probability is at most $5/(6T) \leq 1/T$ by Lemma C.10; in expectation this adds at most $\Delta_{\max}$. Summing with Step 1 gives the bound. $\square$

Corollary C.13 (Tuned exploration). *In the setting of Theorem C.12, let $n_{\min} = \lceil 8 \ln(6NT)/p_{\min} \rceil$, $C_1 = 2n_B [(2 + K_g)/g_s + (1 + K_g)L_{1/g} s/(\lambda \gamma_s)] \sqrt{\ln(6NT)/p_{\min}}$, so that $2n_B \varepsilon_{\text{SD}}(n_0) = C_1 n_0^{-1/2}$, and $n_0 = \max\{n_{\min}, \lceil (TC_1/(2J\Delta_{\max}))^{2/3} \rceil\}$, reading $y/0 = 0$. Then*

$$\Lambda_T(n_0) \leq 1.9 (J\Delta_{\max})^{1/3} (C_1 T)^{2/3} + J\Delta_{\max} (n_{\min} + 1) + \Delta_{\max}.$$

Proof. If $\Delta_{\max} = 0$ then $\text{Reg}_T \leq T\Delta_{\max} = 0$ and there is nothing to prove. Otherwise the first two terms of $\Lambda_T(n_0)$ are $\mathbf{a}n_0 + \mathbf{b}n_0^{-1/2}$ with $\mathbf{a} = J\Delta_{\max}$ and $\mathbf{b} = C_1 T$; put $n^* = (\mathbf{b}/(2\mathbf{a}))^{2/3}$, so $\mathbf{a}n^* = 2^{-2/3}\mathbf{a}^{1/3}\mathbf{b}^{2/3}$ and $\mathbf{b}(n^*)^{-1/2} = 2^{1/3}\mathbf{a}^{1/3}\mathbf{b}^{2/3}$. If $n_0 = \lceil n^* \rceil \geq n_{\min}$, then $\mathbf{a}n_0 + \mathbf{b}n_0^{-1/2} \leq \mathbf{a}(n^* + 1) + \mathbf{b}(n^*)^{-1/2} = 3 \cdot 2^{-2/3}\mathbf{a}^{1/3}\mathbf{b}^{2/3} + \mathbf{a} \leq 1.9 \mathbf{a}^{1/3}\mathbf{b}^{2/3} + \mathbf{a}$. If $n_0 = n_{\min} > n^*$, then $\mathbf{a}n_0 = \mathbf{a}n_{\min}$ and $\mathbf{b}n_0^{-1/2} < \mathbf{b}(n^*)^{-1/2} \leq 1.9 \mathbf{a}^{1/3}\mathbf{b}^{2/3}$. In both cases the two terms are at most $1.9 \mathbf{a}^{1/3}\mathbf{b}^{2/3} + \mathbf{a}(n_{\min} + 1)$. $\square$

The tuning uses $\Delta_{\max}$, which the curator does not know. Tuning with any known $\bar{\Delta} \geq \Delta_{\max}$ in its place, for instance $\bar{\Delta} = n_B + \kappa B$ (as $|w_i| \leq 1$ and $q_i \leq 1$), gives the same bound with $\bar{\Delta}$ in place of $\Delta_{\max}$ in the first two terms, by the same proof with $\mathbf{a} \leq J\bar{\Delta}$.

Proof of Theorem 2. The hypotheses of Theorem 2 are those of Theorem C.12, with n_B the largest size of a file in $\mathcal{S}'$, so the bound is Theorem C.12. Under the tuning of Corollary C.13, C_1 is $O(\sqrt{\ln(NT)})$ and $n_{\min}$ is $O(\ln(NT))$ with the other parameters fixed, so $\Lambda_T = O(T^{2/3}(\ln NT)^{1/3})$. For the edit-cap term, if $m \leq 2B/c_{\max}$ then $s = \lfloor m/2 \rfloor \geq m/3$, so $J \leq 3N/m + 1$, and $n_B \leq N$ gives $\lceil (s + n_B)/m \rceil \leq N/m + 2$; hence the term is at most $\Delta_{\max} L(4N/m + 3)$. $\square$

Proof of Theorem 3. Instance. Fix $c \in (0, B]$, $p \in (0, 1]$, $\theta > 0$ and $a > 0$ with $\theta g(a) < 1$ and $\kappa c < p\theta g(a)$; such values exist since c may be taken small. All items share c , p , $\theta_i = \theta$ and $a_i = a$; harm $h_i = 0$; gap detection δ ; under instance j the gains are $r_j = 1$ and $r_i = 0$ for $i \neq j$, observed without noise ($X_{i,t} \equiv r_i$). Then $R(S) = p\theta g(|S|a) \mathbf{1}\{j \in S\} - \kappa c|S|$. If $j \in S$ this is at

most $p\theta g(a) - \kappa c = R(\{j\})$, as g is nonincreasing, and if $j \notin S$ it is $-\kappa c|S| \leq 0 < R(\{j\})$. So $S_j^* = \{j\}$, and $R(S_j^*) - R(S_t) \geq \Delta$ whenever $j \notin S_t$; also $R(S) \geq -\kappa C(S) \geq -\kappa B$ for feasible S , so $\Delta_{\max} \leq \Delta + \kappa B$.

Reduction. Let instance 0 have $r_i = 0$ for all i . For a file $S \not\ni j$, the observation law of a session with file S is the same under instance j and instance 0: in-file items have the same compliance $\theta g(|S|a)$ and the same gain 0 under both, and j contributes only $G_{j,t} \sim \text{Bern}(\delta p)$. This holds for every joint law of the flags within a session, since the flag laws do not depend on the gains. Let $\sigma_j = \min\{t : j \in S_t\}$ be the first session that tries j . The law of a history $(S_1, O_1, \dots, S_t, O_t)$ is the product of the policy kernels $P(S_s \mid \text{past})$ and the session laws $K(O_s \mid S_s)$; on the event $\{\sigma_j > t\}$ every factor agrees under instances j and 0, so $\Pr_j(\sigma_j > t) = \Pr_0(\sigma_j > t)$.

Counting. From $S_0 = \emptyset$ with at most m additions per epoch, $|\bigcup_{s \leq t} S_s| \leq \sum_{s \leq t} |S_s \setminus S_{s-1}| \leq m e(t)$ almost surely, so $\sum_j \Pr_0(\sigma_j \leq t) \leq \mathbb{E}_0 |\bigcup_{s \leq t} S_s| \leq m e(t)$, and $\sum_j \Pr_j(j \notin S_t) \geq \sum_j \Pr_0(\sigma_j > t) \geq N - m e(t)$.

Summing. $\text{Reg}_T(j) \geq \Delta \sum_{t \leq T} \Pr_j(j \notin S_t)$, so, using $T \geq Ln_m$ and $N - me \geq 0$ for $e \leq n_m$,

$$\sum_j \text{Reg}_T(j) \geq \Delta \sum_{e=1}^{n_m} L(N - me) = \Delta L \left(Nn_m - \frac{mn_m(n_m + 1)}{2} \right) \geq \Delta L \frac{N(n_m - 1)}{2},$$

because $mn_m \leq N$. The maximum over j is at least the average, $(\Delta L/2)(n_m - 1)$. For $N \geq 4m$, $n_m - 1 > N/m - 2 \geq N/(2m)$. $\square$

D EXPERIMENTAL DETAILS

The plan of each experiment is paraphrased below, and every deviation from the plan is listed. Model calls go to hosted Claude models through their command-line client; the context file is the whole system prompt, with no tools and prompt caching off. Intervals are 95% percentile intervals of a task-cluster bootstrap with 2,000 replicates unless stated.

D.1 Size sweep

Purpose. Proposition 1 bounds the optimal file size independently of the pool, and Corollary 1 says the loss from loading the whole pool grows with the pool. The size sweep grows a pool of real context-file instructions from 200 to 3,200 items and measures value against file size at each pool size, for two models on the same files.

Data. Candidates are root context-file list items from a public corpus of coding-agent configuration files (Galster et al., 2026) (CC BY 4.0 compilation; file contents keep their repository licenses, so only permissive repositories are used and item text stays out of the released data). Tasks are ManyIFEval prompts (Harada et al., 2025) (CC BY 4.0); verifiers are the IFEval (Zhou et al., 2023) and IFBench (Pyatkin et al., 2025) checkers (Apache-2.0). The subjects are Claude Haiku 4.5 and Claude Sonnet 5.5.

Design. The pool holds 3,200 items drawn from the tier whose verbatim text cannot apply to a reply, in one seeded order: one item per repository for the first 1,901 items and a second item per repository after that. $N \in \{200, 800, 3200\}$ takes prefixes. Every fifth item stays verbatim as a zero-value candidate (640 items). The other 2,560 get a verifier on the item’s rarest word across the corpus (document frequency at most 1%): two thirds use IFEval `keywords:existence`, one

third `keywords:frequency` with a count of 2 or 3. Every fifth verifiable item is genre-free; the rest carry one of six genre triggers (“When the reply is an email, ...”). Claude Sonnet 5.5 rewrites each verifiable item’s body, keeping the original wording; the trigger and the verifier arguments are fixed by code, and a body that lacks the keyword falls back to a template (3 items). Tasks are 60 ManyIFEval prompts that name no genre; the genre is assigned round-robin (ten each) and stated in the user turn with a 900–1,200 word range.

The file of size b holds the first b verifiable items of the N pool in a second seeded order, zero-value items last. Files are therefore nested in b , the file $b = N$ is the whole pool, and $n^*(N)$ is the optimum over these nested files. The grid is $N \in \{200, 800, 3200\}$ and $b \in \{5, 10, 20, 40, 80, 200, 400, 800, 1600, N\}$ with $b \leq N$, 60 tasks, 2 replicates that differ in item order: 24 cells and 2,880 runs per model. Both models see the same files. Every run is scored on every verifiable item of its pool that is relevant to its genre, loaded or not.

A keyword also appears unprompted. Its chance rate is the share of relevant unloaded verifiers that pass, per model, pool and task, over the files of at most 40 items. It is 0.045, 0.024 and 0.016 at $N = 200, 800$ and 3200 for Haiku and within 0.003 of that for Sonnet. Per cell, $R_0(S_b)$ is the mean number of followed relevant loaded items minus the chance rate times their number. A singleton screen loads an item alone on 6 tasks; $q_i(\{i\})$ is the mean over them. It covers the 452 items loaded in some file of size at most 200, once per model (2,712 runs each). The load $\ell(S_b)$ is the sum of the subject’s singleton rates over the relevant loaded items; an unscreened item takes the mean rate of the screened items of its kind and verifier family, which is unbiased because files are seeded prefixes. The mean singleton rate is 0.58 for Haiku and 0.80 for Sonnet. $C(S_b)$ is the file’s tokens. The marginal relevant compliance between consecutive sizes is $MC(b) = [R_0(S_{b'}) - R_0(S_b)] / [\ell(S_{b'}) - \ell(S_b)]$. The value at price ρ_{exp} is $R_{\rho_{\text{exp}}}(S_b) = R_0(S_b) - (\rho_{\text{exp}}/c_{\text{med}})C(S_b)$ with $c_{\text{med}} = 39$ tokens, the median verifiable item, so ρ_{exp} is the normalized cost per median-length item; it is distinct from the theoretical break-even compliance ρ . From one task-cluster bootstrap we take $n^*(N) = \arg \max_b R_{\rho_{\text{exp}}}(S_b)$ and the accretion loss $R_{\rho_{\text{exp}}}(S_{n^*}) - R_{\rho_{\text{exp}}}(S_N)$.

Details. At $b = 3200$ the chance hits are about 14 of the 853 relevant loaded items, most of the raw count of followed items (14.3 for Haiku, 18.1 for Sonnet); Tables 2 and 3 give the raw counts next to the corrected ones. The relevant load at $b = N$ is about $0.27N$ items (53, 213, 853). The singleton rate of a genre-free item is pooled over genres. ManyIFEval has too few prompts of each genre, so the genre is assigned and stated in the user turn. The task’s own instruction list is not used. The median item costs 39 tokens after rewriting, against the corpus median of 15. No run was truncated or empty.

Results. Tables 2 and 3 give the cells and Table 4 the measured optima. At the top step, $b = 1600 \rightarrow 3200$, MC is -0.007 $[-0.013, -0.002]$ for Haiku and -0.037 $[-0.052, -0.023]$ for Sonnet. At $N = 3200$ gross value peaks at $b = 200$, at 5.0 followed instructions for Haiku and 23.8 for Sonnet, and falls to 0.6 $[0.0, 1.3]$ and 4.0 $[0.5, 9.5]$ for the whole pool (Figure 1). Without the chance correction the top-step MC is 0.029 $[0.022, 0.035]$ for Haiku and -0.013 $[-0.028, 0.001]$ for Sonnet. Over the three pools the optimum at $\rho_{\text{exp}} = 0.1$ is 10, 5, 10 for Haiku and 40, 20, 40 for Sonnet: it does not grow with the pool. From $N = 800$ to $N = 3200$ the accretion loss grows from 76.2 to 316.7 for Haiku and from 75.9 to 320.4 for Sonnet, a factor 4.2 for each. Sonnet follows about five times as many instructions as Haiku before capacity binds, and its optimal file is four times as large.

The curves are not smooth, because unranked files differ in the difficulty of their items. For Sonnet at $N = 800$ gross value is 28.4 at $b = 400$ and 3.7 for the whole pool, a step of -0.46 $[-0.56, -0.36]$. Loaded items of another genre are followed at 0.02–0.05. Replies fall inside the

word range in 76% of runs for both models. Figure 3 shows the value curves at $\rho_{\text{exp}} = 0.1$ and Figure 4 the marginal compliance against the load. Figures 5 and 6 show the value curves at the other prices: the optimum moves left as ρ_{exp} rises, and it sits at $b = N$ only for Sonnet at $N = 200$ and $\rho_{\text{exp}} = 0.05$.

Monotone dilution, item by item. Assumption 1 is a statement about each instruction: its compliance does not rise when the file grows. Files are nested in b within a pool, so every relevant item of S_b is in every larger file, and the per-item labels test the assumption directly. For each pool we follow the relevant items of the $b = 40$ file, 12 to 16 per task, through the larger files (Figures 7 and 8). At $N = 3200$ their compliance falls from 0.27 to 0.05 for Haiku and from 0.91 to 0.05 for Sonnet. At $N = 800$ it falls from 0.29 to 0.04 and from 0.47 to 0.07, and at $N = 200$ from 0.25 to 0.11 and from 0.55 to 0.31. These are raw rates, with the chance rate of 0.02–0.05 as their floor.

Order matters inside large files, which the set-level model does not represent: at $b = 800$ of the largest pool the first fifth of the file is followed about twice as often as the rest by both models (0.05 against 0.02 for Haiku, 0.14 against 0.06 for Sonnet).

Against their singleton rates, the screened items are followed less often in a file than alone in 23 of the 24 cells for Haiku and 22 for Sonnet, by 0.07 to 0.66. In the other three cells, all files of at most 40 items, they are followed as often or more often, by at most 0.14.

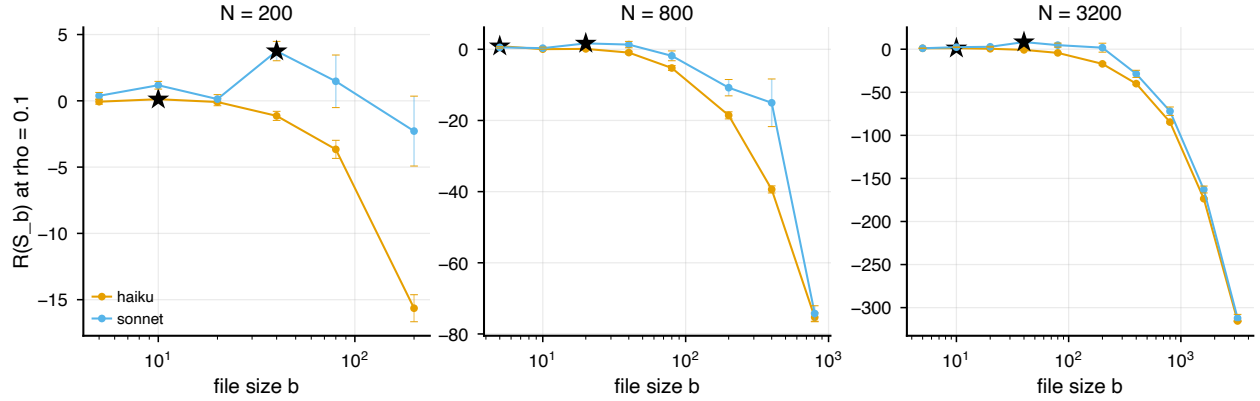


Figure 3: Size sweep: value $R_{\rho_{\text{exp}}}(S_b)$ at price $\rho_{\text{exp}} = 0.1$ against file size b for both models, one panel per pool size N . Points are means over the 60 tasks; bars are ± 1.96 Newey–West standard errors over the task sequence. The stars mark the measured optima $n^*(N)$.

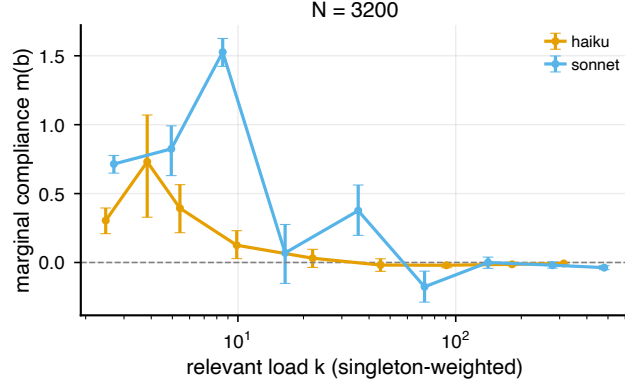


Figure 4: Size sweep: marginal relevant compliance $MC(b)$ against relevant load at $N = 3200$ for both models, with 95% task-cluster bootstrap bars.

Table 2: Size-sweep cells for Claude Haiku 4.5: gross value $R_0(S_b)$ (followed relevant instructions per task, net of chance), the raw count, the singleton-weighted relevant load $\ell(S_b)$, and the marginal compliance MC of the step ending at b with its 95% task-cluster bootstrap interval.

N	b	$R_0(S_b)$	raw	$\ell(S_b)$	MC (step to b)
200	5	0.53	0.60	1.33	
	10	1.19	1.34	1.58	2.67 [1.91, 3.88]
	20	2.05	2.35	3.39	0.48 [0.37, 0.57]
	40	3.18	3.89	8.69	0.21 [0.13, 0.28]
	80	5.11	6.49	14.00	0.36 [0.23, 0.50]
	200	3.84	6.25	25.06	-0.11 [-0.22, -0.01]
800	5	1.37	1.43	2.00	
	10	1.01	1.10	2.53	-0.68 [-1.49, -0.14]
	20	2.16	2.30	3.53	1.16 [0.92, 1.41]
	40	3.33	3.62	7.03	0.33 [0.21, 0.44]
	80	3.65	4.18	13.53	0.05 [-0.03, 0.14]
	200	3.41	4.92	32.50	-0.01 [-0.06, 0.05]
	400	4.05	7.19	62.93	0.02 [-0.01, 0.05]
	800	2.65	7.69	98.36	-0.04 [-0.07, -0.01]
3200	5	1.55	1.57	1.56	
	10	2.11	2.17	3.39	0.30 [0.21, 0.40]
	20	2.76	2.88	4.28	0.73 [0.33, 1.07]
	40	3.65	3.88	6.56	0.39 [0.22, 0.56]
	80	4.49	4.95	13.19	0.13 [0.03, 0.23]
	200	5.04	6.12	30.86	0.03 [-0.04, 0.10]
	400	4.50	6.63	59.47	-0.02 [-0.06, 0.03]
	800	3.24	7.51	121.98	-0.02 [-0.04, 0.00]
	1600	1.67	10.16	240.79	-0.01 [-0.02, 0.00]
	3200	0.63	14.33	385.89	-0.01 [-0.01, 0.00]

Table 3: Size-sweep cells for Claude Sonnet 5.5, as in Table 2.

N	b	$R_0(S_b)$	raw	$\ell(S_b)$	MC (step to b)
200	5	0.97	1.04	1.56	
	10	2.25	2.39	3.19	0.78 [0.67, 0.88]
	20	2.27	2.56	5.28	0.01 [−0.09, 0.12]
	40	8.07	8.74	11.28	0.97 [0.83, 1.11]
	80	10.24	11.55	19.00	0.28 [0.05, 0.51]
	200	17.21	19.45	35.22	0.43 [0.25, 0.60]
800	5	1.02	1.08	2.28	
	10	1.29	1.39	2.78	0.53 [−0.09, 1.00]
	20	3.71	3.87	4.42	1.48 [1.31, 1.66]
	40	5.58	5.91	10.61	0.30 [0.13, 0.47]
	80	7.07	7.66	18.42	0.19 [0.02, 0.38]
	200	11.18	12.85	46.44	0.15 [0.07, 0.23]
3200	400	28.40	31.86	93.65	0.36 [0.26, 0.48]
	800	3.72	9.27	147.59	−0.46 [−0.56, −0.36]
	5	1.60	1.62	1.50	
	10	3.31	3.38	3.89	0.71 [0.65, 0.78]
	20	5.05	5.17	6.00	0.82 [0.63, 0.99]
	40	12.72	12.95	11.03	1.53 [1.42, 1.63]
	80	13.48	13.96	21.92	0.07 [−0.15, 0.28]
	200	23.82	24.94	49.42	0.38 [0.20, 0.56]
	400	15.88	18.07	94.43	−0.18 [−0.29, −0.06]
	800	15.90	20.28	186.93	0.00 [−0.04, 0.04]
	1600	12.32	21.05	369.21	−0.02 [−0.04, 0.00]
	3200	4.01	18.09	591.98	−0.04 [−0.05, −0.02]

 Table 4: Size sweep: measured optimum $n^*(N)$ and accretion loss $R_{\rho_{\text{exp}}}(S_{n^*}) - R_{\rho_{\text{exp}}}(S_N)$ with 95% task-cluster bootstrap interval, per model, pool size and price ρ_{exp} .

	N	$\rho_{\text{exp}} = 0.05$	$\rho_{\text{exp}} = 0.1$	$\rho_{\text{exp}} = 0.2$	$\rho_{\text{exp}} = 0.4$
Haiku	200	40; 6.9 [5.9, 8.0]	10; 15.8 [14.6, 16.8]	5; 34.5 [33.3, 35.5]	5; 72.3 [71.1, 73.3]
	800	40; 37.5 [36.4, 38.7]	5; 76.2 [74.9, 77.3]	5; 153.7 [152.4, 154.8]	5; 308.7 [307.4, 309.8]
	3200	20; 159.1 [158.4, 159.9]	10; 316.7 [316.0, 317.4]	5; 632.2 [631.5, 632.9]	5; 1263.5 [1262.8, 1264.1]
Sonnet	200	200; 0.0 [0.0, 1.3]	40; 6.0 [3.7, 8.3]	10; 21.9 [19.3, 24.5]	5; 59.3 [56.7, 62.0]
	800	400; 41.9 [37.9, 47.6]	20; 75.9 [73.1, 78.4]	5; 152.3 [149.5, 154.6]	5; 307.3 [304.4, 309.6]
	3200	200; 166.9 [161.3, 173.0]	40; 320.4 [315.1, 324.2]	40; 632.1 [626.7, 635.9]	5; 1260.2 [1254.7, 1263.7]

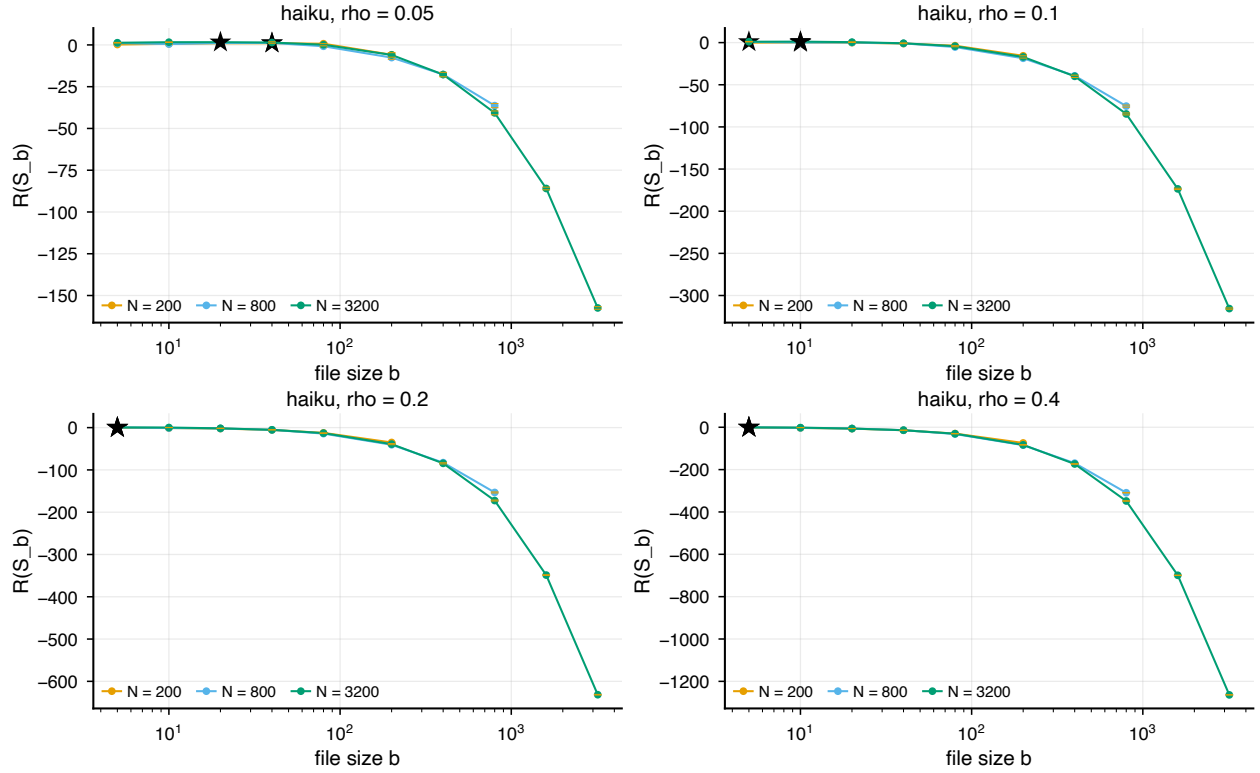


Figure 5: Size sweep, Claude Haiku 4.5: value $R_{\rho_{\text{exp}}}(S_b)$ against file size at each price ρ_{exp} , one series per pool size N , with Newey–West error bars and stars at $n^*(N)$ as in Figure 3.

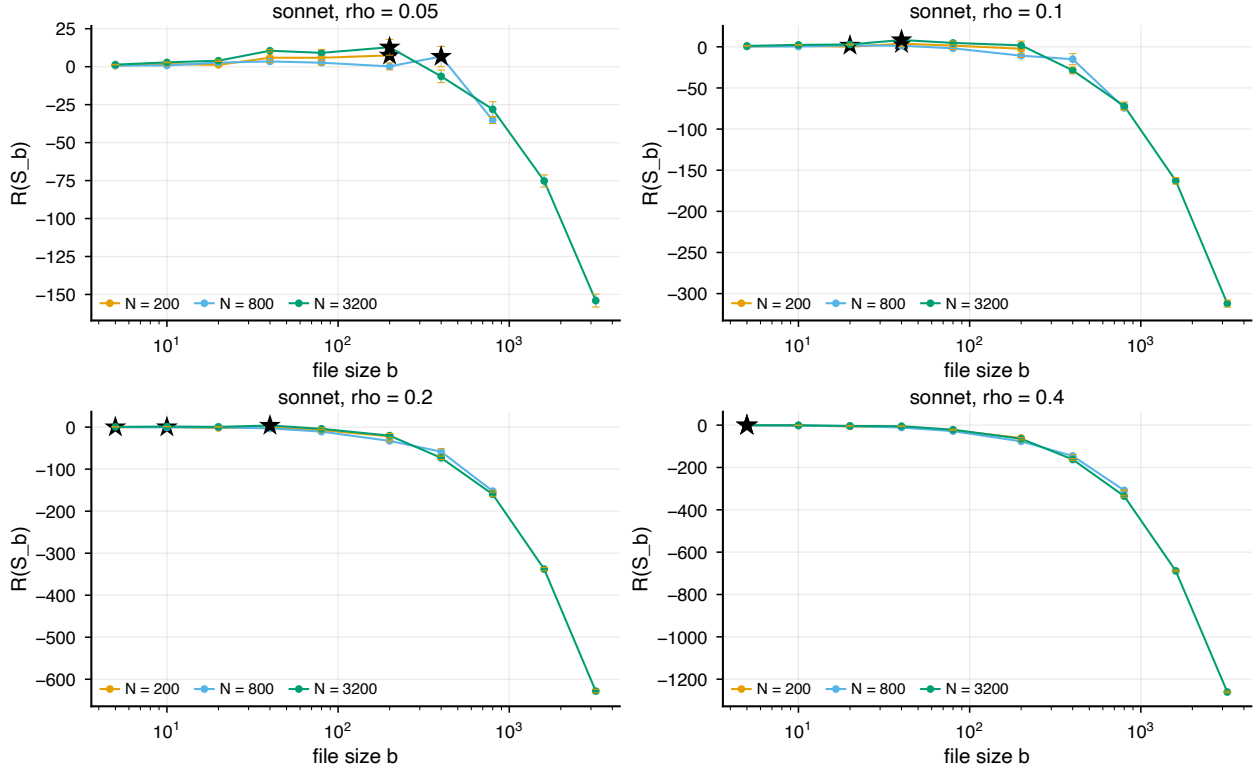


Figure 6: Size sweep, Claude Sonnet 5.5: value $R_{\rho_{\text{exp}}}(S_b)$ against file size at each price ρ_{exp} , as in Figure 5.

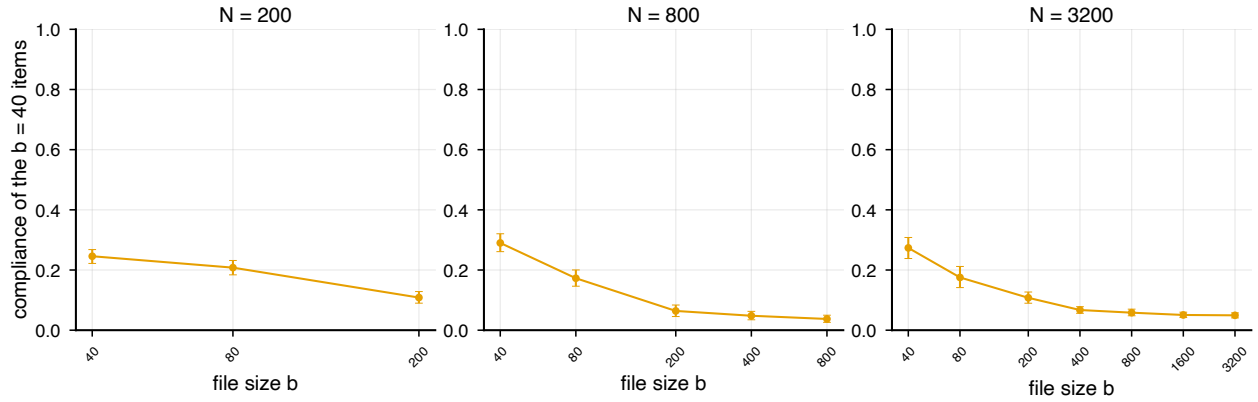


Figure 7: Monotone dilution, item by item, Claude Haiku 4.5: compliance of the relevant items of the $b=40$ file in that file and in every larger file of the same pool, with 95% task-cluster bootstrap bars.

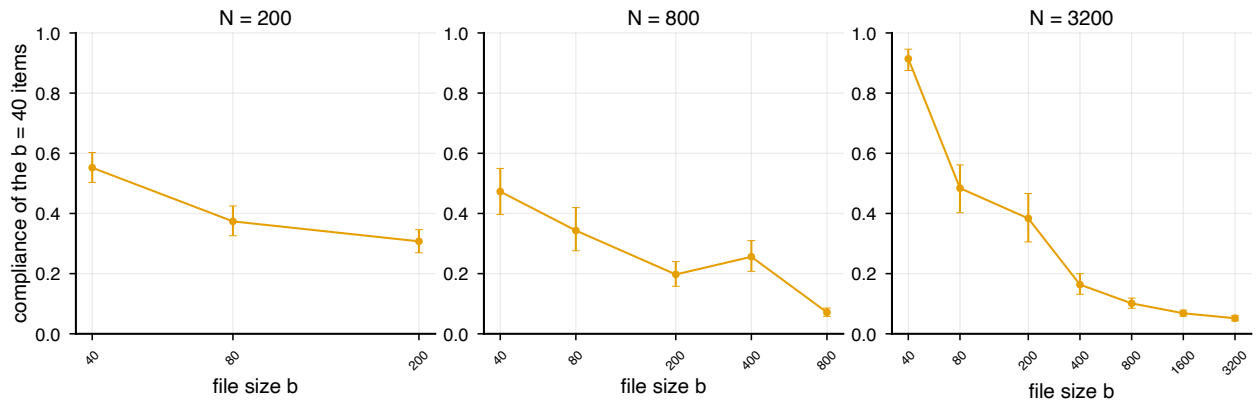


Figure 8: Monotone dilution, item by item, Claude Sonnet 5.5, as in Figure 7.

D.2 Dilution

Purpose. Assumption 1 says that compliance with one instruction does not increase as the file grows. The dilution experiment measures the compliance curve $q(n)$ of 24 model configurations on a published instruction-density benchmark, with no model calls of our own.

Data. The IFScale generations (Jaroslawicz et al., 2025): 5,993 reports from 24 model configurations with $n \in \{10, \dots, 500\}$ keyword instructions and 5 seeds. The repository has no license and the paper is CC BY-NC-SA 4.0, so the raw generations stay local and only aggregates are reported.

Design. Every generation is re-scored with our labels; 52 generations that paste the keyword list are dropped. Per configuration we fit $M(n) = n q(n)$, the number of followed instructions, and report $M(500)/500$, the compliance at $n = 500$. The fitted capacity K of the corrected curve fit sets the class: weak $K < 100$, mid 100–350, strong $K > 350$.

Preregistered call. Descriptive only: $M(500)/500$ is expected between 0.07 and 0.69 in all 24 configurations, with the late/early slope ratio of $M(n)$ below 0.1 in the four weak configurations and the hyperbola $g(x) = 1/(1+x)$ the AIC-best form only in the weak class.

Deviations. The fitted K places 6 configurations in the weak class, 5 in the mid class and 13 in the strong class; the plan projected 4, 8 and 12. The slope ratio is below 0.1 in 2 of the 6 weak configurations, and the hyperbola is AIC-best in 4 of the 6 weak configurations and in none of the others. One configuration, `llama-4-scout`, falls just below the planned 0.07 floor at 0.067.

Results. Compliance at $n = 10$ is 0.92–1.00 in every configuration. $M(500)/500$ lies in $[0.067, 0.689]$ and is below 1 in 24 of 24 configurations (Table 5): no configuration follows its instructions in proportion to their number at $n = 500$, and the strongest configuration follows fewer than 0.7 of them. The curve $q(n)$ is nonincreasing for every class (Figure 9). The hyperbola of the logit case fits only the weak configurations; the monotone dilution Assumption 1 needs no particular curve and holds throughout.

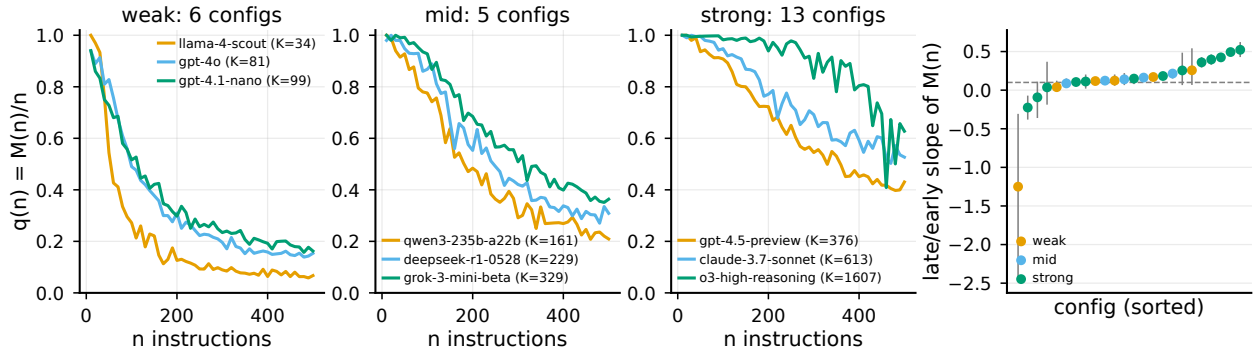


Figure 9: Dilution: $q(n) = M(n)/n$ against n for the lowest-, median- and highest- K configuration of each class (three left panels), and the 24 late/early slope ratios sorted, with 95% seed-bootstrap bars and the 0.1 reference line (right).

Table 5: Dilution: fitted capacity K , class, AIC-best curve form and compliance $M(500)/500$ at $n = 500$, per configuration.

configuration	K	class	best form	$M(500)/500$
llama-4-scout	34	weak	power	0.067
claude-3.5-haiku	52	weak	hyperbola	0.085
gpt-4o-mini	61	weak	hyperbola	0.104
gpt-4o	81	weak	hyperbola	0.154
llama-4-maverick	87	weak	power	0.096
gpt-4.1-nano	99	weak	hyperbola	0.162
qwen3-235b-a22b	161	mid	exponential	0.209
gpt-4.1-mini	202	mid	exponential	0.272
deepseek-r1-0528	229	mid	exponential	0.309
gemini-2.5-flash-preview	281	mid	exponential	0.342
grok-3-mini-beta	329	mid	exponential	0.364
gpt-4.5-preview	376	strong	exponential	0.430
claude-opus-4	428	strong	linear	0.446
claude-sonnet-4	504	strong	linear	0.429
gpt-4.1	526	strong	linear	0.489
claude-opus-4 (high reasoning)	602	strong	exponential	0.521
o4-mini	611	strong	logistic	0.344
claude-3.7-sonnet	613	strong	linear	0.527
claude-sonnet-4 (high reasoning)	648	strong	logistic	0.399
o3	845	strong	logistic	0.516
grok-3-beta	1012	strong	linear	0.619
gemini-2.5-pro-preview	1274	strong	linear	0.666
gemini-2.5-pro-preview (high reasoning)	1333	strong	linear	0.689
o3 (high reasoning)	1607	strong	logistic	0.628

D.3 Regret

Purpose. Theorem 2 bounds the edit-cap term of ETC- m from above and Theorem 3 bounds every policy from below, both of order $(N/m)L$. The regret experiment runs ETC- m on the instances of Theorem 3 and checks both bounds and the order. It is a verification of the theorems, not an empirical test; no dataset has an edit cap. No data and no model calls.

Design. The instances are those of Theorem 3: N identical items, $a = 0.02$, $\theta g(a) = 0.9$, $p = 0.3$, $c = 1$, $\kappa = 0.004$, $B = 10$, $r_1 = 1$, noiseless gains, with $g(x) = 1/(1+x)$ of Appendix B; $S^* = \{1\}$ and every cell is exact. Grid: $N \in \{16, 32, 64, 128, 256\}$, $m \in \{2, 4, 8, 16\}$, $L \in \{10, 20, 40\}$, 60 cells. ETC- m is Algorithm 1 with the estimator of Appendix C.5, under tied loads $\theta_i = \lambda a_i$. Regret is exact wherever the schedule fixes the file, so the only random element is the committed file. The cap term is the regret of the first epoch of every exploration batch plus the transition; the learning term is the later exploration epochs plus the commit phase. $n_0 = \lceil 8 \ln(6NT)/p_{\min} \rceil$ and $T = L(2J + \lceil (s + n_B)/m \rceil) + 4n_0J$ are solved as a fixed point. The lower bound is $(\Delta L/2)(n_m - 1)$ with $n_m = \lfloor N/m \rfloor$ and $\Delta = R(S^*)$.

Preregistered call. The ratio of the cap term to the lower bound lies in $[1, 4(\Delta_{\max}/\Delta)(1 + m(1 + n_B/s)/(2N))]$ in every cell with $N \geq 4m$, and the log-log slope of the cap term on N/m is 1.0 ± 0.1 at each L . A ratio below 1 refutes the bound or the simulator; a slope outside the band refutes the order N/m .

Deviations. None. Gains are observed without noise and $\eta = 1$; the pool has no spurious candidates, so ε is unused.

Results. Both calls pass. Over the 51 cells with $N \geq 4m$ the ratio to the lower bound is 4.08–5.38, inside the interval in every cell (Table 6); the cap term is under the upper bound in all 60 cells, and ETC- m committed to S^* in every cell. The log-log slope of the cap term on N/m is 1.058 $[1.029, 1.088]$ with $R^2 = 0.997$ at each L , over a span of a factor 192 in the cap term (Figure 2). The ratio does not depend on L and falls toward 4 as N grows, the constant of Theorem 2.

The three parameters move the cap term as the bounds predict. More candidates increase the cost, because the policy must test more instructions. A larger edit cap decreases the cost. Longer epochs increase the time that the policy must keep each test batch.

Table 6: Regret experiment: cap-term regret of ETC- m over the lower bound $(\Delta L/2)(n_m - 1)$, with the plan’s upper limit $4(\Delta_{\max}/\Delta)(1 + m(1 + n_B/s)/(2N))$ in parentheses. The ratio does not depend on L . Cells with $N < 4m$ are outside the call.

N	$m = 2$	$m = 4$	$m = 8$	$m = 16$
16	4.35 (7.77)	4.83 (8.05)	$N < 4m$	$N < 4m$
32	4.20 (6.18)	4.42 (6.33)	5.02 (6.61)	$N < 4m$
64	4.13 (5.39)	4.26 (5.46)	4.57 (5.61)	5.38 (5.90)
128	4.09 (5.00)	4.19 (5.03)	4.40 (5.10)	4.87 (5.25)
256	4.08 (4.80)	4.15 (4.82)	4.32 (4.85)	4.66 (4.93)