Measuring in-context algorithmic reasoning in language models against an exact Bayes-optimal standard

Hector Zenil[†,*,1,2] and Luan Ozelim[†,1]

[1] Oxford Immune Algorithmics, Oxford University Innovation & London Institute for Healthcare Engineering, U.K

[2] Department of Biomedical Computing, School of Biomedical Engineering and Imaging Sciences & King's Institute for AI, King's College London, U.K

arXiv:2608.01575v1 [cs.LG] 3 Aug 2026

Abstract

Whether large language models perform genuine algorithmic reasoning or mere pattern completion is hard to test, because most benchmarks lack a ground truth for correct inductive inference. We introduce F-ICL, an in-context-learning benchmark that supplies one exactly. Using the Turing-complete machine F, complement-symmetrised into sF to remove output-polarity bias, we exhaustively enumerate all 1.5 billion programs of length $L \leq 13$ and compute the Bayes-optimal posterior in closed form under a bounded universal (Levin–Solomonoff) prior; models are scored by how closely their served distributions approach it at matched evidence. Each task is paired with its bitwise complement, on which the optimum scores identically, so an original-twin gap isolates the model's inductive bias. Across 105 serving configurations spanning 37 open models (0.8B–675B) and frontier systems from four laboratories, models answer up to 92% of queries correctly, yet 45 of 46 models yield distributions farther from the optimum than a keystroke reference, and their behaviour is bracketed by low-order prefix statistics fitted only on visible evidence. That reference is itself an algorithmic mixture, induced by a print-only machine with no loops, so the panel's implied measure sits closer to a loop-free mixture than to the loop-bearing optimum, independently of the reference machine. Updating is also non-monotone, which no prior explains: a Bayes-rational solved set can only grow in this realisable, noiseless setting, yet added examples produce 6,545 solved-to-unsolved transitions against 13,702 gains. The gap is not predicted by accuracy (Spearman $\rho = -0.19$, $p = 0.21$), does not close with scale or across frontier generations in the serving modes that expose distributions, and is widened by instruction and reasoning post-training. F-ICL is released as an open, reproducible benchmark and toolkit.

1 Introduction

In-context learning, the ability of a sequence model to infer a task from a few input–output examples placed in its prompt and then generalise to a new query, is among the most consequential capabilities of modern language models[6]. It is also among the hardest to interpret. When a model answers a few-shot prompt correctly, it is rarely clear whether it has inferred the underlying rule or has retrieved a surface pattern that happens to fit. The difficulty is not only empirical but definitional:

*Corresponding author: hector.zenil@kcl.ac.uk

†These authors contributed equally to this work.

on most benchmarks there is no agreed notion of what the *ideal* inference from the given examples would even be, so a model's behaviour can only be compared against other models or against human answers, never against the correct posterior.

A principled answer to "what should be inferred from these examples?" exists in algorithmic information theory. Solomonoff's theory of inductive inference defines a universal prior over computable hypotheses and a corresponding Bayes-optimal predictor; under mild conditions no other predictor achieves lower expected loss[7, 16, 9, 10]. The obstacle to using this standard as a yardstick is that it is uncomputable in general and expensive even when bounded. Exhaustive enumeration of small machines has long been used to estimate algorithmic complexity numerically[1, 2, 18]; what has been missing is its use as an *exact inferential standard* for learning systems. As a result, the rich recent literature connecting ICL to Bayesian inference[11, 12, 13, 22, 23] and language modelling to compression[14, 27, 28] has had to rely on tractable but non-universal hypothesis classes (e.g. linear functions or parametric mixtures), trading the exactness of the optimum for analytic convenience. Benchmarks of algorithmic reasoning face the same gap from the other side: suites such as ARC[15] and CLRS[30, 29] probe rule induction and algorithm execution but score against the single intended answer, not against what the evidence licenses; SuperARC[20] instead scores frontier models on human-agnostic, algorithmic-information terms, but along an orthogonal axis to ours, grading compression-based model abstraction and recursive prediction *ability* against approximate complexity estimates rather than grading a served *distribution* against an exactly enumerated posterior; meta-learners trained toward Solomonoff induction[31] sample a closely related Brainfuck-family machine to build a training stream, but sampling leaves the target itself out of reach, so models there are compared against a loose upper bound on the log-loss rather than against the posterior (we return to this in the Discussion); and mechanistic accounts of ICL[24, 25, 26] describe *how* models update in context, but not how close that update is to the normative posterior.

Here we make the Bayes-optimal standard explicit and exact at a small but genuinely algorithmic scale. We fix a minimal Turing-complete reference machine, a five-instruction tape automaton we call F, and *exhaustively enumerate* every program up to a length budget, weighting each by a bounded universal prior. Because the enumeration is exhaustive, the induced map from program behaviour to prior mass is a *sufficient statistic* for Bayesian inference, and the posterior predictive for any few-shot context can be computed in closed form. This yields, for each task, the provably optimal inductive prediction under the F-prior, a ground-truth target with no tuned parameters (Fig. 1). An LLM's in-context predictive distribution can then be compared directly against this optimum, and the speed of its approach as examples accumulate is a clean, information-theoretic measure of algorithmic reasoning: a Bayes-rational learner drives its divergence from the optimum to zero, whereas an evidence-ignoring pattern-matcher plateaus above it.

Turning this idea into a usable benchmark requires solving two problems. First, the enumeration and the optimum must be *certified*: a ground-truth standard is only as trustworthy as the machine that defines it. We address this with an accelerated GPU enumeration that scales to more than a billion programs at the longest length, an independent pure-Python reference implementation, and a chain of exact cross-checks: byte-for-byte agreement of per-output program counts, re-execution of every stored witness program, and bit-identical agreement between the served few-shot posterior and the exhaustive joint. Second, the comparison must be *discriminative and fair*: it should reward genuine inference, expose pattern completion, and summarise a model's behaviour in a way that is robust to task difficulty. We instantiate three task families with exact optima; models are scored on two (induction and continuation) by a bounded, symmetric divergence, and the third (abduction) validates the design. Models are summarised by a single difficulty-aware score alongside an interpretable facet profile.

We describe the benchmark, establish its validity, and use it to evaluate a panel of open-weight

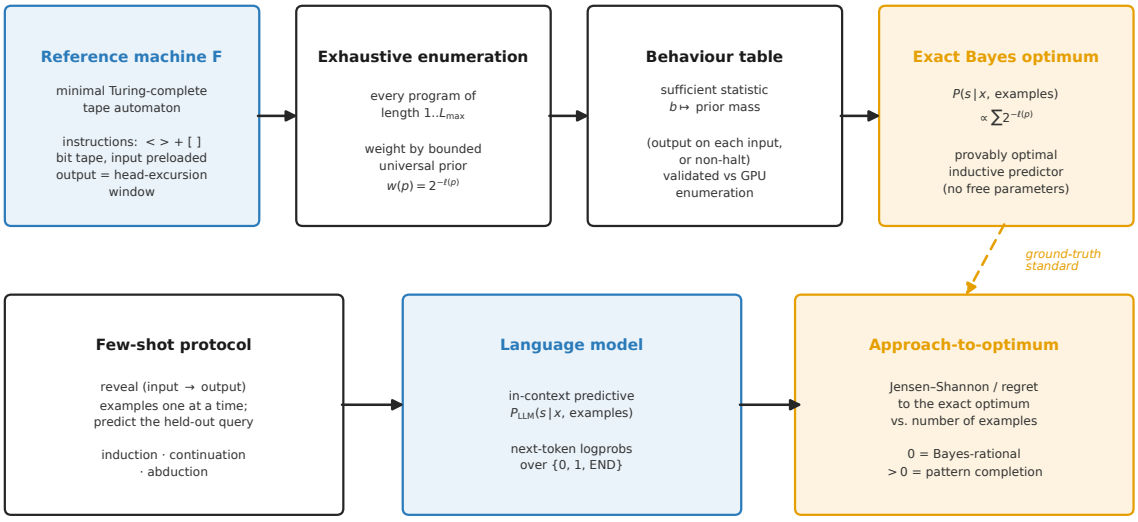

Figure 1: **Measuring in-context learning against an exact Bayes-optimal standard.** Top: the reference machine F is enumerated exhaustively up to a length budget; weighting each program by the bounded universal prior $w(p) = 2^{-\ell(p)}$ gives a behaviour-to-mass table, a sufficient statistic from which the exact Bayes-optimal posterior follows in closed form (both defined in §2.1). An independent reference implementation cross-checks the accelerated enumeration. Bottom: in the few-shot protocol, examples are revealed one at a time and a language model's predictive distribution is compared against the optimum; the divergence to the optimum as a function of the number of examples is the measure of algorithmic reasoning (0 = matches the optimum exactly; larger values indicate greater departure from it).

and frontier language models. F-ICL separates Bayes-rational inference from strong heuristic foils, and reveals that current open models track the optimal trend but fall well short of Bayes efficiency: they solve the easiest tasks while needing far more examples than necessary on harder ones, and produce token-level distributions that are confidently mismatched with the algorithmic prior. The benchmark is exact by construction, inexpensive to run once the optima are frozen, and model-agnostic.

2 Results

2.1 An exact standard for in-context inference

This subsection defines the standard the model results are read against; its certification is summarised in §2.2 and the full construction is given in Methods. F is a minimal Turing-complete tape automaton in the Brainfuck family [5]: five instructions acting on a bit tape preloaded with a binary input (Table 1). It is small enough to enumerate exhaustively up to a length and runtime budget, yet expressive enough that its short programs realise counting, arithmetic and bounded recursion. Weighting each program by a bounded universal (Levin–Solomonoff) prior makes "the ideal inference from these examples" a computable object: for any few-shot context the Bayes-optimal posterior predictive under that prior is computed *exactly*, in closed form, with nothing tuned to the models (Eq. (1)). That optimum is relative to a fixed set of conventions, stated once here and tabulated in Table 2: the symmetrised machine sF, programs of length $L \leq 13$, a 1024-step and 128-cell run budget, normalisation over halting outputs, and a per-family weighting that is length-only for induction and Levin-style (length plus an Elias-δ halt-time code) for the accumulating-evidence families.

Enumerating every program on a fixed ordered set of inputs and recording each program's *behaviour* gives $w(b)$, the total prior mass of each distinct behaviour under the length prior $w(p) = 2^{-\ell(|p|)}$ (Eq. (4), Methods), a sufficient statistic for Bayesian inference over the F-prior restricted to those inputs. Given examples $\mathcal{E} = \{(x_i, o_i)\}$, a behaviour is *consistent* if $b(x_i) = o_i$ for all i, and the exact posterior predictive for a query input x_q is

$$P(s \mid x_q, \mathcal{E}) = \frac{\sum_{b\,\text{consistent},\, b(x_q)=s} w(b)}{\sum_{b\,\text{consistent}} w(b)}. \tag{1}$$

When tasks are drawn from the bounded F-prior, no predictor attains lower expected log-loss than Eq. (1). The evaluated sample is *curated* from that prior (quotas over output length and complexity, plus the learnability gates of Methods), so Eq. (1) is Bayes-optimal for the prior and not for the curated evaluation law. §2.5 measures the consequences of that gap directly, and reports the headline under corrections for it.

F's all-zero tape makes zeros cheap, so the raw prior prefers zero-heavy outputs (first-bit law $P(\texttt{1} \mid \varepsilon) = 0.278$; mean conditional complement asymmetry 0.24 bits Jensen–Shannon), a confound a frozen benchmark would reward. To keep the standard free of that bias, all optima are served from the complement-symmetrised machine sF, and every task ships with its bitwise-complemented *twin*, on which the optimum provably scores identically (Methods 4.3). The enumeration, the stored optima and the serving path (an exhaustive small-program joint over programs $L \leq 6$, and a full-budget $L \leq 13$ program index; two backends that cross-check each other) are certified by exact, independent checks (§2.2; Methods).

Models are scored on two task families. In *induction*, a hidden program is shown through input→output examples and the model predicts its output on a held-out query; the target is the

Table 1: **The reference machine F.** Five instructions over a bit tape; `[ ... ]` is a while-the-current-bit-is-1 loop. The only validity test is bracket matching. Input is preloaded on the tape; output is the bit string on the head-excursion window at halt.

Symbol	Name	Effect
`<`	left	move head one cell left (abort if off-tape)
`>`	right	move head one cell right (abort if off-tape)
`+`	flip	flip the bit under the head
`[`	loop-begin	if current bit = 0, jump past the matching `]`
`]`	loop-end	if current bit = 1, jump back past the matching `[`

exact posterior over outcomes. In *continuation*, a program's output is revealed one token at a time and the model gives its next-token distribution over $\{0, 1, \text{END}\}$; the target is the exact conditional at every position. A third family, *abduction*, validates the design (worked example in Appendix K); §2.3. The headline F-ICL-A score anchors the Jensen–Shannon divergence between model and optimum to three exactly computable points (Eq. (20), Methods): the worst possible predictor scores 0, the keystroke reference (a uniform random keystroke over $\{0, 1, \text{END}\}$ at every step) scores exactly $\frac{1}{2}$, and the Bayes optimum scores 1. The keystroke reference is a fixed, computable, model-independent point, but not an uninformed one: it is itself an algorithmic mixture, the one induced by a print-only machine with no loops (§2.5). A score's position relative to $\frac{1}{2}$ therefore locates a model between that loop-free mixture and the loop-bearing optimum, and its distance states how much of the corresponding gap it closes. We call the quantity this score measures a model's *posterior fidelity*: how closely its served predictive distribution matches the exact Bayes-optimal posterior. We avoid the term "calibration", which in the machine-learning literature denotes confidence–accuracy agreement[36]; posterior fidelity is the stronger, distribution-level property of matching a known-correct posterior. Raw Jensen–Shannon geometry compresses all predictors toward the top of its scale (the keystroke reference scores ≈0.85), which is why the headline is the anchored score rather than the raw one; the raw per-family divergences and the keystroke references are reported alongside it and never blended. Two robustness summaries accompany it: a difficulty-balanced score weighting easy and hard tasks equally, and a median *sample-complexity efficiency* (optimal examples-to-solve divided by model examples-to-solve; 1 is Bayes-efficient). Every induction task additionally carries its *Bayes sample complexity* (BSC), the number of examples the optimal predictor itself needs to identify the answer, including zero when the prior alone identifies it; the frozen set is balanced across BSC $\in \{0, 1, 2, 3+\}$. One consequence of scoring against an exact posterior should be stated here: the induction candidate set contracts as evidence eliminates programs, and collapses to a single candidate in 63% of (task, shot) cells overall and 93% by the final shot, where the divergence is identically zero for model, keystroke reference and worst-case predictor alike. The anchors therefore dilute together and the anchored ratio is preserved, but the induction divergence facet carries its signal on the roughly one third of cells where the posterior has not yet collapsed. Reference foils and classical nulls are scored through the identical pipeline, so the region around and below the keystroke reference is itself charted (§2.3; Methods). With these anchors in hand we first establish that distance to the optimum separates Bayes-rational updating from pattern completion, then report the panel on that established axis. The construction of the instrument is given in Methods.

2.2 The standard is certified

A benchmark whose targets are computed by a bespoke accelerator invites the worry that the accelerator, not the model, is being measured. A chain of exact checks rules this out (Methods):

- **Gate 1: count cross-check.** An independent, dependency-free pure-Python F reproduces the GPU enumeration's per-(input, output) program counts *byte-for-byte* ($L = 5$, $M = 8$, all 511 inputs of length $0..M$ for $M = 8$, M denoting the maximum input length of the enumerated input set).
- **Gate 2: witness re-execution.** Sampled stored entries decode their *witness* program and re-execute it with exact CPU semantics, confirming halt time and output: every recorded behaviour is a real, re-runnable program.
- **Gate 3: rule-set identity.** The dumped program index serving the multi-example optima is re-enumerated and checked, per (input, output), for identity of the exact *set* of program identities, not just their count.
- **Serving check.** The posterior served from that index is *bit-identical* to the exhaustive small-program joint across 0–5 examples, including non-halting examples (Appendix N).

A build is accepted only on exact agreement. Two metric-compatible backends compute the optimum and cross-check each other (a small-program exact joint, and the production dumped program index serving the complete $L \leq 13$ budget; Methods), and a pre-release audit recomputed the frozen datasets' stored targets from independent code paths, catching two builder defects invisible to every accuracy-style metric. The build therefore asserts twin-mirror equality, normalisation and the exact first-position law on every dataset it writes.

2.3 The benchmark separates inference from pattern completion

We verify that approaching the optimum is genuinely diagnostic by scoring reference predictors from Bayes-rational to purely superficial (Fig. 2), on a realisable few-shot family drawn from the sF prior ($M = 4$, $L_{\max} = 6$, the same small-program joint that cross-checks the induction optima; 683 distinct behaviours over 199 outcomes). The exact Bayes floor falls from 3.99 bits with no examples to $\approx$0.01 bits by eight (Fig. 2a); the optimal predictor sits at zero regret by construction. The foils never approach it (Fig. 2b): the *marginal* predictor (output frequencies with the inputs ignored, the natural formalisation of pattern completion) still pays $\approx$2.2 bits of excess loss at twelve examples and spikes to a 17.2-bit regret at one example by confidently committing to an output the algorithmic map does not support; the *nearest-input* predictor (surface analogy) stays $\approx$13 bits above the optimum; the *uniform* predictor pays a constant $\log_2 |\mathcal{O}| \approx 7.6$ bits. The separation is large and stable.

At scale, the production backend ($L \leq 13$, $M \leq 8$; 500 length-balanced tasks per family, Methods) instantiates two accumulating-evidence families whose optima sharpen as evidence is revealed token by token (Fig. 3): *continuation* (next-token prediction over $\{0, 1, \text{END}\}$ against the exact Bayesian continuation) and *abduction* (inferring which of eight candidate inputs produced the revealed output; the exact posterior's entropy collapses from 3 toward $\approx$1.3 bits). Their foil separation differs informatively. In continuation, even an input-agnostic "pooled" predictor stays within 0.12 bits (Jensen–Shannon) of the optimum (typically below 0.07) and a uniform next-token guess within $\approx$0.26 bits (Fig. 3a); the family is a sensitive fidelity probe but a weak foil separator. Abduction is far more discriminative (Fig. 3b): a *prior-only* predictor drifts *away* from the sharpening optimum

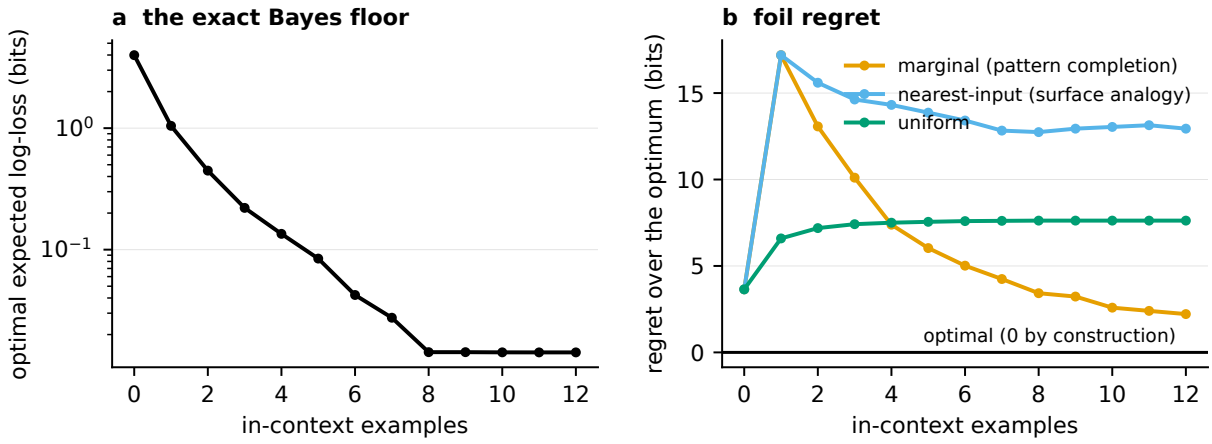

Figure 2: **The benchmark separates Bayes-rational inference from pattern-matching foils** (realisable sF family, $M = 4$, $L_{\max} = 6$, 300 tasks; sF optima under the mass–time weighting of Eq. (7), the same joint that serves the pipeline-scale build induction optima). **a**, The exact Bayes floor (optimal expected log-loss, log scale) falls from 3.99 bits to ≈ 0.01 bits as examples accumulate; no predictor can lie below this curve. **b**, Regret (excess bits over the optimum) versus number of in-context examples. The optimal predictor is at zero by construction; the marginal (output-frequency) foil spikes to 17.2 bits at one example by confidently committing to an unsupported output and still pays 2.2 bits at twelve; the uniform foil's regret climbs to $\log_2 |\mathcal{O}| \approx 7.6$ bits as the floor falls; the nearest-input (surface-analogy) foil remains near 13 bits. None approaches the optimum.

(to ≈0.40 bits) and a *greedy-match* heuristic improves (from 0.72 to ≈0.31 bits) yet never reaches it. Continuation tests distributional fidelity; abduction, Bayes-rational integration of evidence.

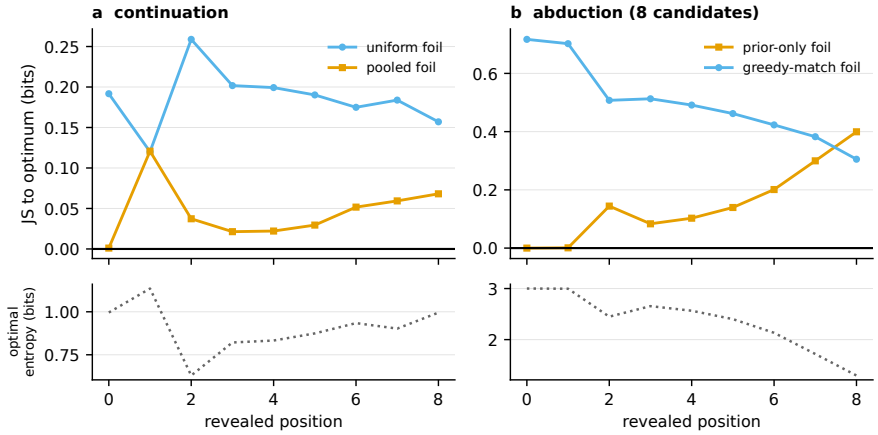

Figure 3: **Accumulating-evidence task families at scale** ($L \leq 13$, $M \leq 8$, 500 length-balanced tasks each under the sF mixture; output revealed token by token). **a**, Continuation: Jensen–Shannon divergence of the uniform and input-agnostic "pooled" foils to the optimal next-token distribution (upper panel); the optimal next-token entropy (lower panel, dotted) is non-monotonic and peaks near 1.1 bits. Continuation is a sensitive fidelity probe but only weakly separates these foils. **b**, Abduction over eight candidate inputs: the optimal posterior entropy (lower panel, dotted) collapses from 3 bits toward ≈1.3 bits as evidence accumulates; the prior-only foil drifts *away* from the optimum (it ignores evidence while the optimum sharpens) and the greedy-match foil improves but never reaches the optimum (upper panel).

In the panel of §2.4, abduction accordingly serves as a *validation* family: models are scored on induction and continuation, whose metrics read directly off token log-probabilities; scoring abduction requires the model's distribution over a candidate *input* set, which current serving interfaces do not expose cleanly (Discussion).

One scope condition should be carried forward. The separation demonstrated here is established on *realisable* families drawn directly from the sF prior, whereas the panel is scored on curated task sets selected for learnability and length balance (Methods), for which the optimum is not the best predictor of the realised sample (§2.5). We take the foil separation as evidence that distance to the optimum is diagnostic of Bayes-rational updating in general, not as a measurement transferred cell by cell to the curated sets; the classical nulls of §2.4, which are scored on the curated sets themselves through the identical pipeline, are what bracket the panel there.

2.4 Evaluating a panel of language models

We evaluated **105** model runs[1]: an API panel of **49** rows spanning NVIDIA NIM (30 open models), OpenAI (a dated generational ladder from GPT-4o May 2024 to GPT-5.6, nine models), Google (Gemini 2.5/3.x through Vertex and the developer API, plus gemma-4 through the developer API), xAI (Grok 4.20 non-reasoning and Grok 4.5 at low reasoning effort) and Anthropic (Claude Fable 5,

[1]Missing calls are dropped from the affected means, never scored. One API model (`mistral-large-3-675b`) was retired by the provider mid-evaluation, its continuation phase permanently unservable (HTTP 410, end-of-life); its row is reported from the completed induction family only. The episode is itself a datum for the reproducibility discussion: benchmark rows measured through commercial APIs can become unrepeatable overnight.

Opus 4.8, and the Fable-with-Opus-fallback stack), and a local panel of **56** `llama.cpp` GGUF builds spanning quantisation ladders, a parameter-scale ladder, and base/instruction/reasoning triads (Methods; serving routes per provider are documented there).

Counts below refer to four nested populations, named here once and used consistently thereafter. The *panel* is all 105 runs. The *log-probability runs* are the 81 of these, over 46 distinct models, whose endpoint exposes next-token distributions; only these carry a fidelity score. The *macro-scored runs* are the 80 of those 81 that completed both scored families, the exception being the retired `mistral-large-3-675b` row, which has induction only and is reported separately wherever a macro score is required. Answer-level statistics (accuracy, agreement, alignment) use the larger set of runs that returned answers, which does not require log-probabilities and is stated at each use. The headline below-anchor count is over the log-probability runs (78 of 81); the robustness re-checks of §2.5, which require a macro score, report the macro-scored count (77 of 80). Continuation analyses quote two cell counts, which differ by bookkeeping granularity rather than by dataset or route: the frozen family contains 1,950 (task, position) cells (one per output bit plus the terminator, over 300 tasks of output length 3–8), but a cell's prompt and optimum depend only on its (input, prefix) pair, and because tasks share inputs (204 distinct inputs across the 300 tasks) and overlapping output prefixes, 1,736 of those cells are distinct measurements; the harness issues each distinct prompt once and reuses the result. Task-level scoring iterates all 1,950 cells, while analyses that operate on distinct measurements (the read-out audit and the tokenisation check) quote 1,736. The model-level statement in the abstract counts distinct models rather than runs.

The panel was scored against the *pipeline-scale build* of the benchmark: the continuation family at the full $L \leq 13$ budget (identical, by construction, to the extended build) and an induction family served from the exact small-program joint (hidden programs $L \leq 6$, mass–time weighting); an extended build, released alongside it and used for the validation figures, scales induction to hidden programs of $L = 5$–9 under the complete $L \leq 13$ mixture with length-only weights served by program-set intersection (Methods, Reproducibility). The two instruments are equivalent on these tasks, and for a principled reason: as program length grows, the bounded mixture thins (ever fewer programs realise new behaviour per unit of prior mass, the same collapse that degrades the correlation between mixture complexity and minimal program length in our companion analysis[19]), so short-program support carries the posterior either way. We verified the equivalence exactly on the panel's own tasks by re-serving all 120 induction tasks from the complete $L \leq 13$ store under the frozen-release conventions using the released robustness tool: across all 1,080 (task, shot) cells the two optima differ by a mean Jensen–Shannon divergence of 0.028 bits with 98% maximum-a-posteriori (MAP) agreement, and the difference decomposes into a weighting-convention component (0.027 bits) and a budget component of only 0.002 bits, confirming that programs beyond $L = 6$ contribute negligible mass on these tasks. Models are tested equivalently: rescoring every stored model distribution against the swapped optima on identical candidate sets reproduces the per-run F-ICL-A(induction) with rank correlation 0.975 (mean absolute change 0.015, $n = 81$ runs), and the headline count is unchanged, 77 of 80 macro-scored runs below the keystroke reference under either instrument (one of the 81 log-probability runs, the retired mistral row, has no continuation leg and so no macro score). What this equivalence does not cover is the extended build's harder task regime (hidden programs of length 7–9, beyond the pipeline-scale build's $L \leq 6$), on which no model has yet been scored; that regime is out of scope here. Because the optimum is fixed, every run is measured against the same target, and the same model served two ways can be compared directly. The pipeline's missing-data guards proved load-bearing here: an initial local serving of the gemma-4 family returned (near-)uniform log-probability read-outs (divergence equal to the keystroke reference to within 2×10^{-4}), which the degeneracy audit flagged as a serving artefact (a chat-template/token-mapping fault in the serving stack) rather than a result. After the template

fix the family was re-evaluated cleanly and re-entered all analyses automatically; its true continuation divergence places it among the families *farthest* from the optimal posterior in the panel, where the artefactual read-out would have placed one gemma variant at the keystroke reference. The audit has an answer-side twin: six API rows returned empty replies on the majority of answer calls (53–99.9%; broken template or truncation at the serving layer, not models scoring zero), and their accuracy facets are excluded with rates disclosed under the released answer audit. Where a provider delivers no answer, we report no answer. A third audit closes the remaining gap on the log-probability side. The uniform test above detects read-outs that collapse toward the anchor, but not the opposite failure, in which the served distribution collapses onto a single class because the others never appeared in the endpoint's top-k and were filled by the smoothing floor. Its signature is exact: two of the three classes carrying bitwise identical probability, which cannot happen when a model genuinely resolves them. Four locally served rows failed this test on a majority of cells under the released read-out audit; two of them, both base checkpoints, failed on 100% of cells and consequently scored *numerically identically* to one another despite different architectures, sizes and tokenisers: the instrument was reading its own floor back rather than resolving the model. The cause is a harvesting limit rather than a property of the models, so it is repairable: we re-served nineteen local rows (the four failures and every row sharing their serving configuration) requesting a larger top-k, after which all nineteen resolve all three classes on all 1,736 cells and the two base checkpoints separate. The repair moves fourteen of the nineteen rows by less than 0.01 bits; the largest move is Qwen3-30B-A3B-Instruct, whose continuation divergence rises by 0.096 bits and whose F-ICL-A falls from 0.343 to 0.308, which was 94% tie-degenerate and is the middle rung of one of the two post-training triads, so the repair *enlarges* that triad's base-to-instruct drop rather than creating it. The repaired read-outs are what the reported panel uses; the originals are retained in the release as the evidence for this audit. Two caveats follow from repairing rather than discarding. The re-serve was scoped to the affected serving configuration, so nineteen of the fifty-six local rows were harvested at the larger top-k and thirty-seven at the original depth; the base-to-instruct comparisons above are configuration-matched, but the third triad rung and part of the gpt-oss quantisation ladder are not, and harvesting depth is recorded per row in the registry. And the threshold is a threshold, not a guarantee: after the repair every local row resolves three classes on every cell, while every API row still shows some epsilon ties, fourteen of the twenty-four on 5% or more of cells and none above 17.2%, which we cannot repair because the providers cap the harvest. That residue flatters the affected rows slightly (across API rows the tie fraction is negatively rank-correlated with divergence), so it is conservative for the below-anchor headline but not for orderings: within the OpenAI ladder the oldest and most faithful row carries 10.9% ties against 0.3% for the next, and the ladder claim should be read with that asymmetry in mind. Two lessons generalise: a degeneracy guard written for one failure mode (near-uniform) is blind to its mirror image (near-point-mass), and two rows scoring identically to six significant figures is a diagnostic worth testing for automatically.

The measurement, made concrete. The benchmark asks how fast a model's in-context inference approaches the Bayes-optimal predictor as examples accumulate (Fig. 4). The optimum locks onto the truth quickly: its induction accuracy is 0.98 by four examples and exact by five. The panel, in contrast, rises slowly and plateaus far short (mean accuracy 0.62 at eight examples; Fig. 4a), leaving a large and persistent shortfall. The posterior-divergence view is even more telling (Fig. 4b): a *single* example moves the panel mean *away* from the optimum, as models over-commit to one datum before slowly recovering, exactly the failure the pattern-matching foils exhibit (§2.3). This one-example over-commitment is nearly universal: 69 of the 81 runs with

usable log-probabilities, frontier rows included, have a larger divergence after one example than after none (sign test $p < 10^{-9}$). The shortfall is *not*, however, an over-commitment to simplicity: panel agreement with the Bayes-MAP answer tracks truth-accuracy to within ± 0.02 at every shot, so models favour neither the Occam answer nor the truth; they are simply far from both. The remaining paragraphs quantify where this gap sits and what does (and does not) move it.

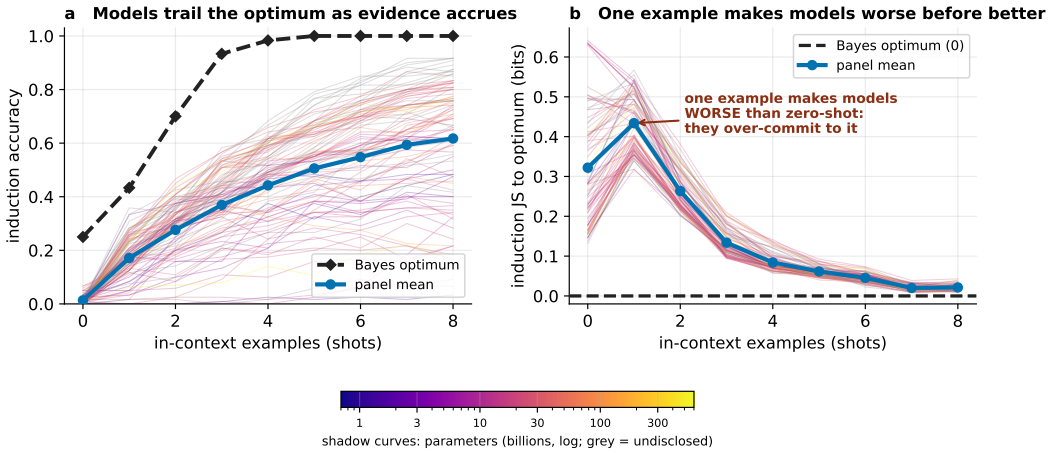

Figure 4: **Approach to the Bayes-optimal predictor.** Faint lines are individual models, coloured by parameter count (log scale, as in Fig. 5; grey = undisclosed API size); the absence of a colour gradient previews the scale-invariance result. The bold line is the panel mean. **a**, Induction accuracy versus in-context examples: the optimum (black) reaches 1.0 by five examples while the panel plateaus near 0.62. **b**, Posterior Jensen–Shannon divergence to the optimum: one example *increases* the mean divergence (models over-commit to a single datum) before it slowly declines, never matching the optimum's zero.

Models track accuracy but are confidently far from the posterior. The headline is a dissociation the exact optimum makes visible and accuracy alone conceals (Fig. 5). Of the **81** runs of **46** distinct models with usable log-probabilities, 78 runs (45 of the 46 models) score *below* the keystroke reference on the anchored F-ICL-A scale, i.e. distributionally further from the posterior than the keystroke reference on the algorithmic-reasoning axis. The single exception is locally served `gpt-oss-20b`, which sits *at* the keystroke reference (full-precision 0.51, task-bootstrap 95% CI $[0.50, 0.52]$; its best single run, at Q6, reaches 0.53) (Fig. 5a; the full named ranking is Fig. 6). The frontier is no exception: every proprietary full row, from OpenAI's GPT-4o through GPT-5.6 and Google's Gemini 2.5 Pro (0.42), lands below the anchor, and the current flagship `gpt-5.6-sol` pairs near-ceiling answer accuracy (0.86) with an F-ICL-A of 0.33, below the panel median: whatever frontier training improves, it is not fidelity to the algorithmic posterior. Point-prediction accuracy and posterior fidelity are also uncorrelated (base-model Spearman $\rho = -0.19$, $p = 0.21$), so models answering up to 92% of in-context queries correctly still sit in the sub-reference zone, far from the Bayes optimum. The gap is also an efficiency statement: the panel's median sample-complexity efficiency is 0.40, i.e. the typical model consumes about two and a half times the information-theoretically minimal number of examples on the tasks it does solve. The continuation family localises the failure (Fig. 5b): the panel-mean next-token divergence exceeds the keystroke reference (0.19 bits) at *every one* of the nine revealed positions (0.29–0.50 bits). The models are confidently wrong about the algorithmic continuation, not just uncertain about it. This shortfall is difficulty-

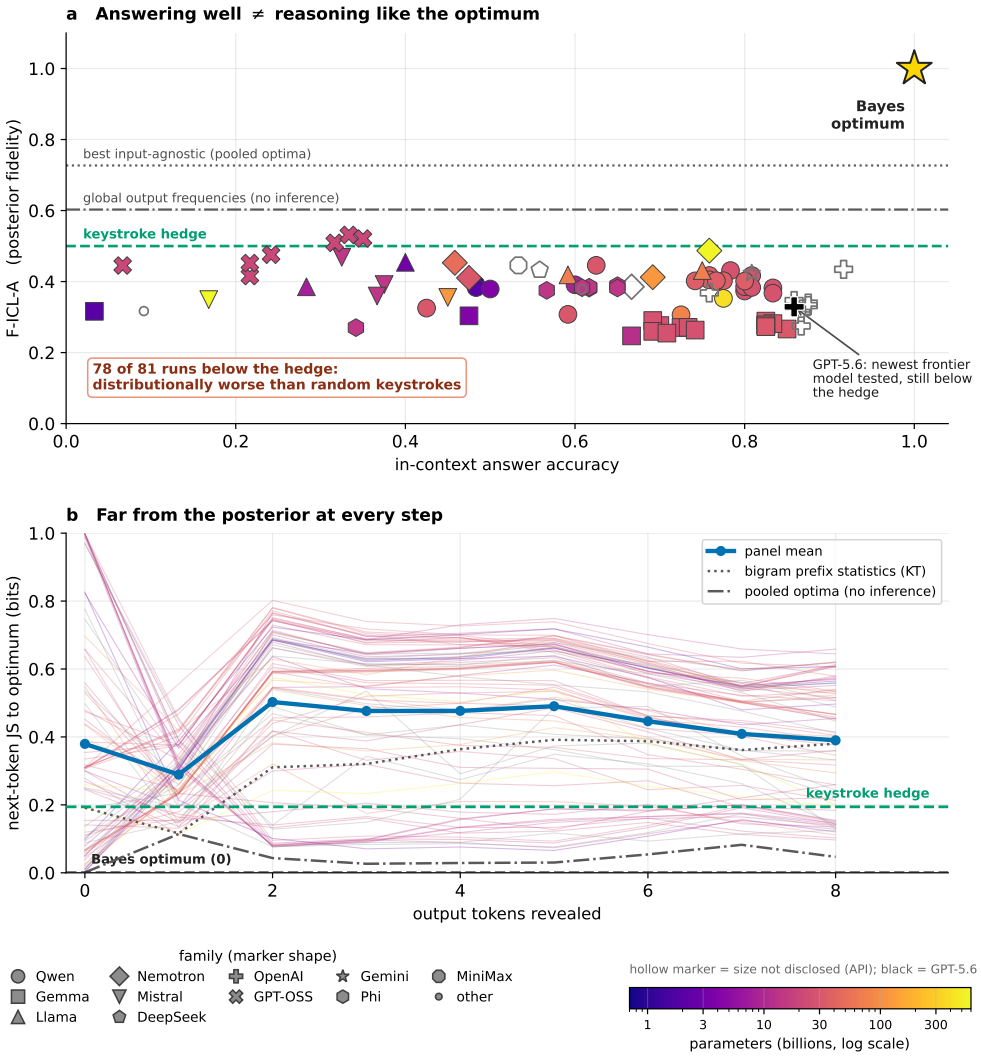

Figure 5: **The algorithmic-reasoning gap. a**, Every model's in-context answer accuracy (final shot) against its anchored F-ICL-A posterior fidelity to the algorithmic optimum; marker shape encodes the model family and colour the parameter count (log scale), with hollow markers for undisclosed API sizes. The dashed line is the keystroke reference and the star the Bayes optimum; grey reference lines are the stronger classical nulls (Methods): the dataset's global output frequencies (0.60) and the pooled per-shot optima (0.73), both above every model in the panel. Models reach up to 92% accuracy yet cluster below the keystroke reference (78 of 81 runs; two-sided sign test $p < 10^{-9}$): answering well does not imply reasoning like the optimum. **b**, Faint lines are individual models (coloured by parameter count as in (a); grey = undisclosed API size) and the bold line the panel mean ($n = 81$ runs) of the continuation Jensen–Shannon divergence to the optimum, which exceeds the keystroke reference (0.19 bits) at every one of the nine revealed positions and never reaches the optimum's zero; the dotted grey curve is the order-1 Krichevsky–Trofimov prefix-statistics null, which tracks the panel band, and the dash-dotted curve the pooled optima.

wide, not a hard-task tail: panel-mean induction accuracy is 0.69 even on tasks the optimum solves from the *prior alone* (BSC=0), drops to 0.53 on tasks the optimum solves from a single example, and stays at 0.53–0.58 across the harder buckets, against the optimum's 1.0 (per-family learning curves and difficulty buckets are provided in the released analysis notebook).

Classical nulls bracket the panel from both sides. The keystroke reference is deliberately the *weakest* evidence-free predictor, so we scored stronger classical nulls through the identical pipeline (Methods; horizontal references in Fig. 5). The dataset's *global output frequencies* (a single distribution ignoring the query, the examples and the position) reach F-ICL-A 0.60, and the best input-agnostic predictor (the per-shot pooled optimum) reaches 0.73: every model in the panel is outscored on posterior fidelity by a lookup table of output frequencies that performs no inference at all. Both of these nulls are constructed from the frozen optima themselves (Methods), so neither is achievable by a predictor that lacks the target; the comparison bounds how much task-independent structure the models fail to recover, and is not the claim that a deployable predictor beats them. From the other side, Krichevsky–Trofimov add-$\frac{1}{2}$ estimators fitted on the visible prefix, the natural "sequence statistics" predictors, land *below* the keystroke reference (0.41 memoryless, 0.45 bigram), and their per-position continuation divergences (0.38 and 0.29 bits) bracket the panel's band (0.29–0.50 bits; Fig. 5b). The panel's distributional behaviour is thus indistinguishable in level from low-order prefix statistics. This bracketing also gives the result its most defensible form, and one that does not depend on the identity of the target. Our anchor is itself an algorithmic-probability mixture, namely the one induced by a *print-only* machine with no loops; the optimum is the mixture induced by a loop-bearing machine. Models sit between the two, at the level of low-order prefix statistics. The claim we are entitled to is therefore about the *shape* of the measure these models bring to such strings: it resembles a loop-free algorithmic mixture or a Markov measure more than a loop-bearing one. That is a structural distinction, and unlike the absolute distance to any one optimum it is not at the mercy of a translation constant of the same order as our entire description budget (Methods).

Error anatomy: what models do when they are wrong. Because every task carries its exact posterior, each wrong answer can be classified rather than only counted (answer-level records cover the 42 complete API runs; empty replies are *non-answers* (refusals or delivery failures, 1,269 calls) and are excluded from the error taxonomy, though they still score as wrong in the accuracy facets). Restricted to task–shot cells whose posterior support is non-degenerate, 31% of wrong answers carry exactly zero posterior mass under the benchmark's truncated, halting-conditioned support: no program in that support both matches the shown examples and produces them, so models violate the *deductive* part of the task, not only the inductive part; 46% of errors do not even have the right output length. The errors' positive structure is the pattern-completion foil's: 41% are verbatim copies of a shown example output, rising to 70% of errors by eight examples, with a recency preference for the most-recent example. Only 2.8% of errors are the Occam (Bayes-MAP) answer, confirming at the item level that the shortfall is not over-simplification. The served distributions show the complementary failure: on the consistent support, models *under*-commit. By eight examples they assign the truth 0.58 where the optimum assigns 0.999, and their entropy after a single example is 1.46 bits against the true posterior's 0.93. Models thus pay the penalty twice: too flat where the posterior is sharp, and confidently wrong off its support.

Termination is where model and target disagree most, and both are miscalibrated. The continuation infidelity concentrates on the *termination* symbol, which carries 89% of the di-

vergence (§2.5). The disagreement is genuine: across the 24 API models with stored next-token distributions (46,800 position-calls) the model's `END` mass is *anti*-correlated with the optimum's (Pearson -0.16, Spearman -0.11). But the realised data show the error is two-sided rather than the models'. Where the optimum is $\geq$50% sure the output ends, models assign termination 0.24 and the output actually ends 0.20 of the time, so the models are the better calibrated of the two; where the optimum is $<$10% sure, the true rate is 0.007 and models assign 0.43, so here the models are badly wrong in the opposite direction. The one unambiguous model failure is structural: at the first output position, where termination is impossible and the optimum's `END` mass is exactly zero, the median model still assigns 0.44 and some assign 0.99, which is consistent with chat-format end-of-sequence leakage rather than with a claim about programs, though we do not test that mechanism directly. The honest summary is that models over-predict termination where it cannot or does not happen, while the target over-predicts it wherever the length quota has removed short outputs from the sample; we therefore report the bit coordinate separately (§2.5) and rest no conclusion on the termination coordinate alone.

In-context learning is non-monotone: models un-solve tasks. In the realisable, noiseless setting a Bayes-rational agent's solved set can only grow with evidence; the optimum's accuracy curve is monotone. The panel's is not: across the 98 complete runs, additional examples produce 6,545 solved→unsolved transitions against 13,702 gains (one forgetting event per two learnings, a median of 66 per run on 120 tasks), and 24% of ever-solved (run, task) pairs end *wrong* at eight examples. This instability is uncorrelated with accuracy ($|\rho| < 0.05$), so it is not an artefact of weak models, and it tracks post-training in both triads (Ministral-3-14B base→instruct→reasoning: $9 \to 39 \to 54$ events; Qwen3-30B-A3B: $35 \to 60 \to 69$), giving the post-training result a purely behavioural, distribution-free counterpart: we read this as alignment tuning destabilising hypothesis maintenance.

Failures are idiosyncratic; difficulty is shared but not Bayesian. Item-level errors correlate only weakly across the panel: mean pairwise Cohen's κ on final-shot correctness is 0.17, and even serving and quantisation replicas of the *same* gpt-oss-20b weights agree only at $\kappa = 0.34$, the item-level face of the serving-reproducibility result below. The union of runs solves every task, and a simple plurality vote over the API panel scores 0.93, above every single run in the panel (the best three are tied at 0.92): the panel's answer knowledge is complementary, yet no aggregation touches the distributional gap. At the same time, *which* tasks are hard is a stable property of models as a class (split-half reliability 0.90), and it is mostly orthogonal to Bayesian difficulty: Bayes sample complexity, program length and prior mass of the truth jointly explain only 13% of the variance, while surface copyability does better (tasks whose truth coincides with the *modal* example output score 0.68 versus 0.59 otherwise, $p = 0.014$; the six tasks whose truth never appears among the example outputs are the hardest of all, at 0.42). Models share a difficulty axis, but it is the pattern-completion axis, not the algorithmic one.

Scale, quantisation, post-training: what moves the gap. The controlled comparisons show that the interventions which most improve conventional benchmarks do not close the fidelity gap, and one widens it (Fig. 7). These orderings are measured on the full divergence, whose ranking is carried by the termination coordinate (Spearman +0.97) and not reproduced by the bit coordinate (+0.18, $p = 0.12$); they are termination-coordinate findings (§2.5). *Quantisation* turns out to be benign for most of the panel: four of the six ladders are flat in both facets down to the lowest precision served (0.76 accuracy at 1.75 bits per weight for Qwen3.6-35B-A3B). Where quantisation

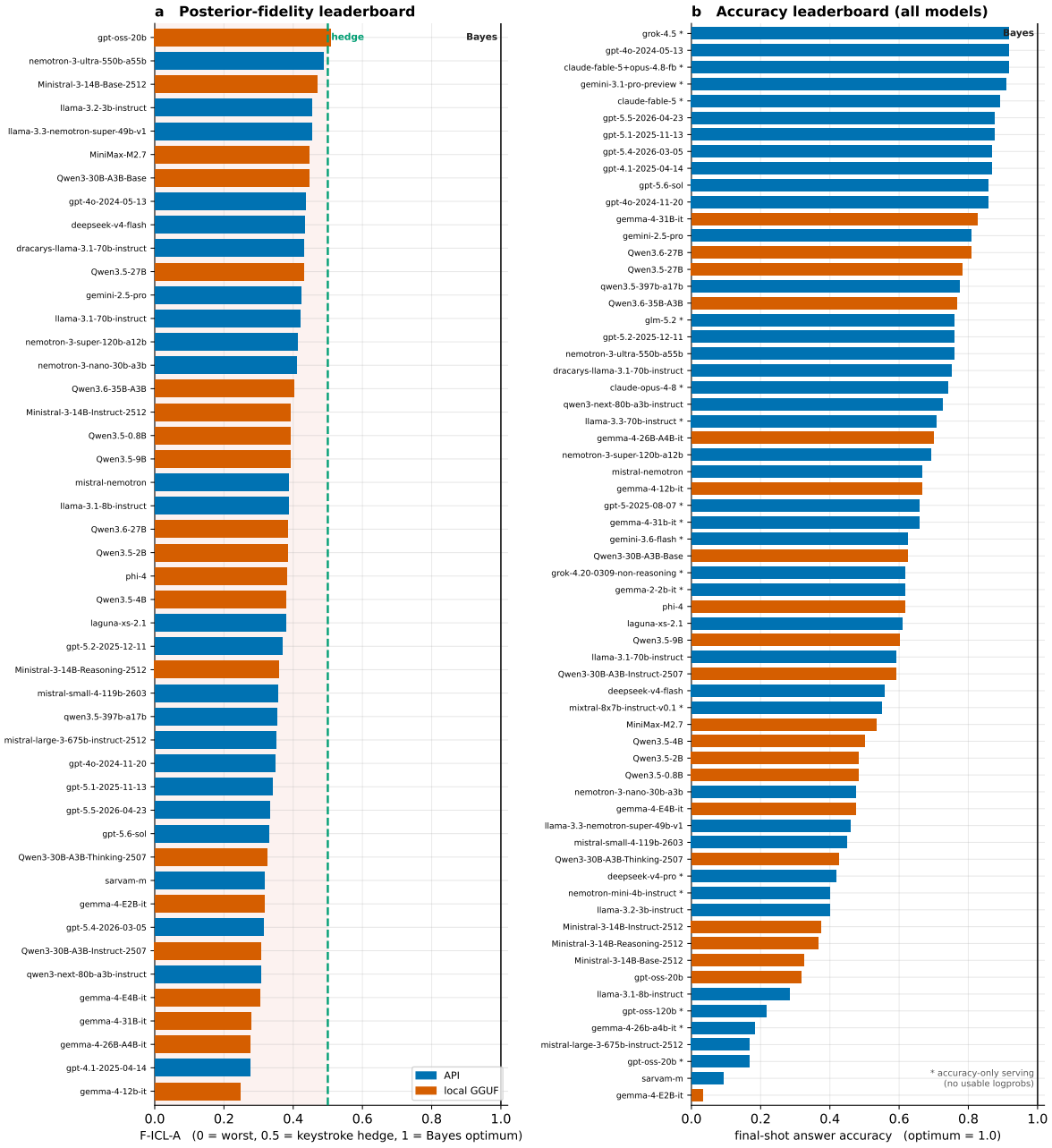

Figure 6: **Panel leaderboards.** Best-scoring precision per model ($n = 46$ models with usable log-probabilities in (a), 64 models with non-degenerate answer serving in (b)); blue = API, orange = local GGUF. **a**, The anchored F-ICL-A scale (models with usable log-probabilities). The dashed line is the keystroke reference and the shaded band the sub-anchor region; the solid line is the Bayes optimum. All models but one score below the keystroke reference; the exception, locally served `gpt-oss-20b` (plotted at its best-scoring precision, Q6, 0.53), is the only bar to cross it. **b**, Final-shot answer accuracy for *every* model, including the accuracy-only API rows (marked $*$) whose serving exposes no usable log-probabilities and which therefore cannot appear in (a); the fidelity and accuracy orderings are visibly unrelated.

does bite, it is accuracy that breaks: `gpt-oss-20b` collapses from 0.32 to 0.07 and phi-4 halves from 0.62 to 0.34 at 2-bit, while their fidelity declines far less ($0.51 \to 0.45$ and $0.38 \to 0.27$, the latter only at the extreme 2-bit point; Fig. 7b,d). The a-priori expectation was the opposite, that the log-probability metrics would erode first; in no ladder does fidelity degrade before accuracy. A distribution already far from the optimum has little further to fall, whereas the argmax that accuracy reads off is exactly what low-bit quantisation perturbs. *Scale* raises accuracy but barely moves posterior fidelity (Qwen3.5 0.8B→27B: accuracy +0.30, F-ICL-A +0.04, still below the keystroke reference; Fig. 7a). *Post-training* moves it the wrong way: in both families the *base* model has the highest fidelity and instruction tuning lowers F-ICL-A (Ministral-3-14B $0.47 \to 0.39 \to 0.36$; Qwen3-30B-A3B $0.45 \to 0.31 \to 0.33$; the base and instruct 95% CIs do not overlap in either family, and the base→instruct drops exceed the CI half-widths several-fold; Fig. 7c) even as instruction tuning holds accuracy. The base→instruct step is the large and consistent one; the further step to explicit reasoning continues the decline in one family ($0.39 \to 0.36$) and reverses slightly in the other ($0.31 \to 0.33$), so we claim the instruction-tuning effect and not a monotone ordering across all three stages (Fig. 7c). Post-training sharpens a model's distributions toward its aligned answer format and away from the algorithmic prior, the dissociation the benchmark was built to detect.

Spend does not buy fidelity either. Across the seventeen metered frontier rows, cost trends with answer accuracy (Spearman $\rho = +0.51$, $p = 0.036$) but shows no detectable association with posterior fidelity ($\rho = +0.30$, $p = 0.43$, $n = 9$ rows with a usable distributional read-out); the panel's most faithful row costs \$4.54 against \$8.80 for the most expensive row that yields a fidelity score. Only the accuracy leg reaches significance, and at this sample size we report both as description rather than inference.

Posterior fidelity is scale-invariant where accuracy is not. The within-family scale ladder generalises to the whole panel as a double dissociation (Fig. 8). These orderings are measured on the full divergence, whose ranking is carried by the termination coordinate (Spearman +0.97) and not reproduced by the bit coordinate (+0.18, $p = 0.12$); they are termination-coordinate findings (§2.5). Across the 32 base models with disclosed parameter counts, spanning 0.8B to 675B, nearly three orders of magnitude, answer accuracy rises significantly with scale (Spearman $\rho = +0.39$, $p = 0.028$) while F-ICL-A shows no detectable association ($\rho = +0.10$, $p = 0.59$), pinned just below the anchor at every size. Both coefficients are Spearman rank correlations against $\log_{10}$ parameters, two-sided, estimated on the *same* 32 models and therefore at identical power; that shared power is what makes the contrast informative rather than an artefact of sample size. We state the null with the caution it deserves: at $n = 32$ the smallest detectable $|\rho|$ at $\alpha = 0.05$ is 0.35, so the fidelity result excludes a large positive association but not a moderate one (95% CI -0.26 to $+0.43$). What it does establish is that the two axes move differently under the same test on the same models. On this evidence, scale does not deliver fidelity to the Bayes-optimal inductive standard over the range accessible to open models today: whatever additional capability parameters buy, we detect none of it on the fidelity axis.

Evidence narrows the induction gap where scale does not; it does not close it. The headline aggregates over all evidence levels, so we also report the anchored score conditioned on minimum evidence (Fig. 9): F-ICL-A recomputed on the cells with at least n_0 examples, with the per-cell hedge and worst-case anchors recomputed on the same cells so both branches of Eq. (20) remain exact. On induction, whose read-out has no termination coordinate, the panel mean rises from 0.414 over all shots to 0.489 at n_0=7 (task-bootstrap 95% CI $[0.43, 0.61]$), and the number of

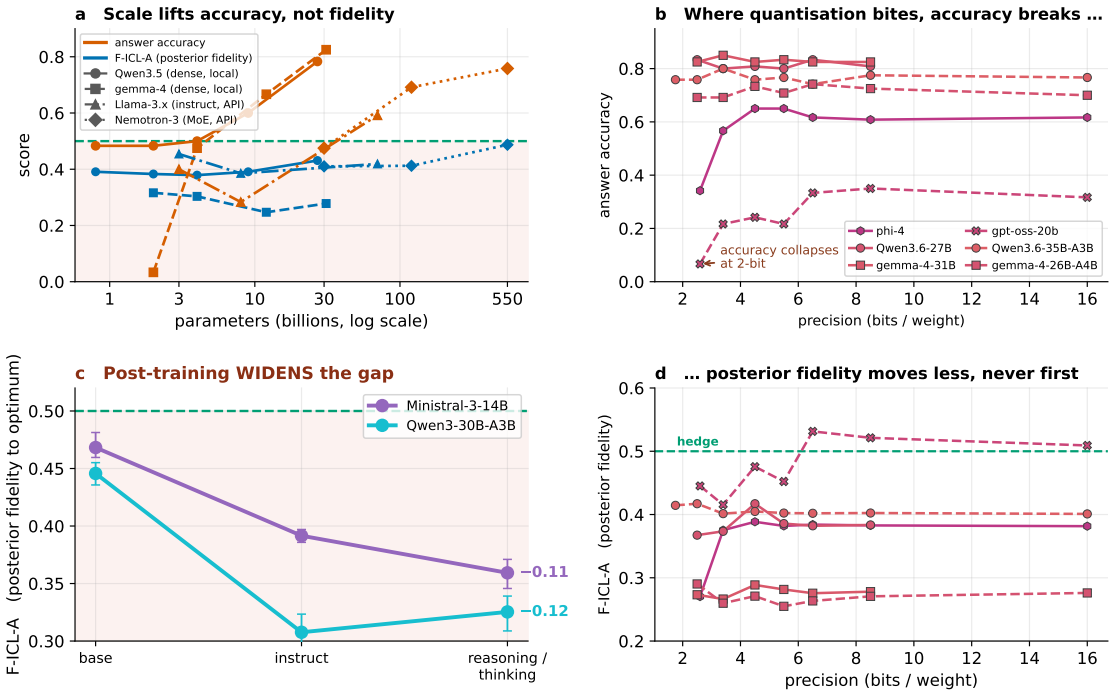

Figure 7: **What moves the gap.** The keystroke reference is marked in every F-ICL-A panel. **a**, Scale: the four within-family ladders the panel supports, two local dense families (Qwen3.5, circles; gemma-4, squares) and two API ladders held to one generation and post-training stage (Llama-3.x instruct, triangles; Nemotron-3 mixture-of-experts (MoE), diamonds). Answer accuracy (orange) climbs with parameters in every family, e.g. +0.30 across Qwen3.5's 34× range, while posterior fidelity (blue) moves little and stays at or below the keystroke reference throughout. **b**, **d**, Quantisation, all six ladders (marker = family, colour = parameter count as elsewhere, dashed = MoE). Four ladders are flat in both facets down to the lowest precision served; in the two that degrade (gpt-oss-20b, phi-4), answer accuracy (**b**) breaks earlier and harder than F-ICL-A (**d**), and in no ladder does fidelity degrade first. **c**, Post-training: F-ICL-A falls sharply from base to instruct in both families and then continues down in one and edges back in the other in the only two open families that ship all three stages at matched size (change annotated; error bars are 95% task-bootstrap CIs, twins resampled with their base task); alignment widens the gap.

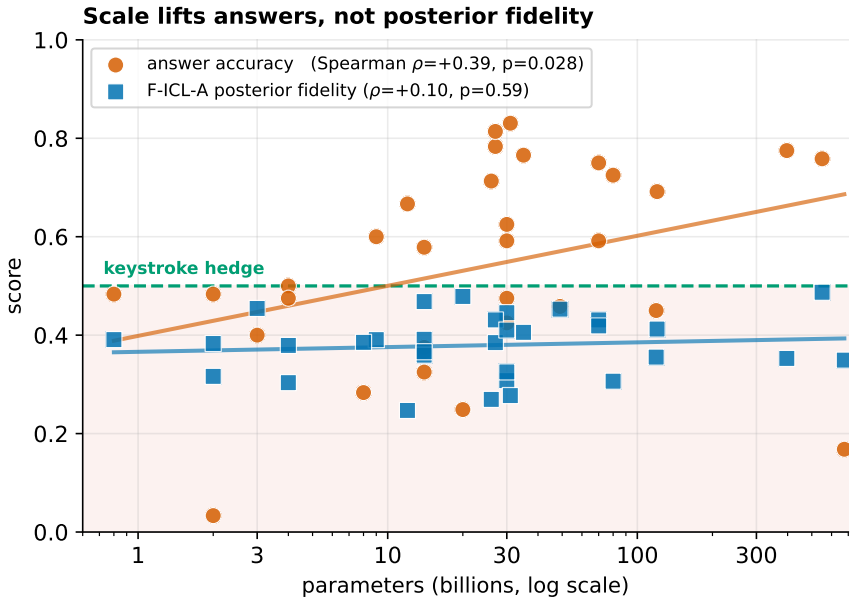

Figure 8: **Scale lifts answers, not posterior fidelity.** Base-model means for the 32 panel members with disclosed parameter counts. Answer accuracy (orange) rises significantly across three decades of scale (Spearman $\rho = +0.39$, $p = 0.028$); F-ICL-A posterior fidelity (blue) shows no detectable association ($\rho = +0.10$, $p = 0.59$; 95% CI -0.26 to $+0.43$) and remains below the keystroke reference at every size. Both are two-sided Spearman rank correlations against $\log_{10}$ parameters on the same 32 models, so the two tests have identical power. Lines are least-squares fits in log-parameters and are shown for orientation only; the reported statistics are rank-based and unaffected by the log transform.

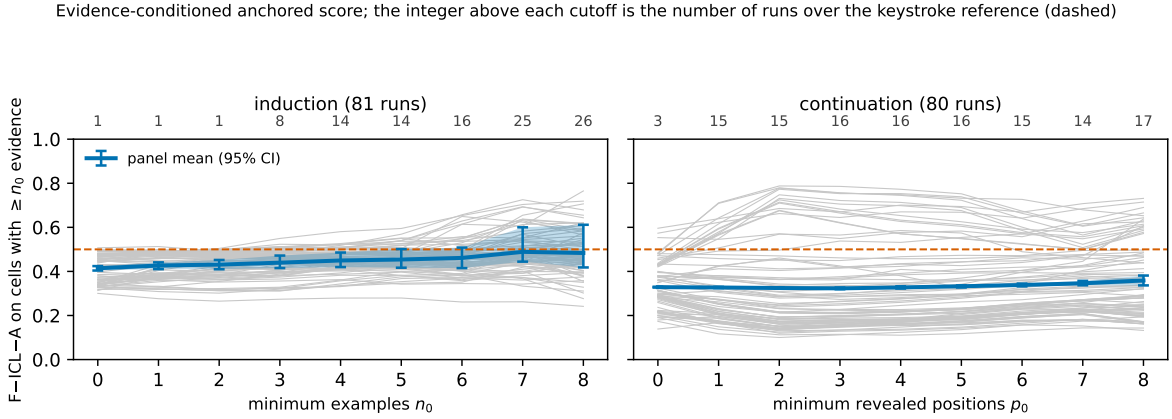

Figure 9: **The evidence-conditioned anchored score.** F-ICL-A recomputed on the cells with at least n_0 examples (induction, left) or p_0 revealed positions (continuation, right); thin lines are the 81 (respectively 80) runs with a distributional read-out, the bold line is the panel mean with a task-bootstrap 95% band, and the count above each cutoff is the number of runs over the keystroke reference (dashed). Induction rises with evidence but its mean stays below $\frac{1}{2}$ at every cutoff; continuation is flat at every depth: the termination miscalibration does not fade with evidence. Computed entirely from the frozen artefacts; the driver is released with the code.

runs above the keystroke reference rises from 1 of 81 to 25 of 81; the crossers concentrate in the newest model families (the Qwen3.6 ladders and Gemini 2.5 Pro from n_0=3–4, GPT-5.6 from n_0=5). Two bounds on the reading: the late-cutoff estimate rests on the cells where the posterior has not yet collapsed (§2.1), so the intervals widen accordingly; and the panel mean stays below $\frac{1}{2}$ at every cutoff, so the sub-reference headline is not an artefact of averaging over the low-evidence regime. On continuation the picture is unchanged at every depth (mean 0.329 to 0.359): the termination miscalibration does not fade with evidence, although excluding the first revealed position alone moves 12 of 80 runs above the reference on this facet, quantifying that single structural cell's contribution (§2.5). Evidence is thus the one lever in this paper that *improves* posterior fidelity, and it improves only the bit-level facet.

Polarity twins expose a model bias the yardstick lacks. Because every task appears with its complemented twin and the sF optimum is provably identical on the pair (Methods 4.3), the polarity gap Δ_{pol} (Eq. (12)) is exactly zero for the optimum and for *any* complement-equivariant predictor, so a nonzero value is attributable to the model alone. Quantisation ladders are correlated replicas of one model, so the unit of analysis is the *base model* (cluster mean over its runs). Across the 46 base models the mean gap is small and positive but sensitive to panel composition: with the open panel alone the clustered mean is +0.0030 ($t(36) = 2.10$, $p = 0.043$; 24 of 37 positive), while the frontier rows, several of which carry comparatively large gaps of either sign (the frontier GPT-5.6's is −0.023), move the full panel to +0.0020 ($t(45) = 1.40$, $p = 0.17$; Wilcoxon signed-rank $p = 0.037$; 28 of 46 positive). We read this as a per-model bias that is real and measurable on a provably neutral instrument, with no panel-level sign consensus. On the accuracy route the per-model bias is larger and two-signed (absolute mean 0.06), a confound invisible to any benchmark whose targets are not polarity-balanced. The small mean, however, hides an item-level symmetry violation: any solver that operates at the level of *rules* solves a task if and only if it solves its complement twin, yet within-run original–twin correctness correlation is only $\phi = 0.29$, and the median model produces the bitwise-complemented answer on its twin task just 52% of the time (range 20–72%; 40% even restricting to pairs where both answers are wrong, a pure style probe with accuracy removed). Complement-equivariance, which the symmetrised machine grants the optimum by construction, is itself a model capability the twins measure, and most of the panel lacks it.

Serving method is not neutral: it changes delivery, content, and even the sign of comparisons. Three models served two ways each (identical weights, identical prompts, temperature 0; calls paired exactly through the content-addressed cache) decompose the serving effect into two mechanisms. The first is *delivery*: on the API side, `gpt-oss-20b` returned an empty reply on 52% of paired answer calls and MiniMax-M2.7 on 45%, while the local builds always answered; scored naively over all paired cells, empties made the local build look better for gpt-oss (0.27 against 0.24), though the paired difference is not significant on its own (McNemar exact $p = 0.085$). The second is *content*: restricted to pairs where both stacks deliver, only 28% (34%) of greedy outputs are byte-identical, and the accuracy comparison *inverts*: the API side is right on 0.50 (0.65) of delivered pairs against the local 0.24 (0.36). The gemma-4 pair isolates content cleanly: served by Google's own API with zero empty replies on either side, the same checkpoints agree on only 18% of outputs and the direction flips again, local 0.47 against API 0.18, and the 31B pair the same way (0.57 against 0.38, with 36% of outputs identical). Serving thus affects whether an answer arrives at all, what the answer is when it does, and which side of a comparison wins, model-dependently. A leaderboard that mixes serving providers inherits all three effects; F-ICL reports the serving

path per row and audits reply delivery explicitly. No same-model serving pair exposes usable API log-probabilities, so the analogous comparison for the *distributional* readout remains open.

2.5 Robustness of the measurement, and what the metric licenses

A benchmark whose target is a designated distribution invites a specific objection: the score measures agreement with that target, and any normative reading of it depends on the target being the right object for the data actually evaluated. Ours is not, in one respect that we now quantify. Tasks are sampled from the F-prior and then *curated* (equal quotas over output length and over complexity terciles, plus learnability gates), so the evaluated law differs from the prior and Eq. (1) is not the Bayes predictor for it. The measurements below are all computed from the frozen artefacts (the full audit is released with the code).

The optimum is not the best predictor of the curated sample. On the 1,950 revealed continuation positions the optimum's mean log-loss on the *realised* token is 2.65 bits against the keystroke reference's 1.59, and it is the worse of the two on 57% of cells; 39 of the 80 scored runs predict the realised data better than the optimum does. On induction the two are indistinguishable (0.93 bits each, averaged over the nine shot counts; the keystroke reference is ahead at zero and two examples). This is what curation does: quotas over length and complexity deliberately over-sample outcomes the prior finds improbable, and a predictor faithful to the prior is penalised on them. It follows that F-ICL-A measures agreement with the algorithmic posterior and *not* predictive skill on this sample, and that "below the keystroke reference" cannot be read as "worse than making no inference". We use the weaker and correct reading throughout: a model below the anchor is further from the algorithmic posterior than a fixed length-prior null is.

Most of the raw divergence is the termination coordinate, where the target is miscalibrated by construction. Splitting the continuation divergence into a termination coordinate (END versus not) and a bit coordinate ($\{0, 1\}$ renormalised) shows the termination coordinate carries 89% of it (mean 0.387 of 0.435 bits) and reproduces the full ranking at Spearman +0.97. That coordinate is also where the curation bites hardest: the shortest sampled output has length three, so a string *cannot* end at positions 1 or 2, yet the optimum places mean END mass 0.37 and 0.82 there. A model that correctly refuses to terminate early is penalised at exactly the positions where the target is wrong about the evaluated sample.

The headline survives both corrections; its magnitude does not. Re-weighting cells to undo the length quota leaves the count unchanged at 77 of 80 runs below the anchor. Discarding the termination coordinate entirely and scoring only the bit coordinate, the half of the measurement the quota cannot distort, leaves 69 of 80 runs below the corresponding anchor (0.121 against 0.057 bits). The two sets are not nested: eleven runs leave and three enter, the three being `gpt-oss-20b` replicas, so on the bit coordinate the panel's sole at-anchor exception falls below it with the rest. The effect is roughly a third of its raw size once termination is removed, and the qualitative result is not an artefact of either the sampling design or the termination miscalibration. But the two coordinates do not order the panel alike. Spearman between the full metric and the bit coordinate is +0.18 ($p = 0.12$), against +0.97 between the full metric and termination. Every ordering result in this paper (ladders, triads, quantisation, clustering) is therefore measurably a termination-coordinate finding, and we label them as such rather than as statements about the bit-level posterior. Predictive regret tells the same story in absolute terms: the panel pays +1.86 bits per token against the anchor and +0.79 against the optimum.

A budget change does not manufacture the result. The mismatch objection has a computable special case: if "below the anchor" were an artefact of the particular bounded prior, a different budget should move it. Re-serving the induction optima from the complete $L \leq 13$ store and rescoring every cached model distribution against them (§2.4) leaves the per-run ordering at Spearman 0.975 across a budget range of $L \leq 6$ to $L \leq 13$ and a change of weighting convention. The count is not quite unchanged: on the induction facet 80 of 81 runs are below the anchor under the pipeline-scale build optima and 76 of 81 under the $L \leq 13$ store, so four runs cross when the budget changes, all of them within 0.02 of the anchor to begin with. The macro count is stable at 77 of 80. That is a control over *budget*, not over the reference machine itself. The machine axis is covered separately and externally: the certified description lengths of sF agree with published Coding Theorem Method values, computed over a different machine family by output frequency, at Pearson 0.91–0.92 (Methods). The two controls therefore bracket the objection, one varying the machine at fixed object and the other the object at fixed machine, while their conjunction, a conditional posterior served from another family, remains untested and unclaimed.

The anchor is a length-prior null, not an uninformed one. The induction anchor renormalises $c(s) = (1/3)^{|s|+1}$ over the candidate set, and that measure is length-decreasing exactly as the posterior is: it sits closer to the optimum (0.110 bits) than a flat distribution over the same candidates does (0.122). It is therefore not a "no inference" point, and our classical nulls say the same thing from the other side, the frequency table scoring 0.60 and the pooled predictor 0.73 while performing no inference at all. The anchor's role is to be a fixed, computable, model-independent reference with an exact value of $\frac{1}{2}$; we name it the keystroke (print-prior) reference and no longer read its sign as the presence or absence of inference.

Prompt format moves the absolute score, and the direction is diagnostic. (The unmodified prompts are in Appendix C.) A five-model ablation over wording, templating, the grammar constraint, example order and the terminator glyph, run on the same pipeline-scale build the panel is scored against and with each model served through the template its panel row uses, is released with the code (Qwen3.5-4B, Qwen3.5-9B, Ministral-3-14B-Reasoning, gemma-4-12b-it, phi-4; 50 continuation tasks per arm). Two quantities appear below and must not be conflated: the anchored score F-ICL-A (dimensionless, higher is better, Eq. (20)) and the raw Jensen–Shannon divergence to the optimum (in bits, lower is better). Movements are reported in the score unless bits are named. Wording, the grammar constraint and example order are immaterial: rewording the instruction moves the score by at most 0.041, removing the grammar by at most 0.002, and shuffling the example order moves full-shot induction divergence by at most 0.017 bits, against an optimum that is order-invariant by construction.

Changing the *template family* is not immaterial, but the benchmark's own geometry identifies most of that movement as a read-out failure rather than a fidelity gain. Serving under a bare chat template raises F-ICL-A by as much as 0.26 in the four models whose template actually changes, while driving their answer accuracy from 0.26–0.46 down to 0.00–0.11, and the resulting continuation divergences collapse onto a single value: 0.189 bits for Ministral-3, Qwen3.5-9B and gemma-4-12b alike. That number is not a property of any model. It is $\mathrm{JS}(P^\star \,\|\, U)$ on exactly these 50 tasks, 0.1888 bits, the value returned when the endpoint resolves none of the three classes, and it is the same signature the degeneracy audit of §2.4 uses. phi-4 supplies the internal control: its panel row already uses the default chat template, so this arm is a no-op for it, and its score moves by +0.001 with accuracy unchanged. Two arms are genuine format effects rather than breakage, and we report them as such: Ministral-3 under a raw completion wrapper gains 0.090 while its

accuracy *rises* (0.260 → 0.284) and its divergence stays well clear of the floor at 0.331, and phi-4 gains 0.023 at accuracy 0.362. The terminator glyph is likewise not immaterial: replacing the full stop with `x` moves continuation divergence by up to 0.24 bits (Qwen3.5-4B, 0.469 → 0.229) in four of five models. Adding an explicit chain-of-thought instruction without suppression costs score in four of the five (up to −0.136) and is neutral in the fifth, which is the behaviour the no-think wrappers exist to prevent.

Because a uniform read-out scores 0.189 while these models score 0.333–0.668 when they are following the format, *breaking a model's output format improves its F-ICL-A score*: the numbers we report are what models achieve when the instrument is working, and this failure mode inflates rather than deflates them. We therefore treat the *relative* comparisons at fixed format (every ladder, triad, quantisation and panel ordering in this paper) as the load-bearing results, and read any single row's absolute distance from the anchor as conditional on the serving format, of which a template change can consume a large fraction.

2.6 Models cluster in distribution space by family

The panel's runs are not just points on a score scale: each one is a *function* from context to distribution, so the served distributions themselves define a geometry. For the 80 runs with usable continuation read-outs we computed the pairwise mean Jensen–Shannon *distance* (the square root of the divergence, which unlike the divergence satisfies the triangle inequality, so the embedding, linkage and silhouette computations below are metric-valid) over their shared (task, position) cells, and embedded the resulting matrix by principal coordinate analysis (Fig. 10a). Classical PCoA is used rather than a neighbour-embedding method such as t-SNE deliberately: our claims are about *global* distances, how far apart two families sit, which neighbour embeddings distort by construction. Two axes carry 70% of the positive eigenvalue mass with a non-Euclidean residue of 2%, so the picture is a faithful one. The geometry passes the obvious sanity check: quantisation replicas of one model sit at mean distance 0.20 from each other against 0.51 between different models, a factor of 2.5.

Structure appears at two scales. The silhouette-optimal partition is a single clean split ($k = 2$, silhouette 0.51) that separates 13 runs from the remaining 67, and it is a split in *style*, not in quality: the two groups have indistinguishable answer accuracy (0.617 versus 0.616) and similar posterior fidelity (0.40 versus 0.37). They differ in how they spread probability mass, not in how well they score. Cutting finer ($k = 6$, Fig. 10b) recovers the quantisation ladders as clusters in their own right (phi-4's, gpt-oss's, gemma-4's and Qwen's ladders each hold together), confirming at the level of served distributions what the score ladders showed pointwise: compression barely moves a model. A permutational multivariate analysis of variance on the distance matrix confirms that model family explains distributional position ($F = 3.93$, $p = 0.0002$, 4,999 permutations), whereas the API-versus-local contrast returns no detectable effect ($F = 1.02$, $p = 0.33$). That second contrast is *not identified* as a test of the serving route: no model in this panel is served both ways with a usable distributional read-out (§2.4), so route is perfectly confounded with model identity and the family factor above absorbs it. We therefore report the null as uninformative about serving rather than as evidence that the geometry is free of it. What does bound measurement noise is the within-model quantisation replica above, which places replicas of one model adjacent under the same read-out path; and serving does change greedy *answers* substantially (above), so the two read-outs are affected differently. Position is only weakly tied to score: Mantel tests relate distributional distance to differences in fidelity ($\rho = 0.27$, $p = 0.001$) and in accuracy ($\rho = 0.12$, $p = 0.008$, $n = 80$), both significant but far from determinative. Models can be distributionally remote and still score alike.

That last fact is what makes the panel's aggregate behaviour intelligible. Clustering the API rows by answer disagreement and drawing matched four-model ensembles, a vote among models from a *single* cluster scores 0.60, while a vote among models drawn from *different* clusters scores 0.76. The ensemble advantage reported above is therefore a diversity effect and not a head-count effect: what the panel supplies collectively is not more of the same signal but complementary errors. The limit of that logic is equally clear. Aggregation operates on *answers*; the deficiency we measure is in *distributions*. A committee of predictors that individually sit below the keystroke reference can vote its way to the panel's best accuracy, but no vote converts sub-reference posteriors into a calibrated one: the ensemble inherits the axis it was built on.

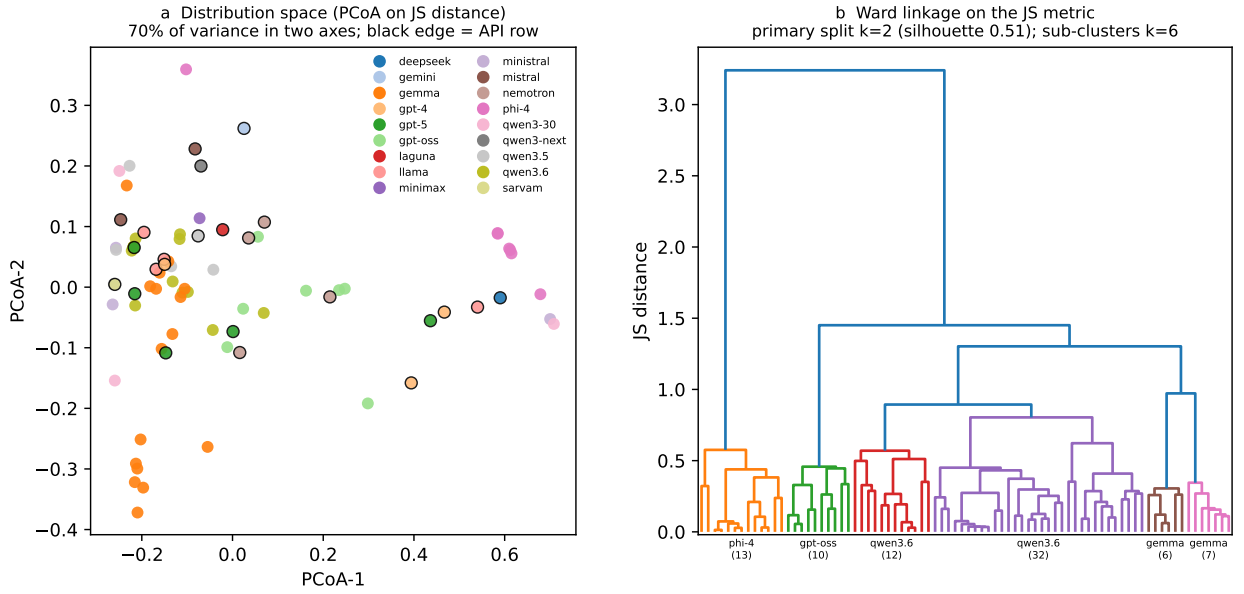

Figure 10: **The panel in distribution space. a**, Principal coordinate analysis of the pairwise Jensen–Shannon distance between the served continuation distributions of the 80 runs with usable read-outs (two axes carry 70% of the positive eigenvalue mass); colour is model family, black edges mark API rows. **b**, Ward linkage on the same metric: the silhouette-optimal split is $k = 2$ (silhouette 0.51), and the finer $k = 6$ cut recovers quantisation ladders as clusters (labels give each sub-cluster's dominant family and size). The dominant split separates runs by how they spread mass rather than by how well they score, and replicas of one model remain adjacent; the matched-ensemble decomposition is reported in the text.

2.7 The frontier under an exact lens: no better, and harder to measure

The panel's frontier rows support a three-part finding that the exact instrument makes visible: frontier releases have not improved on the fidelity axis, the distributional readout needed to measure it is being withdrawn, and what remains measurable is increasingly decided by the serving and safety layers rather than by the models.

Fidelity has not improved across two years of frontier releases. The per-run ordering behind this comparison is shown in Fig. 6. Parameter count is one axis along which capability grows; release date at a frontier laboratory is another, and it folds in every ingredient of frontier

progress at once (data, architecture, post-training, scale). Because OpenAI still exposes token log-probabilities (see below), we evaluated a dated generational ladder of nine models from GPT-4o (May 2024) to GPT-5.6 (July 2026) through the identical harness (Fig. 11). Fidelity does not improve with frontier time. The oldest model is the best: May-2024 GPT-4o reaches F-ICL-A 0.43 (task-bootstrap 95% CI $[0.42, 0.45]$), together with the highest answer accuracy in the entire panel (0.92, later matched by the two-model Fable-with-fallback stack and by Grok 4.5 at low reasoning effort, below), and every later release sits in a 0.27–0.37 band whose CIs never reach 0.39, so the separation of the 2024 point from the entire later era is itself significant ($0.35 \rightarrow 0.27$ over 2024–25, then 0.31–0.37 across the five GPT-5.x releases), echoing at release timescale the within-family result that post-training moves models away from the posterior. Two caveats bound the reading: nine releases support no trend statistic, and the GPT-4 era is measured over a different serving route (Chat Completions) than the GPT-5.x era (Responses), a difference we cannot control for, and our local-versus-API comparison shows that serving route is not generally neutral; and every frontier row is served with deliberation disabled or minimised, because log-probabilities are exposed only in those modes (Methods), so the comparison speaks to the serving configurations that expose distributions rather than to the systems at full deliberation. The conservative statement is that two years of frontier development produced no improvement in fidelity as served through the distribution-exposing modes. The pattern is not ours alone: a compression-based test of frontier models over a different measurement axis independently reports that the newest releases often regress[20], so two unrelated algorithmic-information instruments agree that recency is not fidelity. Accuracy meanwhile wanders non-monotonically (0.66 at the original GPT-5, 0.76 at GPT-5.2, near 0.86 elsewhere). The ladder also carries a serving datum in its interior: the original GPT-5 release served no log-probabilities in any mode, a one-generation gap in the distributional readout that later releases reversed.

The distributional readout is disappearing from frontier APIs. A benchmark that scores served distributions depends on providers exposing them, and direct probes of the four frontier laboratories (July 2026; probe scripts in the repository) show that this dependency is now precarious. OpenAI is the only laboratory that still returns token log-probabilities for its newest models: the GPT-4-era models expose them through Chat Completions, the GPT-5.1 line and later through the Responses API with reasoning effort `none`, and the original GPT-5 release exposed them through neither, a one-generation gap later reversed. Google has withdrawn them at the frontier: on Vertex the native API returns HTTP 400 ("Logprobs is not supported for this model") for every Gemini generation after 2.5, while the 2.5 generation, which still serves them and provides our full Gemini row, has already been retired from the consumer developer API. Google's OpenAI-compatibility layer strips log-probabilities from *all* models it hosts, including third-party models whose native APIs nominally support them. xAI accepts the log-probability parameters in its request schema and silently returns none for every current Grok model; its completions endpoint refuses outright ("Raw sampling is not supported for reasoning models"), and Anthropic's Messages API has never exposed them (no such parameter exists; probed directly, July 2026). Each withdrawal has a distinct signature (explicit refusal, silent omission, schema-level removal), which matters practically: silent omission is exactly the failure mode our degeneracy audit exists to catch. Distribution-level measurement of frontier models currently runs through a single provider, benchmark rows can become unrepeatable when a provider retires a model (the mistral-large episode of the panel footnote), and open-weight local serving is the only path whose distributional readout is fully under the evaluator's control. There are some sampling approachs that could ne used to obtain distributional features of the outputs, but these are out of scope for the current paper.

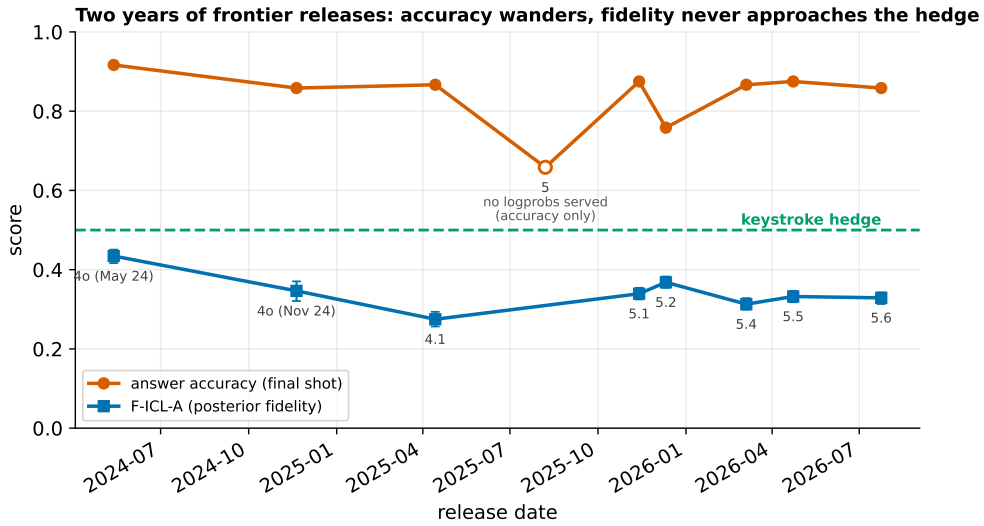

Figure 11: **Two years of frontier releases at one laboratory.** Final-shot answer accuracy (orange) and F-ICL-A posterior fidelity (blue) for nine dated OpenAI releases evaluated through the identical harness (chat-logprob route for the GPT-4 era, Responses route with reasoning effort none for GPT-5.1 and later). Error bars on the fidelity series are 95% task-bootstrap CIs (twins resampled with their base task; $n = 60$ induction and 150 continuation base tasks, twins resampled with their base). Fidelity is highest for the oldest model, and no release approaches the keystroke reference; the open marker is the original GPT-5, which exposed no log-probabilities in any mode and is therefore accuracy-only.

The safety layer is part of the serving stack. Anthropic's Claude rows make the point most directly. Claude enters the panel accuracy-only (no log-probabilities, as above); what the probes additionally revealed is that a *safety classifier* can set a frontier model's benchmark score, and we measured this as a controlled three-configuration experiment on identical prompts. *Pure Fable 5*: the cyber-content filter refuses 16% of our plain binary-string induction queries (`stop_reason: refusal`, category `cyber`; false positives on strings that encode no code and no instructions), peaking exactly in the ambiguous-evidence regime (35 of 120 tasks at four examples) and falling to 12 by eight; scored as served it reaches 0.89 final-shot accuracy, but on the queries it answers it reaches 0.99, and the comparison survives matching: restricted to the same 108 answered cells, May-2024 GPT-4o scores 0.95 and GPT-5.6 scores 0.91, so the margin is not an artefact of the classifier removing hard cells. *Pure Opus 4.8*: one refusal in 1,080 cells but far weaker reasoning, 0.74, so the two Anthropic failure modes, capability and classifier taxation, are cleanly dissociated. *Fable 5 with the server-side Opus 4.8 refusal fallback* (the provider's recommended production configuration): 0.92 as served, tying May-2024 GPT-4o and Grok 4.5 for the best accuracy in the entire panel (all three at 110/120). Yet 4% of cells *still* return nothing, because on 38 of those 43 cells the fallback chain refuses as well. A same-lab sibling, `claude-opus-5`, is stricter still: its classifier refuses *every* induction query (pilot 27/27, confirmed by replay), and a system prompt declaring the benchmark context does not unblock it, so the model cannot be measured on this benchmark at all. Two further serving facts fall out of the experiment. Re-running Fable 5 on byte-identical prompts (the fallback row repeats every non-refused call; $n = 898$ paired calls, hours apart) reproduces its answers only 86% of the time. The `temperature` parameter is *rejected* on current Claude models ("deprecated for this model", HTTP 400), so no client-side setting can remove this variation; the

design cannot separate sampling nondeterminism from silent server-side updates, and either way the 86% is a floor on single-run reproducibility for every accuracy-only Claude row. And an offline merge of the two pure rows (Fable's answers, Opus's on refused cells) predicts the fallback stack's score to within one task (0.91 vs 0.92), confirming the in-call fallback behaves as documented. Whether a frontier system can be benchmarked, and what score it gets, can here be decided by its safety layer rather than its reasoning: entirely so for `claude-opus-5`, by a sixteen-percent tax for Fable 5, and not at all for Opus 4.8.

Summary

Across a heterogeneous panel the findings are consistent. Models select answers well above chance, but their predictive distributions sit at or below the keystroke reference, below a frequency lookup table, and at the level of low-order prefix statistics. A single example makes nearly every model worse before it gets better; a large share of errors is inconsistent with the benchmark's support for the shown evidence, and halting predictions anti-correlate with the truth. None of the levers that improve conventional benchmarks close the gap: not scale, not two years of frontier releases, and not post-training, which widens it. Within a task, added evidence narrows the induction gap without closing it, and leaves the termination miscalibration unmoved. Two conditions bound the reading. The gap is distance from the algorithmic posterior, not evidence that models predict the curated sample worse than the optimum does, which on those sets they often do not (§2.5); and the orderings above are carried by the termination coordinate, which the bit coordinate does not reproduce (§2.5).

3 Discussion

F-ICL reframes the question "does this model reason algorithmically?" as a measurement against an exact, computable optimum rather than against other models or human intuition. Its defining feature is that the target is the provably Bayes-optimal inductive predictor under a bounded universal prior, computed in closed form from an exhaustively enumerated and independently certified reference machine. This makes "how close is the model to ideal inference, and how fast does it get there?" a quantity we can actually compute, at a scale that is small yet genuinely algorithmic.

Three properties make the benchmark practical. It is *exact*: every target is an exact posterior, not an estimate, so a nonzero gap is a real distance from a designated posterior rather than baseline noise. That is a weaker statement than predictive inferiority, and deliberately so: on the curated evaluation sets the optimum is not itself the best predictor of the realised tokens (§2.5), so the gap measures departure from the algorithmic posterior, not failure to predict the data. It is *cheap to run*: the expensive enumeration is frozen once into a self-contained dataset, after which evaluating a model costs only its own inference. It is also *difficulty-aware*: tagging tasks by Bayes sample complexity makes the optimum a per-task difficulty scale, so a model's effective sample complexity can be compared directly against the information-theoretic minimum. The ordering results reported above (scale, generational ladder, post-training) are measured on the full divergence, which §2.5 shows is carried mainly by the termination coordinate; they should be read as statements about that coordinate rather than about the bit-level posterior. Across 37 open models and frontier systems from four proprietary laboratories the panel returns a consistent result: in-context answer selection well above chance, yet predictive distributions at or below the keystroke reference relative to the algorithmic optimum, at a level indistinguishable from low-order prefix statistics. The framework states this in exact, reproducible numbers: posterior fidelity to the optimum shows no detectable improvement across three orders of magnitude of scale or across two years of frontier releases in

the serving modes that expose distributions, is degraded rather than improved by post-training, is invisible to accuracy-only evaluation, and is not delivered by frontier capability: the most faithful frontier model in the panel is the oldest one evaluated.

Sampled universal-machine data and exhaustive enumeration. The closest neighbour to this construction is the line of work that meta-trains networks on data generated by a universal Turing machine[31]. The reference machines are close relatives: their BrainPhoque is Brainfuck-equivalent and F is a minimal Turing-complete tape automaton in the same family, so the universality argument is common to both, and in both cases it is a property of the unbounded machine rather than of the instrument actually run. Their instrument is bounded at 1,000 steps, 200 memory cells and 256 output symbols; ours at 1,024 steps, 128 cells and $L \leq 13$. Neither samples nor enumerates anything universal, a point we state plainly above for our own budget.

The difference that matters is therefore not fidelity of the machine but sampling versus exhaustive enumeration, and it decides what can be measured. Sampling yields an unlimited training stream, which is exactly what meta-training needs, but it leaves the Solomonoff target out of reach: the optimal baseline in that work is by necessity a (rather loose, but non-trivial) upper bound on the log-loss computed from the length of the shortened generating program, which presumes knowledge of the program that produced the sequence and bounds a loss rather than supplying a posterior. Sampling also perturbs the object it approximates, since imbalanced brackets are skipped to keep every sampled program valid, which the authors note changes the program distribution. Exhaustive enumeration gives up unlimited data and program depth (a sampled program may run to the step budget, whereas ours is capped at thirteen instructions), and buys the posterior itself in closed form, together with three properties that follow only from completeness: a difficulty scale from Bayes sample complexity, an exact accounting of the missing mass rather than its absorption into a normaliser, and the exact removal of a known machine bias by symmetrisation. The two designs are complementary rather than competing: one is built to train a predictor, the other to score one.

One empirical point of contact deserves recording, because it is not specific to either design. The probability gap left by non-halting programs is handled there by adopting the normalised Solomonoff prior and filling the gap with absorbing tokens; here the same phenomenon surfaces as the termination coordinate, which carries 89% of the continuation divergence (§2.5) and which we therefore report separately from the bit coordinate. Sequence termination is the dominant nuisance coordinate in both constructions, and we suggest that any comparison of a served distribution against an algorithmic target report it separately rather than folded into a single divergence.

Limitations

The construction has clear boundaries, which are part of its design. The bounded prior at this budget reaches counting, arithmetic and bounded recursion, not the full richness of universal computation; F-ICL is a *clean* probe of algorithmic reasoning at small description length, not a proxy for open-ended reasoning. The optimality of the target is relative to the F-prior and to the realisable setting in which tasks are drawn from that prior; against a different reference machine the numbers would differ, though the construction is identical. The symmetrisation F→sF shows that a known structural bias of the reference machine can be removed *exactly* rather than merely noted. Continuation, as we noted, separates posterior fidelity well but the chosen foils weakly; abduction is the sharper separator, and evaluating models on it awaits a harness that can elicit a distribution over candidate inputs (today's serving interfaces expose next-token log-probabilities, which serve induction and continuation directly). On the measurement side, scores are conditional

on a prompt format, which we hold fixed across every row rather than eliminate; a five-model prompt ablation (wording, templating, the grammar constraint, example order and the terminator glyph, plus a chain-of-thought arm) is released with the code and discussed in §2.5; it is run on the pipeline-scale build the panel uses, and finds wording, the grammar constraint and example order immaterial while a template-family change can consume a large fraction of the anchor distance, mostly by degrading the read-out rather than the model, and models served through commercial APIs are measured as served: their quantisation and decoding stack are not observable, so API rows are annotated and, where log-probabilities are not exposed, restricted to accuracy facets. This restriction is tightening: our provider probes (§2.4) find token log-probabilities withdrawn or absent at the frontier for every laboratory except OpenAI, so the distributional facets of future panels may be measurable only for open-weight models served locally. The distributional readout also inherits a tokenisation confound[37, 38]: the harness folds next-token log-probabilities over the exact tokens `0`, `1` and the terminator, so mass that a vocabulary routes through merged multi-digit tokens is not credited, by an amount that can differ across tokenisers; the accuracy facets are immune by construction. A first-character folding readout that recovers merged-token mass bounds the effect directly: recomputing the continuation family under both foldings leaves the scores statistically unchanged (GPT-4o Nov-2024: mean JS 0.399 exact vs 0.402 first-character; GPT-5.5: 0.481 vs 0.480; Gemini 2.5 Pro, the panel's sole non-OpenAI frontier fidelity row and a distinct SentencePiece-family tokeniser: 0.334 vs 0.338, anchored score 0.398 vs 0.395; Llama-3.1-70B, a third vocabulary family, 0.408 vs 0.413 over all 1,736 cells; every difference is an order of magnitude smaller than the divergences being compared, and the largest of them, Llama's, has a task-bootstrap 95% CI of $[+0.004, +0.008]$ bits against a panel range of 0.29–0.51), so the sub-anchor result is not a token-boundary artefact for the four tokeniser families validated, including the digit-merging o200k family that motivated the concern; the induction readout is immune by construction, since candidates are teacher-forced bit by bit. Training-set contamination is unlikely for the current frozen tasks (the realised strings are new), but the machine and dataset are public, so future training runs could target them; the dataset is versioned and the pipeline regenerates fresh task sets cheaply, which is the standing remedy. Finally, the dumped program joint at full length is storage- and bandwidth-bound: the index for $L = 13$, $M = 8$ is large, and its grouping is a (vectorised) host-side `lexsort` after transfer. Regenerating dumps at still larger budgets would therefore benefit from moving the output keying onto the GPU, a planned extension; the present index already serves the complete $L \leq 13$ budget for every posterior we evaluate.

Three boundaries of interpretation deserve to be stated explicitly. First, the optimum is conditional on conventions the prompt never discloses: the reference machine, the length budget $L \leq 13$, the 1024-step and 128-cell bounds, the halting normalisation, the Levin-style halt-time weighting and the sF polarity mixture (Table 2). A model is therefore not asked to be Bayes-rational with respect to a prior it has been shown; distance to the optimum confounds inferential competence with knowledge of an unshared convention. Whether disclosing the machine and budget in-context closes any part of the gap is untested here and is the first control we intend to run. The bracketing of §2.4 is the reading least exposed to this concern, since it compares the panel against two *named* mixtures rather than against an absolute distance.

Second, the terminator glyph is not a neutral implementation choice. Replacing the full stop with `x` moves continuation divergence by up to 0.24 bits in four of five models (§2.5), a swing comparable to the panel's own spread. The direction is toward the uniform read-out floor of 0.189, which is the behaviour the degeneracy signature is designed to detect and which we exclude row by row; but we have not re-served the full panel under a non-terminator-adjacent glyph, so the headline count is established under one terminator convention rather than shown invariant to it.

Third, no evidence-only predictor in this work exceeds the keystroke reference: the Krichevsky–

Trofimov estimators reach 0.41 and 0.45, while the two nulls that do exceed it are constructed from the frozen optima. The benchmark is therefore demonstrated to be *discriminative*, so it separates Bayes-rational updating from pattern completion (§2.3) and orders predictors sensibly, but we do not exhibit a system without access to the target that attains a high score. Establishing that such a system exists, for instance a network meta-trained toward the universal predictor, is a natural next step and would convert the present comparative claim into an absolute one.

Several directions follow naturally. The same pipeline supports larger length budgets and alternative reference machines, allowing the difficulty ceiling to be raised or the prior to be varied as a robustness study, and connecting the difficulty scale to the length-generalisation behaviour of transformers on algorithmic tasks[34, 35]. The per-token and per-example divergence curves are a fine-grained behavioural signal that could be related to model scale, training data and inference strategy, and the certified witness programs make every task *exhibitable*: one can show the exact program a model failed to infer. Because the optimum is explicit, F-ICL can also serve as a target for training, not only evaluation, in the spirit of neural program induction and synthesis[32, 33]. By giving in-context learning a ground-truth standard, F-ICL offers a precise, reproducible instrument for tracking algorithmic reasoning as models improve.

4 Methods

4.1 The reference machine F

A program is a length-L string over the five instructions of Table 1, indexed as a base-5 integer rule $\in [0, 5^L)$. The tape has a fixed width (default 128 bits) with the head initialised at the centre; the input x is written so that $x[0]$ occupies the centre cell and x extends rightward. Execution reads the bit under the head, applies the instruction, and updates the program counter; `[` jumps to the instruction past its matching `]` when the current bit is 0, and `]` jumps to the instruction past its matching `[` when the current bit is 1, so `[...]` is a while-bit-set loop. A run *halts* when the program counter advances past the last instruction; moving the head off either tape edge aborts the run (out-of-bounds); a step budget (default 1024) bounds run time; and, for non-halting detection only, a repeated full machine state $(\text{counter}, \text{head}, \text{tape})$ certifies a cycle. Bracket matching is the sole validity test; invalid (mismatched) strings are excluded from the enumeration universe, and the non-halting outcome $\perp$ is reserved for *valid* programs that exhaust the step budget or provably cycle. The output of a halting run is the bit string on the head-excursion window $[\min_{\text{head}}, \max_{\text{head}}]$; the empty string is a valid output, and non-halting runs are assigned the sentinel $\perp$. With an empty input this convention reproduces the blank-tape behaviour exactly, which anchors the conditional pipeline to the unconditional one. A step-by-step trace is given in Appendix D.

4.2 Bounded universal prior and algorithmic probability

A length-L program is encoded in a halt-terminated six-symbol code (five instructions plus a halt symbol, read until halt), so its description length in bits is

$$\ell(L) = \lceil (L{+}1)\, \log_2 6 \rceil, \tag{2}$$

which is constant across the 5^L programs of one length. The bounded universal prior weights a program p by

$$w(p) = 2^{-\ell(|p|)}, \tag{3}$$

restricted to a bounded sub-probability measure by the length budget $L_{\max}$ (the semimeasure deficiency proper comes from non-halting; both deficits are quantified below); shorter programs dominate, with the per-length prior mass $5^L 2^{-\ell(L)}$ falling geometrically in the mean (the ceiling in Eq. (2) makes it non-monotone from length to length; Appendix E) and falling geometrically. Enumerating every program of length $1..L_{\max}$

Table 2: **Serving conventions of the frozen benchmark.** All families use sF optima (Eq. (10); one polarity bit, then the F program) over valid programs only. Induction is served by program-set intersection (Eq. (13)), which is weighted by program length alone (Solomonoff) and therefore serves the *full* length budget exactly, with no truncation; the marginal families use the time-penalised (Levin) weights of Eq. (7), whose halt certificates the enumeration stores. Every induction and continuation task is emitted in both polarities; abduction is a validation family (foil separation), not part of the model score.

Family	Backend	Lengths	Weights	Role	Twins
Induction	program index (joint)	1..13 (never truncated)	$2^{-\ell(L)}$	scored	yes
Continuation	full-budget conditionals	1..13	$2^{-\ell(L)-\delta(t)}$	scored	yes
Abduction	full-budget conditionals	1..13	$2^{-\ell(L)-\delta(t)}$	validation	–

on a fixed ordered set of inputs and recording each program's *behaviour* $b = \big(b(x_1), \ldots, b(x_K)\big)$ yields the map

$$b \longmapsto \textstyle\sum_{p:\,\mathrm{beh}(p)=b} w(p), \tag{4}$$

the total prior mass of each distinct behaviour, the sufficient statistic on which every optimum in this paper is computed. Halt times are coded with the Elias-δ code (worked example in Appendix F), whose bit-length we write $\delta(\cdot)$; with $\beta(n) = \lfloor \log_2 n \rfloor + 1$ the number of bits in n, and Elias-γ bit-length $\gamma(n) = 2\beta(n) - 1$,

$$\gamma(n) = 2\lfloor \log_2 n \rfloor + 1, \qquad \delta(n) = \gamma\big(\beta(n)\big) + \beta(n) - 1 = 2\lfloor \log_2 \beta(n) \rfloor + \beta(n). \tag{5}$$

For each input x and output s the pipeline records both a length-only mass (Solomonoff weighting) and a time-penalised algorithmic probability (Levin weighting), and the cheapest certified description length:

$$m(s \mid x) = \sum_{p:\,p(x)=s} 2^{-\ell(|p|)}, \tag{6}$$

$$\hat{m}(s \mid x) = \sum_{p:\,p(x)=s} 2^{-\ell(|p|)-\delta(t_p)}, \tag{7}$$

$$\hat{K}(s \mid x) = \min_{p:\,p(x)=s} \big[\ell(|p|) + \delta(t_p)\big], \tag{8}$$

where t_p is the halt time of p on x and the witness attaining (8) is stored. The normalised conditional algorithmic probability used by the marginal families is

$$P(s \mid x) = \frac{\hat{m}(s \mid x)}{Z(x)}, \qquad Z(x) = \sum_{s'} \hat{m}(s' \mid x), \tag{9}$$

the sum running over halting outputs. All served optima are computed over valid programs only, with an explicit per-family weighting (Table 2). The induction family uses the length-only weights of Eq. (6) (Solomonoff): its posterior is served by program-*set* intersection (Eq. (13)), where length-only weighting reduces serving to exact index arithmetic; this is what makes the full length budget servable with no truncation anywhere, 0-shot included. The accumulating-evidence families use the time-penalised weights of Eq. (7) (Levin weighting[8, 17]), which the enumeration stores per (input, output). The weighting convention is explicit and fixed per family (Table 2). The cross-check below compares program *counts* $\#\{p : p(x) = s\}$, which are weighting-independent and so certify the machine under either choice. Worked numerical values for Eqs. (2)–(9) are given in Appendices E–I.

4.3 The symmetrised machine sF, twin tasks and the polarity gap

A worked numerical example is given in Appendix J.

An sF program is a polarity bit β followed by an F program p; the machine reads β, runs p on the input x if $\beta = 0$ or on the complemented input $\bar{x}$ if $\beta = 1$, and complements the output in the latter case ($\perp$ maps to itself). Its description length is $1 + \ell(|p|)$ bits (plus the halt-time code where charged), and its conditional algorithmic probability satisfies the exact identity

$$\mathrm{s}P(s \mid x) = \tfrac{1}{2}\big[P(s \mid x) + P(\bar{s} \mid \bar{x})\big], \tag{10}$$

each branch carrying the polarity bit's factor $\frac{1}{2}$. sF is an ordinary prefix machine, so Eq. (10) is a change of reference machine, not an ad-hoc average; it is complement-invariant by construction ($\mathrm{s}P(s \mid x) = \mathrm{s}P(\bar{s} \mid \bar{x})$, and the first-bit law equals $\frac{1}{2}$ exactly), and the certified minimum tightens to $\hat{K}_{\mathsf{sF}}(s \mid x) = \min\big(\hat{K}(s \mid x), \hat{K}(\bar{s} \mid \bar{x})\big) + 1$, never more than one bit above the raw machine's, and often below it. Legitimacy, exact invariance and this tightening are proved in §A; the companion paper[21] builds its estimator on the same machine, so using sF here keeps the series' reference machine uniform.

For few-shot posteriors the polarity bit flips the *whole* task: an sF program βp is consistent with an example (x_i, o_i) iff $p(x_i) = o_i$ when $\beta = 0$, or $p(\bar{x}_i) = \bar{o}_i$ when $\beta = 1$. Both routes are served from the same F enumeration (route $\beta = 1$ performs the identical lookups at the complemented keys and complements the outcome), so the sF posterior costs two F queries, each at weight $\frac{1}{2}$:

$$\mathrm{s}P(s \mid x_q, \mathcal{E}) = \tfrac{1}{2}\,P(s \mid x_q, \mathcal{E}) + \tfrac{1}{2}\,P(\bar{s} \mid \bar{x}_q, \bar{\mathcal{E}}), \qquad \bar{\mathcal{E}} = \{(\bar{x}_i, \bar{o}_i)\}, \tag{11}$$

normalised over outcomes. Under Eq. (11) the posterior for a complemented task is exactly the complemented posterior for the original.

This invariance is enforced, not assumed. Every induction and continuation task is emitted in both polarities (twin ids share the base id), and the build asserts, for every pair: the twin's stored optimum equals the flipped original to 10^{-9}; every stored distribution normalises; and across paired first positions $P(\mathtt{0}) = \frac{1}{2}$ exactly. The evaluation then scores a model on the union of both polarities and reports the polarity gap as a facet,

$$\Delta_{\mathrm{pol}} \;=\; \text{FICL-A}\big(\text{original tasks}\big) \;-\; \text{FICL-A}\big(\text{twin tasks}\big), \tag{12}$$

the score difference between original and twin tasks. For the optimum, and for any model that treats `0` and `1` equivalently, $\Delta_{\mathrm{pol}} = 0$; a deviation measures the model's own polarity preference on an instrument that provably has none.

4.4 GPU enumeration pipeline

The enumeration is implemented in Taichi/CUDA. The unconditional pass streams every length-L program through the machine on a blank tape and reduces, per unique output, the count, halt-time statistics, minimal certified cost and prior masses, using a GPU hash aggregator with an exact minimum-rule tie-break so that the recorded witness is deterministic. The conditional pass performs the same enumeration with the tape preloaded with each input x and aggregates outputs *per input*; it reuses the validated aggregator verbatim, overriding only the tape-initialisation and output-span steps. The reduction is associative, which we exploit for parallelism: the multi-GPU unconditional driver stripes rule batches across devices and merges the partial dictionaries through a single coalescing path that preserves the minimum-rule tie-break, while the conditional driver parallelises over *inputs* (whose per-input HDF5 groups are independent, so multi-device results are stitched by a pure group copy with no array-level merge). A dense-prefix copy moves only the halting rows across the PCIe bus each batch, which is what makes $L = 13$ feasible. A small micro-benchmark detects power-of-two tape-stride partition camping on the target GPU and selects an odd tape padding when needed.

4.5 Budget adequacy, merge and certification

Because a fixed step and tape budget can silently drop programs that halt only after more steps or a wider head excursion, a coverage scan, coupled bit-for-bit to the simulator's semantics and using Brent cycle detection, captures, per input, the exact set of halts the base budget would miss. These "outliers" are

re-run at the full budget on a faithful CPU replica and merged into the base aggregation; per input, the partition closes exactly (base halts + re-run outliers = coverage halts). The pipeline orchestrates coverage → base → merge → copy into a unified result, each stage in its own process so each obtains a clean CUDA context. Certification (gate 2) samples stored rows, decodes the stored witness, re-simulates it on the input with exact CPU semantics and asserts that the halt time and output match the recorded values; the count cross-check (gate 1) enumerates every length-L program with the independent reference implementation and asserts per-(input, output) count equality with the GPU aggregation (verified at $L = 5$, $M = 8$ across all 511 inputs); and the dumped program index (below) is gated against the same reference machine for per-(input, output) *rule-set* identity (gate 3): the exact *set* of program identities behind each output, a strictly stronger check than counts. Exact agreement on these checks is required for a build to be accepted.

4.6 The dumped program joint and exact few-shot serving

The aggregate conditional pass records only per-(input, output) statistics (counts, witnesses, masses), which yield the marginals $P(\cdot \mid x)$ but discard *which* program produced *which* output, so they cannot answer a query conditioned on several different inputs. The dump reuses the *validated* conditional kernels verbatim but keeps the per-halt (rule, output) records and writes the full joint as an inverted index: one HDF5 shard per length (L, M), and within it one group per input x holding the distinct clean outputs (CSR arrays `bit_length`, `out_word_ptr`, `out_words`) and, for each output, the complete *sorted* list of base-5 length-L program identities realising it (CSR arrays `rules_ptr`, `rules`). Sharding by length is essential: a bare rule index is undecodable without its L, and serving weights each length by the prior $2^{-\ell(L)}$ (Eq. (3)). Outputs are keyed by the clean head-excursion window (stray packing bits masked), so the per-output rule *count* equals the marginal file's count exactly, tying the two backends together.

Given examples $\mathcal{E}$ and a query x_q, the exact posterior of Eq. (1) is served per length and summed:

$$P(s \mid x_q, \mathcal{E}) = \frac{\sum_L 2^{-\ell(L)} \left|\{p \in C_L(\mathcal{E}) : p(x_q) = s\}\right|}{\sum_L 2^{-\ell(L)} \left|C_L(\mathcal{E})\right|}, \qquad C_L(\mathcal{E}) = \bigcap_{(x_i, o_i) \in \mathcal{E}^+} \mathrm{rules}_L(x_i, o_i), \tag{13}$$

where $\mathcal{E}^+$ are the positive (halting) examples and $C_L(\mathcal{E})$ is the consistent set at length L, obtained by sorted-array intersection of their rule lists (smallest first). Negative examples (an input on which the program did *not* halt) filter C_L by index membership at that input: a program absent from every output list of x_i is exactly one that does not halt on x_i. The survivors' outcomes on the query are read from the query input's index by a single sorted-array membership pass (equivalently, the survivors can be re-run on the validated machine; the implementation cross-checks the two routes for bit-exact agreement); survivors absent from the query's index contribute to the non-halting outcome. Under the length-only weighting no run-time quantity is needed at serving time, which is what lets every posterior, at every shot count, mix over the *complete* $L \leq 13$ budget. Indices are loaded lazily per (length, input) and cached (example lists as contiguous slices; full query indexes in a bounded pool), so a query touches only the inputs it mentions and the full joint is never materialised. The same intersection gives the Bayesian evidence $\sum_L 2^{-\ell(L)} |C_L(\mathcal{E})|$ (the marginal likelihood). One subtlety: with *no* positive example the consistent set is the complement of the halting sets, which the index does not store; the 0-shot prior is instead served in closed form from the enumeration's per-length *aggregate* marginals: per (input, output) halting counts, plus the valid-program count #valid(L) (a bracket-matching dynamic program) for the non-halting mass, again over the complete budget. The served posterior is bit-identical to the small-program joint across 0–5 examples (Appendix N), and it is this backend that produces the frozen induction optima in the evaluation dataset, mixing over the complete $L \leq 13$ budget with hidden programs drawn from $L = 5$–9. The sF posterior of Eq. (11) is served by running the identical intersection twice, once on the examples as given and once on their complements at the complemented query, at weight $\frac{1}{2}$ each; the index is never duplicated, since complemented inputs are first-class inputs of the same dump.

Two engineering choices make the dump feasible at $L = 13$. A dense-prefix copier moves only the $\sim n_{\text{halts}}$ halting rows across the PCIe bus per batch (rather than the whole batch buffer), and a single vectorised `lexsort` (primary output bits, finest rule) groups equal-output halts and sorts their rule lists in one pass, replacing a per-halt loop.

4.7 Task families

Induction (joint / index): a hidden program is shown by its (input → output) behaviour on several distinct inputs; the model predicts its output on a held-out query. Tasks are curated to be non-degenerate (examples not all identical), learnable (the optimal posterior converges to the truth by the final example) and non-trivial (the zero-example optimum places at most 0.7 on the truth), and each is tagged with its Bayes sample complexity; the full per-example posterior is stored. The task set is *difficulty-balanced*: equal quotas per BSC bucket (Eq. (21)), with the hidden program length spread within each bucket, so free sampling cannot flood the benchmark with tasks that Occam-guessing already solves. **Continuation** (marginals): a program's output for an input is revealed token by token, with the exact optimal next-token distribution over $\{\mathtt{0}, \mathtt{1}, \textsc{end}\}$ stored at each position; the optimum uses the full conditional and remains exact. Because the bounded universal prior concentrates on short, simple outputs, free sampling would starve both the later reveal positions and the high-complexity regime; the task set is therefore balanced on *two* axes: equal quotas per final output length, and, within each length, equal quotas per tercile of the task's conditional codelength $\kappa = -\log_2 P(s \mid x)$ (the optimum's own total log-loss on the reveal, the continuation analogue of BSC; stored with every task). Conditioning task *selection* on $(|s|, \kappa)$ does not affect exactness: each stored distribution remains the exact conditional given (x, prefix). The hard tercile realises outputs that are correct but deliberately *not* low-complexity: the reveal walks the model through atypical branches of the conditional, and the model is scored against the exact conditional distribution at every step, never on reproducing the realised bit. **Abduction** (marginals): a hidden input from a candidate set produces an output revealed token by token, with the exact optimal posterior over inputs computed from the per-input marginals. Targets are sampled in proportion to prior mass, so the optimal predictor's expected log-loss is the information-theoretic floor. Abduction is used to validate the benchmark's discriminative power (Fig. 3b); model scoring uses induction and continuation (Table 2). Every induction and continuation task is emitted in both polarities (Methods 4.3); twin pairs share their curation tags (BSC, κ), and duplicate rejection during sampling treats a task and any task's mirrored cell as the same cell, so the two polarity halves are disjoint and balanced identically.

4.8 Exact predictors for the accumulating-evidence families

Write $\mathcal{A} = \{\mathtt{0}, \mathtt{1}, \textsc{end}\}$, let r be a revealed output prefix of length k, and let $\mathcal{S}_r = \{s : \ r \sqsubseteq s\}$ be the outputs extending r. The optimal next-token predictor for *continuation* reads, from the conditional $P(\cdot \mid x)$ of Eq. (9),

$$q_{\mathtt{0}} = \sum_{\substack{s \in \mathcal{S}_r \\ |s|>k,\ s_k=\mathtt{0}}} P(s \mid x), \quad q_{\mathtt{1}} = \sum_{\substack{s \in \mathcal{S}_r \\ |s|>k,\ s_k=\mathtt{1}}} P(s \mid x), \quad q_{\textsc{end}} = \sum_{\substack{s \in \mathcal{S}_r \\ |s|=k}} P(s \mid x),$$
$$P^{\star}_{\text{cont}}(a \mid x, r) = \frac{q_a}{q_{\mathtt{0}} + q_{\mathtt{1}} + q_{\textsc{end}}}, \quad a \in \mathcal{A} \tag{14}$$

(uniform 1/3 if the denominator is 0); this is exactly $P(\text{next token} \mid x,\ \text{output starts with } r)$. The *pooled* foil replaces $P(\cdot \mid x)$ in Eq. (14) by the input-agnostic $\bar{P}(s) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} P(s \mid x)$; the *uniform* foil is $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.

For *abduction* over a candidate input set $\mathcal{C}$ with uniform prior $\pi(x) = 1/|\mathcal{C}|$, define the prefix mass $\mu(x, r) = \sum_{s \in \mathcal{S}_r} P(s \mid x)$ (so $\mu(x, \varepsilon) = 1$). The optimal posterior over the hidden cause is Bayes' rule,

$$P^{\star}_{\text{abd}}(x \mid r) = \frac{\pi(x)\, \mu(x, r)}{\sum_{x' \in \mathcal{C}} \pi(x')\, \mu(x', r)}. \tag{15}$$

The *prior-only* foil returns π regardless of r; the *greedy-match* foil places all mass on the candidate x maximising the pair $\big(\textsc{agree}(\hat{s}_x, r),\ P(r \mid x)\big)$, where $\hat{s}_x = \arg\max_s P(s \mid x)$ and $\textsc{agree}$ counts matching leading characters, a surface match rather than Bayesian integration.

4.9 Reference predictors for induction

All induction predictors return a distribution over the outcome set $\mathcal{O}$ (distinct outputs plus the non-halting outcome) and are ε-smoothed, $S_\varepsilon(d)(s) \propto d(s) + \varepsilon$ ($\varepsilon = 10^{-9}$, renormalised over $\mathcal{O}$), so log-loss and KL are

finite. Given examples $\mathcal{E}_n = \{(x_{i_1}, o_1), \ldots, (x_{i_n}, o_n)\}$ and query x_q:

$$\begin{aligned}
\text{optimal:} \quad & S_\varepsilon\big(P(\cdot \mid x_q, \mathcal{E}_n)\big) \text{ from Eq. (1)}, \\
\text{uniform:} \quad & d(s) = 1/|\mathcal{O}|, \\
\text{marginal (pattern completion):} \quad & d(s) \propto \big|\{j \le n : o_j = s\}\big| \ \text{ (uniform if } n{=}0), \\
\text{nearest-input (surface analogy):} \quad & d = S_\varepsilon(\mathbf{1}_{o_{j^*}}), \ \ j^* = \arg\min_{j \le n} \big(d_H(x_{i_j}, x_q),\ \big||x_{i_j}| - |x_q|\big|\big),
\end{aligned} \tag{16}$$

with d_H the Hamming distance (length-padded) and nearest-input uniform at $n{=}0$. The marginal foil tracks output frequencies while ignoring which input produced them; nearest-input copies the most input-similar example without any program inference.

Classical nulls on the panel axes. For the model panel we additionally score four evidence-free or statistics-only predictors through the *identical* pipeline as the models (same candidate sets, same restricted and renormalised optima, same JS and anchoring), so they land on the F-ICL-A axis of Fig. 5: (i) *global output frequencies*, one fixed distribution, the average of the exact optima over all tasks at zero examples (induction) and over all tasks and positions (continuation); (ii) the *pooled optimum*, the per-shot (respectively per-position) average of the exact optima, the best predictor that ignores the individual task; (iii, iv) *Krichevsky–Trofimov* add-$\frac{1}{2}$ estimators of order 0 and 1 fitted on the revealed prefix (continuation) or on the concatenated example outputs (induction), the standard minimax sequence predictors over a binary alphabet. Their scores are reported with the panel (implementations released with the code).

4.10 Losses, divergences and the F-ICL score

Worked numerical examples of the divergence and of the anchored score are given in Appendices L and M. For a predictive distribution p and realised outcome y the log-loss is $\mathrm{LL}(p, y) = -\log_2 p(y)$, the optimal floor is $\mathrm{LL}(P^\star, y)$, and the *regret* is the (non-negative) excess $R = \mathrm{LL}(p, y) - \mathrm{LL}(P^\star, y)$. We also report $\mathrm{KL}(p \,\|\, q) = \sum_s p(s) \log_2 \frac{p(s)}{q(s)}$. The primary distribution-comparison metric is the base-2 Jensen–Shannon divergence,

$$\mathrm{JS}(p \,\|\, q) = \tfrac{1}{2}\mathrm{KL}(p \,\|\, m) + \tfrac{1}{2}\mathrm{KL}(q \,\|\, m), \qquad m = \tfrac{1}{2}(p + q), \tag{17}$$

which is symmetric, finite without smoothing, and bounded in $[0, 1]$. Per shot/position n, every reported curve is the mean over tasks of the corresponding quantity.

For the headline score, each evaluated family f (here induction and continuation) contributes its mean JS to the optimum over all of its (task, position) instances I_f,

$$\bar{D}_f = \frac{1}{|I_f|} \sum_{i \in I_f} \mathrm{JS}\big(P_i^\star \,\big\|\, P_i^{\text{model}}\big), \tag{18}$$

and the F-ICL score *macro-averages* these over the family set $\mathcal{F}$ (every family weighted equally, regardless of its task count):

$$\mathrm{FICL} = 1 - \frac{1}{|\mathcal{F}|} \sum_{f \in \mathcal{F}} \bar{D}_f = \frac{1}{|\mathcal{F}|} \sum_{f \in \mathcal{F}} \big(1 - \bar{D}_f\big) \ \in [0, 1], \tag{19}$$

so FICL $= 1$ iff the model matches the optimum on every family, and the score equals the mean of the per-family facet axes $1 - \bar{D}_f$.

Keystroke-hedge thresholds. Emitting uniform random keystrokes over $\mathcal{A}$ gives a whole output s probability $c(s) = (1/3)^{|s|+1}$. Per family we report a *hedged* reference $\mathrm{JS}(P^\star, U)$ (against the uniform next-token / candidate distribution U) and a *committed* reference $\sum_a U(a)\, \mathrm{JS}(P^\star, \mathbf{1}_a)$; for induction over a candidate set the candidate distribution is c renormalised over $\mathcal{C}$. Both are macro-averaged exactly as in Eq. (19) (1− mean hedged / committed JS) and reported beside the score, never blended in.

The normalised scale. The raw score of Eq. (19) must be read against its keystroke-hedge threshold, not against 0: Jensen–Shannon divergence is bounded, and on a three-symbol alphabet the uniform hedge is never more than $\mathrm{JS}(\mathbf{1}_a \,\|\, U) = \frac{1}{2}\log_2\frac{3}{2} + \frac{1}{6} \approx 0.459$ bits from *any* distribution, so $1 - \bar{D}$ compresses every predictor, informed or not, into a narrow band near the top (the keystroke reference typically scores

0.85). A bounded, task-intrinsic scale follows from the same geometry: $\mathrm{JS}(P^\star \,\|\, \cdot)$ is convex on the simplex, so its maximum is attained at a vertex: the *worst possible predictor* for an instance is the point mass on the optimum's least likely symbol (or candidate), and its divergence $D_i^{\mathrm{w}} = \max_a \mathrm{JS}(P_i^\star \,\|\, \mathbf{1}_a)$ is computable exactly from the stored optimum. The headline scale anchors *three* exactly computable points (the worst predictor, the keystroke reference, and Bayes) by the piecewise-linear map

$$\text{FICL-A}_f = \begin{cases} \frac{1}{2} + \frac{1}{2}\,\dfrac{\bar{D}_f^U - \bar{D}_f}{\bar{D}_f^U} & \bar{D}_f \le \bar{D}_f^U \quad \text{(better than random)}, \\ \frac{1}{2}\,\dfrac{\bar{D}_f^{\mathrm{w}} - \bar{D}_f}{\bar{D}_f^{\mathrm{w}} - \bar{D}_f^U} & \bar{D}_f > \bar{D}_f^U \quad \text{(worse than random)}, \end{cases} \qquad \text{FICL-A} = \frac{1}{|\mathcal{F}|}\sum_{f\in\mathcal{F}} \text{FICL-A}_f, \quad (20)$$

with $\bar{D}_f^U$ the family's mean hedged keystroke JS and $\bar{D}_f^{\mathrm{w}}$ its mean worst-case JS. The scale is bounded in $[0,1]$ by the convexity argument above, and its three anchors are fixed on *every* dataset: the worst predictor scores 0, the keystroke reference scores exactly $\frac{1}{2}$, and Bayes scores 1. A single number therefore carries both facts a reader needs: which side of the keystroke (print-prior) reference the model sits on (the sign of FICL-A $-\frac{1}{2}$), and how much of the corresponding gap it closes. Above $\frac{1}{2}$ this is the fraction of the hedge-to-Bayes gap; below, the position between the worst predictor and the hedge. Distances are linear within each regime (the map has a kink at $\frac{1}{2}$ by construction). Two diagnostics are retained alongside: the worst-anchored $1 - \bar{D}_f/\bar{D}_f^{\mathrm{w}}$ with the hedge reported as a floating random marker, and the signed $\text{skill}_f = 1 - \bar{D}_f/\bar{D}_f^U$ (unbounded below; used for its sign).

Difficulty and efficiency. The Bayes sample complexity of an induction task is the first shot at which the optimal MAP equals the truth,

$$\mathrm{BSC} = \min\left\{n \in \{0,\dots,N\} :\ \arg\max_s P^{\star(n)}(s) = \text{truth}\right\} \quad (= N \text{ if never}), \tag{21}$$

which makes difficulty explicit. The difficulty-robust *BSC-macro* score buckets tasks by BSC (into $\{0\}$, $\{1\}$, $\{2,3\}$, $\{4+\}$), forms a per-bucket skill $1 - \overline{\mathrm{JS}}_{\text{bucket}}$, and averages the buckets with equal weight. These scoring buckets are deliberately coarser than the sampling strata: the frozen induction set is balanced 120/120/120/120 over BSC $\in \{0\},\{1\},\{2\},\{3+\}$, whereas the score pools $\{2,3\}$ so that each bucket retains enough tasks for a stable per-bucket mean. The *sample-complexity efficiency* compares the examples a model needs to the Bayes minimum: writing n^* for the first shot at which the model solves the task,

$$e = \begin{cases} 1 & n^* = 0 \\ 0 & \text{never solved} \\ \min\big(1,\ \mathrm{BSC}/n^*\big) & \text{otherwise}, \end{cases} \qquad \text{SC-eff} = \text{median}_{\text{tasks}}\, e \tag{22}$$

(1 = as few examples as Bayes). Single-answer induction axes (model answer a, optimal MAP s^*, posterior $P^{\star(n)}$, each averaged over (task, shot)) are accuracy $\mathbf{1}[a{=}\text{truth}]$, agreement $\mathbf{1}[a{=}s^*]$, and Bayes-alignment $P^{\star(n)}(a)$.

The truth–complexity dissociation. The true output is, by design, *not* systematically the lowest-complexity consistent answer: whenever BSC $> n$, the posterior MAP s^* at shot n (the Occam-favoured, smallest-description candidate) differs from the truth, and a model that answers s^* is wrong on accuracy while performing exactly the right inference. The metric set is built around this dissociation rather than suffering from it: accuracy is truth-referenced, agreement and alignment are Bayes-referenced, and their divergence on high-BSC tasks *measures* how a model resolves the tension between simplicity and evidence, which is the object of study rather than an artefact. Two safeguards keep the comparison fair. First, the curation gates guarantee the tension resolves: by the final shot the posterior MAP equals the truth, so final-shot accuracy is a fair ask of any Bayes-rational predictor. Second, the distributional induction metric scores the model over a candidate set consisting of the optimum's top $k{-}1$ outputs *plus the true output* ($k{=}6$): without the truth's inclusion the JS facet would be computed only over the Occam-favoured candidates and would be blind to the probability mass a model assigns to a high-complexity truth precisely on the tasks where the dissociation bites. The set is rebuilt at each shot from that shot's posterior, so it contracts as evidence eliminates programs: it is the full $k{=}6$ at $n{=}0$, but a single candidate in 63% of (task, shot) cells overall and in 93% by the final shot. On those singleton cells the divergence is identically zero

for the model, the keystroke reference and the worst-case predictor alike, so they dilute all three anchors equally and leave the ratio the anchored score is built on intact; the induction divergence facet accordingly carries its signal on the cells where the posterior has not yet collapsed.

The facet radar plots all axes on $[0, 1]$ (higher is better); its area depends on the arbitrary axis order and is *not* a metric; ranking uses the scalar anchored FICL score. Worked numerical values for Eqs. (14)–(19) and for Eq. (22) appear in Appendix I onward; Appendix M also locates that worked run on the anchored scale of Eq. (20).

4.11 Language-model evaluation harness

The exact prompts every score is read from are reproduced verbatim in Appendix C. Models are queried through an OpenAI-compatible endpoint, so the same code drives local servers (`llama.cpp`, vLLM) and hosted APIs; requests are submitted concurrently for throughput on batching servers. Few-shot prompts present examples as `input -> output` lines and leave the query open. For the accuracy metric the answer is constrained by a context-free grammar to a non-empty bit string, which yields clean tokens and bypasses the "thinking" preambles of reasoning models; the grammar constrains *generation only* and does not affect the logprob-based metrics. The continuation metric reads the model's next-token log-probabilities over $\{0, 1, \text{END}\}$ and compares them to the stored optimum by Jensen–Shannon. The distributional induction metric teacher-force-scores the optimum's candidate outputs by sequence log-probability (via completions-echo where available, or via bit-by-bit chat log-probabilities) to form a model distribution over candidates, again scored by Jensen–Shannon. An endpoint probe reports which log-probability and grammar features a server supports before a run. A prompt-ablation protocol re-runs a fixed model subset under systematic variations of the prompt template (instruction wording, example formatting, query framing) to bound the score's sensitivity to presentation choices; ablation runs are reported with the panel artefacts.

4.12 Model panel and controlled comparisons

Because F-ICL scores a model against a fixed, exact optimum, every member of an evaluation panel is measured against the same target, and the frozen optima, prompt format, decoding settings and task sample are held constant so that only the model varies. We therefore choose the panel not to maximise leaderboard rank but to isolate specific factors, organised as the four controlled comparisons below. Whether F-ICL tracks general capability or measures a distinct axis could be settled by correlating these rows against published scores on general (e.g. MMLU-Pro, GPQA) and reasoning-oriented (e.g. AIME, ARC-AGI) benchmarks; that comparison is not attempted here and is left to future work. All models are served through the OpenAI-compatible harness; locally served models are run as quantised weights on commodity dual-GPU hardware, with the quantisation level recorded against every result. The panel reported here comprises **105** runs: **30** open models through an NVIDIA NIM endpoint, a nine-model dated OpenAI ladder (GPT-4o May 2024 through GPT-5.6), Gemini and gemma-4 rows through Vertex and the Gemini developer API, two Grok rows through the xAI API, three Claude rows through the Anthropic Messages API (Fable 5, Opus 4.8, and the Fable-with-Opus-fallback stack), and **56** local `llama.cpp` GGUF builds, covering the families GPT, Gemini, Grok, Claude, gpt-oss, phi-4, Qwen3.5, Qwen3.6, Qwen3-30B, Ministral-3, gemma-4, Nemotron-3, DeepSeek-V4, Mistral, Llama-3, MiniMax and a tail of small entries (GLM, Step ×2, gemma-2, Solar, Sarvam, Laguna, Inkling, Dracarys, Qwen3-Next, Mixtral), and instantiating all four controlled comparisons below. The complete run-level registry (every model, serving route, precision, usable readouts, billed cost, reasoning-effort setting, and audit notes) is provided in the Supplementary Information (§B). The 17 metered frontier rows (four providers, including the nine-model generational ladder and the frontier reasoning systems) were billed at $76.70 in total (per-row costs in the registry), so the benchmark is reproducible on a modest budget; the NVIDIA NIM rows were served under a developer account without per-row metering, and the local panel costs only electricity.

API serving routes. Proprietary rows are measured over the exact route that exposes the required readout, established by per-model probes (July 2026; scripts in the repository): OpenAI GPT-4-era models through Chat Completions with `logprobs`; GPT-5.1 and later through the Responses API with reasoning effort `none`; the original GPT-5 (accuracy only) through the Responses API at effort `minimal`; Gemini

2.5 through the native Vertex `generateContent` API with `response_logprobs` (the sole Google route that returns distributions); Gemini 3.x (accuracy only) through the Gemini developer API or Vertex; Grok (accuracy only) through the direct xAI API (chat route for the non-reasoning 4.20 row; the Responses route at low reasoning effort for the 4.5 row; both routes silently ignore log-probability requests); Claude (accuracy only) through the Anthropic Messages API. The `temperature` parameter is rejected on current Claude models ("deprecated for this model", HTTP 400), thinking is bounded with `effort: low` on Fable 5, and the fallback row uses the server-side refusal-fallback beta with Opus 4.8 as the substitute. Every row records its route, and per-call disk caches make each run resumable without re-billing.

Machine dependence, and what invariance does not buy here. Algorithmic probability is invariant across universal reference machines only up to an additive constant [9]: for machines U, V there is c_{UV} with $|K_U(s) - K_V(s)| \leq c_{UV}$ for all s. Which machine to choose is a substantive and much-discussed problem [3], and the freedom is not innocuous: for any computable predictor there is a universal prior making it optimal [4]. That guarantee is asymptotic in description length, and it is worth being explicit that our budget is far below the regime where it bites. At $L_{\max} = 13$ a program costs $\lceil 14 \log_2 6 \rceil = 37$ bits, so every program description in this paper costs at most 37 bits (38 on sF, which prepends a polarity bit; the Levin-weighted families add an Elias-δ halt-time code, at most 17 further bits at the 1024-step budget). Those totals are the same order as a plausible translation constant between F and another machine. The invariance theorem therefore provides no protection at this scale: the object we compute is a specific bounded mixture over a specific finite machine, not an approximation to any machine-independent quantity, and the same is true of every bounded-enumeration method in this literature. Two consequences are worth stating plainly. First, absolute F-ICL-A values are meaningful only relative to the conventions of Table 2. Second, the empirical claims that should be trusted furthest are the ones that do not depend on the target's identity at all: the non-monotone solved-set instability, the polarity and complement-equivariance readouts, the post-training ordering, and the serving results of §2.7.

One piece of external evidence bounds how idiosyncratic the machine is. The Coding Theorem Method (CTM) estimates algorithmic probability from output *frequency* over an exhaustively enumerated space of $(n, 2)$ Turing machines [1, 2]: a different machine family, a different enumeration, and a frequency-based rather than a length-based estimator. Over all 4,095 complement classes of binary strings up to length 12, the certified description lengths of sF agree with published CTM values at Pearson 0.91–0.92 (Spearman 0.84–0.86), against 0.79–0.81 (Spearman 0.65–0.71) for the unsymmetrised machine [21], where that comparison is derived in full; it is posted as a preprint alongside this paper, so the agreement can be checked independently of anything reported here. Two things follow. The complexity *ordering* our target is built on is not an artefact of one instruction set, since two independently constructed machine families rank short strings alike; and the polarity symmetrisation is externally validated rather than merely convenient, because it closes most of the gap to the other family.

What that agreement does *not* do is rescore this panel. CTM is a blank-tape construction and no published conditional CTM table exists, so there is no CTM-derived posterior $P(s \mid x, \text{examples})$ against which to score models.

Read together with the budget control of §2.5, however, the two results bracket the objection from both sides. The CTM comparison varies the *machine* and holds the object fixed: two independently constructed families, one length-based and one frequency-based, rank short strings alike. The budget control varies the *object* and holds the machine fixed: re-serving the conditional posteriors from $L \leq 6$ through the complete $L \leq 13$ store, with a change of weighting convention on top, moves the per-run ordering by a rank correlation of 0.975 and leaves the headline count at 77 of 80. So the complexity ordering our target is built on is robust across machine families, and the conditional posterior built on that ordering is robust to how the mixture is formed. What neither result covers, and what we therefore do not claim, is the conjunction: a conditional posterior served from a *different* family. That is the natural next test, and it is an enumeration project rather than a rescore: the cached model distributions would carry over unchanged, but the target would have to be built from scratch.

A further caution for that test: an instruction-symbol *permutation* is not a change of machine at all under length-only weighting. If π relabels the instruction alphabet, then $F_\pi(p) = F(\pi(p))$ and $p \mapsto \pi(p)$ is a length-preserving bijection of the valid-program set whenever π respects the bracket structure, so the induced output measure is *identical*: $m_{F_\pi} = m_F$. Relabelling tests, and reflections implemented as a swap

of the two move instructions, are therefore vacuous by construction. A meaningful variation has to change the input/output convention, the instruction set, or the family.

Relatedly, the enumerated universe is Turing-complete only as an idealisation: with a 128-cell tape and a 1024-step budget the objects enumerated are a family of finite automata, and universality is a property of the unbounded machine that motivates the instruction set, not of the instrument.

Where the missing mass goes. Solomonoff's mixture is a *semi*measure because programs may fail to halt; a bounded enumeration adds further deficits, and all of them are absorbed into the normaliser Z unless stated. For the frozen datasets the sources are as follows. Non-halting within the step budget is the only one present in the unbounded idealisation and the largest here, carrying 6.2% of the valid-program prior mass over $L \leq 13$ and rising monotonically with length from 0% at $L = 1$ to 24% at $L = 13$. Then programs beyond $L_{\max}$ (which carry 1.4% of that mass), and the rounding in the length code $\ell(L) = \lceil (L{+}1) \log_2 6 \rceil$, which also makes the per-length mass non-monotone: the L=14 shell is heavier than the L=13 shell, so the budget cuts immediately before a heavier one. An exactly normalised code $6^{-(L+1)}$ removes the non-monotonicity and is preferable in future builds.

The two scored families handle the deficit differently, which we state because it affects what each measures. Continuation retains the non-halting mass by mapping it onto the terminator, so its target is a distribution over $\{\texttt{0}, \texttt{1}, \textsc{end}\}$ with no deficit. Induction strips the non-halting outcome and the storage-tail bucket from the candidate set and renormalises, so its target is conditioned on the program halting within budget. Induction scores are therefore halting-conditioned and support-truncated (top-24 storage plus a residual bucket), and the "zero posterior mass" errors of §2.4 are zero under that truncated support rather than under the unbounded mixture.

Statistical reporting. All hypothesis tests are two-sided. Local decoding is deterministic (temperature 0) so local runs are single-seed; API rows are only as reproducible as their serving stack (re-running Claude Fable 5 on byte-identical prompts reproduces 86% of answers, §2.7), and run-to-run variation is instead characterised directly through the quantisation replicas and the paired serving comparison. Analyses whose runs share weights (quantisation ladders) are clustered by base model before inference: the polarity test is a t-test over the 46 base-model means, not over the 81 runs (the open-panel subset reported alongside uses its own 37 means). Confidence intervals are 95% nonparametric task bootstraps. Given the panel's exploratory breadth we designate two findings as confirmatory: the sub-reference headline (78 of 81 runs and 45 of 46 base models; sign test $p < 10^{-9}$) and the post-training fidelity decline (replicated in two independent families, with drops an order of magnitude above the bootstrap CI); all other contrasts, including the polarity-twin bias, are reported descriptively with exact n and unadjusted p-values and should be read as such.

4.12.1 (a) Same model, varying quantisation

For a fixed checkpoint we sweep the quantisation level (e.g. BF16, 8-, 4-, 3- and 2-BIT) and measure how each F-ICL facet moves. This serves two ends. Scientifically, because the continuation and induction-JS metrics are computed from token log-probabilities (Eqs. (14), (17)), they are sensitive to the logit perturbation quantisation introduces; we expect posterior fidelity (JS to the optimum) to degrade *before* single-answer accuracy does, a dissociation the facets can resolve; the panel outcome is reported in §2.4. Practically, the sweep identifies the smallest precision at which a model's score is statistically indistinguishable from its full-precision score, fixing the most economical serving setting for the rest of the panel and bounding the bias that low-bit quantisation injects into the headline number.

4.12.2 (b) Same family, increasing parameters

Within one model family we hold the training recipe fixed and increase the parameter count (a small dense member, a mid-size dense member, and the largest member that fits, including mixture-of-experts (MoE) variants for which the total and active parameter counts differ). This isolates the effect of scale on algorithmic in-context reasoning: whether the per-family mean JS to the optimum (Eq. (18)) falls monotonically, whether the Bayes sample complexity (Eq. (21)) the model can absorb rises, and whether sample-complexity efficiency

(Eq. (22)) approaches one. For MoE members we report both total and active parameters, since the two scale the relevant computation differently.

4.12.3 (c) Same model: base, instruction- and reasoning-tuned

For a family that releases a base (pre-trained-only) checkpoint alongside instruction-tuned and explicit-reasoning ("thinking") variants, we compare the three at matched scale. The base model is a pure next-token predictor and gives the cleanest test of whether a model's *raw* continuation prior is aligned with the algorithmic prior, a comparison the log-probability metrics support directly, even though a base model may not follow the few-shot answer format (the grammar-constrained answer call handles formatting; the JS metrics do not depend on it). The reasoning variant tests the opposite hypothesis: explicit deliberation may raise induction accuracy and lower sample complexity while *worsening* token-level posterior fidelity, if it sharpens the model's distributions past the optimum's own uncertainty. The harness exposes the reasoning mode as a toggle, so the three are evaluated through an otherwise identical pipeline.

4.12.4 (d) Different families

Holding scale roughly constant, we compare models from distinct developers and pre-training pipelines (for example Qwen, Gemma, GPT-OSS, Nemotron, GLM, DeepSeek, Mistral and Llama). Because the optimum is recipe-independent, a difference in the F-ICL score across families at matched size reflects a difference in what pre-training and alignment instil about algorithmic structure, not a difference in the yardstick. This is the comparison behind the family effect of §2.6. We do not report it here as a separate matched-size ladder: the family effect it targets is reported instead in §2.6, where family explains distributional position ($F = 3.93$, $p = 0.0002$). A matched-size cross-family ladder remains a natural extension.

Controls. Across all four comparisons the frozen optima, the curated task set, the prompt templates, the decoding configuration (grammar-constrained for accuracy, log-probabilities for the divergence metrics) and the random seeds are fixed; only the factor under study varies. Confidence intervals are obtained by resampling tasks (decoding is deterministic throughout), so a reported gap between two panel members can be read as significant or not.

4.13 Reproducibility and the precomputed dataset

The exact optima are computed once and frozen into a single self-contained JSON dataset (curated induction tasks with full per-shot posteriors, and continuation tasks with per-position optimal next-token distributions), so model evaluation requires neither the large enumeration dumps nor the joint computation at run time. All randomness (task sampling, input orderings, candidate sets) is seeded. The released dataset is the one every panel row is scored on. An *extended* build is also released, used for the validation figures and available for future work: it holds 480 induction tasks (240 curated, each with its polarity twin; 8 shots; hidden programs of length $L = 5$–9, difficulty-balanced into equal Bayes-sample-complexity quotas) and 300 continuation tasks (150 length-×-complexity-balanced pairs, each with its twin; up to nine revealed positions, one per output bit plus the terminator), built with seed 0; its induction optima mix over the complete $L \leq 13$ budget (Table 2). The model panel of §2.4 was evaluated on the *pipeline-scale build*, which shares the continuation family byte-for-byte (so every cached continuation call transfers) and serves a smaller induction family of 120 tasks (60 curated + twins) from the exact small-program joint (hidden programs $L \leq 6$, mixture over $L \leq 6$); §2.4 verifies exactly that the two instruments test models equivalently on these tasks (optimum divergence 0.028 bits, run-level rank correlation 0.975, headline unchanged). The validation figures use 300 tasks (Fig. 2) and 500 tasks per family (Fig. 3), seed 0; their generating script is released with the code. Local software: Python 3.12.12, `pycdm` 0.1.0, NumPy 2.4.6, SciPy 1.17.1, Matplotlib 3.10.9, scikit-learn 1.9.0 and h5py 3.16.0; API clients `openai` 2.41.0 and `anthropic` 0.120.0. The enumeration and the conditional dumps were produced with Taichi 1.7.4 on the CUDA backend, on a single NVIDIA GeForce RTX 4090 Laptop GPU (16 GiB) with driver 561.09; local model serving used `llama.cpp` on commodity dual-GPU hardware. Every stochastic step in the build (task sampling, input orderings, candidate-set selection) uses seed 0, and each panel row is a single run: local decoding is deterministic at temperature 0, while API rows are only as reproducible as their serving stack, which we measure rather than assume (§2.7).

5 Data availability

The frozen F-ICL benchmark dataset and the precomputed reference-predictor results that support the findings of this study are archived at `10.5281/zenodo.0000000` (final DOI to be disclosed) and are also released with the code (below). The large intermediate enumeration artefacts can be regenerated from the pipeline.

6 Code availability

The F-ICL benchmark, the reference machine and predictors, the GPU enumeration pipeline, the certification tools and the evaluation harness are released under the MIT licence at the release tagged for this manuscript; the repository URL is to be disclosed.

7 Acknowledgements

The authors gratefully acknowledge the support of Oxford Immune Algorithmics and of King's College London.

8 Author contributions

H.Z. and L.O. jointly carried out all of the following roles: Conceptualization; Methodology; Software (reference machine, enumeration pipeline, evaluation harness, analysis tools); Validation (certification gates, degeneracy audits); Formal analysis; Investigation (model panel and serving routes); Resources; Data curation (frozen benchmark and released artefacts); Writing, original draft; Writing, review and editing; Visualization; Supervision; Funding acquisition. Both authors contributed equally to this work.

9 Competing interests

The authors declare no competing interests.

References

[1] Delahaye, J.-P. & Zenil, H. Numerical evaluation of algorithmic complexity for short strings: A glance into the innermost structure of randomness. *Applied Mathematics and Computation* **219**(1), 63–77 (2012). doi: `10.1016/j.amc.2011.10.006`.

[2] Soler-Toscano, F., Zenil, H., Delahaye, J.-P. & Gauvrit, N. Calculating Kolmogorov complexity from the output frequency distributions of small Turing machines. *PLoS ONE* **9**(5), e96223 (2014). doi: `10.1371/journal.pone.0096223`.

[3] Sterkenburg, T. F. *Universal Prediction: A Philosophical Investigation*. PhD dissertation, University of Groningen, Groningen (2018). ISBN: 978-94-034-0319-9.

[4] Leike, J. & Hutter, M. Bad universal priors and notions of optimality. In *Proceedings of the 28th Conference on Learning Theory*, Proceedings of Machine Learning Research **40**, 1244–1259 (2015).

[5] Bohm, C. On a family of Turing machines and the related programming language. *ICC Bulletin* **3**, 185–194 (1964).

[6] Brown, T. B. *et al.* Language models are few-shot learners. *Advances in Neural Information Processing Systems* **33**, 1877–1901 (2020).

[7] Solomonoff, R. J. A formal theory of inductive inference. Parts I and II. *Information and Control* **7**(1), 1–22, and **7**(2), 224–254 (1964). doi: 10.1016/S0019-9958(64)90223-2; 10.1016/S0019-9958(64)90131-7.

[8] Levin, L. A. Universal sequential search problems. *Problems of Information Transmission* **9**(3), 265–266 (1973).

[9] Li, M. & Vitanyi, P. M. B. *An Introduction to Kolmogorov Complexity and Its Applications*, 4th ed. Springer, Cham (2019). doi: 10.1007/978-3-030-11298-1.

[10] Hutter, M. *Universal Artificial Intelligence: Sequential Decisions Based on Algorithmic Probability*. Springer, Berlin and Heidelberg (2005). doi: 10.1007/b138233.

[11] Xie, S. M., Raghunathan, A., Liang, P. & Ma, T. An explanation of in-context learning as implicit Bayesian inference. In *International Conference on Learning Representations* (2022). https://openreview.net/forum?id=RdJVFCHjUMI.

[12] Muller, S., Hollmann, N., Arango, S. P., Grabocka, J. & Hutter, F. Transformers can do Bayesian inference. In *International Conference on Learning Representations* (2022). https://openreview.net/forum?id=KSugKcbNf9.

[13] Garg, S., Tsipras, D., Liang, P. & Valiant, G. What can transformers learn in-context? A case study of simple function classes. *Advances in Neural Information Processing Systems* **35**, 30583–30598 (2022). doi: 10.52202/068431-2217.

[14] Deletang, G. *et al.* Language modeling is compression. In *International Conference on Learning Representations* (2024). https://openreview.net/forum?id=jznbgiynus

[15] Chollet, F. On the measure of intelligence. *arXiv preprint arXiv:1911.01547* (2019). https://arxiv.org/abs/1911.01547.

[16] Solomonoff, R. J. Complexity-based induction systems: Comparisons and convergence theorems. *IEEE Transactions on Information Theory* **24**(4), 422–432 (1978). doi: 10.1109/TIT.1978.1055913.

[17] Schmidhuber, J. The Speed Prior: A new simplicity measure yielding near-optimal computable predictions. In *Computational Learning Theory*, Lecture Notes in Computer Science **2375**, 216–228 (2002). doi: 10.1007/3-540-45435-7_15.

[18] Zenil, H., Hernandez-Orozco, S., Kiani, N. A., Soler-Toscano, F., Rueda-Toicen, A. & Tegner, J. A decomposition method for global evaluation of Shannon entropy and local estimations of algorithmic complexity. *Entropy* **20**(8), 605 (2018). doi: 10.3390/e20080605.

[19] Zenil, H. & Ozelim, L. Finite reference-machine-grounded algorithmic complexity estimates: Witness and frequency measures on the F machine. Unpublished manuscript (2026).

[20] Hernandez-Espinosa, A., Ozelim, L., Abrahao, F. S. & Zenil, H. SuperARC: A test for artificial superintelligence based on compressed modelling, recursive prediction and problem complexity. *Nature Communications* **17**, 4885 (2026). doi: 10.1038/s41467-026-73289-5.

[21] Zenil, H. & Ozelim, L. The Chain Decomposition Method: A differentiable, multidimensional Solomonoff estimator of Kolmogorov complexity on the real-valued hypercube. Unpublished manuscript (2026).

[22] Min, S., Lyu, X., Holtzman, A., Artetxe, M., Lewis, M., Hajishirzi, H. & Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 11048–11064 (2022). doi: 10.18653/v1/2022.emnlp-main.759.

[23] Chan, S. C. Y., Santoro, A., Lampinen, A. K., Wang, J. X., Singh, A., Richemond, P. H., McClelland, J. L. & Hill, F. Data distributional properties drive emergent in-context learning in transformers. *Advances in Neural Information Processing Systems* **35**, 18878–18891 (2022). doi: 10.52202/068431-1371.

[24] Olsson, C. *et al.* In-context learning and induction heads. *Transformer Circuits Thread* (2022); *arXiv preprint arXiv:2209.11895*. https://arxiv.org/abs/2209.11895.

[25] Akyurek, E., Schuurmans, D., Andreas, J., Ma, T. & Zhou, D. What learning algorithm is in-context learning? Investigations with linear models. In *International Conference on Learning Representations* (2023). https://openreview.net/forum?id=0g0X4H8yN4I.

[26] von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A. & Vladymyrov, M. Transformers learn in-context by gradient descent. In *Proceedings of the 40th International Conference on Machine Learning*, Proceedings of Machine Learning Research **202**, 35151–35174 (2023).

[27] Mahoney, M. V. Text compression as a test for artificial intelligence. Doctoral Consortium abstract. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Annual Conference on Innovative Applications of Artificial Intelligence*, p. 970. AAAI Press, Menlo Park, CA (1999).

[28] Huang, Y., Zhang, J., Shan, Z. & He, J. Compression represents intelligence linearly. In *Conference on Language Modeling* (2024). https://openreview.net/forum?id=SHMj84U5SH

[29] Velickovic, P. & Blundell, C. Neural algorithmic reasoning. *Patterns* **2**(7), 100273 (2021). doi: 10.1016/j.patter.2021.100273.

[30] Velickovic, P., Badia, A. P., Budden, D., Pascanu, R., Banino, A., Dashevskiy, M., Hadsell, R. & Blundell, C. The CLRS algorithmic reasoning benchmark. In *Proceedings of the 39th International Conference on Machine Learning*, Proceedings of Machine Learning Research **162**, 22084–22102 (2022).

[31] Grau-Moya, J. *et al.* Learning universal predictors. In *Proceedings of the 41st International Conference on Machine Learning*, Proceedings of Machine Learning Research **235**, 16178–16205 (2024).

[32] Graves, A., Wayne, G. & Danihelka, I. Neural Turing machines. *arXiv preprint arXiv:1410.5401* (2014). https://arxiv.org/abs/1410.5401.

[33] Ellis, K., Wong, C., Nye, M., Sabl'e-Meyer, M., Morales, L., Hewitt, L., Cary, L., Solar-Lezama, A. & Tenenbaum, J. B. DreamCoder: Bootstrapping inductive program synthesis with wake–sleep library learning. In *Proceedings of the 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation*, 835–850 (2021). doi: 10.1145/3453483.3454080.

[34] Anil, C. *et al.* Exploring length generalization in large language models. *Advances in Neural Information Processing Systems* **35**, 38546–38556 (2022). doi: 10.52202/068431-2793.

[35] Zhou, H., Bradley, A., Littwin, E., Razin, N., Saremi, O., Susskind, J. M., Bengio, S. & Nakkiran, P. What algorithms can transformers learn? A study in length generalization. In *International Conference on Learning Representations* (2024). https://openreview.net/forum?id=AssIuHnmHX

[36] Guo, C., Pleiss, G., Sun, Y. & Weinberger, K. Q. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, Proceedings of Machine Learning Research **70**, 1321–1330 (2017).

[37] Pimentel, T. & Meister, C. How to compute the probability of a word. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 18358–18375 (2024). doi: 10.18653/v1/2024.emnlp-main.1020.

[38] Oh, B.-D. & Schuler, W. Leading whitespaces of language models' subword vocabulary pose a confound for calculating word probabilities. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 3464–3472 (2024). doi: 10.18653/v1/2024.emnlp-main.202.

Supplementary Information

Supplementary sections D–N are worked numerical examples.

A The symmetrised machine: legitimacy, invariance, tightening

Write F for the reference machine, p for one of its programs, $\ell(p)$ for its code length, and $\bar{\cdot}$ for bitwise complement. Define sF as in the main text: read one polarity bit β, run F, and complement the output iff $\beta = 1$. Its programs are the strings βp, so $\ell(\beta p) = \ell(p) + 1$ and the program set is F's doubled. Three properties hold exactly rather than asymptotically.

(i) Legitimacy. sF is an ordinary prefix machine and its prior is a bounded sub-probability measure with the same total mass as F's:

$$\sum_{\beta\in\{0,1\}} \sum_{p} 2^{-\ell(p)-1} \;=\; 2 \cdot \tfrac{1}{2} \sum_{p} 2^{-\ell(p)} \;=\; \sum_{p} 2^{-\ell(p)}. \tag{A.1}$$

The polarity bit is paid for out of the doubled population, so nothing is created or destroyed. Consequently every quantity we report on sF is an estimator *computed on a machine*, not a post-hoc average of two runs of F, which matters because a post-hoc average is not the algorithmic probability of anything.

(ii) Exact invariance. For any input x and output s, grouping the sF programs by their polarity bit gives

$$\mathrm{s}P(s \mid x) \;=\; \tfrac{1}{2}\big[P(s \mid x) + P(\bar{s} \mid \bar{x})\big], \tag{A.2}$$

since βp maps $x \mapsto s$ exactly when p maps $x \mapsto s$ ($\beta = 0$) or p maps $\bar{x} \mapsto \bar{s}$ ($\beta = 1$). Substituting $(\bar{s}, \bar{x})$ for (s, x) in (A.2) exchanges the two terms and leaves the sum unchanged, so

$$\mathrm{s}P(s \mid x) \;=\; \mathrm{s}P(\bar{s} \mid \bar{x}) \quad \text{identically.} \tag{A.3}$$

The identity is a symmetry of the machine, not a property of any dataset, and it holds cell by cell rather than on average. Reversal invariance is inherited from F unchanged, since the polarity bit acts only on output polarity. The first-position law follows: summing (A.2) over the outputs beginning with `1` and over their complements beginning with `0` gives equal totals, so $P(\texttt{0}\text{ first}) = \frac{1}{2}$ exactly.

(iii) Tightening. sF admits every description F admits, via $\beta = 0$, *and* the complement route, via $\beta = 1$. A one-sided upper bound on description length can therefore only improve:

$$\hat{K}_{\mathsf{sF}}(s \mid x) \;=\; \min\big(\hat{K}(s \mid x),\, \hat{K}(\bar{s} \mid \bar{x})\big) + 1 \;\le\; \hat{K}(s \mid x) + 1, \tag{A.4}$$

with equality exactly when the complement is not the cheaper face. The $+1$ is the polarity bit, the price of recording which of the two runs produced s, and paying it is what makes the estimator a description length on a machine rather than an unnormalised pooled count. In the companion enumeration the complement route supplies the shorter program for a substantial minority of outputs, so the mean effect of symmetrisation is a *reduction* in certified description length despite the extra bit.

B Panel registry

Table 3: Panel registry: every run, its serving route and precision, the families with usable readouts (`/lp` = log-probabilities exposed), the billed API cost in USD and the reasoning-effort setting used (local rows carry no metered cost), and audit notes.

Model	Precision	Route	Readouts	USD	Effort	Notes
Local panel						
MiniMax-M2.7	Q8_0	local	ind+cont/lp	–	–	–
Ministral-3-14B-Base-2512	Q8_0	local	ind+cont/lp	–	–	–
Ministral-3-14B-Instruct-2512	Q8_0	local	ind+cont/lp	–	–	–
Ministral-3-14B-Reasoning-2512	Q8_0	local	ind+cont/lp	–	–	–
Qwen3-30B-A3B-Base	Q8_0	local	ind+cont/lp	–	–	–
Qwen3-30B-A3B-Instruct-2507	Q8_0	local	ind+cont/lp	–	–	–
Qwen3-30B-A3B-Thinking-2507	Q8_0	local	ind+cont/lp	–	–	–
Qwen3.5-0.8B	Q8_0	local	ind+cont/lp	–	–	–
Qwen3.5-27B	Q8_0	local	ind+cont/lp	–	–	–
Qwen3.5-2B	Q8_0	local	ind+cont/lp	–	–	–
Qwen3.5-4B	Q8_0	local	ind+cont/lp	–	–	–
Qwen3.5-9B	Q8_0	local	ind+cont/lp	–	–	–
Qwen3.6-27B	Q3_K_M	local	ind+cont/lp	–	–	–
Qwen3.6-27B	Q4_K_M	local	ind+cont/lp	–	–	–
Qwen3.6-27B	Q5_K_M	local	ind+cont/lp	–	–	–
Qwen3.6-27B	Q6_K	local	ind+cont/lp	–	–	–
Qwen3.6-27B	Q8_0	local	ind+cont/lp	–	–	–
Qwen3.6-27B	UD-IQ2_M	local	ind+cont/lp	–	–	–
Qwen3.6-35B-A3B	BF16	local	ind+cont/lp	–	–	–
Qwen3.6-35B-A3B	Q8_0	local	ind+cont/lp	–	–	–
Qwen3.6-35B-A3B	UD-IQ1_M	local	ind+cont/lp	–	–	–
Qwen3.6-35B-A3B	UD-IQ2_M	local	ind+cont/lp	–	–	–
Qwen3.6-35B-A3B	UD-Q3_K_M	local	ind+cont/lp	–	–	–
Qwen3.6-35B-A3B	UD-Q4_K_M	local	ind+cont/lp	–	–	–
Qwen3.6-35B-A3B	UD-Q5_K_M	local	ind+cont/lp	–	–	–
Qwen3.6-35B-A3B	UD-Q6_K	local	ind+cont/lp	–	–	–
gemma-4-12b-it	Q8_0	local	ind+cont/lp	–	–	–
gemma-4-26B-A4B-it	BF16	local	ind+cont/lp	–	–	–
gemma-4-26B-A4B-it	Q8_0	local	ind+cont/lp	–	–	–
gemma-4-26B-A4B-it	UD-IQ2_M	local	ind+cont/lp	–	–	–
gemma-4-26B-A4B-it	UD-Q3_K_M	local	ind+cont/lp	–	–	–
gemma-4-26B-A4B-it	UD-Q4_K_M	local	ind+cont/lp	–	–	–
gemma-4-26B-A4B-it	UD-Q5_K_M	local	ind+cont/lp	–	–	–
gemma-4-26B-A4B-it	UD-Q6_K	local	ind+cont/lp	–	–	–
gemma-4-31B-it	Q3_K_M	local	ind+cont/lp	–	–	–
gemma-4-31B-it	Q4_K_M	local	ind+cont/lp	–	–	–
gemma-4-31B-it	Q5_K_M	local	ind+cont/lp	–	–	–
gemma-4-31B-it	Q6_K	local	ind+cont/lp	–	–	–
gemma-4-31B-it	Q8_0	local	ind+cont/lp	–	–	–
gemma-4-31B-it	UD-IQ2_M	local	ind+cont/lp	–	–	–
gemma-4-E2B-it	Q8_0	local	ind+cont/lp	–	–	–
gemma-4-E4B-it	Q8_0	local	ind+cont/lp	–	–	–
gpt-oss-20b	F16	local	ind+cont/lp	–	–	–
gpt-oss-20b	Q2_K	local	ind+cont/lp	–	–	–
gpt-oss-20b	Q3_K_M	local	ind+cont/lp	–	–	–
gpt-oss-20b	Q4_K_M	local	ind+cont/lp	–	–	–

Model	Precision	Route	Readouts	USD	Effort	Notes
gpt-oss-20b	Q5_K_M	local	ind+cont/lp	–	–	–
gpt-oss-20b	Q6_K	local	ind+cont/lp	–	–	–
gpt-oss-20b	Q8_0	local	ind+cont/lp	–	–	–
phi-4	F16	local	ind+cont/lp	–	–	–
phi-4	Q2_K	local	ind+cont/lp	–	–	–
phi-4	Q3_K_M	local	ind+cont/lp	–	–	–
phi-4	Q4_K_M	local	ind+cont/lp	–	–	–
phi-4	Q5_K_M	local	ind+cont/lp	–	–	–
phi-4	Q6_K	local	ind+cont/lp	–	–	–
phi-4	Q8_0	local	ind+cont/lp	–	–	–
API panel						
claude-fable-5	API	Anthropic	ind	8.10	low	accuracy only
claude-fable-5+opus-4.8-fb	API	Anthropic	ind	8.23	low	accuracy only
claude-opus-4-8	API	Anthropic	ind	4.90	off	accuracy only
deepseek-v4-flash	API	NIM	ind+cont/lp	–	–	–
deepseek-v4-pro	API	NIM	ind	–	–	accuracy only
dracarys-llama-3.1-70b-instruct	API	NIM	ind+cont/lp	–	–	–
gemini-2.5-pro	API	Vertex	ind+cont/lp	8.80	128 tok	–
gemini-3.1-pro-preview	API	Vertex	ind	4.30	128 tok	accuracy only
gemini-3.6-flash	API	Gemini API	ind	4.37	managed	accuracy only
gemma-2-2b-it	API	NIM	ind	–	–	accuracy only
gemma-4-26b-a4b-it	API	Gemini API	ind	–	–	accuracy only
gemma-4-31b-it	API	Gemini API	ind	–	–	accuracy only
glm-5.2	API	NIM	ind	–	–	accuracy only
gpt-4.1-2025-04-14	API	OpenAI	ind+cont/lp	1.64	–	–
gpt-4o-2024-05-13	API	OpenAI	ind+cont/lp	4.54	–	–
gpt-4o-2024-11-20	API	OpenAI	ind+cont/lp	2.39	–	–
gpt-5-2025-08-07	API	OpenAI	ind	0.30	minimal	accuracy only
gpt-5.1-2025-11-13	API	OpenAI	ind+cont/lp	1.96	none	–
gpt-5.2-2025-12-11	API	OpenAI	ind+cont/lp	1.93	none	–
gpt-5.4-2026-03-05	API	OpenAI	ind+cont/lp	2.91	none	–
gpt-5.5-2026-04-23	API	OpenAI	ind+cont/lp	5.96	none	–
gpt-5.6-sol	API	OpenAI	ind+cont/lp	4.88	none	–
gpt-oss-120b	API	NIM	ind	–	–	accuracy only
gpt-oss-20b	API	NIM	ind	–	–	accuracy only
grok-4.20-0309-non-reasoning	API	xAI	ind	0.15	n/a	accuracy only
grok-4.5	API	xAI	ind	11.34	low	accuracy only
inkling	API	NIM	ind	–	–	answers degenerate (excl.); accuracy only
laguna-xs-2.1	API	NIM	ind+cont/lp	–	–	–
llama-3.1-70b-instruct	API	NIM	ind+cont/lp	–	–	–
llama-3.1-8b-instruct	API	NIM	ind+cont/lp	–	–	–
llama-3.2-3b-instruct	API	NIM	ind+cont/lp	–	–	–
llama-3.3-70b-instruct	API	NIM	ind	–	–	accuracy only
llama-3.3-nemotron-super-49b-v1	API	NIM	ind+cont/lp	–	–	–
minimax-m3	API	NIM	ind	–	–	answers degenerate (excl.); accuracy only
mistral-large-3-675b-instruct-2512	API	NIM	ind+cont/lp	–	–	cont. unservable (retired)
mistral-nemotron	API	NIM	ind+cont/lp	–	–	–

Model	Precision	Route	Readouts	USD	Effort	Notes
mistral-small-4-119b-2603	API	NIM	ind+cont/lp	–	–	–
mixtral-8x7b-instruct-v0.1	API	NIM	ind	–	–	accuracy only
nemotron-3-nano-30b-a3b	API	NIM	ind+cont/lp	–	–	–
nemotron-3-super-120b-a12b	API	NIM	ind+cont/lp	–	–	–
nemotron-3-ultra-550b-a55b	API	NIM	ind+cont/lp	–	–	–
nemotron-mini-4b-instruct	API	NIM	ind	–	–	accuracy only
nvidia-nemotron-nano-9b-v2	API	NIM	ind	–	–	answers degenerate (excl.); accuracy only
qwen3-next-80b-a3b-instruct	API	NIM	ind+cont/lp	–	–	–
qwen3.5-397b-a17b	API	NIM	ind+cont/lp	–	–	–
sarvam-m	API	NIM	ind+cont/lp	–	–	–
solar-10.7b-instruct	API	NIM	ind	–	–	answers degenerate (excl.); accuracy only
step-3.5-flash	API	NIM	ind	–	–	answers degenerate (excl.); accuracy only
step-3.7-flash	API	NIM	ind	–	–	answers degenerate (excl.); accuracy only

C The prompts

Every score in this paper is read from a model's response to one of two prompts, held fixed across all 105 rows up to the per-row serving wrapper recorded in the registry. They are reproduced here verbatim; the ablation of §2.5 varies exactly these strings.

Induction. The examples are rendered one per line as `input -> output`, the query line is left open, and the instruction asks for a bare bit string. Answers are read either as free text or under a grammar constraint (Methods).

```
Each line maps an input bit string to a fixed hidden program's output bit
string.  Give ONLY the output for the last line:  a bit string like 011, with
no words, no code, no explanation.  If unsure, still answer with your best bit
string.

0010 -> 01
010 -> 01
0000 ->
```

The empty input is written `()` so that a zero-length string is visible rather than blank. At $n = 0$ examples the block between the instruction and the query line is empty, and the model is asked for the output of an unseen program on one input, which is what makes the zero-shot column a pure prior read-out.

Continuation. The next-token family reveals the output one character at a time and reads the served distribution over $\{0, 1, \text{END}\}$ from the log-probabilities of a single-token reply. The terminator is the full stop.

```
A fixed program over bit strings was run on the input below and produced an
output, shown so far.  Reply with EXACTLY ONE character:  the next output bit
('0' or '1'), or '.'  if the output has ended.

input:  01101111
output so far:  1000
next character:
```

`output so far` is written `(empty)` at the first revealed position. The three classes are recovered from the returned top-k tokens by exact single-character match, with the first-character folding described in the Limitations as the robustness variant.

Serving wrappers. Chat-served models receive the prompt as a single user turn. Models whose default template emits a reasoning block before the answer, and base checkpoints with no chat template at all, are served through `/v1/completions` with a fixed wrapper that prefills an empty, already-closed reasoning block so the first generated token is the answer. The wrappers are recorded per row in the registry; as an example, the Qwen family uses

```
<|im_start|>user
{prompt}<|im_end|>
<|im_start|>assistant
<think></think>
```

with the reasoning block opened and closed immediately so the next generated token is the answer, and the Mistral family the analogous `[INST]{prompt}[/INST][THINK][/THINK]`. Equivalent wrappers exist for the gpt-oss, MiniMax, DeepSeek and GLM families; all are in the released harness. Substituting the raw wrapper for a model's chat template is one of the ablation arms, and is the arm that moves the score most (§2.5).

The worked examples below use the raw machine F *except where* sF *is itself the subject, and, where a weighting is needed, the length-only prior: the minimal setting that keeps every number legible and hand-checkable. The frozen benchmark serves* sF *optima with time-penalised weights (Table 2); the constructions are identical, with the polarity mixture of Eq.* (10) *layered on top.*

D Worked example: the reference machine

Program $p =$ `>+<` (base-5 index rule = 35, length $L = 3$) on input $x =$ `10`. The head starts at the tape centre c, with x written rightward (cell_c=`1`, cell_{c+1}=`0`). Execution halts when the program counter passes the last instruction:

step	instruction	head	pc	$(\text{cell}_c, \text{cell}_{c+1})$
0 (start)	–	c	0	(1, 0)
1	`>` (right)	c+1	1	(1, 0)
2	`+` (flip)	c+1	2	(1, 1)
3	`<` (left)	c	3 ⇒ halt	(1, 1)

The head visited cells $\{c, c{+}1\}$, so the output is the head-excursion window `11`, with halt time $t_p = 3$ and output length 2. Mapped over the input set, this program has behaviour (`01`, `01`, `11`) and contributes one of the behaviour entries counted in Appendix G.

E Worked example: encoding length and prior weight

Equation (2) with $\log_2 6 = 2.584963\ldots$ gives, and Eq. (3) the per-program weight $w = 2^{-\ell(L)}$; the last column is the total prior mass of all programs of that length:

L	$\ell(L)$ (bits)	$2^{-\ell(L)}$	5^L	$5^L\, 2^{-\ell(L)}$
1	6	1.56×10^{-2}	5	0.0781
2	8	3.91×10^{-3}	25	0.0977
3	11	4.88×10^{-4}	125	0.0610
4	13	1.22×10^{-4}	625	0.0763
5	16	1.53×10^{-5}	3,125	0.0477
6	19	1.91×10^{-6}	15,625	0.0298

A single program at L=1 carries $2^5 = 32\times$ the weight of one at L=3; the prior concentrates on short programs, which is why summing $L = 1..L_{\text{eval}}$ is exact to far beyond float precision once any short consistent program exists.

F Worked example: Elias-δ time penalty

Equation (5) for representative halt times, with the resulting Levin time weight $2^{-\delta(t)}$:

t	$b(t)$	$\gamma(t)$	$\delta(t)$	$2^{-\delta(t)}$
1	1	1	1	5.00×10^{-1}
2	2	3	4	6.25×10^{-2}
4	3	5	5	3.13×10^{-2}
8	4	7	8	3.91×10^{-3}
16	5	9	9	1.95×10^{-3}
1024	11	21	17	7.63×10^{-6}

A length-3 witness ($\ell = 11$) that halts at $t = 4$ ($\delta = 5$) therefore has certified description length $\hat{K} = \ell + \delta = 11 + 5 = 16$ bits (Eq. (8)) and Levin weight $2^{-(\ell+\delta)} = 2^{-16} \approx 1.53 \times 10^{-5}$ (Eq. (7)).

G Worked example: the behaviour table (sufficient statistic)

Take $M = 1$, $L_{\max} = 3$, so the input set is $(\varepsilon, \mathtt{0}, \mathtt{1})$ and $\sum_{L=1}^{3} 5^L = 155$ programs are enumerated. They realise 29 distinct behaviours with total mass $Z_{\text{tot}} = 0.236816$, the total enumerated prior mass including the non-halting outcome (distinct from the halting-only normaliser $Z(x)$ of Eq. (9)). Writing each behaviour as $b = (b(\varepsilon), b(\mathtt{0}), b(\mathtt{1}))$ and using Eqs. (4), (3):

behaviour b	mass $\sum_{p \vdash b} w(p)$	#programs
$(\bot, \bot, \bot)$ (never halts)	1.333×10^{-1}	106
$(\mathtt{00}, \mathtt{00}, \mathtt{01})$	2.148×10^{-2}	6
$(\mathtt{1}, \mathtt{1}, \mathtt{0})$	1.611×10^{-2}	2

The map behaviour $\mapsto$ mass is everything the optimal predictor consumes; the program count column is what the GPU cross-check (gate 1) reproduces byte-for-byte.

H Worked example: the exact induction posterior

Hidden behaviour $b^\star = (\texttt{00}, \texttt{00}, \texttt{01})$ from Appendix G; the query is the output on the empty input (truth = 00), and examples are revealed in the order $(\texttt{0} \to \texttt{00})$, then $(\texttt{1} \to \texttt{01})$. The posterior predictive (Eq. (1)) and the marginal foil's regret (Eq. (16)) are:

n	examples	optimal $P(\cdot \mid x_q, \mathcal{E}_n)$ (top)	opt. LL / marg. regret
0	–	$\perp$:0.565, 00:0.186, 1:0.070, …	2.43 / 1.66 bits
1	$\texttt{0} \to \texttt{00}$	00:1.000	0.00 / 0.00 bits
2	$+\,\texttt{1} \to \texttt{01}$	00:1.000	0.00 / 1.00 bits

One example pins the program (the posterior collapses to 00), so this task has Bayes sample complexity BSC = 1 (Eq. (21)). The pattern-completion foil, shown two *different* outputs by n=2, hedges between them and pays ≈ 1 bit of regret while the optimum pays none.

I Worked example: continuation next-token

This example enumerates programs of length $L \leq 5$ under raw F with length-only weights, so every value below is hand-checkable; the full benchmark uses the Levin-weighted sF mixture over $L \leq 13$. For input $x = \texttt{10}$ the conditional marginal $P(\cdot \mid x)$ is tabulated in Appendix J. Applying Eq. (14) as the prefix r grows:

prefix r	$P^\star(\texttt{0}, \texttt{1}, \text{END} \mid x, r)$	entropy (bits)
ε	(0.576, 0.424, 0.000)	0.983
0	(0.329, 0.412, 0.259)	1.559
01	(0.081, 0.046, 0.873)	0.669

The distribution sharpens toward END once the prefix completes a high-mass output. The input-agnostic pooled foil is close here ($\mathrm{JS}(P^\star \,\|\, \text{pooled}) = 0.0058$ bits at $r = \varepsilon$), illustrating why continuation is a sensitive fidelity probe yet only weakly separates these foils (cf. Fig. 3a).

J Worked example: the symmetrised machine sF

Continuing from Appendix I ($L \leq 5$, length-only weights, valid programs): the sF conditional for $x = \texttt{10}$ mixes the raw conditional at $x = \texttt{10}$ with the complemented conditional at the complemented input $\bar{x} = \texttt{01}$ (Eq. (10)):

s	$P(s \mid \texttt{10})$	$P(\bar{s} \mid \texttt{01})$	$\mathrm{s}P(s \mid \texttt{10}) = \frac{1}{2}[\cdot + \cdot]$
01	0.207	0.050	0.128
10	0.207	0.234	0.221
0	0.149	0.137	0.143
1	0.036	0.094	0.065
00	0.105	0.053	0.079
11	0.089	0.239	0.164

Both columns are normalised distributions, so the mixture is one as well. Computing the full table both ways confirms the exact invariance: $\max_s \left|\mathrm{s}P(s \mid \texttt{10}) - \mathrm{s}P(\bar{s} \mid \texttt{01})\right| = 0$ to machine precision: the complement twin's optimum *is* the flipped original. The first-position next-token distribution $(q_{\texttt{0}}, q_{\texttt{1}}, q_{\text{END}})$ makes the correction visible: raw F gives the (0.576, 0.424, 0.000) of Appendix I on $x = \texttt{10}$ (the tape's zero preference), whereas sF gives (0.420, 0.580, 0.000) on $x = \texttt{10}$ and exactly (0.580, 0.420, 0.000) on $x = \texttt{01}$. The residual asymmetry now tracks the *input* (short programs tend to echo its leading bit), not the tape: averaged over any complement-closed input set the first-bit law is $\frac{1}{2}$ exactly.

K Worked example: abductive posterior

Candidates $\mathcal{C} = \{\texttt{0}, \texttt{1}, \texttt{10}, \texttt{11}\}$, hidden true input `10`, revealed output `01`, uniform prior $\pi = 1/4$. Equation (15) reweights each candidate by its prefix mass $\mu(x, r)$:

prefix r	$P^\star(\cdot \mid r)$ over $(\texttt{0}, \texttt{1}, \texttt{10}, \texttt{11})$	$P^\star$(true)	entropy (bits)
ε	(0.250, 0.250, 0.250, 0.250)	0.250	2.000
`0`	(0.299, 0.233, 0.233, 0.235)	0.233	1.991
`01`	(0.131, 0.269, 0.269, 0.330)	0.269	1.932

Each revealed token reweights the candidates by how much of their output mass is consistent with the prefix; the evidence-ignoring prior-only foil stays at π, giving $\mathrm{JS}(P^\star \,\|\, \text{prior-only}) = 0.0181$ bits at $r = \texttt{01}$. With these tiny length-only marginals the update is modest; at L=13 the posterior concentrates sharply (Fig. 3b).

L Worked example: Jensen–Shannon divergence

For $p = (0.7, 0.2, 0.1)$ and $q = (0.4, 0.4, 0.2)$ over $\{\texttt{0}, \texttt{1}, \textsc{end}\}$, Eq. (17) gives $m = \frac{1}{2}(p + q) = (0.55, 0.30, 0.15)$ and

$$\mathrm{KL}(p \,\|\, m) = 0.0681, \quad \mathrm{KL}(q \,\|\, m) = 0.0652, \quad \mathrm{JS} = \tfrac{1}{2}(0.0681) + \tfrac{1}{2}(0.0652) = 0.0667 \text{ bits}, \tag{L.5}$$

finite and symmetric without any smoothing.

M Worked example: the F-ICL macro-average

From the frozen Qwen evaluation, the continuation family has mean divergence $\bar{D}_{\mathrm{cont}} = 0.3973$ (Eq. (18)). To keep the arithmetic legible we take continuation alone, so the macro-average (Eq. (19)) reduces to a single family:

$$\mathrm{FICL} = 1 - \tfrac{1}{1}\,\bar{D}_{\mathrm{cont}} = 1 - 0.3973 = 0.6027, \tag{M.6}$$

with keystroke-reference thresholds $1 - \overline{\mathrm{JS}}_{\mathrm{hedge}} = 0.8695$ and $1 - \overline{\mathrm{JS}}_{\mathrm{committed}} = 0.4792$, and median sample-complexity efficiency 0.333 (Eq. (22)). With both families scored, $\mathrm{FICL} = \frac{1}{2}\big[(1 - \bar{D}_{\mathrm{ind}}) + (1 - \bar{D}_{\mathrm{cont}})\big]$, each family weighted equally regardless of its task count. On the anchored scale of Eq. (20) this run falls on the *lower* branch: its mean divergence 0.3973 exceeds the hedged keystroke divergence $\overline{\mathrm{JS}}_{\mathrm{hedge}} = 0.1305$, so FICL-A $< \frac{1}{2}$ and the run is scored by its position between the worst-case predictor and the keystroke reference rather than between the reference and Bayes. This is the headline configuration of §2.4.

N Worked example: the dumped program joint

The dump records, per length L and input x, an inverted index clean output $\mapsto$ sorted rule list. For $L = 3$, $x = \texttt{0}$ (the $5^3 = 125$ length-3 programs) the index begins:

output s	#rules	first rule identities
`00`	10	$5, 12, 19, 26, 37, 44, 60, 61, \ldots$
`000`	4	$1, 6, 25, 30$
`0`	3	$79, 84, 89$
`01`	3	$7, 35, 51$
`10`	3	$11, 27, 55$

This input has 35 halting programs over 11 distinct outputs (the CSR arrays `rules_ptr`/`rules` pack the lists; `bit_length`/ `out_word_ptr`/`out_words` pack the outputs). Serving the hidden program of Appendix H ($b^\star = (\texttt{00}, \texttt{00}, \texttt{01})$, query on the empty input, examples $\texttt{0} \to \texttt{00}$ then $\texttt{1} \to \texttt{01}$) by Eq. (13) (intersecting the rule lists across examples, re-running the survivors on the query, and weighting lengths $L = 1, 2, 3$ by $2^{-\ell(L)}$) reproduces the behaviour-table joint *exactly*:

n	served $P(\cdot \mid x_q, \mathcal{E}_n)$ (top)	max $\lvert$served − joint$\rvert$
0	⊥:0.565, 00:0.186, 1:0.070, 000:0.041, …	0
1	00:1.000	0
2	00:1.000	0

The 0-shot row is served by the bounded universe scan (no positive example to intersect); the 1- and 2-shot rows by list intersection plus seed-and-rerun. Agreement is bit-exact (0 to machine precision), which is the property that lets the dumped joint serve the induction optima at the complete $L \leq 13$ budget, where the small-program joint is intractable.