

Three Things Called Budget Awareness: Observability, Forecasting and Allocation in LLM Agents, Why Every Published Allocation Gain Is Keyed to a Signal Measured After the Fact, and the Run-to-Run Variance No Forecast Is Scored Against

Mahendrakar, Pranay

SONYTECH

ORCID: 0009-0003-7224-029X

Life of Research

2026-09-22

Pranay Mahendrakar

Abstract

Two 2024-2026 literatures make claims about resource use in language-model agents that look incompatible. One reports that allocating test-time compute according to problem difficulty beats spending it uniformly, by margins up to a factor of four in compute at equal accuracy. The other reports that agents cannot estimate their own resource consumption, that the estimates are systematically optimistic, and that the ability to produce them is only weakly related to the ability to do the task. If intelligent allocation were the decisive lever, the most capable models should be the best allocators, and several independent measurements say they are not. This paper argues that the incompatibility is apparent and dissolves under a partition. The phrase budget awareness is carrying at least four distinct decisions: adherence to a budget someone else set, observability of the remaining budget inside the agent's context, forecasting of one's own future consumption, and allocation of a known budget across competing demands. The four are measured on different instruments and do not have the same answer. Adherence is largely solved in the tested ranges: Aggarwal and Welleck (2025) report output lengths that closely track requested targets at 512, 1024, 2048 and 3600 tokens. Observability is not a competence at all, and it accounts for a share of the reported failure large enough to move headline numbers. Liu et al. (2025b) report that a standard ReAct agent issues 13.77 search calls when given a budget of 30 and 14.24 when given 100, and that a plug-in which does nothing but keep the remaining budget in context reaches 12.8 percent on BrowseComp at a budget of 10 against 12.6 percent for the unmodified agent at a budget of 100, using 40.4 percent fewer search calls at 31.3 percent lower cost. Wen et al. (2025) reach the same conclusion at the token level and state the sharper form of it: naming the budget once in the prompt is insufficient, and the remaining budget has to be re-inserted periodically during generation. Forecasting is poor and structurally biased. Lin et al. (2026) find that across twenty model-environment pairs, optimistic misses outnumber conservative ones at every rollout-progress bin, that weaker models are more optimistic rather than less, and that on failed trajectories models predict feasibility above 70 percent after 60 percent of the budget is spent. Bai et al. (2026) find that eight frontier models correlate with their own realized token usage at up to 0.39 and underestimate it systematically. Allocation fails on a different axis again: Fan et al. (2026) find that under a shared budget, solving order tracks presentation position at $\rho = +0.68$ while effort-value correlations run from 0.00 to $+0.11$, and that the set of questions receiving substantive work overlaps the highest-value-density set at 0.59 against a chance reference of 0.59. The central claim is about what the allocation results license. In every case located by this run's search, the per-instance signal that tells

an allocator which question deserves more compute is a quantity obtained by running the model on that question and measuring the outcome: Snell et al. (2024) define the compute-optimal strategy as an argmax indexed by the ground-truth answer and bin difficulty from 2048 samples per question, stating in their own text that this "assumes oracle access to a ground-truth correctness checking function, which is of course not available upon deployment"; Damani et al. (2024) train a separate predictor on eight sampled responses per query scored by an 8B reward model, and state that Monte Carlo estimation of the quantity "may in general require more computation than we eventually wish to allocate"; Fan et al. (2026) compute value density from an independent 40,960-token reference run; Zhou et al. (2026b) gate budget on rollout-derived solvability and write that solvability "is a posterior signal, while budget investment must be committed before reasoning begins"; the two methods that advertise self-assessed difficulty, Huang et al. (2025) and Singh et al. (2025), train the self-assessment against a label computed from the pass rate over sixteen or N sampled rollouts; Nazi and Dipta (2026) score plans against an oracle that knows solvability and cost for every item; and the ground-truth target in the founding self-knowledge result, Kadavath et al. (2022), is the fraction of sampled answers that are correct. Two located designs are genuine partial exceptions and are treated as such: Zuo and Zhu (2025) pay for difficulty estimation out of the same budget they allocate, and Nogueira et al. (2025) read certainty off the forward pass and fit only a single global threshold. The two literatures are therefore not in conflict, because the out-of-band allocation gains are produced by an estimator fitted to measured outcomes at a compute cost that the same papers decline to put on the axis, and the budget-awareness results ask whether the agent can supply that quantity itself. What survives the partition is narrower and less comfortable than either headline. Feasibility prediction trains easily: Lin et al. report Qwen-7B moving from 25.5 percent to roughly 90 percent with supervised fine-tuning alone, which they read as a formatting and calibration problem rather than a capability gap. What does not train is the interval, and what does not transfer is anything, with cross-task reward retention of 17 to 36 percent. Against this, two results cut the other way and are given their weight here: Sun et al. (2026) remove a budget-state scaffold after training and find the behaviour persists, with re-adding it at inference buying nothing, and Chen et al. (2026) find that exposing resource metadata is not sufficient and that the benefit is model-dependent, concluding that resource-aware orchestration is a distinct capability rather than an automatic consequence of exposure. The paper's main open problem is a missing denominator. Bai et al. report that runs of the same agent on the same SWE-bench Verified task differ by up to 30 times in total tokens. No budget forecaster located here is scored against the spread of the quantity it forecasts, so a reported 47 percent interval coverage cannot presently be read as either near or far from what any forecaster could achieve, and the share of forecast error that belongs to the forecaster rather than to the variance of the trajectory is unmeasured. One incidental finding is recorded because the claim it affects is repeated everywhere: in Snell et al., the running text and the caption of the figure it describes state the pretraining-versus-test-time tradeoff in opposite directions on the difficulty axis, the text putting pretraining ahead on the hardest bins and the caption putting it ahead on the easy questions, and the caption's gloss of its own ratio is inverted against the definition given two paragraphs earlier.

1. Introduction

An agent that runs for a hundred steps spends something at every one of them. It spends tokens on its own reasoning, calls on its tools, seconds of wall clock, and, in some deployments, money. Until recently the standard practice was to add all of this up after the run and report it beside the accuracy, if at all. Kapoor et al. (2024) documented the consequence: for what they call substantially similar accuracy on HumanEval, cost differed by almost two orders of magnitude across agent architectures, one architecture cost over fifty times more than a simple retry-with-warming baseline, and the cost of running the agents was not a top-line metric in any of the papers they examined. Their conclusion was

that accuracy alone cannot identify progress, because accuracy can be raised by methods that are, in their phrase, scientifically meaningless, such as retrying.

Erol et al. (2025) built a framework around the same observation, formalising the expected monetary cost of generating a correct solution and the frontier cost achievable across available models, and reporting that lightweight models are most cost-effective for basic quantitative tasks while large models are most cost-effective for knowledge-intensive ones. Which model is economical is therefore not a fact about the model alone.

The field has since moved, and it has moved in two directions at once. One direction treats the budget as something to be spent well. Snell et al. (2024) showed that choosing a test-time strategy as a function of problem difficulty could match the accuracy of best-of-N sampling using up to four times less generation compute, and titled the result as a claim that scaling test-time compute optimally can be more effective than scaling model parameters. Damani et al. (2024) reduced computation by up to 50 percent at no cost to response quality by predicting, per query, how much an additional unit of compute would buy. A survey of this area (Alomrani et al., 2025) organises it into methods that operate under fixed budgets and methods that scale compute with input difficulty. The direction of travel is clear: allocation beats uniformity, and the gains are large.

The other direction treats the budget as something the agent should know about. Lin et al. (2026) asked whether a frontier agent, mid-execution, can say how much budget it still needs and whether the task is still finishable. Their answer is that it cannot do this well, that the failure has a specific shape, and that the ability to do it is only weakly related to the ability to complete the task. They report the weak relation as a correlation of about 0.35, a number Section 11 returns to because its own source attaches it to three different pairs of quantities. Their framing is that budget is less a metric for after-the-fact accounting and more a control signal that resource-limited agents currently lack.

Read together, the two directions appear to make incompatible commitments. If intelligent allocation is the lever that matters, then allocation is a skill, and one would expect the most capable models to have more of it, in the way they have more of most skills. They do not. Nazi and Dipta (2026) report that object-level capability and metacognitive control dissociate empirically, that extended reasoning raises task accuracy while leaving allocation quality unchanged or worse, and that no configuration they tested was simultaneously a strong monitor and a strong controller on more than one benchmark. Fan et al. (2026) report that strong per-question reasoning does not imply effective global compute allocation. Wu et al. (2026) report that completion under a budget and economical action selection are distinct properties. Chen et al. (2026) report that strong logical planning does not imply strong resource-aware scheduling. Four further groups, four instruments, one dissociation.

This paper argues that the apparent conflict is an artifact of a word. Budget awareness, as the 2024-2026 literature uses it, names at least four decisions that are not the same decision, are measured by different instruments, and have different answers. Separating them removes the conflict and leaves a sharper question in its place.

The four are named in Section 4 and then taken one at a time. Section 5 argues that an identifiable share of the reported failure is not a failure of self-knowledge at all but a failure of observability: the remaining budget is frequently not in the agent's context, and three independent lines of work converge on the same remedy, which is to put it there and keep it there. Section 6 takes the forecasting results at face value and reports what they establish and what they do not. Section 7 makes the paper's central claim: in every case this run's search could locate, the per-instance signal that tells an allocator where to spend is a quantity obtained by running the model and measuring what happened, several of the papers concerned say so in their own text, and the two designs that escape this are the ones that do not need a per-instance estimate at all. Section 8 works through the canonical compute-optimal result in

detail, because the definitional form of its objective is the cleanest statement of the point. Section 9 reports what happens when the budget is known and the agent has to divide it, which is where the failures are most severe. Section 10 reports the one decision of the four that is largely solved, and argues that its being solved is evidence for the partition rather than against it.

Section 11 examines the correlation that the popular version of this argument rests on, and reports that the number is attached to three different pairs of quantities in its own source. Section 12 states the paper's main open problem: the quantity being forecast is itself highly variable across runs of the same task, and no forecaster located here is scored against that variance, so the published coverage figures have no denominator. Section 13 gives the strongest case against this paper's reading, including the results that cut the other way and the two designs that are genuine partial counterexamples to Section 7. Section 14 states what is not known. Section 15 proposes six experiments that would settle parts of it, five of which could be run on data that already exists.

No experiments were run for this paper. It is an analysis of what a body of published measurements licenses, and its contribution is a partition, an argument about the provenance of a class of signals, and a missing denominator.

2. Scope, terms, method, and what this paper is not

Terms. *Budget* here means any scalar or vector resource whose consumption by an agent is bounded: generated tokens, tool calls, wall-clock time, money, or a physical capacity such as memory. Lin et al. (2026) divide these into *internal* budget, the compute generated by the model's own reasoning, and *external* budget, the cost the agent commits in its environment, and that division is used here. *Adherence* means producing a trajectory whose realized consumption satisfies a bound given from outside. *Observability* means that the remaining budget is present in the agent's context at decision time. *Forecasting* means producing, before or during execution, an estimate of consumption or feasibility that will be scored against the realized value. *Allocation* means dividing a bounded resource among competing demands, whether those demands are questions in a queue, tool calls in a trajectory, or steps in a chain of thought. These four are stipulated, not inherited; no cited paper uses all four in these senses.

Method and its limits. The evidence base was built by searching DataCite for arXiv-registered DOIs across about sixty query formulations, following citation chains out of the papers so found, and retrieving the full HTML rendering of twenty-five of them to read their tables, their appendices and their own limitations sections. Metadata for every reference was taken from the DataCite record rather than written from memory. The remaining references were read at abstract level only, and no claim in this paper that depends on a table, an appendix or a limitations section rests on one of them. Three limits on this procedure matter and are stated here so that the coverage claims later in the paper can be discounted appropriately.

First, the arXiv query API returned HTTP 406 throughout the day of the search, so every search ran through a single index. A paper whose DataCite abstract does not contain the query terms, or which is not registered with DataCite under the arXiv client, would not have been found. Coverage claims in this paper should be read as claims about what one index surfaced on one day, and they are phrased that way throughout.

Second, the two papers treated here as the clearest independent measurements of forecasting failure are not fully independent: three authors appear on both Lin et al. (2026) and Bai et al. (2026). Section 6 says so at the point of use.

Third, the area is moving fast enough that the survey literature lags it. The most directly relevant survey located here (Alomrani et al., 2025) predates every benchmark this paper leans on. It

taxonomises controllability and adaptiveness and reports its own measurements on budget adherence, but it does not treat forecasting as a separate capability and could not have cited the 2026 results that establish the dissociation. A reader looking for prior statements of this paper's partition should look at the individual benchmark papers, several of which draw one or two of the same lines within their own scope, rather than at a survey.

What this paper is not. It is not a survey of efficient reasoning; that exists several times over (Sui et al., 2025; Alomrani et al., 2025). It is not an argument that test-time compute allocation does not work; the cited allocation results are taken as sound within their stated scope, and Section 8 is careful to attack the interpretation rather than the measurement. It is not a proposal for a method. It is not an argument that language models lack a self-model; Section 5 argues the opposite for one important class of failure, and Section 13 gives the strongest objection to that argument.

Boundary with the author's prior records. A companion record (Mahendrakar, 2026a) analyses abstention in tool-using agents and the cost term that both readings of it require; the overlap with the present paper is confined to the single finding of Kirichenko et al. (2025) that both cite, and the decision studied there is whether to act, not how much to spend. A second record (Mahendrakar, 2026b) analyses what router results establish about an agent's knowledge of its own failures; the decision there is a choice between models at query time given a difficulty estimate, not an online estimate of remaining resource over a trajectory. Where the present paper touches either, it says so.

3. The claim on the table, and the two literatures that make it

The claim this paper is responding to can be stated in two sentences, one from each side.

From the allocation side: uniform compute allocation is suboptimal, and choosing how much compute to spend as a function of the problem yields large savings at equal accuracy. Snell et al. (2024) put a number on this: with difficulty-conditioned strategy selection, best-of-N performance can be approached or exceeded using up to four times less test-time compute. Damani et al. (2024) put a different number on it: up to 50 percent less computation at no cost to quality, or up to 10 percent better quality at a fixed budget. Zhou et al. (2026a) report that the optimal thinking budget on MATH-500 varies from about 1.0 thousand tokens for Level 1 problems to about 7.5 thousand for Level 5, which is the size of the opportunity if one could tell the levels apart in advance. The same paper reports the shape that makes the opportunity real: marginal utility for one 32B reasoning model runs at about +3.2 percent per 500 tokens early and turns negative beyond 12 thousand, and 67.5 percent of its negative accuracy flips involve the model explicitly abandoning an answer that was already correct. Two independent results say the monotonicity assumption fails outright rather than merely flattening. Gema et al. (2025) construct tasks on which extending reasoning length makes accuracy worse, spanning counting with distractors, regression with spurious features, constraint tracking and advanced AI risks, and identify five distinct failure modes. Cuadron et al. (2025) report the agentic version, in which models substitute extended internal deliberation for environment interaction, and Chen et al. (2024) report the single-problem version, in which models spend heavily on problems whose answers are trivial. Iacobacci et al. (2025) report that simply increasing the thinking budget is not the most effective use of compute, and Agarwal et al. (2025), in a study spanning over thirty billion generated tokens across eight open-source models and four reasoning datasets, report that no single test-time-scaling strategy universally dominates.

From the awareness side: agents do not know what they are about to spend. Lin et al. (2026) report that all twenty of their model-environment pairs underestimate remaining budget more often than they overestimate it, at every bin of rollout progress, and that the bias tracks model confidence rather than task difficulty, with weaker models on a task being more optimistic rather than less. Bai et al. (2026), studying eight frontier models on SWE-bench Verified, report correlations between predicted and

realized token usage reaching at most 0.39, and that all models systematically underestimate.

The tension is a syllogism. If allocation is what matters, allocation is a skill. Skills scale with capability. Therefore capable models should allocate well. The conclusion is false on the measurements, so one of the premises must go.

The premise this paper rejects is that the allocation results are about a skill the model has. They are about a skill some estimator has, and in the located cases that estimator is external to the model whose compute it allocates, is trained or computed from measured outcomes, and costs compute that the published comparisons do not charge. That is Section 7's claim and Section 8's worked example.

Before getting there, one more piece of the picture is needed, because it changes what the failures mean.

4. Four decisions under one name

Consider an agent with a bounded resource. Four different things can be asked of it, and the literature asks all four under the same phrase.

Adherence. Given a bound set from outside, produce a trajectory that respects it. The bound is known, fixed, and externally enforced or externally checked. Nothing is being estimated. This is the decision that length-controlled generation methods target (Aggarwal and Welleck, 2025; Muennighoff et al., 2025; Li et al., 2025; Wen et al., 2025; Han et al., 2024).

Observability. Is the remaining budget in the agent's context at the moment it chooses an action? This is a property of the harness, not of the model. It admits no training and requires no self-knowledge; it requires a field in a prompt. It is nonetheless the variable that several of the strongest reported improvements move.

Forecasting. Estimate, before or during execution, how much will still be spent, or whether the task is still finishable. The estimate is scored against the realized value. This is what Lin et al. (2026) formalize as progressive interval estimation, what Bai et al. (2026) call pre-execution token consumption prediction, and what Kadavath et al. (2022) called P(IK) in the answer-level case.

Allocation. Given a bounded resource and several competing demands, decide how much each gets. The demands may be questions in a queue (Fan et al., 2026; Nazi and Dipta, 2026), tool calls of different kinds (Liu et al., 2025b), or positions in a single chain of thought (Zhou et al., 2026b).

The four come apart in an ordinary way. Adherence can succeed while forecasting fails: a model can be trained to stop at 1024 tokens without having any view on whether 1024 tokens will be enough. Forecasting can succeed while allocation fails: knowing that question three will be expensive does not entail spending the budget on questions one and two instead. Observability can be absent while all three abilities are present, in which case none of them can be exercised, because the quantity they operate on is not in the input.

The last of these is not a hypothetical. It is the situation Liu et al. (2025b) describe when they write that standard agents lack inherent budget awareness and that, without explicit signals, they fail to utilise additional resources even when available. The next section takes that seriously.

5. Observability: the budget is frequently not in the loop

The setting for this section is the one Shen et al. (2025) named when they argued that for tool-using agents the axis that matters is interaction rather than internal deliberation, so the budget an agent spends is a count of steps in an environment and not only a count of thinking tokens.

Liu et al. (2025b) ran a ReAct agent with a Gemini-2.5-Pro backbone on BrowseComp under different

tool-call budgets and recorded what it actually did. At a budget of 30 it made 13.77 search calls; at a budget of 100 it made 14.24. The agent's behaviour is close to invariant to a budget that varies by a factor of more than three. The paper's gloss is that accuracy "barely changes"; the table it points at shows 10.5 percent at budget 30 and 12.6 percent at budget 100, a 2.1-point difference on a base near 10 percent, so the gloss is doing some work and the underlying observation is better stated as the tool-call counts, which are nearly identical. Their diagnosis is that the agent follows a fixed strategy, terminating when it believes it has found a plausible answer, "unaware of unused resources, even though the context window is not filled up".

Their remedy is a plug-in they call Budget Tracker, which supplies the agent with a continuous signal of resource availability and changes nothing else in the loop. With it, the same backbone at a budget of 10 reaches 12.8 percent, against 12.6 percent for the unmodified agent at a budget of 100, using 8.48 search calls against 14.24, which the paper reports as 40.4 percent fewer search calls, 19.9 percent fewer browse calls, and 31.3 percent lower total cost. The paper also reports the browse-to-search ratio for the two arms, 9.6 percent against 12.9 percent: the budget-aware agent spends a larger share of its calls reading pages in full rather than issuing further searches.

Wen et al. (2025) reach the same conclusion on internal budget, and state the sharper version. Their argument is that merely stating a budget constraint in the initial prompt is insufficient, and that the model needs to be continuously reminded of its remaining token budget as it generates. Their method inserts special control tokens periodically during inference for exactly this purpose, and reports an average accuracy improvement of 4.9 percent over its baselines across tested budgets on MATH-500, AMC 2023 and AIME 2024 for DeepSeek-distilled Qwen 2.5 models at 1.5B and 7B. The same observation appears from the other side in Alomrani et al. (2025), who find that Claude 3.7 Sonnet's observed thinking length grows with the allocated budget but that there is a long tail of problems where the model significantly exceeds it.

Chen et al. (2026) find the same effect in a resource dimension that has nothing to do with tokens. Their benchmark measures whether an agent parallelising tool calls respects infrastructure capacity, and they report that exposing resource information reduces avoidable overflows and improves resource utilisation. That is three independent lines of work, on three different resources, converging on one mechanism. The same paper publishes the finding that most constrains what follows from this, and Section 13 gives it in full: the benefit of exposure is model-dependent, so exposure is necessary and not sufficient.

Sun et al. (2026) use the same mechanism as a training device rather than an inference-time one. Their first phase trains a search agent with explicit budget-state injection and structured reasoning prompts under a budget decayed linearly from five tool calls to one; the second phase removes the scaffold entirely, leaving the agent with only the total budget stated in natural language at episode start. The scaffold is a device for making the budget observable during learning.

What this section establishes is narrow and should be stated narrowly. It does not establish that observability is the whole of budget awareness; Section 13 gives the result that rules that reading out. It establishes that an identifiable share of what is reported as an agent's failure to manage resources is a failure to be told about them, that the share is large enough to move headline numbers by an order of magnitude in budget at equal accuracy, and that this share costs no training to recover. A claim that a model lacks a self-model of its resources is not supported by a measurement taken in a harness where the resource state was never shown to it. Several of the measurements in this literature were taken in exactly such a harness, and the papers concerned say so.

6. Forecasting: what the self-estimates establish

Lin et al. (2026) give the most detailed account of forecasting located here, and its protocol matters for reading the results. They record a full agent rollout *without any budget constraint*, then re-query the same agent on every prefix of the logged trajectory, asking for an interval over remaining budget or an explicit declaration that the task is impossible. They score the predictions against the realized remaining budget and the realized outcome. Five frontier models are evaluated across four environments: Sokoban, Search-R1 and SWE-bench for internal budget, and a warehouse supply-chain environment with three coupled cost dimensions for external budget.

The protocol deliberately separates estimation from execution, and the separation is what makes the results interpretable as a measurement of forecasting rather than of control. It also bounds them. The agent being asked is not the agent that produced the trajectory under any constraint, because the trajectory was produced without one; the estimator is not in the control loop; and the paper's own limitations section lists "closing the loop so estimator predictions feed back into actor decisions beyond early stopping" as an open direction. Whatever these numbers measure, they do not measure an agent managing its own budget.

Within that scope the findings are consistent and specific.

Optimism is universal. Across all twenty model-environment pairs, optimistic misses outnumber conservative ones at every rollout-progress bin. Gemini and Qwen are the most optimistic; Sonnet and Opus are closest to calibrated and still skew low. The bias is not removed by averaging over more turns.

Optimism tracks confidence, not difficulty. Within an environment, weaker models are more optimistic about finishing, not less, which the paper notes is the opposite of what limited reasoning about hard tasks would predict.

Recognition is late. On trajectories that ultimately fail, models predict feasibility at rates above 70 percent even after 60 percent of the budget is consumed, and the prediction drops sharply only in the final 20 percent of execution.

Feasibility is a formatting problem; the interval is not. Base Qwen-7B reaches 25.5 percent feasibility-prediction accuracy on Sokoban; supervised fine-tuning alone raises it to roughly 90 percent, with no reinforcement learning needed. Interval estimation moves far less: 10.5 percent coverage at base, 26 to 53 percent after supervised fine-tuning depending on the target interval width, and 47 percent after fine-tuning followed by reinforcement learning, with a median midpoint relative error of 28 percent. The paper's own reading is that feasibility prediction was latent and needed a format, while interval estimation is bottlenecked on reasoning.

Nothing transfers. Cross-task evaluation retains only 17 to 36 percent of in-task reward. The paper's conclusion is that output format transfers but budget estimation is largely task-specific.

Bai et al. (2026) measure the same capability with a different instrument and get a compatible answer. Their agents act in the environment to produce a pre-execution cost estimate rather than being replayed over a logged trajectory, and they report weak-to-moderate correlations with realized usage, up to 0.39, with output-token usage easier to predict than input-token usage and all models systematically underestimating. They note that their own conclusion is not that the signal is useless: self-prediction provides a coarse-grained signal of relative cost that could support early budget alerts. This paper's authors overlap with those of Lin et al. (2026) -- Bai, Pei and Wang appear on both -- so the two should be treated as two measurements from overlapping groups rather than as independent replications.

Two further observations from outside this pair are worth recording because they are made in passing by papers whose subject is something else. Liu et al. (2025b) state in their limitations that their empirical analysis suggests models often underestimate their actual resource consumption. Nazi and Dipta (2026) report that models rarely honour the budgets they themselves declare even when

explicitly instructed to. Neither paper set out to measure forecasting, and both report the same sign.

Finally, the structure of the failure has a precedent. Kadavath et al. (2022) trained a value head to predict the probability that a model knows the answer to a question, found that models perform well on the distribution they are trained on, that the prediction partially generalises across tasks, and that models "struggle with calibration of $P(IK)$ on new tasks". Their ground-truth target is defined as the fraction of unit-temperature samples to a question that are correct. Four years on, the budget case reproduces the shape: trains in-distribution, generalises partially, does not stay calibrated off distribution, and is fitted to a target measured by sampling.

7. The per-instance allocation signal is a quantity measured after the fact

This is the paper's central claim, and it is a claim about provenance rather than about validity. The allocation results are not disputed here. What is disputed is the inference from them to any statement about an agent's own resource competence, and the reason is that in every case this run's search could locate, the signal that tells the allocator which instance deserves more compute is a quantity obtained by running the model on that instance and observing what happened.

The claim is restricted to *per-instance* signals, and the restriction is load-bearing. Methods exist that read an internal quantity off the forward pass and turn it into a decision with a single global threshold; those are treated separately in Section 13, because they are the real partial counterexample and they are much cheaper. What follows concerns the signal that varies from question to question.

The pattern is easiest to see laid out.

Snell et al. (2024) define question difficulty as the model's pass@1 rate estimated from 2048 samples per question, binned into five quantiles. Their oracle variant bins by ground-truth correctness; their model-predicted variant bins by a learned verifier's averaged final-answer score over the same 2048 samples. The compute-optimal strategy is then selected per difficulty bin on a validation fold.

Damani et al. (2024) train a separate lightweight predictor of the marginal reward of an extra unit of compute. Its targets come from generating eight responses per query at temperature 0.7 and scoring them with an 8B reward model. The authors state the obstacle in their own words: without querying the language model, the quantity to be maximised cannot be known, and "Monte Carlo estimation of $q(x_i, b_i)$ (as in Snell et al., 2024) may in general require more computation (e.g. more LM samples) than we eventually wish to allocate to x_i ".

Fan et al. (2026) need a per-question cost in order to define value density, and obtain it from an independent high-budget reference condition in which each question is solved alone with up to 40,960 tokens. They are scrupulous about what this is: they write that they do not interpret the resulting weight as the minimum or necessary cost of solving the question, only as the token usage observed in one independent high-budget attempt.

Zhou et al. (2026b) gate budget investment on empirical solvability derived from rollout groups under the current policy, and state the structural problem exactly: solvability and efficient cost are posterior signals, while budget investment must be committed before reasoning begins.

Nazi and Dipta (2026) score a planner's committed plan against an oracle with full knowledge of solvability and cost for every item in the pool, and normalise between that oracle and a random reference.

Nguyen et al. (2026) allocate a sampling budget using average negative log-likelihood extracted from a first-phase generation, so the difficulty signal is produced by spending part of the budget.

Lin et al. (2026) train their own budget estimators on realized budgets taken from logged rollouts.

Kadavath et al. (2022) define ground-truth $P(\text{IK})$ as the fraction of sampled answers that are correct.

The two papers that most explicitly claim a model assessing its own difficulty belong on the same list, and it is worth being exact about them because the claim in their titles points the other way. Huang et al. (2025) train a model for what they call self-assessed problem difficulty and difficulty-aware budgeting. The cold-start stage labels problems using an external dataset's own difficulty levels and filters generated trajectories against ground-truth answers; the reinforcement stage then rewards the model for emitting a difficulty tag that matches a label computed from the frequency of correct answers among sixteen rollouts sampled per prompt. Singh et al. (2025) similarly estimate task difficulty "as the pass rate during GRPO rollout" and use it to set how aggressively a reasoning trajectory is compressed. In both cases the model ends up able to emit a difficulty judgement without sampling, which is a genuine result; the judgement it was trained to emit is a target that was produced by sampling. Li et al. (2026) take a third route, scoring relative progress with a residual value predictor rather than absolute state quality, and give as their reason the well-known overconfidence of language model self-evaluation.

An obvious objection arrives here and is answered in Section 13: all supervised learning fits models to measured targets, so noticing that these targets were measured is noticing nothing. The answer is that the objection would be right if the cost of measuring the target were incidental. In these cases the authors themselves describe it as substantial, and in one case as potentially exceeding the allocation it informs, and then leave it off the axis. Snell et al. write that their experiments "do not account for this cost largely for simplicity", and note in their appendix that the procedure "is extremely costly because it requires generating many samples" and that "in a practical production setting, this cost would be problematic". Damani et al. state the same concern as a reason for building a cheaper predictor. Neither paper prices it.

One located design does put the estimation cost on the same axis, and it shows that doing so is possible rather than merely desirable. Zuo and Zhu (2025) formulate allocation as a bandit problem in which difficulty is estimated on the fly by spending compute from the same budget the allocation draws on, with no separate training stage and no out-of-band sampling pass. They report gains of up to 11.10 percentage points on MATH-500, 10.82 on AIME25 and 11.23 on LiveCodeBench over uniform allocation, and report that on a set containing both solvable and unsolvable queries their algorithms learn to prioritise the solvable subset. This is the design the rest of the literature's cost accounting would have to match, and it is the exception rather than the norm.

What follows is a scope restriction, not a refutation. The compute-optimal results establish that a *policy over strategies*, conditioned on a statistic that costs substantial compute to obtain, outperforms a fixed policy. They do not establish that the statistic can be obtained cheaply at deployment, that the model can produce it, or that an agent which produced it would thereby allocate well. Each of those is a separate empirical question, and the budget-awareness literature is the one asking it.

8. The compute-optimal objective is indexed by the answer

It is worth spending a section on Snell et al. (2024) specifically, because the definitional form of their objective states this paper's point more precisely than any commentary could.

Their compute-optimal scaling strategy is defined as the hyperparameter setting that maximises the probability that the produced output equals the ground-truth correct response, for a given prompt and compute budget. In their notation the optimal setting is written as a function of both the prompt and the ground truth. The object the literature calls optimal allocation is, as introduced, indexed by the answer.

The paper is entirely candid about the consequence. It states that assessing a question's difficulty in the

way described "assumes oracle access to a ground-truth correctness checking function, which is of course not available upon deployment where we are only given access to test prompts that we don't know the answer to", and introduces model-predicted difficulty precisely to remove that dependence. Model-predicted difficulty is a real advance on the oracle version, and the paper reports that the two give largely similar trends. It also reports where they part: in the low-budget regime both versions approach or beat best-of-N at up to four times less compute, while "in the higher budget regime, some of these benefits diminish with the use of predicted difficulty", with continued improvement visible only under oracle bins.

Model-predicted difficulty removes the ground truth but not the sampling. It bins the same 2048 samples per problem by a learned verifier's score instead of by correctness. The appendix says what this means without hedging: the procedure is technically extremely costly, the cost is not accounted for in the analysis, and in production it would be problematic. A cheaper approach, the paper suggests, would be to fine-tune a model to predict correctness directly from the question, and it explicitly leaves that to future work.

Two further scope facts about this result are worth putting next to its title, because the title travels further than the paper does.

First, the best strategy is selected per difficulty bin using two-fold cross-validation on the test set, which is a defensible way to avoid confounding difficulty estimation with strategy selection and is also an external optimisation over a menu the model never sees.

Second, the paper's own comparison against pretraining is conditional in a direction the title does not carry, and the two places it is stated do not agree. The running text says that if one expects only very difficult questions, difficulty bins four and five, or has a larger inference token count, it is often more effective to allocate budget towards pretraining, and that for mostly easy or intermediate questions, bins one to three and sometimes four, or lower inference requirements, test-time compute is better. The caption of the figure this text is describing says the opposite on the difficulty axis: "We see that on the easy questions or in settings with a lower inference load (e.g. R greater than 1), pretraining is a more effective way to improve performance." Easy in the caption and very difficult in the text cannot both be the condition under which pretraining wins. The caption is also inverted against the paper's own definition on the second axis, since the paper defines R as inference tokens divided by pretraining tokens, which makes R greater than one a higher inference load rather than a lower one. Reading the two together, the running text is the self-consistent version and the caption appears to be in error, but a reader who takes the headline claim from the caption gets the tradeoff backwards. This was checked against the v1 HTML rendering with the mathematics preserved, because an earlier pass with comparison operators stripped could not have distinguished the two readings.

The reason to record this in a paper about budget awareness is not score-keeping. It is that "scale test-time compute rather than parameters" is one of the most-repeated claims in this area, and the paper it comes from states the condition under which the claim holds in two mutually exclusive ways.

A later reanalysis sharpens the same point from inside the allocation literature. Liu et al. (2025a) examined how the optimal test-time scaling strategy depends jointly on the policy model, the process reward model used as verifier, and problem difficulty, and found that the right strategy is a function of all three rather than of difficulty alone. If the object being selected depends on which verifier is in the loop, then the selection is a property of an assembled system and not of the model whose compute is being allocated.

None of this makes the result weaker as a result. It makes it a result about something other than what an agent knows about its own spending.

9. Allocation under a known budget, when the agent must do it

The cleanest test of allocation is one where observability is not a confound and forecasting is not the object: give the model the budget, give it the competing demands, and see how it divides.

Fan et al. (2026) do this with an exam. A model receives N questions with visible point values, one shared reasoning-token budget, and freedom to decide which to attempt, in what order, and with how much effort. Questions are drawn from Omni-MATH with difficulty label at most 5; N takes the values 5, 10 and 20; five open-weight reasoning models and two DeepSeek-V4 API models are evaluated. They then recover per-question effort and solving order from the trace, using partial Spearman correlations to separate position effects from difficulty effects.

The results are unambiguous within their scope. Token effort falls with presentation position, at $\rho = -0.34$ averaged over models and exam lengths, strengthening from -0.17 at $N = 5$ to -0.38 at $N = 10$ and -0.48 at $N = 20$. The order in which questions are substantively worked on tracks presentation position at $\rho = +0.68$ overall and is stable across exam lengths ($+0.68$, $+0.66$, $+0.69$). Point values, which are visible, move almost nothing: effort-value correlations are 0.00 , $+0.04$ and $+0.11$ across the three lengths, and order-value correlations are -0.03 , -0.09 and -0.07 . Difficulty moves effort, but reactively and decreasingly: $+0.33$ at $N = 5$, $+0.26$ at $N = 10$, $+0.11$ at $N = 20$, which the paper reads as the model continuing once inside a hard question rather than deciding in advance to spend more on it.

Coverage collapses as the exam grows. Averaged over the locally deployed open models, the set of questions receiving substantive work grows from 4.0 at $N = 5$ to 8.1 at $N = 20$, so coverage falls from 80 percent to 40 percent. One API model works on 4.5 questions at $N = 10$ and 4.7 at $N = 20$, reaching 23 percent of the longer exam. At $N = 20$ the fraction of questions receiving no attributed reasoning tokens at all is 51 percent for the open-model average, and 69 and 61 percent for the two API models.

The selection test is the sharpest number in the paper. Treating the exam as a knapsack with the shared budget as capacity, and measuring at ten questions under random scoring with questions presented in ascending difficulty, the work set overlaps the highest-value-density set at 0.59, against a chance reference of 0.59, while overlapping the earliest-presented set at 0.76. Under other scoring schemes top-density overlap stays at or below chance while early-position overlap stays between 0.75 and 0.81. Questions with zero value density -- answered incorrectly even in the independent high-budget reference -- account for 24 percent of the work set and 32 percent of all reasoning tokens.

Prompting the model to plan first changes the spread and not the priorities. Coverage improves for most models, from 0.23 to 0.33 for one API model, but the solving-order correlation with prompt position moves only from 0.64 to 0.60 for the open-model average, and one API model becomes more sequential, from 0.68 to 0.83. Effort remains nearly uncorrelated with point value. The same pattern appears on code reasoning, where order tracks presentation position at $\rho = 0.99$.

Nazi and Dipta (2026) test the same capability under a different and in one respect harsher design. Their model receives a task pool, per-problem point values and a token budget calibrated to its own baseline cost on the pool, and must commit before any execution to a single ordered plan specifying which problems to attempt, how many tokens each gets, and in what order. They then run two regimes: one where the per-problem allocations are advisory and one where they are binding. Twenty configurations are evaluated across AIME 2024-25, GPQA Diamond, LiveCodeBench and Humanity's Last Exam.

Their headline is the gap between the regimes. When the allocations are advisory, efficiency varies sharply and some configurations do well on some benchmarks. When the model's own allocations are

enforced, efficiency is negative for the large majority of configurations at a budget covering half the pool's baseline cost, and a substantial fraction approach the full-regret bound, meaning essentially no value is captured. The collapse is sharpest on the reasoning-heavy benchmarks, where binding the budget cuts off solver outputs before they produce an answer. They confirm the mechanism behaviourally: models rarely honour the budgets they themselves declare, even when told to.

Their finding about extended reasoning is the one that bears hardest on the capability question. Base accuracy rises with extended reasoning on most model-dataset pairs, sometimes by a lot. Advisory regret follows no such pattern, staying unchanged or worsening on roughly half of paired configurations. What does change consistently is the advisory-to-enforced gap, which extended reasoning *enlarges*, because longer traces are less likely to fit inside the allocation the model set for itself. The paper's summary is that object-level capability and metacognitive control dissociate empirically, and that no configuration was simultaneously a strong monitor and a strong controller on more than one benchmark.

Wu et al. (2026) add a third instrument with priced actions. Their 304 tasks each specify an explicit budget and a price list over actions, and the four decisions tested are whether to escalate, when to escalate, which model tier to select, and when to stop. Tool-API agents attain 3.9 to 24.0 percent micro strict success and at most 7.3 percent on an economic-consistency score defined as the worse of upgrade-oriented and save-oriented accuracy. Two scripted always-escalate controls reach 76.3 and 73.0 percent micro success while scoring 12.2 and 0.0 percent on the save-oriented group, which is the failure the consistency score exists to expose.

Their budget sweep is the part that matters here, because it varies the budget while holding the task fixed. On 115 tasks requiring deep research, one model's escalation rate moves from 0 percent to 3 percent as the budget crosses the escalation path's cost. On 72 tasks already solvable with cheap evidence, all three backbones spend *more* as the budget grows, and one raises unnecessary deep-research use to 19.4 percent at the loose budget. Budget conditioning is therefore present and pointed the wrong way: the models respond to a larger budget by spending more where spending was unnecessary, and do not respond by spending where spending was required. The authors' own summary is that completion, budget feasibility and conditional resource selection are distinct capabilities, and that low expenditure can reflect premature stopping just as high completion can conceal indiscriminate escalation. Their protocol is single-shot at temperature zero with one episode per task, and they say so.

A fourth instrument narrows where the remaining headroom is. McCleary and Ghawaly (2026) hold the budget fixed and sweep design decisions instead, running six models over three question-answering benchmarks under explicit caps on tool calls and completion tokens. They report that accuracy improves with additional searches only up to a small cap, that hybrid retrieval with light re-ranking gives the largest average gains in their grid, and that larger completion budgets help most on synthesis-style questions. Read against the preceding results, this says that the configuration space around the agent carries a substantial share of what a good allocation would buy, which is consistent with the agent itself contributing little.

One more result belongs in this section because it removes an easy dismissal. Wang et al. (2024) asked whether the reasoning strategies the field was proposing beat simple baselines once compute is held equal, and found that chain-of-thought with self-consistency frequently outperforms multi-agent debate, Reflexion, Plan-and-Solve, Least-to-Most and Progressive Hints when the budget is matched. Their conclusion is that the observed improvements are primarily attributable to increased budget allocations rather than to the inherent advantages of the methods. They also report that some strategies get *worse* with more budget, and give a mechanism: multi-agent debate conditions each round on the previous round's answers, and the entropy of its solutions declines across rounds while self-consistency's rises. So even on the allocation side, part of what looked like better allocation was more compute.

10. Adherence is largely solved, and that is evidence for the partition

The one decision of the four with a clean positive answer is the one where nothing is being estimated.

Aggarwal and Welleck (2025) train a model with a reward penalty for deviating from a user-specified target length, and report that observed output lengths usually closely match requested lengths across targets of 512, 1024, 2048 and 3600 tokens. Against the prior budget-forcing approach of Muennighoff et al. (2025), they report gains of over 100 to 150 percent relative and 20 to 25 percent absolute at the 512- and 1024-token budgets, and attribute the difference to adapting the chain of thought to fit rather than truncating it mid-reasoning. Wen et al. (2025) report superior adherence from control tokens that carry the remaining budget. Sun et al. (2026) report a single policy that operates across budget levels and generalises to budgets outside its training range.

Two caveats belong with this. First, the ranges are small. Aggarwal and Welleck evaluate up to 3600 thinking tokens, and Alomrani et al. (2025) note that it is unclear whether the method generalises to larger context sizes. Sun et al. train with a maximum of five search calls. Second, adherence to an externally given bound is not the same as adherence to a self-set bound, and on the latter the evidence is negative: Nazi and Dipta (2026) find that models rarely honour allocations they declared themselves.

That second caveat is where adherence and allocation interact, and it is worth being precise about the interaction because it complicates this paper's own partition. In the enforced regime of Nazi and Dipta, a plan can fail for two reasons: the plan selected the wrong problems, or the plan selected the right problems and gave them budgets the model could not work within. These are an allocation failure and an adherence failure respectively, and the reported efficiency score does not separate them. Their observation that extended reasoning enlarges the advisory-enforced gap is consistent with the second reading dominating, since longer traces overrun self-set allocations more often. If that is what is happening, then part of what reads as an allocation failure in that benchmark is an adherence failure against a target the model itself produced, and the two decisions are not as separable in measurement as they are in definition. Section 15 proposes a re-scoring that would tell them apart.

11. The correlation the argument rests on, and what it is between

The popular one-line version of the budget-awareness result is that capability and budget-awareness come apart, with a correlation of about 0.35. That number deserves a paragraph of its own, because in its source it is attached to three different pairs of quantities.

Lin et al. (2026) write in their abstract and introduction that "Task success correlates only weakly with interval hit rate (r approximately 0.35)". In Section 4.1 they write that "task success rate correlates only weakly with feasibility prediction (r approximately 0.35)". In Section 4.2 they write that interval hit rate correlates strongly with midpoint bias and width adequacy "but only weakly with feasibility F1 (r approximately 0.35)". Task success against interval hit rate, task success against feasibility prediction, and interval hit rate against feasibility prediction are three different correlations among three quantities, and the same rounded value is reported for each. All three could in principle round to 0.35, and the paper does not say they do. What is clear is that the abstract and Section 4.1 attach the number to the same sentence-level finding with different second variables, which is a transcription difference rather than three separate computations, and that a reader cannot tell from the paper which pair the coefficient belongs to.

This is not a reason to discard the finding. The dissociation it names is supported in the same paper by results that do not depend on the coefficient: on Search-R1, the model with the highest task success (75.8 percent) produces worse intervals than a model with lower success (23.1 against 36.5 percent hit rate); on SWE-bench the rankings split three ways across task success, feasibility prediction and interval estimation; on Sokoban the model with the highest binary feasibility F1 (61.9 percent) has the lowest

interval hit rate (8.8 percent). It is supported independently by Nazi and Dipta (2026), Fan et al. (2026), Wu et al. (2026) and Chen et al. (2026), none of whom share authors with Lin et al. and none of whom use the same instrument.

It is a reason to stop quoting the coefficient. A paper that cites $r = 0.35$ for the proposition that capability and budget-awareness come apart is citing a number whose referent is ambiguous in its own source, and the claim does not need it. The qualitative dissociation is the finding; the coefficient is bookkeeping that did not survive the paper's own editing.

One related fact should be recorded in the same spirit. Lin et al. also report that their agents' midpoint estimates beat a deterministic linear-extrapolation baseline on warehouse occupancy in most model-progress bins, especially early in the rollout, while for cumulative cost extrapolation "often remains competitive". So the agents' forecasts are not uniformly worse than arithmetic on the trajectory so far; they are better on one budget dimension and not clearly better on another, in the one environment where the comparison was run. That is a more interesting result than the headline and it is reported in a single paragraph.

12. The variance nobody subtracts

Every forecasting number in Section 6 is a number without a denominator.

Bai et al. (2026) report, as a property of agentic coding tasks, that runs on the same task can differ by up to 30 times in total tokens. They report it to make the point that token usage is highly variable and inherently stochastic, and they connect it to context construction, retrieval and tool-driven exploration rather than to reasoning length; the same paper notes that output tokens are easier to predict than input tokens, which is consistent with the variance living mostly in what the environment returns.

Now put that beside the forecasting scores. Lin et al. (2026) report 47 percent interval coverage after supervised fine-tuning followed by reinforcement learning, with a median midpoint relative error of 28 percent, and treat the remaining gap as the central open problem. Bai et al. report self-prediction correlations up to 0.39. Neither number is stated relative to the spread of the quantity being predicted.

The missing comparison is not exotic. For a given task, environment and decoding configuration, the realized budget has a distribution. An oracle that knew that distribution exactly, but not which run it was about to observe, would achieve some coverage at a given interval width, and that coverage is strictly less than one whenever the distribution is wide relative to the interval. That number is the ceiling. A forecaster at 47 percent coverage is performing well or badly depending entirely on whether the ceiling is 55 percent or 95 percent, and nothing located by this run's search reports the ceiling for any of these environments.

The consequences run in both directions, which is why the omission matters rather than merely being untidy.

If the ceiling is low, then a substantial share of the reported forecasting failure is not a failure of self-knowledge. It is the agent being scored on predicting a quantity that its own stochastic policy and a stochastic environment jointly determine, and the correct conclusion would be that budget intervals should be wide and that narrow intervals are the wrong target. Lin et al.'s own width ablation is suggestive on this point and is not read this way in the paper: training with narrow targets, at 10 percent of remaining budget, yields 26 percent coverage at 45 percent midpoint relative error, while training with wide fixed targets of 1,000 tokens yields over 90 percent coverage at 170 percent error. Coverage is purchasable with width. Whether the width required to reach high coverage is the width the trajectory's own variance demands, or is instead evidence of a bad point estimate, is exactly what the missing ceiling would tell.

If the ceiling is high, then the forecasting deficit is real and larger than the raw numbers suggest, because a well-behaved quantity is being predicted badly. That would strengthen the papers' own conclusions rather than weakening them.

Either way, the number that would decide it is not in the literature located here. The closest adjacent result is Oladri et al. (2026), who study a related form of predictability: on a chain-of-thought model answering AIME problems, generations either terminate within the budget or exhaust it, with converged generations reaching 90.3 percent accuracy and non-converged ones 6.6 percent, at an overall convergence rate of 62.0 percent. They then ask how early this outcome is detectable from internal state, and report that linear probes on layer-20 activations at token 150 reach an area under the ROC curve reported as 0.608 plus or minus 0.080 under five-fold cross-validation, above chance even at token 50. That is a measured limit on one internal signal for one binary version of the question, on one model, and it is a limit on that probe rather than on the predictability of the outcome. It is the only such number this search found, and it is weak enough to be worth knowing.

13. The case against this paper's reading

Six objections, in descending order of how much they cost, and a note on selection.

Exposing the budget is not enough, and one paper shows it. Chen et al. (2026) supply both halves of the observability argument and then close the door on the strong version of it. They report that exposing resource information reduces avoidable overflows and improves utilisation, and they also report that the benefit remains model-dependent, concluding that resource-aware orchestration is "a distinct capability rather than an automatic consequence of exposing resource metadata". They further report that models without resource profiles are not merely greedy schedulers: they often add conservative delays that reduce capacity violations at substantially higher scheduling latency, so the no-information behaviour is not simply absence of the capability. Section 5 is therefore correct in the weak form -- a share of the failure is observability -- and wrong in any strong form that treats budget awareness as reducible to a field in a prompt. This paper adopts the weak form.

Budget behaviour can be internalised, and one paper shows that too. Sun et al. (2026) train with an explicit budget-state scaffold, remove it, and report that the resulting policy achieves the highest accuracy across budget levels with the scaffold absent, that re-enabling the scaffold at inference yields virtually identical accuracy, and that the scaffold without reinforcement learning is ineffective. Their own reading is that if budget awareness were not internalised, re-enabling the scaffold should produce a measurable gain, and it does not. That is a well-designed ablation and it is direct evidence for the "separate trainable competence" horn of the original question. Two scope limits belong with it, and the paper states both: the budget is a discrete count of external search calls with a training maximum of five, and training and evaluation use a static 2018 Wikipedia dump. Whether the same internalisation holds for a token budget spanning three orders of magnitude, on a live environment, is untested.

Capability sometimes does help. Wu et al. (2026) report that the stronger backbone improves micro success within both command-line agent systems they tested, Opus over Haiku and one GPT tier over a smaller one, and that the strongest configuration attains the highest economic-consistency score among their evaluated agents, at 54 percent on the upgrade-oriented group and 96 percent on the save-oriented group. Capability and resource competence are not orthogonal everywhere. The dissociation results establish that the relation is weak and non-monotone across models and instruments, not that it is absent.

An internal signal plus one global threshold is a real counterexample. Nogueira et al. (2025) periodically interrupt generation to decode a short probe answer and compute a certainty score from the model's predicted probabilities over the answer tokens, terminating once the score crosses a

threshold. The per-instance signal here is internal, costs a probe decode capped at four tokens in their implementation, and requires no separate predictor, no reward model and no sampling pass. What is fitted to measured outcomes is one scalar: they sweep the threshold over four values and report that the most conservative of them stays within 1.1 percent relative of baseline accuracy while saving over two million tokens in aggregate on AIME2025. That is a different and far weaker dependence on measurement than 2048 samples per question, and Section 7 is restricted to per-instance signals for exactly this reason. What the design does not do is forecast: a stopping rule that fires when certainty is high says nothing about how much budget the task will need, and Nogueira et al. do not claim otherwise. The same observation applies to Zuo and Zhu (2025), whose bandit pays for its own exploration from the budget it allocates.

"Measured after the fact" describes all supervised learning. Section 7's pattern could be restated as the observation that predictors are fitted to labels, which is not a finding. The answer given there is that the authors themselves describe the cost of producing the label as substantial relative to what the prediction saves, and then do not price it. The objection can be pressed further: if a cheap predictor could be distilled from expensive labels once, offline, the deployment cost would be amortised and the criticism would evaporate. That is a real possibility and Snell et al. name it as future work. It has not been demonstrated in anything located here, and it would have to survive the transfer results, which are poor: 17 to 36 percent cross-task reward retention in Lin et al. (2026), strong self-assessment in one domain rarely carrying to another in Nazi and Dipta (2026), and poor off-distribution calibration in Kadavath et al. (2022).

The partition may be less clean than presented. Section 10 already concedes the sharpest instance: in an enforced-allocation regime, allocation failures and adherence failures against a self-set target are not separated by the reported score. There is a second instance. Observability and forecasting interact, because an agent shown its remaining budget at every step is being handed exactly the input a forecaster would need, and the improvement from Budget Tracker might be an improvement in implicit forecasting rather than a bypass of it. Nothing located here separates those. The partition is offered as an analytic tool that makes the literature's results commensurable, not as a claim that the four decisions are empirically independent.

A note on selection. Everything above was found through one index on one day, under the conditions stated in Section 2. The claim that the out-of-band allocation papers do not price their estimator's compute is a claim about a search, and it is written that way wherever it appears. One counterexample to the broader pattern was already found inside this search and is reported in Section 7 rather than buried here; more would be more useful than this paper.

14. What is not known

Stated flatly, because hedged ignorance reads as evasion.

The forecasting ceiling is unmeasured. No paper located here reports the run-to-run distribution of realized budget on a fixed task under fixed conditions, and no budget forecaster is scored against it. The one variance figure found, up to 30 times on SWE-bench Verified, is reported as a property of agents rather than as a bound on forecasters.

The observability contribution is not isolated. No located result holds a policy fixed and varies only whether the remaining budget is present in the context. The nearest is the Budget Tracker arm of Liu et al. (2025b), which changes the prompt and the agent's decision path together.

The estimator's compute is usually not on the axis. No located comparison in the out-of-band family charges the cost of obtaining the allocation signal against the compute the allocation saves. Snell et al. state that they do not account for it; Damani et al. state that it may exceed what is being allocated. Both

leave it unpriced. Zuo and Zhu (2025) do pay for estimation from the allocated budget, which makes their comparison cost-honest by construction, but no located work compares an out-of-band estimator against an in-budget one on the same task at the same total cost, so which family wins under honest accounting is unknown.

Cross-modality transfer of budget forecasting is unmeasured. Lin et al. (2026) measure cross-task transfer within a modality and find 17 to 36 percent retention. Whether an estimator trained on tokens transfers to tool calls, money or wall-clock on the same trajectories has not been measured, despite their warehouse environment carrying three coupled dimensions.

Whether optimism is a property of the model or of the elicitation is unknown. Lin et al. elicit forecasts by replaying logged prefixes; Bai et al. elicit them by letting an agent act in the environment before predicting. Both find underestimation. Neither varies the elicitation while holding the model fixed, so the common finding could be a common property of models or a common property of asking.

The disagreement rate between internal and external difficulty estimates is unmeasured. An agent's own feasibility judgement and an external solvability or difficulty estimator are quantities that would often be available for the same instance, and no located work reports how often they disagree, or which is right when they do.

Whether budget forecasting improves budget allocation is untested. This is the one the field's own limitation sections name. Lin et al. list closing the loop between estimator and actor as future work. Liu et al. state that they do not explore how an agent should allocate its resources. The intuition that a better forecast produces a better allocation is stated in several places and tested in none of them located here.

15. What would settle the open part

Six experiments. Five of them can be run on data or environments that already exist, which is stated per item because it changes who can run them.

(1) The variance floor. Re-run a fixed agent on a fixed task k times under fixed decoding, record the empirical distribution of realized budget, and compute the coverage attained by an oracle that knows the distribution but not the run. Report every forecaster's coverage as a fraction of that oracle's. This converts "47 percent coverage" into a statement with a denominator, and it would resolve whether narrow intervals are the wrong target. Runnable on the released environments of Lin et al. (2026) with no new task authoring; the cost is k times one rollout suite.

(2) The single-bit observability ablation. Hold weights, prompt template and decoding fixed; vary exactly one thing, whether the remaining budget appears in the observation at each step; measure behaviour and accuracy. The Budget Tracker result is close to this and changes more than one thing at once. Runnable inside any existing agent harness.

(3) Cost-accounted allocation curves. Re-plot the compute-optimal results with the estimator's own generation cost added to the horizontal axis. For Snell et al. this means adding the 2048 samples per problem; for Damani et al. the eight responses per query plus reward-model scoring; for Huang et al. and Singh et al. the sixteen and N rollouts per prompt their difficulty labels require during training, amortised over deployment volume. The question is whether any of the reported advantages survive, and at what deployment scale the amortisation pays. The arithmetic is runnable from the published experimental descriptions, for the sample counts the papers state, without re-running any model. A second and harder version compares an out-of-band estimator against the in-budget bandit of Zuo and Zhu (2025) at matched total cost on one benchmark, which would establish which family is actually ahead.

(4) The modality-transfer matrix. Train a budget estimator on one budget dimension and evaluate it on the others over the same trajectories. Lin et al.'s warehouse environment already carries money, time and occupancy as coupled dimensions on a single rollout, so the matrix can be filled without new data collection. The interesting cell is whether an estimator that learned token consumption has learned anything about monetary commitment.

(5) The elicitation ablation for optimism. Hold the model fixed and vary only how the forecast is requested: replay over a logged prefix, a pre-execution agentic probe, and an in-loop control token. If optimism survives all three, it is a property of the model; if it does not, part of the reported bias belongs to the protocol. Requires re-eliciting, not re-training.

(6) Separating allocation failure from self-adherence failure. Re-score the enforced regime of Nazi and Dipta (2026) twice: once as published, and once with each per-problem allocation raised to the nearest budget the model is measured to honour for that problem class. The difference between the two scores is the share of the enforced collapse attributable to the model failing its own target rather than choosing the wrong targets. Requires the published plans and one calibration pass, not a new benchmark.

An honest note on which of these is worth doing first. Experiment (1) is the one this paper would run, because every other number in the forecasting literature is currently uninterpretable without it, and because it is the cheapest of the six.

16. Conclusion

The question this paper started from was whether budget-awareness in language model agents is a separate trainable competence or a symptom of the same absent self-model that shows up elsewhere. The answer available from the published evidence is that the question is underdetermined as asked, because the phrase names four decisions with four different answers.

Adherence to an externally set budget is largely solved in the ranges tested. Observability is not a competence and accounts for a share of the reported failure large enough to move headline numbers by an order of magnitude in budget at equal accuracy; the fix is a field in a prompt, refreshed each step, and three independent lines of work found it separately. Forecasting is poor, systematically optimistic, easy to train for the binary case, hard to train for the interval, and does not transfer across tasks. Allocation fails on an axis of its own: under a shared budget, effort follows presentation order rather than value, and when a model's self-set allocations are enforced it captures essentially nothing.

The apparent conflict with the test-time compute literature does not survive the partition. Those results are about a policy over strategies conditioned on a statistic, and in every case located here the per-instance form of that statistic was obtained by running the model and measuring the outcome -- 2048 samples per question, eight responses scored by a reward model, sixteen rollouts per prompt, a 40,960-token reference run, an oracle that knows solvability and cost. The papers that built these estimators say this plainly: one states that estimating the quantity may cost more compute than the allocation it informs, and another that its estimation procedure would be problematic in production, immediately after saying that its analysis does not charge for it. One located design, an in-budget bandit, does pay for its own estimation from the budget it allocates, and it is the exception. Until the out-of-band family is compared against it at matched total cost, "intelligent allocation beats brute-force scaling" is a claim about what an external estimator with a measurement budget can do, and it licenses nothing about what an agent can do for itself. A reader tracing that claim to its most-cited source should also know that the source states the condition under which it holds in two incompatible ways, once in running text and once in the caption of the figure that text describes.

What the evidence does establish, and what deserves to travel further than the correlation coefficient

everyone quotes, is a dissociation reported by five groups using five instruments: the best solver is not the best estimator, and extended reasoning improves the first while leaving or making worse the second. That is a real finding and it does not need a number that its own source attaches to three different pairs of quantities.

The most useful thing that could be added to this literature is also the cheapest. The quantity these agents are being asked to forecast varies by up to thirty times across runs of the same task. Nothing located by this run's search measures how much of it is forecastable in principle. Every reported coverage figure is a numerator without a denominator, and a field that has spent two years establishing that agents predict their own spending badly does not yet know how well it could be predicted at all.

Acknowledgements

This paper is a literature analysis and depends entirely on measurements made and published by others. Several of its load-bearing observations are quotations of authors stating limitations of their own results: Snell et al., who write that their difficulty estimate assumes oracle access to ground-truth correctness and that its cost would be problematic in production; Damani et al., who state that estimating the quantity they allocate on may cost more than the allocation; Liu et al., who record in their limitations that models often underestimate their resource consumption and that they did not study allocation; Zhou et al., who state that solvability is a posterior signal while the investment must be committed first; Fan et al., who decline to interpret their own cost proxy as a necessary cost; and Chen et al., who publish the result that most constrains the reading their own paper otherwise supports. An analysis of this kind is only possible because those sentences were written.

References

- Agarwal, A., Sengupta, A., & Chakraborty, T. (2025). The Art of Scaling Test-Time Compute for Large Language Models. arXiv:2512.02008. doi:10.48550/arXiv.2512.02008. [PREPRINT]
- Aggarwal, P., & Welleck, S. (2025). L1: Controlling How Long A Reasoning Model Thinks With Reinforcement Learning. arXiv:2503.04697. doi:10.48550/arXiv.2503.04697. [PREPRINT]
- Alomrani, M. A., Zhang, Y., Li, D., Sun, Q., Pal, S., Zhang, Z., Hu, Y., Ajwani, R. D., Valkanas, A., Karimi, R., Cheng, P., Wang, Y., Liao, P., Huang, H., Wang, B., Hao, J., & Coates, M. (2025). Reasoning on a Budget: A Survey of Adaptive and Controllable Test-Time Compute in LLMs. arXiv:2507.02076. doi:10.48550/arXiv.2507.02076. [PREPRINT]
- Bai, L., Huang, Z., Wang, X., Sun, J., Mihalcea, R., Brynjolfsson, E., Pentland, A., & Pei, J. (2026). How Do AI Agents Spend Your Money? Analyzing and Predicting Token Consumption in Agentic Coding Tasks. arXiv:2604.22750. doi:10.48550/arXiv.2604.22750. [PREPRINT]
- Chen, X., Xu, J., Liang, T., He, Z., Pang, J., Yu, D., Song, L., Liu, Q., Zhou, M., Zhang, Z., Wang, R., Tu, Z., Mi, H., & Yu, D. (2024). Do NOT Think That Much for $2 + 3 = ?$ On the Overthinking of o1-Like LLMs. arXiv:2412.21187. doi:10.48550/arXiv.2412.21187. [PREPRINT]
- Chen, Z.-K., Zhong, X.-X., Li, S.-Y., Zhan, D.-C., & Ye, H.-J. (2026). PeakBench: Benchmarking Resource-Aware Tool Invocation in LLM Agents. arXiv:2608.24509. doi:10.48550/arXiv.2608.24509. [PREPRINT]
- Cuadron, A., Li, D., Ma, W., Wang, X., Wang, Y., Zhuang, S., Liu, S., Schroeder, L. G., Xia, T., Mao, H., Thumiger, N., Desai, A., Stoica, I., Klimovic, A., Neubig, G., & Gonzalez, J. E. (2025). The Danger of Overthinking: Examining the Reasoning-Action Dilemma in Agentic Tasks. arXiv:2502.08235. doi:10.48550/arXiv.2502.08235. [PREPRINT]
- Damani, M., Shenfeld, I., Peng, A., Bobu, A., & Andreas, J. (2024). Learning How Hard to Think: Input-Adaptive Allocation of LM Computation. arXiv:2410.04707. doi:10.48550/arXiv.2410.04707. [PREPRINT]
- Erol, M. H., El, B., Suzgun, M., Yuksekgonul, M., & Zou, J. (2025). Cost-of-Pass: An Economic Framework for Evaluating Language Models. arXiv:2504.13359. doi:10.48550/arXiv.2504.13359. [PREPRINT]
- Fan, C., Cheng, Y., Li, M., Liang, Y., Zhou, T., & Feizi, S. (2026). Thinking Hard, Not Smart: Reasoning Models Fail to Ration Test-Time Compute Across Questions. arXiv:2608.07968. doi:10.48550/arXiv.2608.07968. [PREPRINT]

Gema, A. P., Hagele, A., Chen, R., Arditi, A., Goldman-Wetzler, J., Fraser-Taliente, K., Sleight, H., Petrini, L., Michael, J., Alex, B., Minervini, P., Chen, Y., Benton, J., & Perez, E. (2025). Inverse Scaling in Test-Time Compute. arXiv:2507.14417. doi:10.48550/arXiv.2507.14417. [PREPRINT]

Han, T., Wang, Z., Fang, C., Zhao, S., Ma, S., & Chen, Z. (2024). Token-Budget-Aware LLM Reasoning. arXiv:2412.18547. doi:10.48550/arXiv.2412.18547. [PREPRINT]

Huang, S., Wang, H., Zhong, W., Su, Z., Feng, J., Cao, B., & Fung, Y. R. (2025). AdaCtrl: Towards Adaptive and Controllable Reasoning via Difficulty-Aware Budgeting. arXiv:2505.18822. doi:10.48550/arXiv.2505.18822. [PREPRINT]

Iacobacci, I., Qian, Z., AL-Tam, F., AL-Qurishi, M., & Souissi, R. (2025). Increasing the Thinking Budget is Not All You Need. arXiv:2512.19585. doi:10.48550/arXiv.2512.19585. [PREPRINT]

Kadavath, S., Conerly, T., Askell, A., Henighan, T., Drain, D., Perez, E., Schiefer, N., Hatfield-Dodds, Z., DasSarma, N., Tran-Johnson, E., Johnston, S., El-Showk, S., Jones, A., Elhage, N., Hume, T., Chen, A., Bai, Y., Bowman, S., Fort, S., Ganguli, D., Hernandez, D., Jacobson, J., Kernion, J., Kravec, S., Lovitt, L., Ndousse, K., Olsson, C., Ringer, S., Amodei, D., Brown, T., Clark, J., Joseph, N., Mann, B., McCandlish, S., Olah, C., & Kaplan, J. (2022). Language Models (Mostly) Know What They Know. arXiv:2207.05221. doi:10.48550/arXiv.2207.05221. [PREPRINT]

Kapoor, S., Stroebel, B., Siegel, Z. S., Nadgir, N., & Narayanan, A. (2024). AI Agents That Matter. arXiv:2407.01502. doi:10.48550/arXiv.2407.01502. [PREPRINT]

Kirichenko, P., Ibrahim, M., Chaudhuri, K., & Bell, S. J. (2025). AbstentionBench: Reasoning LLMs Fail on Unanswerable Questions. arXiv:2506.09038. doi:10.48550/arXiv.2506.09038. [PREPRINT]

Li, J., Zhao, W., Zhang, Y., & Gan, C. (2025). Steering LLM Thinking with Budget Guidance. arXiv:2506.13752. doi:10.48550/arXiv.2506.13752. [PREPRINT]

Li, Y., Deng, W., Li, J., & Li, X. (2026). Spend Less, Reason Better: Budget-Aware Value Tree Search for LLM Agents. arXiv:2603.12634. doi:10.48550/arXiv.2603.12634. [PREPRINT]

Lin, Y., Wang, Z., Liu, M., Shan, Y., Bai, L., Zhang, J., Jin, X., Chen, B., Su, J., Wang, X., Pei, J., & Li, M. (2026). BAGEN: Are LLM Agents Budget-Aware? arXiv:2606.00198. doi:10.48550/arXiv.2606.00198. [PREPRINT]

Liu, R., Gao, J., Zhao, J., Zhang, K., Li, X., Qi, B., Ouyang, W., & Zhou, B. (2025a). Can 1B LLM Surpass 405B LLM? Rethinking Compute-Optimal Test-Time Scaling. arXiv:2502.06703. doi:10.48550/arXiv.2502.06703. [PREPRINT]

Liu, T., Wang, Z., Miao, J., Hsu, I.-H., Yan, J., Chen, J., Han, R., Xu, F., Chen, Y., Jiang, K., Daruki, S., Liang, Y., Wang, W. Y., Pfister, T., & Lee, C.-Y. (2025b). Budget-Aware Tool Use Enables Effective Agent Scaling. arXiv:2511.17006. doi:10.48550/arXiv.2511.17006. [PREPRINT]

Mahendrakar, P. M. (2026a). Not Acting Is Not One Decision: Two Abstention Triggers in Tool-Using Agents, Why the Calibration and Permission Readings Cover Different Ones, and the Cost Term Neither Supplies. Life of Research. doi:10.5281/zenodo.22803148

Mahendrakar, P. M. (2026b). Predicting Your Own Failure Is Not Predicting Another Model's Advantage: What Router Results Establish About Agent Self-Knowledge, and the Incremental-Validity Test Nobody Has Run. Life of Research. doi:10.5281/zenodo.22288852

McCleary, K., & Ghawaly, J. (2026). Quantifying the Accuracy and Cost Impact of Design Decisions in Budget-Constrained Agentic LLM Search. arXiv:2603.08877. doi:10.48550/arXiv.2603.08877. [PREPRINT]

Muennighoff, N., Yang, Z., Shi, W., Li, X. L., Fei-Fei, L., Hajishirzi, H., Zettlemoyer, L., Liang, P., Candes, E., & Hashimoto, T. (2025). s1: Simple test-time scaling. arXiv:2501.19393. doi:10.48550/arXiv.2501.19393. [PREPRINT]

Nazi, Z. A., & Dipta, S. R. (2026). TRIAGE: Evaluating Prospective Metacognitive Control in LLMs under Resource Constraints. arXiv:2605.13414. doi:10.48550/arXiv.2605.13414. [PREPRINT]

Nguyen, M., Gupta, S., & Le, H. (2026). Uncertainty-Aware Budget Allocation for Adaptive Test-Time Reasoning. arXiv:2605.26849. doi:10.48550/arXiv.2605.26849. [PREPRINT]

Nogueira, J. P., Sun, W., Silva, A., & Zumot, L. (2025). Certainty-Guided Reasoning in Large Language Models: A Dynamic Thinking Budget Approach. arXiv:2509.07820. doi:10.48550/arXiv.2509.07820. [PREPRINT]

Oladri, R., Jawahar, N., & Mohamed, A. (2026). Token Budget Saturation and Mechanistic Early Detection of Reasoning Non-Convergence in Chain-of-Thought Models. arXiv:2607.21433. doi:10.48550/arXiv.2607.21433. [PREPRINT]

Shen, J., Bai, H., Zhang, L., Zhou, Y., Setlur, A., Tong, S., Caples, D., Jiang, N., Zhang, T., Talwalkar, A., & Kumar, A.

- (2025). Thinking vs. Doing: Agents that Reason by Scaling Test-Time Interaction. arXiv:2506.07976. doi:10.48550/arXiv.2506.07976. [PREPRINT]
- Singh, J., Chen, J. C.-Y., Prasad, A., Stengel-Eskin, E., Nambi, A., & Bansal, M. (2025). Think Right: Learning to Mitigate Under-Over Thinking via Adaptive, Attentive Compression. arXiv:2510.01581. doi:10.48550/arXiv.2510.01581. [PREPRINT]
- Snell, C., Lee, J., Xu, K., & Kumar, A. (2024). Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. arXiv:2408.03314. doi:10.48550/arXiv.2408.03314. [PREPRINT]
- Sui, Y., Chuang, Y.-N., Wang, G., Zhang, J., Zhang, T., Yuan, J., Liu, H., Wen, A., Zhong, S., Zou, N., Chen, H., & Hu, X. (2025). Stop Overthinking: A Survey on Efficient Reasoning for Large Language Models. arXiv:2503.16419. doi:10.48550/arXiv.2503.16419. [PREPRINT]
- Sun, X., Li, J., Hong, Y., Fu, Y., & Xiao, Y. (2026). One Policy, Any Budget: Internalizing Budget-Aware Search via Reinforcement Learning. arXiv:2609.00813. doi:10.48550/arXiv.2609.00813. [PREPRINT]
- Wang, J., Jain, S., Zhang, D., Ray, B., Kumar, V., & Athiwaratkun, B. (2024). Reasoning in Token Economies: Budget-Aware Evaluation of LLM Reasoning Strategies. arXiv:2406.06461. doi:10.48550/arXiv.2406.06461. [PREPRINT]
- Wen, H., Wu, X., Sun, Y., Zhang, F., Chen, L., Wang, J., Liu, Y., Liu, Y., Zhang, Y.-Q., & Li, Y. (2025). BudgetThinker: Empowering Budget-aware LLM Reasoning with Control Tokens. arXiv:2508.17196. doi:10.48550/arXiv.2508.17196. [PREPRINT]
- Wu, J., Gong, M., Cheng, F., & Zhao, Q. (2026). EcoAgent-Bench: Evaluating Economic Decision-Making in Budget-Constrained LLM Agents. arXiv:2608.05519. doi:10.48550/arXiv.2608.05519. [PREPRINT]
- Zhou, S., Ling, R., Chen, J., Wang, X., Fan, T., & Wang, H. (2026a). When More Thinking Hurts: Overthinking in LLM Test-Time Compute Scaling. arXiv:2604.10739. doi:10.48550/arXiv.2604.10739. [PREPRINT]
- Zhou, Z., Zhang, L., Wang, J., Yuan, M., & Lin, J. (2026b). Nice Fold or Hero Call: Learning Budget-Efficient Thinking for Adaptive Reasoning. arXiv:2605.11625. doi:10.48550/arXiv.2605.11625. [PREPRINT]
- Zuo, B., & Zhu, Y. (2025). Strategic Scaling of Test-Time Compute: A Bandit Learning Approach. arXiv:2506.12721. doi:10.48550/arXiv.2506.12721. [PREPRINT]

(c) 2026 Pranay Mahendrakar. Licensed under CC BY 4.0.

The literature search, drafting and citation verification for this paper were carried out with AI assistance under the author's direction. Every citation was machine-verified against the DataCite record for its DOI before inclusion, and every quantitative claim was read back against the cited source's own table or text before it was written down. No experiment was run and no number in this paper was measured or recomputed by its author; every figure is quoted from the paper credited with it, and where two figures from one source are set side by side to make a point that the source does not itself make, the juxtaposition is flagged in the text. Section 2 states the search procedure and its limits, including the fact that one of the two usual metadata routes was unavailable on the day of the search, so that the coverage claims in Sections 7, 13 and 14 can be checked and, if wrong, corrected. The author is responsible for the final text and for all claims made in it.