# Prompting Strategies for Conceptual Ideation in Problem Solving

**Anonymous ACL submission**


## Abstract

LLMs have improved at executing well-specified chains of reasoning but remain weak at *conceptual ideation*, which identifies what is worth reasoning about in the first place. We cast this as bilevel optimization over (i) a compact hypothesis space $\Theta \subseteq \Theta_0$ and (ii) a point $\theta \in \Theta$, and derive a domain-agnostic prompting strategy — GENERATE-$\Theta$ — that makes hypothesis-space construction an explicit pre-solving step. On two challenging puzzle domains, GENERATE-$\Theta$ matches or exceeds the max-thinking variant of the same base model (Claude Opus 4.6 or DeepSeek-v3.2/v4) and outperforms reasonable prompting baselines. Results may provide insight into strategies learned by "reasoning" models during training. Our prompting strategy could be integrated with RL training to produce better exploratory data for reasoning models.


## 1 Introduction

Scientific progress, and ambitious problem solving, often requires *conceptual ideation*: looking at a problem that does not fit a familiar template and proposing a tractable formulation for it. Galois proved general results about algebraic solutions to polynomials by inventing new abstractions that revealed hidden symmetries and groups, leading to group theory. Maxwell invented classical electrodynamics by imagining a hypothetical machine that could explain the known interactions between electricity and magnetism, leading to field theory. In these and other cases, finding the right formulation was the principle breakthrough.

Our work here builds on important results in cognitive science. For instance, our characterization of the problem is consistent with Simon (Simon, 1973), who imagined an outer loop on his General Problem Solver (Newell and Simon, 1972) that retrieved relevant structures from a knowledge bank and "noticed" ways the problem statement should be revised in light of the new information. Cognitive historical work provides a complementary view onto conceptual ideation strategies that lead to breakthroughs in practice. Nersessian's analyses of how scientists and skilled problem-solvers bootstrap novel problem formulations from old ones suggest an iterative process based on analogy and abstraction (Nersessian, 2008). Relatedly, a large body of work has explored both qualitative and formal algorithmic accounts of analogical structure mapping for constructing problem representations (Gentner, 1983; Hofstadter and Sander, 2013; Mitchell, 2021; Hofstadter et al., 1995). Research on "insight" problem solving has examined the mechanisms that both cause and alleviate impasses for the solver on tricky problems, which involve problem formulation and reformulation (Ohlsson, 1992; Knoblich et al., 1999).

These lines of work have only begun to make contact with research on problem-solving in LLMs (Kargupta et al., 2025). Current LLM-based agents can be remarkably strong at executing well-specified chains of reasoning but weak at the (re)formulation step. Recent empirical work documents three converging failure modes: LLMs anchor strongly on the first formulation presented to them, and existing mitigations such as chain-of-thought or self-reflection fail to remove the effect (Lou and Sun, 2024; Hu et al., 2025); they undersample distinct hypotheses, with temperature-based sampling producing highly redundant rollouts that diversity-promoting interventions noticeably improve (Wang et al., 2025; Anonymous, 2025); and they exhibit a marked choice-supportive bias — inflating confidence in their first answer and sticking with it at rates beyond rational benchmarks even with no new evidence (Kumaran et al., 2026).

Prior LLM scaffolding largely targets the soundness of inference chains rather than the construction of the hypothesis space. Chain-of-thought (Wei et al., 2022), Self-Consistency, Plan-and-

Solve (Wang et al., 2023), SELF-DISCOVER (Zhou et al., 2024), STEP-BACK (Zheng et al., 2024), and ReAct (Yao et al., 2023b) all assume the correct hypothesis space is already understood. Tree- and graph-structured search over reasoning steps (Yao et al., 2023a; Besta et al., 2024; Hao et al., 2023) tests individual hypotheses but is not designed to explore the hypothesis-space itself. Recent hypothesis-generation techniques (Qiu et al., 2024; Wang et al., 2024) are closely related to ours in that they show performance benefits to explicitly generating and testing hypotheses but they do not explicitly output a representation of the space.

We make hypothesis-space construction a first-class operation and cast conceptual ideation as bilevel optimization: a solver maintains a compact parameter space $\Theta$ and a program $g : \Theta \to \Pi$ decoding $\theta \in \Theta$ into a candidate solution $y_\theta = g(\theta)(x)$, with $\dim(\Theta) \ll \dim(\Pi)$ enforcing that $\Theta$ functions as a tractable hypothesis space. The full search is

$$\operatorname*{arg\,min}_{\Theta \subseteq \Theta_0,\ \theta \in \Theta} \ \sigma\big(g(\theta)(x)\big), \tag{1}$$

where $\Theta_0$ is the solver's concept library and $\sigma$ is a (possibly implicit) scoring function. Two problem instances are siblings under a parent class if their known solutions $\theta^*$ agree on at least one dimension. Closely-related problems share most dimensions and are useful because they narrow the search *a priori*. Good ideation, on this view, involves bootstrapping a circumscribed $\Theta$ from prior problems.

From this formalism we derive a four-stage prompt graph, GENERATE-$\Theta$: (1) *generate variables* — propose $\Theta$ as the variables that define this class of problem, independent of the instance; (2) *enumerate values* — discretize the support of each axis; (3) *generate combinations* — enumerate candidate $\theta$'s; (4) a bounded keep-generating loop. To test this approach, we carefully select two decoding-style inference puzzles that require conceptual ideation: cryptic crosswords and few-shot translation with constructed languages.

On our new Minute Cryptic benchmark of cryptic crossword clues, GENERATE-$\Theta$ on a non-thinking Claude Opus 4.6 attains 100% on the easier subset and 97.6% on the harder subset, matching or exceeding the max-thinking variant of the same model (90.7% and 97.6%) and beating SELF-DISCOVER, STEP-BACK, iterated chain-of-thought, and a no-strategy baseline. The same pattern holds on DeepSeek v3.2 (chat vs. reasoner). On a random 50-problem subset of BIG-Bench Rosetta (few-shot translation), GENERATE-$\Theta$ on DeepSeek v4-pro ties the max-thinking baseline at 38.0% — $\sim$20 points above all other baselines.

**Contributions.**

1. An optimization-based formalization of conceptual ideation (Eq. 1) that makes the hypothesis space $\Theta$ a first-class object.
2. GENERATE-$\Theta$, a domain-agnostic, four-stage prompting scaffold derived from the formalism (Section 3.3).
3. Empirical evaluation against four general-purpose prompting baselines and "reasoning"-mode backbones on two ideation-heavy benchmarks (Minute Cryptic and BIG-Bench Rosetta), spanning three LLM families (Claude Opus 4.6, DeepSeek v3.2, DeepSeek v4-pro).
4. Analyses that jointly disentangle ideation *quality* from output *volume* as drivers of strategy success (Sections 4.2.2–4.2.1).

## 2 Related Work

**Structured prompting.** STEP-BACK (Zheng et al., 2024), Plan-and-Solve (Wang et al., 2023), and SELF-DISCOVER (Zhou et al., 2024) target the soundness of inference chains; chain-of-thought (CoT) (Wei et al., 2022) and its tree- and graph-structured descendants (Yao et al., 2023a; Besta et al., 2024; Hao et al., 2023) implement deduction-oriented search algorithms that are complementary to GENERATE-$\Theta$, which can include deduction as an inner loop for hypothesis testing.

**Inference-time scaling and reasoning models.** Recent "thinking" modes (OpenAI, 2024; DeepSeek-AI, 2025) are trained to allocate large internal CoT budgets at inference time, and analyses of test-time compute trade-offs (Snell et al., 2024) show that structured deliberation can rival parameter scaling and be combined with reasoning-specific reinforcement learning to improve output quality (Xu et al., 2025). We show that a deliberately simple external scaffold recovers the non-thinking-to-thinking gap on conceptual-ideation puzzles, raising the question of whether reasoning models have implicitly learned a similar, hypothesis-space construction strategy. Our result suggests the possibility of building stronger reasoning models for novel problem solving, either

by characterizing what productive ideation consists of in algorithmic terms or by using more powerful prompting strategies to bootstrapping better exploratory data during RL training.

**Hypothesis-space search.** Wang et al. (2024) prompt an LLM to propose diverse natural-language hypotheses then test them as programs on ARC; Qiu et al. (2024) iteratively refine inductive hypotheses against examples. Both are closely related to our work, but they generate hypotheses without explicitly outputting the space.

**LLMs on hard puzzle benchmarks.** Puzzles and games have commonly been used to benchmark LLMs. Cryptic crosswords specifically have been studied as a stress-test of LLM linguistic reasoning (Efrat et al., 2021). While solvers have been developed specifically for cryptic crosswords (Andrews and Witteveen, 2025), here we target domain-general strategies, with cryptics as one test case. Other linguistic puzzles have been studied, including remote-associates-style games like New York Times's Connections (Todd et al., 2025; Samdarshi et al., 2024). Video game puzzles have also served as a rich testbed for concept learning (Ahmed et al., 2025; Tsividis et al., 2026). Our evaluation domains here are chosen to de-confound logic (correct application of domain rules to reach valid conclusions) from ideation strategies, and therefore avoid visuospatial and physical reasoning.

**Automated-science / agentic systems.** Lu et al. (2024); Jansen et al. (2026) and related end-to-end systems operate at the research-pipeline level; our work targets the ideation primitive any such system might deploy. Similar to literature reviewed above, the AI Scientist leveraged hypothesis chaining but did not generate the hypothesis space itself.

**Topology of LLM reasoning.** Tan et al. (2025) apply persistent-homology analysis to reasoning traces and find that topological features predict trace quality better than standard graph metrics. We share their motivation — that the *shape* of a search trajectory contains signal — but operationalize it with a domain-specific coding and information theoretic idea-diversity measures.

# 3 Methods

## 3.1 Tasks and Datasets

**Minute Cryptic.** Daily cryptic crossword puzzles from `minutecryptic.com`, scraped from September 2025 through March 2026 (186 clues), split by the puzzle's published *par* rating — the average number of in-site hints a human solver took — into par-1 ($n = 43$), par-2 ($n = 85$), and par-3+ ($n = 58$, not used). Each clue is a self-contained string with a length enumeration; solutions are graded by literal, case-insensitive string match. Example: "Fifty-one trillion litres, to be precise (7)" → `LITERAL`. Cryptic clues require conceptual ideation because the solver cannot simply use logical deduction to infer the wordplay sub-type but rather must search over a large inventory of possibilities (anagrams, hidden-words, containers, etc.).

**BIG-Bench Rosetta.** The first 50 problems of the BIG-Bench Linguistics Puzzles task (Srivastava et al., 2022). Each problem provides six aligned English/conlang pairs and asks for the seventh translation. Example: *given pairs of the form "other: nu loduqu zewa ... / English: birds chase the small person", translate a new conlang utterance into English*. The task requires the solver to induce the relevant linguistic dimensions (word order, case marking, number marking, agreement, etc.).

**Selection criteria.** Both tasks require search over problem formulations before any candidate solution can be evaluated, both admit short verifiable answers, and both allow generation of fresh instances outside training-set memorization.

## 3.2 Models

We evaluate three LLM families. **Claude Opus 4.6** (Anthropic), two modes: *instant* (no extended thinking) and *thinking* (max extended thinking). **DeepSeek v3.2** via the DeepSeek cloud API: *chat* (no reasoning) and *reasoner* (reasoning on). **DeepSeek v4-pro**, two modes (*instant*, *thinking*). DeepSeek deprecated v3.2 API access between our Minute Cryptic and Rosetta experiments, so Rosetta results use v4. We did not run Opus on Rosetta because of API cost constraints.

The per-turn output token budget was 8,192 for all non-thinking DeepSeek runs, the per-model maximum for all reasoning-mode runs, and the provider default for Opus. Temperature was 0.7 throughout. All runs shared the system prompt: "You are a precise and thorough problem-solver who is an excellent scientific thinker."

## 3.3 Prompting Strategies

All strategies are instantiated as directed *prompt graphs* executed by a multi-turn agent. Nodes are prompt templates; edges define transitions; multi-turn loops are bounded by a per-edge `max_repeats` counter (set to 5 for the `keep_generating` edge in

all multi-turn strategies). A shared side-channel subgraph runs after each "solve"-eligible node to extract a one-line answer and check it against ground truth; on success, the run terminates early. This gives every strategy the same answer-extraction machinery and the same early-stop semantics.

**DIRECT** Single turn: present the problem, ask for a solution. No continuation loop.

**ITERATED-COT** One "solve, think step-by-step" turn, then up to $k$ "you haven't found the answer yet, keep generating new ideas" continuations.

**STEP-BACK** Paraphrase the problem into a more generic, higher-level question, answer that first, then solve grounded in those principles (Zheng et al., 2024).

**SELF-DISCOVER** Four stages: select reasoning modules from a fixed list of 39; adapt them to the task; operationalize them into a JSON reasoning plan; execute that plan (Zhou et al., 2024). Keep thinking for up to $k$ additional turns.

**GENERATE-Θ (ours)** Two forced phases before any solving: (i) *generate variables* (exhaustively generate the variables that define this class of problem, independent of the instance); (ii) *enumerate values* (for each variable, exhaustively generate possible values), followed by (iii) *generate combinations* (work through reasonable combinations). Continue (iii) for up to $k$ additional turns.

**GENERATE-Θ-strict-chain (ablation).** Same three phases, with intervening thought tokens suppressed and outputs forced into a strict hypothesis chain.

Strategies were domain-general except for slight modifications. Specifically, for Minute Cryptic we simply noted that the puzzle was a cryptic crossword clue, which was only necessary to disambiguate a small number of examples that could be misclassified without additional context. For Rosetta, we stated that the answer should not be overly complex to discourage pathological thinking modes observed during piloting. In addition, we found that a stricter version of GENERATE-Θ prompting was necessary for DeepSeek to adequately follow the strategy on Rosetta, which was a more complex task. Specifically, we forced hypotheses to be output in a standardized format; without this restriction outputs tended to ignore the initially generated dimensions. The GENERATE-Θ-strict-chain variant was developed to test the role of intervening thoughts on search policies.

### 3.4 Evaluation Protocol

We run evaluation through a standard task harness, scoring with the same checker the agent uses for early stopping. We sample three epochs per problem for all DeepSeek v3.2 runs (Minute Cryptic); single-epoch evaluation for Claude Opus and DeepSeek v4-pro (due to API cost).

To confirm that strategy-level accuracy differences are not artifacts of single-epoch sampling, we conduct a statistical test using the three-epoch runs. A paired bootstrap on the per-problem mean accuracy difference (10,000 samples; GENERATE-Θ minus each non-thinking baseline) shows large, significant gaps over DIRECT, ITERATED-COT, and STEP-BACK on both par-1 and par-2 ($+12$ to $+31$ pp, all $p < .01$; 95% CIs excluding zero), and a statistical tie with SELF-DISCOVER (par-1: $+1.6$ pp, $p = .71$; par-2: $+6.7$ pp, $p = .055$).

### 3.5 Per-Facet Idea Coding

To ask not just whether each strategy solves a clue but how it explored Θ, we code solvers' turn-by-turn outputs into a faceted idea space, taking Minute Cryptic par-1 subset as a case study. For each solver run we extract the complete chain of outputs, and pass it to a coding LLM (claude-opus-4-6, temperature 0.3) which is carefully prompted to decompose the output into *ideas*. Each idea commits to up to four facets of the problem:

**Parse** which substring of the clue is treated as the definition (position $\in$ {start, end, whole, other}).

**Mechanism** ordered list of wordplay types, drawn from a fixed inventory of 13 standard types (anagram, hidden_word, reversal, container, deletion, charade, double_def, homophone, cryptic_def, &lit, acrostic, substitution, letter_selection).

**Execution** step-by-step instantiation referencing indicator and fodder substrings of the clue.

**Output** the candidate answer guess.

A second LLM pass normalizes synonym mechanism names to canonical ones, deduplicates, and runs simple consistency checks. Ideas are organized into a hierarchy: idea $P$ is treated as a parent of idea $C$ when $P$'s non-null facets form a strict subset of $C$'s with matching values, capturing the natural refinement relation under which committing to additional facets specializes a partial idea. Each leaf node is counted as a distinct hypothesis.

**Features.** For each (strategy, clue) we compute per-facet Shannon entropy (`parse_entropy`, `mechanism_entropy`, `execution_entropy`, `output_entropy`), the number of distinct ideas (`n_ideas`), and the mean nearest-neighbor Jaccard distance between idea sets within the strategy (`mean_nn_jaccard`). Since the hypothesis space may be unbounded, we assume a discrete space spanned by the set of ideas generated across all solvers. Thus, a solver that discovered most ideas that other solvers came up with plus new ones would likely receive the highest entropy score.

# 4 Results

## 4.1 Accuracy

Table 1 reports accuracy by strategy, model, and dataset. GENERATE-Θ is the best prompt-only performer in every column, tying the reasoning-mode DIRECT on par-1/Opus, par-2/Opus, and Rosetta/DSv4-pro, and slightly exceeding it on the two DSv3.2 cryptic columns. SELF-DISCOVER is the second-best prompt-only scaffold — consistent with its long seed-module list plus enforced JSON structure (an implicit partial formulation step). STEP-BACK and ITERATED-COT tend to cluster below the top two on both domains, although all strategies besides GENERATE-Θ and the reasoning model performed similarly on Rosetta.

Figure 1 traces cumulative solve rate as a function of conversation turn. In some cases strategies appear to asymptote within the budget but not all, suggesting the ceiling was not always reached.

## 4.2 What explains performance?

A possible account of performance differences is that GENERATE-Θ simply produces more ideas, either because it spends more output tokens generating them, or because more of its ideas land in useful regions of Θ. A second possibility is that GENERATE-Θ pursues a qualitatively different search policy through Θ, diversifying along the dimensions that matter most before committing. But a key confounder is that harder problems require more search, so stronger strategies will on average produce more tokens. We organize the remaining results around three sub-questions that distinguish these accounts.

### 4.2.1 Can output length explain the gap?

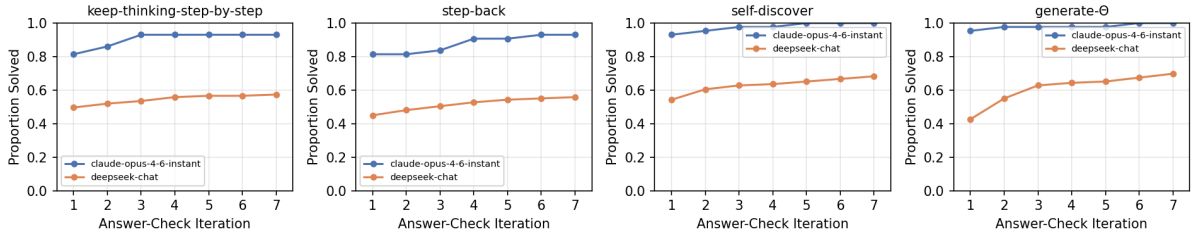
If accuracy were a monotone function of output volume, more output tokens would mean more accuracy, and prompting strategy would matter only insofar as it elicits longer outputs. We present three observations that speak to this question below, which collectively rule out its strongest version.

First, on Rosetta, GENERATE-Θ-strict-chain spends $83{,}440$ mean output tokens per problem and reaches $28.0\%$, while standard GENERATE-Θ spends only $73{,}373$ tokens and reaches $38.0\%$ (Table 1). Suppressing intervening thought tokens spends more and accomplishes less, indicating they may help in efficiently choosing the next hypothesis to test given those already tried. Second, SELF-DISCOVER sometimes uses substantially more output tokens than GENERATE-Θ but yields lower accuracy. On par-1/DSv3.2 SELF-DISCOVER spends $189{,}320$ tokens for $68.2\%$, while GENERATE-Θ spends $125{,}147$ tokens for $69.8\%$. The token cost / accuracy trade is therefore likely strategy-specific, not driven by total output volume alone.

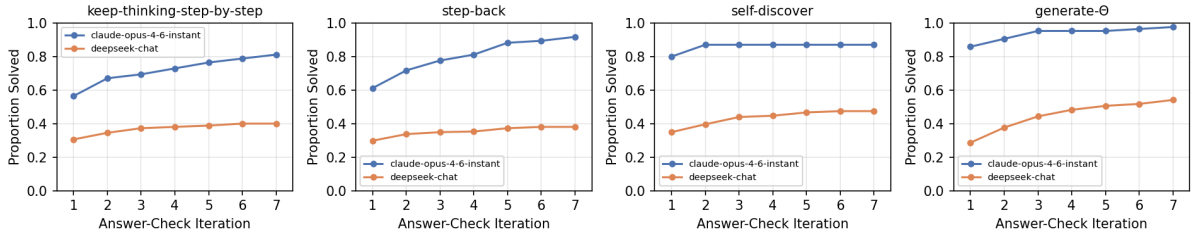
We also plotted tokens versus estimated problem difficulty to compare token-usage behaviors more systematically. Figure 2 plots, per strategy, observed output tokens spent on each problem against an empirical expected token count. For each problem, we estimated the number of tokens required to solve the problem by averaging the token counts across the non-thinking-mode strategies for a fixed LLM model. In general, we can see that weaker strategies under-generate compared to their stronger counterparts (although estimates are less reliable for harder problem, where only one or two models managed to solve them). This is consistent with several possible explanations related to LLM biases. E.g., LLMs may in some sense be progress-aware, reluctant to spend more tokens when little progress is being made. Or, they may fake answers more on challenging problems, with fake answers usually being shorter. Consistent with these, STEP-BACK and ITERATED-COT are more reluctant to spend tokens relative to the expected budget on Rosetta (which is harder) as compared to Minute Cryptic. To interrogate this question further, we reran ITERATED-COT on Rosetta with

Table 1: Accuracy by strategy, model, and dataset. Best prompt-only result in each column in bold; reasoning-mode baselines italicised. n.r. = not run.

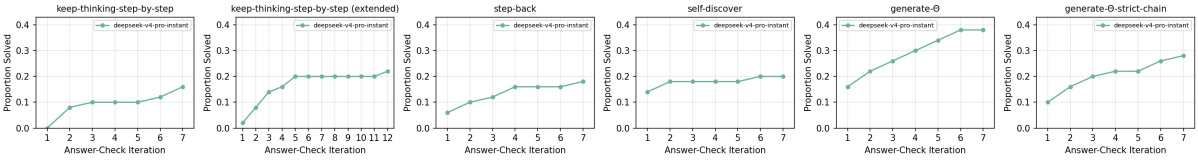
| Strategy | **MC par-1** ($n = 43$) Opus-4.6 | DSv3.2 | **MC par-2** ($n = 85$) Opus-4.6 | DSv3.2 | **Rosetta** ($n = 50$) DSv4-pro |
|---|---|---|---|---|---|
| DIRECT (no-think) | 76.7% | 38.8% | 56.5% | 30.2% | 2.0% |
| DIRECT (*thinking*) | *90.7%* | *68.2%* | *97.6%* | *51.0%* | *38.0%* |
| ITERATED-COT | 93.0% | 57.4% | 81.2% | 40.0% | 16.0% |
| STEP-BACK | 93.0% | 55.8% | 91.8% | 38.0% | 18.0% |
| SELF-DISCOVER | 100% | 68.2% | 87.1% | 47.5% | 20.0% |
| GENERATE-Θ (ours) | **100%** | **69.8%** | **97.6%** | **54.1%** | **38.0%** |
| GENERATE-Θ (strict-chain) | n.r. | n.r. | n.r. | n.r. | 28.0% |




(a) MC par-1




(b) MC par-2




(c) Rosetta

Figure 1: Cumulative solve rate as a function of turn count (excluding DIRECT, which has no continuation loop).

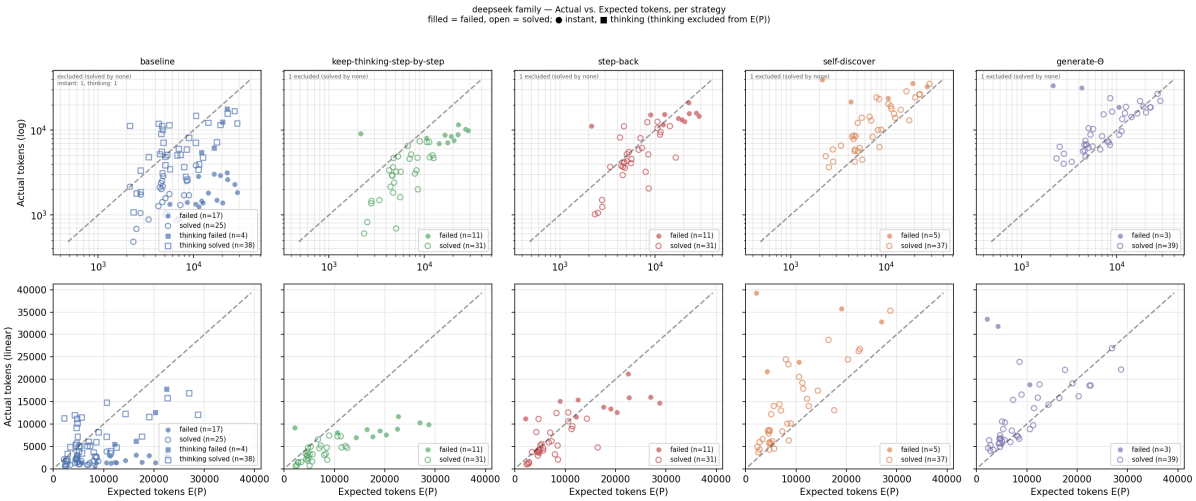
5 additional iterations available to it ("extended"). Improvement was modest: accuracy rose from 16% to 22% (8/50 to 11/50 solves), suggesting that extending the iteration budget on this scaffold has diminishing returns at this difficulty.

Taken together, we cannot rule out that successful strategies find more solutions in part because they disrupt certain LLM biases. E.g., GENERATE-Θ may be successful in part because its Cartesian product instruction effectively creates a long to-do list. Future work should determine the fuller space of strategies that achieve this effect and whether ours is particularly effective among these.

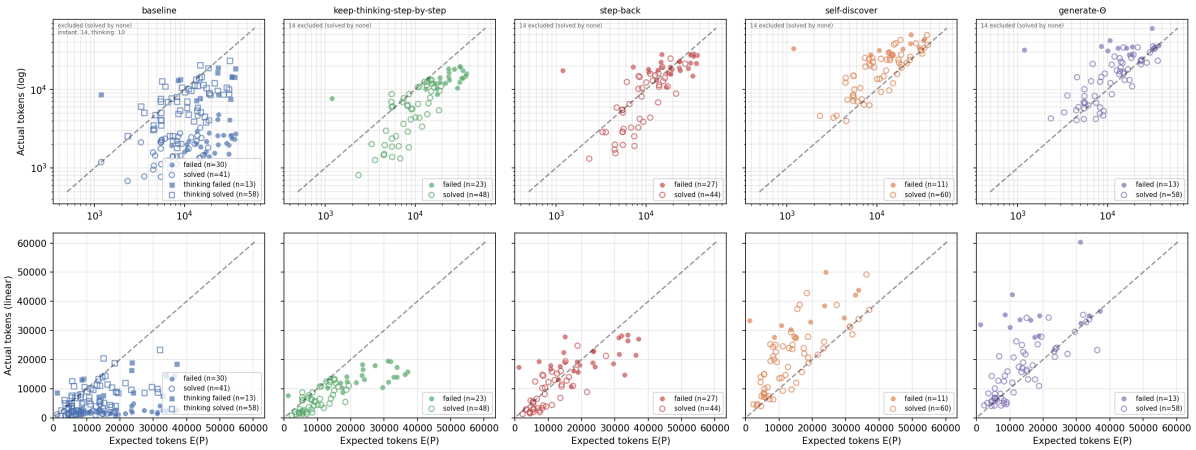
#### 4.2.2 Do strategies differ in how they search?

The per-facet coding analysis lets us look at which ideas each strategy generates, not just how many. Table 2 reports per-facet Shannon entropy, distinct idea counts, and mean nearest-neighbor Jaccard distance for the four non-thinking-mode strategies on MC par-1. (Fig. 7 visualizes the same data.)

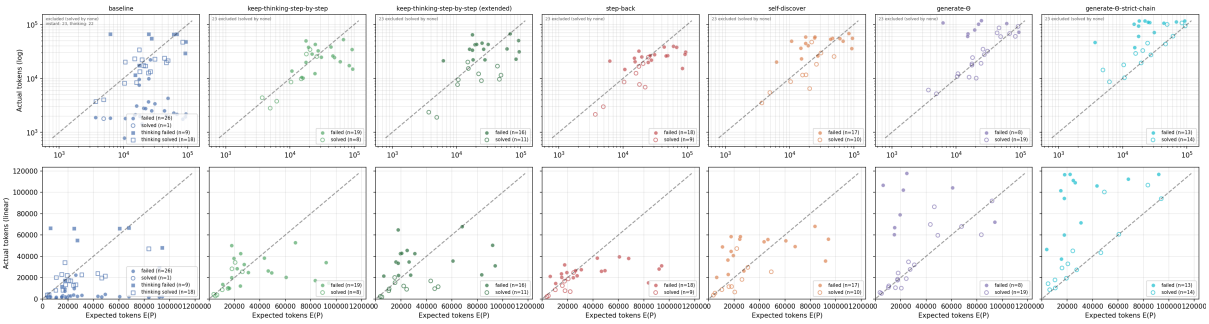
Three qualitatively different ideation profiles emerge. GENERATE-Θ front-loads variability on the prerequisite facets: highest parse-entropy (0.63 vs. $\leq$ 0.40 for every other strategy) and highest mechanism-entropy (1.84 vs. $\leq$ 1.57). It ties for highest execution-entropy despite generating *fewer* ideas overall than SELF-DISCOVER (6.54 vs. 7.40), and its ideas are the most spread out in idea space (nn_jaccard 0.82 vs. 0.72–0.79). SELF-DISCOVER




(a) MC par-1




(b) MC par-2




(c) Rosetta

Figure 2: Observed output tokens vs. expected tokens. Expected tokens $E(P)$ are estimated from instant-variant runs on solved problems of similar predicted difficulty; failed (filled) and solved (open) markers are overlaid. Problems unsolved by any strategy are omitted, since their token cost cannot be estimated.

works backward from candidate outputs: highest n_ideas and highest output-entropy (1.17), but substantially lower parse-entropy — the extra ideas inflate the raw count without adding coverage of the prerequisite facets, consistent with a guess-and-check pattern. STEP-BACK and ITERATED-COT both lean into step-by-step formats, parse the clue once and rarely revisit; they have the lowest parse- and mechanism-entropy, consistent with the anchoring failure mode. The accuracy differences in the coded subset (last row of Table 2) track the prerequisite-facet entropies more closely than they track raw idea count, suggesting qualitative differences in search strategy at least partially determine

Table 2: Per-facet Shannon entropy, idea counts, and intra-strategy nearest-neighbor Jaccard distance on MC par-1 (DeepSeek-v3.2; mean ± SE across clues). Bold = best in row. "Acc" counts the problems in the coded subset solved at least once across three epochs.

| | GENERATE-Θ | SELF-DISCOVER | STEP-BACK | ITERATED-COT |
|---|---|---|---|---|
| n_ideas | 6.54 | **7.40** | 6.28 | 5.58 |
| parse ent. | **0.63** | 0.40 | 0.33 | 0.40 |
| mech. ent. | **1.84** | 1.57 | 1.22 | 1.18 |
| exec. ent. | 2.06 | **2.09** | 1.89 | 1.70 |
| output ent. | 0.63 | **1.17** | 0.96 | 0.98 |
| nn_jaccard | **0.82** | 0.79 | 0.77 | 0.72 |
| Acc | 30/43 | 29/43 | 24/43 | 26/43 |

outcomes. An illustrative case study — the reconstructed idea forests for a single clue across all four strategies — appears in Appendix B (Fig. 6).

## 5 Discussion

**Summary.** A deliberately simple, formulation-first prompting scaffold closes the non-thinking-to-thinking gap on two conceptual-ideation puzzle benchmarks. Analyses suggest the gain is not driven by output volume alone. Per-facet entropy analysis shows GENERATE-Θ explores the prerequisite facets of Θ more thoroughly than the other prompt-only strategies, while SELF-DISCOVER expends its idea budget on output guesses and STEP-BACK/ITERATED-COT anchor on the first parse.

**Backbone capability matters.** GENERATE-Θ works well on strong recent models (Opus 4.6, DeepSeek v3.2 chat/reasoner, DeepSeek v4-pro) but works much less well on weaker LLMs we tested during development (e.g., "flash" variant of DeepSeek v4 and earlier DeepSeek versions). On Rosetta, even DeepSeek v4-pro required the strict-chain wrapper to follow the strategy reliably. This is consistent with the view that disciplined hypothesis-space construction is a capability that has only recently become usable for LLMs; below a threshold, models have too many reasoning and knowledge failure modes to take advantage.

**Reasoning-mode training.** GENERATE-Θ on a non-thinking model matches or exceeds the max-thinking variant of the same model, which suggests that part of what thinking-mode training has produced could be internalized hypothesis-space construction. This invites two follow-ups. First, if true, neural decoding should reveal a sequence of variable proposals before solution attempts. Second, continuing to interrogate what search algorithms work best for conceptual ideation is an obvious collaboration point between AI and cognitive science, and explicitly representing this process provides additional potential impact when tackling non-linguistic problem domains, such as strategizing and puzzle solving. This approach allows for parameters of cognitive behaviors such as information foraging (Fu and Pirolli, 2007) to be modeled and learned in diverse domains.

**Outer-loop revision.** GENERATE-Θ assumes a reasonable Θ can be proposed up-front; it does no meta-control over which facet is most worth varying next, and does not explicitly revise Θ when the initial frame is wrong. The natural next steps include Θ-revision operators motivated by human problem solving (e.g., reasoning-base pruning, abstraction, analogy to known problems) as well as integration with specialized reasoning agents that return structured failure feedback (proof attempts, code execution, calculations) to drive that revision.

**Token efficiency.** A practical limitation is overhead: GENERATE-Θ routinely spends more output tokens than DIRECT or ITERATED-COT. Internalizing the formulation stage — via fine-tuning or RL — is one direction for reducing this overhead while retaining the ideation gains.

## Limitations

We showed relative success for our new prompting strategy in just two puzzle domains. Future work will be needed to confirm generality and disentangle confounders related to LLM biases. Furthermore, we deliberately chose domains where reasoning was not a principle bottleneck. To extend our results to more challenging domains like math and coding, our strategy may need to be combined with specialized reasoning modules and additional prompting strategies for revising the hypothesis space mid-solve. Finally, due to API costs, we were unable to run multiple epochs on every dataset, potentially shifting results.

## References


Zergham Ahmed, Joshua B Tenenbaum, Christopher J Bates, and Samuel J Gershman. 2025. Synthesizing world models for bilevel planning. *arXiv preprint arXiv:2503.20124*.

Martin Andrews and Sam Witteveen. 2025. Generating code to verify cryptic crossword reasoning. In *ICLR 2025 Third Workshop on Deep Learning for Code*.

Anonymous. 2025. Outcome-based exploration for LLM reasoning. *arXiv preprint arXiv:2509.06941*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *AAAI*.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Avia Efrat, Uri Shaham, Dan Kilman, and Omer Levy. 2021. Cryptonite: A cryptic crossword benchmark for extreme ambiguity in language. In *EMNLP*.

Wai-Tat Fu and Peter Pirolli. 2007. Snif-act: a cognitive model of user navigation on the world wide web. *Hum.-Comput. Interact.*, 22(4):355–412.

Dedre Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *EMNLP*.

Douglas R Hofstadter, Melanie Mitchell, and 1 others. 1995. The copycat project: A model of mental fluidity and analogy-making. *Advances in connectionist and neural computation theory*, 2(205-267):2–3.

Douglas R. Hofstadter and Emmanuel Sander. 2013. *Surfaces and Essences: Analogy as the Fuel and Fire of Thinking*. Basic Books, New York.

Yuan Hu and 1 others. 2025. An empirical study of the anchoring effect in LLMs: Existence, mechanism, and potential mitigations. *arXiv preprint arXiv:2505.15392*.

Peter Jansen, Peter Clark, Doug Downey, and Daniel S Weld. 2026. Generating literature-driven scientific theories at scale. *arXiv preprint arXiv:2601.16282*.

Priyanka Kargupta, Shuyue Stella Li, Haocheng Wang, Jinu Lee, Shan Chen, Orevaoghene Ahia, Dean Light, Thomas L Griffiths, Max Kleiman-Weiner, Jiawei Han, and 1 others. 2025. Cognitive foundations for reasoning and their manifestation in llms. *arXiv preprint arXiv:2511.16660*.

Günther Knoblich, Stellan Ohlsson, Hilde Haider, and Detlef Rhenius. 1999. Constraint relaxation and chunk decomposition in insight problem solving. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 25(6):1534–1555.

Dharshan Kumaran, Stephen M. Fleming, Larisa Markeeva, Joe Heyward, Andrea Banino, Mrinal Mathur, Razvan Pascanu, Simon Osindero, Benedetto de Martino, Petar Veličković, and Viorica Pătrăucean. 2026. Competing biases underlie overconfidence and underconfidence in LLMs. *Nature Machine Intelligence*. Preprint: arXiv:2507.03120.

Jiaxu Lou and Yifan Sun. 2024. Anchoring bias in large language models: An experimental study. *arXiv preprint arXiv:2412.06593*.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*.

Melanie Mitchell. 2021. Abstraction and analogy-making in artificial intelligence. *Annals of the New York Academy of Sciences*, 1505(1):79–101.

Nancy J. Nersessian. 2008. *Creating Scientific Concepts*. MIT Press, Cambridge, MA.

Allen Newell and Herbert A. Simon. 1972. *Human Problem Solving*. Prentice-Hall, Englewood Cliffs, NJ.

Stellan Ohlsson. 1992. Information-processing explanations of insight and related phenomena. In M. T. Keane and K. J. Gilhooly, editors, *Advances in the Psychology of Thinking*. Harvester-Wheatsheaf.

OpenAI. 2024. Learning to reason with LLMs. Technical report, OpenAI. https://openai.com/index/learning-to-reason-with-llms/.

Linlu Qiu, Liwei Jiang, Ximing Lu, Melanie Sclar, Valentina Pyatkin, Chandra Bhagavatula, Bailin Wang, Yoon Kim, Yejin Choi, Nouha Dziri, and Xiang Ren. 2024. Phenomenal yet puzzling: Testing inductive reasoning capabilities of language models with hypothesis refinement. In *ICLR*.

Prisha Samdarshi, Mariam Mustafa, Anushka Kulkarni, Raven Rothkopf, Tuhin Chakrabarty, and Smaranda Muresan. 2024. Connecting the dots: Evaluating abstract reasoning capabilities of llms using the new york times connections word game. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21219–21236.

Herbert A Simon. 1973. The structure of ill structured problems. *Artificial intelligence*, 4(3-4):181–201.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Aarohi Srivastava and 1 others. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Xiao Wen Tan, Ngoc Tan, Gillian Lee, and Stanley Kok. 2025. The shape of reasoning: Topological analysis of reasoning traces in large language models. *arXiv preprint arXiv:2510.20665*.

Graham Todd, Tim Merino, Sam Earle, and Julian Togelius. 2025. Benchmarking language models with the new york times connections puzzle. *IEEE Transactions on Games*.

Pedro Tsividis, Joao Loula, Jake Burga, Juan Pablo Rodriguez, Sergio Arnaud, Nate Foss, Andres Campero, Ajay Subramanian, Thomas Pouncy, Samuel J Gershman, and 1 others. 2026. Human-level learning of complex novel tasks as theory-based modelling, exploration and planning. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 384(2320).

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Ruocheng Wang, Eric Zelikman, Gabriel Poesia, Yewen Pu, Nick Haber, and Noah D. Goodman. 2024. Hypothesis search: Inductive reasoning with language models. In *ICLR*.

Tianchun Wang, Zichuan Liu, Yuanzhou Chen, Jonathan Light, Weiyang Liu, Haifeng Chen, Xiang Zhang, and Wei Cheng. 2025. On the effect of sampling diversity in scaling LLM inference. *arXiv preprint arXiv:2502.11027*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Fengli Xu, Qianyue Hao, Chenyang Shao, Zefang Zong, Yu Li, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, and 1 others. 2025. Toward large reasoning models: A survey of reinforced reasoning with large language models. *Patterns*, 6(10).

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations (ICLR)*.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. Take a step back: Evoking reasoning via abstraction in large language models. In *The Twelfth International Conference on Learning Representations (ICLR)*. ArXiv:2310.06117.

Pei Zhou, Jay Pujara, Xiang Ren, Xinyun Chen, Heng-Tze Cheng, Quoc V. Le, Ed H. Chi, Denny Zhou, Swaroop Mishra, and Huaixiu Steven Zheng. 2024. Self-Discover: Large language models self-compose reasoning structures. In *Advances in Neural Information Processing Systems*.

## A Per-difficulty token spend

Figures 3–5 plot total output tokens spent vs. an empirical difficulty rank (mean solve rate across strategies), per strategy and per backbone, for the three datasets. Note that problems where a strategy failed do not have associated datapoints but points on either side of a missing datapoint will still be connected by a line.

## B Idea-forest visualization on a representative clue

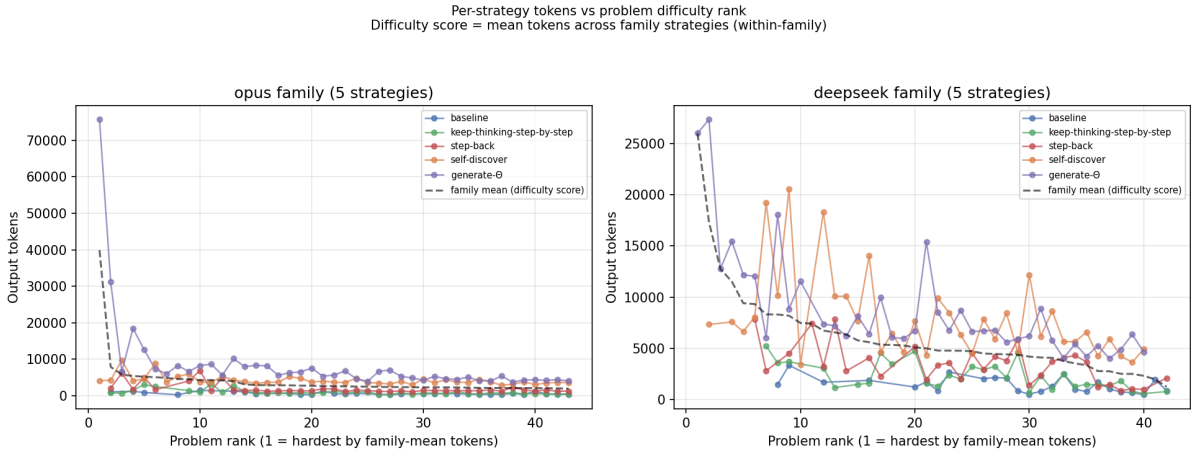



Figure 3: Minute Cryptic par-1. Output tokens vs. difficulty rank.

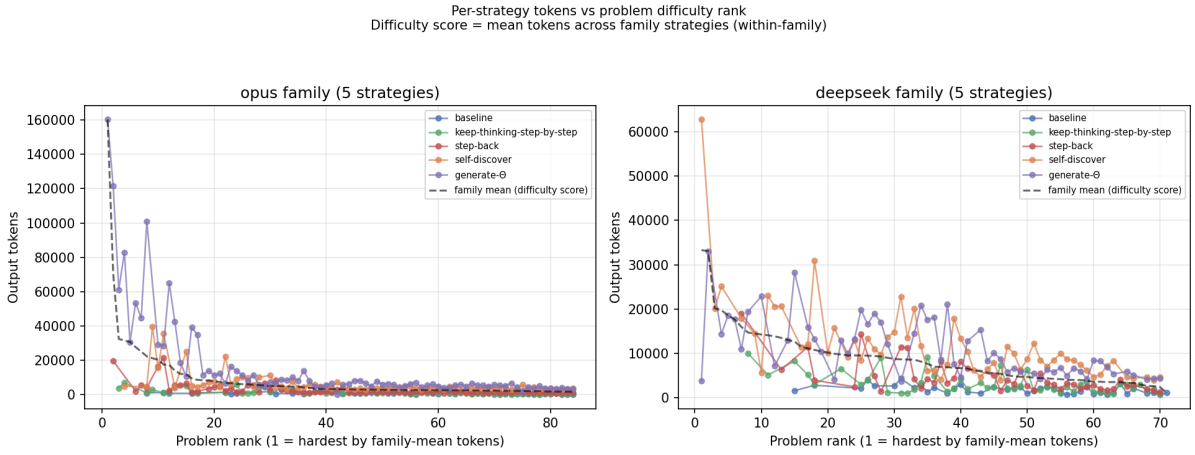



Figure 4: Minute Cryptic par-2. Output tokens vs. difficulty rank.

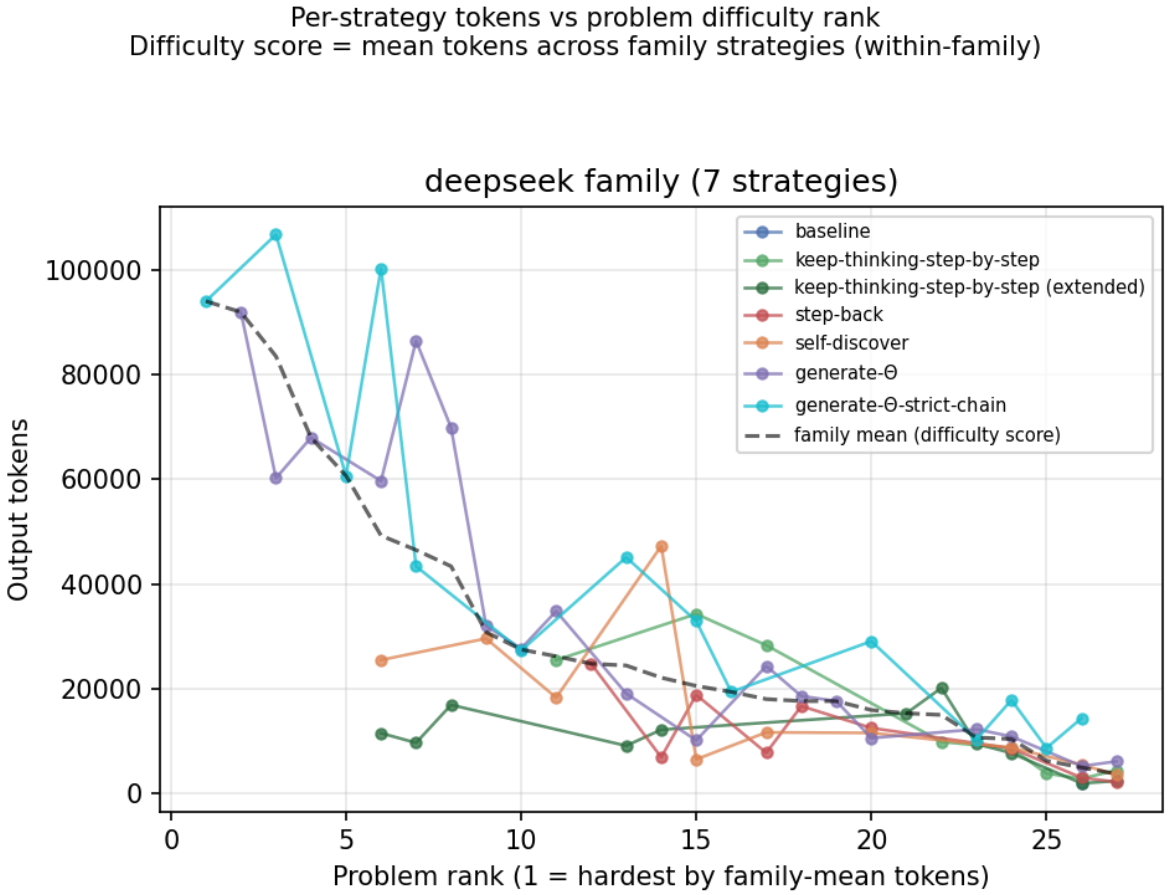



Figure 5: BIG-Bench Rosetta. Output tokens vs. difficulty rank.

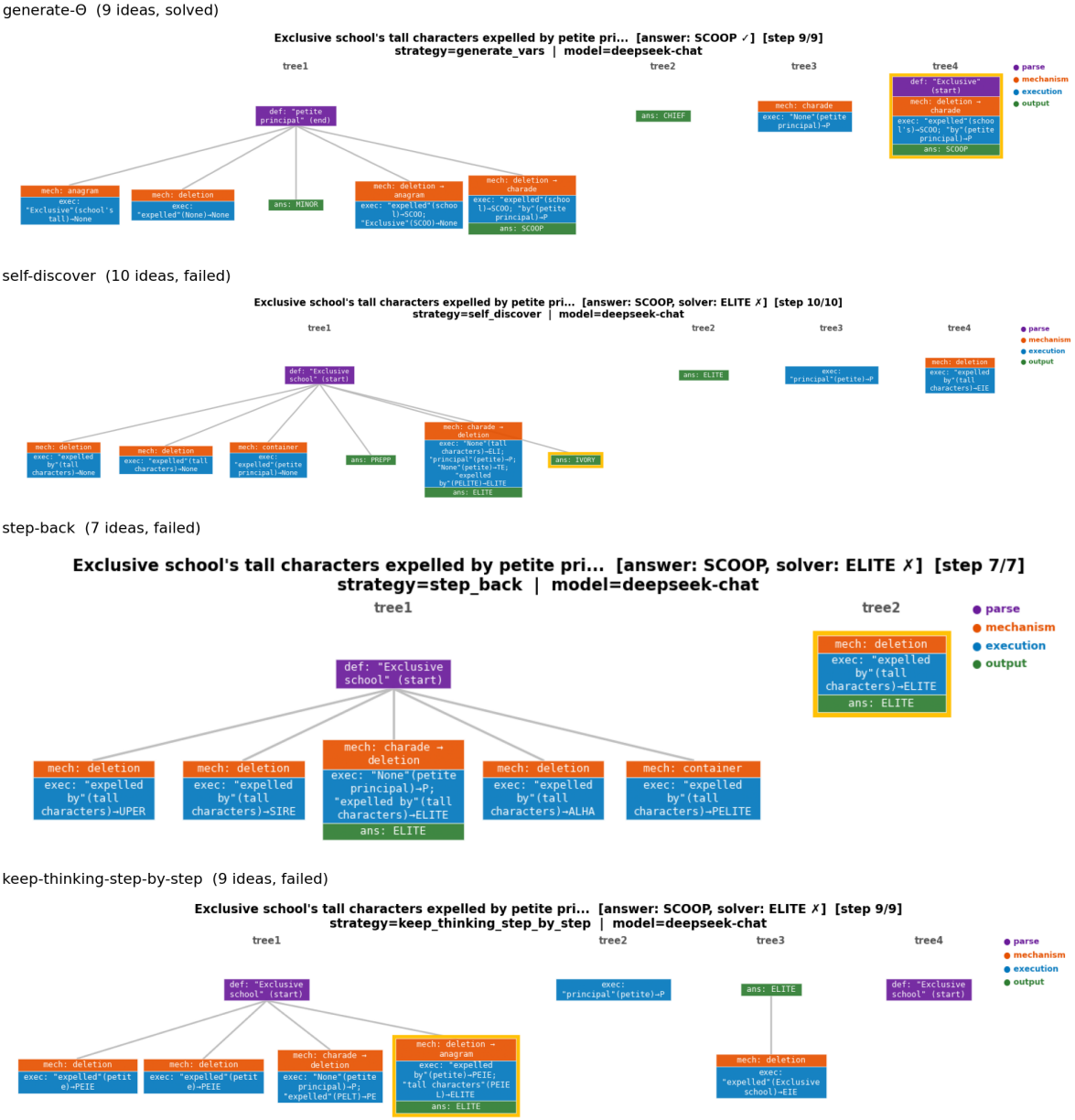



Figure 6: Reconstructed idea forests for all four prompt-only strategies on par-1 problem 16: *"Exclusive school's tall characters expelled by petite principal"* → SCOOP. Wordplay: definition = *Exclusive* (a news scoop); from SCHOOL, remove the typographically "tall characters" (H, L) to leave SCOO, then append *petite principal* (the first letter of *petite*, P) to give SCOOP — mechanism: deletion then charade. Panels (top to bottom): GENERATE-Θ (9 ideas) considers two alternative parses — *petite principal* as definition (suffix) and *Exclusive* as definition (prefix) — and converges on the latter, which matches the gold parse/mechanism/execution in the coding analysis (highlighted gold); SELF-DISCOVER (10 ideas), STEP-BACK (7 ideas), and ITERATED-COT (9 ideas) all anchor on *Exclusive school* as the definition — the salient surface reading — and return ELITE. Node colors encode facet (purple = parse, orange = mechanism, blue = execution, green = output). All four strategies were run on DeepSeek-chat.

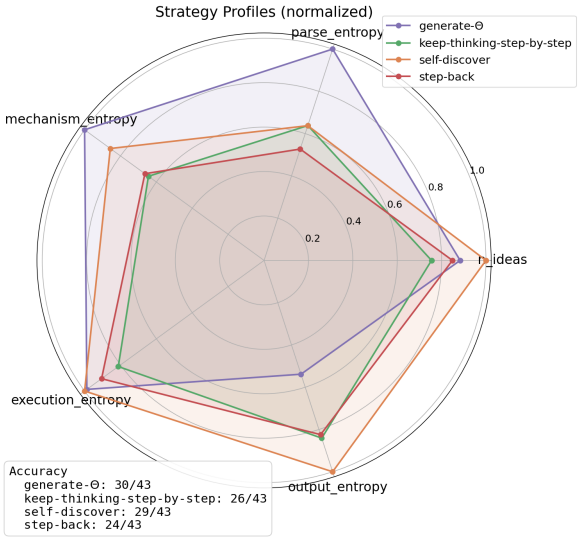



Figure 7: Per-strategy radar profile on MC par-1 (DeepSeek-chat, three-epoch any-correct). Axes are normalised across strategies; the outer ring is the maximum within-row value. GENERATE-Θ dominates on prerequisite facets (parse, mechanism) and on idea spread (nn_jaccard); SELF-DISCOVER dominates on output entropy and total idea count.