

Three Rounds on Emergent Analogy: What the Webb, Hodel-West and Lewis-Mitchell Exchange Settled, How Its Third Round Tested a System While Still Claiming the Model, and Whether 'Counting' Names the Step That Analogy Theory Calls Inference

Pranay Mahendrakar

SONYTECH · ORCID 0009-0003-7224-029X

pranaymahendrakar@sonytech.in · mahendrakarpranay@gmail.com · +91 6361723454

Abstract. In 2023 a large language model was reported to solve text-based analogy problems zero-shot at or above the level of college students. Two critiques followed. They showed that performance on letter-string analogies collapses when the alphabet is permuted or replaced by symbols, while human performance does not. The original authors replied that the failures come from an auxiliary difficulty with counting, and that GPT-4 solves the permuted problems at a human level once it can write and execute code. This paper reads all three rounds in full, including the preprint and published versions of the reply, and sorts their claims into three groups. Settled: the replications agree, and the unaided model fails the counterfactual variants under answer-only prompts. Moved: the evidence. The third round's decisive result is for a model-plus-interpreter system, while its title claim remains about language models. Unsettled: which operations count as auxiliary, and whether the models represent the new alphabet at all. Published checks from both sides, run in different studies with different alphabets, suggest splitting "counting" into two operations: GPT-4 names the one-step successor of a letter in a permuted alphabet almost perfectly, but identifies the interval (of up to two steps) between two given letters about one time in ten. A study of four newer models, however, finds every model worse at naming items two steps away, and its authors conclude that the models do not build representations of novel alphabets on the fly. If the difficulty lies in recognising the relation in the source pair, it lies in what componential theories of analogy call inference; if it lies in representing the order, it lies in encoding, which the same theories also count as part of analogy. On that decomposition, either way, the reply's "auxiliary" label is not yet earned. The paper states a procedure for reading capacity claims from counterfactual tasks, in which the decomposition of the task is fixed before data are seen, and proposes five experiments on existing materials that could decide between the readings. Every number in it is taken, or summed, from the published sources.

Keywords: analogical reasoning; large language models; letter-string analogies; counterfactual tasks; tool use; code execution; auxiliary task demands; human-machine comparison

1. Introduction

The argument over whether large language models reason by analogy became a formal exchange in three rounds, and it is unusually well documented. In round one, Webb, Holyoak and Lu (2023) compared GPT-3 (text-davinci-003) with university students on four kinds of analogy problem: a text-based matrix reasoning task they built for the purpose (Digit Matrices), letter-string analogies in the tradition of Hofstadter and Mitchell's Copycat (Mitchell and Hofstadter, 1990), four-term verbal analogies, and story analogies drawn from Gentner's materials (Gentner, 1983). GPT-3 matched or exceeded the average student on the first three. The authors concluded that "large language models such as GPT-3 have acquired an emergent ability to find zero-shot solutions to a broad range of analogy problems" (Webb et al., 2023).

Round two came from two directions. Hodel and West (2023) rebuilt the letter-string problems on a randomly permuted alphabet with an interval of two letters, and found GPT-3's accuracy on four of six transformation types near zero while 121 undergraduates stayed high.

Lewis and Mitchell (2024a, 2024b) ran a larger and more graded version: alphabets with 2, 5, 10 or 20 letters displaced, alphabets of non-letter symbols, and matching human studies. Averaged over all their alphabets, humans held at about three quarters correct; GPT-3, GPT-3.5 and GPT-4 fell to between 0.35 and 0.49. Lewis and Mitchell (2024b) extended the test to Digit Matrices and story analogies, where they found a drop when the blank in the matrix moves away from the bottom-right corner and an answer-order bias in the stories. Stevenson et al. (2025) found the same pattern with children: 8-year-olds transfer letter-string analogies from the Latin alphabet to the Greek alphabet and to a list of symbols, and four frontier models do not.

Round three is the reply. Webb, Holyoak and Lu (2024; published as Webb et al., 2025) argued that the permuted-alphabet problems require converting each letter into its position in the new order, which requires precise counting, a known weakness of language models. They gave GPT-4 the ability to write and execute code. It wrote code that converted letters to indices, solved the problems at a level statistically indistinguishable from human participants, and explained its answers. The

published version adds two controls that the preprint lacked and, in an argument against contamination, describes the object tested as a system: "the system as a whole was presented only with the counterfactual problems, and the ability to solve them thus cannot be explained by dependence on similarity to the training data" (Webb et al., 2025). The sentence argues that training-data similarity cannot explain the result; it does not say that the capacity belongs to the system rather than to the model.

A tempting summary of the exchange is that it quietly changed the question from whether the model reasons analogically to whether the system does, and that nobody in the exchange conceded this. Reading the full record, that summary does not survive as written, and the way it fails is informative. Both sides noticed that the evidence had shifted. Webb et al. (2025) describe what was tested as "the system as a whole", if for a different purpose, and concede that "the use of code execution to solve these particular problems is not a human-like solution". Lewis and Mitchell (2024b) object that "this notion of counting violates the spirit of the letter-string domain." What neither side does is treat the shift as changing the claim. The reply's title and conclusion still speak of "emergent analogical reasoning in large language models"; the critique's objection is about the domain's intent, not about which object the evidence is evidence for. The shift is visible in both papers and absorbed by neither.

This paper makes three claims, and it is worth stating them early because the rest of the paper is organised around them.

The first is a ledger. Several things were settled by the exchange and are now agreed by the parties' own data. The strongest is that the unaided model, answering without a trace, fails the counterfactual letter-string variants: the reply's own published numbers put GPT-4 at 21.7% on them, against humans who were not significantly different from the code-augmented system (Webb et al., 2025). The reply did not dispute this; it explained it.

The second is about the unit being tested. The exchange compared three different computational objects under one name. Rounds one and two tested a model producing an answer directly. The published reply added a model producing a chain of thought, which helped somewhat, and a model driving an external interpreter, which helped a great deal. These are different objects in a precise sense: theoretical work on chain-of-thought shows that a fixed-depth transformer allowed to write intermediate steps can compute functions that the same transformer answering directly cannot (Li et al., 2024; Feng et al., 2023), and an interpreter adds exact indexing that neither has. A capacity demonstrated by the third object is not thereby demonstrated by the first.

The third, which is the paper's main analytic contribution, is about what "counting" names. The reply's account is that analogy was intact and an auxiliary operation failed. Two published numbers, from the two sides of the exchange, suggest splitting that operation in two. When asked for the successor or predecessor of a given letter in a partially permuted or symbol alphabet, GPT-4 is correct between 97% and 100% of the time (Lewis and Mitchell, 2024b, Table 6). When asked to identify the interval between two given letters in the Hodel-West alphabet, GPT-4 is correct 10.8% of the time (Webb et al., 2025). On these numbers, producing a named one-step relation is intact and recognising which relation holds between two items is not. The split is weaker than it looks. The production checks covered one step only, while the interval probe covered intervals from -2 to 2; and the one study that tested production at two steps found every model it tested worse at two steps than at one, and concluded that the models do not build representations of novel alphabets on the fly (Stevenson et al., 2025). The evidence therefore leaves two readings open: a failure to recognise relations that the model can produce, or a weak representation of the new order that shows up in any operation beyond one step. In componential theories of analogical reasoning, recognising the relation between the first two terms is not an auxiliary step; it is the inference component, one of the operations that make a problem an analogy problem, and encoding the terms is another (Sternberg, 1977). On the first reading, the reply's own evidence locates the failure inside what a cognitive theory of analogy would count as analogy. The code condition did not remove that step. It rerepresented letters as integers, after which recognising the relation became integer subtraction, and the reply defends that rerepresentation as itself human-like (Webb et al., 2025). Which reading is right, and whether that defence holds, are the real open questions, and the exchange never fixed, in advance of the data, a criterion that could decide them.

Two deflations should be answered now. The first is that this is a quarrel about words: whether one calls a model-plus-interpreter "the model" is a convention. It is a convention with consequences. The claim at stake in round one was that a system trained only on next-token prediction acquires an ability that cognitive science treats as central to fluid intelligence (Webb et al., 2023). An interpreter was not trained on anything; it executes exact operations the language model cannot perform reliably on its own. If the interpreter supplies the step on which the counterfactual variants turn, the evidence bears on the language model's ability to plan and write code, which is interesting but is not the claim. The second deflation is that current reasoning models solve these problems anyway, so the exchange is history. The best 2026 data point located, from a human-AI teaming study, has frontier chat models correct on 84.6% of letter-string advice turns with permuted and symbol

alphabets (Welsch et al., 2026). That is progress, but it was obtained with a reasoning setting, and so it is evidence about the second object in the list above (or the third, if tools were available; Section 15), not the first. The question the exchange left open is not answered by it; it is made harder to ask.

The paper proceeds as follows. Section 2 fixes terms and scope. Sections 3 to 5 read the three rounds in order, with the numbers each reported. Section 6 states what was settled. Section 7 develops the three-unit account and shows that the critics' own later work also evaluated the unit the reply used. Section 8 develops the production/recognition split, its limits, and what it implies for the auxiliary-demand argument. Section 9 notes that the extended critique, published after the preprint reply, found failures in two further domains that the counting account does not reach. Section 10 consolidates the numbers in one table. Section 11 states the decision structure as an algorithm. Section 12 proposes tests. Sections 13 to 15 give the strongest case against this paper, what is not known, and its limitations.

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

The exchange. "Round one" is Webb et al. (2023). "Round two" is Hodel and West (2023) together with Lewis and Mitchell (2024a) and its extended version (2024b); Stevenson et al. (2025) is independent work from overlapping authors that bears on the same question, and is treated alongside round two. "Round three" is the reply, which exists in two versions that differ in content: the April 2024 arXiv version (Webb et al., 2024) and the May 2025 PNAS Nexus version (Webb et al., 2025). The chronology matters. Lewis and Mitchell (2024b) was posted in November 2024, seven months after the preprint reply; it comments on that reply and adds the Digit Matrix and story-analogy studies. The published reply was received by its journal in December 2024 and cites only the conference version of the critique (2024a), which covered letter strings alone. Strictly, then, the exchange has a fourth move; its new results are discussed with round two, and its comments on the reply with round three, but they were not before the reply's authors when the preprint was written. Where the two versions of the reply differ, this paper says which one it is citing.

Three units. The paper distinguishes three computational objects that the exchange tested under the name "the model". Unit A is a language model that emits its answer directly, without a written trace. The short completion prompts of rounds one and two elicit this unit, as does the no-code condition of the reply, whose prompt instructed the model to "only provide the answer" (Webb et al., 2024). Unit B is the same model allowed to write a trace before answering, either by a zero-shot chain-of-thought instruction (Kojima et al., 2022; Wei et al., 2022b) or because it is a reasoning

model trained to do so. Unit C is a model that can call an external executor, here a Python interpreter, whose results return into its context (Schick et al., 2023; Gao et al., 2023; Gou et al., 2024). These are not rhetorical categories. Unit B differs from Unit A in computational power, not only in prompt (Nye et al., 2021; Li et al., 2024; Feng et al., 2023), and Unit C adds exact operations that neither performs reliably.

Emergent. The word does two jobs in this literature. For Webb et al. (2023) an emergent ability is one acquired without direct training on the task. For Wei et al. (2022a) it is one absent in smaller models and present in larger ones, and Schaeffer et al. (2023) argue that the apparent sharpness of such transitions can be an artefact of the metric. The exchange is about the first sense. Scale enters only through round one's remark that GPT-4 did better than GPT-3 in preliminary tests (Webb et al., 2023); no round tested a scale series, and this paper says nothing about the second sense.

Auxiliary demand. Following Hu and Frank (2024), an auxiliary task demand is a challenge introduced by the evaluation that is not part of the capacity being measured, such as producing an answer in free text rather than choosing among options. Firestone (2020) makes the same distinction in terms of competence and performance, and argues that fair human-machine comparisons must equate such constraints. The reply's central move is to classify counting as an auxiliary demand. Section 8 examines whether that classification can be made without first saying what analogy consists of.

What this paper is not. It is not a survey of analogical reasoning in language models; it cites that literature where it bears on the three rounds and nowhere else. It is not about "analogical prompting", in which a model is asked to recall related problems before solving a new one (Yasunaga et al., 2024); that uses the word analogy for a prompting method. It proposes no system and reports no experiment.

What the paper takes as given. Letter-string problems have no uniquely correct answer, and every study in the exchange scores against the answer its authors judged most abstract (Lewis and Mitchell, 2024b, following Webb et al., 2023). The paper accepts that convention, as the parties do. Where error analyses count "alternative rule" answers separately, that is reported.

3. Round one: the claim and what it rested on

Webb et al. (2023) evaluated GPT-3 zero-shot on four domains and compared it with UCLA undergraduates. On Digit Matrices, a text rendering of the rule structure of Raven's Standard Progressive Matrices, GPT-3 surpassed the average participant when generating answers (odds ratio 1.88, $p = 0.005$, $N = 40$ participants) and when choosing among options (odds ratio 6.27). Its pattern of difficulty across 32 problem subtypes correlated with the

human pattern ($r = 0.39$), and it reproduced signatures of human performance: rules defined by progression were harder than rules defined by features, and difficulty rose with the number of unique rules. The authors argued that because they built Digit Matrices for the study, "we can be sure GPT-3 had never been exposed to problems of this type" (Webb et al., 2023).

On letter strings, GPT-3 outperformed the average of 57 participants (odds ratio 1.76), an effect "driven primarily by stronger performance on zero-generalization problems", in which source and target differ only in the letters used. Both GPT-3 and humans got worse as the number of generalizations between source and target rose, and their error patterns across problem subtypes correlated ($r = 0.7$). The paper also reported that GPT-3 could generalize from letters to real-world concepts in problems such as "a b c $\rightarrow$ a b d, cold cool warm $\rightarrow$?". It noted that performance "was somewhat sensitive to the way in which problems were formatted" (Webb et al., 2023).

On verbal analogies GPT-3 matched or exceeded human participants on several datasets, and on story analogies it was clearly worse than students (odds ratio 0.37), picking the analogous story above chance in near but not far analogies. A preliminary test of GPT-4 did better on the stories. The paper closed with a qualitative test on Gick and Holyoak's radiation problem (Gick and Holyoak, 1980), which GPT-3 solved when prompted to consider an analogous story, and on a physical problem it could not solve. For that last failure the authors wrote that GPT-3's "difficulty with this problem likely stems from its lack of physical reasoning skills, rather than being due to a difficulty with analogical mapping per se" (Webb et al., 2023).

That sentence matters for the later rounds. The move that the reply would make with counting, attributing a failure to an auxiliary skill rather than to analogy, was already part of round one's method. The authors made it explicitly as a matter of principle: cognition consists of "interacting, but dissociable, processes", and it is "important to disambiguate domain-specific failures in processes such as physical reasoning" from core competence (Webb et al., 2024). The principle is sound in cognitive science, where dissociations are established by showing that the auxiliary process can fail while the core process succeeds on independent tasks. Whether the exchange ever supplied that independent evidence for counting is the subject of Section 8.

The claim of round one was also explicitly about the model. The discussion section asks "how GPT-3 achieves the analogical capacity", proposes that self-attention's pairwise similarity computation and key-value indirection may be relevant, and concludes that the approach "may be sufficient to approximate human-like reasoning abilities, albeit through a radically different route" (Webb et al., 2023). The object of the claim is a

network answering prompts. That is the object round two tested.

4. Round two: two critiques and their instruments

4.1 Hodel and West: a permuted alphabet and a wider interval

Hodel and West (2023) made two arguments and ran one experiment. The first argument was about contamination: letter-string problems are discussed on the web, and the original authors had acknowledged in peer review that GPT-3 might have been trained on some. Without ruling out memorization, the response argued, one cannot claim zero-shot reasoning. The second argument was that tests designed for humans need not measure the same capacity in a model, and that human comparison therefore cannot by itself support or refute a zero-shot claim.

The experiment built variants of the letter-string problems with a synthetic alphabet, a random permutation of the 26 letters stated in the prompt, and with an interval of two letters instead of one. GPT-3 (text-davinci-003, temperature 0, 50 instances per type) was compared with 121 University of Washington undergraduates. On the variants with both modifications, GPT-3's generative accuracy was "close to zero (< 0.1)" for extend-sequence, successor, predecessor and fix-alphabetic-sequence problems; only the remove-redundant-letter and sort types, which do not require generating a new letter, held up. Human accuracy on the modified problems stayed close to their accuracy on the originals. Comprehension checks showed that GPT-3 could process the synthetic alphabet, with accuracy at least 30% of the original level on marginally modified tasks for all but one condition (Hodel and West, 2023). The response concluded that the originals' success most likely resided in the training data.

4.2 Lewis and Mitchell: graded counterfactuals, a comprehension check, and two more domains

Lewis and Mitchell (2024a) designed a more graded manipulation. Their alphabets displaced 2, 5, 10 or 20 letters from their usual positions, and two further alphabets used non-letter symbols. They collected data from 136 Prolific participants and tested GPT-3, GPT-3.5 and GPT-4 at temperature 0 with the prompt that had given the models their best performance. Averaged over all alphabets (the permuted and symbol alphabets together with the unpermuted one, which was included in the design) and all problem types, humans scored 0.753, GPT-3 0.488, GPT-3.5 0.350 and GPT-4 0.452 (1,876 human samples and 6,840 per model; Lewis and Mitchell, 2024a, Table 2).

The design has a feature that later turns out to be important. Following Wu et al. (2024), who introduced counterfactual task variants as a way to separate

transferable skill from task-specific procedures, Lewis and Mitchell ran a counterfactual comprehension check: given the permuted alphabet and a letter, name its successor, or its predecessor, one place away. GPT-4 scored 1.0 on successor in every alphabet type, including letters that had been displaced and symbol alphabets, and 1.0, 0.97 and 1.0 on predecessor (Lewis and Mitchell, 2024b, Table 6). The extended paper states the implication plainly: humans held up and the models fell "in spite of the fact that GPT models seemed to comprehend basic relationships in the variant alphabets, as shown by our counterfactual comprehension checks" (Lewis and Mitchell, 2024b).

The extended paper also tested harder letter-string problems. With one, two or three generalizations between source and target, humans fell to 0.358, 0.317 and 0.260, and GPT-4 to 0.271, 0.219 and 0.195 (Lewis and Mitchell, 2024b, Table 1). The human-model gap is widest on the simplest problems and narrows as both struggle, which the authors note.

Two further domains were added. On Digit Matrices, moving the blank from the bottom-right cell to a random position left human accuracy unchanged (0.715 original, 0.771 alternate blank) but dropped GPT-3.5 from 0.813 to 0.480 and GPT-4 from 0.810 to 0.477. Replacing digits by symbols changed nobody's performance much (Lewis and Mitchell, 2024b, Table 3). A comprehension check confirmed that GPT-4 could identify the position of the blank in 9 of 9 cases. On story analogies, the original data showed GPT-4 correct on 16 of 18 problems when the analogous story was listed first and 11 of 18 when it was listed second; the replication found 18 of 18 and 13 of 18, while humans were at 0.78 regardless of order (Lewis and Mitchell, 2024b, Table 4). Paraphrasing the correct story reduced both GPT-4 (0.86 to 0.72) and humans (0.78 to 0.70), with too few items to test the difference (Lewis and Mitchell, 2024b, Table 5).

An error analysis of the simple letter-string problems found that 38.6% of human errors applied a valid alternative rule, against 22.0% of GPT-4's, and that a larger share of GPT-4's errors than of humans' fell in the "incorrect rule" category (32.7% against 14.7%). The two groups' errors in that category also differed in kind: human errors there were mostly a rule applied in the wrong situation, while the GPT errors included adding two letters instead of one, reversing or shifting the target, using the unpermuted alphabet instead of the one given, and, for GPT-4, responses that were too long (Lewis and Mitchell, 2024b, Table 8 and Appendix A.5).

The critics scoped their claim carefully. Their purpose was "to evaluate the robustness of published claims of zero-shot analogical reasoning with simple prompts, so we did not test other prompting formats or models specifically trained for chain-of-thought reasoning." They added: "we agree with WHL's assessment that using

simple zero-shot prompts is the most appropriate way to evaluate LLMs' abilities for analogical reasoning" (Lewis and Mitchell, 2024b). At the end of round two, both sides agreed on the unit of test: a model answering a simple prompt directly.

4.3 Children, and a second comprehension check

Stevenson et al. (2025) compared 42 children aged 7 to 9, 62 adults, and four models (Claude-3.5, Gemma-2 27B, GPT-4o and Llama-3.1 405B) on letter-string analogies in the Latin alphabet, the Greek alphabet and a list of symbols. Children and adults performed similarly across the three; every model degraded significantly from Latin to Greek and again from Greek to symbols. The study includes its own comprehension check, a next-previous letter task in which the model is given the ordered list and asked which item is one or two places before or after a given item. The results of that check are mixed, and they matter for Section 8. Claude-3.5 "performed well and similarly in all three domains, which is in contrast to its letter-string analogy performance that degrades from baseline to near to far domains" (Stevenson et al., 2025). GPT-4o performed well on the check in the Latin and Greek alphabets while its analogy performance fell from Latin to Greek. Llama-3.1 405B did well on both tasks in the Latin and Greek alphabets and poorly on both with the symbol list, and Gemma-2 27B was uneven. Every model did worse when asked for the item two places before or after than for the adjacent one, and the authors note that this could explain the models' trouble with their second-successor analogy item (Stevenson et al., 2025). Their overall reading of the check is not that the alphabet was grasped: they write that the models "appear unable to create on-the-fly representations of novel alphabets", in the next-previous task as well as in the analogies, and they conclude that the problem "seems to lie in that the conceptual abstraction of what constitutes an alphabet ... does not flexibly map to less familiar domains" (Stevenson et al., 2025).

An earlier study by overlapping authors found that models solving Dutch verbal analogies at levels ranging from about that of 7-year-olds (the weakest model) to slightly above that of 11-year-olds (the strongest) each lost one to two years of equivalent performance once associative processes were controlled (Stevenson et al., 2023), and a study of child-friendly visual analogy items found both children and adults outperforming most models, with models and young children sharing a "fallback" strategy of copying part of the problem (Opielka et al., 2024). These are different tasks, but they share round two's method: hold the relational structure fixed, change the surface, and watch whether performance travels.

5. Round three: the reply, in two versions

5.1 The preprint

The April 2024 reply opens by rejecting the contamination argument for the original materials: Digit Matrices were novel, and some letter-string variants, such as generalizing from letters to real-world concepts, are not plausibly on the web (Webb et al., 2024). Its main move concerns the permuted-alphabet problems. These, it argues, "require that letters be converted into the corresponding indices in the permuted alphabet, a process that depends on the ability to precisely count the items in a list", and language models are known to count poorly (Webb et al., 2024).

To test this the authors evaluated "a more recently released variant of GPT-4, augmented with the capacity to write and execute code" on the Hodel-West problems. With code, GPT-4 performed "on par with human participants" (logistic regression, $p = 0.496$); without it, significantly worse. The model used code "to convert letters into their corresponding indices, using code that GPT-4 generated on its own", without instruction to do so. Correct answers were accompanied by accurate explanations, and 46% (23 of 50) of the code model's errors used a valid alternative rule, close to the 39% reported for humans by Lewis and Mitchell (Webb et al., 2024).

The methods section of the preprint shows that the two GPT-4 conditions differed in more than the interpreter. The no-code condition used the Chat Completions API at temperature 0 on 300 problems per interval size with a prompt that instructed "Please only provide the answer. Do not provide any additional explanation." The code condition used the Assistants API at the default temperature of 1, on 60 problems per interval size, with a prompt containing no such instruction; the model "typically responded with an extended 'chain-of-thought' response"; refusals were regenerated until an answer was given; and responses were scored by hand (Webb et al., 2024). The contrast between the conditions therefore bundled the interpreter with permission to write a trace, a different sampling temperature, a different interface, regeneration of refusals and manual scoring. In the terms of Section 2, the preprint compared Unit A with Unit C and attributed the difference to the interpreter.

The preprint's supplementary tables also show where code did not help. Summing the error counts in its Tables 1 and 2 (a computation by this paper's author, not a figure the reply reports), the code model solved 40 of 60 problems at interval one and 30 of 60 at interval two. It gave the intended answer on none of the ten fix-alphabetic-sequence problems at either interval and on 3 of 10 sort problems at each, although the same tables classify many of those errors as valid alternative rules: 6 of the 10 fix errors at interval one, all 10 at interval two, and 3 of the 7 sort errors at each interval. On extend-sequence at interval two it solved 1 of 10; the reply

explains that on those problems GPT-4 did not invoke code to measure the interval in the source and "mistakenly identified an interval size of 1" (Webb et al., 2024). That explanation is itself evidence for the argument of Section 8: the failure occurred at the step of recognising the relation in the source pair.

The preprint also addressed contamination by the Hodel-West problem set, which had been public before the code model's training cutoff. The code model did as well on a new synthetic alphabet, and an older GPT-4 engine with a cutoff before the problem set was published did as poorly without code as the newer one. "Thus, performance was driven almost entirely by the presence or absence of code execution" (Webb et al., 2024).

5.2 The published version

The PNAS Nexus version (Webb et al., 2025) keeps the argument and changes the evidence in three ways that bear directly on the confounds above.

First, it adds a zero-shot chain-of-thought condition. Chain-of-thought "did improve performance to some extent (GPT-4 vs. GPT-4 + chain-of-thought: $P = 0.004$)", but remained worse than humans ($P = 3 \times 10^{-8}$) and worse than the code condition ($P = 4.4 \times 10^{-5}$). Second, it adds a control in which the code-execution model was instructed not to invoke its interpreter: accuracy was 22.5%, against 21.7% for GPT-4 alone ($P = 0.73$). The authors conclude that "improved performance was causally related to the use of code execution, and not to any auxiliary prompts that may be employed in the code execution model." Third, it adds a direct probe: GPT-4 "could not reliably identify the interval between two letters in our synthetic alphabet (overall accuracy of 10.8% for identification of intervals between -2 and 2)" (Webb et al., 2025). Humans and the code model were now at $P = 0.92$, and 38% of the code model's errors used a valid alternative rule, against 39% for humans.

These additions close most of the confound in the preprint. The code-disabled control shows that the Assistants setup without its interpreter is no better than plain GPT-4, and the chain-of-thought condition shows that a trace alone recovers only part of the gap. The published evidence therefore supports the conclusion that, among Units A, B and C, Unit C is the one that reaches human level on these problems, and that the interpreter is what distinguishes it. This paper accepts that conclusion. What it does not accept is the step from there to a claim about Unit A.

The published version also adds two paragraphs of argument. One anticipates the objection that converting letters to indices turns the counterfactual problem into a familiar numerical one, and replies that "human reasoners also make use of rerepresentation to solve insight problems", citing the Gestalt tradition, and that mapping an unfamiliar target onto a familiar source is

"the basic role of analogy in problem solving". "This aspect of the code-augmented model's approach is thus arguably human-like." The other narrows the claim: the findings "pertain specifically to the domain of analogical reasoning", and "should not be interpreted as suggesting that code execution is generally required for language models to solve counterfactual analogy problems" (Webb et al., 2025). The abstract, however, reports that "language models are also capable of generalizing to these new counterfactual task variants when they are augmented with the ability to write and execute code", and concludes that the results "further corroborate the emergence of a capacity for analogical reasoning in LLMs."

5.3 The critics' rejoinder

Lewis and Mitchell (2024b) responded to the preprint in the related-work section of their extended paper. They disagreed "that the letter-string problems with permuted alphabets ... 'require that letters be converted into the corresponding indices.' One doesn't have to compute that, say, b is the 6th letter and p is the 21st letter to solve the problem given above. Rather, one just needs to understand general abstract concepts such as successorship and predecessorship, and what these mean in the context of the given 'fictional' alphabet." They quoted the design principle of the letter-string domain, that problems "should not depend on arithmetical facts about letters", and observed that code generation succeeded on problem types where familiar numerical patterns and counting helped but "was less successful on 'Fix alphabetic sequence' and 'Sort' problems", and that the reply did not test code on symbol alphabets (Lewis and Mitchell, 2024b).

The rejoinder objects to the necessity of counting and to its fit with the domain. It does not say that the code condition tests a different object from the one round one made claims about. Neither does the reply, whose phrase "the system as a whole" is offered against the contamination worry rather than as a statement of which object has the capacity. That is the sense in which the shift is visible and unabsorbed.

6. What the exchange settled

A long exchange tempts a reader to conclude that nothing was resolved. That would be wrong here. Four points are now supported by data from both sides, and none is disputed in any of the papers read. A fifth, which summaries of the exchange often include, is narrower than it appears and is stated separately below.

The replications agree. Lewis and Mitchell (2024b) reran GPT-3 on the original letter-string problems and found results "similar to those of WHL"; their human participants scored somewhat higher than the original UCLA sample but followed the same pattern. On Digit Matrices their GPT results "very closely match those of WHL". Hodel and West's undergraduates also slightly

outperformed the original sample on the original problems (Hodel and West, 2023). The original numbers were not artefacts of one lab's setup.

Unit A fails the counterfactual letter strings. Every study that tested a model answering directly on a permuted or symbol alphabet found a large drop that humans did not show: Hodel and West (2023) for GPT-3, Lewis and Mitchell (2024a, 2024b) for GPT-3, GPT-3.5 and GPT-4, Stevenson et al. (2025) for four 2024 models. The reply's own published data agree: 21.7% for GPT-4 alone (Webb et al., 2025). The reply offers an explanation of this failure, not a denial of it.

Contamination by the critics' own problem set does not explain the reply's result. The code model did as well on a fresh alphabet, and a pre-cutoff engine did as poorly without code (Webb et al., 2024). This closes one contamination worry for round three. It does not address the round-one worry about the original materials, which remains where the parties left it: Digit Matrices were novel, some letter-string problems were web-available, and the counterfactual variants were designed to be neither. Why such questions are hard to settle without access to training data is set out by Sainz et al. (2023) and by Golchin and Surdeanu (2024).

Unit C is not distinguishable from human participants on the Hodel-West problems, and the interpreter is the cause. After the published controls, this is as well supported as anything in the exchange (Webb et al., 2025). The critics did not dispute the measurement; they disputed its interpretation (Lewis and Mitchell, 2024b).

The fifth point is the one that needs care. **GPT-4 can produce one-step neighbours in the new alphabet; whether models represent the new order is disputed.** Lewis and Mitchell's comprehension checks show GPT-4 naming the successor or predecessor of a given item in partially permuted and symbol alphabets with accuracy between 0.97 and 1.0 (Lewis and Mitchell, 2024b, Table 6), and they read this as the models generally grasping what successor and predecessor mean in each fictional alphabet. Stevenson et al. (2025), testing four newer models at one and two steps, found every model worse at two steps, one model failing its check on the symbol list much as it failed the analogies, and a clean pattern (good check, degraded analogies) only for Claude-3.5 across all three alphabets and for GPT-4o across two; they conclude that the models do not form representations of novel alphabets on the fly. What is settled is therefore narrow: for GPT-4, at one step, the permuted order was taken in well enough to name neighbours. Whether it was represented well enough to support anything more is not settled, and Section 8 depends on that difference.

One further matter is open by default rather than by disagreement. The extended critique (Lewis and Mitchell, 2024b) found failures in two further domains:

Digit Matrices with the blank moved, and an answer-order bias in story analogies. Those results appeared seven months after the preprint reply and shortly before the published reply was submitted, which cites only the conference version of the critique; no later paper located for this study responds to them. Section 9 asks whether the counting account could reach them.

7. What moved: the unit being tested

7.1 *The claim sentence in each round*

It helps to line up the sentence in which each round states what it has shown. Round one: "large language models such as GPT-3 have acquired an emergent ability to find zero-shot solutions to a broad range of analogy problems" (Webb et al., 2023). Round two: "despite previously reported successes of LLMs on zero-shot analogical reasoning, these models often lack the robustness of zero-shot human analogy-making" (Lewis and Mitchell, 2024b). Round three, preprint: "the inability of language models to solve these particular reasoning problems stems from a specific difficulty with counting, rather than a general inability to solve analogy problems" (Webb et al., 2024). Round three, published: "language models are also capable of generalizing to these new counterfactual task variants when they are augmented with the ability to write and execute code" (Webb et al., 2025).

The published sentence is accurate about what was tested. The conditional clause names Unit C. The next sentence of the abstract drops the condition: "the emergence of a capacity for analogical reasoning in LLMs". The argument that licenses dropping it is the auxiliary-demand argument: the interpreter supplied counting, counting is not analogy, so the analogy was the model's all along. If that argument holds, the unit question is harmless. Section 8 argues it does not hold as stated. This section argues something prior: even if it held, the three units are different enough that evidence about one does not transfer to another without an explicit bridging argument, and the exchange supplied none.

7.2 *Why the units differ in kind*

The difference between Unit A and Unit B is not stylistic. Nye et al. (2021) showed that models which fail multi-step computations answered directly can perform them when trained to write intermediate steps to a scratchpad. Theoretical work makes the difference precise: a constant-depth transformer answering directly is limited to a small circuit class, while the same transformer allowed to write T steps of chain-of-thought can compute any function computable by boolean circuits of size T (Li et al., 2024; see also Feng et al., 2023). Counting is exactly the kind of computation at issue. Barbero et al. (2024) prove that decoder-only transformers can map distinct input sequences to arbitrarily close final representations, a collapse that produces errors "in, e.g.,

tasks involving counting or copying", and Zhang et al. (2024) show that counting in general-purpose models depends on both chain-of-thought and tokenization. The permuted-alphabet problems require locating items in a 26-element list and measuring distances between them. The separations these theoretical results establish are asymptotic: for a fixed list of 26 items, a sufficiently large constant-depth network could in principle compute positions and distances directly, and the theory does not say whether models of GPT-4's size do. What it supplies is a reason to expect a written trace to add serial computation of exactly this kind, which makes Unit B a different object from Unit A rather than the same object prompted differently.

The difference between Unit B and Unit C is that the executor is exact. An interpreter indexes a list without error regardless of length or position, which neither a forward pass nor a sampled trace guarantees. The published reply found that the trace recovered part of the gap and the interpreter recovered the rest (Webb et al., 2025), which is what this account predicts.

There is also a psychological reason not to treat Units A and B as the same participant. Commenting on Binz and Schulz's (2023) proposal to study GPT-3 as a participant in psychology experiments, Shiffrin and Mitchell (2023) observed that "Working memory is not part of GPT-3", and asked which aspect of a model's responses should be compared with humans at all. A written trace is the nearest thing such a model has to a working memory. A human participant solving a permuted-alphabet problem holds positions in mind or points along the list; Unit A has to do the equivalent inside one forward pass per token, and Unit B can write it down.

None of this makes Unit C evidence illegitimate. Tool-augmented language models are a mature research programme with their own evaluation norms (Schick et al., 2023; Gao et al., 2023; Gou et al., 2024), and Kambhampati et al. (2024) argue that language models cannot plan on their own but can help planning as components of frameworks that include external verifiers. The point is that in those programmes the claim is about the system: PAL reports that a model plus interpreter solves arithmetic, not that the model has become arithmetically competent. The exchange's round-three claim is about the component.

7.3 *A philosophical precedent, and its conditions*

There is a respectable argument for attributing the system's capacity to the agent. Clark and Chalmers (1998) proposed that when an external resource plays the functional role an internal process would play, it can count as part of the cognitive process. The reply's rerepresentation paragraph can be read as an appeal of this kind: the model decided on its own to write code, as a person might decide to count on their fingers, and the decision to rerepresent is itself a cognitive act (Webb et al., 2025).

The extended-mind argument does not, however, dissolve the distinction the exchange needed. It licenses attributing a capacity to an agent-plus-resource coupling, and on its own terms the coupling is the unit that has the capacity. It does not license the further claim that the unaugmented agent has it. A person who can solve permuted-alphabet problems only with a calculator would not be described as having shown that people solve them zero-shot. The reply's own concession, that code execution "is not a human-like solution (humans do not need to write code to count the items in a list)" (Webb et al., 2025), is an acknowledgement that the human comparison is between Unit C and an unaided human, which is precisely the kind of unequal comparison Firestone (2020) and Lampinen (2022) warn about, in the opposite direction from the usual one.

7.4 The critics' own lab took up the same unit

A telling development is that Mitchell's group took up the same unit after the exchange. Beger et al. (2025), with Mitchell as senior author, evaluated reasoning models on ConceptARC (Moskvichev et al., 2023) with Python tools on and off and at two reasoning budgets, and reported their rule-level analysis in the "medium effort + tools" setting. On text input, tools did not raise accuracy for three of the four models: o3 went from 77.1% at medium effort to 75.6% with tools, and Claude Sonnet 4 and Gemini 2.5 Pro were about five points lower with tools (60.2% to 55.0%, and 66.0% to 60.4%); o4-mini was the exception (70.8% to 77.7%). On visual input, tools helped o3 (5.6% to 29.2%) and o4-mini (8.1% to 25.0%) substantially and the other two models only slightly (5.2% to 6.9%, and 4.2% to 5.8%) (Beger et al., 2025, Table 1). The authors attribute the visual gains to the models using "computer vision libraries to partially compensate" for failing to recognise the grid size. That is structurally the reply's argument: an external executor compensates for a peripheral deficit, here perception rather than counting. What Beger et al. did with it differs. They did not infer from output accuracy that the abstraction was present. They graded the rules the models stated, and found 29% of o3's rules correct-but-unintended against 2.7% of humans' (Beger et al., 2025). The critics evaluated Units B and C side by side, chose the tools condition for their rule-level analysis, and added a check of whether the intended abstraction was used.

The same drift is visible elsewhere. McCoy et al. (2024) found that o1, a reasoning model, "substantially outperforms previous LLMs in many cases, with particularly large improvements on rare variants of common tasks", while remaining sensitive to the probability of examples and tasks, as McCoy et al. (2023) had found for earlier models. Camposampiero et al. (2025) found o3-mini dropping from 86.6% on I-RAVEN to 17.0% on I-RAVEN-X despite spending 3.4 times more reasoning tokens. The ARC Prize 2024 report attributes the year's gains to "deep learning-guided program

synthesis and test-time training" (Chollet et al., 2024), both system-level methods; Akyurek et al. (2024) reach average human performance on the ARC public validation set only when test-time training is ensembled with program synthesis. And the 2026 teaming study cited in the introduction measured frontier models at a low reasoning setting (Welsch et al., 2026). By 2026 the field's working unit for abstract-reasoning evaluation is B or C. The exchange's original question, about A, has not been answered; it has been set aside.

7.5 Why Unit A still matters

One might conclude that Unit A is obsolete and the question with it. There are two reasons not to. The first is the original claim's interest. Round one was exciting because it suggested that next-token training alone produces zero-shot analogical competence (Webb et al., 2023; Mitchell and Krakauer, 2023). Evidence about Unit C is evidence about next-token training plus instruction tuning plus tool-use training plus an exact executor. The second is mechanistic work, which is necessarily about Unit A. Yang et al. (2025) identified a three-stage "emergent symbolic architecture" in open models, with symbol abstraction heads, symbolic induction heads and retrieval heads, and extended the analysis to letter-string analogies. Their letter-string task used the standard alphabet with two worked examples, on which Llama-3.1 70B scored 99.2% on successor and 82.0% on predecessor problems. That is evidence about how Unit A solves familiar-alphabet problems. It is silent on the counterfactual problems, which are where the exchange turned. Opielka et al. (2025) found invariant concept vectors for verbal relations such as antonymy, but "for more abstract concepts like 'previous' and 'next', we do not observe invariant linear representations", a finding they link to the generalization failures. Lee et al. (2026) found that failures of analogical reasoning in models reflect missing relational information in mid-upper layers or failure to apply it to new entities. If the question about Unit A is dropped, the mechanistic findings lose the behavioural question they were meant to explain.

8. What was not settled: which step is auxiliary

8.1 The auxiliary-demand argument, stated carefully

Hu and Frank (2024) showed that evaluation methods with greater task demands yield lower measured performance, and that the gap is largest for smaller and less-trained models; their analogical-reasoning test was Digit Matrices, contrasting free production with forced choice. Firestone (2020) argued that human-machine comparisons should equate performance constraints so that superficial failures are not read as missing competence. The reply draws on both (Webb et al., 2025). The argument has a valid form: if a task requires operations X and Y, the capacity of interest is X, and the

system fails at Y for reasons unrelated to X, then failure on the task is not evidence against X.

Everything depends on two premises: that Y is not part of X, and that the failure is at Y. The exchange addressed the second premise with evidence and the first with assertion. The reply asserts that counting is auxiliary. The rejoinder asserts that successorship in the given alphabet is the target capacity. Neither paper states, before looking at data, what operations constitute analogical reasoning on a letter-string problem, so neither can show that the other has classified an operation wrongly. The comprehension checks and the interval probe, read together, allow more than either side drew from them.

8.2 What analogy theory says a letter-string problem requires

The classic componential analysis of analogical reasoning, developed for four-term problems of the form $A : B :: C : ?$, decomposes solution into encoding the terms, inferring the relation between A and B, mapping A onto C, and applying the inferred relation to C to produce the answer (Sternberg, 1977). Structure-mapping theory places the same emphasis on relations: analogy is the mapping of relational structure from a source to a target, and relations must be represented before they can be mapped (Gentner, 1983). Later models differ in how relations are represented and compared (Holyoak, 2012; Lu, Ichien and Holyoak, 2022), but none treats identifying the relation in the source as outside the analogy.

A zero-generalization successor problem, such as $[x\ y\ l\ k] \rightarrow [x\ y\ l\ w]$; $[j\ r\ q\ a] \rightarrow ?$, maps onto these components directly. Encoding: read the letters. Inference: recognise that w stands to k as its successor in the given order. Mapping: align the target string with the source. Application: produce the successor of a in the given order. In the Hodel-West variant the relation is "two places after", and inference requires recognising that too.

8.3 Two published numbers that may split "counting"

The comprehension check of Lewis and Mitchell tests application in isolation, at one step: given a letter and a named relation, produce the result. GPT-4 is at 1.0 for successor and 0.97 to 1.0 for predecessor, in partially permuted and symbol alphabets (Lewis and Mitchell, 2024b, Table 6). The next-previous task of Stevenson et al. (2025) tests the same operation at one and two steps, in four other models, and its results do not support "production intact" without qualification. Every model was worse at two steps than at one; Llama-3.1 405B failed on the symbol list in this task as it did in the analogies; Gemma-2 27B was uneven. Only Claude-3.5, in all three alphabets, and GPT-4o, in the Latin and Greek alphabets, showed a good check alongside degraded analogies. The authors read their check as

evidence that the models do not represent the novel alphabets, and suggest that the two-step weakness could explain the models' trouble with their second-successor analogy item (Stevenson et al., 2025).

The interval probe of Webb et al. (2025) tests inference in isolation: given two letters, say which relation holds between them. GPT-4 is at 10.8% for intervals between -2 and 2 in the Hodel-West alphabet. The main text reports only the overall figure, so the probe's one-step and two-step items cannot be separated here.

Figure 1(a) places the GPT-4 values beside the analogy accuracies from the same sources. The pattern is a dissociation between producing a named one-step relation and recognising an unnamed relation of up to two steps. Taken with analogy theory, it suggests that the failure of Unit A lies at the inference component. A second reading fits the same numbers, and Stevenson et al.'s two-step results favour it: that Unit A's representation of the new order supports one-step lookups and little else, so that recognition fails because the order is weakly encoded, not because inference fails on a well-encoded order. The reply's interpretation of its own probe is that it shows "counting is the source of GPT-4's difficulty on this task" (Webb et al., 2025). That is compatible with both readings here; on the first, the disagreement is about what to call the step. If identifying the relation between two items in an ordered set is counting, then counting is the inference step of these analogies. On the second, counting names the encoding of the order, which is also a component of analogy under the decomposition of Section 8.2.

Three caveats bound this reading. The comprehension checks and the interval probe come from different laboratories, alphabets and GPT-4 snapshots, so Figure 1(a) juxtaposes rather than controls. Lewis and Mitchell's permutations displaced at most 20 letters, while Hodel and West permuted all 26. And the production side was measured at one step, the recognition side at up to two, and the only two-step production data, from other models, show production weakening too. The within-source contrasts are the firmest part: Lewis and Mitchell's GPT-4 produced one-step relations near-perfectly and solved 0.452 of the zero-generalization analogies (averaged over all their alphabets, including the unpermuted one); Webb et al.'s GPT-4 recognised intervals 10.8% of the time and solved 21.7% of the analogies. Stevenson et al. (2025) do not replicate the dissociation in general: one of their four models shows it cleanly across all three alphabets, one shows it in two, and all four produce two-step neighbours less reliably than one-step ones. The dissociation is therefore a hypothesis supported by juxtaposing two laboratories' one-step production and mixed-step recognition data, not an established result, and Section 13 takes up the reading on which it is an artefact.

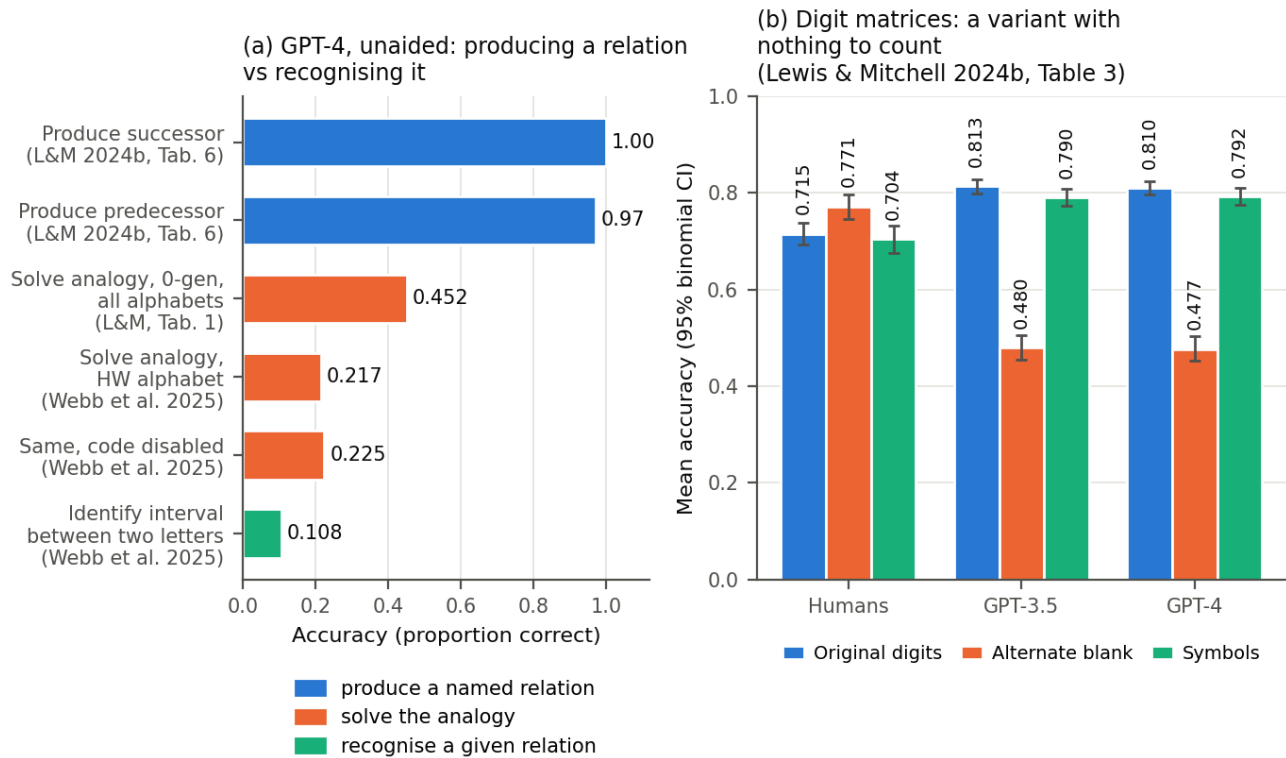


Figure 1. Published values only; no number here was measured by the author. (a) GPT-4 answering directly (Unit A) on operations inside a counterfactual letter-string problem. Blue: producing the successor or predecessor of a given item, displaced letters in partially permuted alphabets (Lewis and Mitchell, 2024b, Table 6; the symbol-alphabet values are 1.0 for both). Orange: solving the analogy, zero-generalization problems averaged over all alphabets, including the unpermuted one (Lewis and Mitchell, 2024b, Table 1), and on the fully permuted Hodel-West alphabet with answer-only prompting and with the code model told not to use code (Webb et al., 2025). Green: identifying the interval between two given letters, intervals -2 to 2, Hodel-West alphabet (Webb et al., 2025). The two sources used different GPT-4 snapshots and different alphabets (2 to 20 letters displaced versus a full permutation), and the production checks are one-step while the interval probe mixes one- and two-step intervals, so bars from different sources are not a controlled comparison; the within-source contrast is the point. (b) Digit matrices with the blank in its usual position, in a random position, and with digits replaced by symbols; means with 95% binomial confidence intervals (Lewis and Mitchell, 2024b, Table 3). Moving the blank requires no counting; the result was reported after the preprint reply and is not discussed in the published one.

8.4 What the code did, on this reading

On this reading the interpreter did not supply a peripheral operation that analogy then used. It rerepresented the problem so that the inference step became arithmetic on integers, a domain in which the model is reliable, and the model then performed the analogy over integers. The reply's example shows exactly this: the code maps [q a h v] and [q a h m] to [12, 13, 14, 15] and [12, 13, 14, 17], and the model observes that the last index rises by 2 (Webb et al., 2024). Lewis and Mitchell (2024b) describe the same operation as translating the problem into "the equivalent numerical one".

The reply's defence is that rerepresentation is part of human analogy, and that is true (Webb et al., 2025). But the defence changes which part of the performance is doing the analogical work. What the code condition shows is that GPT-4 can recognise that an unfamiliar problem would become tractable if recast in a familiar representation, and can write the program that recasts it. That is a real and interesting capacity, and arguably

closer to the problem-solving use of analogy emphasised by Gick and Holyoak (1980) than a letter-string problem is. It is not evidence that the model can recognise successorship in an unfamiliar order, which is what the counterfactual variants were built to test. In the terms of Chollet (2019), the difficulty of generalizing to a task is measured relative to the system's priors and experience; recasting the problem into integers moves it into the region of greatest experience, so success after recasting is weaker evidence of generalization than success on the original representation would be. Where rerepresentation into integers helped least, on fix-alphabetic-sequence and sort problems, the code model rarely gave the intended answer: it gave it on none of ten fix problems at either interval in the preprint's data, although 6 and 10 of those errors respectively followed a valid alternative rule (Webb et al., 2024). On those problems the code model found a rule, but not the one the problems were built around.

8.5 A criterion the exchange could have fixed in advance

The general lesson is that "auxiliary" cannot be assigned after seeing which operation failed. A criterion that could be fixed in advance follows from the logic the parties already used. An operation Y in a task is auxiliary to a capacity X when three conditions hold. First, Y is not a component of X under the theory of X that the claim appeals to; for analogy on Sternberg's account, that excludes encoding the terms and inferring the source relation. Second, Y can be probed in isolation using the task's own materials. Third, the claim that Y caused the failure predicts something testable beyond the task score: for example, that supplying Y's output directly, without supplying anything else, restores performance. Round two applied a version of the second condition and found one-step application intact. Round three applied the second condition to a different operation and found it broken, but did not address the first condition. Section 12 proposes the test the third condition calls for.

Stated this way, the disagreement becomes empirical, although the test locates the failure rather than settling the label. If supplying the model with the indices in the prompt, with no interpreter and no trace, restores Unit A to human level, then the bottleneck was locating letters in a list, a lookup. Under Sternberg's decomposition such a lookup belongs to encoding, which is itself a component, so Algorithm 1 (Section 11) would still record a failure inside analogy; but it would be a failure at encoding, not at inference, and this paper's localisation to inference would be wrong. The reply's classification would then be defensible on any account that treats encoding the terms as preparation for analogy rather than part of it, which is how the published reply describes counting: an operation needed "to generate the inputs to analogical reasoning" (Webb et al., 2025). If supplying the indices does not restore performance, the bottleneck survives the removal of counting, and the failure lies at inference or later, as both the rejoinder's reading and the first reading of Section 8.3 predict. Neither result is in the literature located for this paper.

9. Two domains the counting account does not reach

The counting account explains, at most, the permuted-alphabet failures. The extended critique reported failures in two other domains (Lewis and Mitchell, 2024b). As Section 2 notes, those results were posted in November 2024, seven months after the preprint reply, and the published reply, which was received by its journal the following month, cites only the conference version of the critique, which covered letter strings alone. The reply cannot fairly be faulted for not answering results that postdate its argument. The question here is different: whether the account it offers could explain them.

The Digit Matrix alternate-blank variant is the cleanest. Nothing about it involves counting or an unfamiliar order: the digits and rules are the originals, and only the position of the missing cell changes. Humans were unaffected (0.715 to 0.771); GPT-3.5 and GPT-4 each dropped by 0.33 (0.813 to 0.480, and 0.810 to 0.477), and GPT-4 identified the blank's position correctly in a comprehension check (Lewis and Mitchell, 2024b). If the auxiliary-demand argument explains the letter-string failures, something else has to explain this one. Speculatively, one candidate follows the reasoning of McCoy et al. (2023): a missing cell at the end of the last row coincides with next-token prediction, while a missing cell elsewhere does not. That is a hypothesis about Unit A's architecture, which is again a claim that Unit C evidence would not touch. Digit Matrices were the evidence round one presented as strongest, because they were novel. The variant shows that novelty of content did not guarantee robustness to format. The reply's defence of Digit Matrices as novel (Webb et al., 2024) was written before the variant was reported, and so could not engage it; no later response located for this paper does.

The story-analogy order effect is smaller in sample (18 problems) but also unanswered: GPT-4 was correct on every problem when the analogous story came first and on 13 of 18 when it came second, while humans showed no order effect (Lewis and Mitchell, 2024b). Stories were the one domain in which round one found GPT-3 clearly worse than humans, so this result corrects the record less than the other two. It does show that round two's case did not rest on letter strings alone.

An account that reaches one of three domains, and in that domain establishes its result for a different unit, does not yet answer the critique as it now stands. That is a statement about the current state of the evidence, not a criticism of the reply's authors, who wrote before two of the three domains were reported. The two results the counting account does not reach are also the ones that most directly threaten round one's novelty argument and its human-signature argument respectively.

10. The numbers in one place

Table 1 collects the measured values the argument rests on. Every row names its source; the unit column uses the classification of Section 2. Rows marked "derived" were computed by this paper's author from counts published in the source, and the computation is stated.

Three things are visible in the table that no single row shows. First, the column of units: the rows that establish the counterfactual failure (4 to 8, 13, 16, 17) are all Unit A, and the rows that establish human-level success on those problems (14, 18) are Unit C. No row shows Unit A at human level on a counterfactual letter-string problem. Second, rows 7, 8 and 17 are the direct probes of sub-operations for GPT-4, and they point in opposite

Table 1. Measured values from the sources, with the unit each was measured on (A: direct answer; B: with a trace; C: with an external executor). "Derived" rows are sums of published counts. Values from different sources used different materials and model snapshots and are not directly comparable across rows.

#	Source	What was measured	Value	Unit
1	Webb et al. (2023)	GPT-3 vs 40 students, Digit Matrices, generative	odds ratio 1.88, $p = 0.005$	A
2	Webb et al. (2023)	GPT-3 vs 57 students, letter strings, overall	odds ratio 1.76	A
3	Webb et al. (2023)	GPT-3 vs 54 students, story analogies	odds ratio 0.37 (students better)	A
4	Hodel and West (2023)	GPT-3, synthetic alphabet and interval 2, four of six transformation types	accuracy below 0.1	A
5	Lewis and Mitchell (2024a), Table 2	Letter strings, all alphabets (including the unpermuted one): humans / GPT-3 / GPT-3.5 / GPT-4	0.753 / 0.488 / 0.350 / 0.452	A
6	Lewis and Mitchell (2024b), Table 1	Same, one / two / three generalizations: humans vs GPT-4	0.358 / 0.317 / 0.260 vs 0.271 / 0.219 / 0.195	A
7	Lewis and Mitchell (2024b), Table 6	GPT-4 comprehension check, successor (in place / displaced / symbols)	1.0 / 1.0 / 1.0	A
8	Lewis and Mitchell (2024b), Table 6	GPT-4 comprehension check, predecessor (same columns)	1.0 / 0.97 / 1.0	A
9	Lewis and Mitchell (2024b), Table 3	Digit Matrices original / alternate blank / symbols, humans	0.715 / 0.771 / 0.704	human
10	Lewis and Mitchell (2024b), Table 3	Same, GPT-4 (GPT-3.5 in brackets)	0.810 / 0.477 / 0.792 (0.813 / 0.480 / 0.790)	A
11	Lewis and Mitchell (2024b), Table 4	GPT-4 story analogies, correct answer first vs second	18/18 vs 13/18 (humans 0.78 vs 0.78)	A
12	Lewis and Mitchell (2024b), Table 8	Share of errors using a valid alternative rule, GPT-4 vs humans	22.0% vs 38.6%	A
13	Stevenson et al. (2025)	Four models, analogy accuracy Latin to Greek to symbols; next-previous check at one and two steps	significant decline at each step; every model worse at two steps; check good for Claude-3.5 in all three alphabets, GPT-4o in two, Llama-3.1 poor on symbols	A
14	Webb et al. (2024), Tables 1-2 (derived)	GPT-4 with code, problems solved at interval 1 / interval 2	40/60 / 30/60	C
15	Webb et al. (2024), Tables 1-2	GPT-4 with code, fix-alphabetic-sequence, intended answer at interval 1 / 2 (errors using a valid alternative rule)	0/10 (6 of 10) / 0/10 (10 of 10)	C
16	Webb et al. (2025)	GPT-4 alone vs code model told not to use code, Hodel-West problems	21.7% vs 22.5%, $p = 0.73$	A
17	Webb et al. (2025)	GPT-4 identifying interval between two letters, intervals -2 to 2	10.8%	A
18	Webb et al. (2025)	Humans vs GPT-4 with code; GPT-4 vs GPT-4 with chain-of-thought	$p = 0.92$; $p = 0.004$ (trace still below humans, $p = 3 \times 10^{-8}$)	C; B
19	Yang et al. (2025)	Llama-3.1 70B, standard alphabet, 2-shot, successor / predecessor	99.2% / 82.0%	A
20	Beger et al. (2025), Table 1	o3 ConceptARC, medium effort, text without / with tools; visual without / with tools	77.1 / 75.6; 5.6 / 29.2 (%)	B; C
21	Beger et al. (2025)	Correct-but-unintended rules, o3 vs humans	about 29% vs 2.7%	C
22	Camposampiero et al. (2025)	o3-mini, I-RAVEN vs I-RAVEN-X with perceptual uncertainty	86.6% vs 17.0%	B
23	Welsch et al. (2026)	Frontier chat models, letter-string advice correct (3,384 trials)	0.846	B
24	Hellwig et al. (2026)	3-layer transformer trained with copy tasks, seen vs new permuted alphabets (few training alphabets)	90.7% vs 33.3%	trained

directions for one-step production and mixed-step recognition; row 13 shows that in other models production itself weakens at two steps. Third, rows 9 to 11 are failures reported after the preprint reply, and row 10 is a failure no counting account can reach.

11. The decision structure

The exchange can be summarised as a procedure for reading a capacity claim from a counterfactual-task result. The procedure below states the questions that the

three rounds answered, and the ones they left implicit, in the order in which they have to be answered. It is not a method for building a system.

Steps 1 and 2 are where the exchange's evidence moved while its claim did not. Step 4 is the one the exchange skipped: neither side fixed a decomposition in advance, so step 7 could be argued either way. Step 8 is the test the auxiliary-demand argument calls for and nobody ran; it is also what would tell an encoding failure from an inference failure. Step 10 is where the counting account

Algorithm 1. Reading a claim "model M has capacity X" from a task T and a counterfactual variant T'.

```

Input : task T, variant T' that preserves X's structure and changes
        surface; results for M on T and T'; human results on both.
Output: the unit the evidence supports, and the claim it licenses.

1: U <- the unit each result was measured on:
    A (direct answer) | B (self-generated trace) | C (external executor)
2: if results on T and T' come from different units:
    report them as results about different objects (Sec. 7)
3: if M succeeds on T and fails on T' in unit A, and humans do not:
    X is not shown robust in unit A (Sec. 6)
4: D <- decomposition of T' into operations under the theory of X
    (for analogy: encode, infer relation, map, apply) (Sec. 8.2)
    -- fix D before running any probe
5: for each operation d in D and each proposed auxiliary operation y:
    probe it in isolation on T' materials (comprehension checks)
6: if every probe of a component of X passes and T' still fails:
    the failure is in combining components, i.e. in X (Sec. 8.3)
7: if a probe fails:
    if the failing operation is a component of X under D:
        the failure is in X, whatever name it is given (Sec. 8.3)
    else:
        candidate auxiliary failure
        in either case go to 8 to locate it
8: test: supply the failing operation's output directly in the prompt,
    with no trace and no executor
    if unit-A performance on T' reaches the human level:
        the failure is at that operation: an auxiliary failure if it
        lies outside D, else a failure in the component of D that
        contains it (Sec. 8.5)
    else:
        that operation is not the cause; the failure lies elsewhere
9: if an executor restores performance (unit C):
    record which operation it performed, and whether that operation
    is in D; report the capacity as the system's (Sec. 7.3)
10: for every other variant T'' on which M failed:
    check whether the auxiliary account reaches it (Sec. 9)
11: return (unit, capacity supported | not shown | not tested, reasons)

```

is incomplete.

Applied to the exchange, the procedure returns the following. For Unit A on permuted and symbol letter strings, analogy on unfamiliar orders is not shown: step 3 applies; the one-step production probes pass for GPT-4; and step 7 places the one failing probe (interval recognition) inside D, where step 8 has not been run to say whether it lies at encoding or at inference. For Unit C, the capacity to solve the Hodel-West problems at human level is supported, with the executor performing index lookup and arithmetic, which is how the inference step was carried out (step 9). For Digit Matrices with a moved blank, step 10 finds no auxiliary account on offer.

12. Tests that would decide the open questions

Each proposal below uses existing materials and publicly released problem sets (Hodel and West, 2023; Lewis and Mitchell, 2024a, 2024b; Webb et al., 2024). None requires a new model or a new benchmark. The predictions are the ones the two readings make; the paper does not claim to know the results.

Test 1: supply the indices, remove the executor. Present the Hodel-West problems to a model answering directly, with each letter in the problem annotated by its position in the permuted alphabet, for example "k(4) w(5) b(6) f(7)". This supplies the output of counting without supplying a trace or an interpreter. If the reply's

account is right, Unit A should approach the code condition, because the only missing operation has been provided. If the recognition reading is right, Unit A should improve only as far as the problem has become the familiar integer-sequence problem, and the improvement then shows rerepresentation, not analogy over the unfamiliar order. A matched human condition, in which participants see the same annotations, would show whether the annotation helps humans too, which is what a species-fair comparison requires (Firestone, 2020).

Test 2: a recognition probe on the same materials as the production probe. Run Lewis and Mitchell's comprehension check, extended to two steps, and Webb et al.'s interval probe, reported by interval size, on the same alphabets, the same snapshot and the same prompt format, for Units A, B and C. The dissociation in Figure 1(a) currently rests on two laboratories' data and is partly contradicted by a third's. A single study that measured production, recognition and analogy accuracy together, at matched step sizes, would either confirm it or show that it was an artefact of different materials or of step size.

Test 3: code on the variants where counting is not the issue. Give the code model the symbol-alphabet problems, which the reply did not test, and the alternate-blank Digit Matrices. The counting account predicts benefit on the first and none on the second. A benefit on

the second would mean the executor is doing more than counting; no benefit on either would narrow the reply's result to the Hodel-West alphabet.

Test 4: reasoning models with the executor disabled, at matched budgets. Reasoning models blur Units A and B. Evaluating them on the Lewis-Mitchell alphabets with tools off, across reasoning budgets, would show how much of the gap a trace closes on its own. The published chain-of-thought condition used zero-shot prompting on GPT-4 (Webb et al., 2025); a reasoning model's trained trace is a stronger test. Welsch et al. (2026) suggest that performance is now high with a trace, but their design does not separate the trace from any other factor.

Test 5: grade the rule, not just the answer. Following Beger et al. (2025), collect the model's stated rule for each letter-string answer and classify it as correct-intended, correct-unintended or incorrect. Webb et al. (2024) report that correct answers were "always accompanied by an accurate explanation" in their code condition, and Lewis and Mitchell (2024b) found that GPT-4's errors differed in kind from humans'. A rule-level comparison across units would show whether the code condition's correct answers rest on successorship in the given order or on integer arithmetic, which is the question on which the reply and rejoinder disagree.

A sixth recommendation is procedural. A study claiming or denying a reasoning capacity from counterfactual variants should state its decomposition of the task, and which operations it treats as auxiliary, before collecting data. Lin (2025) argues that psychological research on language models needs both psychometric validation of what a score means and causal identification of what produced it; the auxiliary-demand dispute is a case in which the absence of the first made the second impossible to adjudicate.

13. The strongest case against this paper

The best objection is that the paper takes one cognitive theory's decomposition as authoritative. Sternberg's component analysis was developed for four-term verbal and geometric analogies, and letter strings in the Copycat tradition were designed around a different view, in which perception of the situation and analogy-making are intertwined rather than sequential (Mitchell and Hofstadter, 1990; Mitchell, 2021). On that view there is no clean "inference" step to place counting inside or outside, and the production/recognition split may be an artefact of forcing a sequential model onto a fluid process. This paper's answer is that the objection cuts against the reply at least as much as against this paper. If the processes cannot be separated, the reply cannot classify counting as auxiliary either, and the auxiliary-demand argument has nothing to stand on. The paper's claim is conditional: on the component theory both sides implicitly used, the failing step is inside analogy. Under

a theory with no components, no step is auxiliary.

A second objection is that the comprehension checks are too easy to show anything. Naming the successor of a letter in a list stated in the prompt may be solvable by copying the next token from the context, which is what induction heads do (Olsson et al., 2022), without any grasp of order. If so, the near-perfect production scores measure copying, not successorship, and the dissociation in Figure 1(a) is between copying and reasoning rather than between production and recognition. This objection has force, and the paper cannot fully answer it. If the only operation Unit A performs reliably on the counterfactual alphabet is local copying, then nothing in the evidence shows that Unit A represents the order at all, which is close to the conclusion Stevenson et al. (2025) draw from their own two-step check. In that case the application component was never tested in any rich sense, the failure could lie at encoding the order as much as at inferring the relation, and the localisation to inference in Section 8.3 loses its support. What survives is weaker: the evidence does not show the failure to be auxiliary, because on either reading it falls inside a component of analogy under the decomposition the paper adopts. Tests 1 and 2 of Section 12 would separate the two readings.

A third objection is that the code model's success does show analogy, since the model had to recognise, from the source pair expressed in integers, that the last index rose by two, and then apply that to the target. That is an analogy, just over integers. This is correct, and the paper does not deny it. The point is that integer sequences are the most familiar ordered domain a language model knows, and the counterfactual variants were built to test analogy over an unfamiliar one. A result showing analogy over a familiar domain after a rerepresentation is a result about the rerepresentation. Whether rerepresentation into a familiar domain is itself the essence of analogy, as the reply's appeal to Gestalt psychology suggests (Webb et al., 2025), is a genuine theoretical question that this paper cannot settle.

A fourth objection is that the paper relies on derived numbers from the preprint's error tables. The derived values (rows 14 and 15 of Table 1) are sums of counts the reply printed, and the reply's own text states that the code model answered 9 of 10 extend-sequence problems at interval two wrongly. The published version reports a different alternative-rule share (38% rather than 46%), which suggests that its data differ from the preprint's; the derived numbers should be read as describing the preprint's sample only.

14. What is not known

It is not known whether supplying letter indices in the prompt, without an executor or a trace, restores the unaided model's performance on permuted alphabets. It is not known whether production and recognition of

order relations dissociate when measured in one study with one model and one set of materials, or whether GPT-4's production of neighbours in a permuted alphabet holds up at two steps. It is not known whether the code-augmented model benefits on symbol alphabets, or on Digit Matrices with the blank moved. It is not known what explains the alternate-blank drop at all. It is not known whether current reasoning models, with tools disabled, solve the Lewis-Mitchell alphabets at human level, or whether their stated rules on those problems use the intended relations. The mechanistic studies located for this paper examined familiar alphabets; nobody has traced how a model processes a counterfactual letter-string problem, successful or not. Finally, the exchange never produced an agreed decomposition of a letter-string problem into component operations, and without one the question "was the failure auxiliary?" has no answer that both sides would accept.

15. Limitations

The most serious limitation is that the central dissociation is assembled from two laboratories' measurements with different materials, prompts, model snapshots and step sizes, and that the one study measuring two-step production, in other models, points partly against it. Figure 1(a) and the argument of Section 8.3 depend on reading within-source contrasts across sources. A single controlled study could overturn it, and Test 2 is proposed for that reason.

The paper read the main text of the published reply but not its supplementary material. Some details of the published conditions, such as the temperature and interface used for the chain-of-thought condition, are known only for the preprint. Where the paper describes the preprint's methods it says so, and the confound analysis in Section 5.1 is scoped to the preprint.

The paper classifies units by what the sources report. Some sources do not state whether tools were available; Welsch et al. (2026) report a reasoning setting, and the paper treats them as Unit B on that basis. If tools were available in their setup, row 23 of Table 1 belongs to Unit C.

The three rounds were written by cognitive scientists and published partly in cognitive-science venues, and the paper judges them by cognitive-science standards of decomposition and dissociation. A reader who holds that capacity claims about machine learning systems should be read off task performance under stated conditions, with no reference to cognitive components, will find Section 8 beside the point, although Sections 6 and 7 should still apply.

Finally, the paper covers one exchange closely and the surrounding literature selectively. Verbal analogies, which round one tested and round two deliberately did not (Lewis and Mitchell, 2024b), are outside its scope, as are narrative-analogy benchmarks such as ARN (Sourati

et al., 2024), on which models also struggle with far analogies. The paper's conclusions are about letter-string and matrix problems in this exchange and should not be extended past them without separate evidence.

16. Conclusion

The exchange settled more than its tone suggests. Replications agree; the unaided model fails counterfactual letter strings; a model with an interpreter solves the permuted problems at human level, and the interpreter is why. GPT-4 can also name one-step neighbours in a permuted alphabet, though whether models represent such an alphabet beyond one step is disputed. What moved was the object the evidence was about. Rounds one and two tested a model answering directly. Round three's decisive evidence came from a model with an executor, and its conclusion stayed about language models. What was not settled is which operations belong to analogy. The published numbers suggest splitting "counting" into producing a named one-step relation, which GPT-4 does well, and recognising an unnamed one, which it does poorly; evidence from other models that two-step production also weakens leaves open whether the deeper problem is the representation of the order itself. If the difficulty is recognition of the source relation, it is the inference step of a componential account of analogy; if it is the representation of the order, it is encoding, which the same account also counts as part of analogy. On that account, either way, the reply's "auxiliary" label names a component of analogy, and the code solved the problem by rerepresenting it into the domain the counterfactual variants had been built to avoid.

None of this shows that language models cannot reason by analogy. It shows that the exchange's third round answered a different question from its first, and that the dispute would have been decidable if either side had committed, before looking at data, to a decomposition of the task and a test of the auxiliary claim. Those commitments cost little, and the materials to act on them are already public.

Acknowledgements

This paper was prepared with AI assistance, as described in the front matter. No external reviewer read it before release. The author thanks the authors of all five studies in the exchange for releasing code, data and, in Webb et al.'s case, the complete model transcripts, without which the confound analysis would not have been possible. Errors of reading or classification are the author's.

References

- Akyurek, E., Damani, M., Zweiger, A., Qiu, L., Guo, H., Pari, J., Kim, Y. & Andreas, J. (2024). The Surprising Effectiveness of Test-Time Training for Few-Shot Learning. arXiv preprint. arXiv:2411.07279. doi:10.48550/arXiv.2411.07279.

- Barbero, F., Banino, A., Kapturowski, S., Kumaran, D., Araujo, J. G. M., Vitvitskiy, A., Pascanu, R. & Velickovic, P. (2024). Transformers need glasses! Information over-squashing in language tasks. *NeurIPS 2024*. arXiv:2406.04267. doi:10.48550/arXiv.2406.04267.
- Beger, C., Yi, R., Fu, S., Denton, K., Moskvichev, A., Tsai, S. W., Rajamanickam, S. & Mitchell, M. (2025). Do AI Models Perform Human-like Abstract Reasoning Across Modalities? arXiv preprint. arXiv:2510.02125. doi:10.48550/arXiv.2510.02125.
- Binz, M. & Schulz, E. (2023). Using cognitive psychology to understand GPT-3. *Proceedings of the National Academy of Sciences* 120(6), e2218523120. arXiv:2206.14576. doi:10.48550/arXiv.2206.14576.
- Camposampiero, G., Hersche, M., Wattenhofer, R., Sebastian, A. & Rahimi, A. (2025). Can Large Reasoning Models do Analogical Reasoning under Perceptual Uncertainty? *NeSy 2025*. arXiv:2503.11207. doi:10.48550/arXiv.2503.11207.
- Chollet, F. (2019). On the Measure of Intelligence. arXiv preprint. arXiv:1911.01547. doi:10.48550/arXiv.1911.01547.
- Chollet, F., Knoop, M., Kamradt, G. & Landers, B. (2024). ARC Prize 2024: Technical Report. arXiv preprint. arXiv:2412.04604. doi:10.48550/arXiv.2412.04604.
- Clark, A. & Chalmers, D. (1998). The Extended Mind. *Analysis* 58(1), 7-19. doi:10.1093/analysis/58.1.7.
- Feng, G., Zhang, B., Gu, Y., Ye, H., He, D. & Wang, L. (2023). Towards Revealing the Mystery behind Chain of Thought: A Theoretical Perspective. *NeurIPS 2023*. arXiv:2305.15408. doi:10.48550/arXiv.2305.15408.
- Firestone, C. (2020). Performance vs. competence in human-machine comparisons. *Proceedings of the National Academy of Sciences* 117(43), 26562-26571. doi:10.1073/pnas.1905334117.
- Gao, L., Madaan, A., Zhou, S., Alon, U., Liu, P., Yang, Y., Callan, J. & Neubig, G. (2023). PAL: Program-aided Language Models. *ICML 2023*. arXiv:2211.10435. doi:10.48550/arXiv.2211.10435.
- Gentner, D. (1983). Structure-Mapping: A Theoretical Framework for Analogy. *Cognitive Science* 7(2), 155-170. doi:10.1207/s15516709cog0702_3.
- Gick, M. L. & Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology* 12(3), 306-355. doi:10.1016/0010-0285(80)90013-4.
- Golchin, S. & Surdeanu, M. (2024). Time Travel in LLMs: Tracing Data Contamination in Large Language Models. *ICLR 2024*. arXiv:2308.08493. doi:10.48550/arXiv.2308.08493.
- Gou, Z., Shao, Z., Gong, Y., Shen, Y., Yang, Y., Huang, M., Duan, N. & Chen, W. (2024). ToRA: A Tool-Integrated Reasoning Agent for Mathematical Problem Solving. *ICLR 2024*. arXiv:2309.17452. doi:10.48550/arXiv.2309.17452.
- Hellwig, P., Zuidema, W., Stevenson, C. E. & Lewis, M. (2026). Transformer See, Transformer Do: Copying as an Intermediate Step in Learning Analogical Reasoning. *COLM 2026*. arXiv:2604.06501. doi:10.48550/arXiv.2604.06501.
- Hodel, D. & West, J. (2023). Response: Emergent analogical reasoning in large language models. arXiv preprint. arXiv:2308.16118. doi:10.48550/arXiv.2308.16118.
- Holyoak, K. J. (2012). Analogy and Relational Reasoning. In *The Oxford Handbook of Thinking and Reasoning*, 234-259. Oxford University Press. doi:10.1093/oxfordhb/9780199734689.013.0013.
- Hu, J. & Frank, M. C. (2024). Auxiliary task demands mask the capabilities of smaller language models. *COLM 2024*. arXiv:2404.02418. doi:10.48550/arXiv.2404.02418.
- Kambhampati, S., Valmeekam, K., Guan, L., Verma, M., Stechly, K., Bhambri, S., Saldyt, L. & Murthy, A. (2024). LLMs Can't Plan, But Can Help Planning in LLM-Modulo Frameworks. *ICML 2024*. arXiv:2402.01817. doi:10.48550/arXiv.2402.01817.
- Kojima, T., Gu, S. S., Reid, M., Matsuo, Y. & Iwasawa, Y. (2022). Large Language Models are Zero-Shot Reasoners. *NeurIPS 2022*. arXiv:2205.11916. doi:10.48550/arXiv.2205.11916.
- Lampinen, A. K. (2022). Can language models handle recursively nested grammatical structures? A case study on comparing models and humans. arXiv preprint. arXiv:2210.15303. doi:10.48550/arXiv.2210.15303.
- Lee, T., Song, M., Yoon, C., Park, J. & Kang, J. (2026). The Curious Case of Analogies: Investigating Analogical Reasoning in Large Language Models. *AAAI 2026*. arXiv:2511.20344. doi:10.48550/arXiv.2511.20344.
- Lewis, M. & Mitchell, M. (2024a). Using Counterfactual Tasks to Evaluate the Generality of Analogical Reasoning in Large Language Models. *Proceedings of the Annual Meeting of the Cognitive Science Society* 46. arXiv:2402.08955. doi:10.48550/arXiv.2402.08955.
- Lewis, M. & Mitchell, M. (2024b). Evaluating the Robustness of Analogical Reasoning in Large Language Models. arXiv preprint. arXiv:2411.14215. doi:10.48550/arXiv.2411.14215.
- Li, Z., Liu, H., Zhou, D. & Ma, T. (2024). Chain of Thought Empowers Transformers to Solve Inherently Serial Problems. *ICLR 2024*. arXiv:2402.12875. doi:10.48550/arXiv.2402.12875.
- Lin, Z. (2025). From Prompts to Constructs: A Dual-Validity Framework for Large Language Model Research in Psychology. *Annual Review of Psychology* 78 (in press). arXiv:2506.16697. doi:10.48550/arXiv.2506.16697.
- Lu, H., Ichien, N. & Holyoak, K. J. (2022). Probabilistic analogical mapping with semantic relation networks. *Psychological Review* 129(5), 1078-1103. doi:10.1037/rev0000358.
- McCoy, R. T., Yao, S., Friedman, D., Hardy, M. & Griffiths, T. L. (2023). Embers of Autoregression: Understanding Large Language Models Through the Problem They are Trained to Solve. arXiv preprint. arXiv:2309.13638. doi:10.48550/arXiv.2309.13638.
- McCoy, R. T., Yao, S., Friedman, D., Hardy, M. D. & Griffiths, T. L. (2024). When a language model is optimized for reasoning, does it still show embers of autoregression? An analysis of OpenAI o1. arXiv preprint. arXiv:2410.01792. doi:10.48550/arXiv.2410.01792.
- Mitchell, M. (2021). Abstraction and Analogy-Making in Artificial Intelligence. *Annals of the New York Academy of Sciences*. arXiv:2102.10717. doi:10.48550/arXiv.2102.10717.
- Mitchell, M. & Hofstadter, D. R. (1990). The emergence of understanding in a computer model of concepts and analogy-making. *Physica D: Nonlinear Phenomena* 42(1-3), 322-334. doi:10.1016/0167-2789(90)90086-5.
- Mitchell, M. & Krakauer, D. C. (2023). The Debate Over Understanding in AI's Large Language Models. *Proceedings of the National Academy of Sciences* 120(13). arXiv:2210.13966. doi:10.48550/arXiv.2210.13966.
- Moskvichev, A., Odouard, V. V. & Mitchell, M. (2023). The ConceptARC Benchmark: Evaluating Understanding and Generalization in the ARC Domain. *Transactions on Machine Learning Research*. arXiv:2305.07141. doi:10.48550/arXiv.2305.07141.
- Nye, M., Andreassen, A. J., Gur-Ari, G., Michalewski, H., Austin, J., Bieber, D., Dohan, D., Lewkowycz, A., Bosma, M., Luan, D., Sutton, C. & Odena, A. (2021). Show Your Work: Scratchpads for Intermediate Computation with Language Models. arXiv preprint. arXiv:2112.00114. doi:10.48550/arXiv.2112.00114.
- Olsson, C., Elhage, N., Nanda, N., Joseph, N., DasSarma, N., Henighan, T., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., Drain, D., et al. (2022). In-context Learning and Induction Heads. arXiv preprint. arXiv:2209.11895. doi:10.48550/arXiv.2209.11895.
- Opielka, G., Rosenbusch, H. & Stevenson, C. E. (2025). Analogical Reasoning Inside Large Language Models: Concept Vectors and the Limits of Abstraction. arXiv preprint. arXiv:2503.03666. doi:10.48550/arXiv.2503.03666.
- Opielka, G., Rosenbusch, H., Vijverberg, V. & Stevenson, C. E. (2024). Do Large Language Models Solve ARC Visual Analogies Like People Do? *CogSci 2024*. arXiv:2403.09734. doi:10.48550/arXiv.2403.09734.

- Sainz, O., Campos, J. A., Garcia-Ferrero, I., Etxaniz, J., Lopez de Lacalle, O. & Agirre, E. (2023). NLP Evaluation in trouble: On the Need to Measure LLM Data Contamination for each Benchmark. Findings of EMNLP. arXiv:2310.18018. doi:10.48550/arXiv.2310.18018.
- Schaeffer, R., Miranda, B. & Koyejo, S. (2023). Are Emergent Abilities of Large Language Models a Mirage? arXiv preprint. arXiv:2304.15004. doi:10.48550/arXiv.2304.15004.
- Schick, T., Dwivedi-Yu, J., Dessi, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N. & Scialom, T. (2023). Toolformer: Language Models Can Teach Themselves to Use Tools. arXiv preprint. arXiv:2302.04761. doi:10.48550/arXiv.2302.04761.
- Shiffrin, R. & Mitchell, M. (2023). Probing the psychology of AI models. Proceedings of the National Academy of Sciences 120(10), e2300963120. doi:10.1073/pnas.2300963120.
- Sourati, Z., Ilievski, F., Sommerauer, P. & Jiang, Y. (2024). ARN: Analogical Reasoning on Narratives. arXiv preprint. arXiv:2310.00996. doi:10.48550/arXiv.2310.00996.
- Sternberg, R. J. (1977). Component processes in analogical reasoning. Psychological Review 84(4), 353-378. doi:10.1037/0033-295X.84.4.353.
- Stevenson, C. E., ter Veen, M., Choenni, R., van der Maas, H. L. J. & Shutova, E. (2023). Do large language models solve verbal analogies like children do? arXiv preprint. arXiv:2310.20384. doi:10.48550/arXiv.2310.20384.
- Stevenson, C. E., Pafford, A., van der Maas, H. L. J. & Mitchell, M. (2025). Can Large Language Models generalize analogy solving like children can? Transactions of the Association for Computational Linguistics. arXiv:2411.02348. doi:10.48550/arXiv.2411.02348.
- Webb, T., Holyoak, K. J. & Lu, H. (2023). Emergent Analogical Reasoning in Large Language Models. Nature Human Behaviour 7(9), 1526-1541. doi:10.1038/s41562-023-01659-w. arXiv:2212.09196. doi:10.48550/arXiv.2212.09196.
- Webb, T., Holyoak, K. J. & Lu, H. (2024). Evidence from counterfactual tasks supports emergent analogical reasoning in large language models. arXiv preprint (v2). arXiv:2404.13070. doi:10.48550/arXiv.2404.13070.
- Webb, T. W., Holyoak, K. J. & Lu, H. (2025). Evidence from counterfactual tasks supports emergent analogical reasoning in large language models. PNAS Nexus 4(5), pgaf135. doi:10.1093/pnasnexus/pgaf135.
- Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., et al. (2022a). Emergent Abilities of Large Language Models. Transactions on Machine Learning Research. arXiv:2206.07682. doi:10.48550/arXiv.2206.07682.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q. & Zhou, D. (2022b). Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. NeurIPS 2022. arXiv:2201.11903. doi:10.48550/arXiv.2201.11903.
- Welsch, R., Rausch, M., Knierim, P., Kosch, T., Kuhn, J., Schmidt, A. & Fernandes, D. (2026). Available but Unclaimed: An Empirical Study of Human-AI Synergy. arXiv preprint. arXiv:2609.16793. doi:10.48550/arXiv.2609.16793.
- Wu, Z., Qiu, L., Ross, A., Akyurek, E., Chen, B., Wang, B., Kim, N., Andreas, J. & Kim, Y. (2024). Reasoning or Reciting? Exploring the Capabilities and Limitations of Language Models Through Counterfactual Tasks. NAACL 2024. arXiv:2307.02477. doi:10.48550/arXiv.2307.02477.
- Yang, Y., Campbell, D., Huang, K., Wang, M., Cohen, J. & Webb, T. (2025). Emergent Symbolic Mechanisms Support Abstract Reasoning in Large Language Models. ICML 2025. arXiv:2502.20332. doi:10.48550/arXiv.2502.20332.
- Yasunaga, M., Chen, X., Li, Y., Pasupat, P., Leskovec, J., Liang, P., Chi, E. H. & Zhou, D. (2024). Large Language Models as Analogical Reasoners. ICLR 2024. arXiv:2310.01714. doi:10.48550/arXiv.2310.01714.
- Zhang, X., Cao, J. & You, C. (2024). Counting Ability of Large Language Models and Impact of Tokenization. arXiv preprint. arXiv:2410.19730. doi:10.48550/arXiv.2410.19730.

© 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 arXiv citation was machine-verified against its live arXiv Atom API record, and every other citation against its Crossref or DOI record, during drafting (title and author list checked against the record returned). The three rounds of the exchange were read in full: the arXiv HTML renderings of Webb, Holyoak and Lu (2023), Hodel and West (2023), Lewis and Mitchell (2024a, 2024b) and Webb, Holyoak and Lu (2024), and the Europe PMC full text of the published PNAS Nexus version (Webb, Holyoak and Lu, 2025). Every quantitative claim is taken from the abstract, main text or a table of the source credited with it. Two numbers (GPT-4 with code execution solving 40 of 60 and 30 of 60 problems) were computed by the author by summing the error counts in Tables 1 and 2 of Webb et al. (2024); the text says so where they appear. No experiment was run and no number in this paper was measured by its author. Table 1 represents published numbers, each named on its row; Figure 1 re-plots published values with no transformation. The three-unit account, the production/recognition split, Algorithm 1 and the proposed tests in Section 12 are conceptual synthesis by the author, not empirical results, and are presented as such. The supplementary material of the PNAS Nexus version was not read.

DOI: 10.5281/zenodo.23049431