وهم التفكير:

فهم نقاط القوة والقيود في نماذج الاستدلال من منظور تعقيد المشكلة

بارشين شجاعي*† إيمان ميرزاده*

كيوان عليزاده ماكسويل هورتون

سامي بنجيو مهرداد فرجتبار

Apple

الملخص

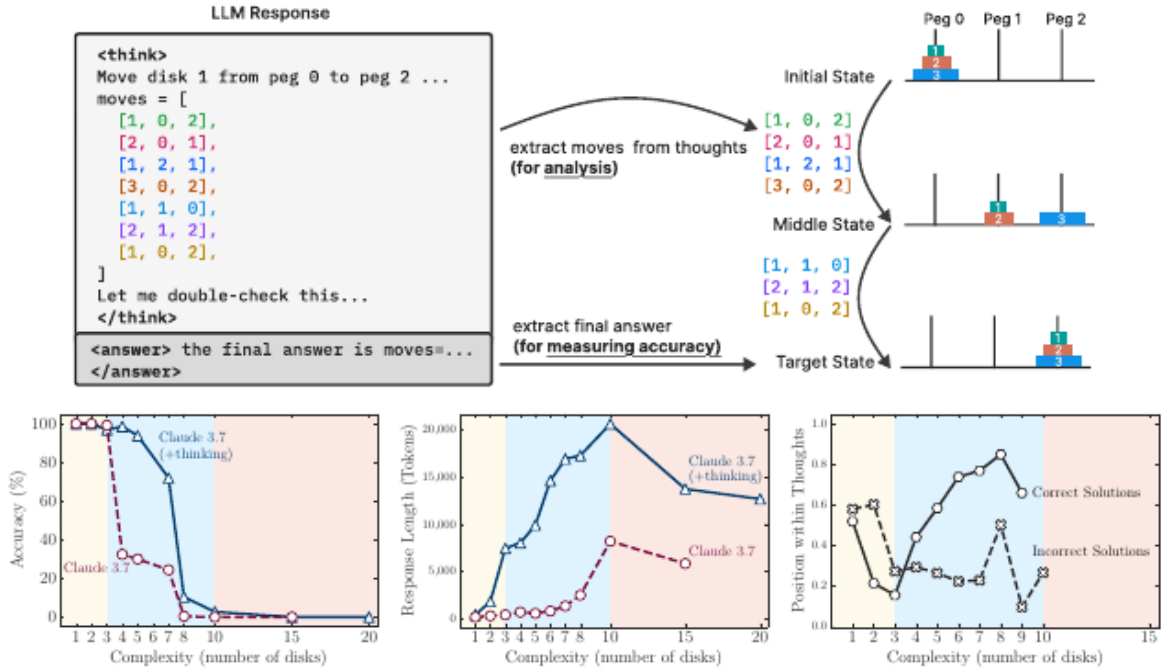
قدمت الأجيال الحديثة من نماذج اللغة الرائدة نماذج استدلال كبيرة (LRMs) تولّد عمليات تفكير مفصلة قبل تقديم الإجابات. وفي حين تُظهر هذه النماذج أداءً مُحسَّنًا في معايير الاستدلال القياسية، فإن قدراتها الأساسية، وخصائص التوسع، وقيودها ما تزال غير مفهومة بشكل كافٍ. تركز التقييمات الحالية في المقام الأول على معايير الرياضيات والبرمجة القياسية الراسخة، مع التأكيد على دقة الإجابة النهائية. ومع ذلك، غالبًا ما يعاني نموذج التقييم هذا من تلوث البيانات ولا يقدم رؤى حول بنية وجودة مسارات الاستدلال. في هذا العمل، نستقصي بشكل منهجي هذه الثغرات بمساعدة بيئات ألغاز قابلة للتحكم تسمح بالمعالجة الدقيقة للتعقيد التركيبي مع الحفاظ على بنى منطقية متسقة. يتيح هذا الإعداد تحليل الإجابات النهائية ليس فقط، بل وأيضًا مسارات الاستدلال الداخلية، مما يوفر رؤى حول كيفية "تفكير" نماذج الاستدلال الكبيرة (LRMs). من خلال تجارب مكثفة عبر ألغاز متنوعة، نُظهر أن نماذج الاستدلال الكبيرة (LRMs) الرائدة تواجه انهيارًا كاملًا للدقة بعد تجاوز مستويات تعقيد معينة. علاوة على ذلك، فإنها تُظهر حد توسع غير بديهي: يزداد جهد الاستدلال الخاص بها مع تعقيد المشكلة حتى نقطة معينة، ثم يتراجع على الرغم من امتلاكها ميزانية رموز كافية. بمقارنة نماذج الاستدلال الكبيرة (LRMs) بنظائرها من نماذج اللغة الكبيرة (LLMs) القياسية تحت حوسبة استدلالية مكافئة، نحدد ثلاثة أنظمة أداء: (1) المهام منخفضة التعقيد حيث تتفوق النماذج القياسية بشكل مفاجئ على نماذج الاستدلال الكبيرة (LRMs)، (2) المهام متوسطة التعقيد حيث يُظهر التفكير الإضافي في نماذج الاستدلال الكبيرة (LRMs) ميزة، و (3) المهام عالية التعقيد حيث يواجه كلا النموذجين انهيارًا كاملًا. وجدنا أن نماذج الاستدلال الكبيرة (LRMs) لديها قيود في الحساب الدقيق: فهي تفشل في استخدام الخوارزميات الصريحة وتستدل بشكل غير متسق عبر الألغاز. كما نقوم باستقصاء مسارات الاستدلال بعمق أكبر، ودراسة أنماط الحلول المستكشفة وتحليل السلوك الحسابي للنماذج، مما يلقي الضوء على نقاط قوتها، وقيودها، ويثير في نهاية المطاف أسئلة جوهرية حول قدرات الاستدلال الحقيقية لديها.

1. المقدمة

تطورت نماذج اللغة الكبيرة (LLMs) مؤخرًا لتشمل متغيرات متخصصة مُصممة خصيصًا لمهام الاستدلال – وهي نماذج الاستدلال الكبيرة (LRMs) مثل نماذج o1/o3 من شركة OpenAI [1, 2]، ونموذج DeepSeek-R1 [3]، ونموذج Claude 3.7 Sonnet Thinking [4]، ونموذج Gemini Thinking [5].

تُعد هذه النماذج مُنتجات جديدة، تتميز بآليات 'التفكير' الخاصة بها مثل سلسلة الأفكار (CoT) الطويلة مع التأمل الذاتي، وقد أظهرت نتائج واعدة عبر مختلف معايير الاستدلال القياسية.

* مساهمة متساوية.
أُنجز العمل خلال فترة تدريب في شركة Apple.
apple.com@{p_shojaee, imirzadeh, kalizadehvahid, mchorton, bengio, farajtabar}



الشكل 1: أعلى: يتيح إعدادنا التحقق من كل من الإجابات النهائية ومسارات الاستدلال الوسيطة، مما يسمح بتحليل مفصل لسلوك تفكير النموذج. أسفل اليسار والوسط: عند التعقيد المنخفض، تكون النماذج غير المفكرة أكثر دقة وكفاءة في استخدام الرموز. مع ازدياد التعقيد، تتفوق نماذج الاستدلال في الأداء ولكنها تتطلب المزيد من الرموز – إلى أن ينهار كلاهما بعد تجاوز عتبة حرجة، مع مسارات أقصر. أسفل اليمين: بالنسبة للحالات التي تم حلها بشكل صحيح، يميل نموذج Claude 3.7 Thinking إلى إيجاد الإجابات مبكرًا عند التعقيد المنخفض ولاحقًا عند التعقيد الأعلى. في الحالات الفاشلة، غالبًا ما يثبت على إجابة خاطئة مبكرة، مما يهدر ميزانية الرموز المتبقية. تكشف كلتا الحالتين عن أوجه قصور في عملية الاستدلال.

يشير هذا الظهور إلى تحول نموذجي محتمل في كيفية تعامل أنظمة نماذج اللغة الكبيرة (LLM) مع مهام الاستدلال وحل المشكلات المعقدة، حيث يقترح بعض الباحثين أنها خطوات هامة نحو قدرات ذكاء اصطناعي عام أكثر شمولاً.

على الرغم من هذه الادعاءات والتطورات في الأداء، فإن الفوائد والقيود الأساسية لنماذج الاستدلال الكبيرة (LRMs) ما تزال غير مفهومة بشكل كافٍ. لا تزال هناك أسئلة جوهرية قائمة: هل هذه النماذج قادرة على استدلال قابل للتعميم، أم أنها تستفيد من أشكال مختلفة من مطابقة الأنماط [6]؟ كيف يتوسع أداؤها مع زيادة تعقيد المشكلة؟ كيف تُقارن بنظائرها من نماذج اللغة الكبيرة (LLM) القياسية غير المفكرة عند تزويدها بنفس حوسبة رموز الاستدلال؟ والأهم من ذلك، ما هي القيود المتأصلة في مناهج الاستدلال الحالية، وما التحسينات التي قد تكون ضرورية للتقدم نحو قدرات استدلال أكثر قوة؟

نعتقد أن الافتقار إلى التحليلات المنهجية التي تستقصي هذه الأسئلة يرجع إلى القيود في نماذج التقييم الحالية. تركز التقييمات الحالية في الغالب على معايير الرياضيات والبرمجة القياسية الراسخة، والتي، على الرغم من قيمتها، غالبًا ما تعاني من مشكلات تلوث البيانات ولا تسمح بظروف تجريبية محكومة عبر إعدادات ومستويات تعقيد مختلفة. علاوة على ذلك، لا تقدم هذه التقييمات رؤى حول بنية وجودة مسارات الاستدلال. لفهم سلوك الاستدلال لهذه النماذج بشكل أكثر صرامة، نحتاج إلى بيئات تتيح التجريب المحكوم.

في هذه الدراسة، نستقصي آليات الاستدلال لنماذج الاستدلال الكبيرة (LRMs) الرائدة من منظور تعقيد المشكلة. فبدلاً من المعايير القياسية (مثل مسائل الرياضيات)، نعتمد بيئات ألغاز قابلة للتحكم تتيح لنا تغيير التعقيد بشكل منهجي – عن طريق تعديل عناصر اللغز مع الحفاظ على المنطق الأساسي – وفحص كل من الحلول والاستدلال الداخلي (الشكل 1، أعلى). هذه الألغاز: (1) توفر تحكمًا دقيقًا في التعقيد؛ (2) تتجنب التلوث الشائع في المعايير القياسية الراسخة؛ (3) تتطلب فقط القواعد المقدمة بشكل صريح، مع التأكيد على الاستدلال الخوارزمي؛ و (4) تدعم التقييم الصارم القائم على المحاكاة، مما يتيح عمليات فحص دقيقة للحلول وتحليلات مفصلة للفشل.
يكشف استقصاؤنا التجريبي عن عدة نتائج رئيسية حول نماذج الاستدلال الكبيرة (LRMs) الحالية: أولاً، على الرغم من آليات التأمل الذاتي المتطورة التي تعلمتها من خلال التعلم المعزز (Reinforcement Learning)، تفشل هذه النماذج في تطوير قدرات حل مشكلات قابلة للتعميم لمهام التخطيط، مع انهيار الأداء إلى الصفر بعد تجاوز عتبة تعقيد معينة. ثانيًا، تكشف مقارنتنا بين نماذج الاستدلال الكبيرة (LRMs) ونماذج اللغة الكبيرة (LLMs) القياسية تحت حوسبة استدلالية مكافئة عن ثلاثة أنظمة استدلال مميزة (الشكل 1، أسفل).
بالنسبة للمشكلات الأبسط منخفضة التركيب، تُظهر نماذج اللغة الكبيرة (LLMs) القياسية كفاءة ودقة أكبر. مع ازدياد تعقيد المشكلة بشكل معتدل، تكتسب نماذج الاستدلال ميزة. ومع ذلك، عندما تصل المشكلات إلى تعقيد عالٍ بعمق تركيبي أطول، يواجه كلا نوعي النماذج انهيارًا كاملًا في الأداء (الشكل 1، أسفل اليسار).
والجدير بالذكر، بالقرب من نقطة الانهيار هذه، تبدأ نماذج الاستدلال الكبيرة (LRMs) في تقليل جهد الاستدلال الخاص بها (مقاسًا برموز وقت الاستدلال) مع ازدياد تعقيد المشكلة، على الرغم من أنها تعمل أقل بكثير من حدود طول التوليد (الشكل 1، أسفل الوسط). يشير هذا إلى وجود قيود أساسية في توسع وقت الاستدلال في قدرات الاستدلال لنماذج الاستدلال الكبيرة (LRMs) بالنسبة لتعقيد المشكلة. أخيرًا، يكشف تحليلنا لمسارات الاستدلال الوسيطة أو الأفكار عن أنماط تعتمد على التعقيد: في المشكلات الأبسط، غالبًا ما تحدد نماذج الاستدلال الحلول الصحيحة مبكرًا ولكنها تستمر بشكل غير فعال في استكشاف بدائل غير صحيحة – وهي ظاهرة 'التفكير المفرط' (Overthinking). عند التعقيد المعتدل، لا تظهر الحلول الصحيحة إلا بعد استكشاف مكثف للمسارات غير الصحيحة. بعد تجاوز عتبة تعقيد معينة، تفشل النماذج تمامًا في إيجاد الحلول الصحيحة (الشكل 1، أسفل اليمين). يشير هذا إلى أن نماذج الاستدلال الكبيرة (LRMs) تمتلك قدرات محدودة على التصحيح الذاتي، والتي، على الرغم من قيمتها، تكشف عن أوجه قصور أساسية وقيود توسع واضحة.
تسلط هذه النتائج الضوء على كل من نقاط القوة والقيود في نماذج الاستدلال الكبيرة (LRMs) الحالية، مما يثير تساؤلات حول طبيعة الاستدلال في هذه الأنظمة مع آثار هامة على تصميمها ونشرها. مساهماتنا الرئيسية هي:
نشكك في نموذج التقييم الحالي لنماذج الاستدلال الكبيرة (LRMs) على معايير الرياضيات القياسية الراسخة ونصمم منصة اختبار تجريبية محكومة من خلال الاستفادة من بيئات الألغاز الخوارزمية التي تتيح التجريب المحكوم فيما يتعلق بتعقيد المشكلة.

نُظهر أن أحدث نماذج الاستدلال الكبيرة (LRMs) (مثل o3-mini، و DeepSeek-R1، و Claude-3.7-Sonnet-Thinking) ما تزال تفشل في تطوير قدرات حل مشكلات قابلة للتعميم، مع انهيار الدقة في نهاية المطاف إلى الصفر بعد تجاوز مستويات تعقيد معينة عبر بيئات مختلفة.
نجد أنه يوجد حد للتوسع في جهد الاستدلال لنماذج الاستدلال الكبيرة (LRMs) فيما يتعلق بتعقيد المشكلة، ويتضح ذلك من خلال الاتجاه التنازلي غير البديهي في رموز التفكير بعد نقطة تعقيد معينة.
نشكك في نموذج التقييم الحالي القائم على دقة الإجابة النهائية ونوسع تقييمنا ليشمل الحلول الوسيطة لمسارات التفكير بمساعدة محاكيات ألغاز حتمية. يكشف تحليلنا أنه مع ازدياد تعقيد المشكلة، تظهر الحلول الصحيحة بشكل منهجي في مواضع لاحقة في التفكير مقارنة بالحلول غير الصحيحة، مما يوفر رؤى كمية حول آليات التصحيح الذاتي داخل نماذج الاستدلال الكبيرة (LRMs).
نكشف عن قيود مفاجئة في قدرة نماذج الاستدلال الكبيرة (LRMs) على إجراء الحساب الدقيق، بما في ذلك فشلها في الاستفادة من الخوارزميات الصريحة واستدلالها غير المتسق عبر أنواع الألغاز المختلفة.

2. الأعمال ذات الصلة

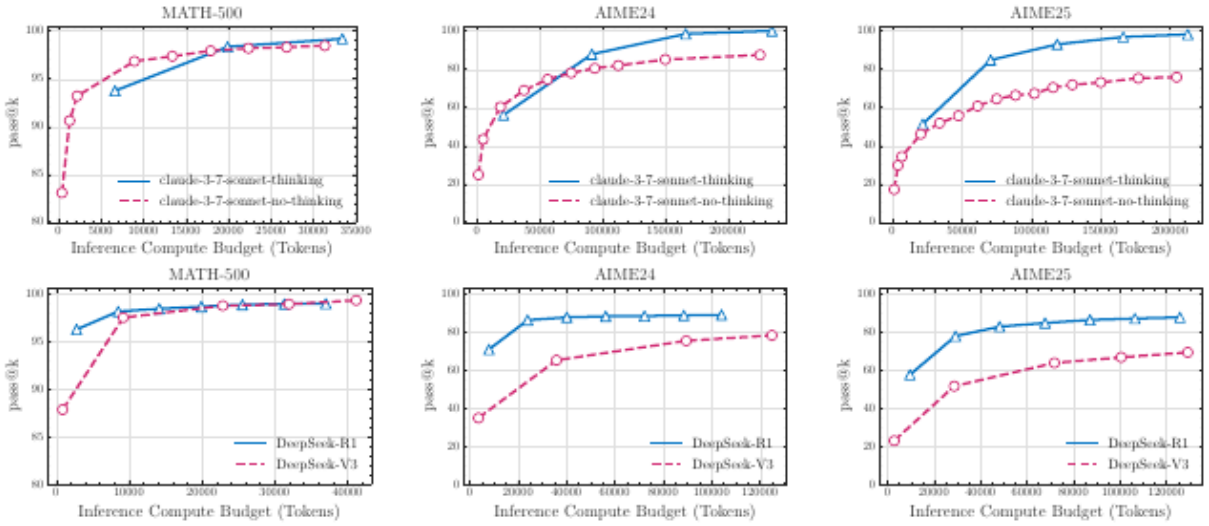
الاستدلال في نماذج اللغة. تخضع نماذج اللغة الكبيرة (LLMs) لمراحل تدريب متعددة ومكلفة باستخدام كميات هائلة من بيانات التدريب. وفي حين تُظهر نماذج اللغة الكبيرة (LLMs) هذه فهمًا واعدًا للغة مع قدرات ضغط قوية، فإن قدراتها على الذكاء والاستدلال ما تزال موضوع نقاش علمي حاسم [7، 8]. أظهرت التكرارات السابقة لنماذج اللغة الكبيرة (LLMs) [9، 10، 11] أداءً ضعيفًا في معايير الاستدلال القياسية [12، 13، 14، 6]. لمعالجة أوجه القصور هذه، تم استكشاف عدة مناهج يتمثل الموضوع المشترك بينها في "توسيع نطاق" كل من بيانات التدريب والحوسبة في وقت الاختبار. على سبيل المثال، أُثبت أن توليد سلسلة الأفكار (CoT) [15، 16، 17، 18] ودمج التحقق الذاتي [19، 20، 21] قبل الإجابة النهائية يُحسّن أداء النموذج. ومع ذلك، فإن الحصول على بيانات سلسلة الأفكار (CoT) عالية الجودة وقابلة للتوسع أمر مكلف للغاية بسبب ندرتها. يركز خط بحثي آخر على تعويض الافتقار إلى البيانات الموجهة عن طريق تعليم النماذج التفكير بفعالية أكبر من خلال التعلم الموجه أو التعلم المعزز (RL) [22، 23، 24، 25، 26، 27]. ومن الأمثلة البارزة مفتوحة المصدر لهذه التحسينات نموذج Deepseek-R1 [3]، الذي أظهر أن تطبيق التعلم المعزز (RL) مع مكافآت قابلة للتحقق يمكن أن يعزز أداء النموذج بشكل كبير، مما يضاهي أداء النماذج المغلقة مثل نموذج o1 من OpenAI [2]، مما أدى إلى جيل جديد من نماذج اللغة يشار إليها باسم نماذج الاستدلال الكبيرة (LRMs) مثل Gemini flash thinking [5]، و Claude 3.7 Sonnet thinking [4]، إلخ.

فهم نماذج الاستدلال الكبيرة. استكشفت الدراسات الحديثة جوانب مختلفة من سلوك الاستدلال: فقد أظهرت نماذج الاستدلال الكبيرة سلوكيات ناشئة مثل التباين بين مسارات التفكير والإجابات النهائية [28، 29] بالإضافة إلى مخاوف تتعلق بالكفاءة من خلال ما يسميه الباحثون "ظاهرة 'التفكير المفرط'" [30، 31، 32، 33]، حيث تنتج النماذج مخرجات مسهبة ومكررة، حتى بعد إيجاد الحل، مما يخلق عبء حوسبة استدلال إضافي كبير. في هذا العمل، نحلل بشكل منهجي مدى تفكير النموذج بالنسبة لتعقيد المهمة. حديثًا، أظهر بالون وآخرون [34] أنه في نماذج الاستدلال الكبيرة (LRMs) الأحدث، تنخفض الدقة بشكل عام عندما يزداد التفكير في مسائل الرياضيات، في المقابل، نلاحظ أنه عندما يتجاوز مستوى الصعوبة حدًا معينًا في بيئة الألغاز القابلة للتحكم، يبدأ النموذج في التفكير بشكل أقل، وأن الارتباط العكسي بين التفكير وتعقيد المهمة يحدث فقط حتى عتبة معينة. تساءل يو وآخرون [35] عما إذا كان التعلم المعزز (RL) يستحث حقًا أنماط استدلال جديدة ويُظهرون أن مقياس pass@k لنماذج الاستدلال مقابل النماذج غير الاستدلالية يتقارب إلى

نفس النقطة. نلاحظ أيضًا أنه في معيار MATH-500، يكون مقياس pass@k متقاربًا لنماذج الاستدلال مقابل النماذج غير الاستدلالية، لكننا لاحظنا أنماطًا مختلفة تحت مستويات التعقيد المتوسطة والعالية للألغاز، وهو ما لا يمكن ملاحظته بسهولة في معايير الرياضيات القياسية الراسخة المستخدمة في التقييمات الشائعة.

بيئات التقييم القابلة للتحكم. على عكس الدراسات السابقة التي ركزت على المسائل الرياضية لتقييم قدرات الاستدلال لنماذج اللغة، يقدم هذا العمل بيئات ألغاز قابلة للتحكم. تسمح هذه البيئات بالمعالجة الدقيقة لتعقيد المشكلة مع الحفاظ على عمليات منطقية متسقة، مما يتيح تحليلًا أكثر صرامة لأنماط الاستدلال وقيوده. إن البيئات القابلة للتحكم ليست غير شائعة في الأدبيات [12، 36، 37]. ومع ذلك، فإن هدفنا الأساسي ليس اقتراح معيار قياسي جديد؛ بدلاً من ذلك، نستخدم هذه المعايير كأدوات لتصميم التجارب لفهم قدرات الاستدلال لنماذج اللغة. أظهرت دراسة وثيقة الصلة أجراها فالميكام وآخرون [38] أن نماذج o1 تُظهر تحسينات كبيرة في الأداء مقارنة بالنماذج السابقة. يقدم عملنا رؤى إضافية، مثل فحص أزواج من النماذج المفكرة/غير المفكرة (على سبيل المثال، DeepSeek-R1/V3، و Claude 3.7 Sonnet (thinking/non-thinking). علاوة على ذلك، ندرس مسارات الاستدلال لنماذج الاستدلال الكبيرة (LRMs) بعمق أكبر، مما يكشف عن سلوكيات مختلفة عبر مستويات تعقيد متنوعة.

بشكل عام، تثير النتائج الواعدة من نماذج الاستدلال الكبيرة (LRMs) الحديثة سؤالًا حاسمًا: إلى أي مدى تم تحسين القيود المبلغ عنها سابقًا لنماذج اللغة الكبيرة (LLMs)؟ في هذا العمل، نتجاوز مجرد قياس أداء نماذج الاستدلال الكبيرة (LRMs) هذه. نحن نحلل مدى جودة تعامل نماذج الاستدلال الكبيرة (LRMs) هذه مع المشكلات ذات التعقيدات المتفاوتة وندرس خصائص عمليات الاستدلال الخاصة بها.



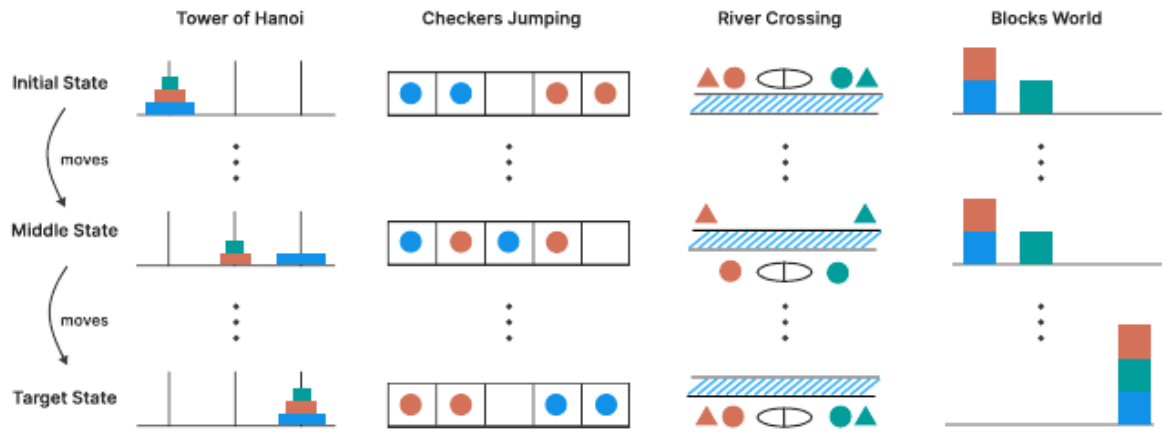
الشكل 2: يكشف التحليل المقارن للنماذج المفكرة مقابل النماذج غير المفكرة عبر معايير الرياضيات القياسية عن أنماط أداء غير متسقة. فبينما تُظهر النتائج على مجموعة بيانات MATH-500 أداءً متقاربًا بين كلا نوعي النماذج، تُظهر النماذج المفكرة أداءً متفوقًا في معياري AIME24 و AIME25. بالإضافة إلى ذلك،

يُبرز تدهور الأداء الملحوظ من معيار AIME24 إلى معيار AIME25 مدى قابلية هذه المعايير لمشكلات تلوث البيانات.

3. بيئات الرياضيات والألغاز

حاليًا، ليس من الواضح ما إذا كانت تحسينات الأداء الملحوظة في نماذج التفكير الحديثة القائمة على التعلم المعزز (RL) تُعزى إلى زيادة التعرض لبيانات معايير الرياضيات القياسية الراسخة، أو إلى الحوسبة الاستدلالية الأكبر بكثير المخصصة لرموز التفكير، أو إلى قدرات الاستدلال التي طُورت بواسطة التدريب القائم على التعلم المعزز (RL). استكشفت دراسات حديثة [35، 39] هذا السؤال باستخدام معايير الرياضيات القياسية الراسخة من خلال مقارنة القدرات الحديّة العليا (pass@k) لنماذج التفكير القائمة على التعلم المعزز (RL) بنظائرها من نماذج اللغة الكبيرة (LLM) القياسية غير المفكرة. وقد أظهروا أنه في ظل ميزانيات رموز استدلال مكافئة، يمكن لنماذج اللغة الكبيرة (LLMs) غير المفكرة أن تصل في نهاية المطاف إلى أداء مماثل لنماذج التفكير في معايير قياسية مثل مجموعة بيانات MATH500 [40] ومعيار AIME24 [41]. أجرينا أيضًا تحليلنا المقارن لنماذج الاستدلال الكبيرة (LRMs) الرائدة مثل نموذج Claude-3.7-Sonnet (مع وبدون تفكير) ونموذج DeepSeek R1 (مقابل V3). تؤكد نتائجنا (الموضحة في الشكل 2) أنه، على مجموعة بيانات MATH500، يكون أداء pass@k لنماذج التفكير مماثلاً لنظائرها غير المفكرة عند تزويدها بنفس ميزانية رموز الاستدلال. ومع ذلك، لاحظنا أن فجوة الأداء هذه تتسع في معيار AIME24 وتتسع أكثر في معيار AIME25.

تمثل هذه الفجوة المتسعة تحديًا تفسيريًا. يمكن أن يُعزى ذلك إما إلى: (1) زيادة التعقيد الذي يتطلب عمليات استدلال أكثر تطورًا، مما يكشف عن مزايا حقيقية لنماذج التفكير للمشكلات الأكثر تعقيدًا، أو (2) انخفاض تلوث البيانات في المعايير الأحدث (خاصة معيار AIME25). ومن المثير للاهتمام أن الأداء البشري في معيار AIME25 كان في الواقع أعلى منه في معيار AIME24 [42، 43]، مما يشير إلى أن معيار AIME25 قد يكون أقل تعقيدًا. ومع ذلك، فإن أداء النماذج أسوأ في معيار AIME25 مقارنة بمعيار AIME24 – مما قد يشير إلى تلوث البيانات أثناء تدريب نماذج الاستدلال الكبيرة (LRMs) الرائدة. بالنظر إلى هذه الملاحظات غير المبررة وحقيقة أن معايير الرياضيات القياسية لا تسمح بالمعالجة المحكومة لتعقيد المشكلة، فقد لجأنا إلى بيئات الألغاز التي تتيح تجريبًا أكثر دقة ومنهجية.



الشكل 3: توضيح لبيئات الألغاز الأربع. تُظهر الأعمدة التقدم من الحالة الأولية (أعلى) مرورًا بالحالة الوسيطة (وسط) إلى الحالة المستهدفة (أسفل) للألغاز: برج هانوي (نقل الأقراص عبر الأوتاد)، قفز الداما (تبديل مواضع الرموز الملونة)، عبور النهر (نقل الكيانات عبر النهر)، وعالم المكعبات (إعادة تشكيل تراص المكعبات).

3-1 بيئات الألغاز

نُقيّم استدلال نماذج الاستدلال الكبيرة (LRM) على أربعة ألغاز قابلة للتحكم تشمل العمق التركيبي، وتعقيد التخطيط، والإعدادات التوزيعية. الألغاز مُعرفة أدناه وموضحة في الشكل 3.

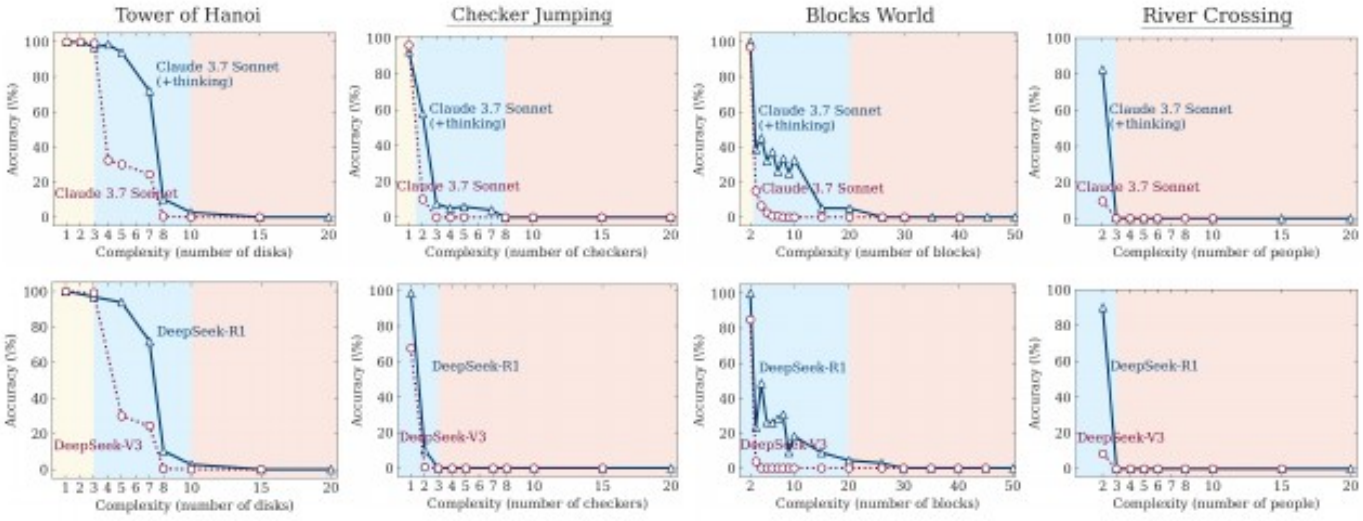
برج هانوي (Tower of Hanoi) هو لغز يتكون من ثلاثة أوتاد وعدد n من الأقراص مختلفة الأحجام مكدسة على الوتد الأول بترتيب الحجم (الأكبر في الأسفل). الهدف هو نقل جميع الأقراص من الوتد الأول إلى الوتد الثالث. تتضمن الحركات الصالحة تحريك قرص واحد فقط في كل مرة، وأخذ القرص العلوي فقط من الوتد، وعدم وضع قرص أكبر فوق قرص أصغر منه أبدًا. يمكن التحكم في صعوبة هذه المهمة من خلال عدد الأقراص الأولية حيث أن الحد الأدنى لعدد الحركات المطلوبة مع وجود n من الأقراص الأولية سيكون 2ⁿ – 1. ومع ذلك، في هذا العمل لا نقوم بتقييم مثالية الحل النهائي ونقيس فقط صحة كل حركة والوصول إلى الحالة المستهدفة.

قفز الداما (Checker Jumping) هو لغز أحادي البعد يرتب قطع داما حمراء، وقطع داما زرقاء، ومساحة فارغة واحدة في خط. الهدف هو تبديل مواضع جميع قطع الداما الحمراء والزرقاء، مما يعكس بشكل فعال التكوين الأولي. تتضمن الحركات الصالحة تحريك قطعة داما إلى مساحة فارغة مجاورة أو القفز فوق قطعة داما واحدة بالضبط من اللون المقابل للهبوط في مساحة فارغة. لا يمكن لأي قطعة داما التحرك للخلف في عملية اللغز. يمكن التحكم في تعقيد هذه المهمة من خلال عدد قطع الداما: مع وجود 2n من قطع الداما، سيكون الحد الأدنى لعدد الحركات المطلوبة هو (n + 1)² – 1.

عبور النهر (River Crossing) هو لغز تخطيط لإرضاء القيود يتضمن عدد n من الفاعلين وعدد n من الوكلاء المقابلين لهم الذين يجب عليهم عبور النهر باستخدام قارب. الهدف هو نقل جميع الأفراد البالغ عددهم

2n من الضفة اليسرى إلى الضفة اليمنى. يمكن للقارب أن يحمل k من الأفراد على الأكثر ولا يمكنه السفر فارغًا. تنشأ مواقف غير صالحة عندما يكون الفاعل في وجود وكيل آخر دون وجود وكيله الخاص، حيث يجب على كل وكيل حماية عميله من الوكلاء المنافسين. يمكن أيضًا التحكم في تعقيد هذه المهمة من خلال عدد أزواج الفاعل/الوكيل الموجودة. بالنسبة لـ n = 2، n = 3 أزواج، نستخدم سعة قارب k = 2، وبالنسبة لعدد أكبر من الأزواج نستخدم k = 3.

عالم المكعبات (Blocks World) هو لغز تكديس مكعبات يتطلب إعادة ترتيب المكعبات من تكوين أولي إلى تكوين هدف محدد. الهدف هو إيجاد الحد الأدنى لعدد الحركات اللازمة لهذا التحويل. تقتصر الحركات الصالحة على المكعب العلوي لأي كومة، والذي يمكن وضعه إما على كومة فارغة أو فوق مكعب آخر. يمكن



التحكم في التعقيد في هذه المهمة من خلال عدد المكعبات الموجودة.

الشكل 4: دقة النماذج المفكرة (Claude 3.7 Sonnet مع التفكير، DeepSeek-R1) مقابل نظائرها غير المفكرة (Claude 3.7 Sonnet، DeepSeek-V3) عبر جميع بيئات الألغاز ومستويات متفاوتة من تعقيد المشكلة.

4. التجارب والنتائج

4.1 إعداد التجارب

تُجرى معظم تجاربنا على نماذج الاستدلال ونظائرها غير المفكرة، مثل Claude 3.7 Sonnet (مفكر/غير مفكر) و DeepSeek-R1/V3. اخترنا هذه النماذج لأنها تتيح الوصول إلى رموز التفكير، على عكس نماذج مثل سلسلة-o من OpenAI. بالنسبة للتجارب التي تركز فقط على الدقة النهائية، نُبلغ أيضًا عن النتائج على نماذج سلسلة-o. بالنسبة لنماذج Claude 3.7 Sonnet، نسمح بأقصى ميزانية للرموز (64 ألف). وبالمثل، بالنسبة لنماذج DeepSeek-R1/V3 على الخوادم المحلية، نسمح بأن يصل الحد الأقصى للطول إلى 64

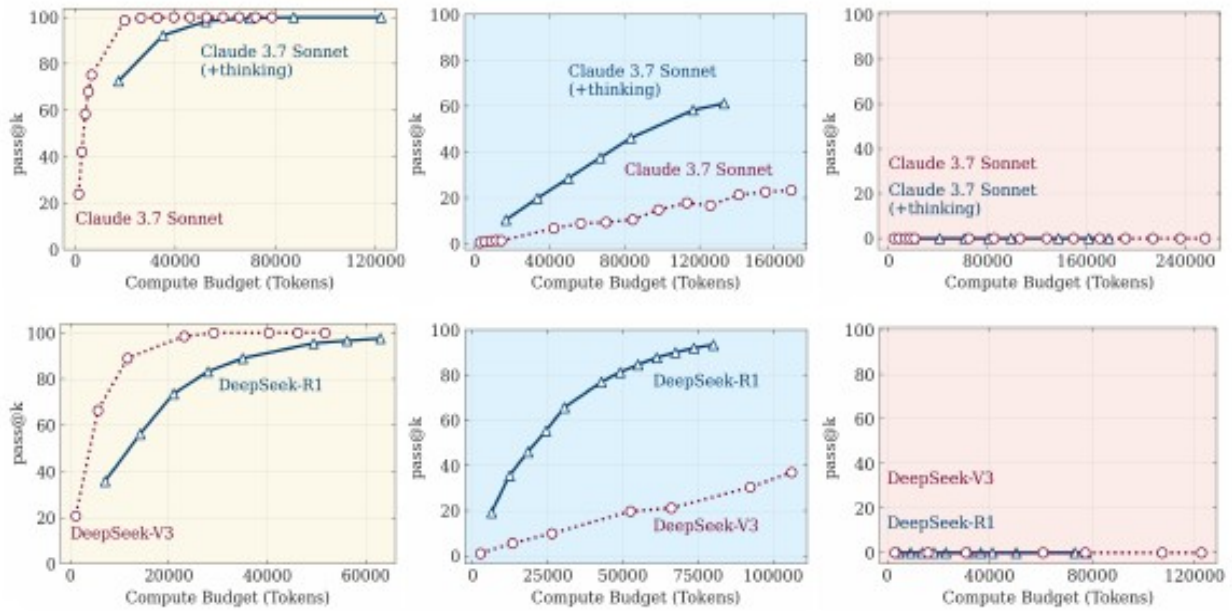
ألف رمز. لكل حالة لغز، نولّد 25 عينة ونُبلغ عن متوسط أداء كل نموذج عبرها. يتم توفير تفاصيل شاملة لإعداد تجاربنا ونتائجنا في الملحق.

4.2 كيف يؤثر التعقيد على الاستدلال؟

4.2.1 ثلاثة أنظمة للتعقيد

بدافع من الملاحظات الواردة في الشكل 2، وللاستقصاء المنهجي لتأثير تعقيد المشكلة على سلوك الاستدلال، أجرينا تجارب تقارن بين أزواج النماذج المفكرة وغير المفكرة عبر بيئات الألغاز القابلة للتحكم الخاصة بنا. ركز تحليلنا على أزواج متطابقة من نماذج اللغة الكبيرة (LLMs) ذات الهياكل الأساسية المتطابقة للنماذج، وتحديدًا Claude-3.7-Sonnet (مع مقابل بدون تفكير) و DeepSeek (R1 مقابل V3). في كل لغز، نغير التعقيد عن طريق معالجة حجم المشكلة N (يمثل عدد الأقراص، أو عدد قطع الداما، أو عدد المكعبات، أو عناصر العبور).

يعرض الشكل 4 دقة كلا نوعي النماذج كدالة لتعقيد المشكلة عبر جميع بيئات الألغاز. واستكمالًا لذلك، يُظهر الشكل 5 قدرات الأداء الحديّة العليا (pass@k) لأزواج النماذج هذه تحت حوسبة رموز استدلال مكافئة (متوسطة عبر جميع الألغاز)، مما يوسع التحليلات السابقة من معايير الرياضيات القياسية (الشكل 2) لتشمل بيئات الألغاز القابلة للتحكم. تُظهر النتائج من كلا هذين الشكلين أنه، على عكس الملاحظات من الرياضيات، توجد ثلاثة أنظمة في سلوك هذه النماذج فيما يتعلق بالتعقيد. في النظام الأول حيث يكون تعقيد المشكلة منخفضًا، نلاحظ أن النماذج غير المفكرة قادرة على الحصول على أداء مماثل، أو حتى أفضل من النماذج المفكرة مع استدلال أكثر كفاءة في استخدام الرموز. في



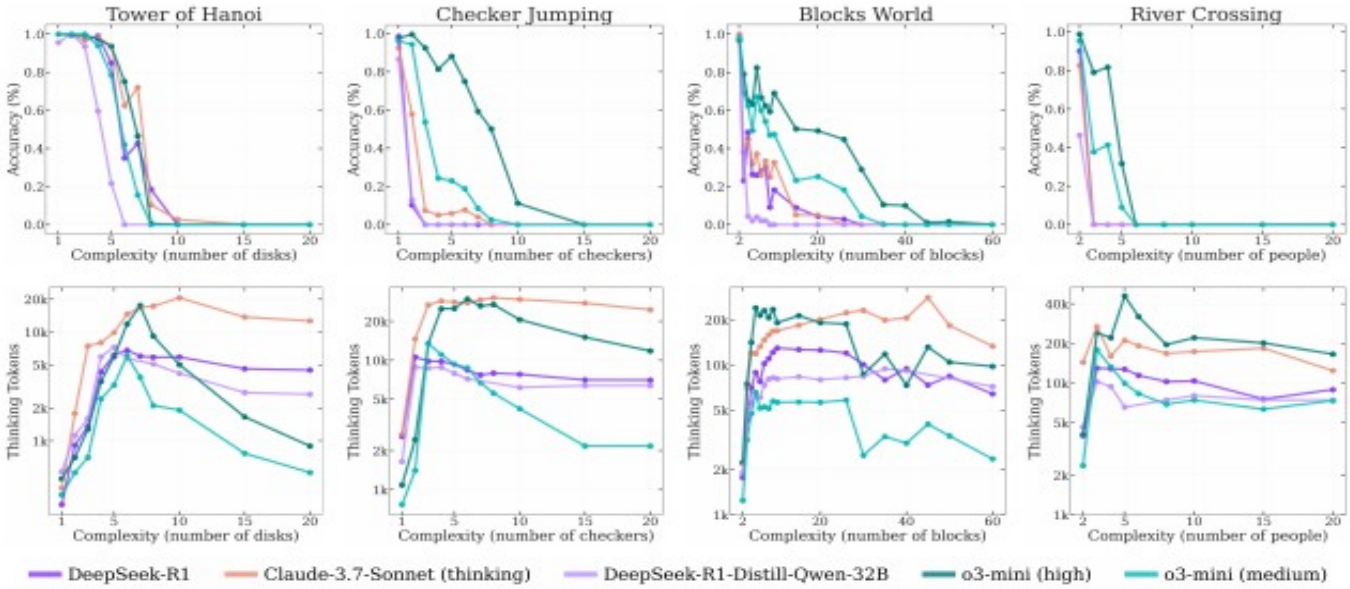
الشكل 5: أداء Pass@k للنماذج المفكرة مقابل النماذج غير المفكرة عبر ميزانيات حوسبة مكافئة في بيئات الألغاز ذات التعقيد المنخفض والمتوسط والعالي. تتفوق النماذج غير المفكرة في المشكلات البسيطة، وتُظهر النماذج المفكرة مزايا عند التعقيد المتوسط، بينما تفشل كلتا الطريقتين عند التعقيد العالي بغض النظر عن تخصيص الحوسبة.

النظام الثاني ذي التعقيد المتوسط، تبدأ ميزة نماذج الاستدلال القادرة على توليد سلسلة أفكار طويلة في الظهور، وتزداد فجوة الأداء بين أزواج النماذج. النظام الأكثر إثارة للاهتمام هو النظام الثالث حيث يكون تعقيد المشكلة أعلى وينهار أداء كلا النموذجين إلى الصفر. تُظهر النتائج أنه بينما تؤخر النماذج المفكرة هذا الانهيار، فإنها تواجه أيضًا في نهاية المطاف نفس القيود الأساسية التي تواجهها نظائرها غير المفكرة.

4.2.2 انهيار نماذج الاستدلال

بعد ذلك، نفحص كيف تستجيب نماذج الاستدلال المتخصصة المختلفة المجهزة برموز التفكير لزيادة تعقيد المشكلة. تُقيّم تجاربنا خمسة نماذج تفكير حديثة: o3-mini (تكوينات متوسطة وعالية)، و DeepSeek-R1، و DeepSeek-R1-Qwen-32B، و Claude-3.7-Sonnet (مفكر). يوضح الشكل 6 أداء هذه النماذج من حيث الدقة (أعلى) واستخدام رموز التفكير (أسفل) عبر مستويات تعقيد متفاوتة. تُظهر النتائج أن جميع نماذج الاستدلال تُظهر نمطًا مشابهًا فيما يتعلق بالتعقيد: تنخفض الدقة تدريجيًا مع زيادة تعقيد المشكلة حتى تصل إلى انهيار كامل (دقة صفرية) بعد تجاوز عتبة تعقيد خاصة بكل نموذج. يكشف تحليل حوسبة رموز التفكير الاستدلالية أيضًا عن نمط مثير للاهتمام في تخصيص رموز التفكير التي تعلمتها هذه النماذج. نلاحظ أن نماذج الاستدلال تزيد في البداية رموز التفكير الخاصة بها بشكل متناسب مع تعقيد المشكلة. ومع ذلك، عند الاقتراب من عتبة حرجة – والتي تتوافق بشكل وثيق مع نقطة انهيار دقتها – تبدأ النماذج بشكل غير بديهي في تقليل جهد الاستدلال الخاص بها على الرغم من زيادة صعوبة المشكلة. هذه الظاهرة أكثر وضوحًا في متغيرات o3-mini وأقل حدة في نموذج Claude-3.7-Sonnet (المفكر). والجدير بالذكر، على الرغم من

أنها تعمل أقل بكثير من حدود طول التوليد الخاصة بها مع توفر ميزانية استدلال وفيرة، تفشل هذه النماذج في الاستفادة من الحوسبة الاستدلالية الإضافية خلال مرحلة التفكير عندما تصبح المشكلات أكثر تعقيدًا. يشير هذا السلوك إلى وجود قيود أساسية في توسع قدرات التفكير لنماذج الاستدلال الحالية بالنسبة لتعقيد المشكلة.



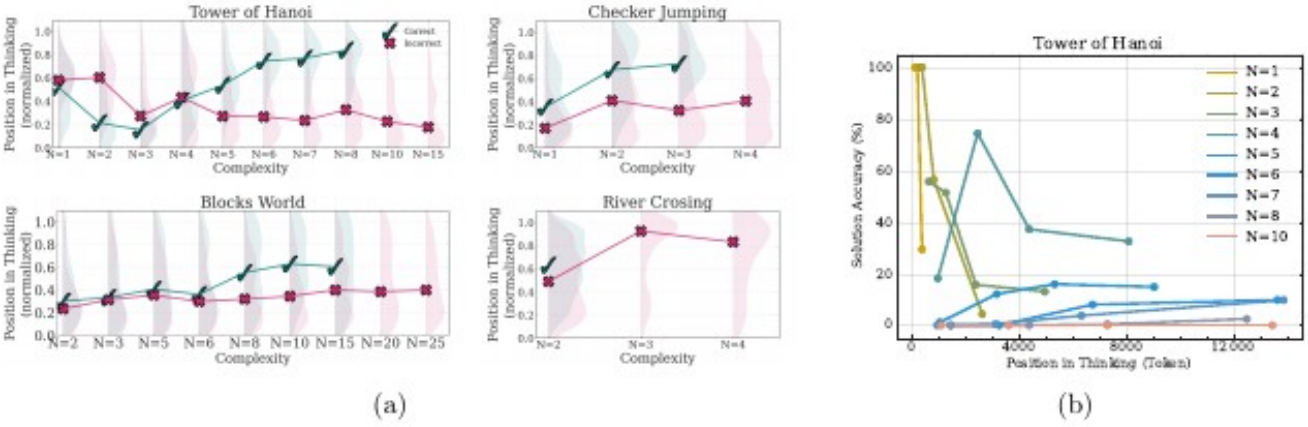
الشكل 6: الدقة ورموز التفكير مقابل تعقيد المشكلة لنماذج الاستدلال عبر بيئات الألغاز. مع زيادة التعقيد، تنفق نماذج الاستدلال في البداية المزيد من الرموز بينما تنخفض الدقة تدريجيًا، حتى تصل إلى نقطة حرجة ينهار عندها الاستدلال – فينخفض الأداء بشكل حاد ويتناقص جهد الاستدلال.

4.3 ماذا يحدث داخل أفكار نماذج الاستدلال؟

لاكتساب رؤى أعمق حول عمليات التفكير لنماذج الاستدلال، أجرينا تحليلًا دقيقًا لمسارات الاستدلال الخاصة بها. كما هو موضح في الشكل 1، يتيح لنا إعدادنا مع بيئات الألغاز النظر إلى ما هو أبعد من الإجابة النهائية والحصول على رؤية أكثر تفصيلاً لمسارات الاستدلال ('الأفكار') التي تنتجها هذه النماذج. نقوم باستخلاص وتحليل الحلول الوسيطة المستكشفة ضمن أفكار النموذج بمساعدة محاكيات الألغاز. يستقصي تحقيقنا أنماط وخصائص هذه الحلول الوسيطة، وصحتها بالنسبة لموضعها التسلسلي في عملية الاستدلال، وكيف تتطور هذه الأنماط مع زيادة تعقيد المشكلة. لهذا التحليل، نركز على مسارات الاستدلال التي يولدها نموذج Claude-3.7-Sonnet-Thinking عبر مجموعة الألغاز الخاصة بنا. لكل حل وسيط تم تحديده ضمن المسارات، سجلنا: (1) موضعه النسبي ضمن مسار الاستدلال (مُطَبَّعًا بطول التفكير الكلي)، (2) صحته كما تم التحقق منها بواسطة محاكيات الألغاز الخاصة بنا، و (3) تعقيد المشكلة المقابلة. يتيح ذلك توصيف تقدم ودقة تطوير الحل على امتداد عملية الاستدلال.

يوضح الشكل 7أ العلاقة بين موضع الحلول الوسيطة ضمن الأفكار، وصحتها، وتعقيد المشكلة عبر جميع بيئات الألغاز. يؤكد تحليلنا لمسارات الاستدلال أيضًا صحة الأنظمة الثلاثة للتعقيد التي نوقشت أعلاه. بالنسبة للمشكلات الأبسط، غالبًا ما تجد نماذج الاستدلال الحل الصحيح في وقت مبكر من تفكيرها ولكنها

تستمر بعد ذلك في استكشاف حلول غير صحيحة. لاحظ أن توزيع الحلول غير الصحيحة (باللون الأحمر) مزاح بشكل أكبر نحو الأعلى باتجاه نهاية التفكير مقارنة بالحلول الصحيحة (باللون الأخضر). هذه الظاهرة، المشار إليها باسم 'التفكير المفرط' (Overthinking) في الأدبيات، تؤدي إلى إهدار الحوسبة. عندما تصبح المشكلات أكثر تعقيدًا بشكل معتدل، ينعكس هذا الاتجاه: تستكشف النماذج أولاً الحلول غير الصحيحة وتصل في الغالب لاحقًا في التفكير إلى الحلول الصحيحة. هذه المرة يكون توزيع الحلول غير الصحيحة (باللون الأحمر) مزاحًا بشكل أكبر نحو الأسفل مقارنة بالحلول الصحيحة (باللون الأخضر). أخيرًا، بالنسبة للمشكلات ذات التعقيد الأعلى، يظهر الانهيار،



(a) (b)

الشكل 7: اليسار والوسط: موضع وصحة الحلول الوسيطة ضمن مسارات الاستدلال عبر أربعة ألغاز عند مستويات تعقيد متفاوتة. تشير ✓ إلى الحلول الصحيحة، وتشير ✗ إلى الحلول غير الصحيحة، مع كثافة التوزيع الموضحة بالتظليل؛ اليمين: دقة الحل مقابل الموضع في التفكير للغز برج هانوي عند مستويات تعقيد مختلفة. تُظهر المشكلات البسيطة (N=1-3) دقة مبكرة تتناقص بمرور الوقت ('التفكير المفرط')، وتُظهر المشكلات المتوسطة (N=4-7) تحسنًا طفيفًا في الدقة مع استمرار الاستدلال، وتُظهر المشكلات المعقدة (N≥8) دقة قريبة من الصفر باستمرار، مما يشير إلى فشل كامل في الاستدلال.

مما يعني أن النموذج يفشل في توليد أي حلول صحيحة ضمن التفكير.

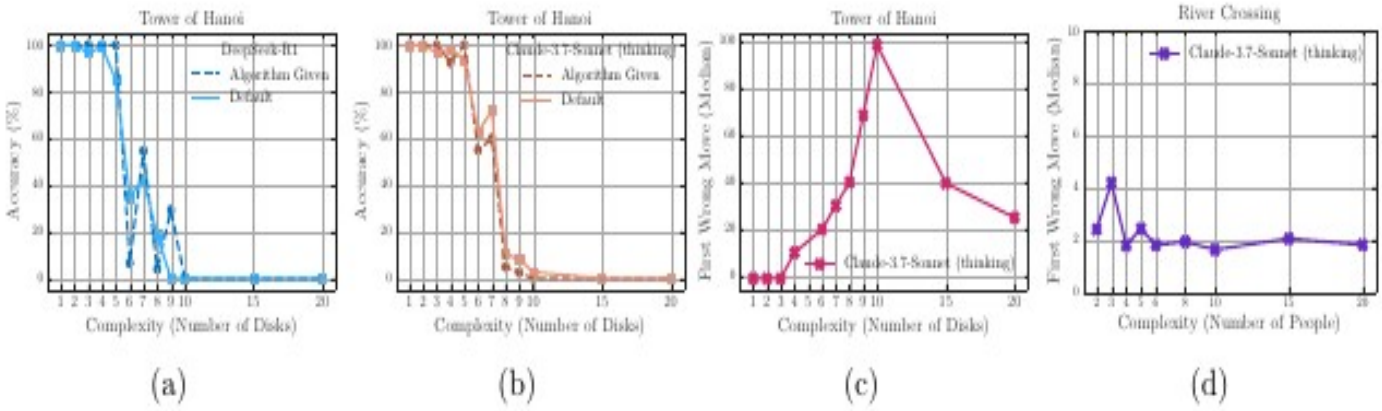
يعرض الشكل 7ب تحليلًا تكميليًا لدقة الحل ضمن مقاطع تسلسلية (فئات) للتفكير في بيئة برج هانوي. يمكن ملاحظة أنه بالنسبة للمشكلات الأبسط (N أصغر)، تميل دقة الحل إلى الانخفاض أو التذبذب مع تقدم التفكير، مما يقدم دليلًا إضافيًا على ظاهرة 'التفكير المفرط'. ومع ذلك، يتغير هذا الاتجاه بالنسبة للمشكلات الأكثر تعقيدًا، حيث تزداد دقة الحل مع تقدم التفكير – حتى عتبة معينة. بعد تجاوز عتبة التعقيد هذه، في 'وضع الانهيار'، تكون الدقة صفرًا.

4.4 أسئلة مفتوحة: السلوك المحير لنماذج الاستدلال

في هذا القسم، نقدم نتائج مفاجئة تتعلق بقيود نماذج الاستدلال في تنفيذ خطوات حل المشكلات الدقيقة، بالإضافة إلى إظهار سلوكيات مختلفة للنماذج بناءً على عدد الحركات.

كما هو موضح في الشكلين 8أ و 8ب، في بيئة برج هانوي، حتى عندما نقدم الخوارزمية في التلقين – بحيث يحتاج النموذج فقط إلى تنفيذ الخطوات المحددة – لا يتحسن الأداء، ولا يزال الانهيار الملحوظ يحدث عند نفس النقطة تقريبًا. هذا جدير بالملاحظة لأن إيجاد وابتكار حل يجب أن يتطلب حوسبة أكبر بكثير (على سبيل المثال، للبحث والتحقق) من مجرد تنفيذ خوارزمية معينة. هذا يسلط الضوء بشكل أكبر على قيود نماذج الاستدلال في التحقق وفي اتباع الخطوات المنطقية لحل مشكلة ما، مما يشير إلى أن هناك حاجة إلى مزيد من البحث لفهم قدرات المعالجة الرمزية لمثل هذه النماذج [44، 6].

علاوة على ذلك، في الشكلين 8ج و 8د، نلاحظ سلوكًا مختلفًا جدًا من نموذج Claude 3.7 Sonnet المفكر. في بيئة برج هانوي، غالبًا ما يحدث الخطأ الأول للنموذج في الحل المقترح في وقت لاحق بكثير، على سبيل المثال، حوالي الحركة 100 لـ (N=10)، مقارنة ببيئة عبور النهر، حيث يمكن للنموذج فقط إنتاج حل صالح حتى الحركة 4. لاحظ أن هذا النموذج يحقق أيضًا دقة شبه مثالية عند حل لغز برج هانوي بـ (N=5)، والذي يتطلب 31 حركة، بينما يفشل في حل لغز عبور النهر عندما يكون (N=3)، والذي له حل من 11 حركة. من المحتمل أن يشير هذا إلى أن أمثلة عبور النهر عندما تكون N>2 نادرة على الويب، مما يعني أن نماذج الاستدلال الكبيرة (LRMs) ربما لم تصادف أو تحفظ مثل هذه الحالات بشكل متكرر أثناء التدريب.



الشكل 8: (أ) و (ب) على الرغم من توفير خوارزمية الحل في التلقين، يحدث فشل التنفيذ عند نقاط مماثلة، مما يسلط الضوء على قيود نماذج الاستدلال في تنفيذ الخطوات المنطقية. (ج) و (د) بشكل ملحوظ، يُظهر نموذج Claude 3.7 Sonnet تسلسلات أطول بكثير خالية من الأخطاء في لغز برج هانوي مقارنة بالأخطاء المبكرة في سيناريو عبور النهر.

5. الخلاصة

في هذا البحث، نفحص بشكل منهجي نماذج الاستدلال الكبيرة (LRMs) الرائدة من منظور تعقيد المشكلة باستخدام بيئات ألغاز قابلة للتحكم. تكشف نتائجنا عن قيود أساسية في النماذج الحالية: على الرغم من آليات التأمل الذاتي المتطورة، تفشل هذه النماذج في تطوير قدرات استدلال قابلة للتعميم بعد تجاوز عتبات تعقيد معينة. حددنا ثلاثة أنظمة استدلال مميزة: تتفوق نماذج اللغة الكبيرة (LLMs) القياسية على نماذج الاستدلال الكبيرة (LRMs) عند التعقيد المنخفض، وتتفوق نماذج الاستدلال الكبيرة (LRMs) عند التعقيد

المتوسط، وينهار كلاهما عند التعقيد العالي. ومما يثير القلق بشكل خاص هو الانخفاض غير البديهي في جهد الاستدلال مع اقتراب المشكلات من التعقيد الحرج، مما يشير إلى وجود حد متأصل في توسع الحوسبة في نماذج الاستدلال الكبيرة (LRMs). كشف تحليلنا المفصل لمسارات الاستدلال أيضًا عن أنماط استدلال تعتمد على التعقيد، بدءًا من 'التفكير المفرط' غير الفعال في المشكلات الأبسط إلى الفشل الكامل في المشكلات المعقدة. تتحدى هذه الرؤى الافتراضات السائدة حول قدرات نماذج الاستدلال الكبيرة (LRM) وتشير إلى أن المناهج الحالية قد تواجه حواجز أساسية أمام الاستدلال القابل للتعميم. أخيرًا، قدمنا بعض النتائج المفاجئة حول نماذج الاستدلال الكبيرة (LRMs) التي تؤدي إلى عدة أسئلة مفتوحة للعمل المستقبلي. والأهم من ذلك، لاحظنا قيودها في إجراء الحساب الدقيق؛ على سبيل المثال، عندما قدمنا خوارزمية الحل للغز برج هانوي للنماذج، لم يتحسن أداؤها في هذا اللغز. علاوة على ذلك، كشف استقصاء حركة الفشل الأولى للنماذج عن سلوكيات مفاجئة. على سبيل المثال، يمكنها أداء ما يصل إلى 100 حركة صحيحة في لغز برج هانوي ولكنها تفشل في تقديم أكثر من 5 حركات صحيحة في لغز عبور النهر. نعتقد أن نتائجنا يمكن أن تمهد الطريق لإجراء تحقيقات مستقبلية في قدرات الاستدلال لهذه الأنظمة.

القيود

نقر بأن عملنا له قيود. فبينما تتيح بيئات الألغاز الخاصة بنا إجراء تجارب محكومة مع تحكم دقيق في تعقيد المشكلة، فإنها تمثل شريحة ضيقة من مهام الاستدلال وقد لا تلتقط تنوع مشكلات الاستدلال في العالم الحقيقي أو تلك التي تتطلب معرفة مكثفة. ومن الجدير بالذكر أن معظم تجاربنا تعتمد على الوصول إلى واجهة برمجة تطبيقات (API) الصندوق الأسود لنماذج الاستدلال الكبيرة (LRMs) الرائدة المغلقة، مما يحد من قدرتنا على تحليل الحالات الداخلية أو المكونات المعمارية. علاوة على ذلك، يفترض استخدام محاكيات الألغاز الحتمية أنه يمكن التحقق من الاستدلال بشكل مثالي خطوة بخطوة. ومع ذلك، في المجالات الأقل تنظيمًا، قد لا يكون مثل هذا التحقق الدقيق ممكنًا، مما يحد من قابلية نقل هذا التحليل إلى استدلالات أخرى أكثر قابلية للتعميم.

شكر وتقدير

يود المؤلفون أن يشكروا سكوت هوانغ، وبيتشن جيانغ، ومينسيك تشو، ومحمد سخاوت، وديفيد هاريسون، ومحمد رضا أرماندبور، وديفي كريشنا على ملاحظاتهم ودعمهم القيم.

المراجع

[1] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. arXiv .preprint arXiv:2412.16720, 2024

.OpenAI. Introducing openai o1. Jan 2024 [2]

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, [3]
,Qihao Zhu

Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms

.via reinforcement learning. arXiv preprint arXiv:2501.12948, 2025

.Anthropic. Claude 3.7 sonnet. Feb 2025 [4]

.Google. Gemini flash thinking. Google AI Blog, Jan 2025 [5]

Seyed Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy [6]
,Bengio

and Mehrdad Farajtabar. GSM-symbolic: Understanding the limitations of mathematical

reasoning in large language models. In The Thirteenth International Conference on Learning

.Representations, 2025

Francois Chollet, Mike Knoop, Gregory Kamradt, Bryan Landers, and Henry Pinkard. [7]
:Arc-agi-2

A new challenge for frontier ai reasoning systems. arXiv preprint arXiv:2505.11831, .2025

Gary Marcus. Five ways in which the last 3 months — and especially the deepseek [8]
era — have

vindicated "deep learning is hitting a wall". Marcus on AI (Substack), February 2025. Blog

.post

Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed [9]
Awadallah, Hany

Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, and et. al. Phi-3

technical report: A highly capable language model locally on your phone. CoRR, ,abs/2404.14219

.2024

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra [10] -Singh Chap

lot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile ,Saulnier

,Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril

Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. CoRR, ,abs/2310.06825

.2023

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al- [11] ,Dahle

Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony

Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur ,Hinsvark

Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste ,Rozière

Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris

Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne ,Wong

Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny

Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego ,Perino

Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael

Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis ,Anderson

Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah ,Korevaar

Hu Xu, Hugo Touvron, and et al. The llama 3 herd of models. CoRR, abs/2407.21783, .2024

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen [12] Lin, Sean

Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya ,Sanyal

Xiang Ren, Allyson Ettinger, Zaïd Harchaoui, and Yejin Choi. Faith and fate: Limits of

transformers on compositionality. In Alice Oh, Tristan Naumann, Amir Globerson, Kate ,Saenko

Moritz Hardt, and Sergey Levine, editors, Advances in Neural Information Processing Systems

Annual Conference on Neural Information Processing Systems 2023, NeurIPS :36 2023, New

.Orleans, LA, USA, December 10 - 16, 2023, 2023

R. Thomas McCoy, Shunyu Yao, Dan Friedman, Matthew Hardy, and Thomas L. [13] .Griffiths

Embers of autoregression: Understanding large language models through the problem they are

.trained to solve, 2023

Marianna Nezhurina, Lucia Cipolina-Kun, Mehdi Cherti, and Jenia Jitsev. Alice in [14] :wonderland

Simple tasks showing complete reasoning breakdown in state-of-the-art large .language models

.arXiv preprint arXiv:2406.02061, 2024

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, [15] ,Ed H. Chi

Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language

models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. ,Oh

editors, Advances in Neural Information Processing Systems 35: Annual Conference on Neural

Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, - November 28

.December 9, 2022, 2022

Mehran Kazemi, Najoung Kim, Deepti Bhatia, Xin Xu, and Deepak Ramachandran. [16] -Lam

bada: Backward chaining for automated reasoning in natural language. arXiv preprint

.arXiv:2212.13894, 2022

Hattie Zhou, Azade Nova, Hugo Larochelle, Aaron Courville, Behnam Neyshabur, [17] and Hanie

Sedghi. Teaching algorithmic reasoning via in-context learning. arXiv preprint ,arXiv:2211.09066

.2022

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke [18] Iwasawa. Large

language models are zero-shot reasoners. Advances in neural information processing ,systems

.2022 ,22213–35:22199

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang [19] Liu, and

Jun Zhao. Large language models are better reasoners with self-verification. In Houda ,Bouamor

Juan Pino, and Kalika Bali, editors, Findings of the Association for Computational :Linguistics

EMNLP 2023, pages 2550–2575, Singapore, December 2023. Association for Computational

.Linguistics

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu [20] .Chen

Making language models better reasoners with step-aware verifier. In Proceedings of the 61st

Annual Meeting of the Association for Computational Linguistics (Volume 1: Long ,Papers)

.pages 5315–5333, 2023

Eric Zhao, Pranjal Awasthi, and Sreenivas Gollapudi. Sample, scrutinize and scale: [21] Effective

.inference-time search by scaling verification. arXiv preprint arXiv:2502.01839, 2025

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STar: Bootstrapping [22] reasoning

with reasoning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, ,editors

.Advances in Neural Information Processing Systems, 2022

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, [23] and

Vaishnavh Nagarajan. Think before you speak: Training language models with pause .tokens

.In The Twelfth International Conference on Learning Representations, 2024

David Herel and Tomas Mikolov. Thinking tokens for language modeling. ArXiv, [24] ,abs/2405.08644

.2024

Zhihong Shao, Peiyi Wang, Runxin Xu Qihao Zhu, Junxiao Song, Mingchuan Zhang, [25] ,Y.K. Li

Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open

.language models, 2024

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, [26] ,Siva Reddy

Aaron Courville, and Nicolas Le Roux. Vineppo: Unlocking rl potential for llm reasoning

.through refined credit assignment, 2024

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish [27] Ivison, Faeze

Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya

Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris

Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh -Ha

jishirzi. Tülu 3: Pushing frontiers in open language model post-training. ArXiv, ,abs/2411.15124

.2024

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, [28] John

Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, et al. Reasoning models

.don’t always say what they think. arXiv preprint arXiv:2505.05410, 2025

Dacheng Li, Shiyi Cao, Tyler Griggs, Shu Liu, Xiangxi Mo, Eric Tang, Sumanth [29] Hegde, Kourosh

Hakhamaneshi, Shishir G Patil, Matei Zaharia, et al. Llms can easily learn to reason from

demonstrations structure, not content, is what matters! arXiv preprint .arXiv:2502.07374, 2025

[30] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the .overthinking of o1-like llms. arXiv preprint arXiv:2412.21187, 2024

[31] Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Hanjie Chen, Xia Hu, et al. Stop overthinking: A survey on efficient reasoning .for large language models. arXiv preprint arXiv:2503.16419, 2025

[32] Sara Vera Marjanović, Arkil Patel, Vaibhav Adlakha, Milad Aghajohari, Parishad BehnamGhader, Mehar Bhatia, Aditi Khandelwal, Austin Kraft, Benno Krojer, Xing Han Lù, et al. Deepseek-r1 thoughtology: Let's< think> about llm reasoning. arXiv preprint .arXiv:2504.07128, 2025

[33] Yuxiao Qu, Matthew YR Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel ,Beeching Ruslan Salakhutdinov, and Aviral Kumar. Optimizing test-time compute via meta reinforcement .fine-tuning. arXiv preprint arXiv:2503.07572, 2025

[34] Marthe Ballon, Andres Algaba, and Vincent Ginis. The relationship between reasoning and performance in large language models–o3 (mini) thinks harder, not longer. arXiv preprint .arXiv:2502.15631, 2025

[35] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base ?model

.arXiv preprint arXiv:2504.13837, 2025

Benjamin Estermann, Luca A. Lanzendörfer, Yannick Niedermayr, and Roger [36] .Wattenhofer

.Puzzles: A benchmark for neural algorithmic reasoning, 2024

Karthik Valmeekam, Alberto Olmo Hernandez, Sarath Sreedharan, and Subbarao [37] .Kambhampati

Large language models still can't plan (A benchmark for llms on planning and reasoning about

.change). CoRR, abs/2206.10498, 2022

Karthik Valmeekam, Kaya Stechly, and Subbarao Kambhampati. Llms still can't [38] plan; can

.lrms? a preliminary evaluation of openai's o1 on planbench. 2024

Wenjie Ma, Jingxuan He, Charlie Snell, Tyler Griggs, Sewon Min, and Matei [39] Zaharia. Reasoning

.models can be effective without thinking. arXiv preprint arXiv:2504.09858, 2025

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, [40] Teddy Lee, Jan

Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. arXiv preprint

.arXiv:2305.20050, 2023

Mathematical [41]

Association

of

.America

American

invitational

-math

ematics

examination

.(aime)

/https://maa.org/math-competitions

.american-invitational-mathematics-examination-aime, 2025. Accessed: 2025-05-15

.Art of Problem Solving [42]

.Amc historical results - aime i (february 1, 2024)

_https://artofproblemsolving.com/wiki/index.php/AMC_historical_results#AIME

.I_.28February_1.2C_2024.29, 2024. Accessed: 2025-05-15

.Art of Problem Solving [43]

.Amc historical results – aime i (february 6, 2025)

_https://artofproblemsolving.com/wiki/index.php/AMC_historical_results#AIME

.I_.28February_6.2C_2025.29, 2025. Accessed: 2025-05-15

Gary F Marcus. The algebraic mind: Integrating connectionism and cognitive [44] science. MIT

.press, 2003

Saul Amarel. On representations of problems of reasoning about actions. In [45] Readings in artificial

.intelligence, pages 2–22. Elsevier, 1981

.Günter Rote. Crossing the bridge at night. Bulletin of the EATCS, 78:241, 2002 [46]

أ. الملحق

في هذا الملحق، نقدم تفاصيل تكميلية للنص الرئيسي، بما في ذلك مواصفات إعداد التجارب، ونتائج إضافية، وتحليل موسع.

أ.1 تفاصيل حول مواصفات وتصميم بيئة الألغاز - أوصاف شاملة لجميع بيئات الألغاز الأربع، بما في ذلك أوصاف مشكلاتها، وتصميمات التلقين، والمحاكيات.

أ.1.1 برج هانوي

أ.1.2 قفز الداما

أ.1.3 عبور النهر

أ.1.4 عالم المكعبات

أ.2 تفاصيل التنفيذ - مواصفات إعداد التجارب الكاملة، وتكوينات النماذج، وتفاصيل خط أنابيب الاستخلاص، وتجارب تنفيذ الخوارزميات المحددة.

أ.3 تفاصيل حول التعقيد الحسابي

أ.3.1 توصيف العمق التركيبي

أ.3.2 الأداء مقابل العمق التركيبي

أ.4 نتائج وتحليلات إضافية - تحليل موسع يشمل أنماط جهد الاستدلال، وتحليل مفصل للفشل عبر جميع النماذج وبيئات الألغاز.

أ.1 تفاصيل حول مواصفات وتصميم بيئة الألغاز

أ.1.1 برج هانوي

وصف المشكلة. برج هانوي هو لغز تكراري كلاسيكي يُعد مشكلة رائعة لتقييم قدرات الاستدلال التسلسلي والتخطيط في نماذج الاستدلال. يتكون اللغز من ثلاثة أوتاد (مرقمة 0، 1، و 2 من اليسار إلى اليمين) وعدد N من الأقراص مختلفة الأحجام، حيث يُرقم كل قرص بشكل فريد من 1 (الأصغر) إلى N (الأكبر). في التكوين الأولي، تكون جميع الأقراص الـ N مكدسة على الوتد الأيسر (الوتد 0) بترتيب تنازلي للحجم، مع وجود القرص الأكبر في الأسفل والأصغر في الأعلى. يكون الوتدان المتبقيان (1 و 2) فارغين في البداية. الهدف هو نقل جميع الأقراص من الوتد 0 إلى الوتد 2، مع الحفاظ على نفس ترتيب الحجم (الأكبر في الأسفل، الأصغر في الأعلى). يخضع هذا اللغز لثلاثة قيود أساسية: (1) حركة قرص واحد: يمكن تحريك قرص واحد فقط في كل مرة؛ (2) الوصول إلى القرص العلوي: يمكن اختيار القرص العلوي فقط من أي وتد للحركة؛ و (3) قيد ترتيب الحجم: لا يجوز أبدًا وضع قرص أكبر فوق قرص أصغر. يُعد هذا اللغز منصة اختبار تقييم جيدة لقدرات الاستدلال والتخطيط للنماذج لأنه يتطلب من النماذج إظهار متطلبات معرفية رئيسية مثل تقسيم المشكلة إلى

مشكلات فرعية (التفكير التكراري)، وتتبع حالات متعددة ومواضع أقراص في وقت واحد (إدارة الذاكرة العاملة)، والالتزام بقواعد وقيود الحركة أثناء التخطيط المسبق (إرضاء القيود)، وتحديد الترتيب الصحيح للعمليات لتحقيق الهدف النهائي (التخطيط التسلسلي).

الحد الأدنى لعدد الحركات المطلوبة لحل لغز برج هانوي التكراري بعدد N من الأقراص هو $2^N - 1$، مما يجعله مشكلة تتوسع بشكل أسي. تتيح هذه الخاصية التحكم الدقيق في الصعوبة عن طريق تعديل حجم المشكلة بعدد الأقراص الأولية. ومع ذلك، في إطار تقييمنا، نركز على صحة الحل بدلاً من المثالية، ونُقيّم صحة كل حركة وقدرة النموذج على الوصول إلى الحالة المستهدفة كمعايير للنجاح.

تصميم التلقين. يبدأ تلقين النظام ببيان واضح للمشكلة يصف إعداد اللغز. ويذكر صراحة قواعد الحركة وهدف نقل جميع الأقراص إلى الوتد الثالث. لتسهيل الفهم، يتضمن التلقين أمثلة توضيحية بالإضافة إلى توقعات التنسيق والاستدلال الحاسمة.

System Prompt - Tower of Hanoi

You are a helpful assistant. Solve this puzzle for me.
There are three pegs and n disks of different sizes stacked on the first peg. The disks are numbered from 1 (smallest) to n (largest). Disk moves in this puzzle should follow:

1. Only one disk can be moved at a time.
2. Each move consists of taking the upper disk from one stack and placing it on top of another stack.
3. A larger disk may not be placed on top of a smaller disk.

The goal is to move the entire stack to the third peg.
Example: With 3 disks numbered 1 (smallest), 2, and 3 (largest), the initial state is [[3, 2, 1], [], []], and a solution might be:

```
moves = [[1, 0, 2], [2, 0, 1], [1, 2, 1], [3, 0, 2],
         [1, 1, 0], [2, 1, 2], [1, 0, 2]]
```

This means: Move disk 1 from peg 0 to peg 2, then move disk 2 from peg 0 to peg 1, and so on.
Requirements:

- When exploring potential solutions in your thinking process, always include the corresponding complete list of moves.
- The positions are 0-indexed (the leftmost peg is 0).
- Ensure your final answer includes the complete list of moves in the format:
  `moves = [[disk id, from peg, to peg], ...]`

يقدم تلقين المستخدم، بعد تلقين النظام، حالة اللغز المحددة بتكوينها الحالي الذي يُظهر توزيع الأقراص عبر الأوتاد، وتكوينها الهدف الذي يحدد الحالة المستهدفة.

User Prompt Template for N Disks - Tower of Hanoi

I have a puzzle with N disks of different sizes with
Initial configuration:

- Peg 0: N (bottom), ... 2, 1 (top)
- Peg 1: (empty)
- Peg 2: (empty)

Goal configuration:

- Peg 0: (empty)
- Peg 1: (empty)
- Peg 2: N (bottom), ... 2, 1 (top)

Rules:

- Only one disk can be moved at a time.
- Only the top disk from any stack can be moved.
- A larger disk may not be placed on top of a smaller disk.

Find the sequence of moves to transform the initial configuration into the goal configuration.

المحاكي. يستخدم إطار التقييم الخاص بنا محاكيات ألغاز منفصلة لكل لغز لضمان تقييم صارم ومتسق للحلول التي تم الحصول عليها من نماذج الاستدلال الكبيرة (LRMs). تم تصميم محاكي برج هانوي كبيئة ذات حالة تتعقب تكوينات الأقراص عبر ثلاثة أوتاد وتتحقق من صحة كل حركة مقترحة مقابل القيود الأساسية للغز. تتبع بنية المحاكي نمط تصميم معياري مع فصل واضح بين إدارة الحالة، والتحقق من صحة الحركة، والتحقق من الحل. في هذا المحاكي، لدينا فئة لغز (puzzle class) تتعقب تكوين القرص الحالي وتفرض القيود الأساسية للغز. لدينا أيضًا طريقة لتنفيذ كل حركة في إعداد اللغز وإجراء تحقق من أربع طبقات: التحقق من شروط حدود الوتد (0-2)، والتحقق من أن أوتاد المصدر تحتوي على أقراص، والتأكد من أن القرص المحدد هو العلوي، وفرض قيد ترتيب الحجم الذي يمنع وضع الأقراص الأكبر فوق الأقراص الأصغر. عند التحقق الناجح، تنفذ الطريقة نقل القرص وتُحدّث حالة اللعبة. بعد ذلك، تتم معالجة التحقق الكامل من الحل عن طريق معالجة قوائم الحركات بشكل تسلسلي، والتحقق من تحقيق الحالة المستهدفة.

أ.1.2 قفز الداما

وصف المشكلة. قفز الداما هو لغز إرضاء قيود أحادي البعد مُصمم لاختبار قدرات الاستدلال التسلسلي، والتخطيط، وفهم القواعد. يتكون اللغز من ترتيب خطي لقطع داما حمراء ('R')، وقطع داما زرقاء ('B')، ومساحة فارغة واحدة ('_'). في التكوين القياسي، يتم وضع N من قطع الداما الحمراء على الجانب الأيسر، تليها مساحة فارغة في المنتصف، و N من قطع الداما الزرقاء على الجانب الأيمن، لتشكيل لوحة خطية بطول 2N + 1. الهدف هو تبديل مواضع جميع قطع الداما الحمراء والزرقاء، مما يعكس بشكل فعال التكوين الأولي، حيث تنتهي قطع الداما الحمراء على اليمين وقطع الداما الزرقاء على اليسار. تخضع الحركة في هذا اللغز لقاعدتين أساسيتين: (1) حركة الانزلاق: يمكن لقطعة الداما أن تنزلق للأمام إلى مساحة فارغة مجاورة؛ و (2) حركة القفز: يمكن لقطعة الداما أن تقفز للأمام فوق قطعة داما واحدة بالضبط من اللون المقابل لتهبط في مساحة فارغة. لذلك، لا يمكن لقطع الداما التحرك للخلف نحو جانب البداية الخاص بها – يمكن لقطع الداما الحمراء التحرك فقط نحو اليمين، ويمكن لقطع الداما الزرقاء التحرك فقط نحو اليسار من التكوين الأولي. يمثل هذا اللغز تحديات معرفية تجعله منصة اختبار رائعة لنماذج الاستدلال. على سبيل المثال، يجب أن تُظهر النماذج بعض جوانب الاستدلال المكاني (تتبع مواضع قطع الداما والحركات الممكنة)، وإرضاء القيود (الالتزام بقواعد الحركة أثناء اللغز)، والتخطيط المسبق (توقع كيفية تأثير الحركات الحالية على المستقبل).

الاحتمالات نحو الهدف)، واستكشاف فضاء الحالات (البحث في تسلسلات الحركات الممكنة لإيجاد مسار حل صالح).

تتوسع صعوبة لغز قفز الداما مع عدد قطع الداما: مع وجود N من قطع الداما من كل لون، يتطلب الحل الأدنى $(N+1)^2 - 1$ حركة، مما يخلق علاقة تربيعية بين حجم المشكلة وتعقيد الحل. في إطار تقييمنا، نركز بشكل أساسي على صحة الحل بدلاً من المثالية، ونُقيّم كل حركة مقابل قيود اللغز ونؤكد أن الحالة النهائية تطابق التكوين الهدف. يتيح لنا هذا النهج تحديد حالات فشل الاستدلال وانتهاكات القيود التي قد تحدث أثناء عملية الحل بدقة.

تصميم التلقين. يبدأ تلقين النظام ببيان واضح للمشكلة يصف إعداد اللغز وقواعد الحركة. ويذكر صراحة الهدف ويقدم مثالاً ملموسًا بتكوين لوحة صغير لتوضيح كيفية تمثيل الحركات.

System Prompt - Checker Jumping

You are a helpful assistant. Solve this puzzle for me.
On a one-dimensional board, there are red checkers ('R'), blue checkers ('B'), and one empty space ('_'). A checker can move by either:

1. Sliding forward into an adjacent empty space, or
2. Jumping over exactly one checker of the opposite color to land in an empty space.

The goal is to swap the positions of all red and blue checkers, effectively mirroring the initial state.
Example: If the initial state is ['R', '_', 'B'], the goal is to reach ['B', '_', 'R']. Your solution should be a list of moves where each move is represented as [checker_color, position_from, position_to]. For example:

```
moves = [['R', 0, 1], ['B', 2, 0], ['R', 1, 2]]
```

This means: Move the red checker from position 0 to 1, then move the blue checker from position 2 to 0, and so on.
Requirements:

- When exploring potential solutions in your thinking process, always include the corresponding complete list of moves.
- The positions are 0-indexed (the leftmost position is 0).
- Ensure your final answer includes the complete list of moves for final solution in the format: `moves = [[checker_color, position_from, position_to], ...]`

يقدم تلقين المستخدم حالة اللغز المحددة بتكوينها الأولي للوحة، وحالتها الهدف.

User Prompt Template for N Checkers - Checker Jumping

I have a puzzle with 2N+1 positions, where N red checkers ('R') on left, N blue checkers ('B') on right, and one empty space ('_') in between are arranged in a line.
Initial board: R R ... R _ B B ... B
Goal board: B B ... B _ R R ... R
Rules:

- A checker can slide into an adjacent empty space.
- A checker can jump over exactly one checker of the opposite color to land in an empty space.
- Checkers cannot move backwards (towards their starting side).

Find the minimum sequence of moves to transform the initial board into the goal board.

المحاكي. يستخدم إطار التقييم الخاص بنا محاكيًا مخصصًا للتحقق من صحة حلول لغز قفز الداما. يُنفذ

المحاكي نظام تحقق شامل يفرض جميع قيود اللغز مع تتبع تطور الحالة على امتداد مسار الحل. تم تصميم محاكي قفز الداما كبيئة ذات حالة تتعقب موضع جميع قطع الداما والمساحة الفارغة، وتتحقق من صحة كل حركة لحل معين مقابل قواعد حركة اللغز. يبدأ المحاكي بالتحقق من أن كلًا من الحالتين الأولية والنهائية جيدة التكوين، وتحتويان على نفس العدد من قطع الداما الحمراء والزرقاء ومساحة فارغة واحدة بالضبط. بعد ذلك، يتم تنفيذ كل حركة باستخدام طريقة تُجري تحققًا متعدد الطبقات: التحقق من حدود الموضع، والتأكد من لون قطعة الداما الصحيح في المصدر، وضمان أن المواضع المستهدفة فارغة، والتحقق من أنواع الحركات إما كانزلاقات (مسافة=1) أو قفزات (مسافة=2). يفرض المحاكي قيودًا اتجاهية تمنع الحركة للخلف (تتحرك قطع الداما الحمراء يمينًا، وتتحرك قطع الداما الزرقاء يسارًا) ويتحقق من صحة حركات القفز من خلال التأكد من وجود قطعة داما ذات لون معاكس في الموضع الأوسط. عند التحقق الناجح، تنفذ الطريقة نقل قطعة الداما عن طريق تحديث المواضع وإخلاء المصدر. بعد ذلك، تتم معالجة تسلسلات الحركات الكاملة مع التحقق من الحالة المستهدفة النهائية.

أ.1.3 عبور النهر

وصف المشكلة. عبور النهر هو لغز تخطيط لإرضاء القيود يختبر التنسيق متعدد الوكلاء وإدارة القيود. يُعد هذا اللغز تعميمًا لمشكلات كلاسيكية مثل مشكلة المبشرين وأكلة لحوم البشر ومشكلة الجسر والشعلة، والتي دُرست على نطاق واسع في أدبيات التخطيط [45، 46]. يتضمن لغز عبور النهر عدد N من الفاعلين (يُرمز لهم بـ $a_1, a_2, \ldots, a_N$) وعدد N من الوكلاء المقابلين لهم (يُرمز لهم بـ $A_1, A_2, \ldots, A_N$) الذين يجب عليهم عبور نهر باستخدام قارب. في الحالة الأولية، يكون جميع الأفراد البالغ عددهم 2N على الضفة اليسرى للنهر. الهدف هو نقل الجميع بأمان إلى الضفة اليمنى. يعمل اللغز بموجب عدة قيود حركة رئيسية: (1) قيد سعة القارب: يمكن للقارب أن يحمل عدد k من الأفراد على الأكثر في المرة الواحدة، حيث يُضبط k عادةً على 2 للألغاز الأصغر($3 \le N$) و 3 للألغاز الأكبر ($5 \le N$)؛ (2) قيد عدم فراغ القارب: لا يمكن للقارب أن يسافر فارغًا ويجب أن يكون على متنه شخص واحد على الأقل؛ (3) قيد السلامة: لا يمكن لفاعل أن يكون في وجود وكيل آخر ما لم يكن وكيله الخاص موجودًا أيضًا، حيث يجب على الوكلاء حماية عملائهم من الوكلاء المنافسين. ينطبق قيد السلامة هذا على كل من الضفاف وفي القارب. يتطلب هذا اللغز تخطيطًا معقدًا وتتبعًا للحالة حيث يجب على المشاركين تنسيق عمليات عبورهم بعناية مع الحفاظ على قيود السلامة في جميع الأوقات. يجب على برنامج الحل أن يستدل من خلال مجموعات مختلفة من الأفراد الذين يمكنهم السفر معًا بأمان، وتحديد من يجب أن يعود بالقارب بعد العبور، والتخطيط الاستراتيجي لتسلسل ينقل الجميع في النهاية إلى الضفة اليمنى دون انتهاك أي قيود. يمكن التحكم في تعقيد هذه المهمة عن طريق تعديل عدد أزواج الفاعل-الوكيل وسعة القارب، مما يخلق تحديًا قابلاً للتوسع لنماذج الاستدلال.

تصميم التلقين. يقدم تلقين النظام الترميز لتمثيل الفاعلين والوكلاء، ويحدد تنسيق الحل كقائمة بحركات القارب، ويقدم مثالًا بسيطًا لتوضيح التنسيق.

System Prompt - River Crossing

You are a helpful assistant. Solve this puzzle for me.
You can represent actors with a_1, a_2, ... and agents with A_1, A_2, ... Your solution must be a list of boat moves where each move indicates the people on the boat. For example, if there were two actors and two agents, you should return:

```
moves=[["A_2", "a_2"], ["A_2"], ["A_1", "A_2"], ["A_1"], ["A_1", "a_1
    "]]
```

which indicates that in the first move, A_2 and a_2 row from left to right, and in the second move, A_2 rows from right to left and so on.
Requirements:

- When exploring potential solutions in your thinking process, always include the corresponding complete list of boat moves.
- The list shouldn't have comments.
- Ensure your final answer also includes the complete list of moves for final solution.

يقدم تلقين المستخدم حالة اللغز المحددة مع عدد N من أزواج الفاعل والوكيل، وسعة القارب k، وقيد السلامة الذي يجب الالتزام به على امتداد الحل.

User Prompt Template for N Pairs - River Crossing

N actors and their N agents want to cross a river in a boat that is capable of holding only k people at a time, **with the constraint that no actor can be in the presence of another agent, including while riding the boat, unless their own agent is also present**, because each agent is worried their rivals will poach their client. Initially, all actors and agents are on the left side of the river with the boat. How should they cross the river? (Note: the boat cannot travel empty)

المحاكي. يستخدم إطار التقييم الخاص بنا محاكيًا مخصصًا للتحقق من صحة حلول لغز عبور النهر المستخرجة. يتتبع المحاكي حالة جميع الأفراد (الفاعلين والوكلاء) وموضع القارب مع فرض جميع قيود اللغز. يتم تنفيذ كل حركة بتحقق متعدد الخطوات: التحقق من حدود سعة القارب، والتحقق من أن جميع الركاب على جانب القارب الحالي، وفرض قيد السلامة الحاسم المتمثل في عدم إمكانية تواجد الفاعلين بصحبة وكلاء آخرين دون وجود وكيلهم الخاص، سواء على متن القارب أو على كل ضفة بعد الحركة. يقوم المحاكي بإدارة تحديد موضع القارب ديناميكيًا، والتبديل تلقائيًا بين الجانبين بعد كل عبور، والتحقق من صحة الحالة الكاملة بعد كل حركة لضمان عدم حدوث انتهاكات للسلامة على أي من الضفتين. بعد ذلك، يتم التحقق من تسلسلات العبور الكاملة للتأكد من وصول جميع الأفراد البالغ عددهم 2N بنجاح إلى الضفة اليمنى.

أ.1.4 عالم المكعبات

وصف المشكلة. عالم المكعبات هو لغز تخطيط كلاسيكي دُرس مؤخرًا لتحليل قدرات التخطيط لنماذج اللغة الكبيرة (LLMs) [37، 38]. يتضمن اللغز عدة أكوام من المكعبات (A، B، C، إلخ) يجب إعادة ترتيبها من تكوين أولي إلى تكوين هدف محدد. يتم تحديد كل مكعب بشكل فريد من خلال حرفه، والهدف هو إيجاد الحد الأدنى من تسلسل الحركات اللازمة لتحويل الحالة الأولية إلى الحالة المستهدفة. يعمل اللغز فقط بموجب قيدين أساسيين: (1) حركة المكعب العلوي: يمكن تحريك المكعب العلوي فقط من أي كومة؛ و (2) الوضع الصالح: لا يمكن وضع المكعب إلا على موضع فارغ أو فوق مكعب آخر. تخلق هذه القيود مشكلة تخطيط يصبح فيها ترتيب العمليات حاسمًا، حيث قد تتطلب بعض التكوينات وضعًا مؤقتًا للمكعبات للوصول إلى تلك الموجودة أسفلها لاحقًا. يُعد عالم المكعبات منصة اختبار رائعة لتقييم قدرات التخطيط في نماذج الاستدلال لأنه يتطلب تفكيرًا مستقبليًا وتتبعًا للحالة. فحصت دراسات حديثة هذا اللغز في تكوينات مختلفة، بما في ذلك الإعدادات المبسطة التي تحتوي على ما لا يقل عن 3 إلى 5 مكعبات، لتقييم أداء نماذج اللغة الكبيرة (LLM) في مهام التخطيط التسلسلي [37، 38]. يجب أن تُظهر النماذج القدرة على تحليل تحويلات الحالة المعقدة إلى حركات تسلسلية صالحة، والاستدلال حول التبعيات بين المكعبات (على سبيل المثال، إزالة حجب المكعبات السفلية قبل الوصول إليها)، والتخطيط الفعال للمسارات إلى الحالة المستهدفة دون حركات غير قانونية.

يمكن توسيع نطاق صعوبة هذا اللغز عن طريق تعديل عدة معلمات: عدد المكعبات، وعدد الأكوام، وتعقيد التكوينات الأولية والنهائية. نحن نتحكم بشكل أساسي في التعقيد من خلال عدد المكعبات N، مع اتباع أنماط هيكلية واضحة في التكوينات الأولية والنهائية. في تصميمنا التجريبي، يقسم التكوين الأولي باستمرار المكعبات N بين كوميتين بترتيب أبجدي، مع ترك الكومة الثالثة فارغة كمساحة عمل. يدمج تكوين الهدف جميع المكعبات في الكومة الأولى بنمط متشابك منهجي يتناوب بين المكعبات من الكومتين الأوليتين، مع تحديد موضع محدد يتطلب تفكيكًا كاملاً وإعادة تجميع للأكوام الموجودة. على سبيل المثال، بالنسبة لـ N = 4، تحتوي الحالة الأولية على مكعبات مقسمة بين كوميتين [["A", "B"], ["C", "D"], []] وتتطلب الحالة المستهدفة [["A", "C", "B", "D"], [], []] تشابك المكعبات من كلتا الكومتين؛ وبالنسبة لـ N = 6، يجب تحويل الحالة الأولية [["A", "B", "C"], ["D", "E", "F"], []] إلى [["A", "B", "D", "E", "C", "F"], [], []]، لتشكيل نمط متناوب معقد. مع زيادة N، ينمو فضاء الحالة بشكل عاملي، ويزداد طول الحل الأدنى خطيًا تقريبًا مع N. بالنسبة للقيم الصغيرة لـ N (2-7)، تختبر الألغاز التخطيط الأساسي؛ وبالنسبة للقيم المتوسطة (8-20)، فإنها تتطلب استدلالًا أكثر تعقيدًا مع آفاق تخطيط أطول؛ وبالنسبة للقيم الكبيرة (N > 20)، فإنها تتحدى حدود قدرات الاستدلال التسلسلي من خلال طلب حركات مؤقتة واسعة النطاق والتعرف على الأنماط عبر مسارات حل طويلة.

تصميم التلقين. يقدم تلقين النظام القواعد الأساسية للغز عالم المكعبات، ويحدد تنسيق تمثيل الحركة، ويقدم مثالًا بسيطًا لتوضيح بنية الحل.

System Prompt - Blocks World

You are a helpful assistant. Solve this puzzle for me.
In this puzzle, there are stacks of blocks, and the goal is to rearrange them into a target configuration using a sequence of moves where:

- Only the topmost block from any stack can be moved.
- A block can be placed either on an empty position or on top of another block.

Example: With initial state `[["A", "B"], ["C"], []]` and goal state `[["A"], ["B"], ["C"]]`, a solution might be:

```
moves = [["C", 1, 2], ["B", 0, 1]]
```

This means: Move block C from stack 1 to stack 2, then move block B from stack 0 to stack 1.
Requirements:

- When exploring potential solutions in your thinking process, always include the corresponding complete list of moves.
- Ensure your final answer also includes the complete list of moves for final solution in the format: `moves = [[block, from stack, to stack], ...]`

يقدم تلقين المستخدم حالة اللغز المحددة مع التكوينات الأولية والنهائية المعطاة، ويُذكِّر النموذج صراحةً بقيد الحركة.

User Prompt Template for N Blocks - BlocksWorld

I have a puzzle with N blocks.
Initial state:

Stack 0: $blocks_0$ (top)

Stack 1: $blocks_1$ (top)

...

Stack m: $blocks_m$ (top)

Goal state:

Stack 0: $goal_blocks_0$ (top)

Stack 1: $goal_blocks_1$ (top)

...

Stack m: $goal_blocks_m$ (top)

Find the minimum sequence of moves to transform the initial state into the goal state. Remember that only the topmost block of each stack can be moved.

المحاكي. يستخدم إطار التقييم الخاص بنا محاكيًا مخصصًا للتحقق من صحة حلول لغز عالم المكعبات المستخرجة. يدير المحاكي حالة جميع المكعبات عبر الأكوام مع فرض قيود حركة اللغز. يتم تنفيذ كل حركة في إعداد اللغز بتحقق من ثلاث طبقات: التحقق من أن مؤشرات الكومة ضمن الحدود، والتأكد من أن كومة المصدر تحتوي على مكعبات، وضمان أن المكعب المحدد في أعلى كومته (فرض قاعدة حركة المكعب العلوي فقط). عند التحقق الناجح، يتم تنفيذ نقل المكعب ويُسحب المكعب من كومة المصدر ويُلحق بكومة الوجهة. أخيرًا، تتم معالجة تسلسلات الحل الكاملة لحركات المكعبات والتحقق من أن التكوين الناتج يطابق الحالة المستهدفة النهائية.

أ.2 تفاصيل التنفيذ

التكوينات. استخدمت تجاربنا بشكل أساسي نماذج الاستدلال ونظائرها غير المفكرة لتمكين التحليل الشامل لعملية التفكير. اخترنا على وجه التحديد Claude 3.7 Sonnet (مفكر/غير مفكر) و DeepSeek-R1/V3 نظرًا لقدرتهما على توفير الوصول إلى مسارات التفكير، وهو مطلب حاسم لتحليلنا. بالنسبة للتجارب التي تركز فقط على مقاييس الدقة النهائية، قمنا أيضًا بتضمين نتائج من نماذج o3-mini من OpenAI، حيث تفتقر إلى الوصول إلى الأفكار. بالنسبة لنماذج Claude 3.7 Sonnet (المفكرة وغير المفكرة)، استخدمنا أقصى ميزانية توليد تبلغ 64,000 رمز، تم الوصول إليها من خلال واجهة برمجة التطبيقات (API). تم ضبط درجة الحرارة (Temperature) على 1.0 لجميع عمليات تشغيل واجهة برمجة التطبيقات (API) (عمليات تشغيل Claude-3.7-Sonnet و o3-mini). أُجريت التجارب باستخدام DeepSeek-R1، و DeepSeek-V3، و DeepSeek-R1-Distill-Qern-32B على خوادم محلية مع ضبط أقصى طول للتوليد على 64,000 وضبط درجة الحرارة على 1.0. في جميع التجارب، قمنا بتوليد 25 عينة لكل حالة لغز عند كل مستوى تعقيد (قيمة N) وأبلغنا عن متوسطات الأداء عبر جميع العينات.

استخلاص الحلول. تم تطوير خط أنابيب استخلاص مخصص لمعالجة استجابات النماذج ومسارات الاستدلال الوسيطة (الأفكار). يتكون خط الأنابيب من عدة مكونات رئيسية. قمنا بتنفيذ مستخلصات مرنة قائمة على التعبيرات النمطية (regex-based extractors) لتحديد محاولات الحل المحتملة في كل من الاستجابة النهائية ومسار التفكير. تحدد عملية الاستخلاص أنماط الحل باستخدام التعبيرات النمطية (كل من أنماط "moves =" الصريحة والحلول البديلة القائمة على الأقواس). نقوم بمعالجة وتنظيف كل حل مرشح مستخلص عن طريق (1) إزالة التعليقات من القائمة (النص الذي يلي "#" في أي سطر)، و (2) تطبيع تنسيقات الحركة إلى ما هو مقترح في السياق لضمان بنية متسقة. بعد ذلك، نتحقق من صحة تنسيق الحل وبنيته لتصفية التطابقات غير الصالحة. أثناء الاستخلاص، نلتقط أيضًا بيانات وصفية لموضع الرمز لكل حل مستخلص. والجدير بالذكر، لتتبع الموضع بدقة ضمن مسارات التفكير، استخدمنا نفس مُرمِّز الرموز (cl100k_base) مثل النموذج المقابل لحساب الرموز عبر جميع التجارب. تم أيضًا تطبيع مواضع الرموز بالنسبة لطول التفكير لتمكين المقارنة عبر العينات. أخيرًا، نتأكد من أن الحلول المسجلة ضمن مسار التفكير فريدة وأن الحلول المكررة (قائمة حركات متطابقة) قد تمت تصفيتها. في حالة الحلول المكررة، يتم تسجيل الحل الأول فقط للتحليل.

تقييم الحلول. بعد الاستخلاص، يتم تمرير كل حل مرشح إلى المحاكي المقابل للغز للتحقق الدقيق. يأخذ المحاكي الحل كقائمة من الحركات ويُقيّمه بالنسبة للغز (راجع الملحق أ.1 للحصول على تفاصيل كل محاكي

لغز). يتم تنفيذ كل حركة في الحل التركيبي بشكل تسلسلي وفقًا للحركات السابقة وقواعد اللغز. بعد ذلك، تتم مقارنة الحالة النهائية التي تم الحصول عليها من جميع الحركات في التسلسل بالحالة المستهدفة للغز لتحديد صحة الحل الكامل. بالنسبة للحلول غير الصحيحة، يتم أيضًا جمع تفاصيل حركة الفشل الأولى ونوع الفشل أثناء التحقق من الحركة باستخدام محاكي اللغز.

تنفيذ الخطوات المحددة. بالإضافة إلى حل المشكلات المفتوح عبر الألغاز المختلفة، أجرينا أيضًا تجارب مركزة لاختبار كيفية تأثير توفير خوارزمية الحل الصريحة للتوجيه مع الخطوات المحددة على سلوك نماذج الاستدلال هذه (القسم 4.4). توقعنا أن إيجاد وابتكار حل من البداية يجب أن يتطلب حوسبة أكبر بكثير للنموذج (على سبيل المثال، للبحث والتحقق) من مجرد اتباع خطوات خوارزمية معينة. ومع ذلك، تُظهر النتائج في الشكلين 8أ و 8ب أن سلوك نماذج الاستدلال لا يتغير كثيرًا وأن الانهيار لا يزال يحدث عند نفس النقاط تقريبًا كما كان من قبل مع هذا الإعداد. تعزز هذه النتيجة الدليل على أن القيد ليس فقط في حل المشكلات واكتشاف استراتيجية الحل ولكن أيضًا في التحقق المنطقي المتسق وقيد تنفيذ الخطوات على امتداد سلاسل الاستدلال المولدة.

على سبيل المثال، يتم تزويد النماذج بخوارزمية تكرارية كاملة لحل لغز برج هانوي على النحو التالي. تم إلحاق مسودة الخوارزمية هذه بتلقين المشكلة القياسي لاختبار تأثيرها على سلوك الاستدلال.

Example of Prescribed Algorithm for Tower of Hanoi

Here is a pseudocode of recursive algorithm to solve the puzzle:

```
ALGORITHM Solve(n, source, target, auxiliary, moves)
    // n = number of disks to move
    // source = starting peg (0, 1, or 2)
    // target = destination peg (0, 1, or 2)
    // auxiliary = the unused peg (0, 1, or 2)
    // moves = list to store the sequence of moves

    IF n equals 1 THEN
        // Get the top disk from source peg
        disk = the top disk on the source peg
        // Add the move to our list: [disk_id, source, target]
        ADD [disk, source, target] to moves
        RETURN
    END IF

    // Move n-1 disks from source to auxiliary peg
    Solve(n-1, source, auxiliary, target, moves)

    // Move the nth disk from source to target
    disk = the top disk on the source peg
    ADD [disk, source, target] to moves

    // Move n-1 disks from auxiliary to target
    Solve(n-1, auxiliary, target, source, moves)
END ALGORITHM
```

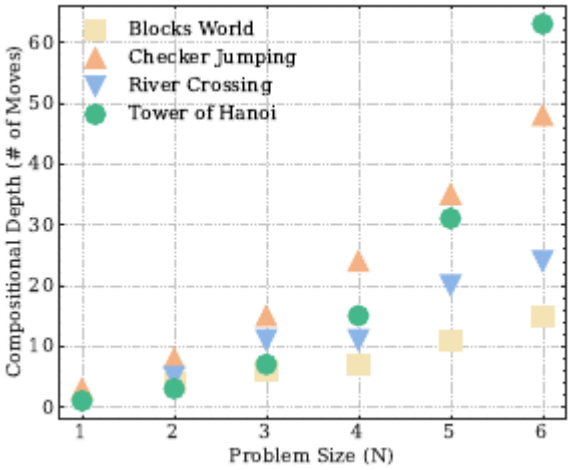
To solve the entire puzzle of moving n disks from peg 0 to peg 2:
1. Initialize an empty list 'moves'
2. Execute Solve(n, 0, 2, 1, moves)
3. The 'moves' list will contain the complete solution

Note: When executing this pseudocode, track which disk is currently on top of each peg. The disk IDs in the moves list should correspond to the actual disk being moved.
You can use this algorithm as a scratchpad to help you solve the problem step by step.

أ.3 تفاصيل حول التعقيد الحسابي

أ.3.1 توصيف العمق التركيبي

العمق التركيبي هو عدد العمليات التسلسلية (أي الحركات) المطلوبة للوصول إلى حل كامل. يوضح الشكل 9 كيف يتوسع هذا العمق مع حجم المشكلة (N) عبر بيئات الألغاز الأربع الخاصة بنا. لكل لغز نمط نمو مميز، يعكس تعقيده الحسابي الأساسي. على سبيل المثال، يُظهر لغز برج هانوي نموًا أسيًا ($2^N - 1$) ويُظهر لغز قفز الداما توسعًا تربيعيًا ($(N+1)^2 - 1$). تُظهر ألغاز عبور النهر وعالم المكعبات نموًا أكثر اعتدالًا، شبه خطي مع N. تتيح لنا ملامح العمق التركيبي المتفاوتة هذه تقييم كيفية تعامل نماذج الاستدلال اللغوية مع أنواع مختلفة من تحديات الاستدلال التسلسلي وما إذا كانت دقتها مرتبطة دائمًا بالعمق التركيبي المطلوب لحل اللغز. يتم توفير مزيد من التفاصيل المتعلقة بهذا التحليل في الشكل 10 في الملحق أ.4.



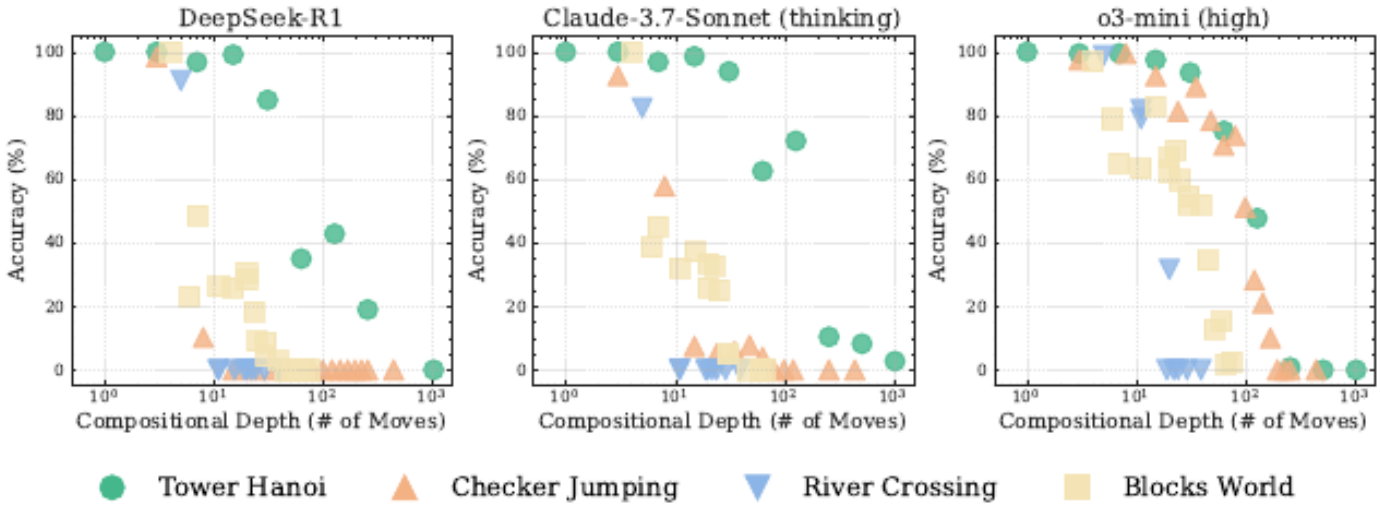
الشكل 9: العمق التركيبي (عدد الحركات المطلوبة) عبر أحجام المشكلات المختلفة لبيئات الألغاز الأربع الخاصة بنا.

أ.3.2 الأداء مقابل العمق التركيبي

بينما يشير الحدس إلى وجود ارتباط سلبي بين تعقيد المشكلة ودقة النموذج، يكشف تحليلنا عن علاقة أكثر دقة بين العمق التركيبي وأداء نماذج الاستدلال الكبيرة (LRM). يوضح الشكل 10 هذا عبر ثلاثة نماذج استدلال حديثة (Claude-3.7-Sonnet مع التفكير، و DeepSeek-R1، و o3-mini) على مجموعة الألغاز الخاصة بنا. ضمن أنواع الألغاز الفردية، نلاحظ الارتباط السلبي المتوقع: مع زيادة العمق التركيبي، تنخفض دقة النموذج باستمرار. ومع ذلك، عبر أنواع الألغاز المختلفة، تنكسر هذه العلاقة. قد تواجه النماذج صعوبة في حل الألغاز ذات العمق التركيبي المنخفض بينما تنجح في حل ألغاز مختلفة ذات عمق تركيبي أعلى. على سبيل المثال، تحقق النماذج دقة >50% في حالات لغز برج هانوي التي تتطلب ما يقرب من 10^2 حركة، ومع ذلك تفشل باستمرار في حل ألغاز عبور النهر ذات العمق التركيبي الأقل بكثير ($\sim 10^1$ حركة).

أ.4 نتائج وتحليلات موسعة

تحليل الفشل. يوفر فهم المواضع التي تفشل فيها النماذج ضمن خطوات الاستدلال التركيبي رؤى تتجاوز مقاييس النجاح الثنائية. يتطلب تقييم الدقة لدينا تنفيذًا مثاليًا لتسلسلات الحركات بأكملها – تؤدي حركة واحدة غير صحيحة إلى الفشل. لفحص أنماط الفشل بشكل أكثر تفصيلاً، نحلل العمق التركيبي الذي ترتكب فيه النماذج أولى حركاتها غير الصحيحة عبر مستويات تعقيد المشكلة المختلفة.



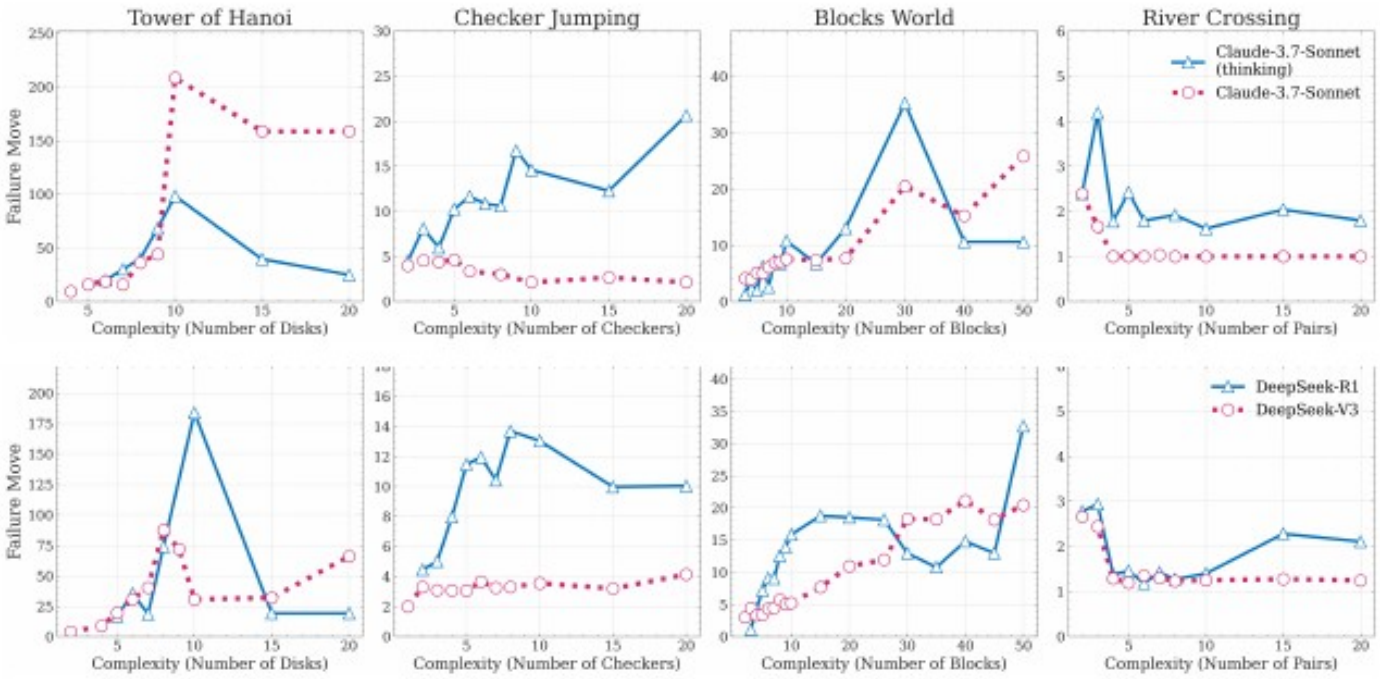
الشكل 10: الدقة مقابل العمق التركيبي (عدد الحركات المطلوبة) لثلاثة نماذج استدلال كبيرة (LRMs) (DeepSeek-R1، و Claude-3.7-Sonnet مع التفكير، و o3-mini) عبر أربع بيئات ألغاز.

يوضح الشكل 11 معرّف حركة الفشل مقابل تعقيد المشكلة (N) ضمن تسلسل الحل. يقارن الصف العلوي نموذج Claude-3.7-Sonnet مع وبدون قدرات التفكير، بينما يقارن الصف السفلي نموذج DeepSeek-R1 (مفكر) مع نموذج DeepSeek-V3 (غير مفكر). توضح هذه المقارنات كيف تؤثر آليات التفكير لنماذج الاستدلال الكبيرة (LRMs) على أنماط الفشل في مهام الاستدلال التركيبي للألغاز. تظهر عدة أنماط غير بديهية من تحليلنا. أولاً، تُظهر النماذج سلوك فشل غير رتيب فيما يتعلق بتعقيد المشكلة – وهي حالات تفشل فيها النماذج في وقت مبكر من تسلسل الحل لقيم N أعلى على الرغم من أنها تتطلب حلولًا أطول بشكل عام. على سبيل المثال، في لغز برج هانوي، تفشل النماذج أحيانًا عند أقل من 50 حركة لـ N = 15 ولكنها تنجح عبر أكثر من 100 حركة لـ N = 8، مما يتناقض مع التوقع بأن التخطيط والتنفيذ الخوارزمي الفعال لنفس اللغز يجب أن يحافظ على أنماط فشل متسقة بالنسبة لتقدم الحل. يشير هذا إلى وجود تناقضات أساسية في كيفية تطبيق النماذج (كل من نماذج الاستدلال الكبيرة (LRMs) ونظائرها من نماذج اللغة الكبيرة (LLM) القياسية غير المفكرة) لاستراتيجيات الحل المكتسبة عبر مقاييس مشكلات مختلفة. أيضًا، نلاحظ أنه في أنظمة التعقيد العالي حيث يواجه كلا متغيري النموذج انهيارًا كاملًا للدقة، على سبيل المثال، لغز برج هانوي مع N ≥ 15 وعالم المكعبات مع N ≥ 40، تحافظ النماذج غير المفكرة أحيانًا على الأداء بشكل أعمق في تسلسل الحل وتكون قادرة على الفشل في حركات لاحقة مقارنة بالمتغيرات التي تدعم التفكير. هذا مثير للاهتمام لأنه يُظهر أن حالات فشل الاستدلال التركيبي في نماذج اللغة الكبيرة (LLMs) ليست ببساطة بسبب عدم كفاية طول السياق أو حوسبة الاستدلال، بل تعكس قيودًا أساسية في كيفية حفاظ النماذج على الاتساق الخوارزمي عبر مقاييس المشكلات.

نقوم أيضًا بتحليل الخصائص التوزيعية لحركات الفشل لفهم اتساق وموثوقية استدلال النموذج. يعرض الشكل 12 توزيعات كثافة مواضع حركات الفشل المجمعة عبر جميع مستويات تعقيد المشكلة لكل بيئة لغز، مع مقارنة النماذج المفكرة وغير المفكرة ضمن نفس العائلة. بناءً على الشكل، تُظهر النماذج المفكرة (Claude-3.7-Sonnet مع التفكير و DeepSeek-R1) باستمرار متوسط مواضع فشل أعلى عبر جميع الألغاز، كما هو موضح بالخطوط العمودية المتقطعة التي تُظهر متوسط الفشل الأول في تسلسل الحركات. ومع ذلك، فإن شكل توزيع النماذج المفكرة غالبًا ما يكون له تباين أعلى في أنماط فشلها. يشير هذا إلى أنه

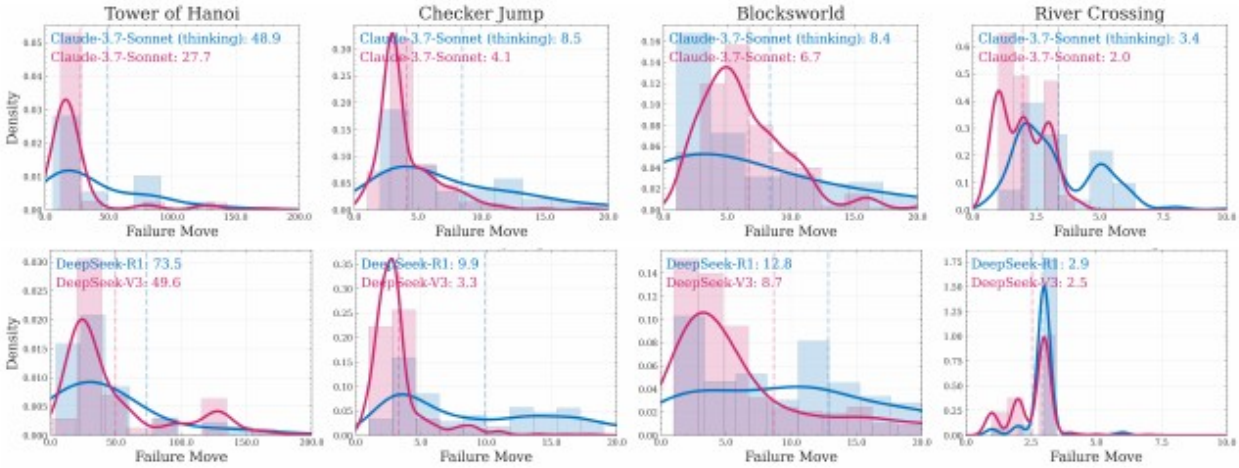
بينما يمكن لهذه النماذج الوصول إلى أعماق أكبر في تسلسلات الحل في المتوسط، فإن عمليات الاستدلال الخاصة بها أكثر عدم استقرار وعرضة لأداء غير متسق.

ديناميكيات جهد الاستدلال. يوضح الشكل 13 جهد الاستدلال (مقاسًا برموز التفكير الاستدلالية) مقابل تعقيد المشكلة عبر بيئات الألغاز الخاصة بنا. تشير النقاط الخضراء إلى

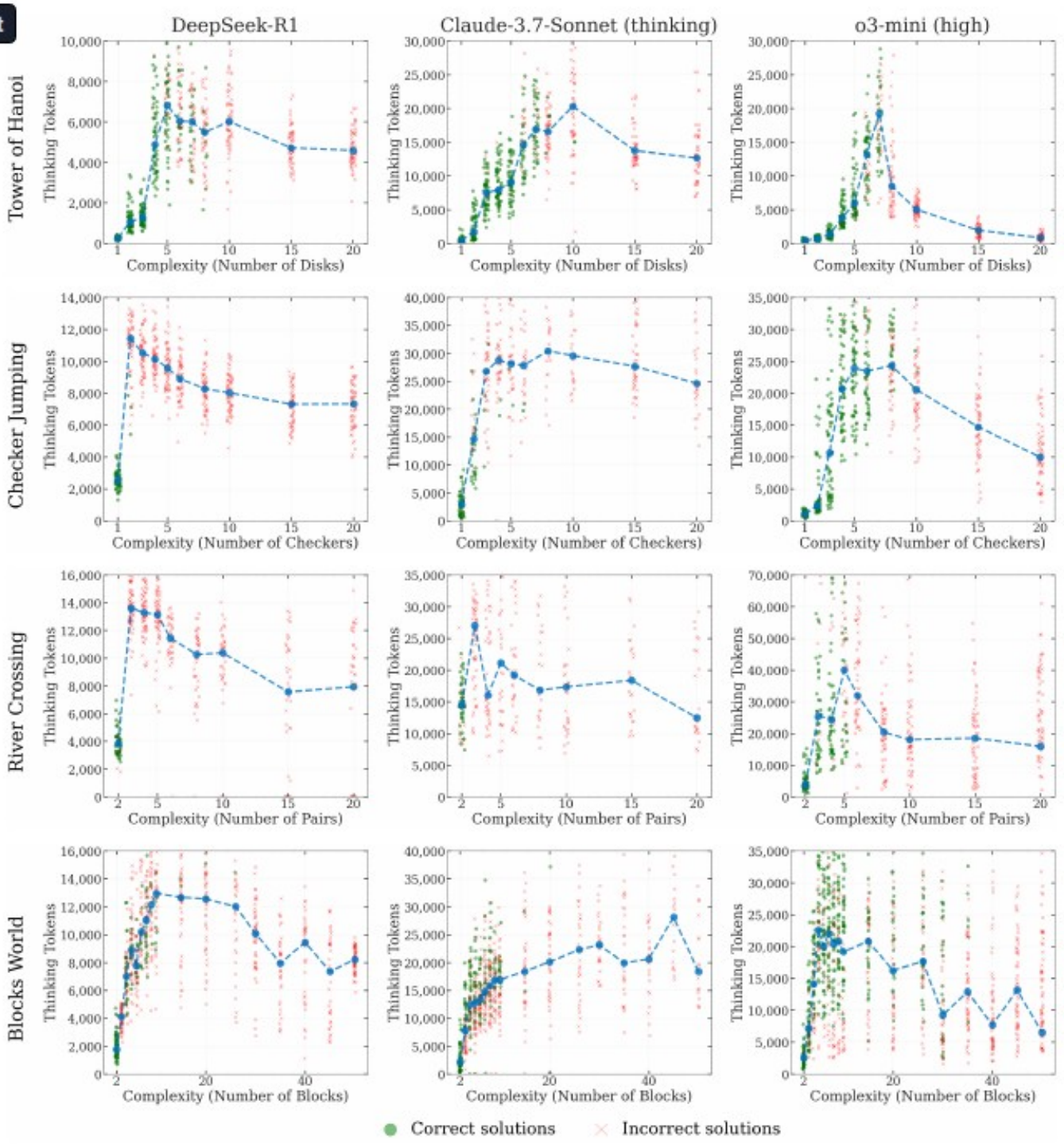


الشكل 11: مقارنة حركة الفشل الأولى مقابل تعقيد المشكلة (N) للنماذج المفكرة وغير المفكرة عبر بيئات الألغاز. أعلى: مقارنة Claude-3.7-Sonnet؛ أسفل: DeepSeek-R1 مقابل DeepSeek-V3.

الحلول الصحيحة، وتُظهر العلامات الحمراء الحلول غير الصحيحة، وتتتبع الخطوط الزرقاء متوسط استخدام رموز التفكير عند كل مستوى تعقيد (N) عبر الألغاز المختلفة ونماذج الاستدلال الكبيرة (LRMs). نلاحظ نمطًا ثابتًا عبر جميع نماذج الاستدلال الثلاثة (DeepSeek-R1، و Claude-3.7-Sonnet-thinking، و o3-mini) حيث يتوسع استخدام رموز التفكير، أي جهد الاستدلال، في البداية مع تعقيد المشكلة ولكنه ينخفض بشكل غير بديهي بعد الوصول إلى عتبة خاصة بكل نموذج. يشير هذا إلى وجود حد توسع مثير للاهتمام وأساسي في عملية تفكير نماذج الاستدلال الكبيرة (LRM) للاستدلال حيث أنه بعد تجاوز عتبات تعقيد معينة، لا تفشل النماذج فقط في حل المشكلات ولكنها تقلل بشكل غير بديهي من حوسبة الاستدلال الخاصة بها على الرغم من مواجهة مشكلات أكثر صعوبة وكونها أقل بكثير من حدود السياق والتوليد.



الشكل 12: توزيع كثافة حركات الفشل الأولى للنماذج المفكرة وغير المفكرة عبر بيئات الألغاز. أعلى: مقارنة Claude-3.7-Sonnet؛ أسفل: DeepSeek-R1 مقابل DeepSeek-V3.



الشكل 13: نتائج مفصلة حول جهد الاستدلال (مقاسًا برموز التفكير الاستدلالية) مقابل تعقيد المشكلة (N) لثلاثة نماذج استدلال كبيرة (LRMs) (DeepSeek-R1، و Claude-3.7-Sonnet مع التفكير، و o3-mini) عبر أربع بيئات ألغاز.