DeepSeek-R1: Förderung des logischen Denkens in LL.M.-Studiengängen durch Bestärkendes Lernen

DeepSeek-KI

research@deepseek.com

Abstrakt

Wir stellen unsere Reasoning-Modelle der ersten Generation vor: DeepSeek-R1-Zero und DeepSeek-R1. DeepSeek-R1-Zero, ein Modell, das durch groß angelegtes bestärkendes Lernen (RL) ohne überwachte Feinabstimmung (SFT) als vorbereitenden Schritt trainiert wurde, weist bemerkenswerte Denkfähigkeiten auf. Durch RL entwickelt DeepSeek-R1-Zero auf natürliche Weise zahlreiche leistungsstarke und faszinierende Denkverhalten. Es stößt jedoch auf Herausforderungen wie schlechte Lesbarkeit und Sprachmischung . Um diese Probleme zu lösen und die Denkleistung weiter zu verbessern, führen wir DeepSeek-R1 ein, das mehrstufiges Training und Kaltstartdaten vor RL enthält. DeepSeek - R1 erreicht bei Denkaufgaben eine mit OpenAI-o1-1217 vergleichbare Leistung. Um die Forschungsgemeinschaft zu unterstützen, stellen wir DeepSeek-R1-Zero, DeepSeek-R1 und sechs dichte Modelle (1,5 B, 7 B, 8 B, 14 B, 32 B, 70 B), die aus DeepSeek-R1 basierend auf Qwen und Llama destilliert wurden, als Open Source zur Verfügung.

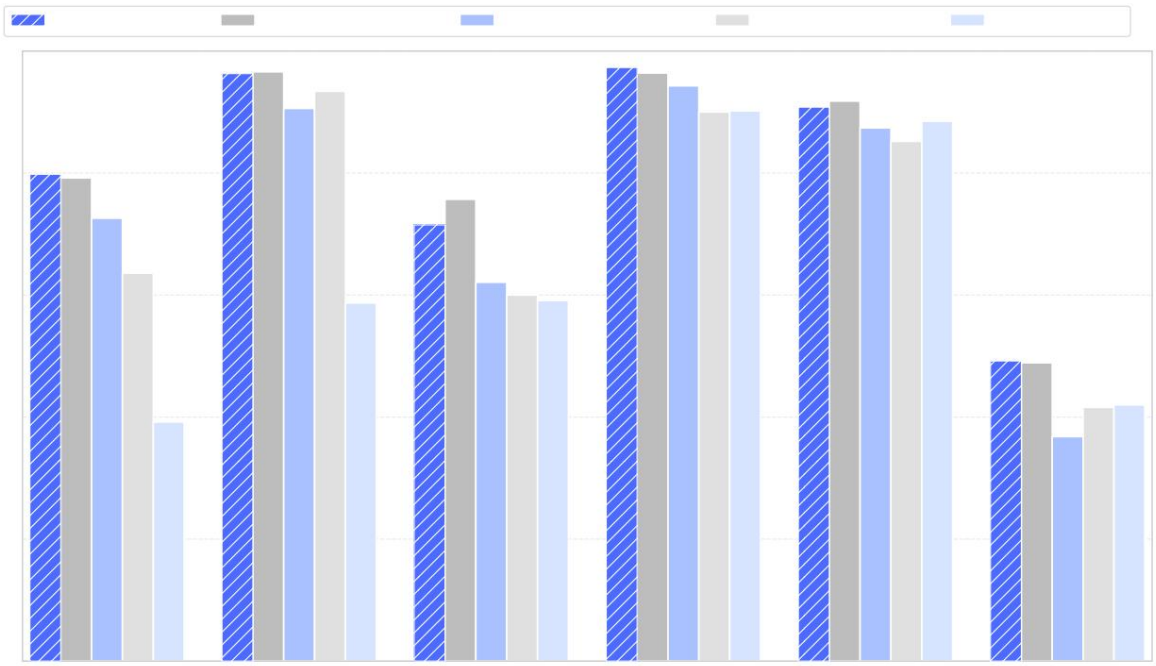

Abbildung 1 | Benchmark-Leistung von DeepSeek-R1.

Inhalt

1 Einleitung **3**

1.1 Beiträge 4

1.2 Zusammenfassung der Evaluierungsergebnisse 4

2 Vorgehensweise **5**

2.1 Übersicht 5

2.2 DeepSeek-R1-Zero: Reinforcement Learning auf dem Basismodell. 5

2.2.1 Algorithmus für bestärkendes Lernen 5

2.2.2 Belohnungsmodellierung 6

2.2.3 Trainingsvorlage 6

2.2.4 Leistung, Selbstentwicklungsprozess und Aha-Moment von DeepSeek-R1-Zero 6

2.3 DeepSeek-R1: Reinforcement Learning mit Kaltstart 9

2.3.1 Kaltstart 9

2.3.2 Argumentationsorientiertes bestärkendes Lernen 10

2.3.3 Ablehnungsstichprobenverfahren und überwachte Feinabstimmung 10

2.3.4 Reinforcement Learning für alle Szenarien 11

2.4 Destillation: Ermöglichen Sie kleinen Modellen die Fähigkeit zum logischen Denken. 11

3 Experiment **11**

3.1 DeepSeek-R1-Auswertung 13

3.2 Destillierte Modellbewertung 14

4 Diskussion **14**

4.1 Destillation vs. bestärkendes Lernen 14

4.2 Erfolglose Versuche 15

5 Schlussfolgerung, Einschränkungen und zukünftige Arbeiten **16**

A Beiträge und Danksagungen **20**

1. Einleitung

In den letzten Jahren haben Large Language Models (LLMs) eine schnelle Iteration und Entwicklung durchlaufen (Anthropic, 2024; Google, 2024; OpenAI, 2024a), wodurch die Lücke zur künstlichen allgemeinen Intelligenz (Artificial General Intelligence, AGI) zunehmend geschlossen wurde.

In jüngster Zeit hat sich die Nachschulung als wichtiger Bestandteil des gesamten Schulungsablaufs herausgestellt. Es hat sich gezeigt, dass es die Genauigkeit bei Denkaufgaben verbessert, sich an soziale Werte anpasst und sich an Benutzerpräferenzen anpasst, und das alles bei einem im Vergleich zum Vortraining relativ geringen Bedarf an Rechenressourcen. Im Kontext der Denkfähigkeiten waren die Modelle der o1-Reihe (OpenAI, 2024b) von OpenAI die ersten, die eine Skalierung der Inferenzzeit einführten, indem sie die Länge des Chain-of- Think-Prozesses verlängerten. Dieser Ansatz hat bei verschiedenen Denkaufgaben, wie Mathematik, Codierung und wissenschaftlichem Denken, erhebliche Verbesserungen erzielt. Die Herausforderung einer effektiven Skalierung der Testzeit bleibt jedoch für die Forschungsgemeinschaft eine offene Frage. Mehrere frühere Arbeiten haben verschiedene Ansätze untersucht, darunter prozessbasierte Belohnungsmodelle (Lightman et al., 2023; Uesato et al., 2022; Wang et al., 2023), bestärkendes Lernen (Kumar et al., 2024) und Suchalgorithmen wie Monte Carlo Tree Search und Beam Search (Feng et al., 2024; Trinh et al., 2024; Xin et al., 2024). Keine dieser Methoden hat jedoch eine allgemeine Argumentationsleistung erreicht, die mit den Modellen der o1-Reihe von OpenAI vergleichbar wäre.

In diesem Artikel machen wir den ersten Schritt zur Verbesserung der Fähigkeiten von Sprachmodellen zum logischen Denken durch reines bestärkendes Lernen (RL). Unser Ziel ist es, das Potenzial von LLMs zu erkunden, um ohne überwachte Daten logisches Denken zu entwickeln, wobei wir uns auf ihre Selbstentwicklung durch einen reinen RL-Prozess konzentrieren. Insbesondere verwenden wir DeepSeek-V3-Base als Basismodell und verwenden GRPO (Shao et al., 2024) als RL-Framework, um die Modellleistung beim logischen Denken zu verbessern. Während des Trainings entwickelte DeepSeek-R1-Zero auf natürliche Weise zahlreiche leistungsstarke und interessante Denkverhalten. Nach Tausenden von RL-Schritten zeigt DeepSeek-R1-Zero bei Denkbenchmarks eine hervorragende Leistung. Beispielsweise steigt der Pass@1-Score bei AIME 2024 von 15,6 % auf 71,0 %, und mit Mehrheitswahl verbessert sich der Score weiter auf 86,7 %, was der Leistung von OpenAI-o1-0912 entspricht.

DeepSeek-R1-Zero stößt jedoch auf Herausforderungen wie schlechte Lesbarkeit und Sprachmischung . Um diese Probleme zu lösen und die Argumentationsleistung weiter zu verbessern, führen wir DeepSeek-R1 ein, das eine kleine Menge Kaltstartdaten und eine mehrstufige Trainingspipeline enthält . Konkret beginnen wir mit der Erfassung von Tausenden von Kaltstartdaten, um das DeepSeek-V3-Base-Modell zu optimieren . Anschließend führen wir argumentationsorientiertes RL wie DeepSeek-R1- Zero durch. Wenn wir uns der Konvergenz im RL-Prozess nähern, erstellen wir neue SFT-Daten durch Ablehnungsstichproben am RL-Kontrollpunkt, kombiniert mit überwachten Daten von DeepSeek-V3 in Bereichen wie Schreiben, faktische Qualitätssicherung und Selbsterkenntnis, und trainieren dann das DeepSeek-V3-Base-Modell neu.
Nach der Feinabstimmung mit den neuen Daten durchläuft der Checkpoint einen zusätzlichen RL-Prozess, bei dem Eingabeaufforderungen aus allen Szenarien berücksichtigt werden. Nach diesen Schritten haben wir einen Checkpoint namens DeepSeek-R1 erhalten, der eine Leistung erreicht, die mit OpenAI-o1-1217 vergleichbar ist.

Wir untersuchen die Destillation von DeepSeek-R1 zu kleineren dichten Modellen weiter. Wenn wir Qwen2.5-32B (Qwen, 2024b) als Basismodell verwenden, ist die direkte Destillation von DeepSeek-R1 besser als die Anwendung von RL darauf. Dies zeigt, dass die von größeren Basismodellen entdeckten Denkmuster für die Verbesserung der Denkfähigkeiten entscheidend sind. Wir stellen die destillierten Serien Qwen und Llama (Dubey et al., 2024) als Open Source zur Verfügung. Insbesondere übertrifft unser destilliertes 14B-Modell das hochmoderne Open-Source-Modell QwQ-32B-Preview (Qwen, 2024a) bei weitem, und die destillierten Modelle 32B und 70B stellen einen neuen Rekord bei den Denk-Benchmarks unter den dichten Modellen auf.

1.1. Beiträge

Nach dem Training: Verstärkungslernen im großen Maßstab anhand des Basismodells

- Wir wenden RL direkt auf das Basismodell an, ohne uns im Vorfeld auf eine überwachte Feinabstimmung (SFT) zu verlassen . Dieser Ansatz ermöglicht es dem Modell, Gedankenketten (CoT) zur Lösung komplexer Probleme zu untersuchen, was zur Entwicklung von DeepSeek-R1-Zero führte. DeepSeek- R1-Zero weist Fähigkeiten wie Selbstverifizierung, Reflexion und die Generierung langer CoTs auf und markiert damit einen bedeutenden Meilenstein für die Forschungsgemeinschaft. Insbesondere handelt es sich um die erste offene Forschung, die bestätigt, dass die Denkfähigkeiten von LLMs rein durch RL gefördert werden können, ohne dass SFT erforderlich ist. Dieser Durchbruch ebnet den Weg für zukünftige Fortschritte in diesem Bereich.

- Wir stellen unsere Pipeline zur Entwicklung von DeepSeek-R1 vor. Die Pipeline umfasst zwei RL -Phasen, die darauf abzielen, verbesserte Denkmuster zu entdecken und sich an menschliche Vorlieben anzupassen , sowie zwei SFT-Phasen, die als Ausgangspunkt für die Denk- und Nicht-Denkfähigkeiten des Modells dienen. Wir glauben, dass die Pipeline der Branche durch die Schaffung besserer Modelle zugute kommen wird.

Destillation: Auch kleinere Modelle können leistungsstark sein

- Wir zeigen, dass die Denkmuster größerer Modelle auf kleinere Modelle heruntergebrochen werden können, was zu einer besseren Leistung führt als die Denkmuster, die durch RL bei kleinen Modellen entdeckt werden. Die Open Source DeepSeek-R1 sowie ihre API werden der Forschungsgemeinschaft dabei helfen, in Zukunft bessere kleinere Modelle herunterzubrechen.
- Mithilfe der von DeepSeek-R1 generierten Argumentationsdaten haben wir mehrere dichte Modelle optimiert , die in der Forschungsgemeinschaft weit verbreitet sind. Die Auswertungsergebnisse zeigen, dass die destillierten kleineren dichten Modelle bei Benchmarks außergewöhnlich gut abschneiden. DeepSeek - R1-Distill-Qwen-7B erreicht 55,5 % bei AIME 2024 und übertrifft damit QwQ-32B-Preview. Darüber hinaus erreicht DeepSeek-R1-Distill-Qwen-32B 72,6 % bei AIME 2024, 94,3 % bei MATH-500 und 57,2 % bei LiveCodeBench. Diese Ergebnisse übertreffen vorherige Open-Source-Modelle erheblich und sind mit o1-mini vergleichbar. Wir haben 1,5B-, 7B-, 8B-, 14B-, 32B- und 70B-Checkpoints basierend auf den Serien Qwen2.5 und Llama3 als Open Source für die Community bereitgestellt .

1.2. Zusammenfassung der Evaluierungsergebnisse

- **Reasoning-Aufgaben:** (1) DeepSeek-R1 erreicht bei AIME 2024 eine Punktzahl von 79,8 % Pass@1 und übertrifft damit OpenAI-o1-1217 knapp. Bei MATH-500 erreicht es eine beeindruckende Punktzahl von 97,3 %, ist damit gleichauf mit OpenAI-o1-1217 und übertrifft andere Modelle deutlich. (2) Bei codierungsbezogenen Aufgaben zeigt DeepSeek-R1 Expertenniveau bei Code-Wettbewerbsaufgaben, da es bei Codeforces eine Elo-Bewertung von 2.029 erreicht und damit 96,3 % der menschlichen Teilnehmer des Wettbewerbs übertrifft. Bei ingenieursbezogenen Aufgaben schneidet DeepSeek-R1 etwas besser ab als DeepSeek-V3, was Entwicklern bei Aufgaben in der realen Welt helfen könnte.
- **Wissen:** Bei Benchmarks wie MMLU, MMLU-Pro und GPQA Diamond erzielt DeepSeek-R1 hervorragende Ergebnisse und übertrifft DeepSeek-V3 deutlich mit Wertungen von 90,8 % bei MMLU, 84,0 % bei MMLU-Pro und 71,5 % bei GPQA Diamond. Während seine Leistung bei diesen Benchmarks leicht unter der von OpenAI-o1-1217 liegt, übertrifft DeepSeek-R1 andere Closed-Source-Modelle und demonstriert damit seinen Wettbewerbsvorteil bei Bildungsaufgaben . Beim faktischen Benchmark SimpleQA übertrifft DeepSeek-R1 DeepSeek-V3 und demonstriert damit seine Fähigkeit, faktenbasierte Abfragen zu verarbeiten. Ein ähnlicher Trend ist zu beobachten, wo OpenAI-o1 bei diesem Benchmark 4o übertrifft.

- **Sonstiges:** DeepSeek-R1 eignet sich auch hervorragend für eine Vielzahl von Aufgaben, darunter kreatives Schreiben, allgemeine Fragen beantworten, bearbeiten, zusammenfassen und mehr. Es erreicht eine beeindruckende Längenkontrollierte Gewinnrate von 87,6 % bei AlpacaEval 2.0 und eine Gewinnrate von 92,3 % bei ArenaHard , was seine starke Fähigkeit zeigt, nicht prüfungsorientierte Abfragen intelligent zu verarbeiten. Darüber hinaus zeigt DeepSeek-R1 eine hervorragende Leistung bei Aufgaben, die erfordern Verständnis von langen Kontexten, übertrifft DeepSeek-V3 bei langen Kontexten deutlich Benchmarks.

2. Ansatz

2.1. Übersicht

Frühere Arbeiten stützten sich in hohem Maße auf große Mengen überwachter Daten, um das Modell zu verbessern Leistung. In dieser Studie zeigen wir, dass die Denkfähigkeit deutlich verbessert werden kann verbessert durch groß angelegtes Reinforcement Learning (RL), auch ohne Einsatz von überwachten Feinabstimmung (SFT) als Kaltstart. Darüber hinaus kann die Leistung weiter gesteigert werden mit die Einbeziehung einer kleinen Menge von Kaltstartdaten. In den folgenden Abschnitten präsentieren wir: (1) DeepSeek-R1-Zero, das RL direkt auf das Basismodell anwendet, ohne SFT-Daten, und (2) DeepSeek-R1, das RL ausgehend von einem Checkpoint anwendet, der mit Tausenden von lange Chain-of-Thought (CoT) Beispiele. 3) Destillieren Sie die Argumentationsfähigkeit von DeepSeek-R1 auf kleine, dichte Modelle.

2.2. DeepSeek-R1-Zero: Reinforcement Learning auf dem Basismodell

Das bestärkende Lernen hat sich bei Denkaufgaben als sehr effektiv erwiesen, wie unsere früheren Arbeiten belegen (Shao et al., 2024; Wang et al., 2023). Diese Arbeiten stark von überwachten Daten abhängig, deren Erfassung zeitintensiv ist. In diesem Abschnitt das Potenzial von LLMs zur Entwicklung von Denkfähigkeiten **ohne überwachte Daten zu erkunden,** Fokussierung auf ihre Selbstentwicklung durch einen reinen Verstärkungslernprozess. Wir beginnen mit einem kurzer Überblick über unseren RL-Algorithmus, gefolgt von der Präsentation einiger spannender Ergebnisse und Ich hoffe, dies bietet der Community wertvolle Erkenntnisse.

2.2.1. Algorithmus für bestärkendes Lernen

Gruppenrelative Richtlinienoptimierung Um die Trainingskosten von RL zu sparen, verwenden wir Gruppenrelative Richtlinienoptimierung. Relative Policy Optimization (GRPO) (Shao et al., 2024), die auf das Kritikermodell verzichtet, das hat normalerweise dieselbe Größe wie das Richtlinienmodell und schätzt die Basislinie stattdessen aus Gruppenwerten. Konkret wählt GRPO für jede Frage eine Gruppe von Ausgaben { 1, 2, · · · } aus, dem alten Richtlinie und optimiert dann das Richtlinienmodell, indem das folgende Ziel maximiert wird:

$$J(\theta) = E[\ddot{y}(), \{\}]_{=1} \quad (|)]$$

$$\frac{1}{\ddot{y}\ddot{y}}_{=1} \text{Min} \frac{(|)}{(|)}, \text{Clip} \frac{(|)}{(|)}, 1\ddot{y}, 1+ \quad \ddot{y}\,\mathbf{D}\,\| \quad , \tag{1}$$

$$\mathbf{G}\,\| \quad = \frac{(|)}{(|)}_{\ddot{y}\ \text{Protokoll}} \frac{(|)}{(|)} \ddot{y}\,1, \tag{2}$$

wobei und Hyperparameter sind, und der Vorteil ist, berechnet mit einer Gruppe von Belohnungen { 1, 2, ... , } entsprechend den Ausgaben innerhalb jeder Gruppe:

$$= \frac{\ddot{y}\ m(\{1, 2, \cdots, \})}{s(\{1, 2, \cdots, \})}. \tag{3}$$

Ein Gespräch zwischen Benutzer und Assistent. Der Benutzer stellt eine Frage und der Assistent beantwortet sie. Der Assistent denkt zunächst über den Denkprozess im Kopf nach und liefert dem Benutzer dann die Antwort. Der Denkprozess und die Antwort sind in die Tags <think> </think> bzw. <answer> </answer> eingeschlossen, d. h. <think> Denkprozess hier </think> <answer> Antwort hier </answer>. Benutzer: Eingabeaufforderung. Assistent:

Tabelle 1 | Vorlage für DeepSeek-R1-Zero. Die Eingabeaufforderung wird während des Trainings durch die spezifische Frage zum logischen Denken ersetzt.

2.2.2. Belohnungsmodellierung

Die Belohnung ist die Quelle des Trainingssignals, das die Optimierungsrichtung von RL bestimmt.

Um DeepSeek-R1-Zero zu trainieren, verwenden wir ein regelbasiertes Belohnungssystem, das hauptsächlich aus zwei Arten von Belohnungen besteht:

- **Genauigkeitsbelohnungen:** Das Genauigkeitsbelohnungsmodell bewertet, ob die Antwort richtig ist. Bei mathematischen Problemen mit deterministischen Ergebnissen muss das Modell beispielsweise die endgültige Antwort in einem bestimmten Format (z. B. in einem Kästchen) bereitstellen, um eine zuverlässige regelbasierte Überprüfung der Richtigkeit zu ermöglichen. Ebenso kann bei LeetCode-Problemen ein Compiler verwendet werden, um Feedback basierend auf vordefinierten Testfällen zu generieren. •
- **Formatbelohnungen:** Zusätzlich zum Genauigkeitsbelohnungsmodell verwenden wir ein Formatbelohnungsmodell, das das Modell dazu zwingt, seinen Denkprozess zwischen den Tags „<think>“ und „</think>“ zu platzieren.

Bei der Entwicklung von DeepSeek-R1-Zero wenden wir weder das Ergebnis- noch das Prozess-Neuralbelohnungsmodell an, da wir festgestellt haben, dass das Neuralbelohnungsmodell im groß angelegten Verstärkungslernprozess unter Belohnungshacking leiden kann und das erneute Trainieren des Belohnungsmodells zusätzliche Trainingsressourcen erfordert und die gesamte Trainingspipeline komplizierter macht.

2.2.3. Trainingsvorlage

Um DeepSeek-R1-Zero zu trainieren, entwerfen wir zunächst eine einfache Vorlage, die das Basismodell anleitet, unsere vorgegebenen Anweisungen zu befolgen. Wie in Tabelle 1 dargestellt, erfordert diese Vorlage, dass DeepSeek-R1-Zero zuerst einen Denkprozess und dann die endgültige Antwort erstellt.

Wir beschränken unsere Einschränkungen bewusst auf dieses strukturelle Format und vermeiden inhaltsspezifische Verzerrungen – wie etwa die Vorgabe reflektierenden Denkens oder die Förderung bestimmter Problemlösungsstrategien – um sicherzustellen, dass wir den natürlichen Verlauf des Modells während des RL- Prozesses genau beobachten können.

2.2.4. Leistung, Selbstentwicklungsprozess und Aha-Moment von DeepSeek-R1-Zero

Leistung von DeepSeek-R1-Zero Abbildung 2 zeigt die Leistungsentwicklung von DeepSeek- R1-Zero im AIME 2024-Benchmark während des RL-Trainingsprozesses. Wie dargestellt, zeigt DeepSeek-R1-Zero eine stetige und konstante Leistungssteigerung im Verlauf des RL-Trainings. Insbesondere der durchschnittliche Pass@1-Score bei AIME 2024 zeigt einen deutlichen Anstieg von anfänglich 15,6 % auf beeindruckende 71,0 % und erreicht damit ein Leistungsniveau, das mit OpenAI-o1-0912 vergleichbar ist. Diese deutliche Verbesserung unterstreicht die Wirksamkeit unseres RL- Algorithmus bei der Optimierung der Leistung des Modells im Laufe der Zeit.

Tabelle 2 bietet eine vergleichende Analyse zwischen DeepSeek-R1-Zero und OpenAIs o1-0912- Modellen anhand einer Vielzahl von Benchmarks im Zusammenhang mit logischem Denken. Die Ergebnisse zeigen, dass RL

Modell	AIME 2024		MATH-500	GPQA LiveCode Diamantbank		CodeForces
	bestanden@1	Nachteile@64	Pass@1	Pass@1	Pass@1	Bewertung
OpenAI-o1-mini	63,6	80,0	90,0	60,0	53,8	1820
OpenAI-o1-0912	74,4	83,3	94,8	77,3	63,4	1843
DeepSeek-R1-Zero 71.0		86,7	95,9	73,3	50,0	1444

Tabelle 2 | Vergleich der Modelle DeepSeek-R1-Zero und OpenAI o1 hinsichtlich des logischen Denkens Benchmarks.

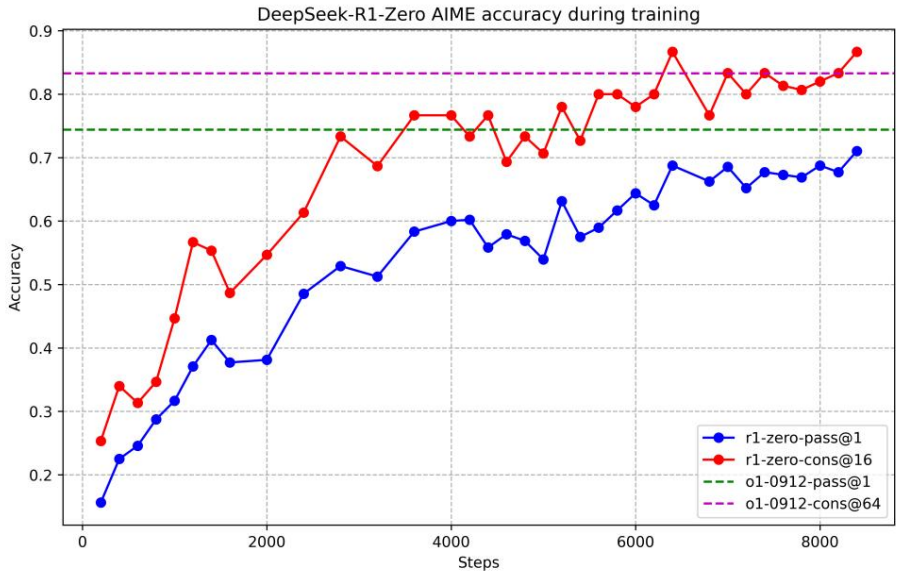

Abbildung 2 | AIME-Genauigkeit von DeepSeek-R1-Zero während des Trainings. Für jede Frage nehmen wir 16 Antworten und berechnen Sie die durchschnittliche Gesamtgenauigkeit, um eine stabile Auswertung zu gewährleisten.

DeepSeek-R1-Zero zur Erlangung robuster Argumentationsfähigkeiten ohne die Notwendigkeit einer überwachten Feinabstimmung der Daten. Dies ist eine bemerkenswerte Leistung, da es die Fähigkeit des Modells unterstreicht, Lernen und Generalisieren effektiv nur durch RL. Darüber hinaus kann die Leistung von DeepSeek- R1-Zero durch die Anwendung von Mehrheitswahl weiter gesteigert werden. Zum Beispiel: Bei der Mehrheitswahl im AIME-Benchmark ist die Leistung von DeepSeek-R1-Zero von 71,0% auf 86,7% und übertrifft damit die Leistung von OpenAI-o1-0912. Die Fähigkeit von DeepSeek-R1-Zerkleinerer, eine solche wettbewerbsfähige Leistung zu erreichen, sowohl mit als auch ohne Mehrheitswahl, unterstreicht seine starken grundlegenden Fähigkeiten und sein Potenzial für weitere Fortschritte bei Denkaufgaben.

Selbstentwicklungsprozess von DeepSeek-R1-Zero Der Selbstentwicklungsprozess von DeepSeek-R1-Zero ist eine faszinierende Demonstration, wie RL ein Modell dazu bringen kann, seine Denkfähigkeiten zu verbessern autonom. Indem wir RL direkt vom Basismodell aus initiieren, können wir die Leistung des Modells genau überwachen. Fortschritt ohne den Einfluss der überwachten Feinabstimmungsphase. Dieser Ansatz bietet eine klare Vorstellung davon, wie sich das Modell im Laufe der Zeit entwickelt, insbesondere im Hinblick auf seine Fähigkeit, komplexe Denkaufgaben.

Wie in Abbildung 3 dargestellt, zeigt die Denkzeit von DeepSeek-R1-Zero eine konsistente Verbesserung.

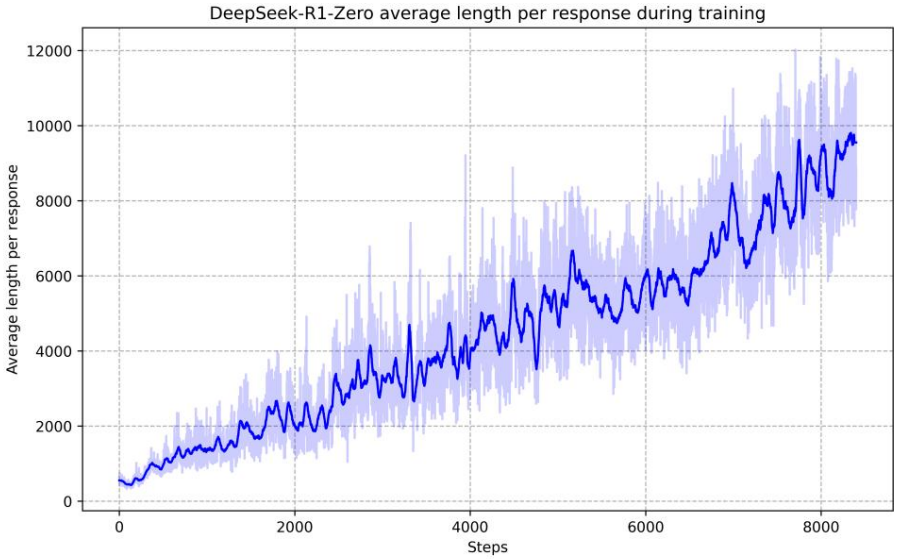

Abbildung 3 | Die durchschnittliche Antwortlänge von DeepSeek-R1-Zero im Trainingsset während des RL-Prozesses. DeepSeek-R1-Zero lernt auf natürliche Weise, Denkaufgaben mit mehr Denkzeit zu lösen.

ment während des gesamten Trainingsvorgangs. Diese Verbesserung ist nicht das Ergebnis externer Anpassungen , sondern vielmehr eine intrinsische Entwicklung innerhalb des Modells. DeepSeek-R1-Zero erlangt auf natürliche Weise die Fähigkeit, zunehmend komplexere Denkaufgaben zu lösen, indem es erweiterte Testzeitberechnungen nutzt . Diese Berechnung reicht von der Generierung von Hunderten bis zu Tausenden von Denktoken, wodurch das Modell seine Denkprozesse eingehender untersuchen und verfeinern kann.

Einer der bemerkenswertesten Aspekte dieser Selbstentwicklung ist die Entstehung ausgefeilter Verhaltensweisen, wenn die Testzeit zunimmt. Verhaltensweisen wie Reflexion – bei der das Modell seine vorherigen Schritte erneut durchgeht und neu bewertet – und die Erforschung alternativer Problemlösungsansätze entstehen spontan. Diese Verhaltensweisen sind nicht explizit programmiert, sondern entstehen als Ergebnis der Interaktion des Modells mit der Umgebung des bestärkenden Lernens. Diese spontane Entwicklung verbessert die Denkfähigkeiten von DeepSeek-R1-Zero erheblich und ermöglicht es ihm, anspruchsvollere Aufgaben effizienter und genauer zu bewältigen.

Aha-Moment von DeepSeek-R1-Zero Ein besonders faszinierendes Phänomen, das während des Trainings von DeepSeek-R1-Zero beobachtet wurde, ist das Auftreten eines „Aha-Moments“. Dieser Moment tritt, wie in Tabelle 3 dargestellt, in einer Zwischenversion des Modells auf. Während dieser Phase lernt DeepSeek-R1-Zero, einem Problem mehr Zeit zum Nachdenken zu widmen, indem es seinen anfänglichen Ansatz neu bewertet . Dieses Verhalten ist nicht nur ein Beweis für die wachsenden Denkfähigkeiten des Modells, sondern auch ein faszinierendes Beispiel dafür, wie bestärkendes Lernen zu unerwarteten und anspruchsvollen Ergebnissen führen kann .

Dieser Moment ist nicht nur ein „Aha-Moment“ für das Modell, sondern auch für die Forscher, die sein Verhalten beobachten. Er unterstreicht die Leistungsfähigkeit und Schönheit des bestärkenden Lernens: Anstatt dem Modell explizit beizubringen, wie es ein Problem lösen soll, geben wir ihm einfach die richtigen Anreize, und es entwickelt selbstständig fortgeschrittene Problemlösungsstrategien. Der „Aha-Moment“ dient als eindringliche Erinnerung an das Potenzial von bestärkendem Lernen, neue Ebenen der Intelligenz in künstlichen Systemen freizusetzen und den Weg für autonomere und anpassungsfähigere Modelle in der Zukunft zu ebnen.

Frage: Wenn > 1, dann ist die Summe der reellen Lösungen von **ÿÿ** ÿ ÿ + = gleich

Antwort: <think>

Zur Lösung der Gleichung **ÿÿ** ÿ ÿ + = , Beginnen wir mit der Quadrierung beider · · ·

ÿÿ ÿ ÿ + ^{2} = ^{2} =ÿ ÿ ÿ + = ^{2}.

Ordnen Sie es neu an, um den inneren Wurzelterm zu isolieren) ? (^{2} ÿ + =ÿ ^{2} ÿ 2 ^{2} + (^{2}) ^{2} = + =ÿ 4 ÿ 2 ^{2} ÿ + (^{2} ÿ) = 0

. . .

Warte, warte. Warte. Das ist ein Aha-Moment, den ich hier markieren kann.

Lassen Sie uns dies Schritt für Schritt erneut auswerten, um zu ermitteln, ob die richtige Summe sein kann · · · Wir begannen mit der Gleichung: **ÿÿ** ÿ ÿ + = Lassen Sie uns zunächst beide Seiten quadrieren: ÿ ÿ + = =ÿ ÿ + = ÿ Als ^{2}

Nächstes könnte ich beide Seiten erneut quadrieren und die Gleichung wie folgt behandeln: · · ·

. . .

Tabelle 3 | Ein interessanter „Aha-Moment“ einer Zwischenversion von DeepSeek-R1-Zero. Das Modell lernt, mit einem anthropomorphen Tonfall umzudenken. Dies ist auch für uns ein Aha-Moment, der uns die Kraft und Schönheit des bestärkenden Lernens erleben lässt.

Nachteile von DeepSeek-R1-Zero Obwohl DeepSeek-R1-Zero starke Denkfähigkeiten aufweist und autonom unerwartete und leistungsstarke Denkverhaltensweisen entwickelt, ist es mit mehreren Problemen konfrontiert. Beispielsweise kämpft DeepSeek-R1-Zero mit Herausforderungen wie schlechter Lesbarkeit und Sprachmischung. Um Denkprozesse lesbarer zu machen und sie mit der offenen Community zu teilen, untersuchen wir DeepSeek-R1, eine Methode, die RL mit benutzerfreundlichen Kaltstartdaten nutzt.

2.3. DeepSeek-R1: Reinforcement Learning mit Kaltstart

Inspiriert von den vielversprechenden Ergebnissen von DeepSeek-R1-Zero stellen sich zwei natürliche Fragen: 1) Kann die Argumentationsleistung weiter verbessert oder die Konvergenz beschleunigt werden, indem als Kaltstart eine kleine Menge hochwertiger Daten einbezogen wird? 2) Wie können wir ein benutzerfreundliches Modell trainieren, das nicht nur klare und kohärente Gedankenketten (CoT) erzeugt, sondern auch starke allgemeine Fähigkeiten zeigt? Um diese Fragen zu beantworten, entwickeln wir eine Pipeline zum Trainieren von DeepSeek-R1. Die Pipeline besteht aus vier Phasen, die im Folgenden beschrieben werden.

2.3.1. Kaltstart

Im Gegensatz zu DeepSeek-R1-Zero erstellen und sammeln wir für DeepSeek-R1 eine kleine Menge langer CoT-Daten, um die frühe instabile Kaltstartphase des RL-Trainings vom Basismodell aus zu verhindern und so das Modell als anfänglichen RL-Akteur zu optimieren. Um solche Daten zu sammeln, haben wir mehrere Ansätze untersucht: Wir verwenden Few-Shot-Prompting mit einem langen CoT als Beispiel, fordern Modelle direkt auf, detaillierte Antworten mit Reflexion und Verifizierung zu generieren, sammeln DeepSeek-R1- Zero-Ausgaben in einem lesbaren Format und verfeinern die Ergebnisse durch Nachbearbeitung durch Menschen Kommentatoren.

In dieser Arbeit sammeln wir Tausende von Kaltstartdaten, um die DeepSeek-V3-Base als Ausgangspunkt für RL zu optimieren. Im Vergleich zu DeepSeek-R1-Zero sind die Vorteile von Kaltstartdaten

enthalten:

- Lesbarkeit: Eine wesentliche Einschränkung von DeepSeek-R1-Zero besteht darin, dass sein Inhalt oft nicht zum Lesen geeignet ist. Antworten können mehrere Sprachen mischen oder es fehlt die Markdown-Formatierung, um Antworten für Benutzer hervorzuheben. Im Gegensatz dazu entwerfen wir beim Erstellen von Kaltstartdaten für DeepSeek-R1 ein lesbares Muster, das am Ende jeder Antwort eine Zusammenfassung enthält und Antworten herausfiltert, die nicht leserfreundlich sind. Hier definieren wir das Ausgabeformat als |special_token| <reasoning_process>|special_token|<summary>, wobei der Reasoning- Prozess das CoT für die Abfrage ist und die Zusammenfassung verwendet wird, um die Reasoning- Ergebnisse zusammenzufassen.

- Potenzial: Durch sorgfältiges Entwerfen des Musters für Kaltstartdaten mit menschlichen Vorkenntnissen beobachten wir eine bessere Leistung gegenüber DeepSeek-R1-Zero. Wir glauben, dass das iterative Training eine bessere Methode zum Begründen von Modellen ist.

2.3.2. Argumentationsorientiertes Verstärkungslernen

Nachdem wir DeepSeek-V3-Base anhand der Kaltstartdaten feinabgestimmt haben, wenden wir denselben groß angelegten Trainingsprozess für bestärkendes Lernen an wie bei DeepSeek-R1-Zero. In dieser Phase geht es darum, die Denkfähigkeiten des Modells zu verbessern, insbesondere bei schlussfolgerungsintensiven Aufgaben wie Codierung, Mathematik, Naturwissenschaften und logischem Denken, bei denen es um klar definierte Probleme mit klaren Lösungen geht. Während des Trainings beobachten wir, dass CoT häufig eine Sprachmischung aufweist, insbesondere wenn RL-Eingabeaufforderungen mehrere Sprachen umfassen. Um das Problem der Sprachmischung zu mildern, führen wir während des RL-Trainings eine Belohnung für Sprachkonsistenz ein, die als Anteil der Zielsprachenwörter im CoT berechnet wird. Obwohl Ablationsexperimente zeigen, dass eine solche Ausrichtung zu einer leichten Verschlechterung der Leistung des Modells führt, entspricht diese Belohnung den menschlichen Vorlieben und macht es lesbarer. Schließlich kombinieren wir die Genauigkeit der Denkaufgaben und die Belohnung für Sprachkonsistenz, indem wir sie direkt summieren, um die endgültige Belohnung zu bilden. Anschließend wenden wir RL-Training auf das feinabgestimmte Modell an, bis es Konvergenz bei Denkaufgaben erreicht.

2.3.3. Ablehnungsstichproben und überwachte Feinabstimmung

Wenn das logisch orientierte RL konvergiert, verwenden wir den resultierenden Kontrollpunkt, um SFT- Daten (Supervised Fine-Tuning) für die nächste Runde zu sammeln. Im Gegensatz zu den anfänglichen Kaltstartdaten, die sich hauptsächlich auf das logische Denken konzentrieren, werden in dieser Phase Daten aus anderen Bereichen einbezogen, um die Fähigkeiten des Modells beim Schreiben, Rollenspielen und anderen allgemeinen Aufgaben zu verbessern. Insbesondere generieren wir die Daten und optimieren das Modell wie unten beschrieben.

Daten zum Denken Wir kuratieren Denkanstöße und generieren Denktrajektorien, indem wir eine Ablehnungsstichprobe vom Kontrollpunkt des obigen RL-Trainings durchführen. In der vorherigen Phase haben wir nur Daten aufgenommen, die mit regelbasierten Belohnungen ausgewertet werden konnten. In dieser Phase erweitern wir den Datensatz jedoch durch die Einbeziehung zusätzlicher Daten, von denen einige ein generatives Belohnungsmodell verwenden, indem wir die Grundwahrheit und die Modellvorhersagen zur Beurteilung in DeepSeek-V3 einspeise
Da die Modellausgabe manchmal chaotisch und schwer zu lesen ist, haben wir außerdem Gedankenketten mit gemischten Sprachen, langen Absätzen und Codeblöcken herausgefiltert. Für jede Eingabeaufforderung prüfen wir mehrere Antworten und behalten nur die richtigen bei. Insgesamt sammeln wir etwa 600.000 Trainingsbeispiele zum Thema logisches Denken.

Nicht-Argumentationsdaten Für nicht-Argumentationsdaten wie Schreiben, sachliche Qualitätssicherung, Selbsterkenntnis und Übersetzung übernehmen wir die DeepSeek-V3-Pipeline und verwenden Teile des SFT-Datensatzes von DeepSeek-V3 wieder. Für bestimmte nicht-Argumentationsdaten rufen wir DeepSeek-V3 auf, um eine mögliche Gedankenkette zu generieren, bevor wir die Frage durch Eingabeaufforderung beantworten. Für einfachere Abfragen wie „Hallo“ geben wir jedoch keine CoT als Antwort. Am Ende haben wir insgesamt etwa 200.000 Trainingsbeispiele gesammelt, die nichts mit Argumentation zu tun haben.

Wir optimieren DeepSeek-V3-Base für zwei Epochen mithilfe des oben kuratierten Datensatzes mit etwa 800.000 Samples.

2.3.4. Reinforcement Learning für alle Szenarien

Um das Modell noch besser an menschliche Vorlieben anzupassen, implementieren wir eine sekundäre Phase des Verstärkungslernens, die darauf abzielt, die Nützlichkeit und Harmlosigkeit des Modells zu verbessern und gleichzeitig seine Denkfähigkeiten zu verfeinern. Konkret trainieren wir das Modell mithilfe einer Kombination aus Belohnungssignalen und verschiedenen Eingabeaufforderungsverteilungen. Für die Denkdaten halten wir uns an die in DeepSeek-R1-Zero beschriebene Methodik, die regelbasierte Belohnungen verwendet, um den Lernprozess in den Bereichen Mathematik, Code und logisches Denken zu steuern. Für allgemeine Daten greifen wir auf Belohnungsmodelle zurück, um menschliche Vorlieben in komplexen und nuancierten Szenarien zu erfassen. Wir bauen auf der DeepSeek-V3-Pipeline auf und übernehmen eine ähnliche Verteilung von Präferenzpaaren und Trainingsaufforderungen . Für die Nützlichkeit konzentrieren wir uns ausschließlich auf die abschließende Zusammenfassung und stellen sicher, dass die Bewertung den Nutzen und die Relevanz der Antwort für den Benutzer hervorhebt und gleichzeitig die Beeinträchtigung des zugrunde liegenden Denkprozesses minimiert. Um die Harmlosigkeit zu ermitteln, bewerten wir die gesamte Antwort des Modells, einschließlich des Denkprozesses und der Zusammenfassung, um mögliche Risiken, Verzerrungen oder schädliche Inhalte, die während des Generierungsprozesses auftreten können, zu identifizieren und zu mindern . Letztendlich ermöglicht uns die Integration von Belohnungssignalen und vielfältigen Datenverteilungen, ein Modell zu trainieren, das sich durch hervorragende Denkfähigkeiten auszeichnet und gleichzeitig Nützlichkeit und Harmlosigkeit priorisiert.

2.4. Destillation: Kleine Modelle mit Schlussfolgerungsfähigkeiten ausstatten

Um effizientere kleinere Modelle mit Denkfähigkeiten wie DeepSeek-R1 auszustatten, haben wir Open-Source-Modelle wie Qwen (Qwen, 2024b) und Llama (AI@Meta, 2024) direkt feinabgestimmt, indem wir die 800.000 mit DeepSeek-R1 kuratierten Beispiele verwendet haben, wie in §2.3.3 beschrieben. Unsere Ergebnisse zeigen, dass diese unkomplizierte Destillationsmethode die Denkfähigkeiten kleinerer Modelle deutlich verbessert. Die Basismodelle, die wir hier verwenden, sind Qwen2.5-Math-1.5B, Qwen2.5-Math-7B, Qwen2.5-14B , Qwen2.5-32B, Llama-3.1-8B und Llama-3.3-70B-Instruct. Wir haben Llama-3.3 ausgewählt, da seine Denkfähigkeit etwas besser ist als die von Llama-3.1.

Für destillierte Modelle wenden wir nur SFT an und schließen keine RL-Phase ein, obwohl die Einbeziehung von RL die Modellleistung erheblich steigern könnte. Unser Hauptziel besteht hier darin, die Wirksamkeit der Destillationstechnik zu demonstrieren und die Erforschung der RL- Phase der breiteren Forschungsgemeinschaft zu überlassen.

3. Experiment

Benchmarks Wir evaluieren Modelle auf MMLU (Hendrycks et al., 2020), MMLU-Redux (Gema et al., 2024), MMLU-Pro (Wang et al., 2024), C-Eval (Huang et al., 2023), und CMMLU (Li et al., 2023), IFEval (Zhou et al., 2023), FRAMES (Krishna et al., 2024), GPQA Diamond (Rein et al., 2023), SimpleQA (OpenAI, 2024c), C-SimpleQA (He et al., 2024), SWE-Bench Verified (OpenAI,

2024d), Aider [1], LiveCodeBench (Jain et al., 2024) (08.2024 – 01.2025), Codeforces [2], chinesisch National High School Mathematics Olympiad (CNMO 2024)3 und American Invitational Mathematics Examination 2024 (AIME 2024) (MAA, 2024). Zusätzlich zu Standard-Benchmarks bewerten wir unsere Modelle auch anhand von offenen Generierungsaufgaben mit LLMs als Richtern. Insbesondere halten wir uns an die ursprünglichen Konfigurationen von AlpacaEval 2.0 (Dubois et al., 2024) und Arena-Hard (Li et al., 2024), die GPT-4-Turbo-1106 als Richter für paarweise Vergleiche nutzen. Hier geben wir nur die endgültige Zusammenfassung zur Bewertung weiter, um Längenverzerrungen zu vermeiden. Für destillierte Modelle berichten wir über repräsentative Ergebnisse zu AIME 2024, MATH-500, GPQA Diamond, Codeforces und LiveCodeBench.

Bewertungsaufforderungen Nach der Einrichtung in DeepSeek-V3 werden Standardbenchmarks wie MMLU, DROP, GPQA Diamond und SimpleQA mithilfe von Aufforderungen aus dem Simple -Evals-Framework bewertet. Für MMLU-Redux übernehmen wir das Zero-Eval-Aufforderungsformat (Lin, 2024) in einer Zero-Shot-Einstellung. In Bezug auf MMLU-Pro, C-Eval und CLUE-WSC ändern wir die Aufforderung leicht in die Zero-Shot-Einstellung, da die ursprünglichen Aufforderungen nur wenige Versuche umfassen. Der CoT in wenigen Versuchen kann die Leistung von DeepSeek-R1 beeinträchtigen. Andere Datensätze folgen ihren ursprünglichen Bewertungsprotokollen mit Standardaufforderungen, die von ihren Erstellern bereitgestellt werden. Für Code- und Mathematik-Benchmarks deckt der HumanEval-Mul-Datensatz acht gängige Programmiersprachen ab (Python, Java, C++, C#, JavaScript, TypeScript, PHP und Bash). Die Modellleistung auf LiveCodeBench wird im CoT-Format ausgewertet, wobei die Daten zwischen August 2024 und Januar 2025 erfasst werden. Der Codeforces -Datensatz wird anhand von Problemen aus 10 Div.2-Wettbewerben sowie von Experten erstellten Testfällen ausgewertet, wonach die erwarteten Bewertungen und Prozentsätze der Wettbewerber berechnet werden. Von SWE-Bench verifizierte Ergebnisse werden über das agentenlose Framework (Xia et al., 2024) erzielt. AIDER-bezogene Benchmarks werden in einem „Diff“-Format gemessen. Die DeepSeek-R1-Ausgaben sind auf maximal 32.768 Token für jeden Benchmark begrenzt .

Baselines Wir führen umfassende Bewertungen anhand mehrerer starker Baselines durch, darunter DeepSeek-V3, Claude-Sonnet-3.5-1022, GPT-4o-0513, OpenAI-o1-mini und OpenAI-o1-1217. Da der Zugriff auf die OpenAI-o1-1217-API auf dem chinesischen Festland eine Herausforderung darstellt, berichten wir über ihre Leistung auf Grundlage offizieller Berichte. Für destillierte Modelle vergleichen wir auch das Open-Source-Modell QwQ-32B-Preview (Qwen, 2024a).

Evaluierungs-Setup Wir haben die maximale Generierungslänge für die Modelle auf 32.768 Token festgelegt. Wir haben festgestellt, dass die Verwendung von Greedy-Decoding zur Auswertung von Long-Output-Argumentation-Modellen zu höheren Wiederholungsraten und erheblicher Variabilität zwischen verschiedenen Prüfpunkten führt. Daher verwenden wir standardmäßig die pass@-Auswertung (Chen et al., 2021) und melden pass@1 unter Verwendung einer Temperatur ungleich Null. Konkret verwenden wir eine Stichprobentemperatur von 0,6 und einen Top-Wert von 0,95, um Antworten (normalerweise zwischen 4 und 64, abhängig von der Größe des Testsatzes) für jede Frage zu generieren. Pass@1 wird dann wie folgt berechnet:

$$\text{Durchgang@1} = \frac{1}{} \ddot{y}\ddot{y}_{=1} ,$$

wobei die Richtigkeit der -ten Antwort bezeichnet. Diese Methode liefert zuverlässigere Leistungsschätzungen. Für AIME 2024 berichten wir auch Konsensergebnisse (Mehrheitsvotum) (Wang et al., 2022) unter Verwendung von 64 Proben, bezeichnet als cons@64.

1https://aider.chat
2https://codeforces.com
3https://www.cms.org.cn/Home/comp/comp/cid/12.html

3.1. DeepSeek-R1-Auswertung

	Benchmark (Metrisch)	Claude-3.5-Sonnet-1022	GPT-4o 0513	DeepSeek V3	OpenAI o1-mini	OpenAI o1-1217	DeepSeek R1
	Architektur	-	-	MoE	-	-	MoE
	# Aktivierte Parameter	-	-	37B	-	-	37B
	# Parameter gesamt	-	-	671B	-	-	671B
Englisch	MMLU (Bestanden@1)	88,3	87,2	88,5	85,2	**91,8**	90,8
	MMLU-Redux (EM)	88,9	88,0	89,1	86,7	-	**92,9**
	MMLU-Pro (EM)	78,0	72,6	75,9	80,3	-	**84,0**
	DROP (3-Schuss F1)	88,3	83,7	91,6	83,9	90,2	**92,2**
	IF-Eval (Prompt Strict)	**86,5**	84,3	86,1	84,8	-	83,3
	GPQA Diamant (Bestanden@1)	65,0	49,9	59,1	60,0	**75,7**	71,5
	SimpleQA (Richtig)	28,4	38,2	24,9	7,0	**47,0**	30.1
	RAHMEN (Zubehör)	72,5	80,5	73,3	76,9	-	**82,5**
	AlpacaEval2.0 (LC-Gewinnrate)	52,0	51,1	70,0	57,8	-	**87,6**
	ArenaHard (GPT-4-1106)	85,2	80,4	85,5	92,0	-	**92,3**
Code	LiveCodeBench (Pass@1-COT)	38,9	32,9	36.2	53,8	63,4	**65,9**
	Codeforces (Perzentil)	20.3	23,6	58,7	93,4	**96,6**	96,3
	Codeforces (Wertung)	717	759	1134	1820	**2061**	2029
	SWE verifiziert (gelöst)	**50,8**	38,8	42,0	41,6	48,9	49.2
	Aider-Polyglot (Acc.)	45.3	16,0	49,6	32,9	**61,7**	53,3
Mathe	AIME 2024 (Pass@1)	16,0	9.3	39.2	63,6	79,2	**79,8**
	MATH-500 (Bestanden@1)	78,3	74,6	90,2	90,0	96,4	**97,3**
	CNMO 2024 (Bestanden@1)	13.1	10.8	43.2	67,6	-	**78,8**
chinesisch	CLUEWSC (EM)	85,4	87,9	90,9	89,9	-	**92,8**
	C-Eval (EM)	76,7	76,0	86,5	68,9	-	**91,8**
	C-SimpleQA (Richtig)	55,4	58,7	**68,0**	40,3	-	63,7

Tabelle 4 | Vergleich zwischen DeepSeek-R1 und anderen repräsentativen Modellen.

Für bildungsorientierte Wissensbenchmarks wie MMLU, MMLU-Pro und GPQA Diamond, DeepSeek-R1 zeigt eine bessere Leistung als DeepSeek-V3. Diese Verbesserung ist hauptsächlich auf die verbesserte Genauigkeit bei STEM-bezogenen Fragen zurückzuführen, bei denen durch groß angelegtes Verstärkungslernen erhebliche Verbesserungen erzielt werden. Darüber hinaus bietet DeepSeek-R1 zeichnet sich durch seine starke Dokumentenanalyse aus, insbesondere bei FRAMES, einer langwierigen kontextabhängigen QA-Aufgabe Fähigkeiten. Dies unterstreicht das Potenzial von Argumentationsmodellen in KI-gesteuerten Such- und Daten Analyseaufgaben. Beim faktischen Benchmark SimpleQA übertrifft DeepSeek-R1 DeepSeek-V3, und demonstriert damit seine Fähigkeit, faktenbasierte Abfragen zu verarbeiten. Ein ähnlicher Trend ist zu beobachten, wo OpenAI-o1 übertrifft GPT-4o in diesem Benchmark. Allerdings schneidet DeepSeek-R1 schlechter ab als DeepSeek-V3 im chinesischen SimpleQA-Benchmark, vor allem aufgrund seiner Tendenz zur Ablehnung Beantwortung bestimmter Anfragen nach Sicherheits-RL. Ohne Sicherheits-RL könnte DeepSeek-R1 eine Genauigkeit von über 70 %.

DeepSeek-R1 liefert auch beeindruckende Ergebnisse bei IF-Eval, einem Benchmark zur Bewertung eines Fähigkeit des Modells, Formatanweisungen zu befolgen. Diese Verbesserungen können mit der Einbeziehung von Anweisungsfolgedaten während der letzten Phasen der überwachten Feinabstimmung (SFT) und RL Training. Darüber hinaus wird eine bemerkenswerte Leistung bei AlpacaEval2.0 und ArenaHard beobachtet, Dies zeigt die Stärken von DeepSeek-R1 bei Schreibaufgaben und der Beantwortung von Fragen zu offenen Themen. Die deutliche Überlegenheit von DeepSeek-V3 unterstreicht die Generalisierungsvorteile von groß angelegten RL, das nicht nur die Denkfähigkeit steigert, sondern auch die Leistung in verschiedenen Domänen. Darüber hinaus sind die von DeepSeek-R1 generierten Zusammenfassungslängen präzise, mit einer Durchschnittlich 689 Token auf ArenaHard und 2.218 Zeichen auf AlpacaEval 2.0. Dies zeigt, dass

DeepSeek-R1 vermeidet die Einführung von Längenverzerrungen bei GPT-basierten Auswertungen und festigt damit seine Robustheit gegenüber mehreren Aufgaben.

Bei mathematischen Aufgaben zeigt DeepSeek-R1 eine Leistung, die mit OpenAI-o1-1217 vergleichbar ist. übertrifft andere Modelle bei weitem. Ein ähnlicher Trend ist bei Kodierungsalgorithmen zu beobachten Aufgaben wie LiveCodeBench und Codeforces, bei denen auf Argumentation fokussierte Modelle dominieren Benchmarks. Bei ingenieurorientierten Codieraufgaben übertrifft OpenAI-o1-1217 DeepSeek-R1 auf Aider, erreicht aber eine vergleichbare Leistung auf SWE Verified. Wir glauben, dass die Technik Die Leistung von DeepSeek-R1 wird sich in der nächsten Version verbessern, da die Menge an zugehörigem RL Trainingsdaten sind derzeit noch sehr begrenzt.

3.2. Destillierte Modellbewertung

Modell	AIME 2024		MATH-500	GPQA LiveCode Diamantbank		CodeForces
	pass@1	Nachteile@64	pass@1	Pass@1	Pass@1	Bewertung
GPT-4o-0513	9.3	13.4	74,6	49,9	32,9	759
Claude-3.5-Sonnet-1022	16,0	26,7	78,3	65,0	38,9	717
OpenAI-o1-mini	63,6	80,0	90,0	60,0	53,8	**1820**
QwQ-32B-Vorschau	50,0	60,0	90,6	54,5	41,9	1316
DeepSeek-R1-Distill-Qwen-1,5B	28,9	52,7	83,9	33,8	16,9	954
DeepSeek-R1-Distill-Qwen-7B	55,5	83,3	92,8	49,1	37,6	1189
DeepSeek-R1-Distill-Qwen-14B	69,7	80,0	93,9	59,1	53,1	1481
DeepSeek-R1-Distill-Qwen-32B	**72,6**	83,3	94,3	62.1	57,2	1691
DeepSeek-R1-Distill-Llama-8B	50,4	80,0	89,1	49,0	39,6	1205
DeepSeek-R1-Distill-Llama-70B	70,0	**86,7**	**94,5**	**65,2**	**57,5**	1633

Tabelle 5 | Vergleich der destillierten Modelle von DeepSeek-R1 und anderer vergleichbarer Modelle auf Benchmarks im Zusammenhang mit dem logischen Denken.

Wie in Tabelle 5 gezeigt, ermöglicht die einfache Destillation der Ausgaben von DeepSeek-R1 dem effizienten DeepSeek- R1-7B (d. h. DeepSeek-R1-Distill-Qwen-7B, im Folgenden ähnlich abgekürzt), nicht-logisch arbeitende Modelle wie GPT-4o-0513 auf ganzer Linie zu übertreffen. DeepSeek-R1-14B übertrifft QwQ-32B- Preview in allen Bewertungsmetriken, während DeepSeek-R1-32B und DeepSeek-R1-70B deutlich übertreffen o1-mini bei den meisten Benchmarks. Diese Ergebnisse zeigen das große Potenzial der Destillation . Darüber hinaus haben wir festgestellt, dass die Anwendung von RL auf diese destillierten Modelle erhebliche weitere Gewinne. Wir glauben, dass dies eine weitere Untersuchung rechtfertigt und präsentieren daher nur die Ergebnisse der einfache SFT-destillierte Modelle hier.

4. Diskussion

4.1. Destillation vs. bestärkendes Lernen

In Abschnitt 3.2 können wir sehen, dass das kleine Modell durch die Destillation von DeepSeek-R1 beeindruckende Ergebnisse. Es bleibt jedoch eine Frage: Kann das Modell eine vergleichbare Leistung erzielen durch das im Artikel besprochene groß angelegte RL-Training ohne Destillation?

Um diese Frage zu beantworten, führen wir ein umfangreiches RL-Training auf Qwen-32B-Base durch, wobei wir Mathematik verwenden, Code und STEM-Daten, Training für über 10.000 Schritte, was zu DeepSeek-R1-Zero-Qwen-32B führte. Die Die in Tabelle 6 dargestellten experimentellen Ergebnisse zeigen, dass das 32B-Basismodell nach groß angelegter

Modell	AIME 2024		MATH-500	GPQA Diamant	LiveCodeBench
	pass@1	Nachteile@64	pass@1	Pass@1	Pass@1
QwQ-32B-Vorschau	50,0	60,0	90,6	54,5	41,9
DeepSeek-R1-Zero-Qwen-32B	47,0	60,0	91,6	55,0	40,2
DeepSeek-R1-Distill-Qwen-32B	**72,6**	**83,3**	**94,3**	**62,1**	**57,2**

Tabelle 6 | Vergleich destillierter und RL-Modelle anhand von Benchmarks im Zusammenhang mit dem logischen Denken.

RL-Training erreicht eine Leistung auf Augenhöhe mit QwQ-32B-Preview. Allerdings ist DeepSeek-R1-Distill-Qwen-32B, das aus DeepSeek-R1 destilliert wurde, bietet eine deutlich bessere Leistung als DeepSeek-R1-Zero-Qwen-32B in allen Benchmarks.

Daher können wir zwei Schlussfolgerungen ziehen: Erstens, die Destillation leistungsfähigerer Modelle in kleinere hervorragende Ergebnisse, während kleinere Modelle, die auf dem in dieses Papiers erfordert enorme Rechenleistung und erreicht möglicherweise nicht einmal die Leistung der Destillation. Zweitens sind Destillationsstrategien sowohl wirtschaftlich als auch effektiv, aber die Weiterentwicklung über die Grenzen der Intelligenz hinaus sind möglicherweise noch leistungsfähigere Basismodelle und bestärkendes Lernen im größeren Maßstab erforderlich.

4.2. Erfolglose Versuche

In den frühen Phasen der Entwicklung von DeepSeek-R1 gab es auch Fehler und Rückschläge. der Weg. Wir teilen hier unsere Erfahrungen mit dem Scheitern, um Einblicke zu geben, aber das bedeutet nicht, dass Mit diesen Ansätzen ist es nicht möglich, wirksame Denkmodelle zu entwickeln.

Process Reward Model (PRM) PRM ist eine sinnvolle Methode, um das Modell zu besseren Ansätze zur Lösung von Denkaufgaben (Lightman et al., 2023; Uesato et al., 2022; Wang et al., 2023). In der Praxis weist PRM jedoch drei Haupteinschränkungen auf, die seinen endgültigen Erfolg behindern könnten . Erstens ist es schwierig, einen feinkörnigen Schritt im allgemeinen Denken explizit zu definieren. Zweitens Zu bestimmen, ob der aktuelle Zwischenschritt korrekt ist, ist eine anspruchsvolle Aufgabe. Automatisierte Die Annotation mithilfe von Modellen kann zu unbefriedigenden Ergebnissen führen, während die manuelle Annotation nicht förderlich für die Skalierung ist. Drittens führt die Einführung eines modellbasierten PRM zwangsläufig zu Belohnungen Hacking (Gao et al., 2022), und das Umschulen des Belohnungsmodells erfordert zusätzliche Trainingsressourcen und es verkompliziert die gesamte Trainingspipeline. Zusammenfassend lässt sich sagen, dass PRM zwar eine gute Möglichkeit, die vom Modell generierten Top-N-Antworten neu zu ordnen oder bei der geführten Suche zu helfen (Snell et al., 2024), sind seine Vorteile im Vergleich zum zusätzlichen Rechenaufwand, den es führt dies während des groß angelegten bestärkenden Lernprozesses in unseren Experimenten ein.

Monte Carlo Tree Search (MCTS) Inspiriert von AlphaGo (Silver et al., 2017b) und AlphaZero (Silver et al., 2017a) haben wir die Verwendung von Monte Carlo Tree Search (MCTS) untersucht, um die Testzeit zu verbessern Skalierbarkeit der Berechnung. Bei diesem Ansatz werden Antworten in kleinere Teile zerlegt, um die Modell, um den Lösungsraum systematisch zu erkunden. Um dies zu erleichtern, fordern wir das Modell auf, mehrere Tags generieren, die den für die Suche erforderlichen spezifischen Denkschritten entsprechen. Für Beim Training verwenden wir zunächst gesammelte Eingabeaufforderungen, um Antworten über MCTS zu finden, geleitet von einem vorab trainierten Wert Modell. Anschließend verwenden wir die resultierenden Frage-Antwort-Paare, um sowohl das Akteurmodell und das Wertemodell, wobei der Prozess iterativ verfeinert wird.

Dieser Ansatz stößt jedoch bei der Ausweitung des Trainings auf mehrere Herausforderungen. Erstens: Im Gegensatz zum Schach, wo der Suchraum relativ gut definiert ist, stellt die Token-Generierung eine

exponentiell größerer Suchraum. Um dies zu beheben, legen wir für jeden Knoten eine maximale Erweiterungsgrenze fest, was jedoch dazu führen kann, dass das Modell in lokalen Optima stecken bleibt. Zweitens beeinflusst das Wertmodell direkt die Qualität der Generierung, da es jeden Schritt des Suchprozesses leitet. Das Trainieren eines feinkörnigen Wertmodells ist von Natur aus schwierig, was es für das Modell schwierig macht, es iterativ zu verbessern. Während der Kernerfolg von AlphaGo darauf beruhte, ein Wertmodell zu trainieren, um seine Leistung schrittweise zu verbessern, ist dieses Prinzip in unserem Setup aufgrund der Komplexität der Token-Generierung schwer zu replizieren .

Zusammenfassend lässt sich sagen, dass MCTS zwar in Kombination mit einem vorab trainierten Wertemodell die Leistung bei der Inferenz verbessern kann, die iterative Steigerung der Modellleistung durch Selbstsuche jedoch weiterhin eine erhebliche Herausforderung darstellt.

5. Schlussfolgerung, Einschränkungen und zukünftige Arbeiten

In dieser Arbeit teilen wir unsere Erfahrungen bei der Verbesserung der Fähigkeiten zum Modellschlussfolgern durch bestärkendes Lernen. DeepSeek-R1-Zero stellt einen reinen RL-Ansatz dar, der ohne Kaltstartdaten auskommt und bei verschiedenen Aufgaben eine starke Leistung erzielt. DeepSeek-R1 ist leistungsstärker und nutzt Kaltstartdaten neben iterativer RL-Feinabstimmung. Letztendlich erreicht DeepSeek-R1 bei einer Reihe von Aufgaben eine mit OpenAI-o1-1217 vergleichbare Leistung.

Wir untersuchen weiter die Destillation der Argumentationsfähigkeit auf kleine, dichte Modelle. Wir verwenden DeepSeek-R1 als Lehrermodell, um 800.000 Trainingsbeispiele zu generieren und mehrere kleine, dichte Modelle zu optimieren. Die Ergebnisse sind vielversprechend: DeepSeek-R1-Distill-Qwen-1.5B übertrifft GPT-4o und Claude-3.5-Sonnet bei Mathematik-Benchmarks mit 28,9 % bei AIME und 83,9 % bei MATH. Andere dichte Modelle erzielen ebenfalls beeindruckende Ergebnisse und übertreffen andere anweisungsoptimierte Modelle, die auf denselben zugrunde liegenden Prüfpunkten basieren, deutlich.

Wir planen, in Zukunft in die Forschung für DeepSeek-R1 in den folgenden Bereichen zu investieren.

- **Allgemeine Leistungsfähigkeit:** Derzeit reichen die Fähigkeiten von DeepSeek-R1 bei Aufgaben wie Funktionsaufrufen, Multi-Turn, komplexem Rollenspiel und JSON-Ausgabe nicht an die von DeepSeek-V3 heran.

  In Zukunft möchten wir untersuchen, wie lange CoT genutzt werden kann, um Aufgaben in diesen Bereichen zu verbessern.
- **Sprachenmischung:** DeepSeek-R1 ist derzeit für Chinesisch und Englisch optimiert, was bei der Verarbeitung von Abfragen in anderen Sprachen zu Problemen mit der Sprachenmischung führen kann. Beispielsweise kann DeepSeek-R1 Englisch für Argumente und Antworten verwenden, selbst wenn die Abfrage in einer anderen Sprache als Englisch oder Chinesisch erfolgt . Wir beabsichtigen, diese Einschränkung in zukünftigen Updates zu beheben .
- **Prompting Engineering:** Bei der Evaluierung von DeepSeek-R1 stellen wir fest, dass es empfindlich auf Eingabeaufforderungen reagiert. Few-Shot-Prompting verschlechtert seine Leistung kontinuierlich. Daher empfehlen wir Benutzern, das Problem direkt zu beschreiben und das Ausgabeformat mit einer Zero-Shot-Einstellung anzugeben, um optimale Ergebnisse zu erzielen.
- **Software-Engineering-Aufgaben:** Aufgrund der langen Auswertungszeiten, die die Effizienz des RL-Prozesses beeinträchtigen, wurde RL im großen Maßstab nicht umfassend bei Software- Engineering-Aufgaben angewendet. Daher hat DeepSeek-R1 bei Software-Engineering-Benchmarks keine große Verbesserung gegenüber DeepSeek-V3 gezeigt. Zukünftige Versionen werden dieses Problem angehen, indem sie Ablehnungsstichproben auf Software-Engineering-Daten implementieren oder asynchrone Auswertungen während des RL-Prozesses einbeziehen, um die Effizienz zu verbessern.

Verweise

AI@Meta. Llama 3.1-Modellkarte, 2024. URL https://github.com/meta-llama/llama-m odels/blob/main/models/llama3_1/MODEL_CARD.md.

Anthropisch. Claude 3.5 Sonett, 2024. URL https://www.anthropic.com/news/claude-3 -5-Sonett.

M. Chen, J. Tworek, H. Jun, Q. Yuan, HP de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, FP Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, WH Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, AN Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever und W. Zaremba. Evaluierung großer, mit Code trainierter Sprachmodelle. CoRR, abs/2107.03374, 2021. URL https://arxiv.org/abs/2107.03374.

A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. Die Lama-3-Modellherde. arXiv-Vorabdruck arXiv:2407.21783, 2024.

Y. Dubois, B. Galambosi, P. Liang und TB Hashimoto. Längengesteuerter Alpakaeval: Eine einfache Möglichkeit zur Entzerrung automatischer Evaluatoren. ArXiv-Vorabdruck arXiv:2404.04475, 2024.

X. Feng, Z. Wan, M. Wen, SM McAleer, Y. Wen, W. Zhang und J. Wang. Alphazero-ähnliche Baumsuche kann die Dekodierung und das Training großer Sprachmodelle leiten, 2024. URL https: //arxiv.org/abs/2309.17179.

L. Gao, J. Schulman und J. Hilton. Skalierungsgesetze für die Überoptimierung von Belohnungsmodellen, 2022. URL https://arxiv.org/abs/2210.10760.

AP Gema, JOJ Leang, G. Hong, A. Devoto, ACM Mancino, R. Saxena, X. He, Y. Zhao, X. Du, MRG Madani, C. Barale, R. McHardy, J. Harris, J. Kaddour, E. van Krieken und P. Minervini. Sind wir mit mmlu fertig? CoRR, abs/2406.04127, 2024. URL https://doi.or g/10.48550/arXiv.2406.04127.

Google. Unser Modell der nächsten Generation: Gemini 1.5, 2024. URL https://blog.google/techno logy/ai/google-gemini-modell der nächsten generation-februar-2024.

Y. He, S. Li, J. Liu, Y. Tan, W. Wang, H. Huang, X. Bu, H. Guo, C. Hu, B. Zheng, et al. Chi-nese simpleqa: Eine chinesische Faktizitätsbewertung für große Sprachmodelle. arXiv-Vorabdruck arXiv:2411.07140, 2024.

D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song und J. Steinhardt. Messen des Sprachverständnisses bei massivem Multitask. arXiv-Preprint arXiv:2009.03300, 2020.

Y. Huang, Y. Bai, Z. Zhu, J. Zhang, J. Zhang, T. Su, J. Liu, C. Lv, Y. Zhang, J. Lei, et al. C-Eval: Eine mehrstufige, multidisziplinäre chinesische Evaluierungssuite für Grundlagenmodelle. arXiv-Vorabdruck arXiv:2305.08322, 2023.

N. Jain, K. Han, A. Gu, W. Li, F. Yan, T. Zhang, S. Wang, A. Solar-Lezama, K. Sen und I. Stoica. Livecodebench: Ganzheitliche und kontaminationsfreie Auswertung großer Sprachmodelle für Code. CoRR, abs/2403.07974, 2024. URL https://doi.org/10.48550/arXiv.2403.07974.

S. Krishna, K. Krishna, A. Mohananey, S. Schwarcz, A. Stambler, S. Upadhyay und M. Faruqui. Fakt, Abruf und Grund: Eine einheitliche Bewertung der durch Abruf erweiterten Generierung. CoRR, abs/2409.12941, 2024. doi: 10.48550/ARXIV.2409.12941. URL https://doi.org/10.485 50/arXiv.2409.12941.

A. Kumar, V. Zhuang, R. Agarwal, Y. Su, JD Co-Reyes, A. Singh, K. Baumli, S. Iqbal, C. Bishop, R. Roelofs, et al. Training von Sprachmodellen zur Selbstkorrektur durch bestärkendes Lernen. arXiv-Vorabdruck arXiv:2409.12917, 2024.

H. Li, Y. Zhang, F. Koto, Y. Yang, H. Zhao, Y. Gong, N. Duan und T. Baldwin. CMMLU: Messung des umfassenden Multitasking-Sprachverständnisses auf Chinesisch. arXiv-Vorabdruck arXiv:2306.09212, 2023.

T. Li, W.-L. Chiang, E. Frick, L. Dunlap, T. Wu, B. Zhu, JE Gonzalez und I. Stoica. Von Crowdsourcing-Daten zu hochwertigen Benchmarks: Arena-Hard- und Benchbuilder-Pipeline. arXiv-Preprint arXiv:2406.11939, 2024.

H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever und K. Cobbe. Lassen Sie uns das Schritt für Schritt überprüfen. arXiv-Vorabdruck arXiv:2305.20050, 2023.

VON Lin. ZeroEval: Ein einheitliches Framework zur Bewertung von Sprachmodellen, Juli 2024. URL https://github.com/WildEval/ZeroEval.

MAA. Amerikanische Einladungsprüfung für Mathematik – aime. In Amerikanische Einladungsprüfung für Mathematik – AIME 2024, Februar 2024. URL https://maa.org/math -competitions/american-invitational-mathematics-examination-aime.

OpenAI. Hallo GPT-4o, 2024a. URL https://openai.com/index/hello-gpt-4o/.

OpenAI. Mit LLMS logisches Denken lernen, 2024b. URL https://openai.com/index/learnin g-to-reason-with-llms/.

OpenAI. Einführung von SimpleQA, 2024c. URL https://openai.com/index/introducing -simpleqa/.

OpenAI. Mit der Einführung von SWE-bench verifiziert veröffentlichen wir eine von Menschen validierte Teilmenge von swe-bench, die mehr als 2024 Tage umfasst. URL https://openai.com/index/introducing-swe-bench -verified/.

Qwen. Qwq: Denken Sie gründlich über die Grenzen des Unbekannten nach, 2024a. URL https:// qwenlm .github.io/blog/qwq-32b-preview/.

Qwen. Qwen2.5: Eine Gruppe von Basismodellen, 2024b. URL https://qwenlm.github.io/b log/qwen2.5.

D. Rein, BL Hou, AC Stickland, J. Petty, RY Pang, J. Dirani, J. Michael und SR Bowman. GPQA: Ein Google-sicherer Q&A-Benchmark auf Hochschulniveau. arXiv-Vorabdruck arXiv:2311.12022, 2023.

Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. Li, Y. Wu und D. Guo. Deepseekmath: Die Grenzen des mathematischen Denkens in offenen Sprachmodellen erweitern. arXiv-Vorabdruck arXiv:2402.03300, 2024.

D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, TP Lillicrap, K. Simonyan und D. Hassabis. Schach und Shogi durch Selbstspiel mit einem allgemeinen Verstärkungslernalgorithmus meistern. CoRR, abs/1712.01815, 2017a. URL http://arxiv.org/abs/1712.01815.

D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, TP Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel und D. Hassabis. Das Go-Spiel ohne menschliches Wissen meistern. Nat., 550(7676):354–359, 2017b. doi: 10.1038/NATURE24270. URL https://doi.org/10.1038/nature24270.

C. Snell, J. Lee, K. Xu und A. Kumar. Die optimale Skalierung der LLM-Testzeitberechnung kann effektiver sein als die Skalierung von Modellparametern, 2024. URL https://arxiv.org/abs/2408.033 14.

T. Trinh, Y. Wu, Q. Le, H. He und T. Luong. Lösung der olympischen Geometrie ohne menschliches Demonstrationen. Nature, 2024. doi: 10.1038/s41586-023-06747-5.

J. Uesato, N. Kushman, R. Kumar, F. Song, N. Siegel, L. Wang, A. Creswell, G. Irving und I. Higgins. Mathematische Textaufgaben mit prozess- und ergebnisbasiertem Feedback lösen. arXiv-Preprint arXiv:2211.14275, 2022.

P. Wang, L. Li, Z. Shao, R. Xu, D. Dai, Y. Li, D. Chen, Y. Wu und Z. Sui. Math-shepherd: Ein markierungsfreier Schritt-für-Schritt-Verifizierer für LLMS in mathematischem Denken. arXiv-Vorabdruck arXiv:2312.08935, 2023.

X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery und D. Zhou. Selbstkonsistenz verbessert das Denken in Gedankenketten in Sprachmodellen. arXiv-Vorabdruck arXiv:2203.11171, 2022.

Y. Wang, X. Ma, G. Zhang, Y. Ni, A. Chandra, S. Guo, W. Ren, A. Arulraj, X. He, Z. Jiang, T. Li, M. Ku, K. Wang, A. Zhuang, R. Fan, X. Yue und W. Chen. Mmlu-pro: Ein robusterer und anspruchsvollerer Benchmark für das Sprachverständnis für mehrere Aufgaben. AdRR, abs/2406.01574, 2024. URL: https://doi.org/10.48550/arXiv.2406.01574.

CS Xia, Y. Deng, S. Dunn und L. Zhang. Agentless: Entmystifizierung von llm-basierten Software- Engineering-Agenten. arXiv-Preprint, 2024.

H. Xin, ZZ Ren, J. Song, Z. Shao, W. Zhao, H. Wang, B. Liu, L. Zhang, X. Lu, Q. Du, W. Gao, Q. Zhu, D. Yang, Z. Gou, ZF Wu, F. Luo und C. Ruan. Deepseek-prover-v1.5: Nutzung von Proof-Assistant-Feedback für bestärkendes Lernen und Monte-Carlo-Baumsuche, 2024. URL https://arxiv.org/abs/2408.08152.

J. Zhou, T. Lu, S. Mishra, S. Brahma, S. Basu, Y. Luan, D. Zhou und L. Hou. Anweisungsbefolgung Auswertung für große Sprachmodelle. arXiv-Preprint arXiv:2311.07911, 2023.

Anhang

A. Beiträge und Danksagungen

Hauptbeitragende

Daniela
Dejian Yang
Haowei Zhang
Junxiao-Lied
Ruoyu Zhang
Runxin Xu
Zhu Qi
Shirong Ma
Peiyi Wang
Xiao Bi
Xiaokang Zhang
Xingkai Yu Yu
Wu ZF
Wu Zhibin
Gou Zhihong
Shao Zhuoshu Li
Ziyi Gao

Mitwirkende

Aixin Liu
Bing Qu
Hui Li
Jianzhong Guo
Jiashi Li
Jingchang Chen
Jingyang Yuan
Jinhao Tu
Junjie Qiu
Junlong Li JL
Cai Jiaqi
Ni Jian
Liang Jin
Chen Kai
Dong Kai
Hu*
Kaichao You
Kaige Gao
Kang Guan
Kexin Huang Kuai
Yu Lean
Wang Lecong
Zhang Liang Zhao
Litong Wang
Liyue Zhang Lei
Xu Leyi Xia

Mingchuan
Zhang Minghua Zhang
Minghui Tang Mingxu
Zhou Meng Li
Miaojun Wang
Mingming
Li Ning Tian Panpan
Huang Peng
Zhang
Qiancheng Wang
Qinyu Chen
Qiushi Du Ruiqi Ge*
Ruisong Zhang
Ruizhe Pan
Runji Wang
RJ Chen RL Jin

Ruyi Chen
Shanghao Lu
Shangyan Zhou
Shanhuang Chen
Shengfeng Ye
Shiyu Wang
Shuiping Yu
Shunfeng Zhou
Shuting Pan SS Li
Shuang Zhou
Shaoqing Wu
Shengfeng Ye
Tao Yun
Tian Pei
Tianyu Sun T. Wang
Wangding Zeng Wen Liu
Wenfeng Liang
Wenjun Gao
Wenqin Yu*
Wentao Zhang
WL Xiao
Wei An
Xiaodong Liu
Xiaohan Wang
Xiaokang Chen
Xiaotao Nie Xin
Cheng Xin Liu
Xin Xie
Xingchao
Liu Xinyu Yang
Xinyuan Li
Xuecheng Su
Xuheng Lin XQ
Li Xiangyue Jin
Xiaojin
Shen Xiaosha
Chen Xiaowen
Sun Xiaoxiang
Wang

YK Li
YQ Wang

YX Wei
Yang Zhang
Yanhong Xu
Yao Li
Yao Zhao
Yaofeng Sonne
Yaohui Wang
Yi Yu
Zhang Yichao
Yifan Shi
Yiliang Xiong
Ying He
Yishi Piao
Yisong Wang
Yixuan Tan
Yiyang Ma*
Yiyuan Liu
Yongqiang Guo
Yuan Ou
Yuduan Wang Yue
Gong Yuheng
Zou Yujia He
Yunfan
Xiong Yuxiang
Luo Yuxiang You
Yuxuan Liu
Yuyang Zhou
YX Zhu Yanping Huang
Yaohui Li Yi Zheng
Yuchen Zhu
Yunxian Ma
Ying Tang
Yukun Zha
Yuting Yan
ZZ Ren Zehui
Ren Zhangli
Sha Zhe
Fu Zhean Xu
Zhenda Xie
Zhengyan Zhang
Zhewen Hao
Zhicheng Ma Zhigang Yan Zhiyu Wu
Zihui Gu

Zijia Zhu
Zijun Liu*
Li Zhang
Ziwei Xie
Ziyang-Lied
Zizheng-Pfanne
Zhen Huang
Zhipeng Xu
Zhongyu Zhang
Zhen Zhang

Innerhalb jeder Rolle werden die Autoren alphabetisch nach Vornamen aufgelistet. Mit * gekennzeichnete Namen bezeichnen Personen, die unser Team verlassen haben.